Ui-Ear: On-Face Gesture Recognition Through On-Ear Vibration Sensing

Guangrong Zhao, Yiran Shen, *Senior Member, IEEE*, Feng Li, *Member, IEEE*, Lei Liu, *Member, IEEE*, Lizhen Cui, *Senior Member, IEEE*, and Hongkai Wen, *Member, IEEE*

***Abstract*—With the convenient design and prolific functionalities, wireless earbuds are fast penetrating in our daily life and taking over the place of traditional wired earphones. The sensing capabilities of wireless earbuds have attracted great interests of researchers on exploring them as a new interface for human-computer interactions. However, due to its extremely compact size, the interaction on the body of the earbuds is limited and not convenient. In this paper, we propose *Ui-Ear*, a new on-face gesture recognition system to enrich interaction maneuvers for wireless earbuds. *Ui-Ear* exploits the sensing capability of Inertial Measurement Units (IMUs) to extend the interaction to the skin of the face near ears. The accelerometer and gyroscope in IMUs perceive dynamic vibration signals induced by on-face touching and moving, which brings rich maneuverability. Since IMUs are provided on most of the budget and high-end wireless earbuds, we believe that *Ui-Ear* has great potential to be adopted pervasively. To demonstrate the feasibility of the system, we define seven different on-face gestures and design an end-to-end learning approach based on Convolutional Neural Networks (CNNs) for classifying different gestures. To further improve the generalization capability of the system, adversarial learning mechanism is incorporated in the offline training process to suppress the user-specific features while enhancing gesture-related features. We recruit 20 participants and collect a realworld datasets in a common office environment to evaluate the recognition accuracy. The extensive evaluations show that the average recognition accuracy of *Ui-Ear* is over 95% and 82.3% in the user-dependent and user-independent tasks, respectively. Moreover, we also show that the pre-trained model (learned from user-independent task) can be fine-tuned with only few training samples of the target user to achieve relatively high recognition accuracy (up to 95%). At last, we implement the personalization and recognition components of *Ui-Ear* on an off-the-shelf Android smartphone to evaluate its system overhead. The results demonstrate *Ui-Ear* can achieve real-time response while only brings trivial energy consumption on smartphones.**

***Index Terms*—On-face gesture recognition, adversarial learning, model personalization, vibration sensing.**

Received 25 December 2022; revised 16 September 2024; accepted 9 October 2024. Date of publication 14 October 2024; date of current version 5 February 2025. This work was supported by the Natural Science Foundation of Shandong Province, under Grant 2022HWYQ-040, Grant ZR2024ZD12, and Grant ZR2022LZH010, and in part by the National Science Foundation China under Grant 61972230, Grant 62072278, and Grant U23A20273. Recommended for acceptance by S. Mazuelas. *(Corresponding author: Yiran Shen.)*

Guangrong Zhao, Yiran Shen, Lei Liu, and Lizhen Cui are with the School of Software, Shandong University, Jinan 250100, China (e-mail: guangrong.zhao@sdu.edu.cn; yiran.shen@sdu.edu.cn; l.liu@sdu.edu.cn; clz@sdu.edu.cn).

Feng Li is with the School of Computer Science and Technology, Shandong University, Qingdao 266237, China (e-mail: fli@sdu.edu.cn).

Hongkai Wen is with the Department of Computer Science, University of Warwick, CV4 7AL Coventry, U.K. (e-mail: hongkai.wen@warwick.ac.uk).

Digital Object Identifier 10.1109/TMC.2024.3480216

I. Introduction

With the development of chip design and manufacturing, the sizes of the electronic units are being miniaturized rapidly. The wireless chips (e.g., Huawei Kirin A1 [1] and Apple H1 [2]) and different types of sensors are encapsulated to create "true" wireless headsets or earbuds [3] (no wire between the earbuds), which is one of the most successful inventions in recent years. Due to its convenient operation and rich functionality, wireless earbuds are quickly grabbing the market of traditional earphones and are becoming the first choice for most of the users in their daily use, especially for sports and fitness. As reported by Polaris Market Research, the global market size of true wireless earbuds will reach $ 14.51 billion by 2028.[1]

The competition on the earbuds market is fierce, the choice of users are normally made according to the sound quality, noise cancellation capability, etc. Besides, the design of interactive interface is an important factor to be reckoned with having a significant impact on user experience. For example, when listening to the music or watching multimedia content, it will be greatly convenient if the users are able to directly interact with their earbuds to start/pause, switch between titles, turn up/down the column without working on their master devices. Different from traditional wired earphones or bulky headphones, true wireless earbuds normally do not feature dedicated interaction hardware, e.g., line controllers, for users to interact with their master devices easily. To provide user-device interactive functionalities, most of the developers of the wireless earbuds coincidentally choose the similar solution, i.e., tapping on the body of the earbuds. Most of the wireless earbuds facilitate Inertial Measurement Units (IMUs) to detect users' tapping once or twice, while Airpods Pro [4] utilizes force sensor which provides higher detection accuracy. However, the tiny surface of the earbuds places a huge limit on the richness of the skill sets of the interactions between users and earbuds.

Utilizing the face area near the ears is an effective way to significantly enlarge the interactive surface so that more on-face gestures can be included in the user-earbuds interaction skill sets. For example, EarBuddy [5] facilitates the microphones on most of the wireless earbuds to record the acoustic signals generated by finger sliding on face in different patterns to distinguish different on-face gestures. However, compared to IMU signals, the higher sampling rate (11.025 kHz v.s.0.2 kHz) may results

[1][Online]. Available: https://www.prnewswire.com/news/polaris-market-research/

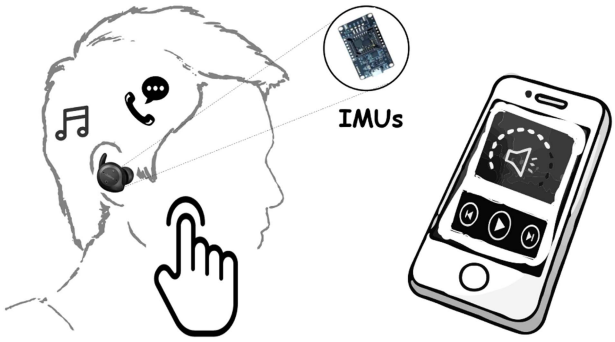


Fig. 1. A typical use case of *Ui-Ear*: the user performs on-face finger gestures to control the music app on smartphones.

in more data transmission energy consumption. Moreover, the network used for gesture prediction is not as lightweight as ours, which consumes more inference time (190 ms v.s.1.25 ms). OESense [6] utilizes the inward-facing microphone on the noise-canceling earbuds to capture the sound generated by on-face gesture and propagating through skin. The inward-facing microphone will not be affected by environment interference, however, it is normally only available on expensive noise-canceling earbuds but not the budget earbuds. At last, capacitive [7] and vision-based sensing [8] solutions are also explored, however, the dependence on the bespoke hardware restrains their application on the commercialized wireless earbuds.

In this paper, we aim to design an on-face gesture recognition system utilizing IMU sensors which are pervasively embedded on most of the wireless earbuds including both budget and high-end ones. To this end, we propose *Ui-Ear*, an on-face finger gesture recognition system by facilitating the sensing capability of IMUs on most of the mainstream wireless earbuds. As shown in Fig. 1, *Ui-Ear* extends the user-earbuds interaction to the face area to significantly enrich the maneuvers of user-device interaction. It is worth noting that, the on-face and on-device gestures are not mutually exclusive. They can work together to form the new skill sets of the user-earbuds interaction.

The contributions of this work are multi-folds:

- We propose, *Ui-Ear*, an on-face finger gesture recognition system to enrich the skill sets of user-earbuds interaction based on the sensor readings of IMUs which are available on most of wireless earbuds. To the best of our knowledge, this is the first piece of work facilitating vibration sensing for on-face gesture recognition.
- We design, **DAL-CNN**, a Domain Adversarial Learning (DAL) framework with a novel ℓ_2-uniform loss for facial gesture recognition to deal with the domain generalization issue. **DAL-CNN** can effectively improve the recognition accuracy with or without training samples from the target users.
- The results from intensive evaluations on realworld datasets show that our proposed **DAL-CNN** outperforms other machine learning approaches (including other domain adaption/generalization approaches) meanwhile the ℓ_2-uniform loss plays an important role in our approach.
- At last, we implement the inference and personalization components of *Ui-Ear* on an off-the-shelf Android smartphone. Experimental tests show that the system overhead of *Ui-Ear* on smartphones is trivial, e.g., it only takes less than 1.3 ms to perform the gesture recognition on the resource-constrained device.

II. Related Work

Functionally, our work is related to smart device interaction expansion. We separate the research on smart device interaction expansion into *device-based interaction* and *skin-based interaction* depending on the surface where the human perform their actions. The *device-based interaction* exploited the smart device itself or the solid medium around the device as the input surface. Based on acoustic signals or IMUs signals, some researchers utilized the existing internal sensors to perceive surface gestures of smartphones [9], [10], [11]. Since smaller volume and no screen, wrist-worn devices such as smartwatches and fitness bands have greater demand for interaction expansion. WatchIt [12] added thin resistive bands to wristbands to provide precise continuous over list scrolling control. WatchOut [13] extended input modalities to the watch body e.g., case, bezel, and band, etc. However, due to the small contact surface of these tiny devices, the interaction experience is still very poor. VibSense [14] used piezoelectric sensors to collect vibration signals to detect touch locations on an extended surface. However, this method required the user to find a solid surface outside the device as a signal propagation medium.

Compared with *device-based interaction*, our work is more related to *skin-based interaction*, which token the human's body surface as an interface. Some recent work [15], [16] pointed out that an adult's average body surface area is 1.73 m^2, which was greater than the touch screen of the smartphone by 400 times. When the human's finger tapped on the skin, the impact created various signals, which were propagated by the soft tissues of the skin surface, and eventually can be received by various sensors. Based on the electrical signal, SkinTrack [17] used capacitive sensors to track various gestures on the arm skin. Similarly, Earput [7] used capacitive sensing based on electrodes to provide ear-based interaction. However, these works depended on additional hardware, which cannot be directly applied to commercial devices. TapSkin [18] utilized IMUs and the microphone of a smartwatch to recognize tapping gestures on the back of the hand. However, its gesture family only contained simple tap gestures, and not included elaborate sliding sequences. EarBuddy [5] utilized the face area near the ears as an input surface. However, compared to *Ui-Ear*, EarBuddy had a higher acoustic signal sampling rate (11.025 kHz v.s. 0.2 kHz). EarBuddy may required more data transmission energy and the inference time of the spectrogram-based deep learning network was significantly longer than ours (190 ms v.s.1.25 ms). OESense [6] utilized the inward-facing microphone on the noise-canceling in-ear earbuds to capture the sound generated by on-face gesture. The inward-facing microphone may not be affected by environment interference. However, it was normally only available on expensive noise-canceling earbuds but not the budget earbuds.

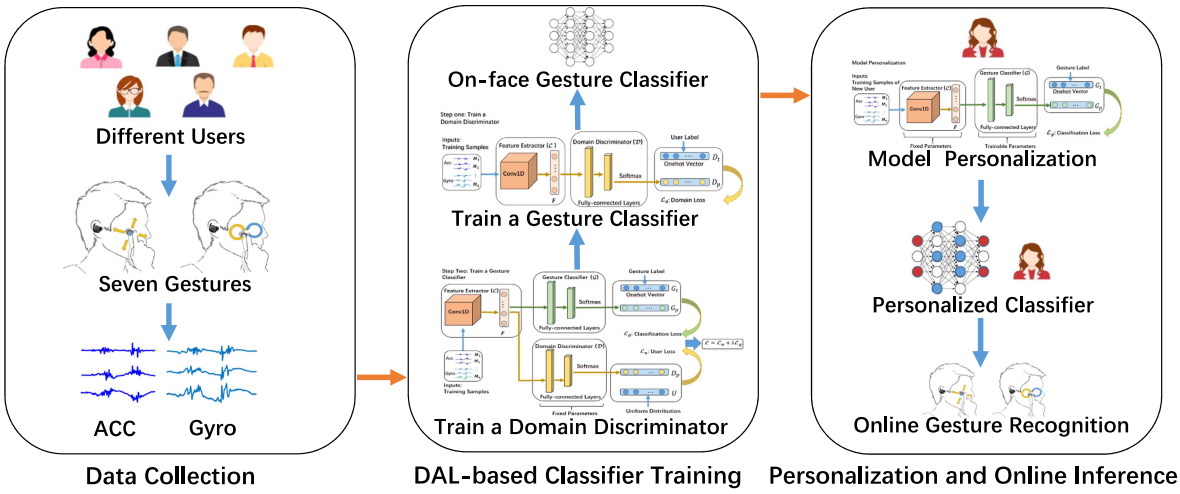


Fig. 2. The system overview of major components of *Ui-Ear*.

Verma et al. proposed a novel facial expression recognition system called ExpressEar [19]. This system utilized the inertial measurement unit (IMU) built into wearable earbuds to capture subtle facial muscle movements, thereby identifying fine facial action units (AUs). However, it did not include a specific module designed for user-independent tasks, leading to a decrease in recognition accuracy from 89.9% in user-dependent tasks to 42.1% in user-independent tasks.

Technically, our work is most related to *domain adversarial learning*. In the early years, in the field of computer vision [20], [21], [22], [23] utilized *domain adversarial learning* to solve the unsupervised domain adaptation (DA) problem. After that, Sriram et al. [24] proposed an end-to-end speech recognition framework based on a generative adversarial network (GAN). And CrossGR [25] used a target-adaptive scheme for Wi-Fi based gesture recognition. In summary, the core ideas of these studies were similar, they all mapped the feature of the target domain and the source Recently, some state-of-the-art works had improved the DA performance. Zhao et al. [26] combined convolution and recurrent neural networks to extract sleep information from radio frequency signals, and discarded information specific to measurement conditions or subjects. Similarly, EI [27] based on the four wireless signals of mmwave, wifi, ultrasonic and visible light to recognize human daily activities, which can remove the environment and subject specific information contained in the activity.

However, all of the above work based on DA method. The main difference between DA and domain generalization (DG) is whether the target domain data can be accessed during training. Consequently, the DA method in above work could not be used in real-time inference tasks. EUIGR [28] token the ongoing gesture into consideration, which integrated a complicated convolution and recurrent neural networks to fuse RFID low-level physical characters and extract space-temporal information. Ultimately, it can obtain strong robustness to subjects diversity and reduced environmental dependence. However, different from the above human activity recognition based on wireless signals, our work *Ui-Ear* exploits IMUs of wireless earbuds to perceive skin gestures. To better promote the domain generalization ability, we design a DAL framework with ℓ_2-uniform loss. Through comparative verification, our method can achieve higher accuracy on IMU-based facial gesture recognition tasks.

III. System Design

In this section, we will provide an overview of the proposed on-face finger gesture recognition system, *Ui-Ear*, for wireless earbuds based on vibration sensing. The overall system design of *Ui-Ear* is presented in Fig. 2 which consists of three major components. First, in training data collection, we recruit a certain number of volunteers to perform seven different on-face gestures to build a realworld dataset for model training and evaluation. A binary classification network is used to detect on-face gesture segments. The data matrix M is then normalized in each axis before being feed to the input of the DAL framework. Second, in DAL-based classifier training phase, the three major components of DAL, i.e., feature extractor ($\mathcal{C}$), domain discriminator ($\mathcal{D}$) and gesture classifier ($\mathcal{G}$) are trained with collected training set. The on-face gesture classifier is trained in two steps: 1) The domain (user) discriminator aims to classify the subject who performs this gesture. 2) The feature extractor tries to fool the domain discriminator meanwhile fosters the performance of the gesture classifier. At last, in personalization and online inference phase, the learned model is fine-tuned with few training samples from the target user and the personalized classifier is used for online gesture inference.

A. Data Collection

Before the DAL-based classifier training, a realworld dataset is collected. The participants are asked to perform seven different on-face gestures including tapping, slide-up, side-down, slide-left, slide-right, clockwise circle and counter-clockwise-circle as shown in Fig. 3 (In mainstream music Apps like Spotify and Amazon Music, the seven different gestures can correspond to wake up the voice assistant, volume up, volume down, previous song, next song, play, and pause, respectively). During the data

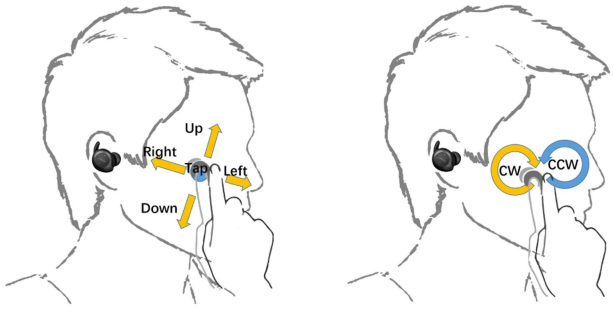


Fig. 3. The seven on-face finger gestures defined in this paper: tap, slide-up. slide down, slide-left, slide-right, clockwise circle (CW) and counter-clockwise circle (CCW).

collection, we observe, different participants may perform the same gesture in distinct ways, which brings significant variance on the data samples we collect. For example, the users may slide on their face in different speed; some users may touch on the face with their nails but some with their pad of the finger. These all introduce significant challenge on learning an on-face gesture classifier with good generalization capability.

We show some examples of the sensor readings when the users are performing different on-face gestures in Fig. 4. The first row is sensor-reading of the x-axis of accelerometer when the same subject performing seven different gestures. The waveforms of the seven different gestures are clearly distinctive so that they are distinguishable. In Fig. 5, the sensor reading of clockwise-circle (CW) from seven different users is presented. Though they are performing the same gesture, the waveforms are variant. Therefore, to obtain a robust on-face gesture recognition system, we have to deal with the inter-users variance carefully.

Signal Preprocessing: We use HI226DK [32] IMU board to collect data. The sampling rate of the IMU is set to be 200 Hz. The raw IMU data consists of six independent time-series from different axes of accelerometer and gyroscope. We first design a positive events detector to detect whether an on-face gesture happens. We apply a 200 ms sliding window with a step size of 50 ms on the raw IMU time series. Then the time-series within the 200 ms-window is feed to the positive event detector to determine if the user is operating an on-face gesture. If the detector provide a positive (i.e.,'1'), we takes a 3s-segment matrix $M = \{M_1, M_2 \dots M_6\}$ of raw IMU signals from the beginning of this 200 ms-time window for the following processing. We choose a long segment because we include CC and CCW gestures and they normally take long time to operate. However, according to the statistics of our collected dataset, if the two circling gestures are excluded, 1s-segment is sufficient to accommodate the operating time of the rest types of gestures. In order to remove the influence of gravity on acceleration data and speed up the convergence of training, the data matrix M is normalized in each axis before being feed to the input of the deep neural network.

B. DAL-Based Classifier Training

Most of the existing DAL-based methods for human activity/gesture recognition [25], [26], [27] are proposed to solve the problem of domain adaption, which requires unlabeled data from target domain (i.e., unseen domain) for DAL-based model training to reduce the distribution divergence between source domain and target domain [21]. However, in our work, we are trying to deal with a Domain Generalization (DG) problem which trains the DAL-based model without accessing any data from the new users. EUIGR [28] is one of the most related domain generalization work to *Ui-Ear* which adopts DAL-based training to realize user-independent activities recognition with RFID signals. The major difference is we design an ℓ_2-uniform loss which is able to better promote the domain generalization than EUIGR. Details about the design and training of the framework are as follows:

1) Domain Discriminator Training: The domain discriminator is used to distinguish the gestures performed by different users. An overview of the structure and training process of the domain discriminator is shown in Fig. 6. $\{M, y_D\}$ is a training sample with normalized data matrix M and its corresponding domain label y_D. As shown in Fig. 6, this step involves training the parameters of feature extractor ($\mathcal{C}$) and domain discriminator ($\mathcal{D}$). The feature extractor is composed of three layers of 1-dimensional convolutions [27], [29], [30], [31]. The kernel size of the convolution is 3 and the stride is 1. The number of output channels are 32, 64 and 128 respectively for each layer. Batch normalization operation is added after each convolutional layer to accelerate the convergence of training and Relu is used as the activation function. MaxPooling layers are also applied after each of the convolutional layer to prevent the network from over-fitting and reduce the complexity of the network. At last, a flatten layer finalized the feature extractor and outputs a 2048-dimensional feature vector F. In mathematics, feature extractor can be expressed as,

$$F = \mathcal{C}(M; \theta_{\mathcal{C}}) \tag{1}$$

where $\mathcal{C}$ is the mapping function and $\theta_{\mathcal{C}}$ is the set of parameters of the feature extractor.

Then the extracted feature vector F is forwarded to the domain discriminator ($\mathcal{D}$) which is composed of two fully connected layers. The number of nodes are 512 and K (i.e, the number of users in trainingset) respectively. The output of the final layer is an K-dimensional probabilistic vector and the domain discriminator can be expressed as,

$$D_p = \mathcal{D}(F; \theta_{\mathcal{D}}) \tag{2}$$

where D_p is predicted vector with real-valued probabilities as its elements and $\theta_{\mathcal{D}}$ is the set of network parameters of the domain discriminator. Then, then the user label is converted to a onehot encoding vector: D_t. Cross entropy is applied to estimate the loss between the prediction and the user label,

$$\mathcal{L}_d = -\frac{1}{N} \sum_{n=1}^{N} \sum_{k=1}^{K} D_t^{(k)} \log D_p^{(k)} \tag{3}$$

where N is number of training samples of a batch (i.e., batchsize). The objective of training on domain discriminator is to minimize the domain loss $\mathcal{L}_d$. Adam optimizer is used for optimization. After the training, the CNN-based feature extractor and domain

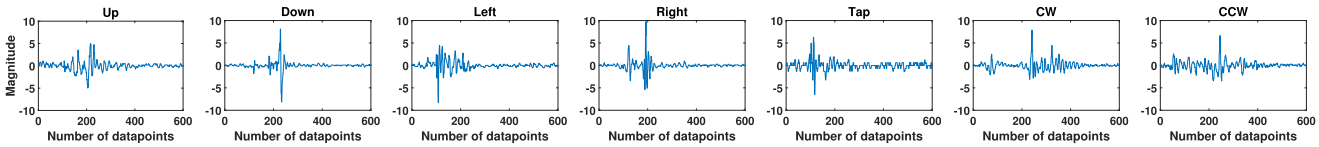


Fig. 4. Comparison of seven different gesture data of the specific user.

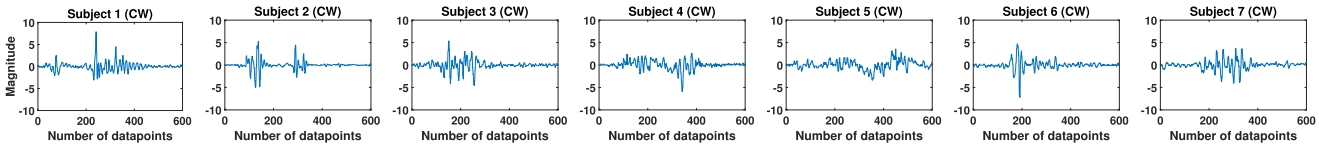


Fig. 5. Comparison of the same gesture (CW) data of seven different users.

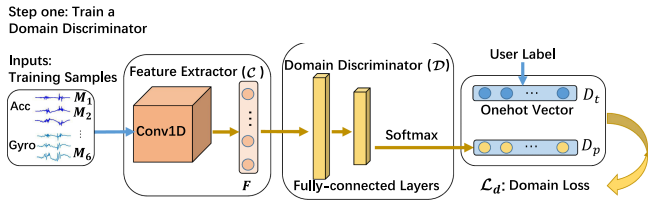


Fig. 6. Step one: train a domain discriminator: a CNN-based discriminator is trained to identify the users according to the on-face gesture.

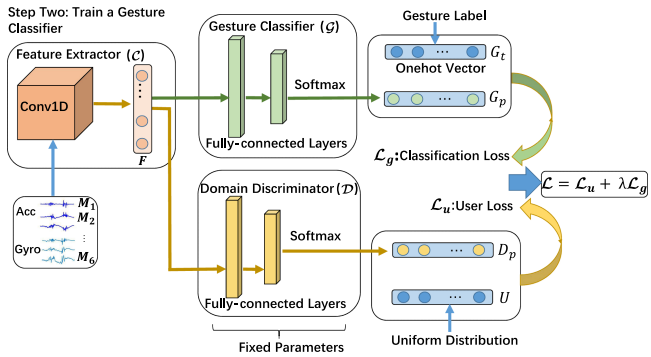


Fig. 7. Step two: train a gesture classifier: a CNN-based gesture classifier is trained with DAL framework to extract user-independent features.

discriminator can work together to distinguish different users according to users on-face gestures.

2) Gesture Classifier Training: In the second step, we will train an on-face gesture classifier based on DAL framework. The domain discriminator trained from the first step is used to suppress the user-dependent information. The workflow of DAL-based gesture classifier training is shown in Fig. 7. When training the classifier, the parameters of domain discriminator are fixed and the optimization only updates the parameters of feature extractor and gesture classifier. In this step, we aim to train a feature extractor independent of users (domains). To achieve this, the objective of the optimization in training is to maximize the accuracy of gesture classification while fooling the domain discriminator so that the users cannot be identified. If the discriminator cannot distinguish different users at all, the expected output should be a uniform distribution $U = \{\frac{1}{M}, \frac{1}{M} \ldots, \frac{1}{M}\}$.

We define an ℓ_2-*uniform* loss $\mathcal{L}_u$ to promote the output from the domain discriminator to be uniform distribution U. Specifically, it can be expressed as,

$$\mathcal{L}_u = \frac{1}{N}\sum_{n=1}^{N}\sum_{k=1}^{K}(D_p^{(k)} - U^{(k)})^2 \tag{4}$$

It is worth noting that, instead of cross-entropy commonly used in existing domain generalization work [21], [25] for activities recognition, the ℓ_2-uniform is specifically-designed for our DAL-based on-face gesture recognition framework. Intuitively, cross-entropy cannot guarantee a uniform distribution over all domains as it takes only one domain into consideration determined by the one-hot vector. Then we choose ℓ_2 over ℓ_1 because ℓ_1 promotes sparse solution and is not differentiable around 0 which contracts the requirement of uniform distribution. To further motivate our design, we also evaluate and compare the performance different choices of loss functions in Section IV-B.

As shown in Fig. 7, the output of feature extractor is also feed to a gesture classifier ($\mathcal{G}$) which consists of two fully-connected layers, the number of nodes are 512 and I (I = 7, i.e., seven different on-face gestures) respectively. Softmax is applied for the gesture classification. The predicted probability vector of the gesture classifier is,

$$G_p = \mathcal{G}(F; \theta_{\mathcal{G}}) \tag{5}$$

where $\theta_{\mathcal{G}}$ is the set of parameters of the gesture classifier. Cross entropy is used calculate the classification loss between the predicted probability vector and onehot vector converted from the gesture label, which is expressed as,

$$\mathcal{L}_g = -\frac{1}{N}\sum_{n=1}^{N}\sum_{i=1}^{I} G_t^{(i)} \log G_p^{(i)} \tag{6}$$

To minimize the user loss and classification loss simultaneously, a combined loss for the DAL-based framework is defined as,

$$\mathcal{L} = \mathcal{L}_u + \lambda \mathcal{L}_g \tag{7}$$

After training with on-face gesture samples from different users, a gesture classifier with user-independent feature extractor is obtained.

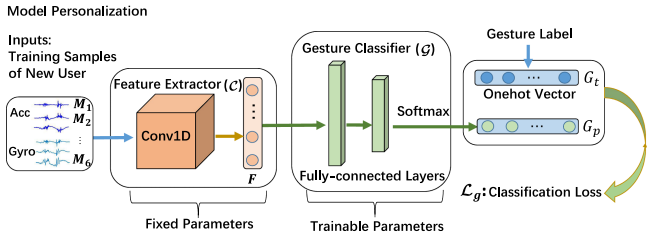


Fig. 8. Model personalization with transfer learning: the values of convolutional layers are fixed and the fully-connected layers are fine-tuned with few samples from the new user.

C. Model Personalization and Online Gesture Inference

After the DAL-based classifier training, the gesture classifier ($\mathcal{G}$) with feature extractor ($\mathcal{C}$) can be implemented on the personal device of the users for on-face finger gesture recognition. Based on the concept of transfer learning, personalizing deep neural network with few training samples from the new user can significantly improve the recognition accuracy. The CNN-based classifier normally captures low-level features with the first few convolutional layers, i.e., the feature extractor, which is significantly reusable. While the higher level (fully-connected) learns the representation of different classes with the low-level features. Therefore, as shown in Fig. 8, during personalization, the CNN-based model learned from step two is transferred to the target user then the parameters of upper layers are fine-tuned. As the coefficients of the feature extractor is reusable, their values are fixed during personalization to reduce the requirement on the amount of training data and the training efforts from users can be significantly saved.

Cross-entropy is adopted to estimate the loss between the predicted probability vector and the onehot vector converted from gesture label. The objective of personalization is to minimize the loss $\mathcal{L}_g$ in (6) with fixed feature extractor. The personalized model combines a gesture classifier fine-tuned with few training samples from current user and feature extractor trained with large number of training samples from other users. The model personalization can significantly improve the accuracy of the gesture classification of current user.

IV. Dataset Evaluation

A. Experimental Setup

Dataset collection: We recruit 20 participants (12 male and 8 female) aged between 18 and 28 years. They have different heights (160 cm–190 cm) and weights (45 kg–90 kg), and each subject needs to participate in two data collection sessions on different days within two weeks. We conduct experiments in a common office environment without controlling ambient noise[2] [33]. HI226DK [34] IMU board is used for picking up the vibration induced by on-face gestures. The IMU singals are transmitted through the serial communication protocol [35].

[2]Scenarios such as airplanes and trains can generate high noise level up to hundreds of db [32], which may have a potential impact on the IMU signal when the noise level is high.

During the data collection, to simulate how the "real" smart earbuds work, it is attached on the auricle of the ear where the wireless earbuds are normally worn. The sampling rate of IMU is set to be 200 Hz which is a common sampling rate for wearable devices. In each session, the participants sit in chair, and perform seven different on-face gestures including tapping, slide-up, slide-down, slide-left, slide-right, clockwise-circle and counter-clockwise-circle. The reading of three-axes accelerometer and three-axes gyroscope is recorded along with the gesture labels and identity labels. One participant contributes 100 samples for each gesture and therefore, 28,000 samples with labels in total are collected from the two sessions. Then, we will evaluate the accuracy of gesture recognition with *Ui-Ear* on the realworld dataset. The evaluation can be vastly categorized into three major tasks which are user-dependent classification, user-independent classification and personalization. The dataset evaluation is conducted on a high performance desktop running Ubuntu 18.04.1. The machine contains two 12-core 2.3 GHz Intel Xeon Gold CPU and 512Gb DDR4 RAM. The deep neural networks are coded and trained with Pytorch [36] on two NVIDIA GeForce RTX 3090 GPUs with 24 GB of G6X memory.

Competing Methods: In the evaluation section, we implement our approach and compare seven different machine learning models for on-face gesture recognition on three major tasks in terms of classification accuracy. Additionally, for the personalization task, we further compare the effectiveness of our transfer learning method with an unsupervised domain adaptation method. The eight machine learning models are:

- **DAL-CNN** is the convolutional neural network trained with Domain Adversarial Learning Framework which is machine learning model used in *Ui-Ear*.
- **DAL-CNN-DA** is an unsupervised domain adaptation method (i.e., tent [37]) applied to the DAL-CNN model for personalization task. This source-free domain adaptation method adjusts the model using only unlabeled target domain data through entropy minimization.
- **DANN** [21], [25] is a domain adaption method to adapt the source domain to the feature distribution of the target domain. The network parameters of discriminator and feature extractor are set to be the same as those of **DAL-CNN**.
- **EUIGR** [28] is proposed to solve the problem of activities recognition with RFID signals collected from different users and environment. It shares a similar network structure to DAL-CNN.
- **CNN** is the traditional convolutional neural network without adversarial learning. CNN is also an end-to-end approach in which the data matrix obtained from IMU is directly used as the input of the network for model learning and inference.
- **XGBoost** [38] we also implement one traditional machine learning model, XGBoost, as one of the benchmarks to demonstrate the effectiveness of deep learning. Different from the CNN-based approaches, XGBoost is not end-to-end. The training and inference are based on the 100-dimensional feature vectors extracted from IMU

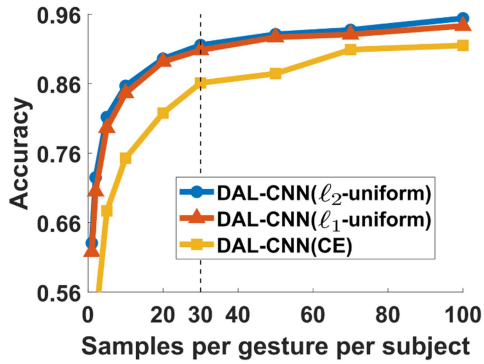


Fig. 9. The recognition accuracy against different number of training samples with different DAL losses.

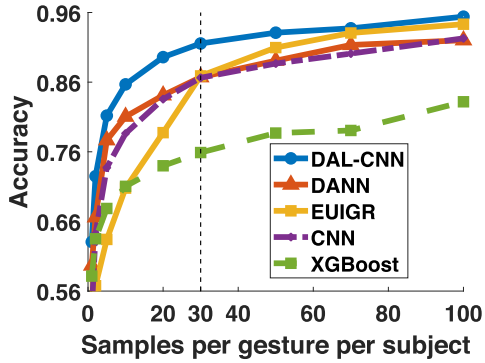


Fig. 10. The recognition accuracy against different number of training samples with different classification methods (all 20 subjects are included in the training dataset) under user-dependent scenario.

time-series. The detailed description and implementation of the feature extraction can be found in AccelWord [39].

B. Evaluation on the Design of Loss Function

We first evaluate design of the loss function of Domain Adversarial Learning (DAL), ℓ_2-uniform, which is one of the key contributions of *Ui-Ear*. The three variances of **DAL-CNN** with different losses for the DAL are considered, which are **DAL-CNN** with ℓ_2-uniform, ℓ_1-uniform and cross-entropy respectively.

During the evaluation the dataset collected from the first session is used for training and the dataset collected from the second session is used for inference. The number of training samples per gesture of each subject is gradually changed from 1 to 100, and hyperparameters (i.e., learning rate, batch size and training epochs, etc) for each method are tuned to achieve their best accuracy. We run the experiments 30 times and report the average results in Fig. 9. Overall, we can observe that with the increase of training samples, the recognition accuracy of all methods rocket before 30 then slows down after that. Moreover, by comparing **DAL-CNN** with different domain loss, our proposed ℓ_2-uniform loss outperforms the popularly used cross-entropy with significant gain on recognition accuracy. Then by comparing the different uniform losses with either ℓ_1-norm or ℓ_2-norm, the ℓ_2-uniform archives slightly better accuracy than its ℓ_1-uniform variation.

C. User-Dependent Classification

We evaluate and compare the recognition accuracy of different machine learning models on the user-dependent classification task. The user-dependent classification evaluation is a fundamental task in many gesture/activity recognition works [18], [40]. The "user-dependent" refers to that the users' identities in the training and inference stages are the same, i.e., all the 20 subjects are included. During the evaluation, the collected dataset from the first session is used as training samples and the learned machine learning models are applied on the dataset collected from the second session for gesture recognition. Generally, providing more training samples can improve the recognition accuracy of the model, but it requires collecting more data from the participants, which can be inconvenient. Therefore, we investigate how many samples are needed to train a user-dependent model with sufficient recognition accuracy. Specifically, We gradually change the number of samples per gesture of each subject from 1 to 100 and calculate the classification accuracy of different gesture recognition methods. To reduce the randomness of the results, the average accuracy from 30 independent trials is reported.

From the results in 10. we can observe, with the growth of number of training samples, the classification accuracy of gesture recognition methods increases sharply at the beginning, and then the improvement becomes insignificant after the number of training samples reach 30 for each gesture of each subject. Moreover, by comparing the accuracy across the competing methods, the end-to-end approaches achieve significantly higher accuracy than the traditional feature-based method and **DAL-CNN** is at least 5% higher than other end-to-end approaches with or without domain adaption component, **DANN**, **EUIGR** and **CNN**, when number of training samples are 30. At last, we also notice that, the accuracy **XGBoost** is far from other methods, therefore, we only consider the end-to-end approaches in the following context.

We then show the specific accuracy of each subject using the four different machine learning models in Fig. 11. The number of training samples are 100. By carefully comparing the recognition accuracy of different methods for each subject, we can observe, our proposed **DAL-CNN** achieves the highest accuracy on gesture recognition for most of the subjects and its accuracy is close to the best one for the rest two subjects (No. 8 and No. 10).

At last, we show the confusion matrices of different classification methods in Fig. 12. For each matrix, vertical labels are the actual classes and the horizontal labels are the predicted classes. The values in the grids are the percentage of classifying one gesture as one of the possible gestures and calculated from averaging the results from all 20 subjects. By comparing across different matrices, we can observe **DAL-CNN** achieves a higher accuracy rate than other machine learning methods. Furthermore, we find that for all gestures, **DAL-CNN** achieves the highest accuracy rate.

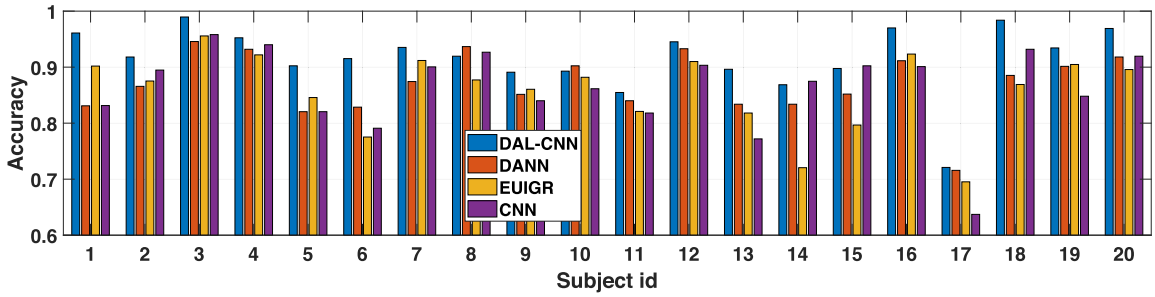
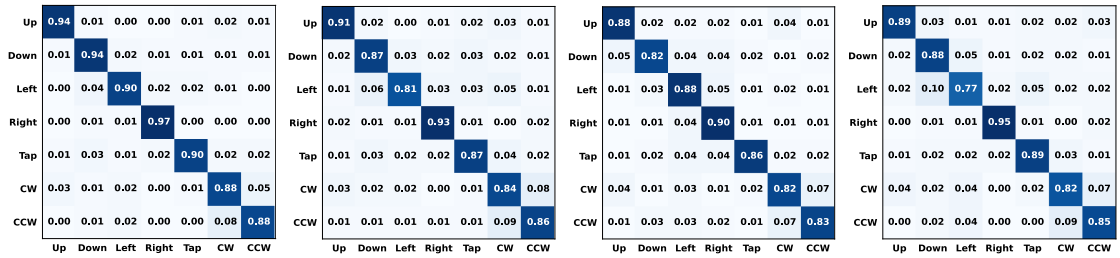


Fig. 11. The recognition accuracy of different subjects.



(a) DAL-CNN (b) DANN (c) EUIGR (d) CNN

Fig. 12. The confusion matrices of different classification methods under user-dependent scenario.

D. User-Independent Classification

In the second task, we will evaluate the accuracy of different machine learning models for user-independent classification. Compared with user-dependent classification task, user-independent models do not require new users to provide any training data before using *Ui-Ear*. It can reduce the workload of collecting data on the side of the user, thereby improving the user experience. However, it puts a higher requirement for the generalization capability of the model. During the evaluation, all competing methods learning across multiple source domains and no target domains data can be accessed during the training process. Specifically, for each round of evaluation K_1 subjects are randomly selected and all the samples of these subjects are used to train the machine learning models. Then the models are applied on all the samples of the rest subjects to show the accuracy of the user-independent classification. During the evaluation, $K_1 = \{2, 5, 10, 15, 19\}$ subjects are randomly selected to train the machine learning model and the classification accuracy on the rest of the subjects is calculated. To reduce the impact of randomness, we repeat the random selection for 20 times and the average of the classification accuracy is presented in Fig. 13. From the results, we can observe, by including more subjects in trainingset, the classification accuracy of all four methods is improved significantly as more variance of the on-face gestures is learned. Moreover, comparing the results between different methods, **DAL-CNN** always achieves the highest recognition accuracy and the improvement over the second best method is over 3% when 19 subjects are used as training.

We also show the classification accuracy of each subject in the leave-one-out cross-validation in Fig. 14 to demonstrate the stability of the gesture recognition methods. By comparing the accuracy of different methods for each subject, we can observe, **DAL-CNN** achieves the highest or close to highest classification accuracy for each subject and it is up to 8% higher than the second best method (subject No. 1).

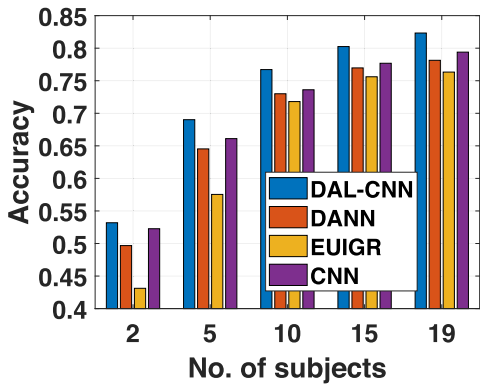


Fig. 13. The user-independent classification accuracy with respect to different number of subjects used in training.

For the user-independent scenario, we also calculate the confusion matrices of classifying the seven gestures using the three different classification methods in Fig. 15. Again, the average percentage of classifying an actual class to a predicted class over all subjects is presented in each corresponding grid. We can find that for the user-independent scenario, **DAL-CNN** achieves the best recognition accuracy for all and it is up to 9% higher than the second best method (gesture **Left**). Therefore, for the user-independent evaluation task, we can claim that **DAL-CNN** is both the most accurate and reliable method.

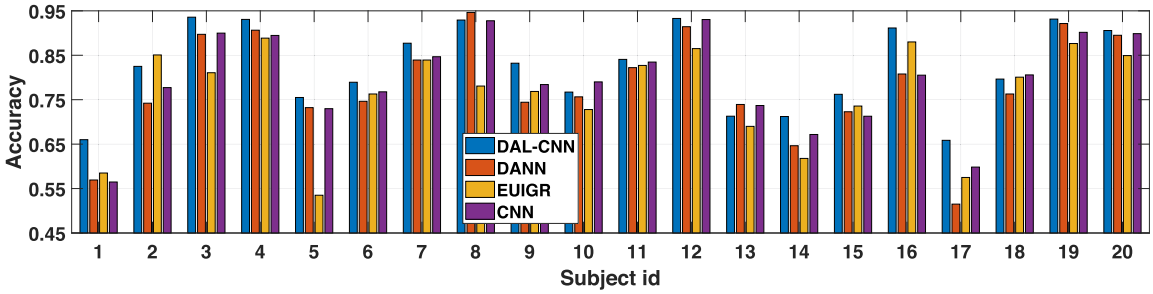
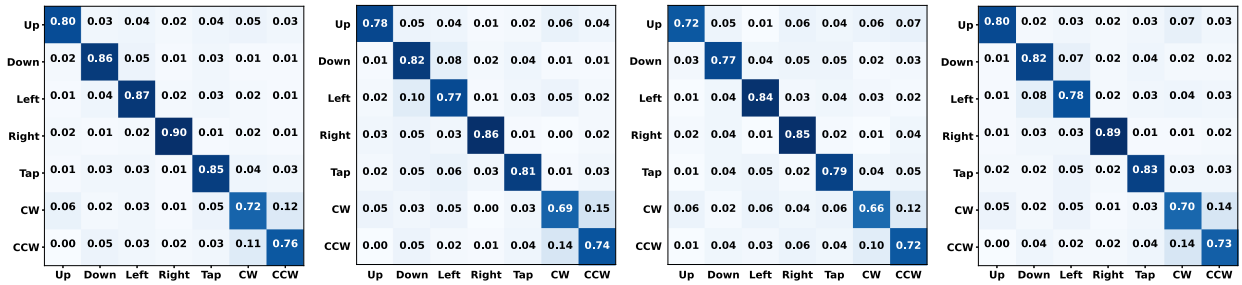


Fig. 14. The user-independent classification accuracy of each subject(leave-one-out cross-validation).



(a) DAL-CNN (b) DANN (c) EUIGR (d) CNN

Fig. 15. The confusion matrices of three different methods under user-independent scenario.

E. Personalization

Personalized learning is a compromise between user-dependent classification and user-independent classification, considering the workload required by each user to provide training samples. Normally, few number of training samples from a new user can enhance the performance of user-independent models [18], [41] while the workload introduced is acceptable. During the evaluation, we compare the performance of different personalization approaches. Leave-one-out cross-validation is applied for evaluation. Specifically, the model trained from 19 subjects is used as pre-trained model, and the rest one subject provides new training/adaptation samples for personalization. For transfer learning based methods, such as **DAL-CNN**, **DANN**, **EUIGR** and **CNN**, the personalization fixes the parameters of the feature extractor obtained from pre-trained model and fine-tuning the parameters of the fully-connected layers. For the unsupervised domain adaption method **DAL-CNN-DA**, the model adapts by using unlabeled data to minimize prediction entropy.

During personalization, we gradually increase the number of training/adaptation samples per gesture from the target users from 1 to 100 and calculate the gesture recognition accuracy. The training/adaptation samples are randomly selected and the average of the recognition accuracy over 30 independent trials is presented in Fig. 16.

Overall, with the growth of the number of training samples, the accuracy of all methods, except for **DAL-CNN-DA**, increase immediately. When the number of training samples per gesture is over 10, the recognition accuracy tends to stabilize. This

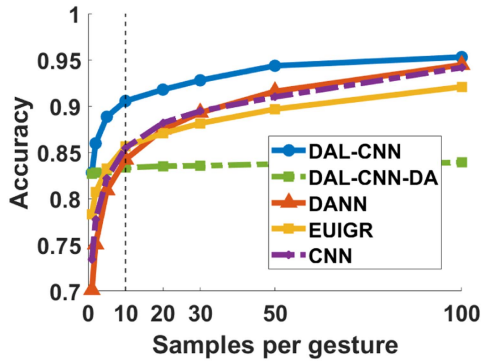


Fig. 16. The recognition accuracy against different number of samples per gesture for personalization.

indicates the transfer learning based approach does not require massive training samples to produce reasonable accuracy. In addition, we can find that as the number of adaptation samples increases, the improvement in recognition accuracy for the unsupervised domain adaptation method **DAL-CNN-DA** is minimal. When the number of adaptation samples per gesture is 10, its accuracy is 83.3%, which is only slightly higher than the user-independent model's recognition accuracy of 82.3%. Therefore, although the unsupervised domain adaptation method does not require labeled data to fine-tune the model, considering the recognition accuracy, we choose transfer learning to achieve personalization.

Moreover, among the transfer learning based methods, **DAL-CNN** achieves the highest classification accuracy and the gap

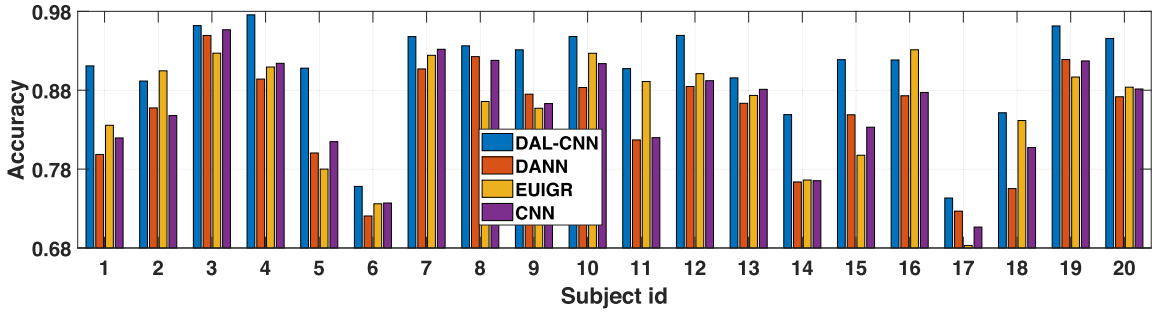
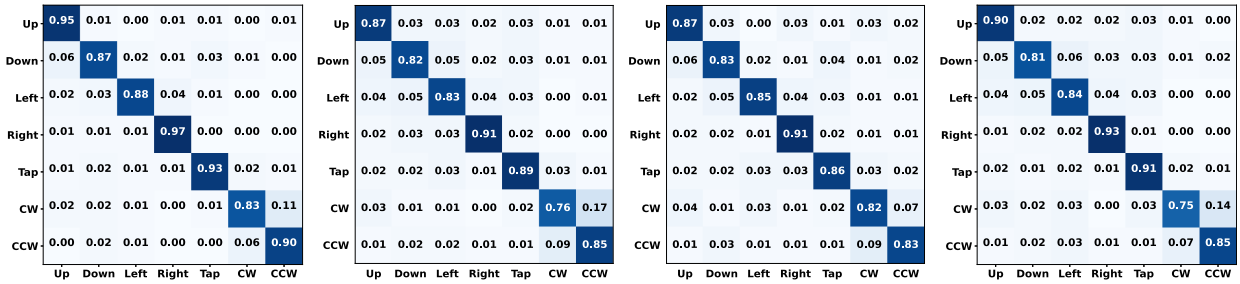


Fig. 17. The recognition accuracy of personalization for different subjects (10 samples per gesture of each subject are provided for fine-tuning).



(a) DAL-CNN (b) DANN (c) EUIGR (d) CNN

Fig. 18. The confusion matrices of different personalization approaches.

is significant when the number of training samples is low. For example, when only one sample per gesture is used for personalization, the accuracy of **DAL-CNN** has reached over 82% and it is about 10% higher than **DANN** and **CNN**, and about 5% than **EUIGR**. Then, when the number of training samples per gesture are 10, the recognition accuracy of **DAL-CNN** achieves over 91% while that of the second best method is about 86%. By comparing the accuracy of the methods with or without DAL (**DAL-CNN** v.s. **CNN**), we can observe, that the DAL framework can significantly improve the recognition accuracy.

Then we show the per-subject accuracy after personalization in Fig. 17. The number of training samples per gesture for personalization is 10. The accuracy reported are obtained from averaging the results from 30 independent trials. First of all, the average accuracy of the four methods is 90.55% (**DAL-CNN**), 84.22% (**DANN**), 85.66% (**EUIGR**) and 85.48% (**CNN**) respectively. The DAL framework brings around 5% improvement on recognition accuracy for both CNN-based classifiers when 10 samples per gesture is used for personalization. Then by comparing the accuracy for each subject, we find, **DAL-CNN** shows good reliability in gesture recognition; it achieves the highest accuracy for all 18 subjects or close to the highest for the rest 2 subjects.

Finally, we show the confusion matrices of different personalization approaches in Fig. 18. The results are obtained from personalizing the machine learning models with 10 newly-acquired training samples per gesture. From the results, we can find that for most of the gestures, the DAL module is most effective to improve the accuracy: **DAL-CNN** achieves the highest recognition accuracy for all the classes and the improvement is over 5% for most cases.

F. Long-Term Evaluation

In this section, we present a long-term evaluation to investigate the temporal stability of our system over an extended period. Approximately two and a half years after the initial data collection involving subjects 1-20, we conduct new experiments with five additional participants, designated as subjects 21-25 (3 males and 2 females), aged between 23 and 28 years. The new experiments are carried out using the same experimental setup described in Section IV-A, including the same collection device and a similar common office environment.

The 20 leave one out user-independent models (i.e., $K_1 = 19$, refer to Section IV-D), initially trained on data from two and a half years ago, are used for direct user-independent testing or serve as pre-trained models for personalization to be applied to the newly collected data from the 5 participants. We use the **DAL-CNN** method to evaluate the system's recognition accuracy over time. Fig. 19 presents the average recognition accuracies for user-independent classification (UIC) and personalized classification (PERS), with the number of training samples per gesture set to 10 for personalization.

As illustrated in Fig. 19, the average user-independent classification accuracy for subjects 21-25 is 80.59%, which shows a minimal decrease compared to the average test accuracy of 82.3% achieved two and a half years earlier with the original 20 subjects. After applying personalization, the recognition

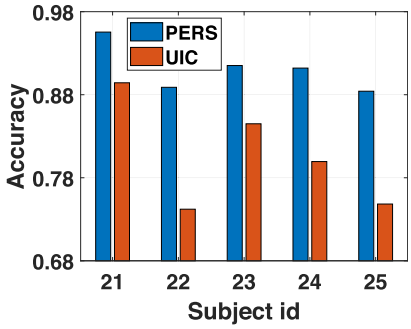


Fig. 19. The recognition accuracy of user-independent classification (UIC) and personalization (PERS) tasks for newly collected data from subjects numbered 21 to 25.

TABLE I
THE TOP-1 ACCURACY OF USER AUTHENTICATION WITH DIFFERENT GESTURES

Gesture	Tap	Up	Down	Left	Right	CW	CCW	Average
Accuracy(%)	89.34	81.98	81.75	74.5	90.5	89.29	89.51	85.27

accuracy significantly improves to 91.12%, which is roughly the same as the average test accuracy of 90.55% recorded for the original 20 subjects two and a half years earlier. This indicates that our system can maintain its stability and effectiveness over a longer period.

G. User Identification

When training the domain discriminator, we find the IMU signals generated by on-face gestures from different users are distinguishable (see Fig. 4 in Section III-A). Basing on the inter-subject variance of the gestures, user authentication layer can be added as the security protection for the wireless earbuds. Therefore, at last, we investigate the accuracy of user identification with on-face gestures. The gesture samples collected from one session is used for training and those from the other session are for inference. The network structure is the same as the domain discriminator with feature extractor in Section III-B. In Table I, the top-1 accuracy of user identification with different gestures are presented. From the results, we can observe, the accuracy of user identification with some of the gestures (Tap, Slide-right, CW and CCW) is close or even over 90%.

V. SYSTEM IMPLEMENTATION & RESOURCE CONSUMPTION

In this section, we implement the personalization and inference phases of *Ui-Ear* on a off-the-shelf smartphone to evaluate the system overhead introduced by the gesture recognition.

A. Positive Events Detection

As the IMU sensor picks up all types of physical activities of users, human movements like walking, head motion, and pose transition will produce significant interference to the trivial vibration caused by finger touch. This interference can lead to inaccurate gesture recognition results. Given this limitation, our system currently cannot work effectively when the user is in motion. Therefore, we recommend performing gesture recognition in still states, such as sitting, standing, and lying down. For these common daily scenarios, we need an on-face gesture detection layer to filter out irrelevant human motions. Before an on-face gesture is found, a 200 ms sliding window with 50 ms step size is applied on the IMU time-series. For each 200 ms time-series, a positive event detector is used to determine if this belongs to the on-face gestures in a still state, including sitting, standing, and lying down. The detector consists of two fully-connected layers with softmax as the classification function and the number of nodes for each fully-connected layers are 64 and 2. We collect a number of different daily activities as the negative events to train the detector along with the on-face gestures. The activities include sitting, standing, lying down, sit-to-stand and vice versa, head moving and walking. According to our evaluation, the lightweight detector achieves an overall accuracy of 97.21%. Then the false positive rate of the detection is 2.3% and false negative rate of the detection is 3.25%.

B. System Overhead

We implement *Ui-Ear* on an Android smartphone and evaluate the system overhead of key components running on user's personal devices including the positive event detection, gesture reference and personalization. Considering the different application scenario, the personalization module either runs in-situ on a smartphone or is offloaded to a PC in local area network according to availability of local PC. Remote cloud is not a choice in our paper for the privacy concern. The smartphone we use for implementation is Xiaomi Mi 11 Ultra with Snapdragon 888 platform including one 2.84 GHz Cortex X1, three 2.4 GHz Cortex A78 and four 1.8 GHz Cortex A55 processors. The running memory of the model we use is 12 GB and its battery capacity is 73.8 KJ (5000 mah). The local PC is running Window 10 operating system with Intel Core i7-10700F processor and a Nvidia RTX-2060 GPU. The WiFi chip of the PC is AX200 160 MHz.

1) Model Personalization: We implement model personalization on both smartphone and local PC. Deeplearning4j [42] and Pytorch [43] frameworks are adopted for fine-tuning on smartphone and local PC respectively. We first calculate the number of coefficients of the CNN-based model. The number of coefficients can be easily obtained from the Deeplearning4j interface. The total number of parameters of the gesture recognition model is 1.08 million, and trainable params for model personalization are 1.05 million. Compared with the networks used for traditional image classification tasks such as VGG, Resnet and Densenet [5], [44], the number of parameters is tens to hundreds of millions, *Ui-Ear* is more lightweight in terms of size. Moreover, for implementation on smartphone, we estimate the running time and energy consumption of personalization with different number of training samples per gesture. The energy consumption is calculated as multiplication of *Current* and *Voltage* and running time T, i.e., $Energy = Current \times Voltage \times T$. The current and voltage can be obtained by Android APIs. The resource consumption of *Ui-Ear* is presented in Table II. From the results we can observe, with the growth of number of

TABLE II
THE RESOURCE CONSUMPTION OF PERSONALIZATION WITH DIFFERENT NUMBERS OF SAMPLES PER GESTURE (IMPLEMENTATION ON SMARTPHONE)

Samples Per Gesture	1	2	5	10	20	30	50	100
Running Time (s)	6.02	6.93	9.39	13.87	22.2	31.70	49.80	94.90
Energy Consumption (J)	11.4	15.1	20.0	29.9	60.1	84.5	228.7	478.6

TABLE III
THE RESOURCE CONSUMPTION OF SMARTPHONE WHEN OFFLOADING PERSONALIZATION TO LOCAL PC

Samples Per Gesture	1	2	5	10	20	30	50	100
Time Delay (s)	1.58	1.61	1.64	1.90	2.09	2.17	2.95	4.47
Communication Cost (J)	0.97	1.1	1.5	2.0	2.7	3.3	4.5	7.7

training samples for fine-tuning, the resource consumption of personalization implemented on smartphones increases rapidly. Though the time and energy consumption is non-trivial, the personalization is an once-off operation and will not be repeated frequently. For example, when personalizating the model with 10 samples per gesture, the running time is 13.87 s and the energy consumption is 29.9J. The energy consumption only accounts for 0.04% of the battery capacity.

For implementation on local PC, the energy budget of the PC can be regarded as unlimited as it is powered by external electric supply. However, offloading and downloading the deep neural networks and training samples introduces time-delay and communication cost on the smartphone. The time-delay includes the time spent during uploading, personalization and downloading. The communication cost is the energy consumed during uploading the model and training samples to local PC and downloading the personalized model on smartphone. We use the FTP protocol [45] to transfer data between smartphone and local PC through WiFi Network. At first, We estimate the size of the data that needs to be transferred. We obtained the model size of *Ui-Ear* from the Pytorch API, we find that the model size is only 4.14 Mb. Then, we also calculated that the size of a single training sample is 19Kb. After that, we estimate the resource consumption of the smartphone in the offloading approach and show the results in Table III. By comparing the resources consumption in Tables II and III, we can observe, the offload approach is able to save the resource consumption of the smartphone significantly. For example, when the number of training samples per gesture is 10, the smartphone-only approach consumes 15.0 and 7.3 times more energy and running time than offloading the personalization to a local PC.

2) Gesture Recognition: At last, we implement the whole system of *Ui-Ear* on smartphone and evaluate the resource consumption of its key components including the positive event detection, on-face gesture inference and personalization on smartphone/PC. Different from model personalization, the positive event detection and on-face gesture inference can be executed frequently whenever the gesture recognition is triggered. Therefore, they are the key impact factor on the system efficiency and user experience when *Ui-Ear* is used in practical scenarios. From the results shown in Table IV, we can observe, each online gesture recognition, including event detection and gesture inference, only takes less than 1.26 ms to execute and consumes less than 2.37 mJ energy. Considering the targeted daily usage of the smartphone is 10–12 hours, the energy consumption of each online gesture recognition with *Ui-Ear* only accounts for 0.0019% to 0.0023% of the energy budget per minute. Therefore, the online components of Ui-Ear only introduces trivial system overhead and can run in-situ on smartphones with unnoticeable time delay.

TABLE IV
SYSTEM OVERHEAD OF *UI-EAR*

Components	Positive Event Detection	On-face Gesture Inference	Personalization on smartphone/offload
Computation Time	0.067ms	1.1871ms	13.87s/398ms
Energy Consumption	0.079mJ	2.29mJ	29.9J/2.0J

Additionally, our system needs to maintain IMU data transmission from the earbuds to the smartphone via Bluetooth. We use the HLK-B10 BLE module [46] to evaluate the impact of IMU data transmission on the battery life of the earbuds. Given that the IMU data generation rate (0.0384 Mbps) is much lower than the Bluetooth transmission bandwidth (> 1 Mbps), the Bluetooth module only needs to remain in transmission mode for a minimal amount of time during a fixed connection period. Moreover, thanks to the characteristics of BLE Bluetooth, energy can be conserved by adjusting the connection interval. For instance, with a 50 ms connection interval, we calculate that the average power consumption for data transmission increases by only 0.27 mW compared to maintaining a continuous standby state. Taking AirPods as an example, which has a 93 mW-hour battery per AirPod [47], and targeting a continuous usage of 5 hours on a single charge, our system accounts for no more than 1.45% of the total energy budget. Therefore, the IMU data transmission imposes only minimal additional overhead on the earbuds and does not significantly impact their battery life.

VI. CONCLUSION

In this article, we propose, *Ui-Ear*, an on-face gesture recognit ion system to enrich the maneuverability of the interaction between human and wireless earbuds. The *Ui-Ear* facilities vibration sensing capability enabled by the IMU sensors embedded in both budget and high-end wireless earbuds to distinguish the on-face gestures. To improve the robustness and accuracy of gesture recognition, we propose a DAL-based framework in training the feature extractor of the gesture classifier. The extensive evaluation on the realworld dataset shows that the feature extractor trained from DAL-based framework can significantly improve the accuracy of the on-face gesture classifier in the user-dependent, user-independent and personalization tasks. At last, *Ui-Ear* is implemented on an off-the-shelf smartphone and its resource-consumption is evaluated. The processing delay and energy consumption shows that *Ui-Ear* can run in-situ on smartphones with trivial system overhead.

REFERENCES

[1] Huawei, "Kirin a1," 2020. [Online]. Available: https://www.huawei-central.com/kirin-a1-the-worlds-first-bluetooth-5-1-and-bluetooth-low-energy-5-1-wearable-chip/

[2] Apple, "Apple H1," 2019. [Online]. Available: https://9to5mac.com/2019/03/20/new-apple-airpods-now-available-h1-chip-wireless-charging-case-hands-free-hey-siri/
[3] E. Nesbo, "Wireless vs. true wireless headphones: What's the difference?," 2022. [Online]. Available: https://www.makeuseof.com/wireless-vs-true-wireless-headphones-whats-the-difference/
[4] "Airpods Pro," 2019. [Online]. Available: https://www.apple.com/uk/airpods-pro/
[5] X. Xu et al., "Earbuddy: Enabling on-face interaction via wireless earbuds," in *Proc. 2020 CHI Conf. Hum. Factors Comput. Syst.*, New York, NY, USA, 2020, pp. 1–14.
[6] D. Ma, A. Ferlini, and C. Mascolo, "Oesense: Employing occlusion effect for in-ear human sensing," in *Proc. 19th Annu. Int. Conf. Mobile Syst. Appl. Serv.*, New York, NY, USA, 2021, pp. 175–187.
[7] R. Lissermann, J. Huber, A. Hadjakos, S. Nanayakkara, and M. Mühlhäuser, "Earput: Augmenting ear-worn devices for ear-based interaction," in *Proc. 26th Australian Comput.-Hum. Interaction Conf. Des. Futures: Future Des.*, New York, NY, USA, 2014, pp. 300–307.
[8] E. Tamaki, T. Miyak, and J. Rekimoto, "Brainyhand: A wearable computing device without HMD and it's interaction techniques," in *Proc. Int. Conf. Adv. Vis. Interfaces*, New York, NY, USA, 2010, pp. 387–388.
[9] K. Sun, T. Zhao, W. Wang, and L. Xie, "Vskin: Sensing touch gestures on surfaces of mobile devices using acoustic signals," in *Proc. 24th Annu. Int. Conf. Mobile Comput. Netw.*, New York, NY, USA, 2018, pp. 591–605.
[10] S. S. A. Shimon, S. Morrison-Smith, N. John, G. Fahimi, and J. Ruiz, "Exploring user-defined back-of-device gestures for mobile devices," in *Proc. 17th Int. Conf. Hum.-Comput. Interaction Mobile Devices Serv.*, New York, NY, USA, 2015, pp. 227–232.
[11] L. Wang et al., "Watching your phone's back: Gesture recognition by sensing acoustical structure-borne propagation," in *Proc. ACM Interactive Mobile Wearable Ubiquitous Technol.*, New York, NY, USA, 2021, pp. 1–26.
[12] N. Shalev, I. Keidar, Y. Moatti, and Y. Weinsberg, "Watchit: Who watches your it guy?," in *Proc. 8th ACM CCS Int. Workshop Manag. Insider Secur. Threats*, New York, NY, USA, 2016, pp. 93–96.
[13] G. Presti et al., "Watchout: Obstacle sonification for people with visual impairment or blindness," in *Proc. 21st Int. ACM SIGACCESS Conf. Comput. Accessibility*, New York, NY, USA, 2019, pp. 402–413.
[14] J. Liu, Y. Chen, M. Gruteser, and Y. Wang, "Vibsense: Sensing touches on ubiquitous surfaces through vibration," in *Proc. 2017 14th Annu. IEEE Int. Conf. Sens. Commun. Netw.*, 2017, pp. 1–9.
[15] S. Singh and A. Kumar, "Review of skinput technology: Input through skin," in *Proc. 2018 Int. Conf. Sustain. Energy Electron. Comput. Syst.*, Greater Noida, India, 2018, pp. 1–5.
[16] C. Harrison, D. Tan, and D. Morris, "Skinput: Appropriating the skin as an interactive canvas," *J. Commun. ACM*, vol. 54, no. 8, pp. 111–118, 2011.
[17] Y. Zhang, J. Zhou, G. Laput, and C. Harrison, "Skintrack: Using the body as an electrical waveguide for continuous finger tracking on the skin," in *Proc. 2016 CHI Conf. Hum. Factors Comput. Syst.*, New York, NY, USA, 2016, pp. 1491–1503.
[18] C. Zhang et al., "Tapskin: Recognizing on-skin input for smartwatches," in *Proc. 2016 ACM Int. Conf. Interactive Surfaces Spaces*, New York, NY, USA, 2016, pp. 13–22.
[19] D. Verma, S. Bhalla, D. Sahnan, J. Shukla, and A. Parnami, "ExpressEar: Sensing fine-grained facial expressions with earables," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 5, no. 3, pp. 1–28, Sep. 2021.
[20] M. L. Wouter and M. Kouw, "An introduction to domain adaptation and transfer learning," 2018, *arXiv: 1812.11806*.
[21] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Proc. 32nd Int. Conf. Int. Conf. Mach. Learn.*, 2015, pp. 1180–1189.
[22] Y. Ganin et al., "Domain-adversarial training of neural networks," *J. J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 2096–2030, 2016.
[23] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proc. 2017 IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2962–2971.
[24] A. Sriram, H. Jun, Y. Gaur, and S. Satheesh, "Robust speech recognition using generative adversarial networks," in *Proc. 2018 IEEE Int. Conf. Acoust. Speech Signal Process.*, 2018, pp. 5639–5643.
[25] X. Li et al., "CrossGR: Accurate and low-cost cross-target gesture recognition using Wi-Fi," *Proc. ACM Interactive Mobile Wearable Ubiquitous Technol.*, vol. 5, no. 1, 2021, Art. no. 23.
[26] M. Zhao, S. Yue, D. Katabi, T. S. Jaakkola, and M. T. Bianchi, "Learning sleep stages from radio signals: A conditional adversarial architecture," in *Proc. 34th Int. Conf. Mach. Learn.*, 2017, pp. 4100–4109.
[27] W. Jiang et al., "Towards environment independent device free human activity recognition," in *Proc. 24th Annu. Int. Conf. Mobile Comput. Netw.*, New York, NY, USA, 2018, pp. 289–304.
[28] Y. Yu, D. Wang, R. Zhao, and Q. Zhang, "RFID based real-time recognition of ongoing gesture with adversarial learning," in *Proc. 17th Conf. Embedded Networked Sensor Syst.*, New York, NY, USA, 2019, pp. 298–310.
[29] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *J. Commun. ACM*, vol. 60, no. 6, pp. 84–90, 2017.
[30] W. Dai, C. Dai, S. Qu, J. Li, and S. Das, "Very deep convolutional neural networks for raw waveforms," in *Proc. 2017 IEEE Int. Conf. Acoust. Speech Signal Process.*, 2017, pp. 424–425.
[31] S. Lee and M. Kim, "Waveform-based end-to-end deep convolutional neural network with multi-scale sliding windows for weakly labeled sound event detection," in *Proc. 2020 Int. Conf. Artif. Intell. Inf. Commun.*, 2020, pp. 182–186.
[32] "Decibel meter pro, what is 55 decibels of sound," 2022. [Online]. Available:https://decibelpro.app/blog/what-is-55-decibels/
[33] "Office noise 2019," (n.d.). [Online]. Available: https://www:youtube:com/watch?v=D7ZZp8XuUTE
[34] "Hipnuc,hi226dk," 2022. [Online]. Available: https://www.hipnuc.com/index.html
[35] Wikipedia, "Serial communication," (n.d.). [Online]. Available: https://en.wikipedia.org/wiki/Serial_communication
[36] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 8024–8035.
[37] D. Wang, E. Shelhamer, S. Liu, B. Olshausen, and T. Darrell, "Tent: Fully test-time adaptation by entropy minimization," in *Proc. Int. Conf. Learn. Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=uXl3bZLkr3c
[38] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, New York, NY, USA, 2016, pp. 785–794.
[39] L. Zhang, P. H. Pathak, M. Wu, Y. Zhao, and P. Mohapatra, "Accelword: Energy efficient hotword detection through accelerometer," in *Proc. 13th Annu. Int. Conf. Mobile Syst. Appl. Serv.*, New York, NY, USA, 2015, pp. 301–315.
[40] J. Ward, P. Lukowicz, G. Troster, and T. Starner, "Activity recognition of assembly tasks using body-worn microphones and accelerometers," *J. IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 10, pp. 1553–1567, Oct. 2006.
[41] C. Zhang, A. Guo, D. Zhang, C. Southern, R. Arriaga, and G. Abowd, "Beyondtouch: Extending the input language with built-in sensors on commodity smartphones," in *Proc. 20th Int. Conf. Intell. User Interfaces*, New York, NY, USA, 2015, pp. 67–77.
[42] "Deeplearning4j," 2020. [Online]. Available: https://deeplearning4j.konduit.ai/android/setup
[43] "PyTorch mobile," 2019. [Online]. Available: https://pytorch.org/mobile/android/
[44] M. Leong, D. Prasad, Y. T. Lee, and F. Lin, "Semi-CNN architecture for effective spatio-temporal learning in action recognition," *J. Appl. Sci.*, vol. 10, 2020, Art. no. 557.
[45] Mehvish, "Guiding tech," 2022. [Online]. Available: https://www.guidingtech.com/use-ftp-server-file-transfer-android/
[46] "Hlk-b10 Bluetooth module," 2024. [Online]. Available: https://www.aliexpress.com/item/1005006624407095.html?src=google
[47] "Airpods," 2024. [Online]. Available: https://en.wikipedia.org/wiki/AirPods#::text=The%20charging%20case%20provides%2024,mAh%20at%203.81%20V%20battery

Guangrong Zhao received the BS degree from the Wuhan University of Science and Technology, China, in 2018, and the MS degree from Harbin Engineering University, China, in 2021. He is the author and co-author of several top papers on wireless sensor networks and computer vision, such as *IEEE Transactions on Mobile Computing*, *IEEE Transactions on Visualization and Computer Graphics*, Neurips, CVPR. His research interests include wireless sensor networks, computer vision, and mobile computing, etc.

Yiran Shen (Senior Member, IEEE) received the BE degree in communication engineering from Shandong University, China, and the PhD degree in computer science and engineering from the University of New South Wales. He is professor with the School of Software, Shandong University. He published regularly at top-tier conferences and journals. Generally speaking, his research interests include merging area of Internet-of-Things (IoTs) and artificial intelligence.

Feng Li (Member, IEEE) received the BS degree in computer science from Shandong Normal University, China, in 2007, the MS degree Shandong University, China, in 2010, and the PhD degree in computer science from Nanyang Technological University, Singapore, in 2015. From 2014 to 2015, he worked as a research fellow with the National University of Singapore, Singapore. He then joined School of Computer Science and Technology, Shandong University, China, where he is currently a professor. His research interests include distributed algorithms and systems, wireless networking, mobile computing, and Internet of Things.

Lei Liu (Member, IEEE) received the master's and PhD degrees in 2005 and 2010, respectively. He is a full professor with the School of Software, Shandong University. He has published more than 70 research papers on international conferences and journals. His research interest includes network performance engineering, 5G technology, quality of service, IoT, and UAVs.

Lizhen Cui (Senior Member, IEEE) received the bachelor's, MSc, and PhD degrees from Shandong University, in 1999, 2002, 2005, respectively. He is a professor with the School of Software and Joint SDU-NTU Centre for Artificial Intelligence Research (C-FAIR) of Shandong University, and also a visiting professor with Nanyang Technological University Singapore. He published more than 100 papers in journals and refereed conference proceedings. His research interests include Big Data management and analysis.

Hongkai Wen (Member, IEEE) received the DPhil degree from the University of Oxford. He is an associate professor with the Department of Computer Science, University of Warwick. He became a postdoctoral researcher in a joint project between Oxford Computer Science and Robotics Institute. Broadly speaking. His research interest includes cyber-physical systems, which use networked smart devices to sense and interactive with the physical world.