A Federated Learning Based Privacy-Preserving Smart Healthcare System

Jiachun Li, *Student Member, IEEE*, Yan Meng, *Student Member, IEEE*, Lichuan Ma, *Member, IEEE*, Suguo Du, Haojin Zhu, *Senior Member, IEEE*, Qingqi Pei, *Senior Member, IEEE*, and Xuemin Shen, *Fellow, IEEE*

***Abstract*—The rapid development of the smart healthcare system makes the early-stage detection of dementia disease more user-friendly and affordable. However, the main concern is the potential serious privacy leakage of the system. In this article, we take Alzheimer's disease (AD) as an example and design a convenient and privacy-preserving system named ADDETECTOR with the assistance of Internet of Things (IoT) devices and security mechanisms. Particularly, to achieve effective AD detection, ADDETECTOR only collects user's audio by IoT devices widely deployed in the smart home environment and utilizes novel topic-based linguistic features to improve the detection accuracy. For the privacy breach existing in data, feature, and model levels, ADDETECTOR achieves privacy-preserving by employing a unique three-layer (i.e., user, client, cloud, etc.) architecture. Moreover, ADDETECTOR exploits *federated learning (FL) based scheme* to ensure the user owns the integrity of raw data and secure the confidentiality of the classification model and implement *differential privacy (DP) mechanism* to enhance the privacy level of the feature. Furthermore, to secure the model aggregation process between clients and cloud in FL-based scheme, a novel *asynchronous privacy-preserving aggregation framework* is designed. We evaluate ADDETECTOR on 1010 AD detection trials from 99 health and AD users. The experimental results show that ADDETECTOR achieves high accuracy of 81.9% and low time overhead of 0.7 s when implementing all privacy-preserving mechanisms (i.e., FL, DP, and cryptography-based aggregation).**

***Index Terms*—Alzheimer's disease (AD) detection, federated learning (FL), Internet of Things (IoT) healthcare, privacy-preserving.**

I. Introduction

IMPROVING the quality of healthcare with a reasonable cost is regarded as a global challenge, which is further exacerbated by the rapid increase of the senior people around the world. According to the latest report from World Health Organization (WHO), there were 703 million persons aged 65 years or over in the world in 2019, which is expected to double to 1.5 billion in 2050. An increase of aging population implies the increase of chronic diseases [e.g., Alzheimer's disease (AD), Parkinson, cardiovascular disease, etc.] that require frequent visits to healthcare providers as well as the increased medical treatment costs. Therefore, leveraging Internet of Things (IoT) techniques to achieve a "smart" healthcare system represents a promising solution.

The rapid development of the IoT is making smart healthcare at home feasible. By deploying smart sensors (e.g., health monitor sensors [1], smart speakers, etc.) in user's home environment, many disease detection schemes including AD detection [2], [3] which collect users' health-related data and process in the data analysis cloud can be deployed. As a remote detection, the smart healthcare system can greatly benefit the senior patients who live in rural communities, have difficulty in moving, or are disabled. The smart healthcare system is expected to play a key role in the era of the pandemic since it can provide remote medical services to senior patients and prevent them from the potential infectious diseases including COVID-19 during visiting the hospitals in person [4].

However, smart healthcare still faces some crucial problems that hinder its further deployment in the real world. We take remote AD detection as a typical example in this study. To make the remote AD detection work in the smart home, existing schemes [5] face several challenges. First, they are not friendly to users since most of them require expensive and specialized medical IoT devices or sensors [6]–[8], which are expensive and difficult to be large-scale deployed [5]. Further, since the health-related data are collected in the user's home and outsourced to the

Manuscript received February 21, 2021; revised June 16, 2021; accepted July 5, 2021. Date of publication July 20, 2021; date of current version December 6, 2021. This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFE0126000, in part by the National Natural Science Foundation of China under Grant 61972453 and Grant 61902292, in part by the Key Research and Development Programs of Shaanxi under Grant 2021ZDLGY06-03, and in part by the Fundamental Research Funds for the Central Universities under Grant XJS201502. Paper no. TII-21-0773. *(Corresponding author: Haojin Zhu.)*

Jiachun Li, Yan Meng, and Haojin Zhu are with the Shanghai Key Laboratory of Scalable Computing and Systems, Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: jiachunli@sjtu.edu.cn; yan_meng@sjtu.edu.cn; zhu-hj@sjtu.edu.cn).

Lichuan Ma and Qingqi Pei are with the State Key Laboratory of Integrated Services Networks and Shaanxi Key Laboratory of Blockchain and Secure Computing, Xidian University, Xi'an 710071, China (e-mail: malichuan@126.com; qqpei@mail.xidian.edu.cn).

Suguo Du is with the Antai College of Economics and Management, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: sgdu@sjtu.edu.cn).

Xuemin Shen is with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: sshen@uwaterloo.ca).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TII.2021.3098010.

Digital Object Identifier 10.1109/TII.2021.3098010

health organization for analysis, the user's privacy [8], [9] is facing the serious privacy leakage challenge [10], [11]. Therefore, it is urgent to propose a *cost-effective* and *privacy-preserving* IoT healthcare-based AD detection scheme.

To overcome the above-mentioned challenges, we propose a privacy-preserving AD detection scheme named ADDetector based on IoT healthcare. First, to achieve the practical AD detection in the smart home, instead of deploying expensive sensors, ADDetector determines the AD only according to the user's voice samples, which can be easily collected by the low-cost smart speaker (e.g., Amazon Echo, Google Home, etc.) in the smart home. Second, ADDetector achieves high accuracy which significantly outperforms state-of-the-art IoT-based AD detection schemes by exploiting novel features from both acoustic data and its corresponding linguistic semantics. More specially, we propose a topic-based method to automatically extract features from linguistic semantics. Lastly, to prevent users' health-related information from being leaked, the detection procedure is deployed with federated learning (FL) based scheme, and all data streaming are securely transferred via differential privacy (DP) and cryptographic solution.

More specifically, the proposed ADDetector achieves AD detection with the privacy-preserving property through its unique FL based three-level (i.e., user home, detection client, data analysis cloud, etc.) architecture. 1) In the user layer, to increase the detection speed, the user extracts the linguistic feature and acoustic feature distributedly with the assist of pretrained model. 2) To avoid user data leakage when transmitting from the user to the client, we implement *DP* on the user feature data to enhance the security level against attackers in the detection client layer. 3) In the cloud layer, to defend against the stealing of user data when transmitting from the client to the cloud, instead of transmitting the original user feature data (linguistic feature and acoustic feature) directly, we implement the method of *FL*. Furthermore, to enhance the security level of the FL framework, we implement the *asynchronous privacy-preserving aggregation module* to protect the weighted average of parameters when updating in FL.

We implement ADDetector on the dataset refined from the ADReSS Challenge dataset from INTERSPEECH 2020 [12] including voice samples of health and AD users. Compared with the state-of-the-art nonprivacy-preserving AD detection schemes (e.g., 78.7% of active-data-representation AD detection system [2]), ADDetector achieves the accuracy of 81.88% on early-stage AD detection when employing all the above FL scheme (i.e., choosing three clients), DP (i.e., setting $\epsilon = 2$), and cryptographic solutions. The deployment of privacy-preserving mechanisms causes ADDetector's accuracy reducing from 89.5 to 81.88%, and the time overhead increases from 0.701 to 0.712 s per user, which are acceptable in IoT healthcare scenarios. The contributions of this article are summarized as follows.

1) We present ADDetector, a privacy-preserving IoT-based AD detection system, which can diagnose early-stage AD by analyzing the audio captured in IoT environment; ADDetector shows advantages of easy-deployment, high-efficiency, and privacy-preserving.

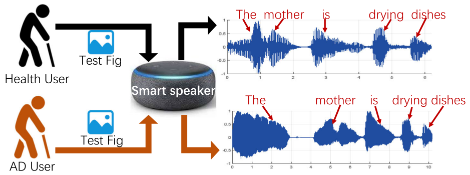


Fig. 1. Illustration of the voice-based AD detection.

2) We devise FL and DP schemes to prevent privacy leakage during the data transmission process. We also propose an asynchronous privacy-preserving aggregation module to secure the module updating within the FL scheme. Furthermore, a topic-based linguistic feature is proposed to enhance the accuracy of AD detection.
3) We evaluate ADDetector in a dataset consisting of 1010 trials from 99 users. The experimental results show that ADDetector achieves high accuracy of 81.9% and low average time overhead of 0.7 s when deploying all privacy-preserving schemes, which demonstrates its effectiveness and efficiency.

The rest of this article is organized as follows. The preliminaries of this article are introduced in Section II. In Section III, we present the overview of ADDetector. In Section IV, we elaborate the detailed design of ADDetector, which is followed by evaluation and discussion in Sections V and VI. Finally, Section VII concludes this article.

II. Related Work

A. IoT Healthcare

IoT healthcare is to collect the patient health data by IoT devices like sensors, sending to the cloud for monitoring, analysis, and remote configuration. People are more likely to accept disease monitoring and treatment at home using IoT devices [13]. For example, people can use wearable devices to continuously take care of the health data (e.g., blood sugar, blood pressure, etc.) and their action activities. The researchers [14] also put efforts on managing the collected user health data. Besides, when considering some serious diseases like dementia disease, comparing with the approaches of screening measures and neuropsychological assessments, IoT healthcare has great advantages over these methods [15] since it does not require any sophisticated medical device and can conduct treatment remotely.

B. Voice-Based AD Detection

The widespread of IoT healthcare brings the opportunity to conduct AD detection relying on user's commercial IoT voice devices rather than sophisticated medical instruments. As shown in Fig. 1, in a typical voice-based AD detection scenario, the user is asked to describe a generated test figure and the voice is recorded by a smart speaker (e.g., Amazon Echo). Such device-free schemes could be further divided into two aspects: acoustic-based and linguistic-based schemes. The former one regards the different patterns among voice samples from health

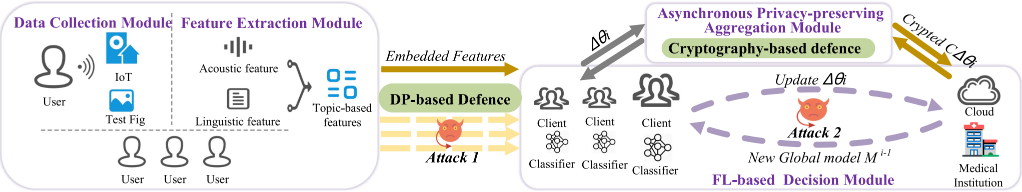


Fig. 2. System architecture of ADDETECTOR.

and AD users as detection features. For instance, it is observed from Fig. 1 that even for the same sentences (i.e., "the mother is drying dishes"), the voice from AD user has a longer time length and contains an obvious pause (i.e., *pause* between "the" and "mother"). The latter is based on the medical phenomenon that the AD users cannot fluently describe the strange objections (e.g., health user speaks "the mother is drying dishes" while the AD user speaks an ambiguous sentence "she must dropped one") and leverages on the linguistic semantics of audio transcripts to achieve AD detection. However, currently, the acoustic-based schemes suffer from low accuracy (e.g., the best accuracies are 62.0% among acoustic-based schemes [16] and 75.6% among linguistic-based schemes [17]).

C. Security Threats Faced by IoT Healthcare

The IoT healthcare system is still of great security threat. First, the vulnerability of smart home gets exposed more often, such as data leakage, security vulnerabilities in interfaces, etc. For example, the VMask Attack [18] can attack the voice interfaces, which is to forge the voice command by adversarial examples to control the smart IoT device. Second, the health data of users has a probability to be stolen and tampered with when transmitting by the attacker. The attacker can implement the man-in-the-middle attack and side-channel Attack to threaten the security of the systems. For instance, the WindTalker [19] can attack and steal the sensitive data by leveraging the channel state information extracted from the Wi-Fi signal. In our study, we mainly consider privacy-related issues while leaving other security issues for future work.

III. OVERVIEW OF SYSTEM

A. Design Goal

Compared with the existing AD detection schemes based on IoT healthcare introduced in Section II, ADDETECTOR is designed to achieve the following goals.

1) Easy-to-be-Deployed: When conducting AD detection, the user is only required to have a voice interaction with commercial voice assistance (e.g., Amazon Echo), and no further analysis or sophisticated device deployment should be conducted by the user.

2) High-Efficiency: ADDETECTOR should achieve the accuracy better than existing acoustic or linguistic-based AD detection schemes. Besides, to deploy ADDETECTOR with the increasing number of users, ADDETECTOR also needs to achieve low time overhead.

3) Privacy-Preserving: The health-related privacy data including user's raw data and features sent to classification cannot be disclosed during the data flow transmission of AD detection.

B. Threat Model

In this article, we consider attackers who are capable of getting access to the communication channel between the user's IoT devices and the data analysis cloud by launching the man-in-the-middle attack. Thus, as shown in Fig. 2, any information that is transmitted by the components of ADDETECTOR such as the processed feature has the noneligible possibility to be obtained by the attacker. Therefore, both a privacy-preserving detection framework (i.e., FL in this article) and data-oriental privacy protection methods (e.g., encryption) should be employed by ADDETECTOR. Note that, in this article, we do not assume the attacker has the ability to inject the user's network and extract the original raw data. Because directly attacking the user's IoT device is a quite strong assumption, it is out of the scope of this study.

C. Overview of System

As illustrated in Fig. 2, to achieve the above-mentioned design goals and address security challenges, we design ADDETECTOR with an architecture of three-layers (i.e., the user layer, the detection client layer, and the cloud layer). ADDETECTOR is composed of four components. In the user layer, the *Data Collection Module* instructs the user to provide voice samples for AD detection, and the *Feature Extraction Module* extracts features from both acoustic and linguistic aspects. Then, the *FL-Based Decision Module* exploits the FL framework in which the features from users are first assigned to multiple detection clients, and then the AD classification is optimized by the interaction between detection clients and the cloud. The FL framework allows the raw data to be preserved at the user level, and DP is exploited to secure the transmission between the user layer and the detection client layer. Finally, the *Asynchronous Privacy-Preserving Aggregation Module* is implemented to ensure the integrity and confidentiality of the interaction between the detection clients layer and the cloud layer.

IV. SYSTEM DESIGN

A. Data Collection Module

In this subsection, we introduce how to collect voice samples from users for AD detection. To collect audio data in IoT healthcare environment, ADDETECTOR requires the user to describe

TABLE I
STATISTICAL ANALYSIS OF TOPIC GENERATION DATASET

Indicators	Health control (HC)	AD
Age	66.35	66.75
Average word amount	12.11	11.67
Pause word	6.25	10.13
Word error	0.30	1.14

a given picture (i.e., cookie theft picture [20]) and utilizes the voice assistance (e.g., Amazon Echo, Google Home, etc.) to collect the audio simultaneously. Then, the collected audio a_u is converted to text sentence s_u, and the generated multimodulated information is transferred to the next module.

1) Topic Generation Dataset: Besides collecting raw data from users, it is important to analyze some labeled data to enable the following feature selection and AD decision. In this article, the topic generation dataset is a subset of the ADReSS Challenge dataset from INTERSPEECH 2020 [12], which contain $N_S = 600$ sentences $S_T = (s_1, s_2, \dots, s_{N_S})$ from $N_U = 30$ users. Furthermore, we extract the $N_W = 7134$ words from S_T and represent them as $W_T = (w_1, w_2, \dots, w_{N_W})$. Note that, in the real-world scenario, the topic generation dataset (S_T, W_T) precollected by the medical institution is for the research/medical purposes. Finally, ADDETECTOR sends the topic generation datasets S_T and W_T to the *Feature Extraction Module* in the user layer.

B. Feature Extraction Module

This module is deployed in each user's smart home. After collecting user's audio a_u and its corresponding textual sentence s_u, ADDETECTOR generates both acoustic feature f_a and linguistic feature f_l by leveraging precollected topic generation dataset S_T and W_T. Then, the generated f_a and f_l are embedded into the final feature f for this medical detection trial.

1) Acoustic Feature Extraction: Given an audio a_u collected from a user, to extract useful features from the acoustic aspect, it is important to understand the difference existing in the audio between health and AD users. Table I shows the statistical results of the topic generation dataset. It is observed that the pauses, word interruption (e.g., word error), can be utilized to characterize the likelihood of AD. Therefore, we extract acoustic feature f_a from a_u in the following ways.

First, to evaluate the pause and word interruption, we select the duration time dur and the chunks chu numbers in the a_u as features. To calculate the chunk numbers, we calculate the power of a_u and regard the number of continuous segments whose powers are larger than the predefined threshold as the feature. Then, to preserve the remaining details of voice samples, we also leverage the Mel Frequency Cepstrum Coefficient (MFCC) [21] as the last feature of f_a.

To calculate MFCC, first, for a given trial sound, during its transmission, the energy density in the high-frequency band decreases sharply. It is harmful to the quality of the extracted feature from the collected audio a_u [22]. Thus, to compensate for such attenuation, we implement the pre-emphasis filter on a_u. The emphasized audio x_n can be represented as follows:

$$x[n] = a_u[n] - \beta a_u[n-1] \tag{1}$$

where $\beta = 0.97$ in this study [23]. Then, for the emphasized $x[n]$, we segment it into multiple frames with the window length of 25 ms and step of 10 ms. For the ith frame $x_i[n]$, we calculate its corresponding MFCC $c_i[n]$. Note that when calculating MFCC, the number of Mel filters M in this study is set to 22, and we choose the first $k_a = 13$ dimension of $c_i[n]$ in this article. Note that N_F represents the number of frames obtained from a given sentence. Finally, by combining time duration dur and chunk numbers chu, we get the acoustic feature f_a with 15 dimension as

$$f_a = \left[\text{dur}, \text{chu}, \sum_{i=0}^{N_F} \frac{c_i(1:k_a)}{N_F}\right]. \tag{2}$$

2) Linguistic Feature Extraction: As mentioned in Section II-B, the sentence from AD user usually has less relevance to the trial image (e.g., "she is drying dishes" from health user vs. "she must dropped me" from AD user when describing "a woman is washing dishes"). Different from previous linguistic-based features [2] that analyze the whole sentence, we notice that focusing on the relationship between sentence semantics and topic from trial images can achieve better detection performance. Therefore, we propose a novel topic-based mechanism to extract the linguistic feature f_l from s_u by the following steps.

Topic Selection: We denote the topic as the most relevant words for a given trial image. Since the information inside a trial image cannot be fully characterized by a single sentence, we leverage the W_T and S_T from precollected train dataset to facilitate the topic selection. For $W_T = [w_1, w_2, \dots, w_{N_W}]$ containing N_T unique words, we calculate each frequency of each unique word to build a tuple $T_{\text{unique}} = (t_1 : f_1, t_2 : f_2, \dots, t_{N_T} : f_{N_T})$ where f_i is the frequency of ith topic word t_i and T_{unique} is sorted in descending order by frequency. Then, we denote the final topic words T as top $k = 30$ topic words from T_{unique}

$$T = [t_1, t_2, \dots, t_k]. \tag{3}$$

Topic-Based Feature Generation: After determining the topic words T, we calculate the linguistic feature by deploying the TextCNN [24] on the user sentence s_u and sentences $S_T = (s_1, s_2, \dots, s_{N_S})$ from topic generation dataset. First, for s_u and ith sentence s_i in S_T, we calculate the number of topic words they contain as d_u and d_i. Then, we define a text as follows:

$$\text{Text} = \overbrace{s_u \oplus \dots s_u}^{d_u \text{ items}} \oplus \overbrace{s_1 \oplus \dots s_1}^{d_1 \text{ items}} \oplus \dots \overbrace{s_{N_s} \oplus \dots s_{N_s}}^{d_{N_s} \text{ items}} \tag{4}$$

where $\oplus$ is the *concatenation and newline* operation and the generated *Text* has a total of $(d_u + \sum_{i=1}^{N_s} d_i)$ lines. Note that if s_u is 0, we regard this detection trial is invalid and directly discard the s_u. Then, we take the $Text$ as an input of a TextCNN model which is pretrained using Dementiabank dataset [25] and receive the clustering result with $(d_u + \sum_{i=1}^{N_s} d_i)$ features. As mentioned in (4), since the first line in $Text$ is s_u, the linguistic feature f_l can be denoted as the first $k_l = 50$ elements of the first feature from TextCNN's results.

3) Feature Combination: We combine the acoustic feature and linguistic feature as the final feature $f = [f_a, f_l]$ of the a_u from a detection trial. Then, for N_D detection trial times in ADDETECTOR, totally N_D features $F = [f_1, f_2, \ldots, f_{N_D}]$ would be extracted and transmitted to the next module.

C. FL-Based Decision Module

1) Security Challenges and Module Overview: The embedded feature f generated on the user layer could then be used to conduct the AD decision. However, directly conducting AD decision faces the following security challenges: 1) Once the classification model's parameters are leaked, the attack can pollute the model via backdoor attack [26]; 2) the feature data may contain sensitive information of users (e.g., age, gender, etc.).

To solve the above challenges, ADDETECTOR first employs FL scheme consisting of the client and the data analysis cloud level to protect the integrity of the classification model. More specifically, in the FL scheme, each client receives features from several users and only uploads model updating information to the data analysis cloud. Then, to prevent feature leakage, the *DP-based mechanism* is exploited when the feature is transmitted from the user to its corresponding client.

2) DP-Based Privacy-Preserving Mechanism: When transmitting feature f from the user level to the client level of the FL scheme, to improve the security level, ADDETECTOR preprocesses f via a lightweight DP-based scheme. The insight of leveraging DP is that adding random noise could hide the corresponding relationships inside the features and preserve their privacy information. The details are shown below [27].

We precollected dataset $D_i = \{(f_1, y_1), \ldots, (f_m, y_m)\}$ in the ith client, in which the embedded feature $f = [f_a, f_l]$. D_i' is the adjacent datasets of D_i, that is, $\|D_i - D_i'\|_1 \leq 1$. Note that D is a collection of the dataset.

Definition 1 (DP). A randomized algorithm $\mathcal{M}$ with domain D is (ε, δ) -differentially private if for all $\mathcal{S} \subseteq \text{Range}(\mathcal{M})$ and for all D_i and D_i'

$$\Pr[\mathcal{M}(D_i) \in \mathcal{S}] \leq \exp(\varepsilon) \Pr[\mathcal{M}(D_i') \in \mathcal{S}] + \delta. \tag{5}$$

Definition 2 (ℓ_1-sensitivity). For a function p which maps the numeric queries of certain database to real numbers, its ℓ_1-sensitivity $D \rightarrow \mathbb{R}^k$ is

$$\Delta p = \max_{D_i, D_i'} \|f(D_i) - f(D_i')\|_1. \tag{6}$$

When choosing the noise mechanism of DP, we consider the following two mechanisms in this study [28].

The Gaussian Mechanism: $n \sim \mathcal{N}(0, \sigma^2)$ preserves $(\epsilon, \delta) -$ DP, where $\mathcal{N}$ represents the Gaussian distribution, the amplitude of noise is defined as $\sigma \geq \alpha \Delta p / \epsilon$ in which the constant $\alpha \geq \sqrt{2 \ln(1.25/\delta)}$ for $\epsilon \in (0, 1)$. In this result, n is the value of an additive noise sample for a data in the dataset, and Δp is the sensitivity of the function p given by (6).

The Laplace Mechanism: In Laplace mechanism, the Laplace distribution with scale $b = \Delta p / \epsilon$ can be defined as follows:

$$\text{Lap}(x \mid b) = \frac{1}{2b} \exp\left(-\frac{|x|}{b}\right). \tag{7}$$

In the evaluation of ADDETECTOR, we implement the two methods on our user features and make the security analysis in Section IV-E.

3) FL-Based Decision Framework: FL is a distributional training framework to address data-silos scenarios raising nowadays. It is composed of a cloud server S (i.e., data analysis cloud in this article) which maintains a global model M and several clients $C = c_1, c_2, \ldots, c_h$. Each client manages several users and conducts the training and provide parameters of its local model to the data analysis cloud S. Then S feedbacks the global parameters to help update the client-side local model so as to make accurate AD decision. By leaving clients' data locally and only transmitting model parameters, deploying FL can prevent users' sensitive feature from being leaked.

When training in each round, the clients only provide the server with the gradient updates of their local models without transmitting user data directly. Then the cloud updates the global model from M^{t-1} to M^t, where t represents the number of rounds. Equation (8) presents the training process of FL from round $t - 1$ to round t

$$M^t = M^{t-1} + \sum_{i=1}^{N} \frac{n_i}{n} \Delta\theta_i^t,$$

$$n_i = \|\mathcal{D}_i\|, n = \sum_{i=1}^{N} n_i \tag{8}$$

where $\Delta\theta_i^t$ is the gradient updates of ith local client c_i in round t, M^{t-1} is the global model in round $t - 1$, and D_i is the dataset of local client i. We elaborate how to calculate $\Delta\theta_i^t$ in the following paragraphs.

Training Process in the Client Level: For a given client, before conducting AD decision for a user's trial, it needs to train a classification model leveraging a precollected dataset $D_i = \{(f_1, y_1), \ldots, (f_m, y_m)\}$, where $F = [f_1, f_2, \ldots, f_m]$ represents m features from all users' trials in this client, and $y = [y_1, y_2, \ldots, y_m] \in \{0, 1\}$ (0 is health and 1 is AD). Suppose there are two parameters (weights θ_1 and bias θ_2) which can split the AD and health items correctly. Specially, the objective function can be written as

$$p(y|F; \theta_1, \theta_2) = (h_{\theta_1,\theta_2}(F))^y (1 - h_{\theta_1,\theta_2}(F))^{1-y},$$

$$h_{\theta_1,\theta_2}(F) = \sigma(\theta_1^T F + \theta_2) \tag{9}$$

where $\theta_1^T F + \theta_2$ is the logits outputted by the linear classifier, followed by the sigmoid function σ. This probability is presented as $h_{\theta1,\theta2}(F)$ (in a range [0, 1]), which can classify the input F as the health group. For further analysis, when the ground truth of F is 1 ($y = 1$), the objective function $p(y|F; \theta_1, \theta_2)$ equals $h_{\theta_1,\theta_2}(F)$, and the optimization solver should maximize this value by optimizing the parameters θ_1, θ_2. It is easy to extend such analysis to the opposite case. The loss function is defined as

$$l(\theta) = \sum_{i=1}^{m} y^{(i)} \log h(F^{(i)}) + (1 - y^{(i)} \log(1 - h(F^{(i)})). \tag{10}$$

By leveraging stochastic gradient descent solver, the ith client can optimize (10) and obtain a local optimal solution $\theta_i^t = (\theta_1^*, \theta_2^*)$ in round t. So the updates of user i uploading to the cloud server S is $\Delta\theta_i^t = \theta_i^t - \theta_i^{t-1}$. In this article, we set the round time between client and server as 5 and the local iteration time as 20.

AD Decision Process: After the training process, the model on the client-side has been trained to aim at minimizing the loss function. When the features f are coming from the user side, the client can predict the result of AD or health (HC).

D. Asynchronous Privacy-Preserving Aggregation Module

According to (8), at the end of tth round, the weighted average of $\Delta\theta_i^t$ is first computed and then added to the global model. The former modules do not consider the privacy leakage existing in the model aggregation between clients and the cloud. In this subsection, inspired by [29] and [30], we design a novel module to protect $\Delta\theta_i^t$. Our scheme is designed by introducing the concept of *secret sharing*. By sharing the aggregator's secret among the clients, the weighted average of local gradient updates cannot be derived unless enough numbers of reports from clients are collected. Considering the case that a small number of clients in AD detection accidentally fail to upload the gradient updates which is not discussed in [29] and [30], a novel privacy-preserving aggregation method for local gradient updates is proposed here. The details of this module are as follows.

1) Secrets Generation: Our secret sharing scheme is based on the discrete logarithm difficulty hypothesis [29] to protect the confidentiality of $\Delta\theta_i^t$. To generate the shared secrets, a trusted third-party entity is adopted in our scheme. First, this entity determines a cyclic group $\mathbb{G}$ of prime order p and g is the generator. For simplicity, we denote the data analysis cloud S as c_0 in the following part. Then, for any two members in $\{c_0, c_1, \ldots, c_h\}$, referred as c_u and c_v, the pairwise secrets sk_{uv} and sk_{vu} for them are randomly sampled which is from $\mathbb{Z}_p$ such that $sk_{uv} + sk_{vu} = 0 \bmod p$. In this manner, the secret vector $\boldsymbol{s}_k$ for c_k is

$$\boldsymbol{s}_k = \{sk_{kv} : v = 0, \ldots, h \text{ and } v \neq k\}.$$

After deriving all the secret vectors, they are sent to the aggregator and clients, respectively.

2) Encrypting the Local Gradient Updates: During the tth round and for the kth client, the following is computed first:

$$sk_k = \sum_{sk \in \boldsymbol{s}_k} sk. \tag{11}$$

For the jth element in $\Delta\theta_k^t$, it is first multiplied with a proper constant to become an integer. In the practical implementation, all the local gradient updates are in the range $[10^{-4}, 1]$. Thus, such a constant is set as 10^4. For simplicity, we do not introduce more symbols to denote the transformed gradient updates. Then, the corresponding ciphertext $C\Delta\theta_{kj}^t$ is obtained by

$$C\Delta\theta_{kj}^t = g^{n_k \cdot \Delta\theta_{kj}^t} \cdot H(t)^{sk_k}. \tag{12}$$

Here, $H : \mathbb{Z} \rightarrow \mathbb{G}$ is a hash function modeled as a random oracle. Refer $\boldsymbol{C\Delta\theta}_k^t$ as the ciphertext vector of $\Delta\theta_k^t$. After encrypting each element in $\Delta\theta_k$, $\boldsymbol{C\Delta\theta}_k^t$ is sent to the aggregator.

3) Deriving the Aggregated Results: After collecting the local gradient updates from clients, the aggregator first computes sk_0 from $\boldsymbol{s}_0$ via (11). Then, he will derive the aggregated results. There are two cases. One case is that all the clients successfully upload the encrypted gradient updates and the other is that some clients accidentally fail to upload the updates.

Case 1: Here, the aggregated results can be directly obtained as in [30]. Specifically, for the jth element, given $C\Delta\theta_{kj}^t$ $(k = 1, \ldots, h)$, the aggregator just computes $C\Delta\theta_{0j}^t$ as

$$\begin{aligned} C\Delta\theta_{0j}^t &= H(t)^{sk_0} \cdot \prod_{k=1}^{h} C\Delta\theta_{kj}^t \\ &= g^{\sum_{k=1}^{h} n_k \cdot \Delta\theta_{kj}^t} \cdot H(t)^{\sum_{k=0}^{h} sk_k}. \end{aligned} \tag{13}$$

Note that, $\Delta\theta_{0j}^t$ represents the update model parameter between c_0 (i.e., the data analysis cloud S) and c_j. According to the manner to generate secrets and (11), when all the clients upload their updates, $\sum_{k=0}^{h} sk_k = 0$. As a result, the aggregator obtains $g^{\sum_{k=1}^{h} n_k \cdot \Delta\theta_{kj}^t}$. Since the local gradient updates are in the range $[10^{-4}, 1]$ and after transformation, this range becomes $[1, 10^4]$. Finally, $\sum_{k=1}^{h} n_k \cdot \Delta\theta_{kj}^t$ is derived via a brute-force search of the integers in the range $[1, h \cdot n_k \cdot 10^4]$. Since the dimension of the model parameter is 33, the time overhead is acceptable for the brute-force search of the integers at the cloud side.

Case 2: When some clients fail to upload their local gradient updates, let C_f denote the set of these clients and C/C_f denote the ones that succeed in uploading. To recover the aggregated results of the uploaded results, after broadcasting C_f by the aggregator, each client k in C/C_f computes $\boldsymbol{s}_k^f$ as

$$\boldsymbol{s}_k^f = \{sk_{vk} : sk_{kv} + sk_{vk} = 0 \bmod p \text{ and } v \in C_f\}.$$

Then, the kth client sets $sk_k^f = \sum_{sk \in \boldsymbol{s}_k^f} sk$. This client obtains $H(t)^{sk_k^f}$ and sends it to the aggregator. Finally, the aggregator computes sk_0^f in the same manner as the clients and computes $H(t)^{sk_0^f}$. At this time, $C\Delta\theta_{0j}^t$ becomes

$$\begin{aligned} C\Delta\theta_{0j}^t &= H(t)^{sk_0} \cdot \prod_{k \in C/C_f} C\Delta\theta_{kj}^t \cdot H(t)^{sk_k^f} \\ &= g^{\sum_{k \in C/C_f} n_k \cdot \Delta\theta_{kj}^t} \cdot H(t)^{sk_0 + sk_0^f + \sum_{k \in C/C_f} (sk_k + sk_k^f)}. \end{aligned} \tag{14}$$

It is easy to verify that $sk_0 + sk_0^f + \sum_{k \in C/C_f} (sk_k + sk_k^f)$ equals 0 mod p. Here, the aggregator obtains $g^{\sum_{k \in C/C_f} n_k \cdot \Delta\theta_{kj}^t}$.

The same as in Case 1, $\sum_{k \in C/C_f} n_k \cdot \Delta\theta_{kj}^t$ can be derived by a brute-force search in the range $[1, |C/C_f| \cdot n_k \cdot 10^4]$.

In this way, the aggregator can obtain the aggregated results without knowing the local updated gradients even when some clients accidentally fail to upload their updates.

E. Security Analysis

Finally, we give a security analysis for the mechanisms employed by ADDETECTOR as follows.

1) DP Privacy-Preserving Mechanism: To realize the protection of the user's feature data, the DP mechanism is implemented to enhance the security level when transmitting from the user to the client side. Even the attacker has some knowledge of the user raw data, and the noise-added feature can prevent the privacy breach with an appropriate choice of ϵ.

2) FL-Based Decision Module: To avoid the feature data transmitting directly in the network, we design the FL framework. Even the transmitting data get leaked, and the attacker can only access the gradient parameter transmitting in the network.

3) Asynchronous Privacy-Preserving Aggregation Module: To protect the update weighted parameters transmitting in FL, we analyze the security of the Asynchronous Privacy-Preserving Aggregation Module. We first focus on Case 1 where all the clients are enrolled in the aggregating phase. Recalling the process to generate the secrets to mask the local gradient updates, any two clients (referred to as c_u and c_v, sk_{uv}, and sk_{vu}) are known to these two clients but the other secrets remain unknown.

As a result, sk_k derived from (11) is only known to c_k. At this time, the encrypted local gradient update $C\Delta\theta_{kj}^t$ derived from (12) is computationally indistinguishable from a number randomly chosen in the cyclic group $\mathbb{G}$. Actually, according to the assumption in the threat model, S is semihonest. Hence, such kind of collusion is impossible. This guarantees the security of the encrypted local gradient updates. When S aggregating the locally encrypted gradient updates via (13), only the aggregated results can be derived as S is not able to recover the clients' secrets used for masking the local gradient gradients.

From this sense, the Asynchronous Privacy-Preserving Aggregation Module is secure under the assumption defined in the threat model. Similarly, the security for Case 2 where some clients fail to upload their local gradient updates can be directly derived.

V. EVALUATION

A. Overall Performance

1) Dataset Explanation: In this article, we utilize the ADReSS Challenge dataset from INTERSPEECH 2020 [12] as our dataset. Since all analyses are done based on this dataset and no human subject is recruited by us for participating real-world experiment, this study is exempt from institutional review board (IRB) approval of our institutions. After excluding the topic generation dataset in Section IV-A and items which cannot generate linguistic features as described in Section IV-B, our dataset contains a total of 1010 trials in which 560 and 450 trials are from 48 health users and 51 AD users, respectively.

TABLE II
TIME OVERHEAD OF AD DETECTION PER USER

Overall time	Acoustic feature	Linguistic feature	DP and FL-based scheme	Encryption scheme
711.55 ms	57.43 ms	639.60 ms	11.42 ms	3.09 ms

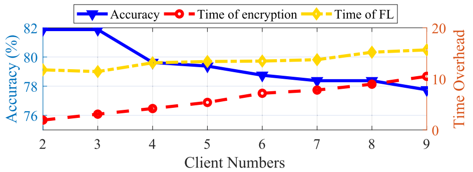


Fig. 3. Impact of FL configurations on AD detection.

2) Performance of ADDETECTOR: When conducting AD detection, we set the client number to 3 in the FL-based scheme and conduct tenfold cross-validation for each client. ADDETECTOR achieves the accuracy of 81.9% under the Laplace-based DP protection ($\epsilon = 2$) and cryptography-based scheme, which demonstrates the effectiveness and privacy-preserving property of ADDETECTOR. For the detection overhead, when implementing ADDETECTOR on a desktop with 64-b Ubuntu 18.04 OS, Intel Core i7 CPU, and 64 GB RAM, the per user AD detection time is only 711.55 ms, which is acceptable in the smart home scenario. As shown in Table II, it is observed that the feature generation costs 97.9% of total time overhead, which means our privacy-preserving schemes (i.e., DP- and FL-based schemes and cryptography-based schemes) are time efficient. Without specification, all evaluation results are conducted in the privacy-preserving framework.

B. Impact of Various Factors on ADDETECTOR

1) Configuration of Federated Learning: In Section V-A, there are three clients in the FL scheme. To evaluate the impact of FL configurations, we change the client number from 2 to 9. It is observed from Fig. 3 that when increasing client numbers, the accuracy decreases from 81.88% in two clients to 77.75% in nine clients, and the time overhead on DP- and FL-based scheme increases from 11.77 ms in two clients to 15.68 ms in nine clients. The reason is that when increasing client numbers, the limited calculation resource of the server (i.e., data analysis cloud) and the distribution of users' data becoming more sparse lead to the increase of time overhead and the decrease of accuracy, respectively. However, even when choosing nine clients, the accuracy is still acceptable and the increasing of time overhead is negligible.

2) Impact of DP-Based Mechanism: To evaluate the impact of the DP-based mechanism, we implement both the Laplace mechanism and the Gaussian mechanism under different ϵ on ADDETECTOR, respectively. As shown in Table III, when decreasing ϵ, the accuracy has a decreasing trend on both Laplace and Gaussian types because the security level of data is increased by improving the noise scale. Besides, the results demonstrate that deploying Laplace noise type in ADDETECTOR can achieve better accuracy than Gaussian noise type. Even the decreasing of ϵ will reduce the availability of data, ADDETECTOR can still

TABLE III
IMPACT OF DIFFERENTIAL PRIVACY

Parameter Setting	Type	Accuracy	Type	Accuracy
ϵ=2	Laplace	81.875%	Gaussian	81.25%
ϵ=1.5	Laplace	81.625%	Gaussian	80.625%
ϵ=1	Laplace	80.75%	Gaussian	80.125%
ϵ=0.8	Laplace	80.25%	Gaussian	79.75%

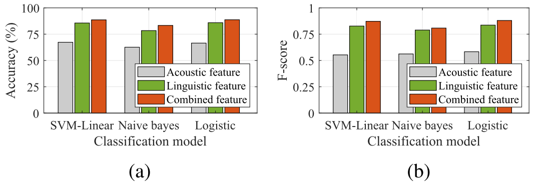


Fig. 4. Performance under different models and features. (a) Accuracy. (b) F-score.

TABLE IV
COMPARISON OF DIFFERENT METHOD

Model Type	Accuracy	F-score	Time overhead (s)
BI-GRU [10]	74.8%	0.768	3.272
HAN [3]	81.5%	0.815	3.337
HAN-AGE [10]	86.9%	0.876	3.438
ADDETECTOR	81.9%	0.860	0.712
ADDETECTOR-beta	89.5%	0.880	0.701

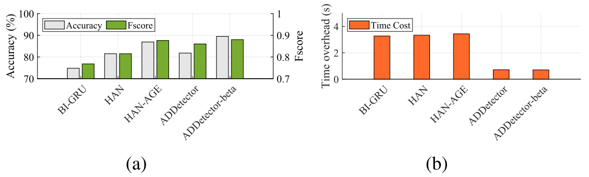


Fig. 5. Graphical comparative analysis of different methods. (a) Accuracy and F-score. (b) Time overhead.

achieve 80.25% accuracy when the ϵ is as small as 0.8. In summary, DP-based scheme can achieve good privacy protection for ADDETECTOR without significantly reducing the performance.

3) Impact of Cryptography-Based Mechanism: To evaluate the efficiency and impact of the Asynchronous Privacy-Preserving Aggregation Module, the open-source OpenSSL library is utilized to implement SHA-1 for the hash function. With respect to the cyclic group, it is constructed following RFC3562, and the computations on this group are achieved by introducing the GNU multiple precision (GMP) arithmetic library. For each client, its local gradient updates are encrypted in less than 1 ms (specifically, 0.748 ms). Thus, we focus on the aggregation phase and utilize the time cost of the aggregator to derive the aggregated result in one epoch as the metric when evaluating the efficiency of this module. The experimental results are shown in Fig. 3. When the number of clients increases from 2 to 9 with a step size of 1, the time cost to derive the aggregated result approximately linearly increases.

This is because multiplications by GMP over the group are very fast and the time cost mainly comes from obtaining the result in a brute-force manner. Since all the local gradient updates are transformed into integers before being encrypted, the search field is linearly expanded as the number of clients increases with a step size of 1. When there are nine clients, such a cost is around 10.50 ms per user. As detecting the AD is not rigorously time-sensitive and personal privacy yields more, such a delay is reasonably acceptable in our scenario.

4) Classifier and Features Selection: In Section V-A, ADDETECTOR utilizes both acoustic and linguistic features and logistic regression classifier. To evaluate the impact of other classifiers and feature combinations, we utilize acoustic and linguistic features separately and also employ support vector machine (SVM) with linear kernel and Naive Bayes classifiers. Note that for the ease of implementation, in this experiment, we evaluate our system without implementing privacy-preserving schemes (i.e., named ADDETECTOR-beta). Fig. 4 shows the accuracy and F-score on different conditions. Among three different classifiers, logistic regression achieves the best detection accuracy. When considering the feature selection, using acoustic features solely can only achieve the accuracy of about 60%. Then, using linguistic features enhances the accuracy to 85%, and using both features can achieve the accuracy of 89.5%.

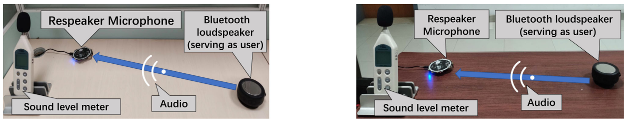
C. Comparison With Existing Works

In this subsection, we make a comparison between ADDETECTOR and existing early-stage AD detection schemes. Note that since employing privacy-preserving mechanisms (i.e., DP- and FL-based schemes and cryptography-based schemes) would decrease the AD detection accuracy, to make a fair comparison, we implement both ADDETECTOR and its nonprivacy-preserving version (i.e., ADDETECTOR-beta) in the performance comparison.

Table IV shows the accuracy and time overhead of ADDETECTOR, ADDETECTOR-beta, and existing schemes. It is observed that without implementing privacy-preserving, ADDETECTOR-beta achieves the accuracy of 89.5% and F-score of 0.88, which outperforms the best case of existing schemes (86.9% accuracy in HAN-AGE [10]). Besides, the time overhead of ADDETECTOR-beta is only 21.3% of the best case in existing schemes (i.e., 3.272 s in BI-GRU [10]). The results in Table IV and Fig. 5 demonstrate the superiority of our purposed AD detection system in the aspects of accuracy and time overhead. Then, when implementing privacy-preserving schemes, ADDETECTOR achieves the accuracy of 81.88% and latency of 0.712 s, which is also comparable with existing solutions. In summary, ADDETECTOR can achieve the privacy protection with a reasonable performance cost.

D. Real-World Evaluation of ADDETECTOR

Similar to existing works [31], in this article, we make an evaluation based on the ADReSS Challenge dataset, which is the only large-scale publicly available dataset to researchers now. To make the experimental results convincing, in this experiment, we set up an IoT healthcare environment to evaluate the performance of ADDETECTOR. Due to the ethical consideration, it is hard to obtain audio from the real AD patient. Thus, as shown in Fig. 6(a) and (b), we utilize a Bluetooth loudspeaker to serve as



(a)

(b)

Fig. 6. Real-world evaluation. (a) Lab. (b) Hall.

TABLE V
REAL-WORLD SCENARIO EVALUATION

Scenario	Original	Laboratory	Hall
Accuracy	81.875%	78.875%	80.500%

TABLE VI
IMPACT OF FEATURE DIMENSIONS

Dimensions of f_l	Accuracy	F-score	Time overhead (s)
10	77.7%	0.738	0.403
30	83.1%	0.804	0.527
50	89.5%	0.880	0.712
200	90.1%	0.889	1.330
700	91.5%	0.898	2.650

the user and play the audio samples from the ADReSS Challenge dataset. Then, a smart speaker Respeaker Core v2.0 is employed to collect audio. After obtaining audio from the user, the features are extracted in the user layer and then are delivered to the detection client layer of ADDetector for further AD detection. The noise levels in the laboratory as shown in Fig. 6(a) and the hallway as shown in Fig. 6(b) are 42.6 and 37.2 dB, respectively, measured by the Smart Sensor AR824 sound level meter.

As shown in Table V, when conducting the real-world test, the accuracy can maintain more than 78% in real-world scenarios. The results demonstrate the effectiveness and robustness of ADDetector overall under different IoT healthcare environments.

VI. Discussion

A. Feature Selection

For the feature selection, as shown in Table II, the time spent on feature generation is a major part of the AD detection. Thus, it deserves to find more effective and useful features to accelerate the detecting time and maintain high accuracy in the future.

B. Feature Dimensions

We conduct additional experiments when setting the dimension of linguistic feature f_l as 10, 30, 50, 200, and 700, respectively. Table VI shows the performance of ADDetector when f_l length changes. Note that, during the experiments, besides f_l, the parameters of ADDetector follow the same configuration as Section V-A.

It is observed from Table VI that when the dimension of the training features used increases, the detection effect is gradually improved. However, the time overhead for each user is increased too. As shown in Table VI, when the linguistic feature dimension is above 50, the performance gets slightly increasing; however, the time overhead per user increases rapidly. Thus, to achieve the tradeoff between the time overhead and the performance, we set the dimension of the linguistic feature data as 50 in this study.

C. Client Number in FL

For the client number selection issue, it is observed in Fig. 3 that when the client number is above 3, the accuracy undergoes an obvious decrease. So owing to the tradeoff of performance and time overhead, we set the client number as 3 so that each client contains 33 users, which is a normal configuration for FL. However, the overall performance under different client numbers from 2 to 9 is acceptable, which is around 80% after implementing all security mechanisms.

D. Classifier Selection

The performance of the classifier depends on the characteristic of the generated features. For instance, [31] and [32] propose different features for the same ADReSS Challenge dataset and then utilize different classifier such as probabilistic linear discriminant analysis (PLDA) and SVM to achieve the best performance in the IoT healthcare environment. Because of the differences between the features, [31] and [32] implement different classifier (i.e., PLDA and SVM) on the same AD detection dataset, respectively. In this study, we propose our novel feature based on the ADReSS Challenge dataset. Due to the extracted feature characteristics, the performance of logistic regression (LR) is superior to other classifiers (i.e., SVM and Naive Bayes). However, all the three classifiers can realize the accuracies of more than 80%, which demonstrates the effectiveness of our feature.

E. Dataset

Since the health-related user data is of high-level privacy protection, the available fully published large-scale dataset of AD detection is only the ADReSS Challenge dataset [12]. Similar to existing works [31], [32], we make an evaluation based on the ADReSS Challenge dataset. In the IoT healthcare environment, the performance on other unpublished datasets is under exploration, which will be left for future research. However, we believe ADDetector is effective and useful on the other datasets. It is because our insight is that the differences of acoustic features and linguistic features between normal and AD users are significant, which is a ubiquitous phenomenon [2], [3].

VII. Conclusion

In this article, we proposed ADDetector, a privacy-preserving smart healthcare system to realize low-cost AD detection. ADDetector utilizes the audio from smart devices as the input and employs the DP-based mechanism and FL-based framework to prevent the leakage of raw data and model details during data transmission. Furthermore, an Asynchronous Privacy-Preserving Aggregation Module was exploited to secure the model updating in the FL scheme. Experimental results demonstrated that ADDetector achieves high accuracy under a strong security protection level. For future work, we will discover more effective features to represent the characteristics of ADs and evaluate the feasibility of ADDETECTOR on a larger dataset.

REFERENCES

[1] S. M. R. Islam, D. Kwak, M. H. Kabir, M. Hossain, and K. Kwak, “The internet of things for health care: A comprehensive survey,” *IEEE Access*, vol. 3, pp. 678–708, 2015.

[2] F. Haider, S. de la Fuente, and S. Luz, “An assessment of paralinguistic acoustic features for detection of Alzheimer’s dementia in spontaneous speech,” *IEEE J. Sel. Topics Signal Process.*, vol. 14, no. 2, pp. 272–281, Feb. 2020.

[3] Y. Pan, B. Mirheidari, M. Reuber, A. Venneri, D. Blackburn, and H. Christensen, “Automatic hierarchical attention neural network for detecting AD,” in *Proc. Interspeech*, 2019, pp. 4105–4109.

[4] B. A. Jnr, “Use of telemedicine and virtual care for remote treatment in response to COVID-19 pandemic,” *J. Med. Syst.*, vol. 44, no. 7, pp. 1–9, 2020.

[5] R. Varatharajan, G. Manogaran, M. K. Priyan, and R. Sundarasekar, “Wearable sensor devices for early detection of Alzheimer disease using dynamic time warping algorithm,” *Cluster Comput.*, vol. 21, no. 1, pp. 681–690, 2018.

[6] P. Gope and T. Hwang, “BSN-Care: A secure IoT-based modern healthcare system using body sensor network,” *IEEE Sensors J.*, vol. 16, no. 5, pp. 1368–1376, Mar. 2016.

[7] B. Xu, L. D. Xu, H. Cai, C. Xie, J. Hu, and F. Bu, “Ubiquitous data accessing method in IoT-based information system for emergency medical services,” *IEEE Trans. Ind. Informat.*, vol. 10, no. 2, pp. 1578–1586, May 2014.

[8] R. Lu, X. Lin, and X. Shen, “SPOC: A secure and privacy-preserving opportunistic computing framework for mobile-healthcare emergency,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 3, pp. 614–624, Mar. 2013.

[9] X. Liang *et al.*, “Enabling pervasive healthcare through continuous remote health monitoring,” *IEEE Wireless Commun.*, vol. 19, no. 6, pp. 10–18, Dec. 2012.

[10] W. Kong, H. Jang, G. Carenini, and T. Field, “A neural model for predicting dementia from language,” in *Proc. Mach. Learn. Healthcare Conf.*, 2019, pp. 270–286.

[11] X. Lin, R. Lu, X. Shen, Y. Nemoto, and N. Kato, “SAGE: A strong privacy-preserving scheme against global eavesdropping for ehealth systems,” *IEEE J. Sel. Areas Commun.*, vol. 27, no. 4, pp. 365–378, May 2009.

[12] *ADReSS Dataset From Interspeech 2020*, 2020. [Online]. Available: http://www.homepages.ed.ac.uk/sluzfil/ADReSS/

[13] L. Fang, Y. Li, Z. Liu, C. Yin, M. Li, and Z. Cao, “A practical model based on anomaly detection for protecting medical IoT control services against external attacks,” *IEEE Trans. Ind. Informat.*, vol. 17, no. 6, pp. 4260–4269, Jun. 2021.

[14] C. Li, M. Dong, J. Li, G. Xu, X. Chen, and K. Ota, “Healthchain: Secure EMRs management and trading in distributed healthcare service system,” *IEEE Internet Things J.*, vol. 8, no. 9, pp. 7192–7202, May 2021.

[15] S. Enshaeifar *et al.*, “The Internet of Things for dementia care,” *IEEE Internet Comput.*, vol. 22, no. 1, pp. 8–17, Jan./Feb. 2018.

[16] L. Hernández-Domínguez, S. Ratté, G. Sierra-Martínez, and A. Roche-Bergua, “Computer-based evaluation of Alzheimer’s disease and mild cognitive impairment patients during a picture description task,” *Alzheimer’s Dementia: Diagnosis, Assessment Dis. Monit.*, vol. 10, pp. 260–268, 2018.

[17] B. Mirheidari, D. Blackburn, T. Walker, A. Venneri, M. Reuber, and H. Christensen, “Detecting signs of dementia using word vector representations,” in *Proc. Interspeech*, 2018, pp. 1893–1897.

[18] L. Zhang, Y. Meng, J. Yu, C. Xiang, B. Falk, and H. Zhu, “Voiceprint mimicry attack towards speaker verification system in smart home,” in *Proc. IEEE Conf. Comput. Commun.*, 2020, pp. 377–386.

[19] M. Li *et al.*, “When CSI meets public WiFi: inferring your mobile phone password via WiFi signals,” in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2016, pp. 1068–1079.

[20] *The AI That Spots Alzheimer’s from Cookie Drawing*, 2020. [Online]. Available: https://www.bbc.com/news/technology-54538228

[21] F. Zheng, G. Zhang, and Z. Song, “Comparison of different implementations of MFCC,” *J. Comput. Sci. Technol.*, vol. 16, no. 6, pp. 582–589, 2001.

[22] X. Zhao and D. Wang, “Analyzing noise robustness of MFCC and GFCC features in speaker identification,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2013, pp. 7204–7208.

[23] W. Han, C.-F. Chan, C.-S. Choy, and K.-P. Pun, “An efficient MFCC extraction method in speech recognition,” in *Proc. IEEE Int. Symp. Circuits Syst.*, 2006, pp. 145–148.

[24] C. Du, Z. Chen, F. Feng, L. Zhu, T. Gan, and L. Nie, “Explicit interaction model towards text classification,” in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 6359–6366.

[25] *The Dementiabank Dataset*. 2017. [Online]. Available: https://dementia.talkbank.org/

[26] S. Li, M. Xue, B. Zhao, H. Zhu, and X. Zhang, “Invisible backdoor attacks on deep neural networks via steganography and regularization,” *IEEE Trans. Dependable Secure Comput.*, to be published, doi: 10.1109/TDSC.2020.3021407.

[27] C. Dwork and A. Roth, “The algorithmic foundations of differential privacy,” *Found. Trends Theor. Comput. Sci.*, vol. 9, no. 3/4, pp. 211–407, 2014.

[28] K. Wei *et al.*, “Federated learning with differential privacy: Algorithms and performance analysis,” *IEEE Trans. Informat. Forensics Secur.*, vol. 15, pp. 3454–3469, 2020.

[29] E. Shi, T. H. Chan, E. G. Rieffel, R. Chow, and D. Song, “Privacy-preserving aggregation of time-series data,” in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2011, pp. 1–17.

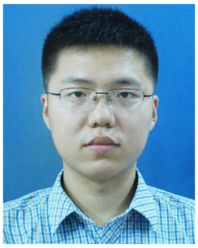
[30] S. Li, H. Zhu, Z. Gao, X. Guan, K. Xing, and X. Shen, “Location privacy preservation in collaborative spectrum sensing,” in *Proc. IEEE INFOCOM*, 2012, pp. 729–737.

[31] R. Pappagari, J. Cho, L. Moro-Velazquez, and N. Dehak, “Using state of the art speaker recognition and natural language processing technologies to detect Alzheimer’s disease and assess its severity,” in *Proc. Interspeech*, 2020, pp. 2177–2181.

[32] N. Cummins *et al.*, “A comparison of acoustic and linguistics methodologies for Alzheimer’s dementia recognition,” in *Proc. Interspeech. ISCA-Int. Speech Commun. Assoc.*, 2020, pp. 2182–2186.

Jiachun Li (Student Member, IEEE) received the B.S. degree in communication engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2020. He is currently working toward the Ph.D. degree in computer science and technology with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China.

His research interests include smart home security and smart healthcare security.



Yan Meng (Student Member, IEEE) received the B.S. degree in electronic and information engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2016. He is currently working toward the Ph.D. degree in computer science and technology with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China.

His research interests include wireless network security and Internet of Things security.

Lichuan Ma (Member, IEEE) received the B.S. degree from the School of Mathematics, Shandong University, Jinan, China, in 2012, and the Ph.D. degree from Xidian University, Xi’an, China, in 2018, both in information security.

He is currently with the School of Cyber Engineering, Xidian University, where he is a Member of the State Key Laboratory of Integrated Services Networks. His research interests include on trust management and privacy-preserving techniques for intelligent systems.

Suguo Du received the Ph.D. degree in mathematical and information science from the University of Coventry, Coventry, U.K., in 2002.

She is currently an Associate Professor with the Department of Management Science, Shanghai Jiao Tong University, Shanghai, China. Her current research interests include risk and reliability assessment, vehicular networks security and privacy protection, and social networks security management.

Haojin Zhu (Senior Member, IEEE) received the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2009.

Since 2017, he has been a Full Professor with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China. His current research interests include network security and privacy enhancing technologies.

Dr. Zhu was the recipient of the Young Scholar Award of Changjiang Scholar Program from the Ministry of Education of P. R. China in 2016.

Qingqi Pei (Senior Member, IEEE) received the B.S., M.S., and Ph.D. degrees in computer science and cryptography from Xidian University, Xi'an, China, in 1998, 2005, and 2008, respectively.

He is currently a Professor and a Member of the State Key Laboratory of Integrated Services Networks, Xidian University. His research interests include privacy preserving techniques, blockchain and edge computing security.

Dr. Pei is a Professional Member of ACM and a Senior Member of the Chinese Institute of Electronics and China Computer Federation.

Xuemin (Sherman) Shen (Fellow, IEEE) received the Ph.D. degree in electrical engineering from Rutgers University, New Brunswick, NJ, USA, in 1990.

He is currently a University Professor with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. His research interests include network resource management, wireless network security, Internet of Things, 5G and beyond, and vehicular *ad hoc* and sensor networks.

Dr. Shen was the recipient of the R.A. Fessenden Award in 2019 from IEEE, Canada, James Evans Avant Garde Award in 2018 from the IEEE Vehicular Technology Society, Joseph LoCicero Award in 2015, and Education Award in 2017 from the IEEE Communications Society. He was also the recipient of the Excellent Graduate Supervision Award in 2006 and Outstanding Performance Award five times from the University of Waterloo and the Premier's Research Excellence Award (PREA) in 2003 from the Province of Ontario, Canada. He was the Technical Program Committee Chair/Cochair for the *IEEE Globecom'16*, *IEEE Infocom'14*, *IEEE VTC'10 Fall*, and *IEEE Globecom'07*, Symposia Chair for the *IEEE ICC'10*, and the Chair for the IEEE Communications Society Technical Committee on Wireless Communications. He was the Editor-in-Chief for the IEEE INTERNET OF THINGS JOURNAL and IEEE NETWORK, and the Vice President on Publications of the IEEE Communications Society. He is a registered Professional Engineer of Ontario, Canada, an Engineering Institute of Canada Fellow, a Canadian Academy of Engineering Fellow, a Royal Society of Canada Fellow, a Chinese Academy of Engineering Foreign Fellow, and a Distinguished Lecturer of the IEEE Vehicular Technology Society and Communications Society.