

# An Improved and Low-dimensional Fingerprint-based Localization Method in Collocated Massive MIMO-OFDM Systems

Seyedeh Samira Moosavi (✉ [seyedeh-samira.moosavi.1@ulaval.ca](mailto:seyedeh-samira.moosavi.1@ulaval.ca))

Universite Laval Faculte des sciences et de genie

Paul Fortier

Universite Laval Faculte des sciences et de genie

---

## Research Article

**Keywords:** Collocated massive MIMO (CM-MIMO), Fingerprinting (FP), Machine learning (ML), Affinity propagation clustering (APC), Gaussian process regression (GPR).

**Posted Date:** March 23rd, 2021

**DOI:** <https://doi.org/10.21203/rs.3.rs-187259/v1>

**License:**  This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

---

---

# An Improved and Low-dimensional Fingerprint-based Localization Method in Collocated Massive MIMO-OFDM Systems

Seyedeh Samira Moosavi · Paul Fortier

Received: date / Accepted: date

**Abstract** Localization has drawn significant attention in 5G due to the fast-growing demand for location-based service (LBS). Massive multiple-input multiple-output (M-MIMO) has been introduced in 5G as a powerful technology due to its evident potentials for communication performance enhancement and localization in complicated environments. Fingerprint-based (FP) localization are promising methods for rich scattering environments thanks to their high reliability and accuracy. The Gaussian process regression (GPR) method could be used as an FP-based localization method to facilitate localization and provide high accuracy. However, this method has high computational complexity, especially in large-scale environments. In this study, we propose an improved and low-dimensional FP-based localization method in collocated massive MIMO orthogonal frequency division multiplexing (OFDM) systems using principal component analysis (PCA), the affinity propagation clustering (APC) algorithm, and Gaussian process regression (GPR) to estimate the user's location. Fingerprints are first extracted based on instantaneous channel state information (CSI) by taking full advantage of the high-resolution angle and delay domains. First, PCA is used to pre-process data and reduce the feature dimension. Then, the training fingerprints are clustered using the APC algorithm to increase prediction accuracy and reduce computation complexity. Finally, each cluster's data distribution is accurately modelled using GPR to provide support for further localization. Simulation results reveal that the proposed

---

Seyedeh Samira Moosavi  
Department of Electrical and Computer Engineering, Laval University, Quebec, QC G1V 0A6, Canada  
Tel.: +1 418-573-3316  
E-mail: seyedeh-samira.moosavi.1@ulaval.ca

Paul Fortier  
Department of Electrical and Computer Engineering, Laval University, Quebec, QC G1V 0A6, Canada  
Tel.: +1 418- 656-2131  
E-mail: paul.fortier@gel.ulaval.ca

method improves localization performance significantly by reducing the location estimation error. Additionally, it reduces the matching complexity and computational complexity.

**Keywords** Collocated massive MIMO (CM-MIMO) · Fingerprinting (FP) · Machine learning (ML) · Affinity propagation clustering (APC) · Gaussian process regression (GPR).

## 1 Introduction

For supporting users with high quality of service, recognizing their location is essential. Therefore, location information has recently become an important characteristic to drive location and context-aware services in wireless communications [1]. Nevertheless, it is a notoriously challenging problem to provide precise and reliable location information of user equipments (UEs) by using multipath propagation in wireless communications [2].

Currently, with the advent of the fifth-generation (5G) of wireless communications, multiple-input multiple-output (MIMO) orthogonal frequency division multiplexing (OFDM) systems have received great attention from the localization community due to their potential for improving user-localization accuracy [3]. Indeed, by employing a very large number of antennas at the base station, massive MIMO-OFDM, besides improving spectral and energy efficiency [4], can obtain a higher multipath resolution in the angle and delay domains to provide high-accuracy localization for location-based services [3]. Furthermore, the resilience against small scale fading is provided through processing measurements across the massive array [5]. Therefore, the potential of massive MIMO-OFDM to support location-based services is one of the main economic drivers of 5G wireless communications [6].

Global positioning system (GPS) is the most well-known localization technique. Although it provides a precise estimation of mobile terminal's (MT's) position, it suffers a loss of accuracy in areas where there is no direct line of sight (LOS) between the transmitter and receiver [7]. Consequently, there are range-based localization methods [8, 9, 10, 11], which are based on radio signal information received from MTs such as angle-of-arrival (AoA), time-of-arrival (ToA), and received signal strength (RSS). However, the AoA-based methods deal with the non-line-of-sight (NLOS) error [8]; in ToA-based methods, the base stations (BSs) need to be synchronized [9] and using RSS has coarse range estimation error in complex environments [12].

Another approach is fingerprinting (FP), which has achieved extensive attention for localization in recent years due to its promising performance in complex multipath environments. It is flexible and can be used in many systems such as WiFi networks [13] and in systems where channel state information (CSI) [14] is used as the location fingerprint. Unlike traditional methods, it uses massive data to train a model and then it is employed for localization [3]. For this purpose, many machine learning (ML) and deep learning (DL) algorithms are used. However, the ML and DL algorithms often have to

deal with data complexity due to the high dimensionality of the data. Some solutions such as using dimensionality algorithms have been proposed to reduce complexity [15]. By considering all benefits of the massive MIMO-OFDM system and the FP methods, integrating 5G and fingerprints can be a good solution for localization in rich scattering environments.

Localization with massive MIMO is still in its nascent stage. In [16], AoA is estimated precisely in massive MIMO systems employing very large uniform rectangular arrays (URAs). The authors in [17] consider the combined information of time delay, angle of delay (AoD), and AoA for localization of MTs in a massive MIMO system. In [18], a compressed sensing approach is proposed to determine MT's location directly from data acquired, such as ToA information at multiple massive MIMO BSs. The Gaussian process regression (GPR) method is employed in [19] to estimate location of users using a vector of RSS, which is considered as a fingerprint in a DM-MIMO system. A fingerprinting method is presented in [20], wherein an angle-delay channel power matrix (ADCPM) is first extracted as a fingerprint. Then a system is employed for clustering the fingerprints with a mathematical model, such as the joint angle-delay similarity coefficient (JADSC). However, efficiency is lost because the effective range of the JADSC is dependent on the scatterers' density.

In the present study, we consider a collocated massive MIMO-OFDM system where the BS is equipped with a large array of antennas to serve single antenna UEs over their coverage area. We propose a machine learning localization method using the CSI of collocated massive MIMO-OFDM system to achieve accurate localization resolution. For this purpose, the fingerprints are extracted from the known channel estimations. Then a combination of principal component analysis (PCA) and t-distributed stochastic neighbor embedding (t-SNE) is used to pre-process data, reduce the dimensions of the data features and visualize the data. Later, an affinity propagation clustering (APC) and Gaussian process regression (GPR)-based fingerprinting positioning system is presented, which consists of two phases: the training phase and the online positioning phase. In the first phase, a large area is divided into small clusters using an optimal clustering method based on the APC algorithm. The data distribution within each cluster is precisely modeled using the GPR method. Then, in the positioning phase, the user's cluster is first specified by a cluster identification algorithm based on a multi-layer perceptron (MLP) neural network. Its location is estimated using the GPR model of that cluster.

The remainder of this paper is structured as follows. In Section II, we present the system model of collocated massive MIMO-OFDM. In Section III, we describe the proposed machine learning localization method in detail. The results obtained through simulations are shown in Section IV. Finally, conclusions are provided in Section V.

## 2 System Description

### 2.1 Massive MIMO-OFDM Channel Model

In this section, we consider the uplink of a multi-user collocated massive MIMO-OFDM system. In this system, we aim to localize the  $K$  single-antenna UEs, which are randomly distributed in the coverage area. The UEs transmit signals to the BS, which is equipped with  $M$  antennas in the form of a uniform linear array (ULA), through  $F \gg 1$  different scattering paths, as shown in Fig. 1.

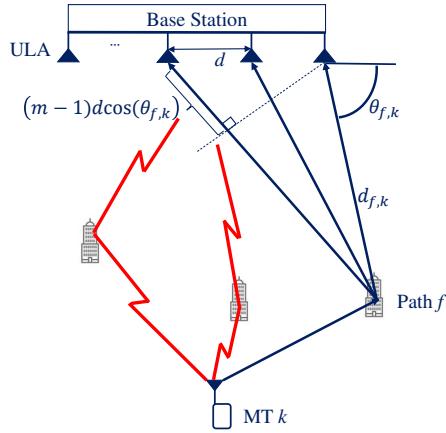


Fig. 1: The wireless channel from an arbitrary user  $k$  to the BS.

According to Fig. 1, the wireless signals are propagated through multiple scattering paths with AoA  $\theta_{f,k} \in (0, \pi)$ , and the distance  $d_{f,k}$  between the UE's antenna and the first receive antenna of the  $f$ th path. Assume that at this multipath wireless channel, the CSI is known at the BS through the uplink channel. The channel impulse response (CIR) vector associated with the  $f$ th path of the  $k$ th user is given by [20]

$$\mathbf{a}_{f,k} = \omega_{f,k} \mathbf{e}(\theta_{f,k}) \quad (1)$$

where  $\omega_{f,k} \sim \mathcal{CN}(0, \sigma_{f,k})$  is the complex attenuation of the  $f$ th path and  $\mathbf{e}(\theta) \in \mathbb{C}^{M \times 1}$  is the array response vector related to the AoA,  $\theta$ , and is given [20] by

$$\mathbf{e}(\theta) = \left[ 1, e^{-j2\pi \frac{d \cos(\theta)}{\lambda}}, \dots, e^{-j2\pi \frac{(M-1)d \cos(\theta)}{\lambda}} \right]^T \quad (2)$$

where  $\lambda$  and  $d$  represent the carrier wavelength and the space between two antennas, respectively. For OFDM systems, the channel frequency response

(CFR) of the  $n$ th subcarrier can be defined as the summation of the time domain CIRs with different delays [20] as

$$\mathbf{h}_{k,n} = \sum_{f=1}^F \omega_{f,k} \mathbf{e}(\theta_{f,k}) e^{-j2\pi \frac{n u_{f,k}}{N_{sc}}} \quad (3)$$

where  $u_{f,k} = \lfloor \frac{\tau_{f,k}}{T_s} \rfloor$  and  $\frac{u_{f,k}}{N_{sc}}$  indicates the temporal propagation delay corresponding to the  $f$ th path.  $\tau_{f,k} = \frac{d_{f,k}}{v}$  is the ToA of each path,  $v$  is the speed of light,  $T_s$  and  $N_{sc}$  are the sample interval and the number of subcarriers in the OFDM system, respectively. Then the overall CFR matrix is defined as

$$\mathbf{H}_k = [\mathbf{h}_{k,0}, \mathbf{h}_{k,1}, \dots, \mathbf{h}_{k,N_{sc}-1}] \quad (4)$$

## 2.2 Fingerprint Extraction

For creating a fingerprint, it is required to extract several characteristics that are constant in the FP method. Therefore, wide-sense stationary features from the instantaneous CSI are considered as a fingerprint. Discrete Fourier transform (DFT) operations are employed to establish a simple mapping from the CFR matrix to a sparse structure [20], and the angle-delay channel power matrix (ADCPM) is given as

$$\boldsymbol{\Omega}_k = \mathbf{F}^H \mathbf{H}_k \mathbf{G}^* \quad (5)$$

where  $\mathbf{F} \in \mathbb{C}^{M \times M}$  indicates a phase-shifted discrete Fourier transform (DFT) matrix represented by  $[\mathbf{F}]_{i,j} \triangleq \frac{1}{\sqrt{M}} e^{-j2\pi \frac{i(j-M/2)}{M}}$ .  $\mathbf{G} \in \mathbb{C}^{N_{sc} \times N_g}$  is defined as  $[\mathbf{G}]_{i,j} \triangleq \frac{1}{\sqrt{N_{sc}}} e^{-j2\pi \frac{ij}{N_{sc}}}$  and  $N_g$  is the number of guard subcarriers. The left multiplication operator,  $\mathbf{F}^H$ , and the right multiplication operator,  $\mathbf{G}^*$ , cause the frequency domain CFR map to the angle and the delay domain, respectively. The complex gain related to the  $i$ th AoA and the  $j$ th ToA is represented by the  $(i,j)$ th element of the ADCPM fingerprint which is defined as

$$\mathbf{R}_k = \mathbb{E}\{\boldsymbol{\Omega}_k \odot \boldsymbol{\Omega}_k^*\} = \begin{bmatrix} r_{1,1} & r_{1,2} & \cdots & r_{1,N_g} \\ r_{2,1} & r_{2,2} & \cdots & r_{2,N_g} \\ \vdots & \vdots & \ddots & \vdots \\ r_{M,1} & r_{M,2} & \cdots & r_{M,N_g} \end{bmatrix} \quad (6)$$

where  $\odot$  denotes the Hadamard product so that  $[\mathbf{R}_k]_{i,j} = \mathbb{E}\{|\boldsymbol{\Omega}_k|_{i,j}|^2\}$ . As shown in Fig. 2, the ADCPM represents the AoA, the ToA, and the channel power of each path related to the scattering environment of user  $k$ .

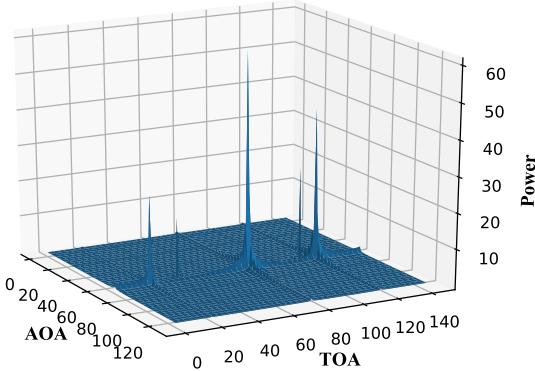


Fig. 2: Example of ADCPM with  $M = 128$  and  $N_g = 144$ .

### 3 Fingerprinting Localization Method Based on Clustering and Regression

Fig. 3 shows an overview of the architecture of the proposed localization method. The ADCPM is extracted as a fingerprint from the channel estimation result known to the BS. The remaining structure of the proposed location estimation method consists of two phases: the training phase and the estimation phase.

In the training phase, the training data are collected from a grid of known location reference points (RPs). Then, the fingerprints are labeled with their corresponding location coordinates, and the original dataset is created. The proposed method is based on four principal component blocks: i) pre-processing, which is used to standardize, reduce dimensions of the features, and visualize data, ii) clustering, where similar data are grouped and stored in the database, iii) cluster identification, which is employed for coarse localization, and iv) regression, where an accurate model is created for each cluster based on their similar data distribution. We will describe each block in detail in the following subsections.

During the estimation phase, we randomly select some unknown locations to collect the testing data. Similar to the training phase, the collected data is processed using the same method to generate fingerprints. Then, its corresponding cluster (cluster ID) is first determined using a cluster identification algorithm. Accordingly, the small region in which the new user is most probably located is identified. Lastly, accurate position estimation is applied using the GPR model of the corresponding cluster to improve the accuracy of the location estimation.

### 3.1 Offline Phase

Let's assume there are  $L$  training RPs. The ADCPM fingerprint of each RP is first converted to a one-dimensional array such that we have

$$\mathbf{r}_l = [r_{1,1}, \dots, r_{1,N_g}, r_{2,1}, \dots, r_{2,N_g}, \dots, r_{M,1}, \dots, r_{M,N_g}] \quad (7)$$

Therefore, each sample  $\mathbf{r}_l$  has  $B = M \times N_g$  attributes, i.e.,  $A_b, b = 1, \dots, B$ , and the radio map is defined as

$$\mathbf{R} = [\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_L]^T \quad (8)$$

where each row of  $\mathbf{R}$  indicates a  $B$ -dimensional training vector  $\mathbf{r}_l$  related to the training  $x$ -coordinates  $x_l$  and the training  $y$ -coordinates  $y_l$ ,  $l = 1, \dots, L$ . Therefore, the corresponding  $L \times 1$  training  $x$ -coordinates  $\mathbf{x}$  and  $L \times 1$  training  $y$ -coordinates  $\mathbf{y}$  are defined as follows

$$\begin{aligned} \mathbf{x} &= [x_1, x_2, \dots, x_L]^T, \\ \mathbf{y} &= [y_1, y_2, \dots, y_L]^T \end{aligned} \quad (9)$$

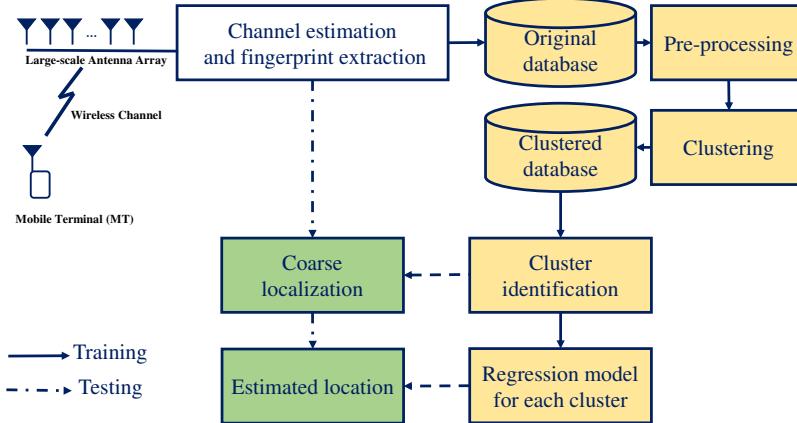


Fig. 3: Overview of the proposed position estimation scheme.

#### 3.1.1 Pre-processing

Data pre-processing is the core stage in data mining and machine learning. Data transformation, dimensionality reduction, and data visualization are well-known techniques in this step [21].

– *Data transformation*

Standardization is the main pre-processing step in data mining, where attributes values are standardized from different dynamic ranges to a particular range [21]. Standardized datasets learn the training model faster and have better quality, efficiency, and accurate clustering results [22]. Based on the nature of the datasets for the analysis, it is essential to select an appropriate standardization method. In our case, for each training vector  $\mathbf{r}_z$ , standardization of each value,  $r_{l,b}, b = 1, \dots, B$ , of attribute  $A_b, b = 1, \dots, B$  is done as follows

$$s_{l,b} = \frac{r_{l,b} - \mu_b}{\sigma_b} \quad (10)$$

where  $\mu_b$  and  $\sigma_b$  represent the mean and the standard deviation of attribute  $A_b$ . After standardization, the training vectors  $\mathbf{r}_l$  converts to  $\mathbf{s}_l$  where  $l = 1, \dots, L$ , and the radio map  $\mathbf{R}$  is changed into  $\mathbf{S} \in \mathbb{R}^{l \times B}$ .

– *Dimensionality reduction*

The number of attributes that describe the dataset is defined as its dimensionality. These dimensions are represented as columns. Dimension reduction involves reducing the number of these columns to obtain a reduced or “compressed” representation of the original data, especially with the presence of curse dimensionality. If the columns are correlated, there is some redundant information that affects the training model results [23]. Therefore, it is crucial to use dimensionality reduction techniques to avoid overfitting and reduce the model’s complexity. In our case study, we applied principal components analysis (PCA) for dimensionality reduction. PCA is a well-established method for dimensionality reduction. PCA is an orthogonal linear transformation that maps the given dataset  $\mathbf{R}$  present in a  $B$ -dimensional space to a  $D$ -dimensional space such that  $D < B$ , while retaining the most relevant information [24]. Before clustering, using PCA is a powerful method for clustering high dimensional datasets. The procedure of PCA is explained in the following. After data standardization, which was explained in the previous section, i.e., the data transformation section, the covariance matrix of radio map  $\mathbf{S}$  is computed by

$$\Sigma = (\mathbf{S})^T \mathbf{S} \quad (11)$$

where  $\Sigma \in \mathbb{R}^{B \times B}$ . In order to find the principal components in the new feature space, we need to compute the eigenvalues  $\lambda_i$  and eigenvectors  $\mathbf{e}_i$  of the covariance matrix  $\Sigma$ , satisfying  $\Sigma \mathbf{e}_i = \lambda_i \mathbf{e}_i$ . For computing eigenvalues  $\lambda_i$ , we have

$$(\Sigma - \lambda_i \mathbf{I}) \mathbf{e}_i = 0 \quad (12)$$

Since  $\mathbf{e}_i$  is a non-zero vector, (12) can be equal to zero if  $\det(\Sigma - \lambda_i \mathbf{I}) = 0$ . Let  $I$  eigenvalues in descending order form the diagonal matrix  $\Lambda$ :

$$\Lambda = \text{diag}[\lambda_1, \lambda_2, \dots, \lambda_I] \quad (13)$$

To discover the eigenvectors of  $\Sigma$  using eigenvalue decomposition, the matrix of eigenvectors is defined as

$$\mathbf{E} = [\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_I] \quad (14)$$

where  $\mathbf{E} \in \mathbb{R}^{B \times I}$ . With the  $D$  largest eigenvalues of  $\Sigma$ , the eigenvector matrix is defined as

$$\mathbf{E} = [\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_D] \quad (15)$$

where  $\mathbf{E} \in \mathbb{R}^{B \times D}$  and its columns represent the principal components (the new dimensions) which are orthogonal to each other and arranged in decreasing order of variance. More precisely, the first eigenvector  $\mathbf{e}_1$  is the direction in which the data varies the most, the second eigenvector  $\mathbf{e}_2$  is the direction of greatest variance among those that are orthogonal (perpendicular) to the first eigenvector, and so on. The last step is transforming the standardized radio map to new dimensions radio map which is computed by

$$\mathbf{N} = \mathbf{SE} \quad (16)$$

where  $\mathbf{N} \in \mathbb{R}^{L \times D}$ .

#### – Data visualization

The t-distributed stochastic neighbor embedding (t-SNE) algorithm is an innovative probabilistic method for data visualization. It is well known in machine learning due to its remarkable ability to transform high dimensional data to lower dimensions by preserving the neighborhood structure of the dataset [25, 26]. If there are  $L$  standardized training data points, in the t-SNE algorithm the similarity of data point  $\mathbf{s}_l$  to  $\mathbf{s}_{l'}$ , where  $1 \leq l, l' \leq L$  and  $l \neq l'$ , is a conditional probability given [25] by

$$p_{l'|l} = \frac{\exp\left(\frac{-\|\mathbf{s}_l - \mathbf{s}_{l'}\|^2}{2\sigma_l^2}\right)}{\sum_{l \neq l''} \exp\left(\frac{-\|\mathbf{s}_l - \mathbf{s}_{l''}\|^2}{2\sigma_l^2}\right)} \quad (17)$$

$p_{l'|l}$  is high for nearby data points, whereas it will be relatively small for widely separated data points.  $\sigma_l$  is associated with a predefined input parameter *Perp* known as “perplexity” and can be loosely interpreted as the number of effective neighbors that each data point has and is defined [27] as

$$\text{Perp}(p_l) = 2^{(H(p_l))} \quad (18)$$

where  $H(p_l)$  is the Shannon entropy, which is given by

$$H(p_l) = - \sum_{l'} p_{l'|l} \log_2 p_{l'|l} \quad (19)$$

Then the  $L \times L$  similarity matrix  $\mathbf{P}_{High}$  in the original high dimensional space is formed and its entries are defined [25] as

$$p_{l,l'} = \frac{p_{l|l'} + p_{l'|l}}{2L} \quad (20)$$

The t-SNE algorithm tends to learn a  $D$ -dimensional map  $\{\mathbf{n}_1, \dots, \mathbf{n}_L\}$ ,  $\mathbf{n}_l \in \mathbb{R}^{1 \times d}$  of the original data that reflects the similarities  $p_{l,l'}$  as well as possible. For this purpose, a Student t-distribution with one degree of freedom is used. Using this distribution, a  $L \times L$  similarity matrix  $\mathbf{Q}_{Low}$  in low dimensional space is defined whose entries are given by

$$q_{l,l'} = \frac{\left(1 + \|\mathbf{n}_l - \mathbf{n}_{l'}\|^2\right)^{-1}}{\sum_{l \neq l''} \left(1 + \|\mathbf{n}_l - \mathbf{n}_{l''}\|^2\right)^{-1}} \quad (21)$$

The t-SNE for finding the projections of the input data  $\mathbf{s}_l$  in lower dimension as  $\mathbf{n}_l$ , employs a gradient-based technique to minimizes the Kullback–Leibler distance, which is defined as the cost function between  $\mathbf{P}_{High}$  and  $\mathbf{Q}_{Low}$  [25] and is given by

$$C = KL(\mathbf{P}_{High} || \mathbf{Q}) = \sum_{l=1}^L \sum_{l'=1}^L p_{l,l'} \log \frac{p_{l,l'}}{q_{l,l'}} \quad (22)$$

The gradient of the Kullback-Leibler distance between  $\mathbf{P}_{High}$  and the Student-t based joint probability distribution  $\mathbf{Q}_{Low}$  is given [25] by

$$\frac{\delta C}{\delta \mathbf{n}_l} = 4 \sum_{l'} (p_{l,l'} - q_{l,l'}) (\mathbf{n}_l - \mathbf{n}_{l'}) \left(1 + \|\mathbf{n}_l - \mathbf{n}_{l'}\|^2\right)^{-1} \quad (23)$$

### 3.1.2 Clustering and Clustering Validation

The training data clustering and cluster validation procedure are presented in Fig. 4. After preprocessing, let us define  $\mathbf{N}$  as the compressed radio map whose  $D$ -dimensions rows are  $\mathbf{n}_l$ ,  $l = 1, \dots, L$ . First a clustering algorithm is employed. Then, the quality of clustering is evaluated by a cluster validity index. In our method, we consider a different clustering algorithm which is explained in the following.

#### – Affinity propagation clustering

The affinity propagation clustering (APC) algorithm [28] divides the training fingerprints into clusters by allocating each fingerprint sample an equal chance to become a cluster-head (CH) [29]. Let us assume that the radio map  $\mathbf{N}$  obtained after preprocessing on the training phase is represented as follows

$$\mathbf{N} = [\mathbf{n}_1, \mathbf{n}_2, \dots, \mathbf{n}_L]^T \quad (24)$$

where, each row of  $\mathbf{N}$  is the fingerprint of the  $l$ th RP.

To divide the training fingerprints into several clusters, the affinity propagation clustering (APC) algorithm [28] is employed by allocating each fingerprint sample an equal chance to become an exemplar. In K-means clustering, the number of output clusters and the corresponding random

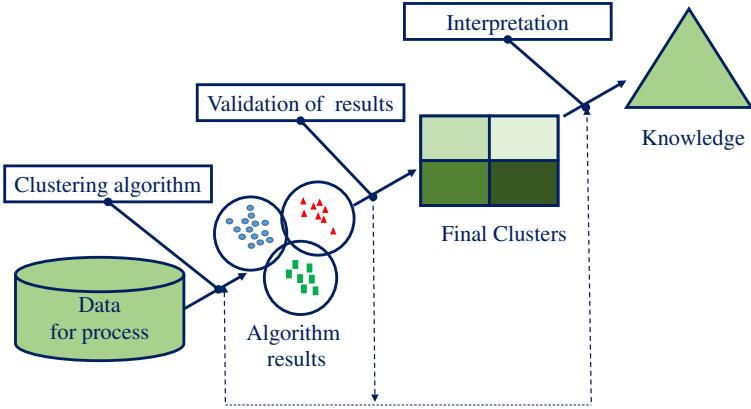


Fig. 4: The pre-processing scenarios.

set of initial exemplars must be identified in advance [29]. Therefore, APC outperforms K-means clustering because it benefits from the initialization independent feature and better selection of the cluster exemplar.

The APC algorithm requires two types of real-valued input to divide  $\mathbf{N}$  into clusters: the similarities matrix  $\mathbf{S}_{Sim}$  and the preference  $pref$ . A pairwise similarity  $s(\mathbf{n}_l, \mathbf{n}_{l'})$  (for  $l \neq l'$ ) is used to indicate the appropriateness of  $\mathbf{n}_l$  to be selected as the exemplar with respect to  $\mathbf{n}_{l'}$ . Since we aim to minimize the squared error, the similarity calculation in the APC is based on the negative squared error (Euclidean distance) as follows:

$$s(\mathbf{n}_l, \mathbf{n}_{l'}) = -\|\mathbf{n}_l - \mathbf{n}_{l'}\|^2 \quad (25)$$

where  $1 \leq l, l' \leq L$  and  $l \neq l'$ . Also  $pref$  is defined as follows

$$pref = \text{median}(s(\mathbf{n}_l, \mathbf{n}_{l'})) \quad (26)$$

To evaluate the quality of clustering results, a clustering index validity such as the silhouette (SI) is considered [30]. The SI is a well-known measure of how similar a training vector  $\mathbf{n}_l, l = 1, \dots, L$  is to its own cluster (cohesion) compared to other clusters (separation). It is then averaged over all training vectors and is defined [31] as

$$SI = \frac{1}{L} \sum_{l=1}^L \frac{d(\mathbf{n}_l) - f(\mathbf{n}_l)}{\max\{d(\mathbf{n}_l), f(\mathbf{n}_l)\}} \quad (27)$$

where  $d(\mathbf{n}_l)$  is the average distance between  $\mathbf{n}_l$  and all training vectors in other clusters and  $f(\mathbf{n}_l)$  is the average distance between  $\mathbf{n}_l$  and all training vectors in the same cluster.

### 3.1.3 Cluster Identification

In this section, to identify the corresponding cluster of each fingerprint, we use an artificial neural network (ANN) classifier. The ANN that we use is the multi-layer perceptron (MLP), which is employed and evaluated for coarse localization. For this purpose, a subset of the clustered training data points is selected as a validation dataset, and it is supposed that their cluster IDs are unknown. This subset is used to obtain the accuracy of the cluster identification model. Let us suppose there are  $D$  attributes after using dimensionality reduction and 20% of the training data are selected as validation dataset. The details of this cluster identification algorithm are described in the following.

- *Multi-layer Perceptron (MLP) Neural Networks (NN)*

MLP is a supervised learning algorithm that learns a function  $h(\cdot) : R^D \sim R^T$  by training on a dataset, where  $D$  is the number of dimensions of attributes for the input and  $T$  is the number of cluster IDs for the output. Given a set of attributes and a target, MLP can learn a non-linear function for classification. MLP consists of three layers: input, hidden, and output. As shown in Fig. 5, we associate the input nodes with reference node  $A_1 \sim A_D$  and the output nodes with predefined cluster-ID. The leftmost layer, known as the input layer, consists of a set of neurons  $\{A_1, A_2, \dots, A_D\}$  representing the input attributes. The hidden layers are between the input and output layers, where the transitional computations are performed. Each hidden layer uses the output of the previous layer to perform a non-linear operation, which is defined as:

$$\mathbf{h}^k = \phi(\mathbf{W}^k \mathbf{h}^{k-1} + \mathbf{b}^k) \quad (28)$$

where  $\mathbf{W}^k$  is a fully connected weight matrix that represents all the connections between each node of the  $(k-1)$ th layer and each node of the  $k$ th layer.  $\mathbf{b}^k$  is the bias vector of  $k$ th layer,  $\mathbf{h}^{k-1}$  represents the output from the previous layer. The weights and biases in a neural network are initially set to random values but the model is trained using the back-propagation (BP) method and the Adam optimizer [32] to minimize the loss function and updates the network parameters (i.e., weights and biases) iteratively until a convergence is achieved.  $\phi(\cdot)$  is the activation function that in our case, we use Rectified Linear Unit (ReLU) [33] (i.e.,  $\phi(n) = \max(n; 0)$ ) as an activation function in hidden layers and the softmax function is used as the activation function of the output layer so that the sum of the output values of all output neurons is equal to 1 and is defined as

$$\phi(\mathbf{n})_i = \frac{e^{n_i}}{\sum_{t=1}^T e^{n_t}} \quad (29)$$

The output layer receives the values from the last hidden layer and transforms them into output values. Then the model is evaluated by comparing the real cluster-ID and the estimated cluster-ID.

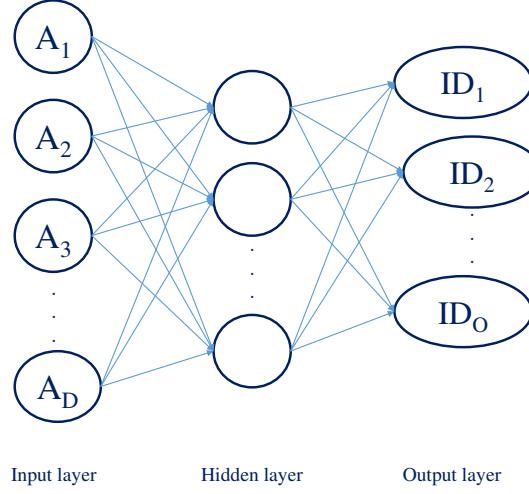


Fig. 5: Structure of MLP classifier for cluster identification.

### 3.1.4 Making a Regression Model

Let us suppose matrix  $\mathbf{N}$  is divided into  $T$  clusters  $\mathbf{N}'_t, t = 1, \dots, T$  with known locations  $[\mathbf{x}_t, \mathbf{y}_t]$ . Therefore, we have  $\mathbf{N} = \{\mathbf{N}'_1, \mathbf{N}'_2, \dots, \mathbf{N}'_T\}$ . Consequently, the training data of each cluster are modeled using a GPR model, which takes the fingerprint as an input and provides the UE's location as an output. For simplicity, the data of a cluster is only considered in this part, but a similar procedure is applied for other clusters as well. Therefore, we consider the  $t^{\text{th}}$  cluster which has  $L'$  training samples.

Let us define  $f_x(\cdot)$  and  $f_y(\cdot)$  as the functions which map the fingerprint vector  $\mathbf{n}_k$  of any user  $k$  into cluster  $\mathbf{N}'_t$  to provide the 2-dimensional user's location coordinates  $(x_k, y_k)$ , such that

$$\begin{aligned} x_k &= f_x(\mathbf{n}_k) + v_x, \\ y_k &= f_y(\mathbf{n}_k) + v_y \end{aligned} \quad (30)$$

where  $v_x$  and  $v_y$  are error terms modeled as i.i.d. Gaussian random variables with zero mean and variance  $\sigma_{v_x}^2$  and  $\sigma_{v_y}^2$ , respectively. From (30), we can see that, estimating the  $x$ -coordinate  $x_k$  (and  $y$ -coordinate  $y_k$ ) from  $\mathbf{n}_k$  is a non-linear regression problem in machine learning. Among non-linear regression methods, we choose GPR because it is a powerful, Bayesian non-parametric approach that provides a probability distribution over all possible values [34]. Also, GPR methods have a good performance in terms of multiple metrics, including the squared prediction error [35]. For simplicity, details of the GPR model are presented for the  $x$ -coordinates of the users and the same procedure can be applied for the  $y$ -coordinates.

In (30), function  $f_x(\cdot)$  is supposed to be random and follows a Gaussian process with zero-mean and covariance matrix  $\mathbf{C}_t$  (also known as a kernel matrix), such that

$$f_x(\cdot) \sim \mathcal{GP}(0, \mathbf{C}_t) \quad (31)$$

The covariance function between user  $k$  and user  $k'$  in cluster  $\mathbf{N}'_t$ , is the weighted sum of the squared-exponential, the linear and the delta function [36] and is defined by

$$c_t(\mathbf{n}_k, \mathbf{n}_{k'}) = \gamma e^{-\frac{1}{2\vartheta^2} \|\mathbf{n}_k - \mathbf{n}_{k'}\|^2} + \varpi \mathbf{n}_k^T \mathbf{n}_{k'} + \sigma_{v_x}^2 \delta_{kk'} \quad (32)$$

where  $c_t(\mathbf{n}_k, \mathbf{n}_{k'})$  is an element of  $\mathbf{C}_t$  and the delta function  $\delta_{kk'}$  is 1 if  $k = k'$  and 0 otherwise. According to (32), we need to estimate and optimize an unknown GPR hyperparameter vector  $\boldsymbol{\phi}_t = [\gamma, \vartheta, \varpi]$  from the training data. Based on the GP assumption, i.e. data can be represented as a sample from a multivariate Gaussian distribution, we know that  $\mathbf{x}_t$  is Gaussian distributed,  $\mathbf{x}_t | \mathbf{N}'_t, \boldsymbol{\phi}_t \sim \mathcal{N}(0, \mathbf{C}_t)$  [37]. The log-likelihood function is used to derive the maximum likelihood estimator of parameter  $\boldsymbol{\phi}_t$ . The estimator  $\hat{\boldsymbol{\phi}}_t$  is obtained by solving

$$\hat{\boldsymbol{\phi}}_t = \arg \max_{\boldsymbol{\phi}_t} \log(p(\mathbf{x}_t | \mathbf{N}'_t, \boldsymbol{\phi}_t)) \quad (33)$$

The optimization problem in (33) can be solved using a limited memory Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm which is an optimization algorithm in the family of quasi-Newton methods that approximates the BFGS algorithm using a limited amount of computer memory. It is a well-known algorithm for parameter estimation in machine learning that is explained in detail in [38].

### 3.2 Online Positioning Phase

In this phase, the position of a test user whose location is unknown is estimated. Let us suppose there are  $\hat{L}$  test users. For this purpose, the  $\hat{L} \times D$  testing matrix  $\hat{\mathbf{N}}$  is used to estimate the  $\hat{L} \times 1$   $x$ -coordinate vector  $\hat{\mathbf{x}}$ . For each test user data  $\hat{\mathbf{n}}$ , the process of location estimation is described below.

- *Step 1: Cluster identification*

Based on the cluster identification algorithm, the cluster ID  $t$  of testing data point  $\hat{\mathbf{n}}$  is estimated.

- *Step 2: Location estimation*

By using the GPR model of related cluster  $t$ , the  $x$ -coordinate  $\hat{x}$  of test user fingerprint  $\hat{\mathbf{n}}$  is estimated.

Now we would like to determine the posterior density of the location, i.e.  $\hat{x} | \mathbf{x}_t, \mathbf{N}'_t, \hat{\mathbf{n}}$ . According to [39], this distribution is Gaussian and the best estimate for  $\hat{x}$  is the mean of this distribution, which is defined as

$$\hat{\mu}_x = \mathbf{c}_t^T [\mathbf{C}_t + \sigma_v^2 \mathbf{I}_{L'}]^{-1} \mathbf{x}_t \quad (34)$$

where  $\mathbf{c}_t = (c_t(\hat{\mathbf{n}}, \mathbf{n}_1), \dots, c_t(\hat{\mathbf{n}}, \mathbf{n}_{L'}))$  and  $L'$  is the number of training data in the cluster  $t$ . Also, the uncertainty in our estimate is given by its variance which is given as

$$\hat{\sigma}_x^2 = \sigma_v^2 - c_t(\hat{\mathbf{n}}, \hat{\mathbf{n}}) - \mathbf{c}_t^T [\mathbf{C}_t + \sigma_v^2 \mathbf{I}_{L'}]^{-1} \mathbf{c}_t \quad (35)$$

## 4 Results and Discussion

In this section, we compare the position estimation performance of the two-stage fingerprint clustering method in [20], which is considered as a benchmark, and the proposed method. To simulate a common urban wireless propagation scenario, a  $120^\circ$  sector with radius  $R = 500$  m is considered. At the center of the sector, we place a BS which is equipped with a ULA where the number of antennas  $M$  is 128. The major wireless parameters which are employed in the simulation are set to those typical in LTE [40], [41], as listed in Table 1. Also, to have a simple analysis, it is assumed that the location of a UE is estimated based on the 20 nearest scatterers. The RPs are uniformly distributed every 10 m in a grid configuration over the whole target area. However, the test users are randomly distributed in the sector. In our case, the number of training RPs  $L$  is 893 and the number of testing data points  $\hat{L}$  is 157. For each path, the location point of scattering is used to compute the AoA, the CFR of each UE is calculated according to (1) and (3).

Table 1: Wireless Parameters

Parameters	Value
Sampling interval	8.138 ns
Sampling frequency	122.88 MHz
Transmission bandwidth	100 MHz
Guard interval	4.7 MHz
Sub-carrier spacing	15 kHz

### 4.1 Preprocessing Results

As mentioned in the offline phase, the fingerprints of the RPs are first extracted and stored into the dataset with their corresponding coordinates. The training data  $\mathbf{R}$  are standardized according to (10) and the pre-processing scenarios are employed. For evaluating the performance of the proposed method, first, the new dimensions of the training data should be specified using dimensionality reduction PCA. In PCA, a vital part is to estimate how many principal components which explain the variance of the data are needed. The relation between the number of principal components and the percentage of data variance is shown in Table 2. We can see that with the first 20 components, approximately 80% of the variance is contained, while we need 58

components to describe close to 95% of the variance. Since we aim to have 90% of the variance, the number of principal components is set to 35 and thus the dimensions of the original data is reduced to 35.

Table 2: The projection loss of the principal components.

Principal components	Variance percentage
20	80
30	85
35	90
58	95

In Fig. 6, shows the t-SNE visualization of applying the PCA on our localization dataset with 35 principal components. We can see that the data are very clearly separated into sub-groups. We can clearly see how all the samples are nicely spaced apart and grouped together with their respective locations. If we now use a clustering algorithm to pick out the separate clusters, we could probably automatically assign new points to a label.

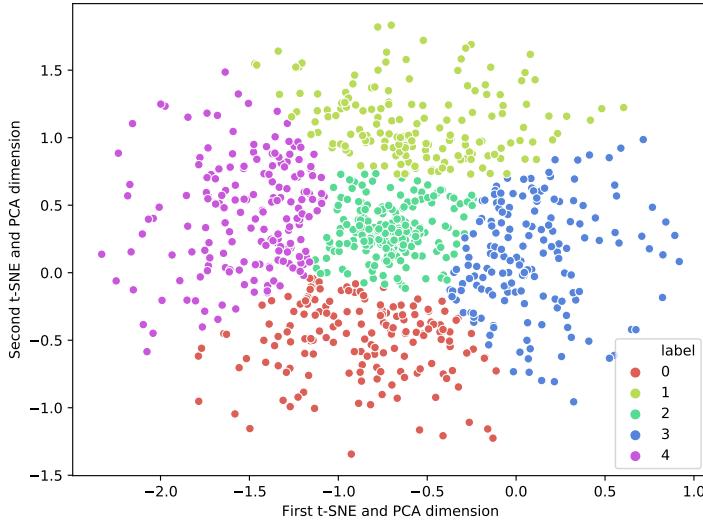


Fig. 6: Preprocessing data using PCA and t-SNE.

#### 4.2 Clustering and Cluster Validation Results

For clustering the compressed training data, k-means, APC and agglomerative algorithms are applied. Before applying APC, according to (26), we select the median value of the similarity matrix in (25) as the initial preference value. Then by increasing this value, we generate a range of preference values because changes in the preference value can result in quite different clustering results. Therefore, APC takes these values as an input to cluster the compressed training matrix  $\mathbf{N}$ . The maximum number of iterations in APC is  $num\_iterations = 200$ . For evaluating the performance of the proposed method, first, the valid number of clusters should be specified. It should be noted that the clustering results affect the cluster identification accuracy, the computational complexity, and further affect the position estimation accuracy, as shown later in this section. For this purpose, first, the valid clusterings which has high quality, are identified. Then, the results of validity indexes are averaged over 100 Monte-Carlo runs.

Fig. 7 shows the quality of clustering of the training data points using k-means, affinity propagation and agglomerative clustering algorithms based on the silhouette,  $SI$ , value. It should be noted that the value of  $SI$  is in the range of 0 to 1. A larger value of  $SI$  represents higher clustering quality. Each cluster can be separated significantly when  $SI > 0.5$ . We have an unsuitable cluster structure when  $SI < 0.2$ . According to the clustering results, we can see that when we use affinity propagation to cluster the training data points, we have better average silhouette which means that the data are well separated. Also, we have a good clustering quality when we use agglomerative clustering, especially when we have 5 clusters.

#### 4.3 Accuracy of Cluster Identification

As mentioned in the offline phase, the MLP algorithm is applied for cluster identification. To evaluate the accuracy of the MLP model, 20% of all training data are considered as a validation dataset and it is supposed that their cluster-ID is unknown. Then, by comparing the estimated cluster ID and the real one, the accuracy is obtained. In our analysis, the accuracy of the model is 92%.

#### 4.4 Positioning Performance

For estimating the position, the GPR model is trained by solving the log-likelihood maximization problem in (33).

During the online positioning phase, the fingerprints of the new UEs are extracted and their position is estimated with our positioning method. Then the estimated positions are compared with the true positions and the performance of the proposed fingerprint wireless positioning system is evaluated.

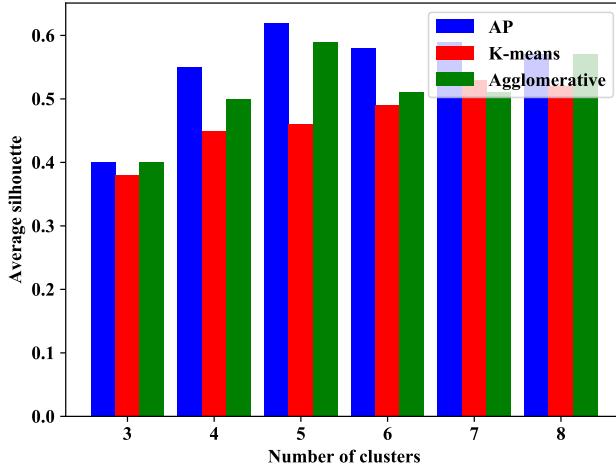


Fig. 7: Comparison of clustering algorithms based on average silhouette.

In this part, by considering the two-stage fingerprint clustering method in [20] as a benchmark, we first present the performance of the localization accuracy of the proposed method. Simulation results are obtained to indicate that the proposed method is suitable in massive MIMO-OFDM systems. Also, the effect of the number of BS antennas in location estimation performance is evaluated. For evaluation, a massive MIMO-OFDM system with 128 BS antennas is considered. Fig. 8 shows the cumulative distribution of the estimation errors for different methods. We can see that the accuracy of the proposed method is better than the benchmark with 93% reliability for 10-meter accuracy. For comparison, in [20], the reliability for 10-meter accuracy was 70%.

The impact of the number of BS antennas on the localization accuracy is demonstrated in Fig. 9. When the number of antennas is increased from 64 to 128, the reliability for 10-meter accuracy is increased from 81% to 90%. Also, we can see that increasing the number of BS antennas increases the localization accuracy.

## 5 Conclusion

We proposed a low dimensional cluster-based approach to estimate the user's location from the CSI in a collocated massive MIMO-OFDM system. In the proposed method, first, all high dimension training data were mapped into a lower dimensional space. Then the whole testbed was divided into clusters using different clustering algorithms, which reduces the computational cost of online positioning. APC was chosen for clustering due to its initialization-

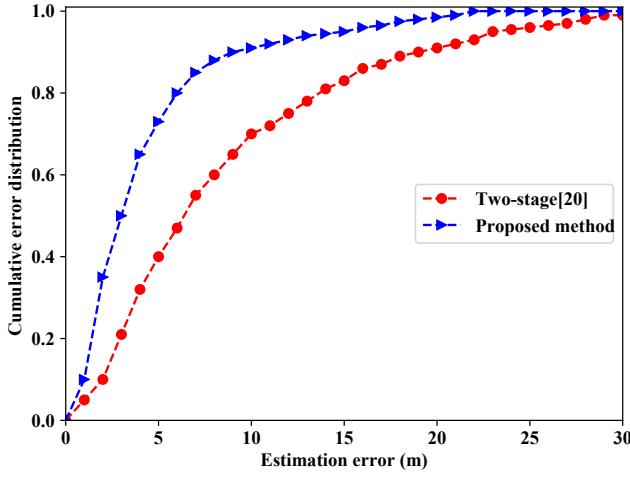
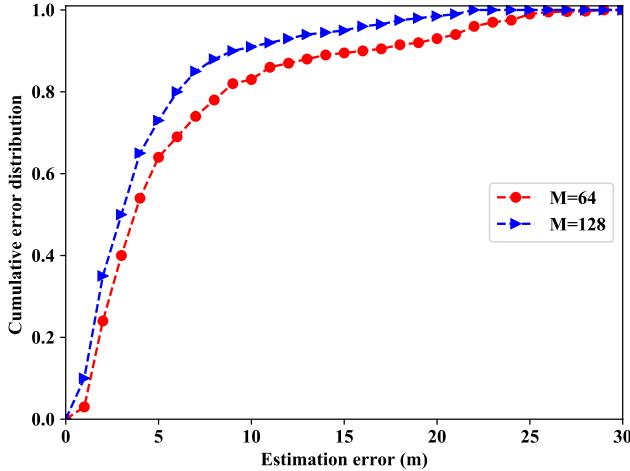


Fig. 8: The CDF of the location errors.

Fig. 9: The CDF of the location errors with different number of antennas  $M$ .

independent property and better selection of cluster representative compared with k-means and agglomerative clustering. A MLP was used for cluster identification to allow for a quick finding of the related cluster. Also, GPR was applied for further location estimation within each cluster. The proposed method was compared with a previous work in terms of localization accuracy. Also,

through simulation results, we showed that with increasing the number of antennas, the localization performance is improved.

### Acknowledgment

The authors would like to thank NSERC for supporting this research.

### References

1. Dammann, A., Raulefs, R., & Zhang, S. (2015) On prospects of positioning in 5G. In: IEEE International Conference on Communication Workshop (ICCW), pp. 1207–1213
2. Mendlzik, R., Meyer, F., Bauch, G., & Win, M. (2019) Localization, mapping, and synchronization in 5G millimeter wave massive MIMO systems. In: 20th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), IEEE, pp. 1–5
3. Wen, F., Wymeersch, H., Peng, B., Tay, W. P. , So, H. C. & Yang, D. (2019) A survey on 5G massive mimo localization. *Digital Signal Processing* 94:21–28
4. Ngo, H., Larsson, E. G., & Marzetta, T. L. (2013) Energy and spectral efficiency of very large multiuser MIMO systems. *IEEE Transactions on Communications* 61(4):1436–1449
5. Vieira, J., Leitinger, E., Sarajlic, M., Li, X., & Tufvesson, F. (2017) Deep convolutional neural networks for massive MIMO fingerprint-based positioning. In: IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), IEEE, pp. 1–6
6. Pirzadeh, H., Wang, C., & Papadopoulos, H. (2019) Machine-learning assisted outdoor localization via sector-based fog massive MIMO. In: IEEE International Conference on Communications (ICC), IEEE, pp. 1–6
7. Prakash, V. C., & Nagarajan, G. (2019) A hybrid RSS-TOA based localization for distributed indoor massive MIMO systems. In: International Conference on Emerging Current Trends in Computing and Expert Technology, Springer, pp. 1359–1370
8. Li, M. , & Lu, Y. (2008) Angle-of-arrival estimation for localization and communication in wireless networks. In: 16th European Signal Processing Conference, IEEE, pp. 1–5
9. Chan, Y., & Ho, K. (1994) A simple and efficient estimator for hyperbolic location. *IEEE Transactions on signal processing* 42(8):1905–1915
10. Zanella, A. (2016) Best practice in RSS measurements and ranging. *IEEE Communications Surveys & Tutorials* 18(4):2662–2686
11. So, H. C., & Lin, L. (2011) Linear least squares approach for accurate received signal strength based source localization. *IEEE Transactions on Signal Processing* 59(8):4035–4040

12. Ning, C., Li, R., & Li, K. (2016) Outdoor location estimation using received signal strength-based fingerprinting. *Wireless Personal Communications* 89(2):365–384
13. Yiu, S., Dashti, M., Claussen, H., & Perez-Cruz, F. (2017) Wireless RSSI fingerprinting localization. *Signal Processing* 131:235–244
14. Chen, H., Zhang, Y., Li, W., Tao, X., & Zhang, P. (2017) ConFi: Convolutional neural networks based indoor Wi-Fi localization using channel state information. *IEEE Access* 5:18066–18074
15. Berruet, B., Baala, O., Caminada, A., & Guillet, V. (2018) Delfin: a deep learning based csi fingerprinting indoor localization in IoT context. In: International Conference on Indoor Positioning and Indoor Navigation (IPIN), IEEE, pp. 1–8
16. Hu, A., Lv, T., Gao, H., Zhang, Z., & Yang, S. (2014) An ESPRIT-based approach for 2-D localization of incoherently distributed sources in massive MIMO systems. *IEEE Journal of Selected Topics in Signal Processing* 8(5):996–1011
17. Shahmansoori, A., Garcia, G. E., Destino, G., Seco-Granados, G., & Wyneersch, H. (2015) 5G position and orientation estimation through millimeter wave MIMO. In: IEEE Globecom Workshops, IEEE, pp. 1–6
18. Garcia, N., Wyneersch, H., Larsson, E.G., Haimovich, A. M., & M. Coulon (2017) Direct localization for massive MIMO. *IEEE Transactions on Signal Processing* 65(10):2475–2487
19. Savic, V., & Larsson, E. G. (2015) Fingerprinting-based positioning in distributed massive MIMO systems. In: Vehicular Technology Conference (VTC Fall), pp. 1–5
20. Sun, X., Gao, X., Li, Y. G., & Han, W. (2017) Fingerprint based single-site localization for massive MIMO-OFDM systems. In: IEEE Global Communications Conference, IEEE, pp. 1–7
21. Obaid, H. S., Dheyab, S. A., & Sabry, S. S. (2019) The impact of data pre-processing techniques and dimensionality reduction on the accuracy of machine learning. In: Information Technology, Electromechanical Engineering and Microelectronics Conference (IEMECON), IEEE, Jaipur, pp. 279–283
22. Ismail, M., & Dauda, U. (2013) Standardization and its effects on k-means clustering algorithm. *Research Journal of Applied Sciences, Engineering and Technology* 6(17):3299–3303
23. Alkhayrat, M., Aljnidi, M., & Aljoumaa, K. (2020) A comparative dimensionality reduction study in telecom customer segmentation using deep learning and PCA. *Journal of Big Data* 7(1):9
24. Jolliffe, I.T. (2002) *Principal Component Analysis, Second Edition*. 2nd ed. Springer series in statistics
25. Maaten, L. V., & Hinton, G. (2008) Visualizing data using t-SNE. *Journal of machine learning research* 9:2579–2605
26. Kingman, F. C., & Kullback, S. (2007) Information theory and statistic. *Neurocomputing* 54(387)
27. Gisbrecht, A., Schulz, A., & Hammer, B. (2015) Parametric nonlinear

- dimensionality reduction using kernel t-SNE. *Neurocomputing* 147:71–82
28. Frey, B. J., & Dueck, D. (2007) Clustering by passing messages between data points. *science* 315(5814):972–976
  29. Gokcay, E., & Principe, J. (2002) Information theoretic clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24(2):158–171
  30. Saitta, S., Raphael, B., & Smith, I. F. (2008) A comprehensive validity index for clustering. *Intelligent Data Analysis* 12(6):529–548
  31. Rousseeuw, P. J. (1987) Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics* 20:53–65
  32. Zhang, Z. (2018) Improved Adam optimizer for deep neural networks. In: 26th International Symposium on Quality of Service (IWQoS), IEEE, pp. 1–2
  33. Soltanolkotabi, M. (2017) Learning relus via gradient descent. In: Advances in neural information processing systems, pp. 2007–2017
  34. Schulz, E., Speekenbrink, M., & Krause, A. (2018) A tutorial on Gaussian process regression: Modelling, exploring, and exploiting functions. *Journal of Mathematical Psychology* 85:1–16
  35. Rasmussen, C. E. (1997) *Evaluation of Gaussian processes and other methods for non-linear regression*
  36. Prasad, K. S., Hossain, E., Bhargava, V. K. (2017) Low-dimensionality of noise-free RSS and its application in distributed massive MIMO. *IEEE Wireless Communications Letters* 7(4):486–489
  37. Rasmussen, C. E., & Williams, C. K. I. (2006) *Gaussian processes for machine learning*, vol 2. MIT Press Cambridge, MA
  38. Byrd, R. H., Lu, P., Nocedal, J., & Zhu, C. (1995) A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing* 16(5):1190–1208
  39. Pérez-Cruz, F., Van Vaerenbergh, S., Murillo-Fuentes, J. J., Lázaro-Gredilla, M., & Santamaria, I. (2013) Gaussian processes for nonlinear signal processing: An overview of recent advances. *IEEE Signal Processing Magazine* 30(4):40–50
  40. 3GPP (2010) Further advancements for e-utra physical layer aspects. *TR 36814*
  41. Miyazaki, N., Nanba, S., & Konishi, S. (2010) MIMO-OFDM throughput performances on MIMO antenna configurations using LTE-based testbed with 100 MHz bandwidth. In: 72nd Vehicular Technology Conference-Fall (VTC-fall), IEEE, Ottawa, Canada, pp. 1–5

## Declarations

- Funding: It is provided by Natural Sciences and Engineering Research Council (NSER).
- Conflicts of interest: There is no Conflicts of interest.
- Availability of data and material: Not applicable

- Code availability: Not applicable



**Seyedeh Samira Moosavi** received her M.Sc. degree in Information Technology Engineering, Network Engineering from the Shiraz University of Technology (SUTech), Shiraz, Iran, in 2015. She is currently pursuing the Ph.D. degree in the Department of Electrical and Computer Engineering at Laval University, Quebec, Canada. Her current research focus is on machine learning methods and their applications in 5G technology.



**Paul Fortier** (S'78-M'89-SM'00) received his B.Sc. degree and his M.Sc. degree in Electrical Engineering from Laval University in 1982 and 1984, respectively and his M.S. degree in Statistics and his Ph.D. degree in Electrical Engineering from Stanford University in 1987 and 1989, respectively. Since 1989, he has been with the Department of Electrical and Computer Engineering at Université Laval where he is currently a full professor. From 1991 to 1996, he was program director for the B.Sc. degree in Computer Engineering and from 1997 to 2003 he was Chairman of the Department of Electrical and Computer Engineering. From 2003 to 2007, he was Associate Dean for Development and Research at the Faculty of Science and Engineering. From 2007 to 2009, he was Vice-president Scientific Affairs and Partnerships at FQRNT (Quebec's granting agency). From 2010 to 2012, he was Vice-President for Research and Innovation at Laval University. He is currently Chair of the Department of Electrical and Computer Engineering at Laval University. His research interests include digital signal processing for communications and the study of complexity and performance tradeoffs in hardware implementations, with applications in wireless communications. He has been involved in the organization of national and international conferences and workshops in these fields. He has done consulting work for several companies and government agencies in Canada. Dr. Fortier is a Fellow of the Engineering Institute of Canada, a Fellow of the Canadian Academy of Engineering and a Senior Member of the Institute of Electrical and Electronics Engineers (IEEE).

puter Engineering and from 1997 to 2003 he was Chairman of the Department of Electrical and Computer Engineering. From 2003 to 2007, he was Associate Dean for Development and Research at the Faculty of Science and Engineering. From 2007 to 2009, he was Vice-president Scientific Affairs and Partnerships at FQRNT (Quebec's granting agency). From 2010 to 2012, he was Vice-President for Research and Innovation at Laval University. He is currently Chair of the Department of Electrical and Computer Engineering at Laval University. His research interests include digital signal processing for communications and the study of complexity and performance tradeoffs in hardware implementations, with applications in wireless communications. He has been involved in the organization of national and international conferences and workshops in these fields. He has done consulting work for several companies and government agencies in Canada. Dr. Fortier is a Fellow of the Engineering Institute of Canada, a Fellow of the Canadian Academy of Engineering and a Senior Member of the Institute of Electrical and Electronics Engineers (IEEE).

## Figures

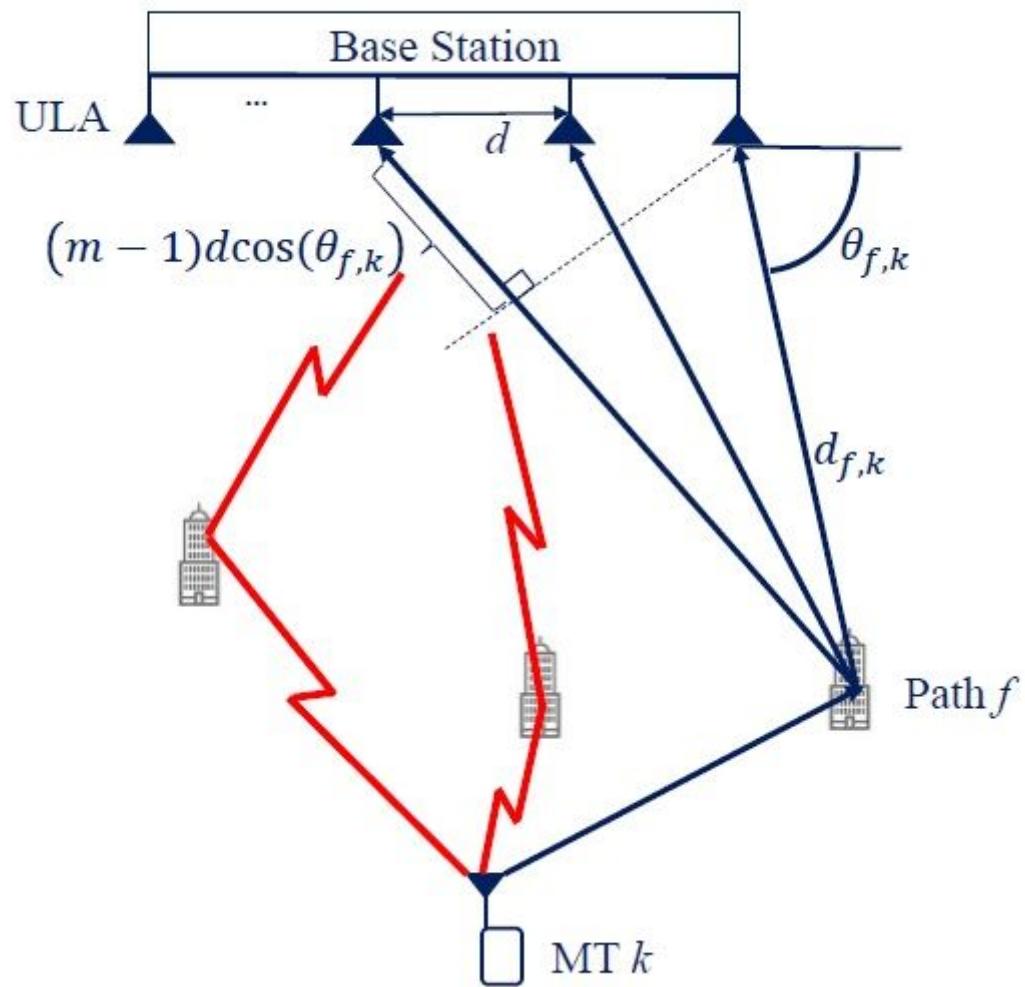
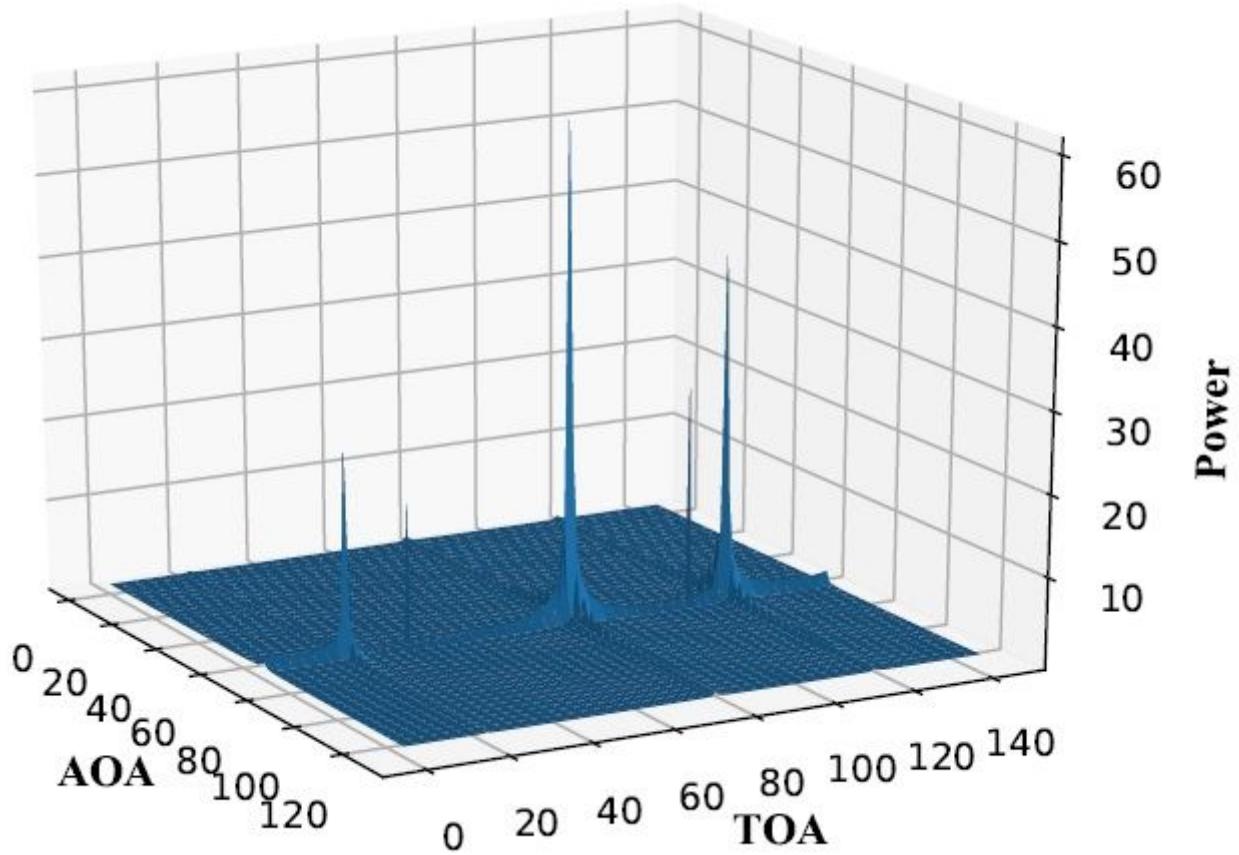


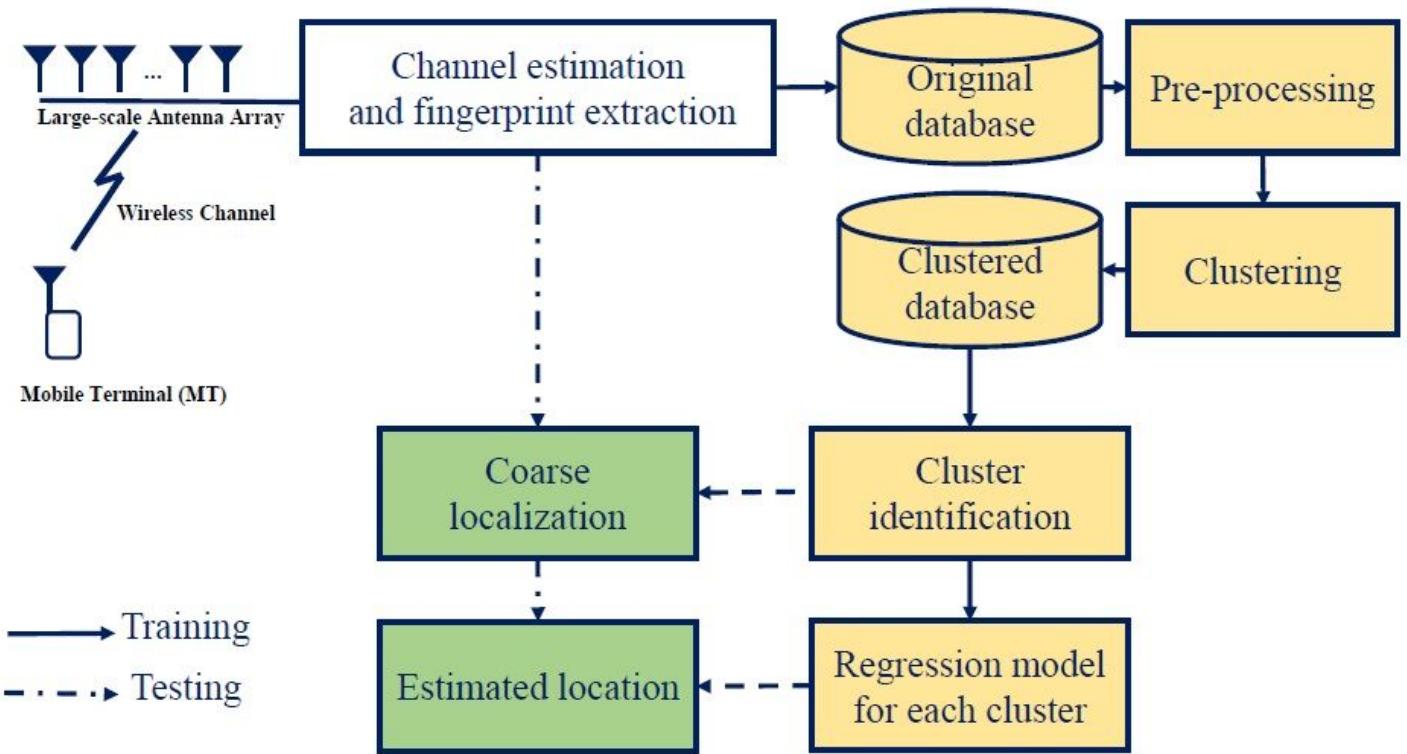
Figure 1

The wireless channel from an arbitrary user  $k$  to the BS.



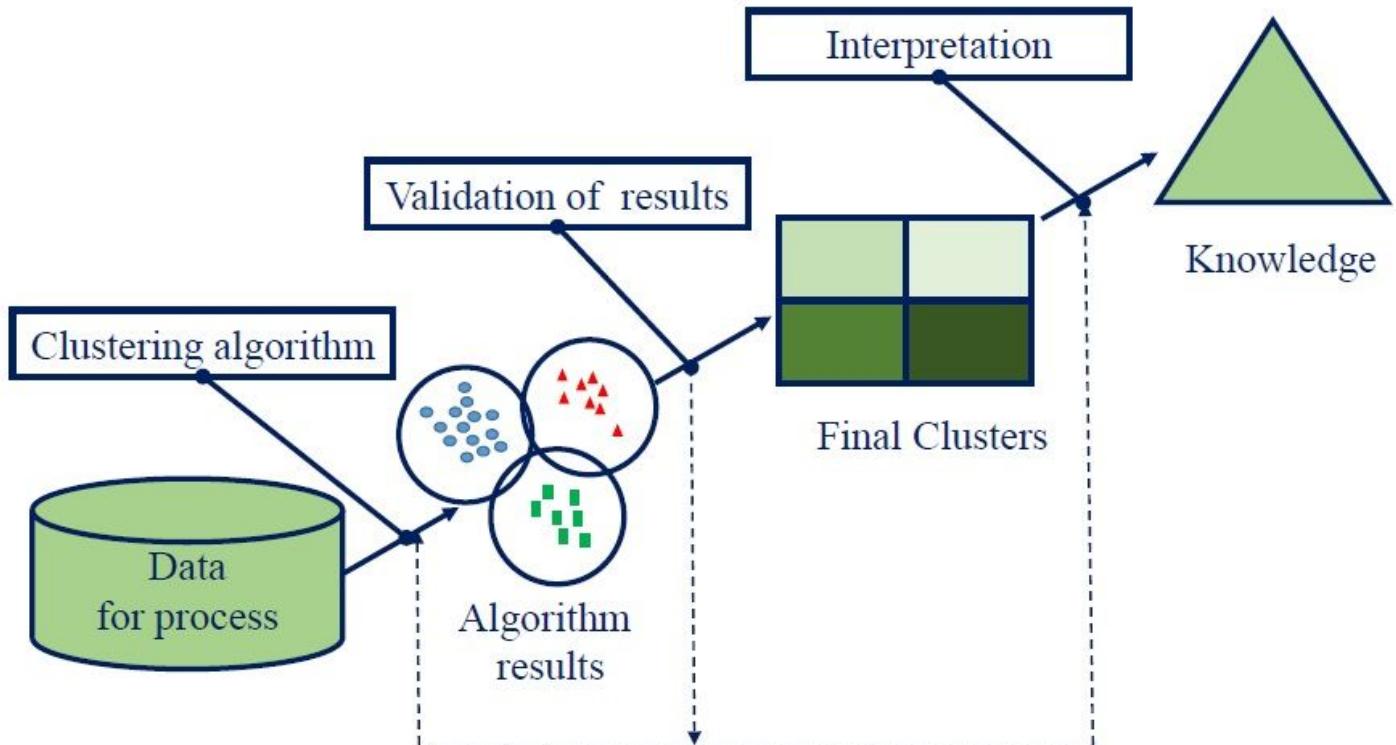
**Figure 2**

Example of ADCPM with  $M = 128$  and  $Ng = 144$ .



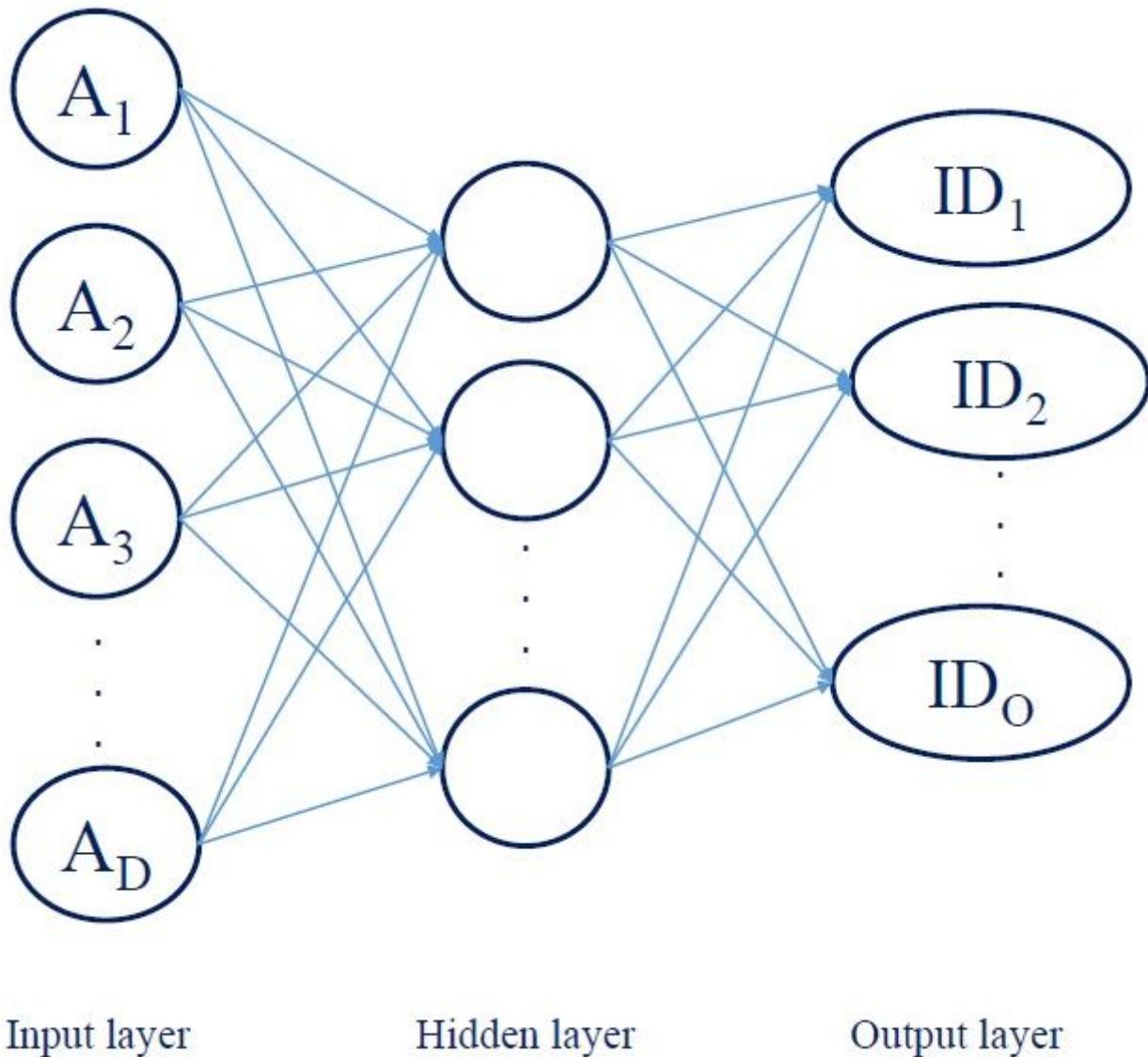
**Figure 3**

Overview of the proposed position estimation scheme.



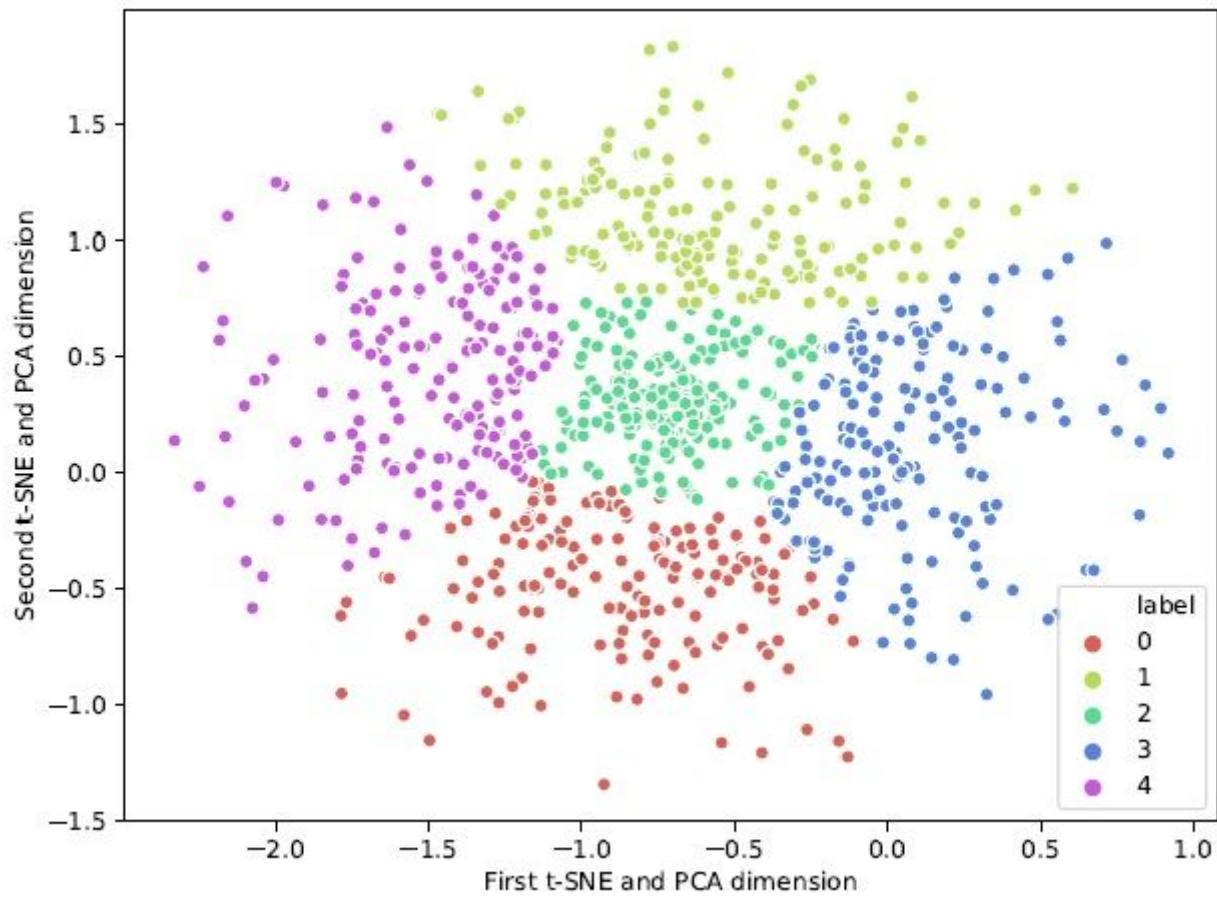
**Figure 4**

The pre-processing scenarios.



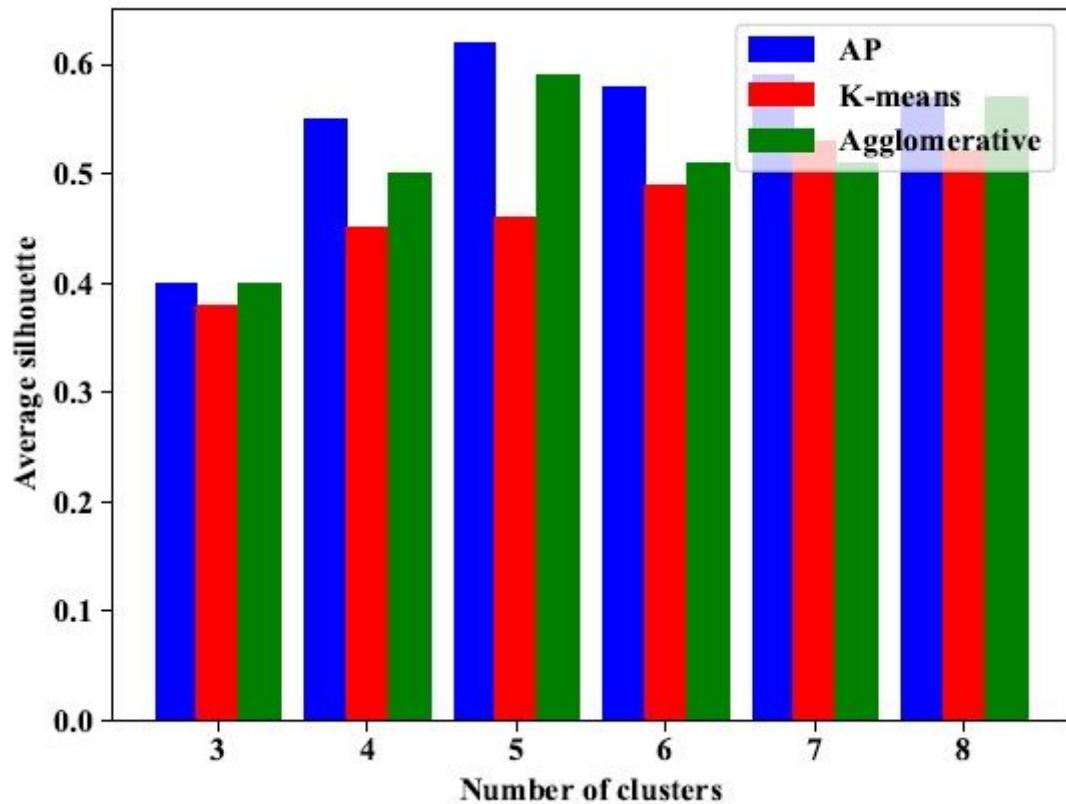
**Figure 5**

Structure of MLP classifier for cluster identification.



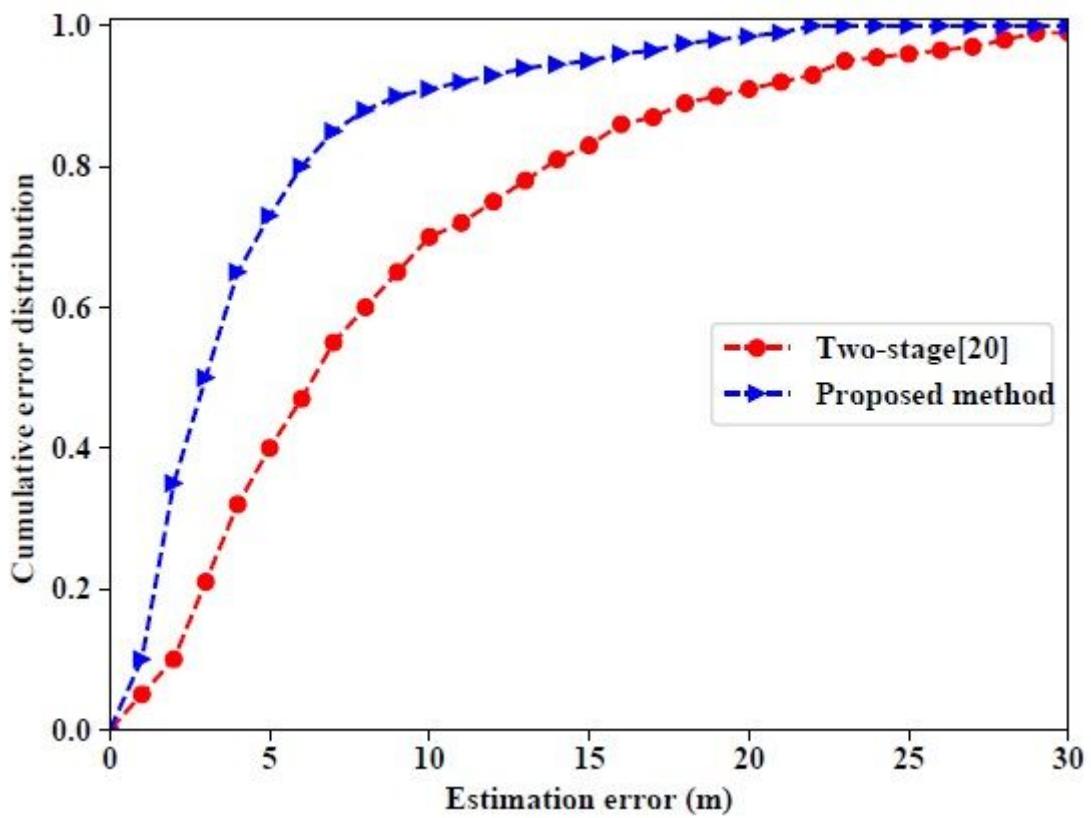
**Figure 6**

Preprocessing data using PCA and t-SNE.



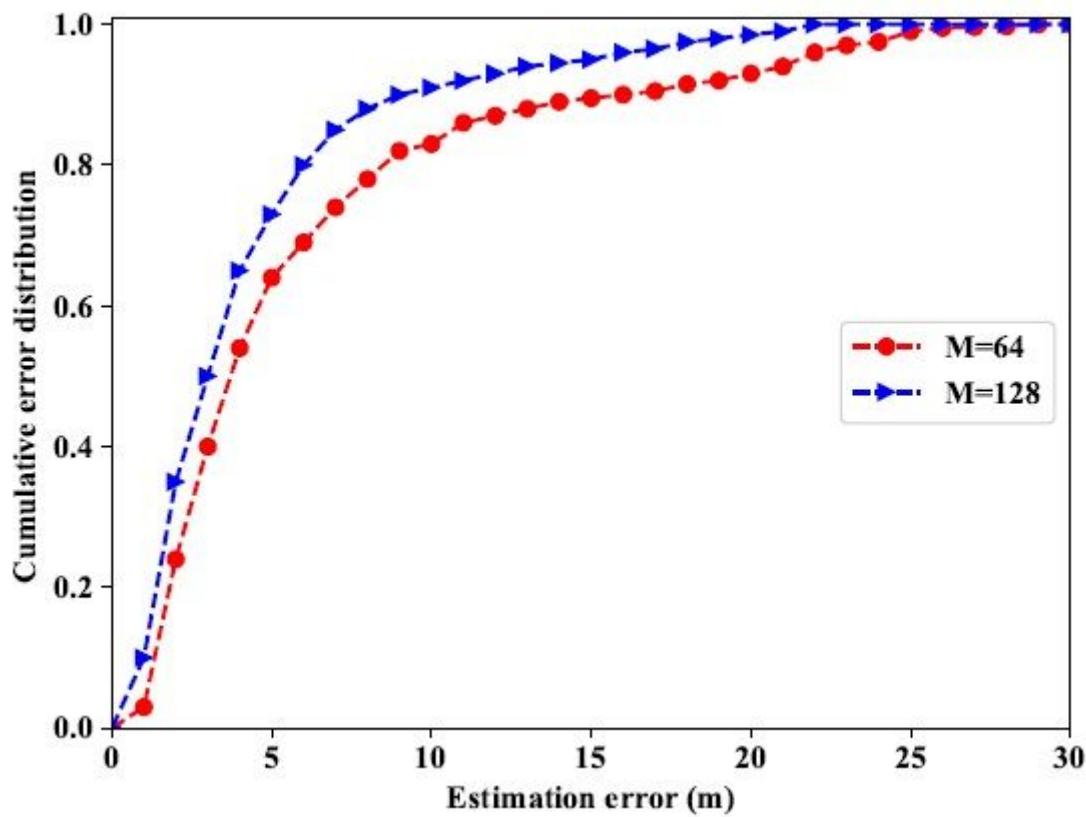
**Figure 7**

Comparison of clustering algorithms based on average silhouette.



**Figure 8**

The CDF of the location errors.



**Figure 9**

The CDF of the location errors with different number of antennas M.