https://doi.org/10.48048/tis.2023.6739

Distributed Representations of Wi-Fi Fingerprints from Non-Contextual Text-Embedding Techniques with Applications in Crowdsourcing Zone-Level Localization

Chotipon Pakdeethammasakul[1] and Nirand Pisutha-Arnond[2,*]

[1]Master's Degree Program in Industrial Engineering, Faculty of Engineering, Chiang Mai University, Chiang Mai 50200, Thailand
[2]Department of Industrial Engineering, Faculty of Engineering, Chiang Mai University, Chiang Mai 50200, Thailand

([*]Corresponding author's e-mail: nirand.p@cmu.ac.th)

Received: 16 February 2023, Revised: 30 March 2023, Accepted: 25 May 2023, Published: 28 August 2023

Abstract

Over the past decade, indoor localization systems have gained increasing attention and found widespread applications in commercial and research environments. Specifically, a Wi-Fi fingerprint-based system offers a low-cost solution over its counterparts such as Bluetooth, ultra-wideband (UWB), and radio frequency identification (RFID) technologies due to the ubiquity of Wi-Fi access points (WAPs) in most buildings. However, the main disadvantage of the fingerprint-based system is intensive survey effort required during system initialization and maintenance. This work explores a solution to alleviate this limitation by considering a crowdsourcing approach for zone-level localization. Instead of relying only on the labelled fingerprint data from trained surveyors, this approach uses the more-attainable unlabelled fingerprint data collected by participating volunteers. This unlabelled data is then used to augment the survey data in a process called pseudo labelling, forming a more comprehensive training dataset for subsequent localization tasks; this semi-supervised approach allows for minimal survey effort during system initialization and maintenance. To enable such solution, this work introduces a novel approach of employing non-contextual word embedding techniques to construct distributed vector representations of fingerprint data to overcome 3 challenges; (a) high memory requirement in the downstream tasks due to high-dimensional non-distributed vector representations from the "standard" vector transformation, (b) inclusion of an arbitrary value that represents missing WAPs which can affect the performance of the downstream localization tasks in a non-transparent manner, and most importantly, (c) poor pseudo-labelling and semi-supervised zone-prediction performances due to poor data separability in a feature space. The choice of the non-contextual text-embedding techniques, as opposed to the contextual counterparts, leads to less computational requirement in model training and distributed-representation generation due to simpler model architectures (no deep learning) and no requirement for pre-trained model during distributed-representation generation. To this end, we considered non-contextual word embedding techniques commonly used in natural language processing such as Word2Vec, GloVe, and Doc2Vec in the distributed-representation transformation, and compared the resulting downstream performances with those from well-recognized dimensionality reduction techniques such as PCA, Isomap, and UMAP. The results show that Word2Vec and GloVe transformations outperform other types of transformations in terms of separability in fingerprint representations, pseudo-labelling performance, and semi-supervised zone-prediction accuracy. Together with the promising robustness property against potential data inhomogeneity, Word2Vec and GloVe transformations are the recommended transformation processes for constructing vector representations of fingerprints in crowdsourcing zone-level localization.

Keywords: Wi-Fi fingerprint, Indoor localization, Classification, Word embeddings

Introduction

Indoor localization is a process where locations of objects or people inside buildings are identified; this technology has many applications such as those in customer tracking for advertisement, asset tracking for inventory management, and occupancy monitoring for safety. Since a global positioning system (GPS) lacks precision or fails entirely in enclosed space, indoor localization is typically realized through other wireless positioning technologies [1,2], such as wireless local area network (known as Wi-Fi) [3,4]. Bluetooth low energy (BLE) [5,6], Radio frequency identification (RFID) [7-9], or ultrawideband (UWB)

systems [10-12]. However, these systems require installation of specialized equipment, which can incur large installation and equipment cost if large coverage area is required.

On the other hand, fingerprint-based technique provides a budget-conscious localization alternative. The technique provides localization through identification of the characteristics of wireless signals (or fingerprint) detectable at the present location with previously collected information called radio maps. This technique takes advantages of the availability and ubiquity of wireless signals in typical buildings such as Wi-Fi and BLE signals from the Wi-Fi access points (WAPs) and BLE beacons, respectively. This means that the technique requires no extra infrastructure cost and can potentially provide large coverage area without large investment [13-15].

In this work, we considered 2 aspects for the fingerprint-based localization. First, as opposed to a grid-based resolution, we considered a zone-level resolution which, while providing less detailed information, is adequate for many applications and allows for less laborious effort in building and maintaining radio maps [3]. Second, without loss of generality, we limited the type of fingerprint to that from the Wi-Fi technology due to the prevalence of the WAPs in most infrastructures; therefore a fingerprint considered in this work is a mapping between *basic service set identifiers* (BSSIDs), and the respective *received signal strength indicator* (RSSI) values, which are measures of signal strength.

The standard procedure for Wi-Fi fingerprint-based localization involves 2 phases. One is a radio-map construction phase (survey phase) where trained personnel collect *labelled* fingerprint data from the specified zones of interest, typically using mobile devices with Wi-Fi receivers. This survey data is then used to construct a zone-prediction model using machine learning (ML) classification algorithms. The other phase is the localization phase where the *unlabelled* fingerprints from the end-users' mobile devices are collected and used to predict the corresponding zones. The main disadvantage of such procedures is the laborious effort required to not only collect sufficiently-comprehensive data to produce a robust ML prediction model (i.e., collecting multiple fingerprints within a zone using multiple types of mobile phones), but also maintain an up-to-date radio map (i.e., repeating the survey phase) due to possible change in landscape structures and WAPs. Such labour intensiveness can become costly for large-scale system deployment and long-term maintenance [16,17].

To alleviate the problem, we resorted to crowdsourcing techniques [16,18], to facilitate constructing and maintaining up-to-date radio maps. This approach lets the end-users, with their consent, assist in the survey phase by automatically and periodically sending fingerprint data from their mobile devices. With this approach, the survey phase can be continuously operated even during the localization phase, enabling access to larger amount of continuously-updated fingerprint data with minimal effort from the administrative personnel.

Nevertheless, since the crowdsourcing data collection does not require end-users' attention, the collected fingerprints will not contain any zone label; as a result, such data cannot directly be used to construct a zone-prediction model from the ML classification algorithms. Alternatively, our approach is to first augment the *unlabelled* crowdsourcing data to the limited *labelled* survey data to create a more comprehensive radio maps before subsequent zone-prediction-model training. In a ML context, our so-called *semi-supervised* approach [19], consist of (a) a pseudo-labelling process where, given a labelled fingerprint data from a particular zone, the task is to find the subset of the unlabelled fingerprint data that is likely to come from the same zone and (b) semi-supervised zone-prediction task where the labelled survey data and unlabelled crowdsourcing data with pseudo labels are used to construct a zone-prediction model from the ML classification algorithms. From this processing pipeline, the zone-prediction performance will depend on the accuracy of the pseudo-labelling process, and, in turn, depends on the separability of the numerical representations of fingerprint data. In other words, fingerprint data should be numerically represented in such a way that fingerprints from the same zone are distinctly grouped into clusters; such behavior will lead to accurate pseudo-labelling performance and should generally result in an accurate and robust zone-prediction model.

Therefore, the main investigation of this work is to find the transformation techniques that convert raw fingerprint data (mappings between BSSIDs and RSSI values) to vector representations with desirable characteristics for the downstream localization tasks. A *standard* method is to "vectorize" a mapping to a non-distributed vector representation where the elements represent the RSSI values of the detected WAPs [17,20]. However, this "standard transformation" has 3 disadvantages. First, being a non-distributed representation, the resulting vector will have the length equal to the number of WAPs presented in the *entire* fingerprint collection and, for a large survey site, can cause the downstream tasks to be memory intensive. Second, an arbitrary value that represents missing WAPs (or the *null* value) must be chosen; a common choice is a value far from the range of valid RSSI values [21]. This synthetic value becomes a part of the fingerprint data and can affect the performance of the downstream localization tasks in a non-transparent

manner. Third, and most importantly, our study shows that the resulting vector representations are not well-separated, which leads to non-optimized pseudo-labelling performance and poor semi-supervised zone-prediction accuracy. Clearly, an alternative method of fingerprint transformation is needed.

Motivated by the techniques in natural language processing (NLP) where text can be represented by a low-dimensional vector with encoded semantic and contextual information, we proposed a novel technique to transform raw fingerprint data to distributed vector representations using non-contextual text-embedding methods from NLP. The resulting vectors are distributedly represented with predetermined dimension, contain no arbitrary null-value, and exhibit improved clustering characteristics, which results in lower memory requirement in the downstream tasks, no synthetic null value in the fingerprint data, and improved performance of the downstream localization tasks, respectively. Also, the use of non-contextual techniques, as opposed to the contextual counterparts, is favorable in terms of computational requirement in model training and distributed-representation generation due to simpler model architectures (no deep learning) and no requirement for pre-trained model during distributed-representation generation. The proposed transformation technique involves 3 steps. First is corpus construction; similar to how a corpus in NLP contains a collection of text, a corpus in our context contains a collection of fingerprints. Each fingerprint is represented by a randomized space-separated list of detected BSSIDs whose numbers of occurrence in the list are proportional to their respective signal strength. The second step involves training the text-embedding models using the previously-generated corpus and the third step is the construction of the fingerprint representation from the trained text-embedding model; the specific details of this steps depend on the type of the text-embedding model.

In this study, we employed 3 commonly-used non-contextual text-embedding models which are Word2Vec [22], Global Vectors for Word Representation (GloVe) [23], and Sentence2Vec (also known as Doc2Vec) [24], in the transformation process and use the resulting vector representations in various localization tasks. We also compared the localization performances with those from the transformations that use dimensional reduction techniques such as Principal Component Analysis, (PCA) [25], Uniform Manifold Approximation and Projection (UMAP) [26], and Isometric Mapping (Isomap) [27]. Compared with the "standard" and "dimensional-reduction" transformations, the "text-embedding" transformations lead to vector representations that yield improved clustering characteristics, pseudo-labelling accuracy, semi-supervised zone-prediction accuracy, and robustness against missing WAPs. Specifically, the text-embedding transformations that employ Word2Vec and GloVe models outperform other types of transformations and are the recommended transformation processes for constructing vector representations of fingerprints in crowdsourcing zone-level localization.

Background

In this section, we explain the motivation for employing text-embedding techniques in localization context, and discuss the choices of text-embedding models and dimensional reduction methods. We also give background information of the chosen models that is relevant to subsequent discussion.

Text-embedding techniques in localization context

Text embedding is a class of models that provide mapping between paragraphs, sentences, words, or sub-words, to distributed vector representations that contain syntactic, semantic, and, in some cases, contextual information. The transformation is learned though unsupervised training on unlabelled corpora using various language tasks such as predicting a missing word from surrounding context words. The goal of text embedding is to encode language information in a way that similar words or sentences yield similar representations (e.g., small Euclidean distance or cosine similarity); such similarity is beneficial for downstream NLP tasks such as semantic similarity classification and clustering [28].

One can draw an analogy between the goal of text embedding and the goal of fingerprint transformation in our study. We are looking for a transformation of fingerprints that encodes location information in such a way that fingerprints from the same zone yield similar representations; such similarity is beneficial for the downstream localization tasks such as pseudo-labelling and semi-supervised zone-prediction. This analogy motivated our investigation into possible utility of text-embedding techniques in localization context. In order to apply the text-embedding methods to the fingerprint transformation, a connection needs to be established between the constituents from language and fingerprint data; here we proposed a connection where a BSSID and a fingerprint are analogous to a word and a sentence, respectively. Once this connection is established, the "corpus" in localization context can be defined and the subsequent "text-embedding" process can be performed to obtain distributed vector representations of fingerprint data.

At first glance it might seem that employing text-embedding techniques introduces unnecessary complexity to the processing pipeline. As a matter of fact, one can apply dimensionality reduction to the *standard* non-distributed vectors to arrive at low-dimensional distributed vector representations using methods such as PCA, UMAP, and Isomap. Nevertheless, since the input of such transformation is the standard non-distributed vectors, the choice of the null value from the input can affect resulting representations and the performance of the downstream localization tasks in a non-straightforward manner. The "text-embedding" transformation, on the other hand, does not require construction of the standard vector representation and thus requires no arbitrary null value to represent missing WAPs, which leads to less dependence of the performance of the localization tasks on hyper-parameters.

On a broader level, we argue that the use of text-embedding techniques in fingerprint transformation is more appropriate than the use of dimensional reduction techniques. Due to signal disturbance and blockage, as well as variation in detection devices (e.g., different types of Wi-Fi adaptors in mobile phones), the fingerprint data can be highly inhomogeneous even when collected from the same zone [29-32], in other words, the RSSI values and the number of detected BSSIDs from repeated fingerprint collection can vary significantly. For the dimensional-reduction transformations, such unwanted "noise" can be undesirably preserved to a certain extend as the dimensional reduction methods are designed to retain as much characteristics of the original data as possible. Instead of trying to preserve the characteristics of the original data, our proposed text-embedding transformations "normalize" the raw fingerprints into a corpus where, within the same context (i.e., sentence), high-signal-strength WAPs detected from the same zones are more likely to co-occur and be in close proximity. Then the corpus is trained with the text-embedding algorithms that learn similarity based on proximity and/or co-occurrence statistics. We believe that such transformation should be more robust in a sense that the resulting representation should depend less on the exact values of RSSI or the number of detected BSSIDs, but more on the underlying localization information that is useful for the downstream localization tasks.

Non-contextual text-embedding models

Here, we discuss the choice of the text-embedding techniques employed in this work. The commonly-used candidates include Word2Vec [22], GloVe [23], Sentence2Vec (Doc2Vec) [24], FastText [33], ELMo [34], and BERT [35], and the differences among these methods include input granularity (sentences, words, or sub-words), out-of-vocabulary handling, and contextual awareness. In this work, we consider text-embedding techniques that belong to the non-contextual category, which are Word2Vec, GloVe, and Doc2Vec, as opposed to the non-contextual counterparts such as ELMo and BERT. The non-contextuality means that the methods assign only a single representation to a word, ignoring semantic variation that could arise from homonyms and word-order differences; such limitation makes these non-contextual methods less attractive for NLP tasks. However, for our downstream localization tasks (pseudo-labelling and semi-supervised zone prediction), we argue that such context-awareness capability is not important as each "word", or BSSID in our case, has a single meaning, which is the existence of a particular WAP and the order of BSSIDs in a "sentence", or list, is not meaningful. By leveraging the non-contextual techniques, our processing pipeline becomes less computational demanding in terms of model training due to simpler model architecture: shallow neural networks for Word2Vec and Doc2Vec, and no neural network for GloVe, as opposed to deep neural networks (LSTM and transformers) for ELMo and BERT. Also, for Word2Vec and GloVe, the distributed-representation can be generated trivially and efficiently from the resulting mappings from the training process without the need to use the computationally-heavy pre-trained models. We provide an example of difference in computational time between the tasks that use non-contextual and contextual techniques in **Appendix A**. It should be noted that our choice of non-contextual methods does not imply that contextuality is not useful localization tasks. In fact, the contextual language models such as BERT have been applied in more sophisticated scenarios where unreliable and adversarial data can be introduced to the collected data [36,37].

The background information for the selected non-contextual text-embedding techniques is given as follows. Word2Vec is a word-embedding algorithm comprising of a shallow neural network that takes an unlabelled corpus as the input and learns the mapping, or weights, by training on 2 types of language tasks. One is predicting the target words given the surrounding context words within a specified window; such task is known as Continuous Bag of Words (CBOW). The other is predicting the context words given a target word; such task is known as Skipgram (SG). Doc2Vec is an extension to Word2Vec where the algorithm can map variable length text to a fixed-length vector representation. The algorithm includes a sentence representation that is calculated during the training and inference processes. The tasks used in training the model consists of 2 types; first is Distributed Memory (DM) which involves predicting the center word from the context words, similar to CBOW. Second is Distributed Bag of Words (DBOW)

which tries to predict the context words from the center word, similar to SG. Lastly GloVe is an algorithm that combines advantages from Word2Vec which learns representation based on local context and methods such as Latent Semantic Analysis (LSA) [38], which uses global co-occurrence statistics; the algorithm yields representation that performs well in a word analogy task while efficiently utilizing global co-occurrence information.

Dimensional reduction models

To compare with the proposed text-embedding transformations, we employed the transformations that use dimensional reduction techniques such as PCA, Isomap and UMAP to produce distributed vector representations of fingerprints. PCA is an unsupervised matrix factorization method that projects data onto lower dimensional subspace while retaining as much variation in the data as possible. UMAP and Isomap are manifold learning algorithms where the high-dimensional data is transferred onto lower-dimensional manifold with the aim of preserving topological structures of the data. It should be noted that t-Distributed Stochastic Neighbor Embedding (t-SNE) [39], is another commonly used dimensional-reduction method; however, the technique is mostly used for visualization (dimensional reduction to 2D and 3D) as increasing the number of reduced dimensions can lead to impractically slow processing time [40].

Methods

In this section, we describe the methods used in our study which consist of data collection, the proposed vector transformation using text-embedding methods, the choices of clustering metrics used to quantify clustering characteristics, and the downstream experiments. We also discuss the choices of parameters and repetition in the calculation; further details can be found in **Appendixs C** and **D**. Finally, the diagram of the processing pipeline is shown in **Figure 1**.

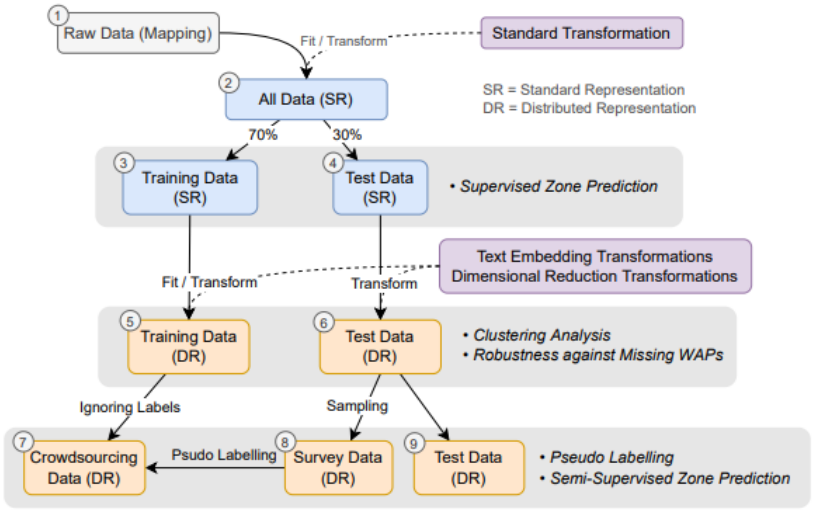


Figure 1 Data processing pipelines.

Data collection

The fingerprint data was collected from the office building at Department of Industrial Engineering, Chiang Mai University, Thailand. The building contains 3 floors with an area of approximately 940 m^2 per floor. A total of 21 zones were predetermined based on common identification such as meeting room, office, and hallway, for examples. We employed an in-house-developed Android application that periodically scanned for nearby WAPs and the corresponding signal strength, and then sent the collected data to a cloud database. At each zone, the surveyors specified the zone in the application, activated the collection process from the application and walked around the area until the collection process was finished. Using 5 different mobiles phones, we scanned around 100 fingerprints per zone which total to 2,060 fingerprints from the site. It should be noted that for optimal zone-prediction accuracy, the fingerprint from each zone should be distinguishable from those of the other zones. This indicates that the zones should be spatially apart, especially those in low WAP-density sites, or physically separated from the neighboring zones by, for example, walls or enclosed partitions. We discuss a zone-distinction test and provide further analysis on zone partitioning in **Appendix B**.

The raw fingerprint data is transformed with the standard transformation to obtain (a) a feature matrix where the feature columns correspond to the BSSIDs and (b) a label column-matrix containing the number-encoded zones. The null value was chosen initially to be −1,000 and we disregarded detection of the WAPs with RSSI value less than −100 by replacing the value of RSSI less than −100 with the null value. The resulting feature matrix contains 443 columns corresponding to the unique WAPs detected in the site, and contains 1,643 rows after the removal of duplicated rows. The data is then split into the training and test sets with 70:30 ratio (Box 3 and 4 in **Figure 1**).

Vector transformation using text-embedding methods

Our proposed vector transformation using text-embedding techniques is divided into 3 parts as follows.

Corpus construction

Similar to how an English text corpus consists of a collection of sentences where each sentence is a space-separated list of words, a corpus in our context also consists of a collection of fingerprints where each fingerprint is a space-separated *shuffled* list of detected BSSIDs. The number of occurrences of a BSSID in the list, n, is determined by $n = RSSI + 100$, where $RSSI$ is the signal strength. In this manner, the BSSIDs from WAPs with high signal strength will appear more frequently in the list and are more likely to be in close proximity to one another. Due to the shuffling operation, multiple unique lists can be generated from a single fingerprint. Also, instead of using the actual BSSID string (i.e., MAC address), we represented a BSSID with a unique ID (a number with a letter prefix) to reduce the corpus memory size and improve readability.

Training of the text-embedding methods

In the training process, the text-embedding models learn the mapping from the generated corpus from the previous step. Specifically, we employed the python implementation of Word2Vec and Doc2Vec from Gensim library [41], and C++ implementation of GloVe from the software developed by the original authors [23].

Construction of fingerprint vector representations

The process of constructing fingerprint vector representations is different for different text-embedding techniques. For Word2Vec and GloVe, the training step only yields a mapping between BSSIDs and the respective distributed vector representations. To construct the representations of a fingerprints, V, we *manually* performed a weighted-average calculation: $V = \sum_i w_i v_i$, where v_i is the distributed vector representations of detected BSSIDs and the summation is performed over all *unique* BSSIDs detected in the respective fingerprint. The weight w_i is calculated from the numbers of occurrence of BSSIDs: $w_i = n_i / \sum_k n_k$. The reason behind weighted averaging is to emphasize the contribution of the representations from the WAPs with strong signal strength as those WAPs are typically in close proximity to the collection location and should be reliable zone indicators. It should be noted that BSSIDs that are not present in the training data will be ignored in this vector-representation construction step. For Doc2Vec, the construction of the fingerprint vector representation consists of generating a shuffled list of BSSIDs (similar to corpus construction) from a selected fingerprint. Then, the BSSID list can be used in the inference process where the representation of the fingerprint is obtained through the internal optimization process.

It should be noted that before the text-embedding transformation, we converted the standard representations (Box 3 and 4 in **Figure 1**) back to the raw format (mapping) before generating the distributed representations. This back-and-forth conversion was performed to ensure consistency of the input data among different choices of transformations. In a production setting, the text-embedding transformation can be performed on the raw data (Box 1 in **Figure 1**) to directly arrive at the distributed representations (Box 5 and 6 in **Figure 1**).

Clustering metrics

To quantify clustering characteristics that benefit the downstream localization tasks, we consider a silhouette value which measures the closeness of a particular data point to its own cluster compared with the nearest cluster (nearest-neighboring cluster). The silhouette value ranges from −1 to +1, where a positive value indicates that the corresponding data point is closer to its own cluster than the nearest-neighboring cluster. On the contrary, the negative value indicates that the data point is more similar to its nearest-neighboring cluster than its own cluster. With the silhouette value calculated for each fingerprint representation, one can consider the mean silhouette, $\bar{S}$, which measures how tightly grouped

representations are on a whole. However, we argue that considering $\bar{S}$ alone is not enough to evaluate the fingerprint representation for the downstream localization tasks. The reason is that the pseudo-labelling and zone-prediction accuracy should be more sensitive to the presence of misclustered fingerprint and $\bar{S}$ alone does not expose such information directly. Therefore, we employed 3 additional metrics which are (I) the fraction of fingerprints with negative silhouette values,$\mathcal{F}^-$, (II) the weighted average of negative silhouette values, $\bar{S}^-$, and (III) the combination of both previously mentioned values:

$$\mathcal{C} = |\bar{s}^-| \times \mathcal{F}^-$$

The quantity $\mathcal{F}^-$ indicates the *amount* of the mismatched fingerprints while the magnitude of $\bar{S}^-$ (i.e., $|\bar{S}^-|$) signifies the *degree* of "overlapping" of the mismatched fingerprints. The value $\mathcal{C}$ takes into account both the amount of mismatched fingerprints and the degree of overlapping. These values reveal information on the misclustered fingerprints and, as will be shown subsequently, exhibit some correlation with the localization performances.

Experiments

Here we summarize the experiments performed in our study as follows.

Supervised zone prediction

The goal of this experiment is twofold. One is to determine the appropriate ML classification algorithm to be used for subsequent tasks (explained below). The second goal is to establish a reference accuracy value of the supervised (no pseudo label) zone-prediction using the representation from the standard transformation. The ML algorithms include Logistic Regression (LR), support vector machine (SVM), decision tree (DT), random forest (RF), k-nearest neighbors (KNN), and gradient boosting. We employed python implementations of LR, SVM, DT, RF, and KNN from Scikit-Learn library [42], and a python implementation of gradient boosting from XGBoost library [43]. We used 10-fold cross-validation to optimize parameters for each algorithm and the details of the parameters are shown in **Appendix C**. The classifiers were trained on vector representations from the standard transformation (Box 3 in **Figure 1**) and the performance metrics were evaluated from the test data (Box 4 in **Figure 1**).

Clustering analysis

In this experiment, we evaluated the clustering characteristics of vector representations from different transformations. We trained the "text-embedding" and "dimensional-reduction" transformations on the training data (Box 3 in **Figure 1**) and used the trained transformations to generate distributed vector representations (Box 5 and 6 in **Figure 1**). We visualized the data structures in 2-dimensions using t-SNE and reported the clustering metrics defined previously.

Pseudo labelling

We simulated the implementation of the crowdsourcing approach by treating the training data as the (unlabelled) crowdsourcing data (Box 7 in **Figure 1**) and the *subset* of the test data as the (labelled) survey data (Box 8 in **Figure 1**); specifically, we sampled 5 survey fingerprints from each zone. We then performed the pseudo labelling process by calculating the mean vector (denoted as the mean fingerprint) of each zone from the sampled fingerprints. For each mean fingerprint, we chose a subset of the crowdsourcing data within a cut-off Euclidean distance; the cut-off distance was calculated from the distance between a particular mean fingerprint and that of the nearest-neighboring mean fingerprint (i.e., from the nearest-neighboring zone). We employed Gaussian mixture (GM) model to partition the subset of the crowdsourcing fingerprints into 2 clusters based on their distances to the selected mean fingerprint. Finally, the members of the cluster with a smaller mean distance is tagged with the zone label corresponding to the mean fingerprint in consideration. The accuracy of the pseudo-labels can be evaluated from the (initially-ignored) ground-truth labels of the crowdsourcing data.

Semi-supervised zone prediction

The semi-supervised zone prediction performance from the crowdsourcing approach was evaluated in this experiment. The ML classification model selected from the supervised zone-prediction experiment was trained on the training data that comprises of the crowdsourcing data with pseudo labels (Box 7 in **Figure 1**) and the survey fingerprints (Box 8 in **Figure 1**). The semi-supervised zone-prediction accuracy was evaluated from the remaining test data (Box 9 in **Figure 1**) that is not part of the survey data.

Robustness against missing WAPs

In this experiment, we demonstrate the ability of the text-embedding transformation to retain meaningful localization information in the presence of data inhomogeneity. Specifically, we considered the standard representations of the fingerprint data (Box 3 and 4 in **Figure 1**) and randomly dropped a specified fraction of the BSSID entries from the fingerprint data (i.e., randomly replace the non-null values with the null value) to simulate temporary signal blockage on certain WAPs (missing WAPs). Then the distributed vector representations were obtained in a similar process to that in the clustering analysis experiment. The selected ML classifier was trained with the training data (Box 5 in **Figure 1**) and the "supervised zone-prediction accuracy" was evaluated from the test data (Box 6 in **Figure 1**).

Parameters and repetition

To make fair comparison between different types of transformations, we evaluated the performance from the experiments multiple times using different parameter sets relevant to different types of transformations. For each parameter set, we also repeated the calculation multiple times (5 in this case) to take into account the randomness in the processing pipeline. Such randomness can come from randomness in the corpus construction which involves a shuffling operation, the stochasticity in the training and/or inference processes of the fingerprint transformations, and the randomness in sampling the test data (the "sampling" arrow in **Figure 1**). For the choices and ranges of parameters used in the experiments, we refer to **Appendix D** for more details.

Results and discussion

In this section, we report the results from the experiments introduced in the previous section which includes supervised zone prediction, clustering analysis, pseudo labelling, semi-supervised zone prediction, and robustness against missing WAPs. Lastly, we give a brief discussion on the optimized sets of parameters of the selected transformations.

Supervised zone prediction

Table 1 shows the supervised zone-prediction performances, which are accuracy, precision, recall and F1, from the different ML classification algorithms. Most classifiers give the performance values higher than 0.940 with LR ranking at the top with the accuracy of 0.977. Therefore, from the results, we chose LR as the classifier for subsequent experiments. From our experience, classification algorithms that produce smooth and simpler decision surfaces such as LR and SVM tend to be less prone to the inhomogeneity (noise) in the data. On the other hand, tree-based algorithms (XGBoost, RF, DF) and KNN do not produce smooth decision surfaces and tend to overfit the training data if the parameters relating to the complexity of the algorithms are not properly tuned. In addition, the accuracy value of 0.977 was established as the reference accuracy of the supervised zone-prediction from the standard transformation; this value will be used subsequently to determine the performance reduction from the crowdsourcing approach.

Table 1 Supervised zone-prediction performances using standard fingerprint representations. All values are weighted-average quantities.

Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)
Logistic regression	97.72	97.91	97.72	97.73
SVM	97.15	97.28	97.15	97.16
K-NN	94.31	94.89	94.31	94.35
XGBoost	97.34	97.45	97.34	97.34
Random forest	95.83	96.67	95.83	95.97
Decision tree	93.36	93.62	93.36	93.33

Clustering analysis

We explored the clustering characteristics of the vector representations from different transformations. For visual comparison, we sampled one representation from each transformation and employed t-SNE to generate 2-dimensional data for plotting. The results are shown in **Figure 2** where unique combinations of markers and filled colors correspond to different zones. From the figure, the

representations exhibit clustering within the same zone with varying degree of cluster sizes and distributions. Also, the clustering metrics are shown in Figure 3 where the values of $\bar{S}$, $\mathcal{F}^-$,$|\bar{s}^-|$ (magnitude of $\bar{s}^-$), and $\mathcal{C}$ from different types of transformations are compared; we note that the variation in these values comes from different parameter sets and repetition as detailed in **Appendix D**.

In general, it is preferable for the fingerprint representations to exhibit high $\bar{S}$, low $\mathcal{F}^-$, low $|\bar{s}^-|$, and low $\mathcal{C}$. For Word2Vec and GloVe transformations, the representations exhibit improvement in terms of cluster separation from that of the standard representation as seen visually from more-tightly-grouped clusters that are further apart, and quantitatively from higher $\bar{S}$. The substantially lower $\mathcal{F}^-$ (resulting in lower $\mathcal{C}$) indicates less mismatched fingerprints, which is a desirable characteristic and, as will be shown subsequently, results in improved localization performances. For the Doc2Vec transformation, the representation yields inferior separability as seen visually from multiple overlapping clusters, and quantitatively from lower $\bar{S}$. The higher $\mathcal{F}^-$ indicates higher amount of mismatched fingerprints, which makes Doc2Vec transformation less attractive in terms of localization performance. It should be noted that large variation in $\mathcal{F}^-$ is attributed to stochasticity in the inference process of the Doc2Vec model [24].

For the PCA and Isomap transformations, the representations show marginal improvement in separability as seen from visual comparison and from slightly higher $\bar{S}$. However, the moderate values of $\mathcal{F}^-$ and $|\bar{s}^-|$ indicate substantial amount of mismatched fingerprints and higher degree of overlapping, respectively, which makes these transformations less promising. For the UMAP transformation, even though visual comparison and the $\bar{S}$ value suggest high separability, the substantial amount of mismatched fingerprints and particularly high degree of overlapping (resulting in higher $\mathcal{C}$) are undesirable characteristics and, as will be shown subsequently, the UMAP transformation yields mediocre localization performances compared with those from other transformations.

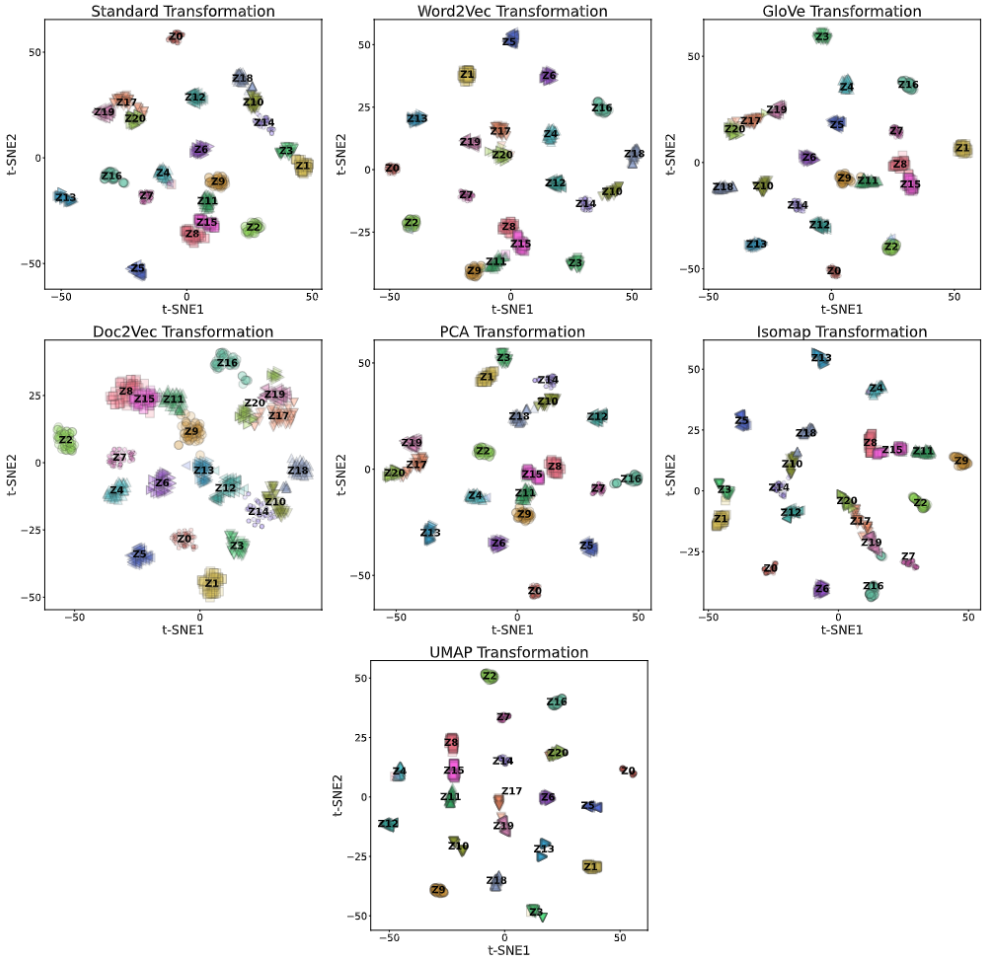


Figure 2 Two-dimensional visualization of fingerprint representations from different types of transformations.

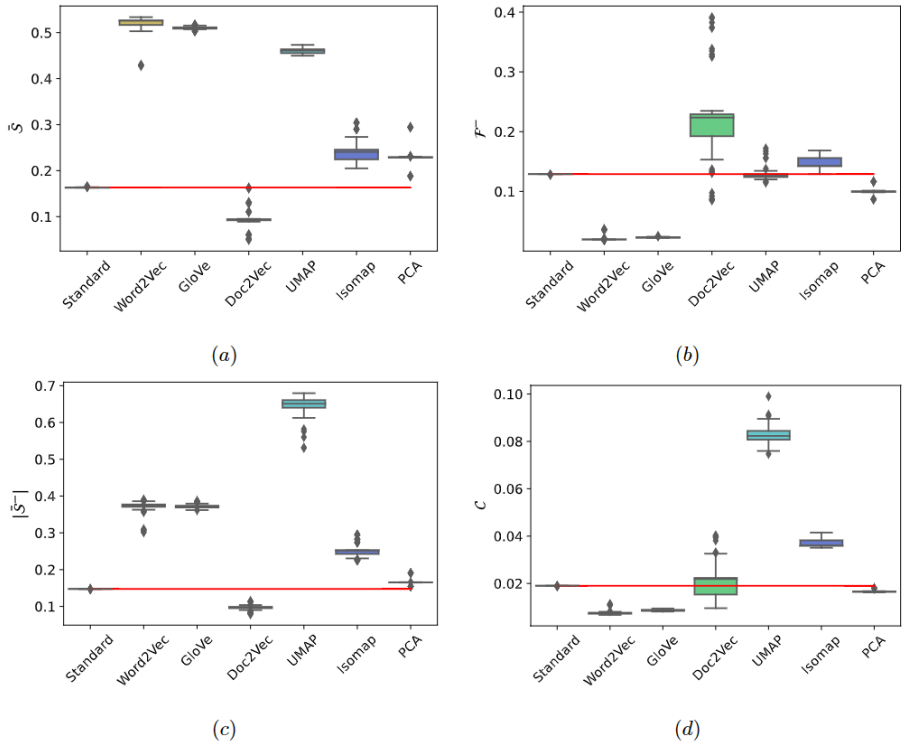


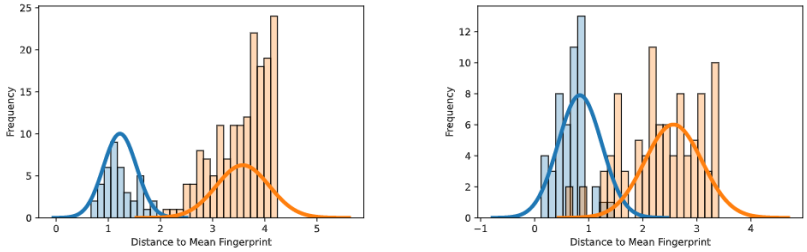
Figure 3 Clustering metrics from fingerprint representations from different types of transformations. The horizontal lines represent the baseline values from the calculations with the standard fingerprint representations.

Pseudo labelling

Figure 4 demonstrates how the pseudo labels are generated. The histograms show distributions of the subset of the crowdsourcing data within a cut-off distance from the mean fingerprint from a particular zone. Superimposed on the histograms are 2 Gaussian distribution functions from the trained GM model. These 2 functions are used to determine cluster-assignment probability and the members of the cluster with a smaller mean distance are tagged with the zone label of the corresponding mean fingerprint. **Figure 4(a)** exhibits an ideal case where the crowdsourcing fingerprints are well-separated with no mismatch. The GM model can predict the 2 Gaussian functions that are well-separated and partition the data correctly. From the figure, the correct partition is indicated by the blue (orange) curve covering the blue (orange) bars and this scenario will lead to high pseudo-labelling accuracy. However, in a situation where the crowdsourcing data from different zones overlap, the GM model will not be able assign the correct partition to the data as seen from the orange bar under the blue curve in **Figure 4(b)**. In such a case, the pseudo-labelling performance will be adversely affected.

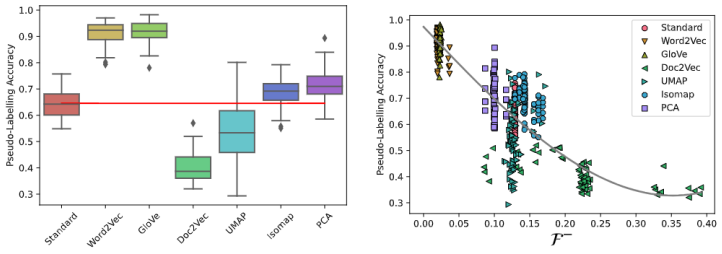
The pseudo-labelling accuracy from different types of transformations is shown in **Figure 5(a)**. From the figure, Word2Vec and GloVe transformations yield improved accuracy from that of the standard transformation; the increase in the mean accuracy is from 0.645 to 0.911 which is approximately 41 %. Such increase is attributed to the desirable clustering characteristics of the representations from the Word2Vec and GloVe transformations as discussed previously. On the hand, the PCA and Isomap transformations yield slight improvement in the accuracy, while UMAP and Doc2Vec transformations yield lower accuracy values. The unpromising performances from the PCA, Isomap, UMAP and Doc2Vec transformations are not unexpected as the representations from these transformations exhibit some undesirable clustering characteristics such as presence of considerable amount of mismatched fingerprints and/or high degree of overlapping.

In addition, the pseudo-labelling accuracy shows some correlation with $\mathcal{F}^-$ value as shown in **Figure 5(b)** where $\mathcal{F}^-$ is inversely proportional to the accuracy. Such correlation is understandable as higher amount of mismatched fingerprints should interfere with the GM model's ability to partition the data correctly. However, the relationship in **Figure 5(b)** still exhibits large variance in the y-values (accuracy), which suggests dependence of the accuracy on other factors. Nevertheless, we believe that $\mathcal{F}^-$ still has some utility as a screening metric for representations that yield promising localization performance.



(*a*) (*b*)

Figure 4 Distributions of the subset of the crowdsourcing data within a cut-off distance from the mean fingerprint from a particular zone. The blue bars indicate the distribution of the crowdsourcing fingerprints that come from the same zone as the mean fingerprint, while the orange bars denote the distribution of the crowdsourcing fingerprints that are collected from other zones. Superimposed on the histograms are 2 Gaussian distribution functions from the trained Gaussian Mixture (GM) model. The Gaussian curves are not drawn to scale. The figures show 2 cases where (a) the fingerprints are well-separated (high pseudo-labelling accuracy) and (b) the fingerprints overlap (low pseudo-labelling accuracy).



(*a*) (*b*)

Figure 5 (a) Pseudo-labelling accuracy from different types of transformations. The horizontal lines represent the baseline values from the calculations with the standard fingerprint representations. (b) Plot of pseudo-labelling accuracy with the fraction of fingerprints with negative silhouette values ($\mathcal{F}^-$). The gray line denotes the polynomial regression fit to the data.

Semi-supervised zone prediction

The performance of the semi-supervised zone-prediction is shown in **Figure 6(a)**. First, we consider the standard representation and compare the performances of the supervised (no pseudo labels) and semi-supervised (with pseudo labels) zone predictions. The reference supervised zone-prediction accuracy, from **Table 1**, is reported to be 0.977 while the semi-supervised zone-prediction accuracy, from **Figure 6(a)**, averages to 0.793, which is about 19 % reduction in accuracy. Therefore, when employing the crowdsourcing approach, using the standard transformation can result in significant reduction in localization performance.

With the Word2Vec and GloVe transformations, both zone-prediction accuracy values average to 0.944 which yields about 19 % increase in the accuracy from that of the semi-supervised zone prediction using the standard transformation, and only around 3 % reduction in the accuracy from that of the supervised zone prediction. However, with the semi-supervised approach, we only used 105 survey (labelled) fingerprints (Box 8 in **Figure 1**) while in the supervised approach, 1,116 labelled fingerprints (Box 3 in **Figure 1**) were used. This means that, with the crowdsourcing approach using Word2Vec and GloVe transformations, significant survey effort can be reduced while maintaining similar level of localization performance.

For the Doc2Vec, Isomap, and PCA transformations, the accuracy values lie in the vicinity of those from the standard transformation, which means that apart from the benefit of lower memory-requirement from the distributed representation, these transformations might not offer much benefit to the crowdsourcing approach. On the other hand, the UMAP transformation yields significantly lower accuracy, which indicates that UMAP transformation is not appropriate for localization tasks. The unpromising zone-prediction performances from the Doc2Vec, Isomap, PCA, and UMAP transformations are attributed to the mediocre pseudo-labelling performances which, in turn, result from undesirable clustering characteristics such as higher amount mismatched fingerprints and/or high degree of overlapping.

Furthermore, the semi-supervised zone-prediction accuracy exhibits some correlation with the $\mathcal{C}$ value as shown in **Figure 6(b)** where the accuracy is inversely proportional to the $\mathcal{C}$ value. This relationship suggests that the zone-prediction accuracy, to a certain extend, depends on both the amount and the degree of the overlapping fingerprints simultaneously. However, the high variance in the y-values still suggest the dependence of the accuracy with other factors. Nevertheless, we suggest the use of the $\mathcal{C}$ value, along with the $\mathcal{F}^-$ value as screening metrics for potential representations that yields high localization performance.

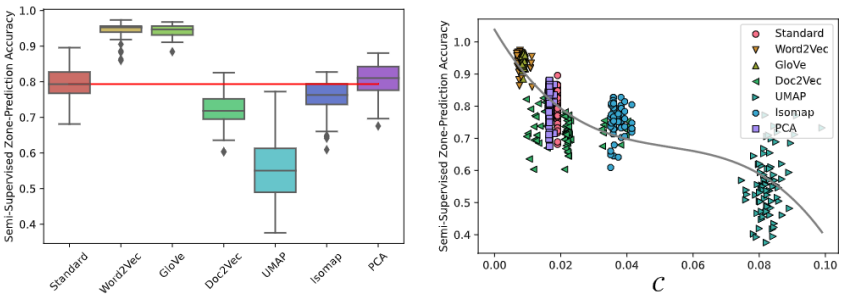


Figure 6 (a) Semi-supervised zone-prediction accuracy from different types of transformations. The horizontal lines represent the baseline values from the calculations with the standard fingerprint representations. (b) Plot of semi-supervised zone prediction accuracy with the $\mathcal{C}$ value. The $\mathcal{C}$ value takes into account the amount of the mismatched fingerprints and the degree of overlapping of the mismatched fingerprints. The gray line denotes the polynomial regression fit to the data.

Robustness against missing WAPs

The result from the robustness experiment is shown in **Figure 7** where the supervised zone-prediction accuracy is plotted as a function of the fraction of missing BSSID entries in the fingerprints; we denote this fraction as the *WAP-missing* fraction. In general, we observe the decrease in accuracy as the WAP-missing fraction increases; this behavior is expected as increasing WAP-missing fraction leads to less localization information. However, the text-embedding transformations, especially Word2Vec and GloVe transformations, exhibit excellent retention of accuracy even when half of the BSSID entries are missing. This result shows that the text-embedding transformations produce fingerprint representations that retain meaningful zone information and can offer increased robustness against data inhomogeneity.

Interestingly, the PCA transformation yields the result that are closer to those from the text-embedding transformations than the dimensional-reduction transformations. In fact, the PCA method is related to word-embedding methods from the fact that Word2Vec algorithm is closely related to matrix factorization [44,45]. This relationship could explain the substantially better performance from the PCA transformation compared with those from Isomap and UMAP transformations.

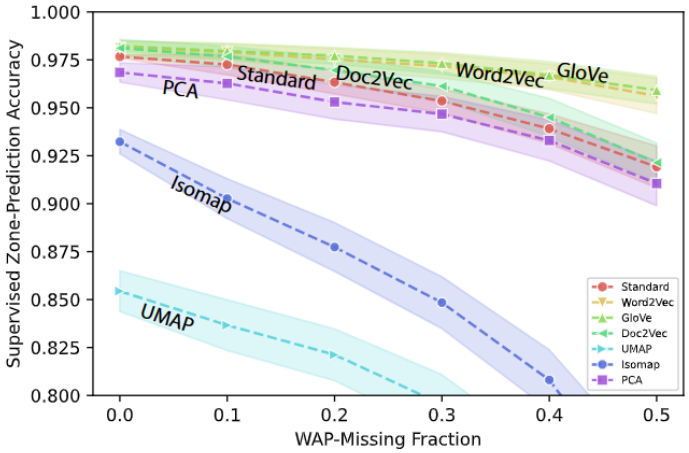


Figure 7 Plot of supervised zone-prediction accuracy as a function of fraction of missing BSSID entries in the fingerprints (WAP-missing fraction).

Parameter optimization

In this section, we discuss the optimized choices of parameters for Word2Vec and GloVe transformations with respect to the semi-supervised zone-prediction accuracy. For both transformations, the optimal parameters yield *vector_size* (number of dimensions of the resulting vector representation) of 50 and *windows_size* (a parameter relating to a number of context words) of 5; these values are intermediate values in our parameter ranges which means that there exist the optimum values of *windows_size* and *vector_size*. Other parameters include

fingerprint_repeat (number of BSSID lists generated per fingerprint) of 1 and *min_count* (minimum number of word occurrence to be included into the vocabulary) of 1. Lastly, the optimal text-embedding language task for the Word2Vec algorithm is CBOW. It should be noted that these parameters are optimized for the dataset in our experiment and more study needs to be performed in order to generalize this finding.

Conclusions

In this work, we proposed a novel technique to transform raw fingerprint data to distributed vector representations using non-contextual text-embedding methods from natural language processing. The resulting vector representations feature 3 benefits. First, the reduced dimension can be specified which results in lower memory requirement in the downstream tasks. Second, the representations contain no arbitrary null-value; thus avoiding the inclusion of a synthetic value in the data that can affect the performance of the downstream localization tasks in a non-transparent manner. Third, and most importantly, the representations result in improved pseudo labelling and semi-supervised zone-prediction performances. Also, the use of non-contextual techniques, as opposed to the contextual counterparts, is favorable in terms of computational requirement in model training and distributed-representation generation due to simpler model architectures (no deep learning) and no requirement for pre-trained model during distributed-representation generation. The transformation technique involves 3 steps. First is corpus construction; a corpus in our context contains a collection of fingerprints and each fingerprint is represented by a randomized space-separated list of detected BSSIDs whose numbers of occurrence in the list are proportional to their respective signal strength. The second step involves training the text-embedding models using the previously-generated corpus and the third step is the construction of the fingerprint representation from the trained text-embedding model.

In particular, we employed 3 commonly-used non-contextual text-embedding models which are Word2Vec, GloVe, and Doc2Vec in the text-embedding transformations and used the resulting vector representations in various localization tasks. We also compared the localization performances with those calculated using representations from the dimensional-reduction transformations; these transformation employs the commonly-used dimensional reduction techniques such as PCA, Isomap, and UMAP. Compared with the standard and dimensional-reduction transformations, our proposed the text-embedding transformations lead to vector representations that yield improved pseudo-labelling accuracy, semi-supervised zone-prediction accuracy, and robustness against missing WAPs. Specifically, the Word2Vec and GloVe transformations result in (a) approximately 41 % increase in the pseudo-labelling accuracy from that of the standard representation, (b) about 19 % increase in the semi-supervised zone prediction accuracy from that of the semi-supervised zone prediction using the standard transformation and (c) only around 3 % reduction in the accuracy from that of the supervised zone prediction. Therefore, the Word2Vec and GloVe transformations not only lead to improved localization performances compared with that from the standard transformation, but also make the semi-supervised localization approach competitive to the supervised approach. Along with the promising robustness property, the Word2Vec and GloVe transformations are the recommended transformation processes for constructing vector representations of fingerprints in crowdsourcing zone-level localization.

Acknowledgements

This work was supported by the Faculty of Engineering, Chiang Mai University and The Murata Science Foundation 2022, Thailand.

References

[1] F Zafari, A Gkelias and KK Leung. A survey of indoor localization systems and technologies. *IEEE Comm. Surv. Tutorials* 2019; **21**, 2568-99.

[2] S Sadowski and P Spachos. RSSI-based indoor localization with the internet of things. *IEEE Access* 2018; **6**, 30149-61.

[3] P Bolliger. Redpin - adaptive, zero-configuration indoor localization through user collaboration. *In*: Proceedings of the MELT'08: First ACM international workshop on Mobile entity localization and tracking in GPS-less environments, California. 2008, p. 55-60.

[4] AH Salamah, M Tamazin, MA Sharkas and M Khedr. An enhanced WiFi indoor localization system based on machine learning. *In*: Proceedings of the 2016 International Conference on Indoor Positioning and Indoor Navigation, Alcala de Henares, Spain. 2016.

[5] F Palumbo, P Barsocchi, S Chessa and JC Augusto. A stigmergic approach to indoor localization using bluetooth low energy beacons. *In*: Proceedings of the 2015 12th IEEE International Conference on Advanced Video and Signal Based Surveillance, Karlsruhe, Germany. 2015.

[6] C Jain, GVS Sashank, N Venkateswaran and S Markkandan. Low-cost BLE based indoor localization using RSSI fingerprinting and machine learning. *In*: Proceedings of the 2021 Sixth International Conference on Wireless Communications, Signal Processing and Networking, Chennai, India. 2021.

[7] L Yang, Y Chen, XY Li, C Xiao, M Li and Y Liu. Tagoram: Real-time tracking of mobile RFID tags to high precision using COTS devices. *In*: Proceedings of the MobiCom'14: 20th annual international conference on Mobile computing and networking, Hawaii. 2014, p. 237-48.

[8] L Mainetti, L Patrono and I Sergi. A survey on indoor positioning systems. *In*: Proceedings of the 2014 22nd International Conference on Software, Telecommunications and Computer Networks, Split, Croatia. 2014.

[9] LM Ni, D Zhang and MR Souryal. RFID-based localization and tracking technologies. *IEEE Wireless Comm.* 2011; **18**, 45-51.

[10] F Ge and Y Shen. Single-anchor ultra-wideband localization system using wrapped PDoA. *IEEE Trans. Mobile Comput.* 2021; **21**, 4609-23.

[11] W Zhao, A Goudar and AP Schoellig. Finding the right place: Sensor placement for UWB time difference of arrival localization in cluttered indoor environments. *IEEE Robot. Autom. Lett.* 2022; **7**, 6075-82.

[12] W Wu, L Shen, Z Zhao, M Li and GQ Huang. Industrial IoT and long short-term memory network enabled genetic indoor tracking for factory logistics. *IEEE Trans. Ind. Informat.* 2022; **18**, 7537-48.

[13] S He and SHG Chan. Wi-fi fingerprint-based indoor positioning: Recent advances and comparisons. *IEEE Comm. Surv. Tutorials* 2015; **18**, 466-90.

[14] AM Hossain, HN Van, Y Jin and WS Soh. Indoor localization using multiple wireless technologies. *In*: Proceedings of the IEEE International Conference on Mobile Adhoc and Sensor Systems, Pisa, Italy. 2007.

[15] P Kriz, F Maly and T Kozel. Improving indoor localization using bluetooth low energy beacons. *Mobile Inform. Syst.* 2016; **2016**, 2083094.

[16] Y Zhuang, Z Syed, Y Li and N El-Sheimy. Evaluation of two wifi positioning systems based on autonomous crowdsourcing of handheld devices for indoor navigation. *IEEE Trans. Mobile Comput.* 2015; **15**, 1982-95.

[17] W Sun, M Xue, H Yu, H Tang and A Lin. Augmentation of fingerprints for indoor wifi localization based on Gaussian process regression. *IEEE Trans. Veh. Tech.* 2018; **67**, 10896-905.

[18] SH Jung, BC Moon and D Han. Unsupervised learning for crowdsourced indoor localization in wireless networks. *IEEE Trans. Mobile Comput.* 2015; **15**, 2892-906.

[19] JEV Engelen and HH Hoos. A survey on semi-supervised learning. *Mach. Learn.* 2020; **109**, 373-440.

[20] A Haider, Y Wei, S Liu and SH Hwang. Pre-and post-processing algorithms with deep learning classifier for wi-fi fingerprint-based indoor positioning. *Electronics* 2019; **8**, 195.

[21] J Torres-Sospedra, R Montoliu, A Martínez-Usó, JP Avariento, TJ Arnau, M Benedito-Bordonau and J Huerta. UJIIndoorLoc: A new multi-building and multi-floor database for WLAN fingerprint-based indoor localization problems. *In*: Proceedings of the International Conference on Indoor Positioning and Indoor Navigation, Busan, Korea. 2014.

[22] T Mikolov, I Sutskever, K Chen, GS Corrado and J Dean. Distributed representations of words and phrases and their compositionality. *arXiv* 2013, https://doi.org/10.48550/arXiv.1310.4546

[23] J Pennington, R Socher and C Manning. GloVe: Global vectors for word representation. *In*: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, Doha, Qatar. 2014, p. 1532-43.
[24] Q Le and T Mikolov. Distributed representations of sentences and documents. *arXiv* 2014, https://doi.org/10.48550/arXiv.1405.4053.
[25] IT Jolliffe and J Cadima. Principal component analysis: A review and recent developments. *Phil. Trans. Math. Phys. Eng. Sci.* 2016; **374**, 20150202.
[26] Z Xu, B Huang, B Jia, W Li and H Lu. A boundary aware wifi localization scheme based on UMAP and KNN. *IEEE Comm. Lett.* 2022; **26**, 1789-93.
[27] JB Tenenbaum, VD Silva and JC Langford. A global geometric framework for nonlinear dimensionality reduction. *Science* 2000; **290**, 2319-23.
[28] SS Birunda and RK Devi. *A review on word embedding techniques for text classification*. Innovative Data Communication Technologies and Application. Springer, Singapore, 2021, p. 267-81.
[29] W Kim, S Yang, M Gerla and EK Lee. Crowdsource based indoor localization by uncalibrated heterogeneous wi-fi devices. *Mobile Inform. Syst.* 2016; **2016**, 4916563.
[30] Y Shu, Y Huang, J Zhang, P Coué, P Cheng, J Chen and KG Shin. Gradient-based fingerprinting for indoor localization and tracking. *IEEE Trans. Ind. Electron.* 2015; **63**, 2424-33.
[31] N Singh, S Choe and R Punmiya. Machine learning based indoor localization using wi-fi RSSI fingerprints: An overview. *IEEE Access* 2021; **9**, 127150-74.
[32] B Ezhumalai, M Song and K Park. An efficient indoor positioning method based on wi-fi RSS fingerprint and classification algorithm. *Sensors* 2021; **21**, 3418.
[33] P Bojanowski, E Grave, A Joulin and T Mikolov. Enriching word vectors with subword information. *Trans. Assoc. Comput. Ling.* 2017; **5**, 135-46.
[34] ME Peters, M Neumann, M Iyyer, M Gardner, C Clark, K Lee and L Zettlemoyer. Deep contextualized word representations. *In*: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, New Orleans, Louisiana. 2018, p. 2227-37.
[35] J Devlin, MW Chang, K Lee and K Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* 2018, https://doi.org/10.48550/arXiv.1810.04805.
[36] B Guo, W Zuo, S Wang, W Lyu, Z Hong, Y Ding, T He and D Zhang. Wepos: Weak-supervised indoor positioning with unlabeled wifi for on-demand delivery. *Proc. ACM on Interact. Mobile Wearable Ubiquitous Tech.* 2022; **6**, 54.
[37] X Sun, H Ai, J Tao, T Hu and Y Cheng. BERT-ADLOC: A secure crowdsourced indoor localization system based on BLE fingerprints. *Appl. Soft Comput.* 2021; **104**, 107237.
[38] ST Dumais. Latent semantic analysis. *Annu. Rev. Inform. Sci. Tech.* 2004; **38**, 188-230.
[39] LVD Maaten and G Hinton. Visualizing data using t-SNE. *J. Mach. Learn. Res.* 2008; **9**, 2579-605.
[40] F Anowar, S Sadaoui and B Selim. Conceptual and empirical comparison of dimensionality reduction algorithms (PCA, KPCA, LDA, MDS, SVD, LLE, ISOMAP, LE, ICA, t-SNE). *Comput. Sci. Rev.* 2021; **40**, 100378.
[41] R Rehurek and P Sojka. *Gensim-python framework for vector space modelling*. Vol 3. NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic, Czechia, 2011.
[42] F Pedregosa, G Varoquaux, A Gramfort, V Michel, B Thirion, O Grisel, M Blondel, P Prettenhofer, R Weiss, V Dubourg, J Vanderplas, A Passos, D Cournapeau, M Brucher, M Perrot and E Duchesnay. Scikit-learn: Machine learning in python. *J. Mach. Learn. Res.* 2011; **12**, 2825-30.
[43] T Chen and C Guestrin. XGBoost: A scalable tree boosting system. *arXiv* 2016, https://doi.org/10.1145/2939672.2939785
[44] Y Liu, M Ott, N Goyal, J Du, M Joshi, D Chen, O Levy, M Lewis, L Zettlemoyer and V Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv* 2019, https://doi.org/10.48550/arXiv.1907.11692
[45] Roberta. Hugging Face. Available at: https://huggingface.co/roberta-base/tree/main, accessed January 2023.

Appendixes

Appendix A: Computational-time comparison between Word2Vec and RoBERTa

In this section, we compare computational time from the tasks that utilize non-contextual and contextual text-embedding techniques to quantify possible increase in computational cost associating with contextual awareness. Specifically, we used Word2Vec as a candidate for the non-contextual technique and utilized a variant of BERT called RoBERTa [44], for the contextual counterpart. The task is to perform unsupervised learning (training) on the corpus generated from the training portion of the survey data (Box 3 in **Figure 1**) and to generate distributed representations (DRs) of the training data. The corpus was generated with one BSSID list per fingerprint (*fingerprint_repeat* = 1). The Word2Vec model was trained with CBOW option with the window size of 3 and the embedding-space dimension of 10. For the RoBERTa model, we utilized the implementation of the technique from the Huggingface library [45]. The training configurations are as follows: The training task is masked language modelling (MLM), the number of attention heads is 2, the number of hidden layers is 2, the hidden layer size is 10, the max position embedding is 512, the learning rate is 1e-5, the early stopping threshold is 1e-3 per 500 steps, the vocab size (after training the tokenizer on the corpus) is 363, which is roughly equal to the number of WAPs in the training data. It should be noted that we did not perform a fine-tuning process as we only focused on the computational time from unsupervised learning. The training loss and loss-change profiles are shown in **Figure 8**. The hardware used in this experiment is a desktop computer with 12th-Gen Intel Core i7-12700H and 32 GB of RAM. To make a fair comparison, we did not utilize GPU hardware because the Word2Vec implementation from Gensim library does not provide GPU support.

The resulting times used in training and DR generation from the Word2Vec and RoBERTa models are shown in **Table 2**. The training times for Word2Vec and RoBERTa models are 1.5 s and 3,629 s, respectively, yielding around 2,400 times faster for Word2Vec model. The DR generation time for the Word2Vec and RoBERTa models are 0.9 and 30 s, respectively, resulting in around 30 times faster for the Word2Vec model. Overall, the Word2Vec model performs significantly faster than the RoBERTa model, especially for the training process. The difference in time for the DR generation is much smaller between the 2 models, but the discrepancy is still substantial. The high computational expense of RoBERTa is not unexpected because this model, as well as other variants of BERT, comprises of a deep neural network with large number of training parameters; these models are typically trained with high-performance and specialized computing resources. On the other hand, the Word2Vec model consists of a shallow neural network with a much lower number of parameters, which results in fast computational time. In fact, in this work, we trained the Word2Vec model, as well as other non-contextual text-embedding models presented in this paper, with larger corpus and embedding dimension of up to 100 (10 times as many embedding dimension as that used in this experiment) on a consumer-grade laptop computer and the computational time is still in the order of minutes. Therefore, if the contextual awareness is not required, Word2Vec, and other non-contextual text-embedding models presented in this work, can be resource-efficient alternatives to the state-of-the-art language models such as BERT.

Table 1 Comparison of computational time used in training and distributed-representation (DR) generation processes with Word2Vec and RoBERTa models.

Model	Training time	DR generation time
Word2Vec	1.5 s	0.9 s
RoBERTa	3,629 s	30.6 s
Ratio (RoBERTa / Word2Vec)	2,419	34

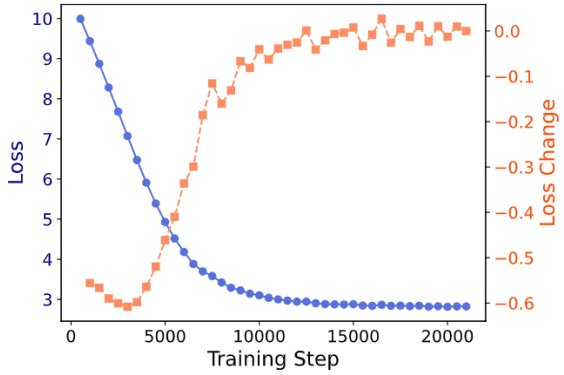


Figure 8 Plot of training loss (blue circles) and loss-change per 500 steps (orange squares) as functions of training time steps.

Appendix B: Zone-distinction test and zone-partitioning analysis

To achieve high zone-prediction accuracy, the zones should be partitioned in such a way that the fingerprint representations from each zone are distinguishable from those of the other zones by the ML classifiers. The challenge is that the distinction of fingerprints among different zones varies due to various factors such as distances between zones, physical obstructions, and WAP density. To this end, we propose a simple zone-distinction test to aid zone-partitioning design and to pinpoint problematic zones from the survey data. The test involves performing binary classification on supervised fingerprint data from a pair of zones to obtain classification metrics. These metrics can then be used as measures of distinguishability. If possible, the data should be collected by devices with similar Wi-Fi adaptors to reduce the effect of fingerprint variation on signal detectors.

To demonstrate the utility of the test, we performed zone-distinction tests on the surveyed data (Box 2 in **Figure 1**) where the subset of the data from similar mobile phones was used. The tests were performed on all possible pairs of zones, with a total of 210 pairs. The fingerprint representations were generated from the Word2Vec transformation. The classification was performed with logistic regression algorithm and F1-score is chosen as the classification metric (accuracy is a possible candidate as the data is balanced). It should be noted that for each test, we repeated the classification multiple times (10 times in this experiment) with different random seeds for the train-test data split to reduce the dependence of the test on the randomness in train-test data division.

Figure 9 shows the boxplots of the F1-scores from the zone-distinction tests; only the tests whose results contain F1-scores less than 1 are shown. Out of 210 tests, the 204 tests (not shown in the figure) whose results yield all F1-scores of 1 indicate that the fingerprints from the corresponding zones in the pairs are distinct enough such that no misclassification occurs; we designate the corresponding zones in the pairs as being completely-distinguishable. On the other hand, the remaining 6 tests (shown in the figure) whose results show some F1-scores lower than 1 indicate that the fingerprints from the corresponding zones in the pairs are not distinct enough and some misclassifications can occur; we refer to the corresponding zones in the pairs as being partially-distinguishable with the "distinguishability" performance described by the average of the F1-scores. The results from the tests show that the majority of the zone-pairs are completely distinguishable and even for the partially-distinguishable pairs, the distinguishability performance degradation is very minimal (the means of F1-scores are close to 1). These test results suggest that our zone-partitioning scheme is adequate for the survey site in terms of fingerprint distinguishability and such high distinguishability results in good classification performances for the supervised zone-prediction as shown in **Table 1**.

To investigate how zone distinction relates to physical space, we show the floor plans of the survey site in **Figure 10**; we also annotated the partially-distinguishable zone-pairs with the corresponding labels from **Figure 9**. First, we do not observe any partially-distinguishable zone-pair whose constituents are from different floors; this indicates that building floors provide complete zone distinction, at least for our survey site. Second, there is no partially-distinguishable zone-pair from the connecting zones (adjacent zone without obstruction), which indicates that sufficient spatial distance (around 10 meters in our case) can provide complete zone distinction. Third, we observe that the 6 partially-distinguishable zone-pairs are adjacent zones that share building walls; this indicates that building walls may not provide complete zone distinction. Nevertheless, in our case, the performance degradation is minimal and the effectiveness of the wall to provide zone distinction is still justified.

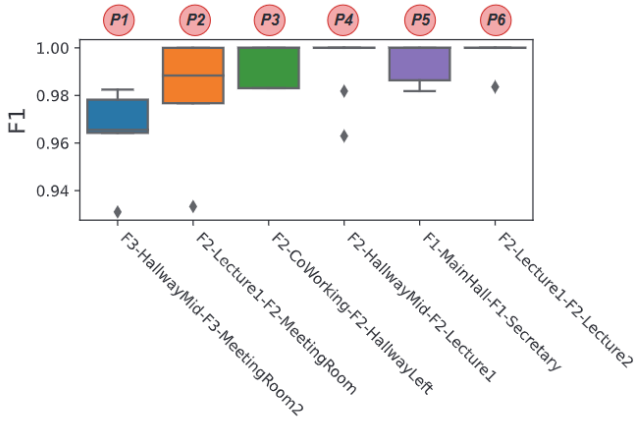


Figure 9 Boxplots of F1-scores from the zone-distinction tests. Out of 210 tests, 6 tests whose results contain F1-scores less than 1 are shown.

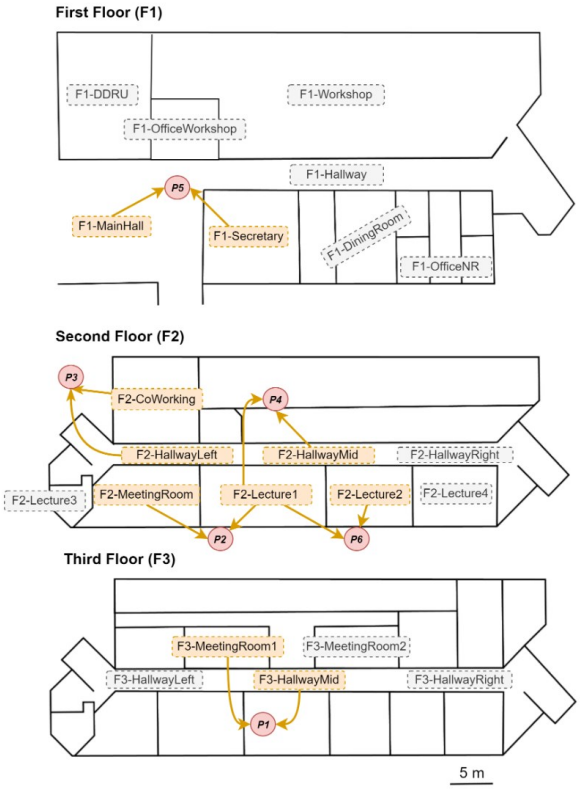


Figure 10 Floor plans of the survey site. The zones from the 6 partially-distinguishable zone-pairs are annotated with the corresponding labels from **Figure 9**.

Appendix C: Machine-learning classification-algorithm parameters

We detail the choices and range of parameters for optimizing ML classifiers. The parameter for logistic regression is *C* (regularization parameter). The parameters for support vector machine are *C* and *gamma* (kernel coefficient). The parameters for k-nearest neighbors are *n_neighbors* (number of nearest data that join majority vote) and *weights* (weight function). The parameters for XGBoost are *gamma* (regularization parameter) and *max_depth* (maximum tree depth). The parameters for random forest (RF) are *max_depth* and *n_estimators* (number of trees). The parameter for decision tree (DT) is *max_depth*. The ranges of parameters are summarized in **Table 3**.

Table 2 Choices and ranges of parameters used to optimize machine-learning classifiers.

Algorithm	Parameter	Range
Logistic regression	C	0.1, 0.5, 1, 5, 10
Support vector machine	C	0.1, 1, 2, 5,10
	gamma	scale, auto
K-nearest neighbor	n_neighbors	3, 5, 7, 9
	weights	uniform, distance
XGBoosted	gamma	0.5, 1
	max_depth	5, 10
Random forest	max_depth	5, 10, 15
	n_estimators	100, 150, 200
Decision tree	max_depth	3, 5, 7, 10, 15

Appendix D: Fingerprint-transformation parameters

We discuss the choices and ranges of parameters of different types of fingerprint transformation. The parameters for the text-embedding transformations are *vector_size* (number of dimensions of the resulting vector representation), *fingerprint_repeat* (number of BSSID lists generated per fingerprint), *task* (type of text-embedding language tasks such as CBOW/SG or DM/DBOW), *window_size* (a parameter relating to a number of context words), and *min_count* (minimum number of word occurrence to be included into the vocabulary). The parameters for the dimensional reduction transformation and the standard transformation include *vector_size*, *null_value* (choice of the null value), and *n_neighbors* (balance between local and global structure preservation). Also, we repeated the calculation multiple times (*repeat* times) in order to take into account the randomness in the processing pipeline. The summarized list of parameters is shown in **Table 4** and the ranges of the parameters are tabulated in **Table 5**. It should be noted that to reduce the number of calculation, we only varied one parameter at a time and the rest of the parameters are fixed at the default values (denoted with asterisks).

Table 3 Choices of parameters of the fingerprint transformations.

Transformation	Parameter
Word2Vec, GloVe, Doc2Vec	*vector_size, fingerprint_repeat, task, window_size, min_count*
PCA	*vector_size, null_value*
UMAP, ISOMAP	*n_neighbors, vector_size, null_value*
Standard	*null_value*

Table 5 Ranges of parameters of the fingerprint transformations. The asterisk indicates the default value.

Parameter	Range
vector_size	25, 50, 100*
fingerprint_repeat	1*, 2
task(Word2Vec)	CBOW*, SG
task (Doc2Vec)	DM*, DBOW
window_size	1, 3, 5*, 7
min_count	1*, 3, 5
null_value	1,000*, 900, 800, 700, 600, 500
n_neighbors	5, 10, 15*, 20, 25, 30
repeat	5