GRAFICS: Graph Embedding-based Floor Identification Using Crowdsourced RF Signals

Weipeng Zhuo*, Ziqi Zhao*, Ka Ho Chiu*, Shiju Li‡, Sangtae Ha†, Chul-Ho Lee‡,§, and S.-H. Gary Chan*,§

*The Hong Kong University of Science and Technology
†University of Colorado Boulder, ‡Texas State University
Email: *{wzhuo,zzhaoas,khchiuac,gchan}@ust.hk, †sangtae.ha@colorado.edu, ‡{shiju.li,chulho.lee}@txstate.edu

Abstract—We study the problem of floor identification for radiofrequency (RF) signal samples obtained in a crowdsourced manner, where the signal samples are highly heterogeneous and most samples lack their floor labels. We propose GRAFICS, a graph embedding-based floor identification system. GRAFICS first builds a highly versatile bipartite graph model, having APs on one side and signal samples on the other. GRAFICS then learns the low-dimensional embeddings of signal samples via a novel graph embedding algorithm named E-LINE. GRAFICS finally clusters the node embeddings along with the embeddings of a few labeled samples through a proximity-based hierarchical clustering, which eases the floor identification of every new sample. We validate the effectiveness of GRAFICS based on two large-scale datasets that contain RF signal records from 204 buildings in Hangzhou, China, and five buildings in Hong Kong. Our experiment results show that GRAFICS achieves highly accurate prediction performance with only a few labeled samples (96% in both micro- and macro-F scores) and significantly outperforms several state-of-the-art algorithms (by about 45% improvement in micro-F score and 53% in macro-F score).

I. INTRODUCTION

While there are plenty of smart city applications based upon radiofrequency (RF) signals, e.g., WiFi, iBeacon and UWB, their success largely depends on the availability of *floor information* associated with the RF signals. For example, to facilitate pedestrian navigation in multi-floor indoor environments, RF signal-based localization systems [1]–[4] need to estimate the floor number accurately before localizing people in a two-dimensional plane. Similar situations happen with indoor unmanned aerial vehicles for scene construction or building quality monitoring [5], [6]. In geofencing for elderly care or pandemic control [7], [8], RF signals are leveraged to assure that people stay on a *certain* floor due to the prevalence of the RF signals. Even in robot rescue [9], RF signals can be used in inferring the floor information when the visual information is lost.

On the other hand, crowdsourcing has emerged as a practically viable solution for the large-scale collection of data, which is crucial for the successful deployment of the above applications in practice. However, the participatory nature of crowdsourcing often makes the collected dataset incomplete. In other words, the crowdsourced RF measurement dataset
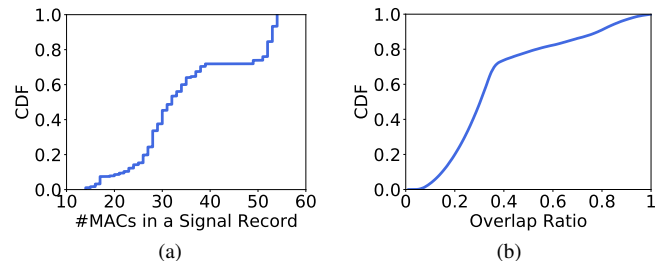
Fig. 1. Statistics of RF signal records on a floor. (a) CDF of the number of MACs in a signal record; (b) CDF of the fraction of common MACs (overlap ratio) for a pair of signal records.

contains many measurement samples lacking the exact floor information, i.e., which floor RF signals were measured. While the activities such as in-shop contactless shopping [10] and QR code check-ins [11] enable the collection of 'floor-labeled' RF signals (or labeled RF signals on each floor), their portion is still relatively *small* in the entire dataset. To sum up, there is a need for an accurate floor identification system based on RF signals, only a few of which have their floor information.

It is, however, non-trivial to build such a floor identification system due to the following technical challenges. In the crowdsourced data collection, different people would contribute different measurements of RF signals, each of which is in the form of a vector of the pairs of detected access points (APs), more precisely, detected medium access control addresses (or MACs for short), and their corresponding received signal strength (RSS) values. The dataset is, however, *heterogeneous*. The MACs observed in one sample may not appear in another. The MACs and their corresponding RSS values in the samples may also vary even when they are measured on the same floor or the same spot. This can be worse due to environmental changes (e.g., installation and removal of APs/MACs) and device heterogeneity.

To illustrate the heterogeneity in the crowdsourced RF measurement dataset, in Figure 1, we show the statistics of 8,274 signal records collected on a floor of a mall, where there are 805 distinct MACs. Figure 1(a) presents the CDF of the number of MACs in each record. Most records contain less than 40 MACs. It indicates that each record only contains a small fraction of the MACs detectable on the floor. In addition, for any pair of signal records, we compute their overlap ratio, defined as the intersection over the union of MACs in the records, and plot the CDF of the overlap ratio in Figure 1(b).

Most pairs (78%) overlap less than half. Hence, it would be *inefficient* to use the dataset *as is*.

In this paper, we present GRAFICS, **gra**ph embedding-based **f**loor **i**dentification using **c**rowdsourced RF **s**ignals, which infers the floor number a user is located from an online measurement of RF signals, along with the crowdsourced RF measurement dataset. While this crowdsourced dataset can be leveraged to build a learning model for the problem, it needs to be judiciously used due to the aforementioned technical challenges. Since the floor-labeled RF measurement samples are scarce in the dataset, supervised learning models are deemed impractical, despite that the supervised learning models have been used for the floor identification [12]–[14]. It also remains questionable how to extract useful information from the highly heterogeneous dataset. Thus, GRAFICS is designed to be a semi-supervised learning-based system, which is built upon graph embedding and proximity-based hierarchical clustering.

GRAFICS first constructs *a bipartite graph* reflecting the observation in each measurement sample while being adaptive to the arrivals (or departures) of new (or old) MACs and arrivals of new measurements. In this graph, we use a node of one type to represent a MAC and a node of another type to indicate a measurement record itself. Since each measurement record has a list of observed MACs and their corresponding RSS values, we create edges between an 'RF-record node' that represents a record and 'MAC nodes' that correspond to the MACs observed in the record. We then assign weight values to the edges based on the RSS values. The connectivity structure in the graph captures the relevance between measurement records while reflecting the information in each record.

To cope with the heterogeneity in the RF samples, GRAFICS next learns *a low-dimensional vector representation* (or embedding) of each node in the graph. As a byproduct, GRAFICS prevents the so-called missing value problem in the first place, which arises when the measurement samples are directly used in a matrix form [15], [16] (see Section II for more details), since the node embeddings of equal length are used on behalf of the measurement samples whose dimensions can be different. To obtain the node embeddings, we adopt LINE [17], yet with our own refinements.

While the original LINE algorithm learns node embeddings based on the relationship between nodes via their common neighbors, we extend the algorithm to learn the embeddings to capture the relationship between nodes based on not only their common 'direct' neighbors but also other common 'multi-hop' neighbors. Our extension of LINE, named E-LINE, is better suited for learning the node embeddings of our bipartite graph for floor classification. For example, two RF measurement records, even though they are taken on the same floor, may not share many common MACs but contain the MACs that rather overlap with the ones in the other records collected on the same floor. We empirically demonstrate that GRAFICS (with E-LINE) indeed outperforms its version with the LINE algorithm for the floor classification problem (see Section VI).

Once the node embeddings are obtained, GRAFICS uses a proximity-based hierarchical clustering to cluster measurement samples in the embedding space so that each cluster forms around each floor-labeled sample, i.e., an RF measurement sample that comes with its floor information. Thanks to graph embedding and hierarchical clustering, GRAFICS is able to work with *much fewer* floor-labeled samples, when compared to the prior supervised learning-based solutions [12]–[14] (see Section VI). Finally, whenever there is an *online* measurement of RF signals for *inference* (i.e., identifying which floor the measurement is taken), its embedding is first obtained through the graph embedding, and its nearest cluster is then identified based on its distance to the centroid of each cluster in the embedding space. The floor associated with the nearest cluster eventually becomes the inference result.

Our contributions can be summarized as follows.

- *GRAFICS is capable of handling the heterogeneity in the RF samples.* The construction of a bipartite graph incorporates both the information in each RF measurement record and the relationship between the records. In addition, the graph is easily extendable for new RF records. It can also be adjusted to reflect installation and removal of APs.

- *GRAFICS learns the representations of RF samples.* Our novel graph embedding algorithm E-LINE maps each node in the bipartite graph onto the same low-dimensional vector space, which enables a better identification of similarities and differences between the heterogeneous RF samples. E-LINE also produces *better* representations or node embeddings in improving the performance of floor identification, as compared to LINE.

- *GRAFICS requires a small amount of labeled data.* The node embeddings are used to cluster RF samples (including few floor-labeled samples) in the embedding space via the hierarchical clustering. This semi-supervised learning makes GRAFICS require *much fewer* floor-labeled samples than the supervised ones. This efficiency stems from the *enhanced* vector representations of the RF samples.

- *GRAFICS significantly outperforms state-of-the-art methods.* We implement GRAFICS and evaluate its performance based on Microsoft's Kaggle open dataset, which includes crowdsourced RF samples obtained in 204 multi-floor buildings in Hangzhou, China, and a newly collected dataset, which has RF samples obtained in 5 buildings in Hong Kong. Experimental results show that GRAFICS achieves higher than 96% in micro-F and macro-F scores across all buildings with only few labels per floor (around 4 out of 1000 on average). Compared with state-of-the-art methods, GRAFICS also outperforms around 45% in micro-F score and 53% in macro-F score.

The rest of this paper is organized as follows. We review related work in Section II and provide an overview of GRAFICS in Section III. In Section IV, we explain the details of bipartite graph modeling, graph embedding, and hierarchical clustering in GRAFICS. We then elaborate on how the online inference is done in GRAFICS in Section V. Extensive experimental results are presented in Section VI. We conclude in Section VII.

II. RELATED WORK

Floor classification with different sensors: Different sensors from smartphones have been leveraged for the floor classification problem [18]–[20]. A heuristic thresholding method is used with barometer readings [21] to detect floor transitions. However, such a threshold is difficult to set in practice since different barometer models may have different sensitivity levels. Thus, in [22], the barometer readings are fused with WiFi and inertial measurement unit (IMU) sensors using Kalman filter to infer floor information. Magnetometers from smartphones are also explored in [23] with IMU readings for floor classification. Despite their satisfactory performance, the collection of such sensor readings introduces high overhead for data storage, data transmission, and battery consumption. Furthermore, users are supposed to follow carefully designed trajectories as ground truth [24] to collect sensor data such that they can be well 'labeled'. In contrast, GRAFICS is *lightweight* and based only on *crowdsourced RF measurements* to achieve accurate and efficient detection of floor numbers.

The need for massive labeled data: Several machine learning algorithms have been utilized for floor classification [12], [13], [25]–[28]. For example, a support vector machine-based method is proposed in [12], but it needs to train support vectors for the classification of every pair of floors, which is inconvenient in practice. To address this problem, a recurrent neural network approach is introduced in [13]. It works well when data has a temporal relationship, which implies that one would need to obtain a group of RF samples collected from the same trajectory. However, crowdsourced RF samples are usually sporadic and contributed by many different users. Furthermore, all these machine learning-based solutions require a large amount of labeled data in training their classification model. In contrast, GRAFICS works with only *few* labeled samples.

Leveraging the locations of APs: There are also a few prior studies [25], [27], [29] that leverage the knowledge of APs' locations for floor classification. For instance, ViFi [29] learns the parameters of a signal propagation model from the RSS measurements to generate virtual reference points and then predicts floor labels for new signals with a weighted k-nearest neighbors algorithm. StoryTeller [27] converts RF signals to images based on APs with strong signal strengths and then trains a convolutional neural network (CNN) model for floor classification. They both require the APs' locations, which are, however, usually unavailable when the RF measurement data is collected in a crowdsourced manner. Thus, GRAFICS is designed to be *independent* of the AP locations and only uses the signal readings collected from different APs to infer floor information accurately.

Assuming fixed-length RF signal vectors: When using RF signals for floor classification, the state-of-the-art systems [14]–[16], [30] first group all the RF signal data together in a matrix form. Each RF signal measurement record contains a set of RSS values from surrounding APs that are represented by their MAC addresses, so the RSS-value set forms a row of
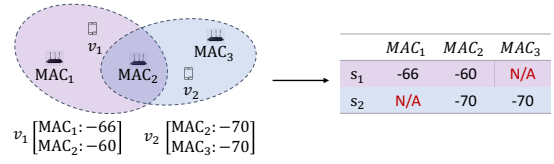

Fig. 2. Missing value problem when representing RF signal data in a matrix.

the matrix. In other words, each sensed MAC address forms a column where the RSS value associated with the MAC address in each record is the value of an entry. Once the matrix is constructed, they transform each row in the matrix to a low-dimensional vector space via a CNN or an autoencoder, and build a classification model for floor identification.

In the matrix representation, however, there is a so-called missing value problem in that some entries in the matrix lack RSS values since the RF samples are of variable length (i.e., the number of detected MAC addresses can be different per sample), as shown in Figure 2. The missing entries are generally filled with extremely small values to indicate non-availability. Nonetheless, some missing values may be due to the limited scanning capability of low-end devices. The entries with the extremely small values could also be misinterpreted as the presence of their corresponding APs, albeit with weak signals, when used for building a classifier. Thus, such ad-hoc data imputation could potentially lead to erroneous feature extraction, resulting in unsatisfactory floor classification and learning inefficiency. In contrast, GRAFICS models RF signal records using a bipartite graph, which naturally *avoids* the missing value problem.

III. GRAFICS: DESIGN

We first discuss the technical challenges that are considered and addressed in designing our floor identification system GRAFICS. We then provide a system overview before we go into the details in the subsequent sections.

A. Technical Challenges and Considerations

Dynamic RF environments: Crowdsourced RF signal samples consist of scattered, noisy RSS values and are of variable sizes. For example, the signal samples collected at two different places yet on the same floor may have no overlap with each other because of the limited coverage of APs. Even on the same spot, the recorded RSS values may differ and detected APs may vary due to environmental changes and device heterogeneity. The RSS records could also vary over time, and APs could be added and removed over time. To address these issues, we propose a novel representation of the scattered, noisy RF signal samples based on a bipartite graph, which allows us to uncover the hidden relationship among the RF samples. We also propose E-LINE that improves on LINE in a way that is better suitable for our bipartite graph to map each node (an RF sample) in the graph into a low-dimensional vector space. Since the RF samples (of variable length) are now represented in the same embedding space, their similarities and differences can be easily measured via the distances among the samples in the embedding space.
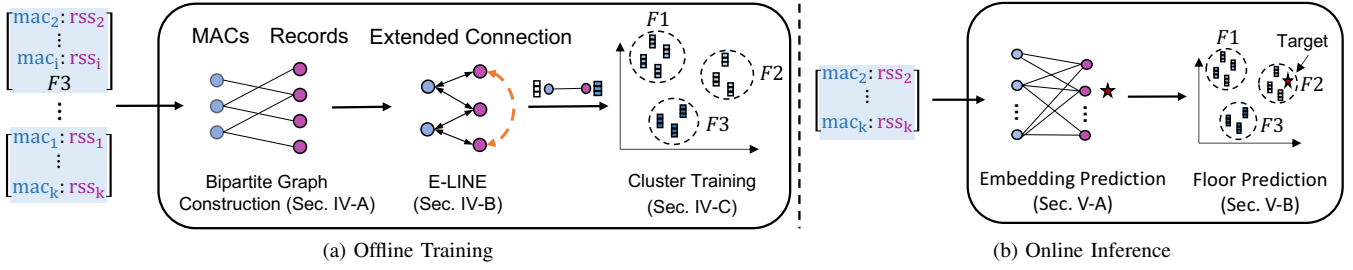
Fig. 3. A system overview of GRAFICS.

Scarce floor-labeled data: Crowdsourcing enables a large-scale collection of RF signal data but does not ensure the collection of their contextual information such as which floor they were obtained. This floor information is the *label* information in the floor identification problem. While we can still collect the floor-labeled RF samples from the events such as in-shop contactless shopping [10] and QR code check-ins [11], their portion in the dataset is largely limited. In other words, not every RF sample is associated with its floor label, and only a very small portion of the RF data has the floor labels. Thus, after obtaining the embeddings of the RF samples, we do not build any supervised learning model for floor classification, which often requires a considerable amount of labeled data. Instead, thanks to the ease of computing the distances among the samples via their embeddings, we are able to build an effective semi-supervised learning model with only a tiny fraction of labeled data, which is done via our proposed proximity-based hierarchical clustering.

B. System Overview

Figure 3 illustrates an end-to-end workflow of GRAFICS, which ranges from offline training to online inference. In the offline training phase, GRAFICS first models RF signal measurement samples using a bipartite graph, inspired by their structure where each RF signal sample consists of sensed MACs with corresponding RSS values (Section IV-A). In this bipartite graph, the RF signal samples and sensed MACs are represented by two different types of nodes, respectively, and they are connected by edges weighted by the RSS values from the sensed MACs (or APs). As a result, changes in the MACs or RF signal samples can be easily captured by adding or removing the nodes or edges in the graph. Then, the bipartite graph is processed with our proposed embedding algorithm E-LINE (Section IV-B) to learn low dimensional coordinates, i.e., a latent representation or embedding for each node in an unsupervised manner. Finally, the learned representations, together with the small set of labels on each floor, are used to build a simple yet effective floor classifier via our proximity-based hierarchical clustering (Section IV-C). During this process, all unlabeled crowdsourced signal samples form clusters, each of which also contains a floor-labeled sample, so their labels are virtually predicted as the label of the floor-labeled sample in the cluster that they belong to.

On the other hand, in the online *inference* phase, for each newly collected RF sample, GRAFICS first generates its embedding through the graph embedding (Section V-A) and then predicts its label (or infer which floor the new sample is taken) as the label (or floor) of the cluster that is closest in the embedding space (Section V-B).

IV. GRAFICS: OFFLINE TRAINING

The offline training of GRAFICS is done via the following three steps: (i) constructing a bipartite graph based on the RF signal samples, (ii) learning node embeddings from the graph by our E-LINE algorithm, and (iii) building a simple yet effective classification model via our proximity-based hierarchical clustering method.

A. Bipartite Graph Construction

Conventionally, RF signal data are represented in a matrix form (or vectors of equal length) for indoor localization and floor identification applications [14]–[16], [30]–[32]. Such a matrix representation, however, often suffers from the missing value problem, as explained in Section II. Thus, we propose to model the variable-length RF signal records as *a weighted bipartite graph*, where the measured signal information is preserved without having the missing value problem. In the graph, each RF signal record is a node of one type and the sensed MAC addresses in each record are nodes of the other type. Edges indicate the presence of sensed MAC addresses in each RF record, and their edge weights are determined based on the RSS values from the corresponding APs.

Let $\mathcal{G} = (\mathcal{M}, \mathcal{V}, \mathcal{E})$ be a weighted bipartite graph. $\mathcal{V}$ is a set of nodes to represent the RF signal records, and $\mathcal{M}$ is a set of nodes for the sensed MAC addresses. Also, $\mathcal{E}$ is a set of edges, where $e_{mv} \in \mathcal{E}$ denotes the edge between $m \in \mathcal{M}$ and $v \in \mathcal{V}$. Note that the edge e_{mv} indicates the presence of MAC address m in record v. For each record v, let RSS_{mv} be the RSS value from an AP whose MAC address is m, which appears in the record. We then associate each edge e_{mv} with weight c_{mv}, which is defined as

$$c_{mv} := f(\text{RSS}_{mv}), \tag{1}$$

where f is a function of RSS_{mv} with $f(\text{RSS}_{mv}) > 0$ for all RSS_{mv}. Note that two nodes m and v are connected as long as there exists a measured value of RSS_{mv} from MAC m in record v, with the edge weight being c_{mv}. Figure 4 illustrates an example where signal record v_1 is connected to MACs 1–2, and v_2 is connected to MACs 2–3.
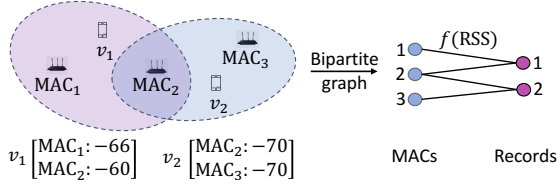
Fig. 4. Example of two signal records with three sensed MACs.

We use the following weight function for graph $\mathcal{G}$:

$$f(\text{RSS}_{mv}) := \text{RSS}_{mv} + \alpha, \qquad (2)$$

with some constant α that satisfies $\alpha > \max\{|\text{RSS}_{mv}|, \forall m, v\}$. In Section VI-D, we empirically show that it achieves the best performance, implying that a constant offset preserves the relationships among nodes better than other schemes.

B. E-LINE

From the bipartite graph $\mathcal{G}$, we next obtain vector representations or embeddings of nodes to be used to build a classifier for floor identification. To this end, we propose a novel graph embedding algorithm E-LINE, which improves on LINE [17] in learning better vector representations of equal length from the bipartite graph $\mathcal{G}$. Among others, LINE is a widely adopted graph embedding algorithm in the literature [33]. Note that graph embedding here is representation learning when each node in a graph is not associated with a set of features (or attributes), and the embedding vectors are learned mainly based on the underlying connectivity structure of a given graph, unlike graph neural networks.

Before going into the details of E-LINE, we below briefly explain LINE since E-LINE is an extension of LINE. LINE considers the following two factors to learn node embeddings from a given graph: (i) whether two nodes are connected, and (ii) whether two nodes share common neighbors. These two factors lead to two different notions of 'proximity', which are first-order proximity and second-order proximity, respectively. We observe that the first-order proximity is no longer useful when LINE is used for a bipartite graph. The edges only exist between the nodes of different types, but we are more interested in the relationships among the nodes of the same type. Our experiment also confirms that LINE performs better with the second-order proximity only than the one using both proximities in learning node embeddings from the bipartite graph for the floor identification problem. Thus, we below focus on the second-order proximity of LINE and explain how it is extended in E-LINE.

The second-order proximity in LINE is defined for a *directed* graph. Note that any undirected graph can be interpreted as a directed graph by replacing each (undirected) edge with two directed edges having the opposite directions. The weight for each undirected edge is also used as the weights for both directed edges, i.e., $c_{ij} = c_{ji}$. The rationale behind the second-order proximity is to make nodes i and j close to each other in the embedding space, if they share many common neighbors, i.e., $|N(i) \cap N(j)|$ is large, where $N(i)$ and $N(j)$ denote the sets of the neighbors of i and j, respectively.

To be precise, there are two different embeddings for each node i. One is 'ego' embedding, defined as $\boldsymbol{u}_i$, to represent the node itself, and it is the representation of each node to use in building a floor classifier. The other is 'context' embedding, defined as $\boldsymbol{u}'_i$, to represent the relationship between node i and other nodes, or encode its (one-hop) neighborhood information. The second-order proximity between i and j is determined by the similarity between their *context* embeddings $\boldsymbol{u}'_i$ and $\boldsymbol{u}'_j$. If two nodes share many common neighbors, their *context* embeddings are similar to each other, so are their *ego* embeddings. Here the embedding dimensions are the same and determined as a hyperparameter.

The second-order proximity is defined based on the conditional probability of the *context* of j determined given node i, which is given by

$$\Pr(\boldsymbol{u}'_j | \boldsymbol{u}_i) := \frac{\exp(\boldsymbol{u}'_j \cdot \boldsymbol{u}_i)}{\sum_{l \in \mathcal{M} \cup \mathcal{V}} \exp(\boldsymbol{u}'_l \cdot \boldsymbol{u}_i)} , \qquad (3)$$

where $\boldsymbol{a} \cdot \boldsymbol{b}$ is the inner product between $\boldsymbol{a}$ and $\boldsymbol{b}$. Its empirical probability is also defined as

$$\hat{\Pr}(\boldsymbol{u}'_j | \boldsymbol{u}_i) := \frac{c_{ij}}{\sum_{l \in N(i)} c_{il}} , \qquad (4)$$

which characterizes the 'influence' from node i to the *context* of its neighbor j. Then, the ego and context embeddings are determined so that the following objective function is minimized:

$$\mathcal{O}_1 = \sum_{i \in \mathcal{M} \cup \mathcal{V}} \lambda_i \sum_{j \in N(i)} \hat{\Pr}(\boldsymbol{u}'_j | \boldsymbol{u}_i) \log \frac{\hat{\Pr}(\boldsymbol{u}'_j | \boldsymbol{u}_i)}{\Pr(\boldsymbol{u}'_j | \boldsymbol{u}_i)},$$

which is defined based on the Kullback–Leibler divergence of $\Pr(\boldsymbol{u}'_j | \boldsymbol{u}_i)$ from $\hat{\Pr}(\boldsymbol{u}'_j | \boldsymbol{u}_i)$ with the convention $0 \log 0 = 0$. After omitting some constant terms, the objective function can be written as

$$\mathcal{O}_1 = - \sum_{i \in \mathcal{M} \cup \mathcal{V}} \sum_{j \in N(i)} \lambda_i \frac{c_{ij}}{\sum_{l \in N(i)} c_{il}} \log \Pr(\boldsymbol{u}'_j | \boldsymbol{u}_i).$$

By setting $\lambda_i := \sum_{l \in N(i)} c_{il}$, we have

$$\mathcal{O}_1 = - \sum_{i \in \mathcal{M} \cup \mathcal{V}} \sum_{j \in N(i)} c_{ij} \log \Pr(\boldsymbol{u}'_j | \boldsymbol{u}_i). \qquad (5)$$

We see that the second-order proximity mainly captures the closeness between nodes in the (ego) *embedding* space via the similarity between their context embeddings. Since each context embedding encodes the one-hop neighborhood information of each node, their similarity indicates how many one-hop neighbors they commonly share. However, it is not able to capture the relevance between nodes when they do not share many neighbors in their one-hop neighborhood but in their local yet few-hop neighborhood. In other words, the context embeddings $\boldsymbol{u}'_i$ and $\boldsymbol{u}'_j$ of nodes i and j should also be similar to each other if they are reachable from each other in a few hops via multiple different local paths. This is particularly important in our floor identification problem. For example, two RF measurement samples taken on the same floor may not
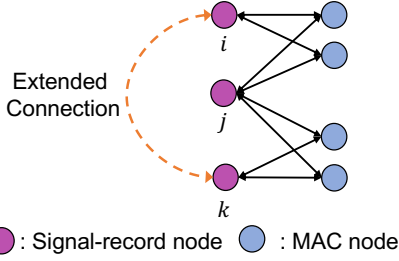
Fig. 5. E-LINE incorporates indirect relationships between nodes into graph embedding.

share many common MACs but contain the MACs that rather overlap with the ones in the other samples collected on the same floor. While the second-order proximity cannot capture such a case, their context embeddings should be similar to each other so that they can be mapped closely on the (ego) embedding space.

To achieve better node embeddings, we below introduce a novel extension of LINE, or E-LINE. Similar to (3), we define the conditional probability of the ego (node) j influenced given the context of i, which is given by

$$\Pr(\boldsymbol{u}_j|\boldsymbol{u}_i') = \frac{\exp(\boldsymbol{u}_j \cdot \boldsymbol{u}_i')}{\sum_{l \in \mathcal{M} \cup \mathcal{V}} \exp(\boldsymbol{u}_l \cdot \boldsymbol{u}_i')} \ . \quad (6)$$

Its empirical distribution is also defined as

$$\hat{\Pr}(\boldsymbol{u}_j|\boldsymbol{u}_i') = \frac{c_{ij}}{\sum_{l \in N(i)} c_{il}}, \quad (7)$$

which corresponds to (4). Following the same line of argument as above, we can construct the following objective function:

$$\mathcal{O}_2 = -\sum_{i \in \mathcal{M} \cup \mathcal{V}} \sum_{j \in N(i)} c_{ij} \log \Pr(\boldsymbol{u}_j|\boldsymbol{u}_i'). \quad (8)$$

Finally, we learn the ego and context embeddings by minimizing the following objective function, which is a combination of the ones in (5) and (8):

$$\mathcal{O}_3 = \mathcal{O}_1 + \mathcal{O}_2$$
$$= -\sum_{i \in \mathcal{M} \cup \mathcal{V}} \sum_{j \in N(i)} c_{ij} \left(\log \Pr(\boldsymbol{u}_j'|\boldsymbol{u}_i) + \log \Pr(\boldsymbol{u}_j|\boldsymbol{u}_i')\right). \quad (9)$$

To summarize, E-LINE enhances LINE in such a way that nodes are mapped into the embedding space based on the similarity between their context embeddings that encode their *local* neighborhood information, which is now more than just their one-hop neighborhood information. See Figure 5 for illustration, where nodes i and k would also be close to each other in the embedding space.

On the other hand, minimizing the objective function in (9) is computationally expensive in practice since the normalizing constants of $\Pr(\boldsymbol{u}_j'|\boldsymbol{u}_i)$ and $\Pr(\boldsymbol{u}_j|\boldsymbol{u}_i')$ involve the summations over the entire node set $\mathcal{M} \cup \mathcal{V}$. Thus, instead of the objective function in (9), we consider its relevant objective function, which is based on the widely used 'negative sampling' method (see [17], [33], [34] and references therein) and is computationally much easier to evaluate. Specifically, we use

the following objective function to minimize in learning the embeddings of each node:

$$\mathcal{L}_{\mathcal{G}} := -\sum_{i \in \mathcal{M} \cup \mathcal{V}} \sum_{j \in N(i)} c_{ij} \Big(\log \left[\sigma(\boldsymbol{u}_j' \cdot \boldsymbol{u}_i)\sigma(\boldsymbol{u}_j \cdot \boldsymbol{u}_i') \right]$$
$$+ K \mathbb{E}_{z \sim \Pr(z)} \left[\log \left(\sigma(-\boldsymbol{u}_z' \cdot \boldsymbol{u}_i)\sigma(-\boldsymbol{u}_z \cdot \boldsymbol{u}_i') \right) \right] \Big), \quad (10)$$

where $\sigma(x) := 1/(1 + \exp(-x))$. The second term in each summand is based on K 'negative' samples. The expectation $\mathbb{E}$ is with respect to node z that is randomly drawn according to a probability distribution $\Pr(z)$, $z \in \mathcal{M} \cup \mathcal{V}$, and is computed via the Monte Carlo approximation with K negative samples. As widely used in the literature [17], [33], [34], we choose $\Pr(z) \propto d_z^{3/4}$, where d_z is the degree of node z.

Recall that the goal here is to make nodes close to each other in the (ego) embedding space if they share many common local neighbors. The rationale behind minimizing the objective function in (10) is as follows. On one hand, the first term in each summand makes the ego (or context) embedding of i and the context (or ego) embedding of j similar to each other for every pair of neighboring nodes i and j. We can also see that due to the presence of the context embedding of each node, the ego embeddings of nodes that are in their local neighborhood would eventually be similar to each other. On the other hand, the second term in each summand, which is based on K negative samples, makes the ego (or context) embedding of i and the context (or ego) embedding of z *dissimilar* to each other. Here node z is most likely to be a node *outside* the local neighborhood of i. Therefore, the ego embeddings of locally closed nodes would be similar to each other, while the ones of distant nodes would be dissimilar to each other.

Figure 6 visualizes how effective E-LINE maps RF signal samples collected in a three-story campus building into the (ego) embedding space, as compared to multidimensional scaling (MDS) and autoencoder used with a matrix representation of the RF signal samples. Here all the samples are 'floor-labeled', and we use a visualization tool called t-SNE [35]. As shown in Figure 6(a), E-LINE makes the embeddings of the signals collected on the same floor naturally form a cluster while separating different clusters apart. However, as can be seen from Figure 6(b)–(c), MDS [36] and autoencoder [37] fail to cluster together the same-floor signals. Such unsatisfactory performance is expected to stem from the aforementioned missing value problem when the RF signal samples are represented in a matrix form.

C. Cluster Training

Given the learned ego embeddings of RF signal samples, only a few of which come with floor labels (which floor they were collected), our goal here is to build a simple yet effective classification model. To this end, we use a proximity-based hierarchical clustering, which works as follows. Initially, each embedding is treated as an individual cluster. Note that only a few embeddings are for *floor-labeled* samples, while most embeddings correspond to *unlabeled* samples. We then repeatedly merge two clusters that are closest to each other such that
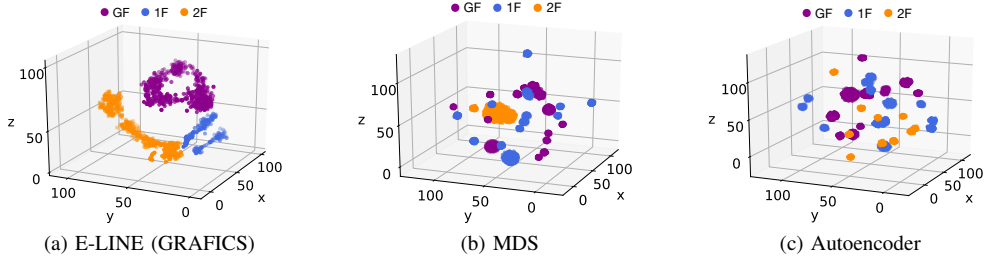
Fig. 6. The E-LINE's embeddings of the RF signal samples from each floor naturally form a cluster in the embedding space, while MDS and autoencoder lead to poor embedding performance.
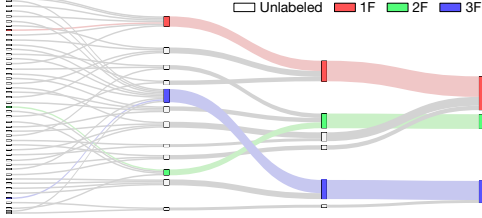

Fig. 7. An illustrative example of proximity-based hierarchical clustering.

the resulting cluster has the embedding of *at most* one floor-labeled sample. In other words, two clusters that both have floor-labeled samples cannot be merged into a cluster. This clustering process is repeated until every embedding is merged into one cluster, and each cluster contains the embedding of exactly one floor-labeled sample, which is used as the *floor label* of the cluster. Here, to compute the distance between two clusters, we use the following distance measure. For clusters i and j, let $\boldsymbol{\Psi}_i$ and $\boldsymbol{\Psi}_j$ be the sets of the embeddings in their clusters, respectively. The distance between clusters i and j is defined as

$$d(\boldsymbol{\Psi}_i, \boldsymbol{\Psi}_j) := \frac{1}{|\boldsymbol{\Psi}_i||\boldsymbol{\Psi}_j|} \sum_{\boldsymbol{u}_i \in \boldsymbol{\Psi}_i} \sum_{\boldsymbol{u}_j \in \boldsymbol{\Psi}_j} \|\boldsymbol{u}_i - \boldsymbol{u}_j\|_2, \quad (11)$$

where $\|\cdot\|_2$ represents the ℓ_2 norm.

Figure 7 visualizes how the embeddings of 48 signal samples obtained on three floors are eventually merged into three clusters, each with the embedding of one floor-labeled sample. While this example assumes that there is only one labeled sample from each floor, the samples collected on each floor may possess several floor-labeled samples. Thus, multiple clusters can be associated with one floor. In addition, we provide a visualization of the clustering process in Figure 8, when there are four labeled samples each floor along with unlabeled samples obtained in the three-story building. As shown in Figure 8, all unlabeled samples are eventually merged into the clusters with labeled samples in the embedding space, which are well separated. Here the labeled samples are colored with the same color for each floor, and the unlabeled samples are colored with the color of the merged cluster.

V. GRAFICS: ONLINE INFERENCE

We turn our attention to the online inference of GRAFICS. For a new RF signal sample, we explain how to obtain its ego and context embeddings. We then move on to the prediction

of its floor label based on its ego embedding and the clusters obtained via the proximity-based hierarchical clustering.

A. Embedding Prediction

When a new RF signal sample becomes available, it is added into the bipartite graph $\mathcal{G}$ as a new 'RF-record' node, say r, and its edges are created with the 'MAC' nodes that appear in the record. Some MAC nodes may also be newly added into the graph, if they are new ones. The edge weights are then determined based on the weight function in (1) and (2) with the recorded RSS values.[1]

Once the new node r is added into the graph, its embeddings $\boldsymbol{u}_r$ and $\boldsymbol{u}_r'$ are learned to minimize the objective function in (9) while the embeddings of the other nodes that have been in the graph remain fixed. This way, the new node r is placed near its local neighborhood in the (ego) embedding space. Similarly for newly added MAC nodes. It is worth noting that minimizing the objective function in (9) with respect to $\boldsymbol{u}_r$ and $\boldsymbol{u}_r'$ is computationally *inexpensive* and can be done in real-time since the other (previously learned) embeddings remain fixed.

B. Floor Prediction

Given the ego embedding $\boldsymbol{u}_r$ of node/sample r, we calculate its distances with the centroids of all the clusters. We then find its predicted floor label as the label of the cluster that is closest to $\boldsymbol{u}_r$, which indicates on what floor the sample r was collected. Specifically, the centroid $\boldsymbol{\psi}_i$ of cluster i is defined as

$$\boldsymbol{\psi}_i := \frac{1}{|\boldsymbol{\Psi}_i|} \sum_{\boldsymbol{u}_i \in \boldsymbol{\Psi}_i} \boldsymbol{u}_i,$$

and the distance between $\boldsymbol{u}_r$ and $\boldsymbol{\psi}_i$ is defined as

$$d(\boldsymbol{u}_r, \boldsymbol{\psi}_i) := \|\boldsymbol{u}_r - \boldsymbol{\psi}_i\|_2.$$

Let l_i be the label of cluster i and $l_{i^\star}$ be the label of the cluster whose centroid is closest to $\boldsymbol{u}_r$, where

$$i^\star := \operatorname*{argmin}_i d(\boldsymbol{u}_r, \boldsymbol{\psi}_i).$$

Then, the predicted label of sample r is simply $l_{i^\star}$. Recall that each cluster has one floor-labeled sample, and the floor label of such a sample in the closest cluster becomes the predicted label of r.

---

[1]It is possible that the new signal sample only contains the MAC addresses that have *never* been seen before. In this case, it may be the one collected outside the building and is thus discarded.
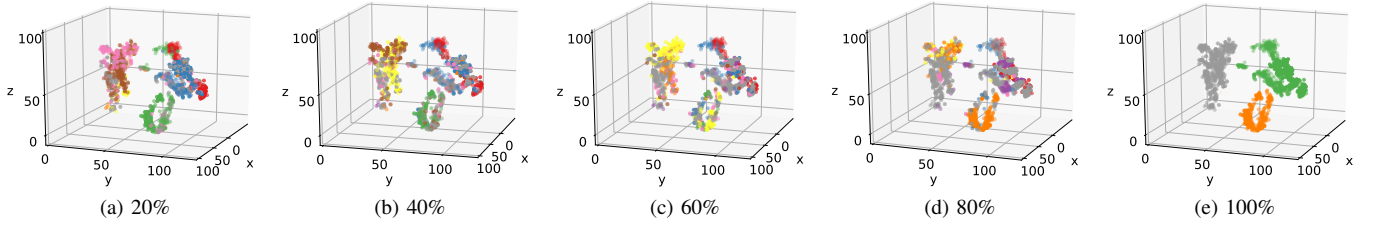
(a) 20% (b) 40% (c) 60% (d) 80% (e) 100%

Fig. 8. GRAFICS gradually clusters the RF signal samples from the same floors in a three-story building, when only four samples are *labeled* for each floor.
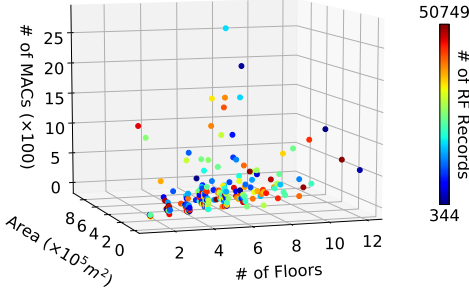

Fig. 9. A summary of building information. Each point represents a building.

Fig. 10. Screenshot of our data collection APP.

VI. ILLUSTRATIVE EXPERIMENTAL RESULTS

In this section, we present the extensive experiment results of GRAFICS. We first explain experiment settings and present the results on performance comparison between GRAFICS and state-of-the-art algorithms. We also study the impact of different system components and parameters on GRAFICS.

A. Experiment Settings

Experiment setup: We conduct experiments on two large-scale datasets. The first one is Microsoft's open dataset in Kaggle competition [38], which contains WiFi signal records collected in 204 buildings, where the smallest one is a two-story building and the tallest one has 12 floors. The other dataset is our own dataset. We develop and use a data collection APP (shown in Figure 10) to collect WiFi signal samples from five facilities in Hong Kong, which are two office towers, a hospital, and two shopping malls. As shown in Figure 9, the datasets cover a wide range of buildings in terms of the number of floors and the building size. The number of sensed MAC addresses varies over the buildings. The number of collected RF signal samples is also quite different across the buildings. Each *floor* is associated with around 1000 WiFi signal samples on average, which are collected in a crowdsourced manner. Unless otherwise mentioned, we present the average results of the 204 buildings from the Microsoft dataset and the ones of the five buildings from the Hong Kong dataset separately.

We set the baseline parameters for performance comparison as follows. The learning and dropout rates of E-LINE are set to 0.001 and 0.1, respectively. We use eight-dimensional vectors for the ego and context embeddings in E-LINE and the embeddings for other algorithms. We use 70% of each dataset for training and 30% for testing. In the training samples, there are only four floor-labeled samples (which are randomly chosen) on each floor, while the rest of the samples are unlabeled. We run each algorithm 10 times for all test cases
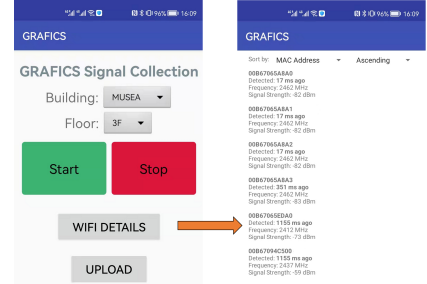
and report their average values. All experiments are conducted on the Ubuntu 18.04 server with 10 Intel Core I9-9900X cores at 3.5GHz, 64GB memory, and an NVidia 2080Ti graphic card. Our code is available online.[2]

Algorithms used for comparison: We consider the following state-of-the-art algorithms where two of them are combined with our proximity-based hierarchical clustering (denoted as Prox) for fair comparison:

- Scalable-DNN [30]: Embeddings are first generated through an encoding network, and floor ids are predicted as one-hot vectors through a feed-forward floor classifier.
- SAE [15]: Stacked autoencoders are used to learn low-dimensional embeddings, and a hierarchical classifier is then used for floor classification.
- Autoencoder [37] + Prox: It learns the embeddings of signal samples through an encoding and decoding process, which are then used with Prox.
- Multidimensional scaling (MDS) [36] + Prox: It learns the embeddings of samples by optimizing some distance matrix, which are then used with Prox.

For scalable-DNN and SAE, we set the parameters as described in [30] and [15], respectively. Recall that a majority of training samples are unlabeled. We assign 'pseudo' labels to the embeddings of unlabeled samples, which are the label of the closest labeled embedding, for training the supervised-learning models of scalable-DNN and SAE. The autoencoder consists of the four layers of 1-D convolution with the ReLU activation function. For MDS, the pairwise distance is set as $1 - \text{cosine}$ similarity between two vectors.

Evaluation metrics: We use micro-F and macro-F to evaluate the classification performance. For floor i, we calculate the number of true positives TP_i, which is the number of correctly
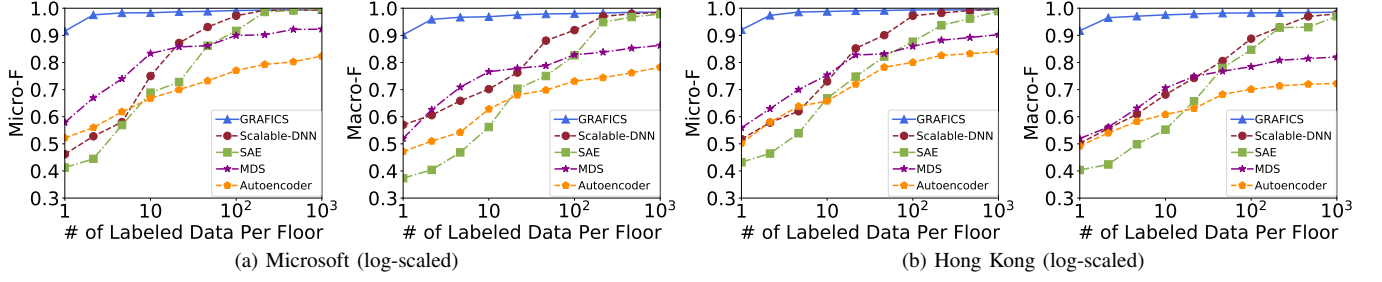
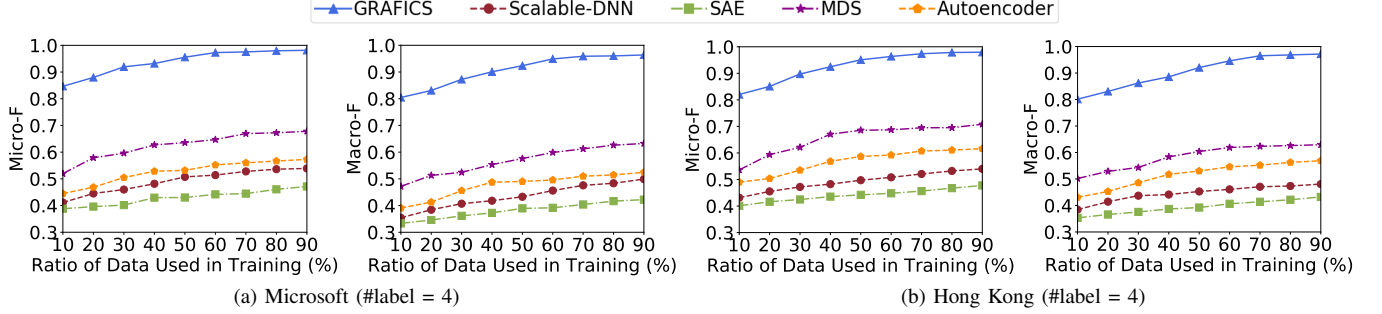Fig. 11. F-scores of GRAFICS with varying number of labeled samples on each floor.

Fig. 12. F-scores vs. the ratio of training data to the entire dataset, where the number of labeled samples is four on each floor.

classified samples. We also calculate the number of false positives FP_i, i.e., the number of samples that are incorrectly labeled as i, and the number of false negatives FN_i, i.e., the number of samples misclassified as one of the other floors. Then, we have $P_i = \mathrm{TP}_i/(\mathrm{TP}_i + \mathrm{FP}_i)$, $R_i = \mathrm{TP}_i/(\mathrm{TP}_i + \mathrm{FN}_i)$ and $F_i = 2P_iR_i/(P_i + R_i)$. Assuming that there are n floors in a building, we define the following micro metrics:

$$\text{micro-}P = \frac{\sum_{i=1}^n \mathrm{TP}_i}{\sum_{i=1}^n (\mathrm{TP}_i + \mathrm{FP}_i)}, \ \text{micro-}R = \frac{\sum_{i=1}^n \mathrm{TP}_i}{\sum_{i=1}^n (\mathrm{TP}_i + \mathrm{FN}_i)},$$
$$\text{and micro-}F = 2\frac{\text{micro-}P \times \text{micro-}R}{\text{micro-}P + \text{micro-}R}.$$

We also use the following macro metrics:

$$\text{macro-}P = \frac{\sum_{i=1}^n P_i}{n}, \ \text{macro-}R = \frac{\sum_{i=1}^n R_i}{n}, \ \text{and}$$
$$\text{macro-}F = 2\frac{\text{macro-}P \times \text{macro-}R}{\text{macro-}P + \text{macro-}R}.$$

B. Comparison with State-of-the-art Algorithms

In Figure 11, we provide comparison results of GRAFICS against other state-of-the-art algorithms for floor classification with varying number of labeled samples on each floor. As can be seen from Figure 11, more labeled samples are available, better the performance of each algorithm. Nonetheless, GRAFICS achieves the best overall performance. It is even able to achieve very high prediction accuracy with just a few labeled samples (i.e., four samples) on each floor, thanks to its novel bipartite graph modeling, high-quality graph embedding via E-LINE, and effective hierarchical clustering.

Scalable-DNN and SAE, however, require a large number of labeled samples. This is somewhat well expected due to their supervised learning nature where their models need to be calibrated with abundant labeled samples. To reach comparable

F-scores to the ones with GRAFICS, they need about 400 to 700 labeled samples per floor, which are $100\times$ more than the ones required by GRAFICS and would be difficult to obtain in practice. On the other hand, MDS and autoencoder do not have much benefit from having more labeled samples since their embeddings are not as good as the ones by E-LINE due to the missing value problem.

We are also interested in how the ratio of training data to the entire dataset affects the performance of each model while the number of labeled samples for training remains unchanged. The results are shown in Figure 12, where the numbers of labeled samples remain fixed as four. We observe that the performance of every model improves with increasing number of samples for training.

C. System Component Study

To see how much GRAFICS benefits from E-LINE over LINE, we present the comparison results between GRAFICS and GRAFICS with LINE (instead of E-LINE) in Figure 13. Here we consider two cases where one is with four labeled samples per floor and the other is with 40 labeled samples per floor. For LINE, we consider its second-order proximity only since it turns out to be better than LINE with first-order and second-order proximities. We omit the results due to space limit. As can be seen from Figure 13, when four labeled samples per floor are only available, GRAFICS with LINE does not perform well and it exhibits high variance in its performance. As the number of available labeled samples increases, the performance of GRAFICS with LINE greatly improves and becomes more stable. However, GRAFICS already achieves the ideal prediction accuracy only with four labeled samples. The results confirm the advantage of E-LINE that learns the embeddings of nodes based on their local
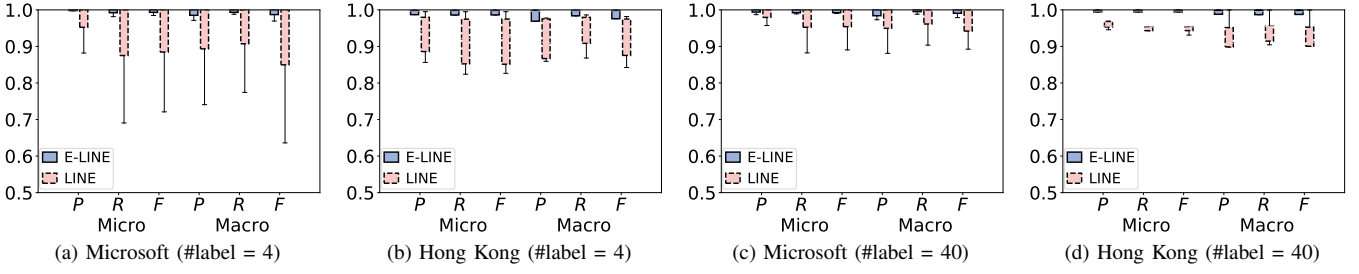
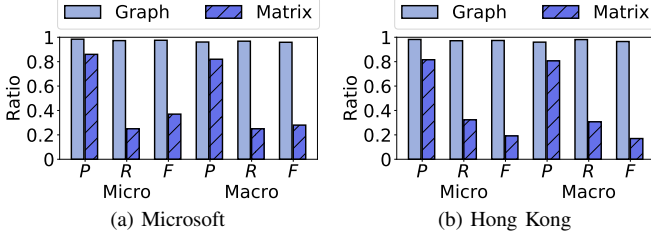Fig. 13. Performance comparison between GRAFICS (with E-LINE) and GRAFICS with LINE.

Fig. 14. Graph modeling and E-LINE vs. matrix representation.

Fig. 15. Insensitivity of GRAFICS to the choice of the embedding dimension.

neighborhood information, which is more than just their one-hop neighborhood information – the case with LINE. In other words, E-LINE makes the embeddings of nodes in their local neighborhood similar to each other while letting the ones of distant nodes dissimilar to each other.

We also evaluate the effectiveness of our bipartite graph modeling and E-LINE in GRAFICS compared to the case when GRAFICS is simply based on a matrix representation of RF signal samples, i.e., the matrix representation is directly used with the proximity-based hierarchical clustering. Here, for the matrix representation, the missing entries are filled with -120dBm, and each row is considered as an embedding of each sample. As shown in Figure 14, the matrix representation leads to quite poor classification performance, which clearly indicates the seriousness of the missing value problem. However, GRAFICS is inherently free of this problem and achieves outstanding performance, because E-LINE effectively captures the similarities and differences among the RF signal samples in the embedding space after their relationships are represented in the bipartite graph.

D. System Parameter Evaluation

GRAFICS learns the embedding vector of each RF signal sample (each node of one type in the bipartite graph) for floor classification. It is thus important to check how the dimension of the embedding vector affects the system performance. Figure 15 shows the F-scores when the dimension varies. GRAFICS consistently performs well regardless of the choice of the dimension. This indicates that GRAFICS does not require a careful choice of the dimension and its real deployment would be easier.

When we construct a (weighted) bipartite graph, we need to ensure that the edge weights are non-negative for graph embedding. To achieve this, we define a proper weight function that adds a valid offset to all collected RSS values, i.e., $f(\text{RSS}) = \text{RSS} + 120$. Another choice of the weight function
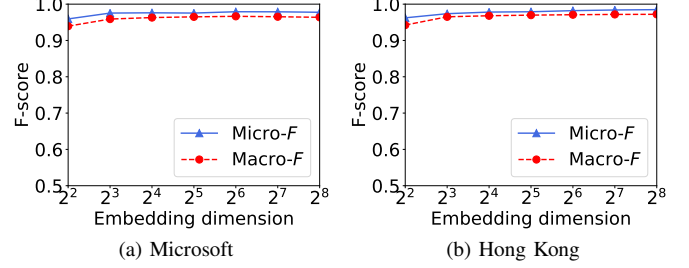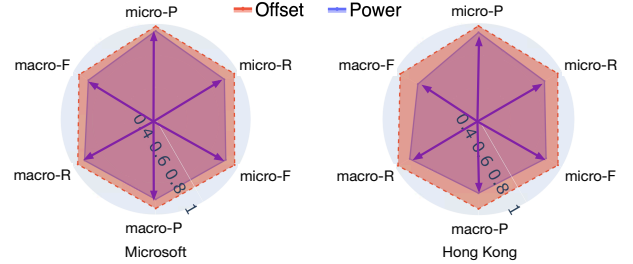

Fig. 16. Impact of different weight functions on the performance.

would be to use a function that converts each RSS value in dBm into the one in power, which is $g(\text{RSS}) = 10^{\text{RSS}/10}$. In Figure 16, we present the comparison results between GRAFICS (with $f(\cdot)$) and GRAFICS that is used with $g(\cdot)$ instead of $f(\cdot)$. We see that the performance of using $f(\cdot)$ is substantially better than the one with $g(\cdot)$. The change in RSS values does not lead to much difference in $g(\cdot)$ compared to $f(\cdot)$, and thus the edge weights would be similar over different edges. Having similar edge weights makes the learned embeddings less effective. In other words, we observe that preserving the differences in RSS values is important to obtain high-quality node embeddings. We also tested different offset values and observed that the performance is more or less the same. We omit the results here for brevity.

To further validate the performance of GRAFICS when the ambient RF signals are from fewer APs (or fewer sensed MAC addresses), we consider the situation where a small fraction of MAC addresses are assumed to exist in the buildings. Figure 17 shows the results of GRAFICS with varying size of the fraction. We see that even with 10% of MAC addresses remaining on-site, GRAFICS can still achieve F scores higher than 0.8, which demonstrates its robustness in sparse RF environments. With 30% to 40% of MACs available on-site, GRAFICS reaches F scores higher than 0.9, which shows that GRAFICS can be readily deployed in practice.
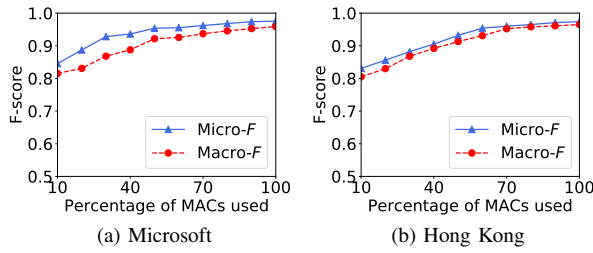
| (a) Microsoft | (b) Hong Kong |

Fig. 17. *F*-scores of GRAFICS with varying number of MACs available.

VII. CONCLUSION

We have presented GRAFICS – a novel graph embedding-based floor identification system, which consists of the novel bipartite graph modeling, high-quality graph embedding via E-LINE, and effective proximity-based hierarchical clustering. We have validated its performance on two large-scale datasets and demonstrated its superior prediction performance with only a few labeled samples over several state-of-the-art algorithms (by about 45% in micro-*F* score and 53% in macro-*F* score). We have also shown its various practical aspects for real deployment.

REFERENCES

[1] A. Shokry, M. Elhamshary, and M. Youssef, "The tale of two localization technologies: Enabling accurate low-overhead WiFi-based localization for low-end phones," in *ACM SIGSPATIAL*, 2017.

[2] L. Yang, Y. Yu, and B. Li, "Multi-floor indoor localization based on RBF network with initialization, calibration, and update," *IEEE Trans. Wirel. Commun.*, vol. 20, no. 12, pp. 7977–7991, 2021.

[3] S. He, B. Ji, and S.-H. G. Chan, "Chameleon: Survey-free updating of a fingerprint database for indoor localization," *IEEE Pervasive Comput.*, vol. 15, no. 4, pp. 66–75, 2016.

[4] S. He, S.-H. G. Chan, L. Yu, and N. Liu, "Calibration-free fusion of step counter and wireless fingerprints for indoor localization," in *ACM UbiComp*, 2015.

[5] B. McCabe, H. Hamledari, A. Shahi, P. Zangeneh, and E. R. Azar, "Roles, benefits, and challenges of using uavs for indoor smart construction applications," in *J. Comput. Civ. Eng.*, 2017, pp. 349–357.

[6] J. P. Queralta, C. M. Almansa, F. Schiano, D. Floreano, and T. Westerlund, "UWB-based system for UAV localization in GNSS-denied environments: Characterization and dataset," in *IEEE IROS*, 2020.

[7] J. Tan, E. Sumpena, W. Zhuo, Z. Zhao, M. Liu, and S.-H. G. Chan, "IoT geofencing for COVID-19 home quarantine enforcement," *IEEE Internet of Things Magazine*, vol. 3, no. 3, pp. 24–29, 2020.

[8] M. F. Monir, A. H. Chowdhury, R. Anzum, and M. A. Amin, "IoT enabled geofencing for COVID-19 home quarantine," in *IEEE ICCCE*, 2021.

[9] V. Spurny, V. Pritzl, V. Walter, M. Petrlik, T. Baca, P. Stepan, D. Zaitlik, and M. Saska, "Autonomous firefighting inside buildings by an unmanned aerial vehicle," *IEEE Access*, vol. 9, pp. 15 872–15 890, 2021.

[10] HERO, "Introducing in-store QR codes, a contactless way to shop." https://www.usehero.com/news/in-store-qr-codes/, 2021.

[11] Gov.UK, "Create a coronavirus NHS QR code for your venue." https://www.gov.uk/create-coronavirus-nhs-qr-poster, 2021.

[12] S. Zhang, J. Guo, W. Wang, and J. Hu, "Floor recognition based on SVM for WiFi indoor positioning," in *Springer CSNC*, 2018.

[13] H.-Y. Hsieh, S. W. Prakosa, and J.-S. Leu, "Towards the implementation of recurrent neural network schemes for WiFi fingerprint-based indoor positioning," in *IEEE VTC-Fall*, 2018.

[14] X. Song, X. Fan, C. Xiang, Q. Ye, L. Liu, Z. Wang, X. He, N. Yang, and G. Fang, "A novel convolutional neural network based indoor localization framework with WiFi fingerprinting," *IEEE Access*, vol. 7, pp. 110 698–110 709, 2019.

[15] M. Nowicki and J. Wietrzykowski, "Low-effort place recognition with WiFi fingerprints using deep learning," in *Springer ICA*, 2017.

[16] A. Alitaleshi, H. Jazayeriy, and S. J. Kazemitabar, "WiFi fingerprinting based floor detection with hierarchical extreme learning machine," in *IEEE ICCKE*, 2020.

[17] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "LINE: Large-scale information network embedding," in *TheWebConf*, 2015.

[18] C. Ascher, C. Kessler, R. Weis, and G. F. Trommer, "Multi-floor map matching in indoor environments for mobile platforms," in *IEEE IPIN*, 2012.

[19] H. Ye, T. Gu, X. Tao, and J. Lu, "F-Loc: Floor localization via crowdsourcing," in *IEEE ICPADS*, 2014.

[20] X. Shen, Y. Chen, J. Zhang, L. Wang, G. Dai, and T. He, "BarFi: Barometer-aided Wi-Fi floor localization using crowdsourcing," in *IEEE MASS*, 2015.

[21] F. Zhao, H. Luo, X. Zhao, Z. Pang, and H. Park, "HYFI: Hybrid floor identification based on wireless fingerprinting and barometric pressure," *IEEE Trans. Ind. Informat.*, vol. 13, no. 1, pp. 330–341, 2015.

[22] Y. Li, Z. Gao, Z. He, P. Zhang, R. Chen, and N. El-Sheimy, "Multi-sensor multi-floor 3D localization with robust floor detection," *IEEE Access*, vol. 6, pp. 76 689–76 699, 2018.

[23] I. Ashraf, S. Hur, M. Shafiq, and Y. Park, "Floor identification using magnetic field data with smartphone sensors," *Sensors*, vol. 19, no. 11, p. 2538, 2019.

[24] J. Chen, S. Wen, and S.-H. G. Chan, "Joint demosaicking and denoising in the wild: The case of training under ground truth uncertainty," in *AAAI*, 2021.

[25] R. Elbakly, H. Aly, and M. Youssef, "Truestory: Accurate and robust RF-based floor estimation for challenging indoor environments," *IEEE Sens. J.*, vol. 18, no. 24, pp. 10 115–10 124, 2018.

[26] F. Dou, J. Lu, T. Xu, C.-H. Huang, and J. Bi, "A bisection reinforcement learning approach to 3-D indoor localization," *IEEE Internet Things J.*, vol. 8, no. 8, pp. 6519–6535, 2020.

[27] R. Elbakly and M. Youssef, "The StoryTeller: Scalable building-and ap-independent deep learning-based floor prediction," *ACM IMWUT*, vol. 4, no. 1, pp. 1–20, 2020.

[28] C. Wang *et al.*, "Secure and reliable indoor localization based on multi-task collaborative learning for large-scale buildings," *IEEE Internet Things J.*, 2021.

[29] G. Caso, L. De Nardis, F. Lemic, V. Handziski, A. Wolisz, and M.-G. Di Benedetto, "ViFi: Virtual fingerprinting WiFi-based indoor positioning via multi-wall multi-floor propagation model," *IEEE Trans. Mobile Comput.*, vol. 19, no. 6, pp. 1478–1491, 2019.

[30] K. S. Kim, S. Lee, and K. Huang, "A scalable deep neural network architecture for multi-building and multi-floor indoor localization based on Wi-Fi fingerprinting," *Springer Big Data Analytics*, 2018.

[31] K. S. Kim, R. Wang, Z. Zhong, Z. Tan, H. Song, J. Cha, and S. Lee, "Large-scale location-aware services in access: Hierarchical building/floor classification and location estimation using Wi-Fi fingerprinting based on deep neural networks," *Fiber and Integrated Optics*, vol. 37, no. 5, pp. 277–289, 2018.

[32] M. Abbas, M. Elhamshary, H. Rizk, M. Torki, and M. Youssef, "WiDeep: WiFi-based accurate and robust indoor localization system using deep learning," in *IEEE PerCom*, 2019.

[33] Y. Ma and J. Tang, *Deep Learning on Graphs*. Cambridge University Press, 2021.

[34] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *NeurIPS*, vol. 26, 2013.

[35] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE." *JMLR*, vol. 9, no. 11, 2008.

[36] M. A. Cox and T. F. Cox, "Multidimensional scaling," in *Handbook of data visualization*, 2008.

[37] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[38] Microsoft, "Kaggle competition on indoor location & navigation," https://www.kaggle.com/c/indoor-location-navigation/overview, 2021.