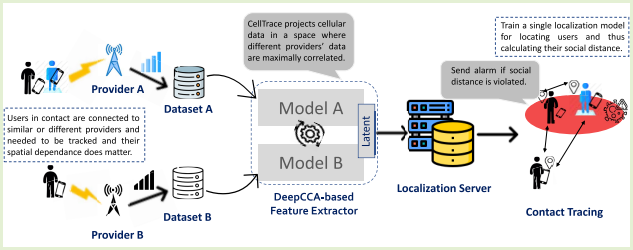
Vaccinated, What Next? An Efficient Contact and Social Distance Tracing Based on Heterogeneous Telco Data

Hamada Rizk, *Senior Member, IEEE*, Asmaa Saeed, and Hirozumi Yamaguchi, *Member, IEEE*

***Abstract*—The demand for safety-boosting systems is always increasing, especially to limit the rapid spread of COVID-19. Real-time social distance preservation is an essential application toward containing the pandemic outbreak. Few systems have been proposed which require infrastructure setup and high-end phones. Therefore, they have limited ubiquitous adoption. Cellular technology enjoys widespread availability and their support by commodity cellphones, which suggest leveraging it for social distance tracking. However, users sharing the same environment may be connected to different telecom providers of different network configurations. Traditional cellular-based localization systems usually build a separate model for each provider, leading to a drop in social distance performance. In this article, we propose *CellTrace*, a deep learning-based social distance preserving system. Specifically, *CellTrace* finds a cross-provider representation using a deep learning version of canonical correlation analysis. Different providers' data are highly correlated in this representation and used to train a localization model for estimating the social distances. In addition, *CellTrace* incorporates different modules that improve the deep model's generalization against overtraining and noise. We have implemented and evaluated *CellTrace* in two different environments with a side-by-side comparison with the state-of-the-art cellular localization and contact tracing techniques. The results show that *CellTrace* can accurately localize users and estimate the contact occurrence, regardless of the connected providers, with a submeter median error and 97% accuracy, respectively. In addition, we show that *CellTrace* has robust performance in various challenging scenarios.**



***Index Terms*—Cellular localization, contact tracing, indoor localization, social distance tracking.**

I. Introduction

THE world is facing a massive threat in the form of a devastating pandemic (COVID-19), which took millions of innocent lives and tremendously affected the world economy and education. On the other hand, experts said it is like any novel disease, but the critical threat lies in its rapid progression infecting millions in a single month. The disease affects the body because a pathogenic infection of COVID-19 mainly comes from direct physical contact with confirmed cases. Therefore, contact tracing approaches have received significant attention as it is the most effective approach for breaking chains of viral transmission [1]. It involves identifying who may have had contact with an infected person with a recursive tracing of their contacts. Despite the presence of vaccines in some countries, we are moving to a post-COVID-19 world where social distancing is the new normality [2]. This confirms the need for an automatic social distance preserving system to ensure such social distancing leading to safe environments, especially indoors, e.g., schools and universities.

Toward realizing contact tracing, Wi-Fi-based systems [3] are proposed due to the prominence of Wi-Fi-based localization systems [4]. These systems leverage the signals received from the installed Wi-Fi access point (AP) infrastructure to estimate the users' locations sharing the same environment. The performance of these systems is obtained only if the area of interest is covered with dense Wi-Fi APs. Nevertheless, neither all environments are well covered with Wi-Fi networks nor do all cell phones enable Wi-Fi by default, limiting the ubiquitous adoption of such systems. A number of systems

Manuscript received 6 March 2022; accepted 21 July 2022. Date of publication 3 August 2022; date of current version 14 September 2022. This work was supported in part by the Japan Society for the Promotion of Science (JSPS), KAKENHI, Japan, under Grant 22K12011, and in part by the Japan Science and Technology Agency (JST), CREST, Japan, under Grant JPMJCR21M5. The associate editor coordinating the review of this article and approving it for publication was Dr. Valerie Renaudin. *(Corresponding author: Hamada Rizk.)*

Hamada Rizk is with the Computer and Control Engineering Department, Tanta University, Tanta 31733, Egypt, and also with the Graduate School of Information Science and Technology, Osaka University, Suita 565-0871, Japan (e-mail: hamada_rizk@f-eng.tanta.edu.eg).

Asmaa Saeed is with the Information Technology Institute (ITI), Giza 12577, Egypt (e-mail: asmaasaeed289@gmail.com).

Hirozumi Yamaguchi is with the Graduate School of Information Science and Technology, Osaka University, Suita 565-0871, Japan (e-mail: h-yamagu@ist.osaka-u.ac.jp).

Digital Object Identifier 10.1109/JSEN.2022.3194540

are proposed for fusing Wi-Fi signals with the onboard inertial sensors (e.g., accelerometers, gyroscopes, and compasses) [5], [6]. However, these systems usually have a location error accumulating over time, which does not hold true for contact tracing applications. Bluetooth Low Energy (BLE) technology has been widely adopted for automatic contact tracing due to its low power consumption and its relatively short communicating distance [7]. These systems utilize the detectable BLE signals to identify encountered users and build contact profiles. However, it has been shown to have higher false negatives due to the relatively low scanning rate compared to the other schemes [7]–[9]. Furthermore, Bluetooth-based contact tracing works only if Bluetooth is enabled on the users' phones, which cannot be ensured. This raises the need for a ubiquitous and more accurate technology for contact tracing and real-time social distance preservation.

On the other hand, cellular technology has several advantages, making it excel its Wi-Fi and Bluetooth counterparts. First, cellular technology is enabled on all cell phones, by definition, including low-end ones. Second, a cellular-based localization system will still work even with a failure in buildings' electrical infrastructure as cellular base stations are better equipped to tolerate power failures. Third, network configuration is somehow fixed (changes rarely occur) as the changing process is complex and costs significant expense when performed frequently. Finally, current cellular-based localization techniques have achieved a fine-grained location accuracy indoors [10], [11] and outdoors [12]. These points suggest cellular to be the underlying ubiquitous technology for social distance and contact tracing.

Current cellular localization systems achieved state-of-the-art performance by using the fingerprinting technique. Fingerprinting is a twofold technique that collects fingerprints of the received signal strength (RSS) from the different cell towers detectable in the area of interest during the offline phase. These fingerprints are used to train a classifier that can be leveraged to differentiate between different locations in the online phase. For this, different classifiers are proposed in the literature, e.g., support vector machines [13], deep feed-forward networks [10], cascaded LSTM layers [11], and stacked autoencoders [14]. Despite the high accuracy achieved by the state-of-the-art cellular-based localization systems, direct application of such techniques for social distance estimation is not possible. Specifically, a social distance tracing system is designed with a view to minimizing the interlocation distance error between different users sharing the same environment. A typical scenario can be found where users of interest are connected to different service providers; each of them has its own spatial distribution of base stations and different power transmission ranges. The naive approach to enable the system for all providers is to build a localization model for each service provider in the country. Although this approach works well for traditional localization systems with a varying localization sensitivity per provider, it leads to performance degradation in social distance tracing. This can be justified as it assumes spatial independence (i.e., independent localization models) between users of different service providers, which is incorrect.

In this article, we propose *CellTrace*: a novel cellular-based social distance and contact tracing system trained with multiple providers' data. To achieve fine-grained accuracy, *CellTrace* builds a deep neural network (DNN) to learn the nonlinear relations between the RSS measured by the users' devices and the corresponding locations in the area of interest and, thus, calculate the social distance between them. Since different users can be connected to different providers, *CellTrace* extracts cross-providers' features enabling the training of a single deep-learning-based localization model. Specifically, *CellTrace* employs a deep canonical correlation analysis (DeepCCA) to learn the complex nonlinear transformations of the RSS of different providers and then project them into a space in which different providers' data at the same location are highly correlated. Then, these projected features are used to train the localization model to detect the locations of different users and their interdistance.

We test *CellTrace* using different Android phones in two indoor environments (small and large sizes). The evaluation results show that *CellTrace* can achieve consistent median localization errors of 0.4 and 0.7 m in the small and large environments, respectively. The system can also correctly detect the exact social distance between two users 97% of the time. This is better than the baseline cellular-based localization system by more than 235% and 117% when used for contact tracing purposes. Moreover, we show that our system is robust when tested across different network operators, lower cell tower densities, device heterogeneity, and unseen locations.

The rest of this article is structured as follows. We briefly describe key concepts relevant to this work in Section II. A literature review is carried out in Section III. In Section IV, we provide an overview of the proposed system, while Section V presents its details. In Section VI, we describe the data collection process and how *CellTrace* is tested. We discuss the system limitations in Section VII. Finally, we conclude this article in Section VIII.

II. Background

In this section, we provide a brief background on the traditional canonical correlation analysis (CCA) on which the DeepCCA is built. The details of our DeepCCA algorithm are given in Section V.

CCA [15], [16] is a standard highly versatile statistical method for finding a common correlation between two multivariate sets of variables (vectors) having the same situations. In particular, CCA **linearly** projects the input sets on another space in which these sets are maximally correlated. This helps in studying the strength of the relationship between two quantitative variables and how they are related. An appealing property of CCA for prediction tasks is that, if there is noise in either set, the learned representations should not contain that noise in the new space.

More formally, assume that $X = [x_1, x_2, \ldots, x_N] \in \mathbb{R}^{d_x \times N}$ and $Y = [y_1, y_2, \ldots, y_N] \in \mathbb{R}^{d_y \times N}$ are two different multivariate variable sets of N samples and feature space of dimensions d_x and d_y, respectively. The goal of CCA is to find K pairs of linear projections (canonical

vectors) $W_x = [w_{x,1}, w_{x,2}, \ldots, w_{x,K}] \in \mathbb{R}^{d_x \times K}$ and $W_y = [w_{y,1}, w_{y,2}, \ldots, w_{y,K}] \in \mathbb{R}^{d_y \times K}$ so that the correlations between $W_x^T X$ and $W_y^T Y$ are maximized. Specifically, CCA aims at finding the projection matrix that maximizes the correlation coefficient α between $W_x^T X$ and $W_y^T Y$ as

$$\alpha\left(W_x^T X, Y^T W_y\right) = \frac{W_x^T X Y^T W_y}{W_x^T X X^T W_x W_y^T Y Y^T W_y}. \tag{1}$$

That is, we want to find

$$\left(W_1^*, W_2^*\right) = \underset{W_x, W_y}{\operatorname{argmax}} \frac{W_x^T X Y^T W_y}{W_x^T X X^T W_x W_y^T Y Y^T W_y}. \tag{2}$$

Since α is scaling-invariant, we can rewrite the correlation as

$$\left(W_1^*, W_2^*\right) = \underset{W_x, W_y}{\operatorname{argmax}} W_x^T X Y^T W_y \quad \text{s.t.} \quad W_x^T X X^T W_x = 1, \quad W_y^T Y Y^T W_y = 1. \tag{3}$$

To find the optimum solution for (3), one has to solve the general eigenvalue problem of the form [17]

$$\begin{bmatrix} \mathbf{0} & \Sigma_{xy} \\ \Sigma_{yx} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{W_x} \\ \mathbf{W_y} \end{bmatrix} = \lambda \begin{bmatrix} \hat{\Sigma}_{xx} & \mathbf{0} \\ \mathbf{0} & \hat{\Sigma}_{yy} \end{bmatrix} \begin{bmatrix} \mathbf{W_x} \\ \mathbf{W_y} \end{bmatrix} \tag{4}$$

where $\hat{\Sigma}_{xx}$, $\hat{\Sigma}_{yy}$ are the covariance matrices. Σ_{xy} and Σ_{yx} are defined as follows: $\Sigma_{xy} = (1/N) X Y^T$ and $\Sigma_{yx} = (1/N) Y X^T$.

By solving (4), we get K eigenvectors $\{[\mathbf{W}_{x,k}; \mathbf{W}_{y,k}]\}_{k=1}^{K}$ and the corresponding Kth eigenvalue that is equal to the correlation coefficient (1). Therefore, the aimed projection matrix W is the set of obtained eigenvectors.

In this article, we adopt a deep learning-based version of CCA, denoted DeepCCA [18], which can be viewed as a **nonlinear extension** of the traditional CCA.

III. Related Work

In this section, we focus on two groups of related work classified into localization and contact tracing systems.

A. Localization Systems

1) Sensor-Based Systems: Over the last decade, many indoor localization systems have been proposed relying on the available sensors of modern smartphones [19], [20]. The systems [5], [6], [19], [21] utilize the smartphones' inertial sensor (e.g., magnetometer, accelerometer, and gyroscope) to track users based on the dead-reckoning technique. However, these measurements include noise components that accumulate quickly over time, leading to a severe deterioration in the localization performance. Over the years, several systems have been proposed to handle this issue, e.g., Zee [19] employs map matching to lessen the localization error, while Unloc [21] and SemanticSLAM [22] opportunistically reset the error on encounters of the detected building landmarks. On the other hand, Headio [20] proposes a computer vision-based solution leveraging the smartphone's front camera to correct the location drift. Sensor-based systems run only on high-end phones equipped with inertial sensors.

CellTrace, on the other hand, utilizes only the ubiquitous cellular technology, which is supported by all mobile phones by definition. In addition, contact tracing can realize at the provider side without either incurring extra processing or fooling the localization system at the edge.

2) Wi-Fi-Based Systems: Wi-Fi is widely available indoors and has been widely adopted for indoor localization due to its relatively short transmission range. To achieve high accuracy for localization, Wi-Fi-based techniques usually depend on building a Wi-Fi radio map of the overheard Wi-Fi APs, which can be leveraged to identify the user location based on matching the received signals. This matching can be either deterministic [23] or probabilistic [24]. Many systems have been proposed to address several Wi-Fi-based challenges over the years, introducing deep learning for the localization task as in [25] and [26]. Despite the fact that Wi-Fi-based techniques have high localization accuracy due to the limited propagation range of Wi-Fi APs [27], they cannot compete favorably with other technologies, e.g., cellular, due to the requirements for high coverage and frequent maintenance.

On the other hand, *CellTrace* leverages the widespread cellular technology, whose network configurations rarely change due to the consequent high expense and complexity of this process when performed frequently. In addition, unlike Wi-Fi networks, cellular signals have a longer propagation range and are less affected by variations in indoor environments [28]–[31]. This leads to a stable infrastructure for localization-based safety systems.

3) Cellular-Based Systems: Due to its high advantages, such as the fact that cellular technology has the most widespread infrastructure and is supported by the vast majority of mobile devices, cellular-based localization has recently gained a lot of attention. Therefore, cellular-based localization systems have been adopted for both outdoor and indoor use cases. The methodology of this technique is that a model is built and trained to learn the relations between the collected RSS measurements and the user locations during the offline phase. Then during the online phase, this model must be able to discriminate between different locations in the area of interest. There have been proposals for both outdoor and indoor cellular-based localization systems. Outdoor cellular-based systems [12], [32], [33], have been proposed as energy-efficient and ubiquitous alternatives for GPS. Furthermore, cellular-based localization has recently been well realized in indoor settings leveraging the computational power of deep learning [10], [11], [14], [34]. For instance, to learn the nonlinear relation between the received signals and the user locations, a deep fully connected neural network is utilized in CellinDeep [10], a deep LSTM network is adopted in MonoDCell [11], and an autoencoders' network is considered in [14] and [34]. However, such typical techniques assume that localization models between users of various service providers are independent, resulting in poor performance when used to track the social distance of users sharing the same environment.

Unlike state-of-the-art cellular-based systems, *CellTrace* extracts latent features, ensuring that different providers are maximally correlated in the shared space. In addition,

the localization model is ensured to generalize and avoid overfitting by using different regularization methods. As a result, both the location estimation accuracy and the contact tracing performance have significantly improved.

B. Contact Tracing Systems

The global outbreak of the Coronavirus has highlighted the importance of contact tracing even after the presence of vaccinations [35], [36]. BLE-based contact tracing is the most prevalent technology. Several systems based on Bluetooth or BLE have been rolled out, supported by the governments of various countries, such as Singapore [37] and Australia [38]. The basic idea of using BLE-based systems is to detect the enabled Bluetooth of the users in the vicinity and their identifier. Initial attempts of this technology face privacy concerns due to its requirement for clients to share contact logs to a central reporting server [36], [39]. Thereby, the authorities can detect people who may have had close contact with the infected one and notify them promptly to break the infection chain of diseases. However, to handle the privacy issues, the Decentralized Privacy-Preserving Proximity Tracing (DP-3T) protocol [40] is developed to facilitate privacy-preserving digital contact tracing of infected cases. This protocol ensures that the central server does not access contact records. However, this comes at the cost of requiring intensive computation on the client side to process infection reports. The Apple/Google Exposure Notification project adopts similar principles on the operating system level [41], [42], which has been widely adopted [43].

Few approaches [44], [45] were recently proposed to track passengers on public transportation depending on the smartphones' inertial sensors (e.g., magnetometer, accelerometer, and gyroscope) using dead reckoning. However, the inherent noise in sensor data leads to an error that accumulates quickly over time resulting in limited accuracy and robustness.

The Wi-Fi-based solution [3], [46], which features dominantly in the indoor positioning research, has become more attractive for pandemic tracking, thanks to the increasing number of indoor and outdoor public Wi-Fi APs. The research [47] demonstrated that, when at least ten Wi-Fi APs were nearby, contact detection using pure Wi-Fi RSS could closely match the accuracy of GPS (used as a reference) in the city center. However, this accuracy may be inadequate for contact tracing, and the expected Wi-Fi coverage cannot be realized indoors. Contact tracing based on GPS cannot work inside buildings due to the absence of line-of-sight to the reference satellites [35], [36].

In contrast, cellular-based contact tracing has shown to be feasible due to its availability and ubiquity, which encouraged some governments to use in emergency time [35], [36], [48]. Moreover, it can be deployed on both the client side and the provider side enabling governments to control the spread of the virus. Motivated by these advantages, *CellTrace* enables cellular-based contact tracing for heterogeneous network providers.

IV. System Overview

Fig. 1 shows the *CellTrace* system architecture. *CellTrace* works in two phases: an offline training phase and an online interdistance location estimation phase. The offline phase starts with the **data collection** process using a client-side application running on the user's cell phone. The application is designed carefully to record the time-stamped cellular information from the overhead towers at sparse predefined points called reference points in the considered area.[1] These collected measurements are uploaded to our online running service for further processing. The **preprocessor** module is used to handle the noise in the input data and prepare the low-level RSS feature vector of each considered provider.[2] As a result, a fixed-size RSS vector across all the recorded samples has been obtained that fits as input to the localization model. The RSS vectors from different providers are then further processed by the **deep feature extractor** module to learn complex nonlinear transformations and project the original low-level features to a cross-provider feature space. The module is based on a combination of a DNN and a CCA process, ensuring that data from different providers are highly correlated in the common space, as described in Section V-B. Thereafter, the projected RSS features are fed to **localization model**; hence, we can calculate the social distancing between each pair of users based on an accurate estimate of their locations. The output of this offline phase is two trained models (i.e., the deep correlation model and the localization model), which are saved for later use in the online phase.

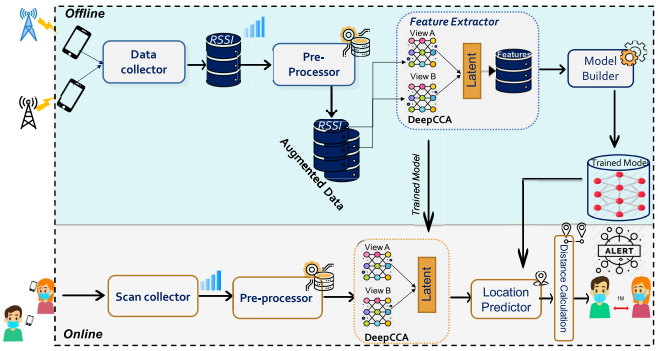


Fig. 1. *CellTrace* system architecture.

During the online phase, users are tracked in real-time by carrying their phones to unknown locations scan for the covering towers. The scans are forwarded to the *CellTrace* server. These data are first handled by the **preprocessor** module to extract the RSS feature vectors. Thereafter, the online predictor module feeds the data to the trained **deep correlated model** to extract the desired features. Finally, the **location estimation model** feeds the data to the localization model trained in the offline phase to estimate the likelihood of the user being at the different reference points trained during the offline phase. Based on this likelihood, the system obtains the user's location in the continuous spatial space. The

[1]During the data collection process, the received signal measurements can be captured by either manual fingerprinting [10], [25] or crowdsourcing [12], [26].

[2]This RSS vector is called a low-level vector due to its tightly coupled to each provider considered in the data collection.

predicted locations of people in the area of interest are used to gauge and warn users who violate the social distance rule.

V. The *CellTrace* System

Fig. 1 shows the different modules of the *CellTrace* system. In the balance of this section, we describe the details of each module.

A. Preprocessor Module

This module runs during both the offline and online tracking phases. Each cell provider has a number of towers covering the area of interest. However, the number of detected cell towers per scan is limited to seven or less, by default [10], [33]. From the whole covering towers, the selected group of towers may vary across different scans; even, the user is stationary at the same location. Therefore, this module unifies the number of detected towers and produces a q-length (i.e., the length of all covering towers) RSS feature vector $s = (s_1, s_2, \ldots, s_q)$ across different scans. The benefit of this is to obtain a consistent set of cell towers that fit the input of the deep model as described in Section V-B. Each entry of the vector represents the RSS from a certain tower, while nondetected towers in an arbitrary scan are set to 0 ASU.[3] It is worth noting that some scans include false network information, e.g., tower IDs of 65636, which is not a valid tower for any location area code (LAC) [50]. This anomalous event usually takes place in a short period of time during radio access technology (RAT) change [51] and can be practically detected by the absence of an LAC ID, which must be included in any scan. Thus, the spurious cell towers are deleted. Toward having a fast convergence time [52], the RSS values received from the covering towers are normalized to the range of [0, 1].

Finally, this module mitigates the burden in collecting a large amount of training data, as required by deep models, by employing our data augmentation framework proposed in [53]. The framework generates a massive amount of synthetic data from samples collected over a short time frame that reflects the typical RSS variations. In addition, *CellTrace* employs spatial augmentation introduced in [11] in order to generate synthetic RSS measurements for nonsurveyed data points to reduce the calibration effort further. These techniques have the advantage of ensuring model generalization and overfitting avoidance.

Note that the deployment of *CellTrace* does not require information about the physical locations of the covering cell towers.

B. Feature Extraction Module

This module aims to transform the preprocessed RSS feature vectors of different providers into a latent space in which they are highly correlated. This has the benefit of consolidating the input to the localization model and considering the spatial dependence of users of heterogeneous providers. Toward achieving this, we adopt DeepCCA [18] to discover that latent space. Compared to the classical CCA, which linearly transforms the input views into highly correlated projections, DeepCCA solves the same objective function by realizing more powerful **nonlinear** projections in a new latent space using DNNs. These projections are learned via the gradient descent technique. The intuition behind leveraging the deep version of CCA is the ability of the DNN to learn complex relations from the noisy cellular data automatically.

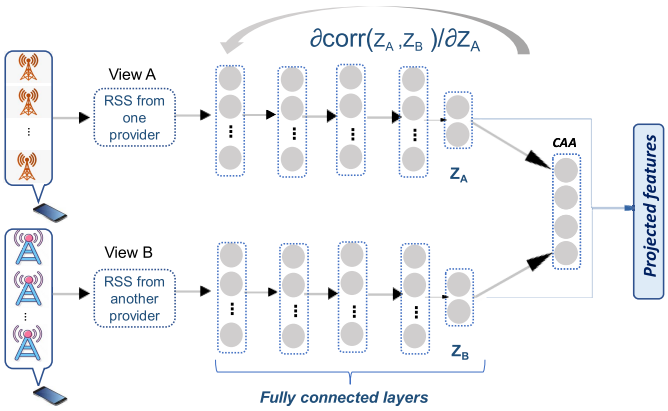


Fig. 2. Network structure of the feature extraction module. It consists of two deep networks learned so that the output layers (topmost layer of each network) are maximally correlated. A CCA layer is stacked on top of a fully connected layer to calculate the correlation between the views.

Fig. 2 shows the schematic of the proposed DeepCCA feature extraction model. As shown in the figure, DeepCCA consists of two independent DNNs, one for each cellular provider. Each DNN consists of cascaded fully connected layers. The input layers of DNN A and DNN B are RSS vectors as detected simultaneously from the corresponding covering towers of each provider. In general, the size of the input vectors varies based on the number of towers of each provider. These DNNs are then trained to encode these inputs to a fixed-size subspace where the corresponding output vectors (z^A and z^B) are maximally correlated.

Specifically, let X_A be a set of RSS input vectors of provider A and X_B be the corresponding set of RSS vectors of provider B, which are collected simultaneously at the same set of reference points. These provider-dependent matrices are fed to the DeepCCA subnetworks to obtain the aimed cross-provider representations. For the instance, the output of the first layer of network A is $h_1^A = \sigma(W_1^A X + b_1^A)$, where σ is a nonlinear activation function (e.g., logistic Sigmoid) applied componentwise, W_1^A is a matrix of weights, and b_1^A is a vector of biases. The output of each layer is used to calculate the output of the next layer and so on until the final layer d whose output is $f_A(X_A) = \sigma(W_d^A h_{d-1} + b_d^A)$, which is the intended latent representation (z), i.e., the spatially correlated feature vector. Similarly, the representation obtained by the second DNN is $f_B(X_B) = \sigma(W_g^B h_{g-1} + b_g^B)$ with different parameters: W_g^B, b_g^B, and g. The objective of DeepCCA is to jointly learn the parameters θ_A and θ_B for both neural networks such that the correlation between z^A and z^B is maximum. Therefore, the objective function of

[3]The RSS is usually measured by the user's phone in the arbitrary strength unit (ASU). It represents an integer range of values of [0–31], which is linearly proportional to the decibel-milliwatts (dBm) unit [10], [49].

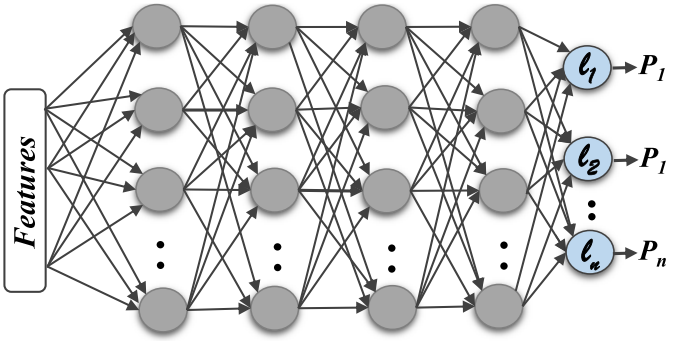


Fig. 3. Deep network structure.

DeepCCA is defined as follows:

$$\left(\theta_A^*, \theta_B^*\right) = \underset{(\theta_A, \theta_B)}{\operatorname{argmax}} \operatorname{Corr}\left(f_A\left(X_A; \theta_A\right), f_B\left(X_B; \theta_B\right)\right). \quad (5)$$

Toward achieving this, we compute the correlation and its gradient with respect to the output layers, and then, the back-propagation is used to update the parameters of both networks. This process is repeated until convergence is obtained.

Given that, in general, optimization of deep models may not achieve the best performance if the model parameters are initialized randomly, therefore, the *feature extraction* module utilized denoising autoencoders [54] to better initialize the parameters of each layer.

Unlike traditional deep learning methods [10], [14], [25] that are trained to maximize the likelihood of target class (location) for radio scans from individual providers, *CellTrace* leverages DeepCCA with the correlation-based objective function. This empowers the **robustness of interdistance** (social distance) calculations of the system as it ensures the **spatial dependence** of the signals received from different providers covering the same area of interest.

C. Location Estimation Model

This module is responsible for utilizing the correlation features (z) extracted from the DeepCCA network to train a localization model and find its optimal parameters. The trained model is used in the online phase by the *online location predictor* module. DNNs have been considered to be one of the staple techniques in machine learning in use today. Based on the universal approximation theorem [55], DNNs could be considered capable of approximating any arbitrary function, provided that they were suitably complex. Therefore, a neural network is adopted in this module.

Architecture: Fig. 3 shows the structure of the considered DNN for localization. Specifically, *CellTrace* adopts a fully connected feedforward neural network. The hierarchical representation of *CellTrace* is obtained by *four* hidden layers of nonlinear processing units. The rectified linear unit (ReLU) (the state of the art of nonlinearity) is used as the activation/transfer function for the hidden layers due to its sparsity and immunity to vanishing gradient problems [56].

The input layer of the network is a vector z of length v, which is obtained from the *feature extraction* module (as described in Section V-B). The output layer consists of a number of neurons corresponding to the number of surveyed reference points in the area of interest considered in this phase. The network is trained to operate as a classifier such that each reference point represents a class. Unlike equivalent regression models, classification models usually have a simpler data collection process (i.e., permits collection at low-density reference points). Therefore, a softmax activation function is leveraged at the output layer. This leads to a probability distribution over the different predefined reference points given an input. In particular, the network outputs the probability that the input sample (the latent representation) comes from a specific reference point. More formally, given a total number of training samples m, where $z_i \in \mathbb{R}^v$ is the projected latent representation of each cell scan $s_i \in \mathbb{R}^q$, which is fed to the model, the sample z_i has a corresponding discrete outputs (i.e., logits), and c_i is $a_i = (a_{i1}, a_{i2}, \ldots, a_{in})$, which captures the score for each reference points from the possible n reference points to be the estimated point. The logit scores a_{ij} (for sample i to be at reference point j) are converted into probabilities using the softmax function as

$$p\left(a_{ij}\right) = \frac{e^{a_{ij}}}{\sum_{j=1}^{j=q} e^{a_{ij}}}. \quad (6)$$

For training purposes in the offline phase, we encode the ground-truth label of each sample using one-hot-encoding. The encoding of the output vector has a probability of one for the correct reference point and zeros for others. We used the Adam optimizer [57] and categorical cross-entropy as our loss function.

To avoid overtraining (i.e., overfitting), *CellTrace* employs the early stopping regularization technique, which automatically selects the optimal number of training epochs. Specifically, early stopping monitors the model's performance for every epoch on a held-out validation set during the training. It terminates the training as soon as the performance stops improving [58].

D. Online Phase

The goal of this phase is to pinpoint the users sharing the same environment and, thus, detect the contact occurrence. Initially, each user's device identifies the provider, measures the received cell signals from the hearable towers in the area of interest, and forwards the scan to our running service to process and extract the corresponding feature vector. Specifically, the RSS vector is submitted to a single view of the trained DeepCCA, which corresponds to the user's network provider to extract a cross-provider feature vector, as described in Section V-B. This vector is then fed to the trained localization model (regardless of the provider) to get a location estimate as one of the already defined reference points at the calibration phase. Then, the user's location l^* is estimated as the one that has the maximum probability given the input vector (z). That is, we want to find

$$l^* = \underset{l}{\operatorname{argmax}} \left[P\left(l|z\right)\right]. \quad (7)$$

One advantage of designing the *localization model* to operate as a classifier rather than a regressor is reducing the amount

of the required reference points and, thus, the data collection burden. However, the classification model can only predict the user locations at the predefined few discrete reference points. As a result, this will lead to a poor user experience as the estimated locations will be spaced out even with a very accurate model. To ensure the required fine-grained tracking of the users in the continuous spatial space,[4] *CellTrace* estimates l^* as the spatial-weighted average of all reference points at the output layer, where the weight of each point is chosen as its corresponding softmax likelihood. More formally, the probabilistic output (obtained by softmax) of the network is denoted as $P = [P_1, P_2, \ldots, P_n]$, where n is the number of reference points in the area of interest and P_i $(1 \leq i \leq n)$ represents the possibility that the input vector is coming from the ith reference point l_i. P_i is formulated as follows:

$$P_i = P\left(l_i|z\right). \tag{8}$$

Thus, the fine-grained location coordinates are defined as

$$l_x^* = \frac{\sum_{i=1}^{n} P_i l_{ix}}{\sum_{i=1}^{n} P_i} \tag{9}$$

$$l_y^* = \frac{\sum_{i=1}^{n} P_i l_{ix}}{\sum_{i=1}^{n} P_i} \tag{10}$$

where l_{ix} and l_{iy} are the coordinates of reference point i.

The objective of *CellTrace* is to detect in real time if two users (carrying their cell phones) are in contact. This can be achieved by calculating interdistance (social distance, d) between users sharing the same environment over the course of a predefined time interval τ time steps, e.g., 16 time steps[5]

$$d = \frac{\sum_{1}^{\tau} \sqrt{\left(l_x^1 - l_x^2\right)^2 + \left(l_y^1 - l_y^2\right)^2}}{\tau} \tag{11}$$

where (l_x^1, l_y^1) and (l_x^2, l_y^2) are the location coordinates of user 1 and user 2 at an arbitrary time step, respectively.

Users whose interdistance is less than one meter (social distance violation) are alerted by vibrating their phone or sending a text message.

E. Design Issues

The proposed system has two modes of operation (i.e., deployment): client- and provider-side modes. First, the client-side mode involves installing the CellTrace app on the user devices. This app is connected to the localization server to provide positioning and contact-tracing services in an emergency. In addition, based on the response from the server, the app notifies encountered users when a contact is detected. In this case, the users install the app and approve the transmission of their signals to the localization server. Despite the simplicity of this approach, it cannot guarantee efficient contact tracking and safety in practice as it requires all users to install the app. On the other hand, the system can be deployed on the provider side. This mode has better

[4]*CellTrace* can locate the user anywhere even in locations different from reference points.

[5]An interval of 16 time steps can be translated to seconds by dividing the number of time steps by the scanning rate.

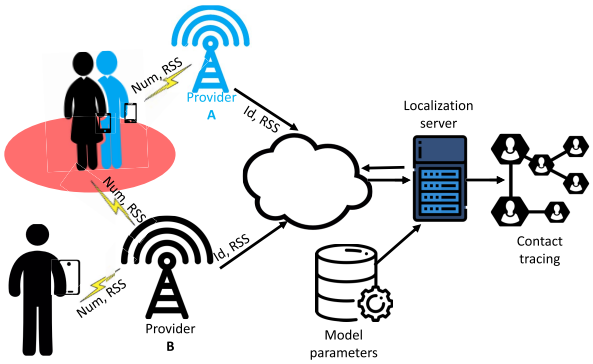


Fig. 4. Social distance and contact tracing.

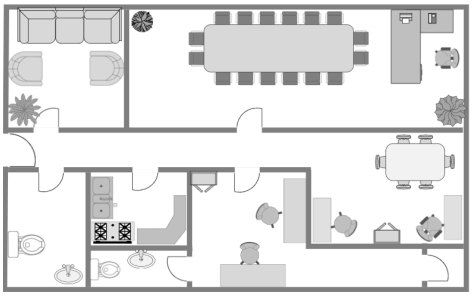


Fig. 5. Layout of testbed 1.

contact tracing efficacy, encouraging some governments to adopt it at least for short periods [48]. Each provider is connected to the shared contact tracing server, which processes anonymized client cellular measurements, and thus, contacts can be identified, as shown in Fig. 4. Then, notification messages are sent from the provider to users upon contact occurrence of infected cases. However, this approach may face some challenges in adoption. For instance, privacy concerns may hinder the adoption in some countries as end-users have not provided their consent to use their data by their providers for contact tracings [48], [59]. Additionally, obtaining consent from different providers to deploy a common contact tracing system is rather difficult, even in an emergency. However, there are tradeoffs in the effectiveness of contact tracing and exposure notification apps with increased privacy [48], [59]. In particular, the effectiveness of the privacy-first apps might be impossible to evaluate due to the lack of recorded data [59].

Nevertheless, *CellTrace* can be extended for further privacy protection by anonymizing users' data, employing differential privacy [60] or inheriting the decentralization concept from other techniques, e.g., inheriting the concept of DP-3T [40] or exposure notification [41], [42].

VI. Evaluation

In this section, the data collection setup and tools used are described first. Then, we show how the system performs by varying the different system parameters. Finally, we compare the performance of *CellTrace* to the state-of-the-art techniques.

A. Data Collection

We deployed our system in two indoor environments with different sizes and characteristics (as described in Table I).

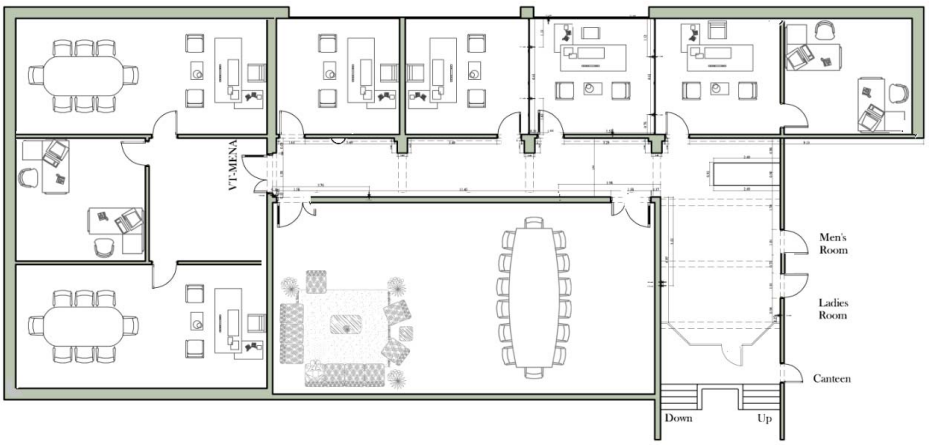


Fig. 6. Layout of testbed 2.

TABLE I
SUMMARY OF THE TEST BEDS CONSIDERED IN EVALUATING *CellTrace*

Criteria	testbed 1	testbed 2
Area (m^2)	11×12	17×37
Spacing of reference points (m)	1	1.5
Number of fingerprint points	55	310
Sampling rate (scan/sec)	3.33	3.33
Building Material	Brick	Brick
Total number of covering cell towers/ provider	16/A, 9/B	37/A, 20/B

The first is a lab on our university campus building (denoted test bed 1) with an area of 132 m^2 containing offices, a meeting room, and corridors (see Fig. 5). The second environment (denoted as test bed 2), as shown in Fig. 6, is a full floor in our university campus having a total area of 629 m^2 containing several offices, labs, and corridors with more furniture. The training data were collected from 55 and 310 reference points located throughout test bed 1 and test bed 2, respectively. The reference points were uniformly distributed over the area of interest with 1- and 1.5-m spacings. The system parameters are listed in Table I.

Data have been collected using an Android app that continuously scans for the network information from the overheard cell towers in the area of interest. The app records cell information, including cell tower identifier (CID), LAC, and the corresponding time-stamped signal strength with a scanning rate set to 3.33 scan/s. To collect different provider's data concurrently, the same application runs synchronously on all mobile devices. Each device is connected to a different provider with one device dedicated to controlling ground-truth profiling and starting/stopping the data collection process on all devices. The application visual interface is designed to depict the test bed floorplan in the foreground of the master device. The user tags her current location on the displayed test bed as a ground truth triggered by a long tap on the map interface. *Five* participants are engaged in the data collection process using different Android phones (e.g., HTC One X9, Google Pixel XL, Tecno Phantom 6, HTC One E9, Motorola Moto G5, and ZTE Blade 7). To consider the time-variability effect on cellular signals, the data were captured and recorded across different days. To scan cell towers in the area of interest, we developed a scanning application using the Android SDK.

To evaluate the learned model and confirm its generalization ability, we adopted K-fold cross-validation (typically $k = 5$). The training set is partitioned into k subsets where each subset includes the data collected from two providers.[6] Each time, $k-1$ subsets are used to form a two-view training set, and the remaining one is used as the validation set. Hence, every subset appears in the validation set exactly once and appears in a training set $k-1$ times. Then, the average error across all k folds is reported and is used to select the model parameters. This significantly reduces the impact of the bias-variance problem due to the interchange of the training and validation sets.

We implemented our deep learning-based training using the Keras learning library on top of the Google TensorFlow framework [61]; the training was carried out on the Google collaboratory cloud platform.[7]

B. Effect of Changing CellTrace Parameters

In this section, we study the effect of the deep models' different hyperparameters, *CellTrace* parameters, and the different techniques used to learn nonlinear transformations for achieving the maximum correlation between the data views on the overall system performance. These parameters include the number of layers, the effect of the feature extraction method, and the size of the feature vector. The default parameters' values used throughout the evaluation section are reported in Table II.

1) Number of Layers in the Network: Fig. 7 shows the effect of changing the number of layers on *CellTrace* accuracy. The figure shows how increasing the number of layers of the location estimation network increases its accuracy until it reaches an optimal value at *four* layers. This can be justified as increasing the number of layers increases the model computing

[6]Without loss of generality, we got permission to use provider-side data for two providers only to ensure the system's validity on both sides, i.e., the client side and the provider side. However, *CellTrace* can work with any number of available providers by creating a view for each provider.

[7]https://colab.research.google.com

TABLE II
DEFAULT PARAMETERS' VALUES USED IN THE EVALUATION

Parameter	Range	Default
Provider	A, B, C	A, B, C
Learning rate	0.0001 - 0.2	0.001
Number of hidden neurons	20 - 1000	300
Batch size	1-Dataset size	256
Number of layers	1 - 30	4
Early Stopping Patience (epochs)	1-10	10
Number of samples per reference point	20 - 3000	3000
Number of epochs	Automatic by Early stopping	
Number of classes	1-55	55
Used devices	HTC One X9, Tecno Phantom 6, Google Pixel XL, ZTE Blade 7, Motorola Moto G5 and Samsung Note 3	
Number of users	5	

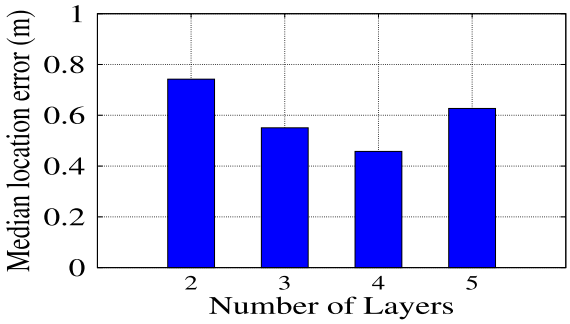


Fig. 7. Effect of changing the number of layers on *CellTrace* accuracy.

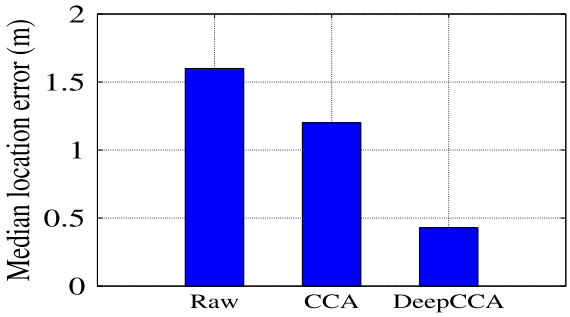


Fig. 8. Effect of feature extraction module on *CellTrace* performance.

power to avoid underfitting and, thus, better fits the function. However, the deeper the model, the more likely it is to overfit the training data, reducing its flexibility and accuracy when handling users from different providers. *Four* layers are set as the default number of layers in *CellTrace* model to achieve the balance between underfitting and overfitting, It is worth noting that applying DeepCCA radically simplifies the classification problem in the projected space. As a result, a four-layer network is sufficient for the classification of features in the projected space.

2) Feature Extraction Method: In this section, we study the influence of the different feature extraction techniques on the overall system performance. Fig. 8 compares the effect of using DeepCCA for extracting provide-invariant features to either using individual provider models or feature projected using traditional CCA [16] on *CellTrace*'s estimation accuracy of social distance. The figure depicts that, in comparison with

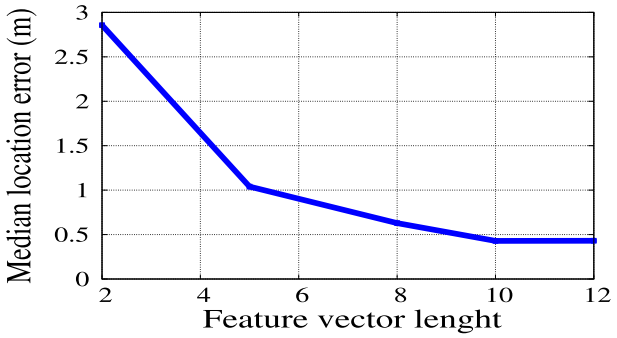


Fig. 9. Effect of changing the feature vector length on *CellTrace* accuracy.

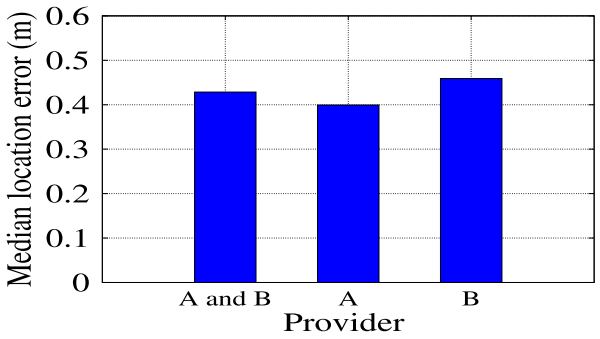


Fig. 10. Effect of using data from different service providers.

the raw features and classic CCA, the proposed DeepCCA method gives an improvement of 235% and 185% in the estimation accuracy of social distance. These results confirm the efficacy of DeepCCA in capturing the correlated signatures of different providers better than other methods, which facilitates locating users sharing the same environment.

3) Feature Vector Length: Fig. 9 shows the location estimation accuracy of *CellTrace* as a function of the latent space dimension size obtained by the DeepCCA network. It is clear from the figure that increasing the size of the latent feature vector z improves the *CellTrace* performance. The figure also shows that a feature vector z of ten dimensions yields the best performance. Beyond that, new dimensions (i.e., features) will be included leading to no further performance enhancement. This can be justified for two opposing reasons: 1) the additional features reduce the correlation between different providers' data and 2) on the other hand, the location discriminative power of the localization model is boosted by increasing the number of features in the latent space.

C. Robustness Experiments

In this section, we evaluate the robustness of *CellTrace* under varying environmental conditions.

1) Resilience to Provider Heterogeneity: In this section, we evaluate the performance of the *CellTrace* system when tested with two different providers individually compared to the heterogeneous providers' scenario. Fig. 10 shows the performance of *CellTrace* when all users are connected to either only A or only B compared to A and B together (heterogeneous providers). It is worth noting that different operators cover the area of interest with different densities of serving towers of 16 and 9 for operators A and B, respectively.

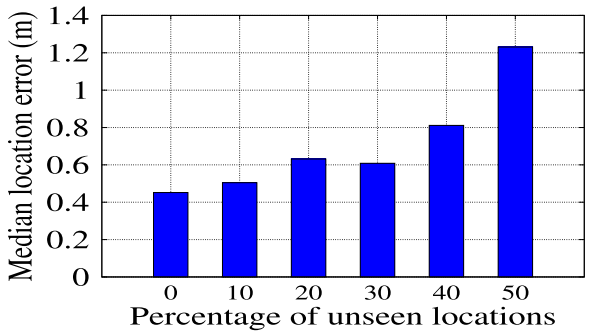


Fig. 11. Effect of testing with unseen locations on the *CellTrace* system performance.

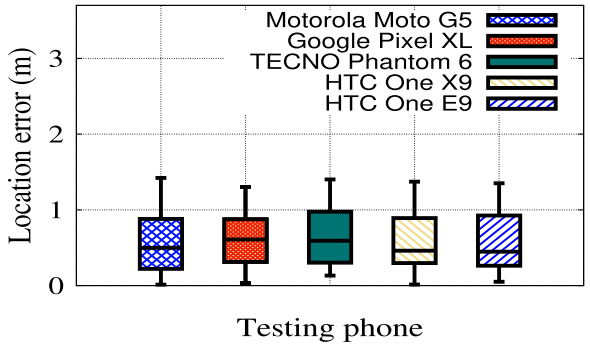


Fig. 12. Performance of the system when tested with heterogeneous phones.

The figure shows that *CellTrace* achieves a consistent accuracy across different providers with a low as 0.5-m median interlocation error (social distance error). This highlights the system's resilience to heterogeneous providers of varying densities of cell towers with different signal distributions. This can be explained as a result of leveraging the feature extraction module that can learn the correlation between two views of data to get a robust and fixed-size feature vector for training a localization model on different providers' data.

2) Unseen Locations: Fig. 11 shows the performance *CellTrace* when testing with unseen locations (i.e., never considered in training). This is done by reducing the number of surveyed locations during the calibration phase (offline) by a particular percentage; the same percentage reduces the fingerprinting overhead. *CellTrace* is robust to unseen locations. As shown in the figure, even with a 50% reduction in the number of training locations (i.e., only 27 locations are considered out of 55), *CellTrace* can obtain a location error of less than 1.3 m. This advantage is due to the employment of data augmentation techniques that compensate for reference points' loss. This result enables *CellTrace* to be deployed at scale.

3) Device Heterogeneity: In this section, we evaluate the resilience of the model to cope with device heterogeneity. To do that, we employed leave-one-out cross-validation [62], which iterates over all devices, and each time holds data captured by one device out for testing while training with data recorded by the remaining devices. Fig. 12 shows the performance corresponding to each testing device. The figure shows that *CellTrace* provides consistent performance even with the device variability. This can be justified due to the ability of the feature extraction network to map the input

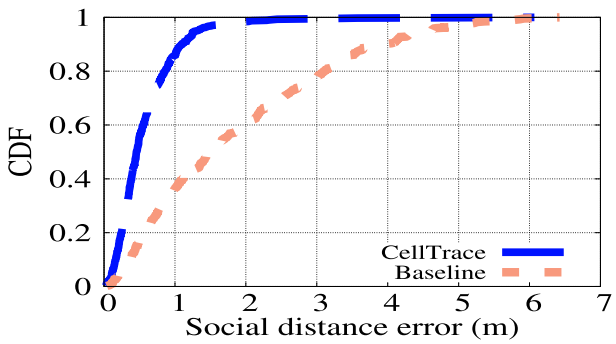


Fig. 13. CDF of the localization error in test bed 1.

cell measurements into the latent space where correlation is maximized. Specifically, as verified in [14], the effect of device heterogeneity can be modeled as random noise added to the measurements. Transformation into the new space using DeepCCA has the effect of reducing the localization model dependence on the absolute input values by considering relative representation that maximizes the correlation. This result highlights the system robustness to the device heterogeneity problem.

D. Comparative Evaluation

In this section, we evaluate the end-to-end performance of *CellTrace* in terms of localization performance and contact tracing accuracy, and compare it to the state-of-the-art cellular location estimation and contact tracing systems.

1) Social Distancing: In this section, we evaluate the performance of *CellTrace* compared to baseline cellular localization techniques [10], which builds an individual model for each provider. Figs. 13 and 14 show the CDF of social distance error for the two techniques in test bed 1 and test bed 2, respectively. Fig. 13 illustrates that *CellTrace* outperforms the baseline, enhancing the median error by 280%. Similarly, for the second test bed, *CellTrace* achieves an improvement in median localization error of 117% compared to the baseline [10]. This can be explained by noting that, unlike *CellTrace*, the baseline technique, which relies on the original signal features, does not consider the interoperability between different operators when their connected clients share the same spatial environment. It is worth noting that a slight drop in the accuracy is observed in test bed 2, which can be justified due to the increase in the spatial uncertainty in a larger space compared to test bed 1. This can be easily handled by space partitioning. Nevertheless, the results in the two test beds confirm the superiority of *CellTrace* due to its ability to capture the correlated features across different providers compared to the baseline.

2) Contact Tracing: In this section, we evaluate the overall accuracy of *CellTrace* in contact detection. Table III summarizes the performance metrics in contact tracing. It is worth noting that positive means that contact is detected by the system, which can be either correct (true) or incorrect (false) detection and similarly for the negative detection. However, a false positive case occurs when users are more than 1 m apart (no physical contact). At the same time, the system reports a contact leading to a false alarm and, thus, a bad user experience. In addition, the false-negative case occurs when the

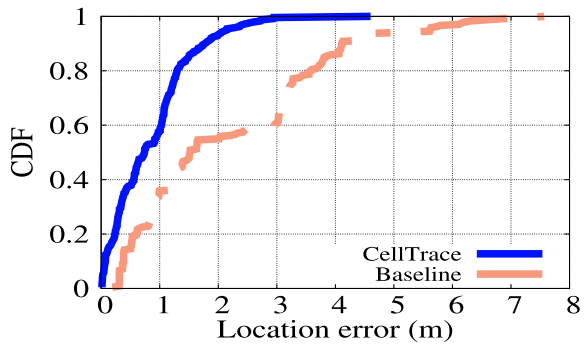


Fig. 14. CDF of the localization error in test bed 2.

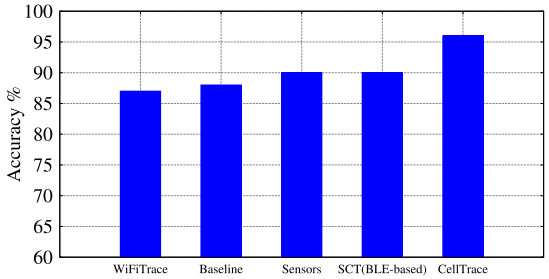


Fig. 15. Performance of the system compared to contact tracing methods.

TABLE III
CONTACT TRACING METRICS CONSIDERED IN EVALUATING *CellTrace*

Social distance threshold	1m		1.5m	
metric	*CellTrace*	Baseline	*CellTrace*	Baseline
True positive rate (TPR)	0.88	0.43	0.95	0.74
False positive rate (FPR)	0.01	0.3	0.0	0.08
True negative rate (TNR)	0.99	0.7	1.0	0.92
False negative rate (FNR)	0.11	0.57	0.06	0.25
Overall accuracy	94%	55%	97%	59%

system missed a real contact reducing the system's feasibility. The results in the table indicate that the *CellTrace* achieves TPR, TNR, FPR, and FNR of 0.95, 1.0, 0.0, and 0.06, and 0.88, 0.99, 0.11, and 0.01 with social distance thresholds of 1.5 and 1 m, respectively. These results confirm the superiority of *CellTrace* compared to the baseline technique in both cases.

Fig. 15 shows the performance of *CellTrace* compared to the state-of-the-art contact tracing systems. The system in [63] (denoted sensors) proposed a contact tracing approach-based pedestrian dead reckoning (PDR) fusing Wi-Fi measurements and smartphone sensors. WiFiTrace [3], on the other hand, collects Wi-Fi network logs, specifically association and dissociation log entries for each device, at various APs in the area of interest and then uses them to reconstruct the locations visited by the user for contact tracing. Furthermore, *CellTrace* was compared against the Bluetooth-based system called SCT [7],[8] which used a decision tree classifier to categorize user contacts as low or high risk based on BLE signals.

The figure illustrates that *CellTrace* outperforms the other techniques with a significant increase in social distancing accuracy over sensor [63], WiFiTrace [3], and SCT [7]. *CellTrace* achieves a 0.45-m median social distancing error, while the baseline has a median error of 1.5 m. This confirms the efficacy of *CellTrace* as a contact tracing technology.

[8]In particular, the effectiveness of the available privacy-first BLE apps might not be easy to evaluate due to the lack of ground-truth data [59].

VII. LIMITATIONS AND DISCUSSION

Although the feasibility of cellular networks as a base technology for reliable contact tracing, using them involves privacy concerns in some countries [48] (as discussed in Section V-E).

1) Cellular technology may have some privacy concerns in some countries [48], despite their feasibility, as discussed in Section V-E. This can be handled by anonymizing users' data, employing differential privacy [60], or inheriting the decentralization concept from other techniques (DP-3T) [40].
2) Fingerprinting approach is challenging in 3G and 4G networks due to the reduction of the available cell information, which only includes the associated serving cell and sometimes the strongest neighboring cells [64], [65]. However, this problem exists only at the client side mode [66], and some solutions have been proposed to mitigate its effect, e.g., [11].
3) Fingerprinting-based localization is expensive in terms of data collection and maintenance. However, some solutions have been proposed to mitigate this issue in cellular [11], [53], [67] and for Wi-Fi [26], which can be reused for cellular as well. It is worth mentioning that collecting site data is usually done by each provider to ensure the quality of service of their clients [67].

VIII. CONCLUSION

In this article, we aimed to realize a flexible solution for contact tracing that can be operated by the provider, client, or even at a third party by handling data from different sources, i.e., different providers. We presented the design, implementation, and evaluation of the *CellTrace* system as a ubiquitous contact and social distance tracing system using cellular signals. As part of the *CellTrace* design, we introduced a novel feature extraction module based on DeepCCA, which yields cross-provider features. These features are further utilized for training a deep localization model tracking users and calculating their social distance regardless of their connected providers. Furthermore, we showed how *CellTrace* includes provisions in the model to avoid overfitting and boost the model generalization ability. *CellTrace* achieved a promising localization and contact tracing performance of submeter median distance error and 97% accuracy, respectively. Nevertheless, *CellTrace* still has to handle privacy-associated issues to ensure effective contact tracing while maintaining privacy. In addition, we plan to study the system performance at scale, i.e., increasing the number of phones, users, providers, and so on.

REFERENCES

[1] L. Willem *et al.*, "The impact of contact tracing and household bubbles on deconfinement strategies for COVID-19," *Nature Commun.*, vol. 12, p. 1524, 2021, doi: 10.1038/s41467-021-21747-7.
[2] M. Tesar, "Towards a post-COVID-19 'new normality?': Physical and social distancing, the move to online and higher education," *Policy Futures Educ.*, vol. 18, no. 5, pp. 556–559, 2020.
[3] A. Trivedi, C. Zakaria, R. Balan, and P. Shenoy, "WiFiTrace: Network-based contact tracing for infectious diseases using passive WiFi sensing," *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 5, no. 1, pp. 1–26, 2021.

[4] X. Wang, L. Gao, S. Mao, and S. Pandey, "DeepFi: Deep learning for indoor fingerprinting using channel state information," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Mar. 2015, pp. 1666–1671.

[5] Y. Shu, C. Bo, G. Shen, C. Zhao, L. Li, and F. Zhao, "Magicol: Indoor localization using pervasive magnetic field and opportunistic WiFi sensing," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 7, pp. 1443–1457, Jul. 2015.

[6] M. Alzantot and M. Youssef, "UPTIME: Ubiquitous pedestrian tracking using mobile phones," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2012, pp. 3204–3209.

[7] P. C. Ng, P. Spachos, and K. N. Plataniotis, "COVID-19 and your smartphone: BLE-based smart contact tracing," *IEEE Syst. J.*, vol. 15, no. 4, pp. 5367–5378, Dec. 2021.

[8] P.-O. Dehaye, "Inferring distance from Bluetooth signal strength: A deep dive," Tech. Rep., 2020. [Online]. Available: https://medium.com/personaldata-io/inferring-distance-from-bluetooth-signal-strength-a-deep-dive-fe7badc2bb6d

[9] D. J. Leith and S. Farrell, "Coronavirus contact tracing: Evaluating the potential of using Bluetooth received signal strength for proximity detection," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 50, no. 4, pp. 66–74, 2020.

[10] H. Rizk, M. Torki, and M. Youssef, "CellinDeep: Robust and accurate cellular-based indoor localization via deep learning," *IEEE Sensors J.*, vol. 19, no. 6, pp. 2305–2312, Mar. 2019.

[11] H. Rizk and M. Youssef, "MonoDCell: A ubiquitous and low-overhead deep learning-based indoor localization with limited cellular information," in *Proc. 27th ACM SIGSPATIAL Int. Conf. Adv. Geograph. Inf. Syst.*, Nov. 2019.

[12] A. Shokry, M. Torki, and M. Youssef, "DeepLoc: A ubiquitous accurate and low-overhead outdoor cellular localization system," in *Proc. 26th ACM SIGSPATIAL Int. Conf. Adv. Geograph. Inf. Syst.*, Nov. 2018, pp. 339–348.

[13] Y. Tian, B. Denby, I. Ahriz, P. Roussel, and G. Dreyfus, "Robust indoor localization and tracking using GSM fingerprints," *EURASIP J. Wireless Commun. Netw.*, vol. 2015, no. 1, p. 157, Dec. 2015.

[14] H. Rizk, M. Abbas, and M. Youssef, "OmniCells: Cross-device cellular-based indoor location tracking using deep neural networks," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. (PerCom)*, Mar. 2020, pp. 1–10.

[15] H. Hotelling, "Relations between two sets of variates," in *Breakthroughs in Statistics*. New York, NY, USA: Springer, 1992, pp. 162–190.

[16] T. W. Anderson, *An Introduction to Multivariate Statistical Analysis* (Wiley Series in Probability and Mathematical Statistics), 1958.

[17] D. Hardoon, S. Szedmak, and J. Shawe-Taylor, "Canonical correlation analysis: An overview with application to learning methods," *Neural Comput.*, vol. 16, no. 12, pp. 2639–2664, Dec. 2004.

[18] G. Andrew, R. Arora, J. Bilmes, and K. Livescu, "Deep canonical correlation analysis," in *Proc. 30th Int. Conf. Mach. Learn.*, vol. 28, S. Dasgupta and D. McAllester, Eds., Atlanta, GA, USA, Jun. 2013, pp. 1247–1255.

[19] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan, and R. Sen, "Zee: Zero-effort crowdsourcing for indoor localization," in *Proc. 18th Annu. Int. Conf. Mobile Comput. Netw.*, 2012, pp. 293–304.

[20] Z. Sun, S. Pan, Y.-C. Su, and P. Zhang, "Headio: Zero-configured heading acquisition for indoor mobile devices through multimodal context sensing," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, Sep. 2013, pp. 33–42.

[21] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, and R. R. Choudhury, "No need to war-drive: Unsupervised indoor localization," in *Proc. 10th Int. Conf. Mobile Syst., Appl., Services*, 2012, pp. 197–210.

[22] H. Abdelnasser *et al.*, "SemanticSLAM: Using environment landmarks for unsupervised indoor localization," *IEEE Trans. Mobile Comput.*, vol. 15, no. 7, pp. 1770–1782, Jul. 2016.

[23] P. Bahl and V. N. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in *Proc. 19th Int. Conf. Comput. Commun. Soc.*, vol. 2, Mar. 2000, pp. 775–784.

[24] M. Youssef and A. Agrawala, "The Horus WLAN location determination system," in *Proc. 3rd Int. Conf. Mobile Syst., Appl., Services*, 2005, pp. 205–218.

[25] M. Abbas, M. Elhamshary, H. Rizk, M. Torki, and M. Youssef, "WiDeep: WiFi-based accurate and robust indoor localization system using deep learning," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. (PerCom*, Mar. 2019, pp. 1–10.

[26] H. Rizk, H. Yamaguchi, M. Youssef, and T. Higashino, "Gain without pain: Enabling fingerprinting-based indoor localization using tracking scanners," in *Proc. 28th Int. Conf. Adv. Geograph. Inf. Syst.*, Nov. 2020, pp. 550–559.

[27] P. Davidson and R. Piché, "A survey of selected indoor positioning methods for smartphones," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 2, pp. 1347–1370, 2nd Quart., 2017.

[28] A. Varshavsky, A. LaMarca, J. Hightower, and E. de Lara, "The SkyLoc floor localization system," in *Proc. 5th Annu. IEEE Int. Conf. Pervasive Comput. Commun. (PerCom)*, 2007, pp. 125–134.

[29] A. Sohail, Z. Ahmad, and I. Ali, "Analysis and measurement of Wi-Fi signals in indoor environment," *Int. J. Adv. Eng. Technol.*, vol. 6, no. 2, p. 678, 2013.

[30] T. K. Geok *et al.*, "Review of indoor positioning: Radio wave technology," *Appl. Sci.*, vol. 11, no. 1, p. 279, Dec. 2020.

[31] H. Rizk, M. Abbas, and M. Youssef, "Device-independent cellular-based indoor location tracking using deep learning," *Pervasive Mobile Comput.*, vol. 75, Aug. 2021, Art. no. 101420.

[32] M. Ibrahim and M. Youssef, "CellSense: An accurate energy-efficient GSM positioning system," *IEEE Trans. Veh. Technol.*, vol. 61, no. 1, pp. 286–296, Jan. 2012.

[33] M. Ibrahim and M. Youssef, "CellSense: A probabilistic RSSI-based GSM positioning system," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, Dec. 2010, pp. 1–5.

[34] H. Rizk, "Device-invariant cellular-based indoor localization system using deep learning," in *Proc. ACM MobiSys Rising Stars Forum (RisingStarsForum)*, New York, NY, USA, 2019, pp. 19–23.

[35] J. Budd *et al.*, "Digital technologies in the public-health response to COVID-19," *Nature Med.*, vol. 26, no. 8, pp. 1183–1192, 2020.

[36] C. T. Nguyen *et al.*, "A comprehensive survey of enabling and emerging technologies for social distancing—Part I: Fundamentals and enabling technologies," *IEEE Access*, vol. 8, pp. 153479–153507, 2020.

[37] (2020). *Tracetogether App COVID-19 Contact Tracing*. [Online]. Available: https://www.tracetogether.gov.sg/

[38] (2020). *COVIDSafe App*. [Online]. Available: https://www.health.gov.au/resources/apps-and-tools/covidsafe-app

[39] A. B. Dar, A. H. Lone, S. Zahoor, A. A. Khan, and R. Naaz, "Applicability of mobile contact tracing in fighting pandemic (COVID-19): Issues, challenges and solutions," *Comput. Sci. Rev.*, vol. 38, Nov. 2020, Art. no. 100307.

[40] C. Troncoso *et al.*, "Decentralized privacy-preserving proximity tracing," 2020, *arXiv:2005.12273*.

[41] *Exposure Notification Framework*, 2020. Accessed: 2022. [Online]. Available: https://developer.apple.com/documentation/exposurenotification

[42] *Exposure Notification API Launches to Support Public Health Agencies*, 2020. Accessed: 2022. [Online]. Available: https://blog.google/inside-google/company-announcements/apple-google-exposure-notification-api-launches/

[43] *Publicly-Available Exposure Notifications Apps*. Accessed: 2022. [Online]. Available: https://developers.google.com/android/exposure-notifications/apps

[44] K. A. Nguyen, Z. Luo, and C. Watkins, "Epidemic contact tracing with smartphone sensors," *J. Location Based Services*, vol. 14, no. 2, pp. 92–128, Apr. 2020.

[45] K. A. Nguyen, C. Watkins, and Z. Luo, "Co-location epidemic tracking on London public transports using low power mobile magnetometer," in *Proc. Int. Conf. Indoor Positioning Indoor Navigat. (IPIN)*, Sep. 2017, pp. 1–8.

[46] G. Li, S. Hu, S. Zhong, W. L. Tsui, and S.-H. G. Chan, "VContact: Private WiFi-based contact tracing with virus lifespan," 2020, *arXiv:2009.05944*.

[47] K. A. Nguyen, Z. Luo, and C. Watkins, "On the feasibility of using two mobile phones and WLAN signal to detect co-location of two users for epidemic prediction," in *Progress in Location-Based Services 2014*. Cham, Switzerland: Springer, 2015, pp. 63–78.

[48] *Israeli Government Reauthorizes Israeli Security Agency Surveillance Owing to COVID-19 Omicron Variant*, 2021. Accessed: 2022. [Online]. Available: https://www.pearlcohen.com/isa-covid-19-surveillance-reauthorized-omicron-variant/

[49] J. Kurose and K. Ross, *Computer Networks: A Top Down Approach Featuring the Internet*. Reading, MA, USA: Addison-Wesley, 2010.

[50] A. Parmar and K. M. Pattani, "Sniffing GSM traffic using RTL-SDR and Kali Linux OS," *Int. Res. J. Eng. Technol.*, vol. 4, no. 1, pp. 1637–1642, 2017.

[51] A. Tudzarov and T. Janevski, "M-RATS: Mobile-based radio access technology selector for heterogeneous wireless environment," in *Proc. 18th Telecommun. Forum (TELFOR)*, 2010, pp. 1–4.

[52] Y. Kang, K.-T. Lee, J. Eun, S. E. Park, and S. Choi, "Stacked denoising autoencoders for face pose normalization," in *Proc. Int. Conf. Neural Inf. Process.* Daegu, South Korea: Springer, 2013, pp. 241–248.

[53] H. Rizk, A. Shokry, and M. Youssef, "Effectiveness of data augmentation in cellular-based localization using deep learning," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2019, pp. 1–6.

[54] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proc. 25th Int. Conf. Mach. Learn. (ICML)*. New York, NY, USA: Association for Computing Machinery, 2008, pp. 1096–1103.
[55] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Math. Control, Signals Syst.*, vol. 2, no. 4, pp. 303–314, 1989.
[56] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. 14th Int. Conf. Artif. Intell. Statist.*, 2011, pp. 315–323.
[57] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
[58] Y. Bengio, "Practical recommendations for gradient-based training of deep architectures," in *Neural Networks: Tricks of the Trade*. Berlin, Germany: Springer, 2012, pp. 437–478.
[59] E. Seto, P. Challa, and P. Ware, "Adoption of COVID-19 contact tracing apps: A balance between privacy and effectiveness," *J. Med. Internet Res.*, vol. 23, no. 3, Mar. 2021, Art. no. e25726.
[60] C. Dwork, "Differential privacy: A survey of results," in *Proc. Int. Conf. Theory Appl. Models Comput.* Xi'an, China: Springer, 2008, pp. 1–19.
[61] M. Abadi *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous distributed systems," 2016, *arXiv:1603.04467*.
[62] C. Sammut and G. I. Webb, Eds., *Leave-One-Out Cross-Validation*. Boston, MA, USA: Springer, 2010, pp. 600–601.
[63] P. Tu, J. Li, H. Wang, K. Wang, and Y. Yuan, "Epidemic contact tracing with campus WiFi network and smartphone-based pedestrian dead reckoning," *IEEE Sensors J.*, vol. 21, no. 17, pp. 19255–19267, Sep. 2021.
[64] H. Zang, F. Baccelli, and J. Bolot, "Bayesian inference for localization in cellular networks," in *Proc. IEEE INFOCOM*, Mar. 2010, pp. 1–9.
[65] A. Ray, S. Deb, and P. Monogioudis, "Localization of LTE measurement records with missing information," in *Proc. 35th Annu. IEEE Int. Conf. Comput. Commun. (IEEE INFOCOM)*, Apr. 2016, pp. 1–9.
[66] A. Mohamed, M. Tharwat, M. Magdy, T. Abubakr, O. Nasr, and M. Youssef, "DeepFeat: Robust large-scale multi-features outdoor localization in LTE networks using deep learning," *IEEE Access*, vol. 10, pp. 3400–3414, 2022.
[67] J. Boulis, M. Hemdan, A. Shokry, and M. Youssef, "Data augmentation using GANs for deep learning-based localization systems," in *Proc. 29th Int. Conf. Adv. Geograph. Inf. Syst.*, Nov. 2021, pp. 672–673.

Hamada Rizk (Senior Member, IEEE) received the M.E. degree in computer science and engineering from Tanta University, Tanta, Egypt, in 2016, and the Ph.D. degree in computer science and engineering from E-JUST in 2020.

He is with Tanta University and Osaka University, Suita, Japan. He has been involved in several projects funded by many academic and industrial organizations, such as NTRA, Egypt, Uber, USA, ASTEP JST, Kakenhi JSPS, and NVIDIA, Japan. He has been working in the mobile and pervasive computing, spatial intelligence, and AI research areas.

Dr. Rizk was a recipient of the Silver Medal in the 4th ACM SigSpatial Competition held in Chicago in 2019. He has been honored as an Outstanding Young Researcher by the HLF foundation in Germany in 2019 and Google in 2019 and 2020, among others.

Asmaa Saeed received the B.E. degree in computer and control engineering from Tanta University, Tanta, Egypt, in 2021.

She is currently a Student Fellow with the Information Technology Institute (ITI), Giza, Egypt. Her research interests include artificial intelligence.

Hirozumi Yamaguchi (Member, IEEE) received the B.E., M.E., and Ph.D. degrees in information and computer science from Osaka University, Osaka, Japan, in 1994, 1996, and 1998, respectively.

He is currently a Professor with Osaka University and leads the Mobile Computing Laboratory. He has been working in the mobile and pervasive computing and networking research areas. He has published papers in top-quality journals and conferences.

Dr. Yamaguchi was awarded the Commendation for Science and Technology by the Minister of Education, Culture, Sports, Science and Technology in 2018. He has also served on International Conference on Distributed Computing and Networking (ICDCN) 2021 and Mobiquitous 2021 as the General Chair and on many mobile and pervasive conferences, such as IEEE International Conference on Pervasive Computing and Communications (PerCom), as the TPC Chair and a member. He has served on the Editorial Board of *Ad Hoc Networks* (Elsevier) journal and *Journal of Reliable Intelligent Environments* (Springer).