SVM-assisted Adaptive Kernel Power Density Clustering Algorithm for Millimeter Wave Channels

Du, Fei ; Zhao, Xiongwen ; Zhang, Yu ; Wen, Yang ; Fu, Zihao ; Geng, Suiyan ; Qin, Peng ; Zhou, Zhenyu ; Xu, Chen ; Liu , Yongsheng ; Fan, Wei

Published in:
I E E E Transactions on Antennas and Propagation

DOI (link to publication from Publisher):
10.1109/TAP.2022.3140538

Publication date:
2022

Document Version
Accepted author manuscript, peer reviewed version

SVM-assisted Adaptive Kernel Power Density Clustering Algorithm for Millimeter Wave Channels

Fei Du, Xiongwen Zhao, *Senior Member, IEEE,* Yu Zhang, Yang Wen, Zihao Fu, Suiyan Geng, Peng Qin, Zhenyu Zhou, *Senior Member, IEEE,* Chen Xu,Yongsheng Liu and Wei Fan, *Senior Member, IEEE*

***Abstract*—Cluster based channel modeling has gradually become a trend in the development of channel model, since it is a good compromise between accuracy and complexity. However, most of the existing clustering algorithms require prior knowledge of clusters, initialization and threshold choices. An accurate and automatic cluster identification algorithm is therefore highly desirable for channel modeling. In this paper, an adaptive kernel-power-density (AKPD) and support vector machine assisted AKPD (SVM-AKPD) algorithms are proposed. Firstly, a new distance-based metric is proposed to calculate an *adaptive-K* for each multipath component (MPC), in which the AKPD can be used in scenarios where we have complex distribution of MPCs, especially for the cluster with small MPCs. Furthermore, the SVM is applied in clustering by the full partition of MPCs feature space to overcome the limitation of the AKPD, where the MPCs lying at a large distance from the cluster centroids will be clustered into surrounding clusters when the clusters are closely-spaced in the AKPD. Finally, the performance of the proposed AKPD and SVM-AKPD is validated with measured and simulated channels data at millimeter wave band, respectively. Both numerical simulations and experimental validation results are provided to demonstrate the effectiveness and robustness of the proposed algorithm. The proposed algorithms enable applications in multiple-input–multiple-output (MIMO) channels with no prior knowledge about the clusters, such as number and initial locations. It also does not need to adjust cluster parameters manually and can be implemented for cluster based channel modeling with a fairly low complexity.**

***Index Terms*—Channel measurement and modeling, SVM, MIMO, cluster, kernel density, multipath component, wireless channel**

I. Introduction

IT is of great importance to understand the radio channel of a wireless communication system for system development and performance evaluation. The concept of clustering, where multipath components (MPCs) with similar parameters are grouped, has been widely adopted in the standard channel models for the fourth generation (4G) radio systems. Due to the high resolution in the delay and space domain introduced by the high system bandwidth and massive antenna configuration in the fifth generation (5G) millimeter wave communications, it is expected that MPCs can be better observed. The MPCs are not independent to each other but rather distributed in groups, i.e., clusters, which has been demonstrated by extensive multiple-input-multiple-output (MIMO) channel sounding campaigns in real deployments [1] [2]. Especially, the mmWave wireless channels exhibit a sparse structure, when measuring the channel using large antenna arrays and large bandwidths [3] [4]. Therefore, clustering has also been an important research topic in 5G research [5]. The concept of cluster has gained popularity mainly due to its simplicity (i.e., reduction of MIMO channel model parameters) and flexibility for the multi-link scenario simulations. In addition, it has been investigated that channel models that do not take the cluster into consideration might overestimate the channel capacity [6]. However, the accuracy of cluster based channel model is directly affected by the clustering algorithm [7]. An accurate and automatic cluster identification algorithm is therefore highly desirable for channel modeling.

The main challenges of MPC clustering are how to incorporate different MPC attributes in clustering and how to reduce the dependence on prior knowledge of cluster. The MPC has several attributes, i.e., power, delay and angle, and each of the above attributes usually has an independent characteristic (i.e., MPCs exhibit different statistical characteristics in each domain). Moreover, the existing clustering algorithms generally need to specify the number of clustering in advance, which, however, is difficult to obtain in the practical application. The state-of-the-art clustering and tracking algorithms that are used for cluster-based channel modeling were reviewed in [8]. Several algorithms have been proposed to cluster MPCs by considering the power, delay, and angular information of MPCs, e.g., K-means algorithm [9] and K-power-means (KPM) [10], where the cluster centroids are identified using the MPC power. The multipath component distance (MCD) calculated with MPC delay and angle is used as a measure to quantify the similarity between the MPCs. However, due to the unsupervised learning property of the KPM, its initialization must be known as a priori. In other words, we need to manually adjust parameters according to different channel dataset, which makes the KPM somehow subjective. The conventional fuzzy c-means (FCM) method was proposed in [11], which was shown to outperform the KPM with random initialization. However, the knowledge of

This work was supported by the National Nature Science Foundation of China (NSFC) under Grant No. 61931001 and 61771194, respectively. Part of results was published by 2018 IEEE 12th International Symposium on Antennas, Propagation and EM Theory (ISAPE), 3-6 Dec. 2018, Hangzhou, China.

F. Du, X. Zhao, Y. Zhang, Y. Wen, Z. Fu, S. Geng, P. Qin, Z. Zhou and C. Xu are with the State Key Laboratory of Alternate Electrical Power System with Renewable Energy Sources, School of Electrical and Electronic Engineering, North China Electric Power University, Beijing 102206, China, and with Hebei Key Laboratory of Power Internet of Things Technology, North China Electric Power University, Baoding, 071003, Hebei, China. (Corresponding author: Xiongwen Zhao, email: zhaoxw@ncepu.edu.cn).

Y. Liu is with China Research Institute of Radiowave Propagation, Qingdao 266107, China. (email: liuys_22s@qq.com).

W. Fan is with Department of Electronic Systems, Aalborg University, Denmark. (email: wfa@es.aau.dk).

the number of clusters is required for the FCM algorithm and it is typically unknown in practice. In [12] and [13], the model-based validation is proposed to find out the desired number of clusters, in which several clustering validity indices are employed to address the problem of the unknown number of clusters. In [14], a new initialization step and balanced MPC distance for the KPM were proposed, which was based on MPC density estimation to calculate the cluster centroids. The proposed initialization step can reduce the computational effort caused by performing the clustering for different number of clusters. In [15], the density-based spatial clustering for application with noise algorithm [16] was proposed to identify local MPCs without knowing the number of clusters as a priori. However, the clusters cannot be distinguished by the presented method with high cluster density. In [17], an improved MCD algorithm was proposed by only considering the distances between the MPCs as clustering basis without taking into account cluster delay and angular spread. In [18], a kernel-power-density-based (KPD) algorithm was proposed without a-priori information, such as the number of clusters. However, it is required that the number of MPCs inside a cluster should be fixed, and the cluster density is calculated based on its fixed nearest MPCs. Therefore, the KPD cannot distinguish a cluster accurately when the number of MPCs inside the cluster is small. In our previous work of [19], we briefly introduce the basic idea of the AKPD algorithm to cluster MPCs. In this paper, we extend our previous work by introducing the measured, simulated channel data and clustering evaluation indices to analyze the performance of the algorithm, and further propose a novel clustering framework by using the support vector machine (SVM) learning model.

In this paper, we firstly propose an adaptive kernel-power-density (AKPD) based clustering algorithm, where an adaptive number of nearest MPCs are chosen to calculate cluster densities. Compared with the classical clustering algorithms as aforementioned, our proposed AKPD algorithm does not need to adjust cluster parameters manually, yet it can distinguish clusters accurately with similar cluster densities and small MPCs within a cluster. In the AKPD algorithms, the MPCs lying at a large distance from the cluster centroids will be clustered into surrounding clusters when the clusters are closely-spaced. To address this problem with the AKPD algorithm, a novel SVM assisted AKPD strategy (SVM-AKPD) is proposed in this paper. The SVM, which is a popular supervised machine learning model that analyzes data for classification and regression analysis, is applied to further improve the clustering algorithm performance in the work. Compared with the traditional algorithm, the proposed SVM-AKPD does not need any prior knowledge and offers automatic cluster identification with excellent performance. Furthermore, the idea of full partition of feature space by the optimal hyperplane in the SVM makes it hard to be affected by irregular MPC distribution. To validate the proposed AKPD and SVM-AKPD algorithms, channel sounding data at 28 GHz mmWave band and simulated channel data generated by the quasi-deterministic radio channel generator (QuaDRiGa) [20] simulation platform is utilized. The proposed algorithms are demonstrated to outperform the reported KPD, KPM and MCD algorithms without prior knowledge of the clusters.

The rest of the paper is organized as follows. Section II describes the double-directional multipath channel and related conventional clustering algorithms. In section III, the proposed AKPD and SVM-AKPD clustering algorithms are introduced. Section IV presents algorithm analysis. Section V is the performance evaluation of the proposed algorithms. Finally, conclusions are drawn in Section VI.

II. Problem Statement

In this section, firstly, we introduce the double-directional radio channel model and the cluster concept. Then, the KPD, KPM and MCD algorithms are briefly summarized, which are widely used in MPC clustering and will be used in this paper as reference methods.

A. Channel Description

The double-directional channel model [21], which has been widely used for MIMO system design, has been considered in this work. For a channel snapshot, the original ray-based channel impulse response (CIR) can be expressed as (1)

$$\begin{aligned} &h(\tau, \Omega_T, \Omega_R, \Theta_T, \Theta_R) \\ &= \sum_{l=1}^{L} \alpha_l e^{j\phi_l} \delta(\tau - \tau_l) \times \delta(\Omega_T - \Omega_{T,l}) \\ &\times \delta(\Omega_R - \Omega_{R,l}) \times \delta(\Theta_T - \Theta_{T,l}) \times \delta(\Theta_R - \Theta_{R,l}) \end{aligned} \tag{1}$$

where

- $\tau, \Omega_T, \Omega_R, \Theta_T, \Theta_R$ are the delay, azimuth angle of departure (AoD), elevation angle of departure (EoD), azimuth angle of arrival (AoA), and elevation angle of arrival (EoA) of the MPCs, respectively, and the subscripts T and R represent transmitter and receiver, respectively,
- L is the total number of MPCs,
- $\alpha_l e^{j\phi_l}$ is the complex amplitude (gain) of the l-th MPC, where α_l is the absolute amplitude and ϕ_l is the phase in the range $[0, 2\pi]$,
- $\delta(\cdot)$ is the Dirac delta function,
- $\tau_l, \Omega_{T,l}, \Omega_{R,l}, \Theta_{T,l}, \Theta_{R,l}$ are the excess delay, excess AoD, excess EoD, excess AoA and excess EoA of the l-th MPC.

Hence the parameter set for the l-th MPC is

$$x_l = \{\alpha_l, \tau_l, \Omega_{T,l}, \Omega_{R,l}, \Theta_{T,l}, \Theta_{R,l}\}. \tag{2}$$

The set of all the MPCs for one snapshot can be denoted by $\Phi = \{x_l | l = 1, 2, \cdots, L\}$. The multipath parameters in a snapshot can be obtained e.g., by using space-alternating-generalized expectation-maximization (SAGE) [22], RiMAX [23] or other multipath estimation algorithms [24] [25].

The objective of clustering is to group MPCs with similar channel parameters into clusters in the joint space and delay

domain according to their parameters. After clustering, the CIR can be expressed as (3),

$$\begin{aligned} &h(\tau,\Omega_T,\Omega_R,\Theta_T,\Theta_R)\\ &=\sum_{m=1}^{M}\{\sum_{l=1}^{L_m}\alpha_{m,l}e^{j\phi_{m,l}}\delta(\tau-\tau_m-\tau_{m,l})\\ &\times\delta(\Omega_T-\Omega_{T,m}-\Omega_{T,m,l})\times\delta(\Omega_R-\Omega_{R,m}-\Omega_{R,m,l})\\ &\times\delta(\Theta_T-\Theta_{T,m}-\Theta_{T,m,l})\times\delta(\Theta_R-\Theta_{R,m}-\Theta_{R,m,l})\} \end{aligned} \tag{3}$$

where

- M is the number of clusters,
- L_m is the number of MPCs in the m-th cluster, and the subscript $\{m,l\}$ means the l-th MPC in the m-th cluster.

B. The KPD Algorithm [18]

The KPD is a density based clustering algorithm, which incorporates the modeled behavior of MPCs by considering MPCs power. The detailed description of the KPD can be found in [18]. A fixed $K=\sqrt{L/2}$ is set for all MPCs as suggested in the KPD, the main steps are as follows:

1) Calculating Density of MPCs: For each MPC sample x_i, use the K nearest MPCs (i.e., set I_i) to calculate the density ρ as follows [18]:

$$\begin{aligned} \rho_{x_i}=&\sum_{x_j\in I_i}\exp(\alpha_{x_j})\times\exp\left(-\frac{|\tau_{x_i}-\tau_{x_j}|^2}{(\sigma_\tau)^2}\right)\\ &\times\exp\left(-\frac{|\Omega_{T,x_i}-\Omega_{T,x_j}|}{(\sigma_{\Omega_T})}\right)\times\exp\left(-\frac{|\Omega_{R,x_i}-\Omega_{R,x_j}|}{(\sigma_{\Omega_R})}\right)\\ &\times\exp\left(-\frac{|\Theta_{T,x_i}-\Theta_{T,x_j}|}{(\sigma_{\Theta_T})}\right)\times\exp\left(-\frac{|\Theta_{R,x_i}-\Theta_{R,x_j}|}{(\sigma_{\Theta_R})}\right) \end{aligned} \tag{4}$$

where $\sigma_{(\cdot)}$ is the standard deviation of the MPCs in $(\cdot)$ domain.

2) Calculating Relative Density: For each MPC sample x_i, use the K nearest MPCs' density to calculate the relative density ρ^* as follows [18]:

$$\rho^*_{x_i}=\frac{\rho_{x_i}}{\max\limits_{x_j\in I_i\cup\{x_i\}}\rho_{x_j}}. \tag{5}$$

3) Searching Initial Cluster Centroids: For each MPC sample x_i, satisfying $\rho^*_{x_i}=1$, label it as the key MPC which is the initial cluster centroid, then the set of initial centroids is

$$\hat{\Phi}:=\{x_i|x_i\in\Phi,\rho^*_{x_i}=1\}. \tag{6}$$

4) Clustering: For those non-key MPCs of x_i, its high-density-neighboring [26] MPC $\tilde{x}_i$ is defined as

$$\tilde{x}_i:=\underset{x_j\in\Phi,\rho^*_j>\rho^*_i}{\arg\min}\ d(x_i,x_j) \tag{7}$$

where d represents the Euclidean distance. With the high-density-neighboring MPC, the non-key MPCs will reach the key MPC and be grouped as one cluster based on the concepts and properties of graph theory [27]. It should be noted that the cluster will only include the key MPC if the corresponding key MPCs cannot be reached.

5) Cluster Merging: Any two clusters can be merged and they are considered as one cluster if the there exist a path where the density of the MPCs are larger than a specific density threshold.

The KPD incorporates the modeled behavior of MPCs and considers the MPCs power [18], and it does not need the prior knowledge about the clusters. However,it is difficult for the KPD algorithm to distinguish the cluster with small number of MPCs, since only fixed *K* nearest MPCs is considered to estimate density. Moreover, too many cluster centriods will be estimated, as natural clusters have small scale fading and intra-cluster power variation [18], which need to merge the clusters close to each other. The additional step of cluster merging will increase the complexity of the algorithm.

C. The MCD Algorithm

The MCD-based method is adopted to group the MPCs [28]. In [28], the MCD between the i-th and j-th MPCs ($i\neq j$) is given by:

$$MCD_{i,j}=\sqrt{MCD^2_{AOA,ij}+MCD^2_{\tau,ij}} \tag{8}$$

where $MCD^2_{AOA,ij}$ and $MCD^2_{\tau,ij}$ are the MCD in delay and angle domain, respectively, $MCD^2_{\tau,ij}$ is calculated as:

$$\begin{aligned} &MCD_{\tau,ij}=\varsigma\frac{|\tau_i-\tau_j|\delta_\tau}{\Delta\tau^2_{\max}}\\ &\Delta\tau_{\max}=\max\{|\tau_i-\tau_j|;\forall i,j\in\{1,\ldots,M\}\} \end{aligned} \tag{9}$$

where ς is a delay scaling factor to balance the weights of the MCD in delay and angle domain. The $MCD^2_{AOA,ij}$ is calculated as:

$$MCD_{AOA,ij}=\frac{1}{2}\left|\begin{pmatrix}\cos\Omega_i\cos\Theta_i\\ \cos\Omega_i\sin\Theta_i\\ \sin\Omega_i\end{pmatrix}-\begin{pmatrix}\cos\Omega_j\cos\Theta_j\\ \cos\Omega_j\sin\Theta_j\\ \sin\Omega_j\end{pmatrix}\right|. \tag{10}$$

The implementation of the algorithm consists of the following three steps as in [29]:

1) : Choose a reference MPC which has the largest power among all the MPCs in a set eligible for extracting clusters.

2) : Calculate the MCD between the reference MPC and all other MPCs in the set, select those MPCs with the MCDs less than a predefined threshold denoted with MCD_{th}, and group them together with the reference MPC as one cluster.

3) : Remove the MPCs already allocated to a cluster from the MPC set, and re-execute step (1) to find the next cluster. This procedure stops until all the MPCs are assigned to certain clusters.

The MCD algorithm is simple to implement and has a fairly low computational complexity. However, the parameter ς and MCD_{th} are determined by a visual inspection evaluated by whether the clustering results can map to the physical environment.

D. The KPM Algorithm [10]

The KPM algorithm is an extension of the K-means algorithm proposed in [30], in which MCD is used as the distance matric considering the MPCs power, the detailed description of the KPM can be found in [10], the main steps are as follows:

1) : Randomly choose initial cluster centroid positions from the MPCs set.

2) : Allocate MPCs to cluster centroids by minimizing the total sum of power weighted distance between the MPC and assigned cluster centroid.

3) : Recalculate the cluster centroids by the intra-cluster MPCs.

4) : Repeat steps 2 and 3 until the cluster centroid is no longer updated. In this paper, a combined validation [10] is used to choose the optimum number of clusters to improve the performance of the KPM algorithm.

III. Proposed Clustering Algorithm

As explained, the conventional KPD algorithms would fail to work when there are only a small number of MPCs inside the cluster. This will become more problematic for high frequency mmWave channels due to its sparse nature [31] [32]. Therefore, in this work, we firstly proposed an AKPD algorithm, where a new distance-based metric is proposed to calculate an *adaptive-K* for each MPC, and we use the *adaptive-K* nearest MPCs to calculate the densities and relative densities for all MPCs in the AKPD algorithm. The main difference between the proposed AKPD and the traditional KPD is that the AKPD uses *adaptive-K* nearest MPCs to calculate the densities and relative densities for all MPCs, while the KPD uses a fixed K. Thus the proposed AKPD algorithm can be used in scenarios where we have more complex distribution of MPCs, as the density of MPCs can be calculated with higher accuracy with the *adaptive-K*.

As we explained, thanks to the small-scale fading and the intra-cluster power variation problem, there is a need for an additional cluster-merging step for conventional KPD. Without such cluster-merging operation, there might exist too many initial clusters based on the estimated key MPCs. This problem is solved in the proposed AKPD algorithm, since the *adaptive-K* enables the algorithm to be better adapted to the cluster densities and their distribution.

However, the relative density variation caused by small-scale fading and intra-cluster power variation will lead to mis-clustering when the clusters are closely-spaced since the AKPD clusters the MPCs according to the high-density-neighboring MPCs. To overcome the limitations of the AKPD, we proposed a novel machine-learning inspired SVM-AKPD algorithm.

A. The AKPD Algorithm

In this part, we firstly discuss how to calculate the *adaptive-K*, which is a key improvement compared to the conventional KPD.

A fixed value $K = L/2$ is initially set for all MPCs, and then value K will be updated. The following are the steps of using the distance-based method to calculate the *adaptive-K*:

1) Calculating Distance Between the MPCs: For MPCs x_i and x_j, the distance between MPCs can be calculated by

$$\hat{d}_{i,j} = [(\tau_i{}' - \tau_j{}')^2 + (\Omega_{T,i}{}' - \Omega_{T,j}{}')^2 + (\Omega_{R,i}{}' - \Omega_{R,j}{}')^2 + (\Theta_{T,i}{}' - \Theta_{T,j}{}')^2 + (\Theta_{R,i}{}' - \Theta_{R,j}{}')^2]^{\frac{1}{2}}. \tag{11}$$

In (11), the parameters τ, Ω or Θ have different dimensions, and they are normalized by

$$X_i{}' = \frac{X_i - \min\limits_{l\in[1,L]} X_l}{\max\limits_{l\in[1,L]} X_l - \min\limits_{l\in[1,L]} X_l} \tag{12}$$

where X represents a specific parameter of MPCs. In [14], a similar normalization of MCD is proposed for a fair treatment of the different parametric dimensions of the estimated MPCs. The distance calculation method between MPCs is also used in the KPD algorithm in this paper.

2) Finding the Leaping Separation Distance: Define $d_i := \{d_{i,1}, \ldots, d_{i,j}, \ldots, d_{i,L}\}$ is the descending order of $\{\hat{d}_{i,1}, \ldots, \hat{d}_{i,j}, \ldots, \hat{d}_{i,L}\}$, then the separation distance $\Delta d_i := \{\Delta d_{i,1}, \ldots, \Delta d_{i,j}, \ldots, \Delta d_{i,K-1}\}$ can be calculated by

$$\Delta d_{i,j} = d_{i,j+1} - d_{i,j},\ d_{i,j} \in d_i, j \in [1, K-1]. \tag{13}$$

3) Calculating adaptive-K: If there exists a large separation distance $\Delta d_{i,j}$ which is bigger than a threshold, we regard that the MPCs x_i and x_j belong to different clusters and K is updated to *adaptive-K* = j. However, the selection of the threshold is crucial in the algorithm. To set the threshold based on the distribution of the MPCs automatically, the maximum distance between x_i and all the MPCs is utilized to avoid misjudging the leaping separation distance that may occur in an intra-cluster MPCs. Therefore, the threshold $\Delta d_{\text{threshold}}$ is defined as follows:

$$\Delta d_{\text{threshold}} = \mu \times d_{\max} \tag{14}$$

$$d_{\max} = \max_{j\in[1,L]} \hat{d}_{i,j} \tag{15}$$

where μ is the scale factor between $\Delta d_{\text{threshold}}$ and $d_{\max}$. It should be noted that the concept of a leaping separation distance has been used to indicate a separation between two clusters in [13].

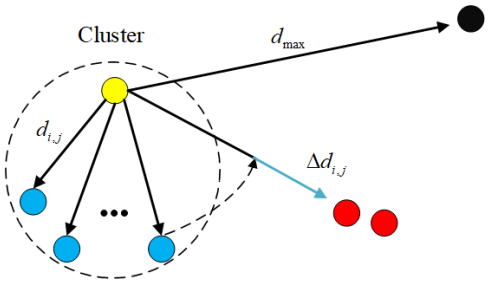


Fig. 1. The schematic diagram of the distance-based metric.

Fig. 1 shows the schematic diagram of the distance-based metric, where the yellow solid circle is the estimated MPC x_i, the blue solid circles represent intra-cluster MPCs with x_i, the black solid circle is the farthest MPC from x_i, and the red solid circles are the K nearest MPCs of x_i which does not belong to the same cluster with x_i. $d_{i,j}$ is the distance between MPCs x_i and x_j, $d_{\max}$ is the biggest distance between x_i and all MPCs, $\Delta d_{i,j}$ is the leaping separation distance.

The pseudo-code of *adaptive-K* calculation is presented as follows:

The AKPD can be implemented by the following steps:

Algorithm 1: The Distance-based metric

Input: $\Phi = \{x_1, \ldots, x_L\}, \mu$
$K = L/2$
for $i \in [1, L]$ **do**
  for $j \in [1, L]$ **do**
    Calculate $\hat{d}_{i,j}$ according to (11)
  end for
  $d_i := \{d_{i,1}, \ldots, d_{i,j}, \ldots, d_{i,L}\}$ = *descend* $\{\hat{d}_{i,1}, \ldots, \hat{d}_{i,j}, \ldots, \hat{d}_{i,L}\}$
  Calculate $\Delta d_{\text{threshold}}$ according to (14)
  for $j \in [1, K-1]$ **do**
    Calculate $\Delta d_{i,j}$ according to (13)
    if $\Delta d_{i,j} > \Delta d_{\text{threshold}}$ **then**
      $Adaptive - K_i = j$
    else
      $Adaptive - K_i = K$
    end
  end for
end for
Output: $Adaptive - K_1$,..., $Adaptive - K_L$

1) Calculating adaptive-K: For each MPC sample x_i, $i \in [1, L]$, the distance-based metric discussed is applied to calculate *adaptive-*K_i, then the set of the *adaptive-*K_i nearest MPCs of x_i is as follows:

$$I_i := \{x_1, x_2, \ldots, x_{adaptive-K_i}\}. \tag{16}$$

2) Calculating Density: For each MPC sample x_i, use the *adaptive-K* nearest MPCs to calculate the density ρ by (4).

3) Calculating Relative Density: For each MPC sample x_i, calculate the relative density ρ^* using the *adaptive-K* nearest MPCs' density by (5). It should be noted that the relative density is equal to 1 when there is only one MPC inside one cluster.

4) Searching Initial Cluster Centroids: For each MPC sample x_i, satisfying $\rho^*_{x_i} = 1$, label it as the key MPC which is the initial cluster centroid.

5) Clustering: Cluster the MPCs according to the high-density-neighboring MPCs as mentioned in the KPD.

With the using of the proposed distance based matric, the number of nearest MPCs of the estimated object is no longer a fixed value, but is calculated based on the actual MPCs distribution. The impact of *adaptive-K* on the clustering results will be analyzed from two aspects: 1) Density, in (4), the density of MPCs is calculated based on the nearest MPCs in the set of *I*, it is obvious that the fixed *K* used for all MPCs will lead to the inaccurate calculation. 2) Relative density, compared with the fixed *K*, if the *K* is smaller than the *adaptive-K*, the relative density would be larger while too small MPCs are used to calculated the relative density, which will lead to the problem that too many cluster centroids are estimated according to the key MPCs. If the *K* is larger than the *adaptive-K*, the relative density may be smaller while too many MPCs are used to calculated the relative density, which will lead to the problem that there are no key MPCs in the clusters with small number of MPCs. Thus, the *adaptive-K* ensures that the MPCs used to calculate the density and relative density belong to the same cluster as the estimated MPCs and adapt to varying densities and distributions of MPCs.

The pseudo-code of the AKPD algorithm is presented in Algorithm 2.

Algorithm 2: The AKPD algorithm

Input: $\Phi = \{x_1, \ldots, x_L\}, \mu$
Calculate *adaptive-K* according to the distance-based method
for $i \in [1, L]$ **do**
  calculate ρ_{x_i} according to (4)
end for
for $i \in [1, L]$ **do**
  calculate $\rho^*_{x_i}$ according to (5)
end for
$\hat{\Phi} := \{x_{C1}, x_{C2}, \ldots, x_{CN_{key}} = \{x_i | x_i \in \Phi, \rho^*_{x_i} = 1\}\}$
for $i \in [1, L]$ **do**
  find $\tilde{x}_i$ according to (7)
end for
$E_1 = \{(x, \tilde{x}) | x \in \Phi \setminus \hat{\Phi}\}$
$\zeta_1 = (\Phi, E_1)$
for $i \in [1, N_{\text{key}}]$ **do**
  $\Phi_i = \{x_{Ci}\}$
  for $j \in [1, L]$ **do**
    if x_{Ci} *is reachable from* x_j *in* ζ_1 **then**
      $\Phi_i = \Phi_i \cup \{x_j\}$
    end
  end for
end for
Output: Φ_1,...,$\Phi_{N_{\text{key}}}$

B. The SVM-AKPD Algorithm

In clustering step (5) of the AKPD algorithms, the MPC is aggregated via the closest MPCs with higher relative density i.e., high-density-neighboring MPCs as explained in (7). However, the MPCs lying at a large distance from the cluster centroids or at the edge of the cluster, will not be able to seek the proper high-density-neighboring MPCs when the clusters are closely-spaced or the shape of the MPCs distribution is not regular. In this case, the high-density-neighboring MPCs of the clustered objects will be misjudged as the closest the higher-density MPCs belong to the surrounding clusters owning to the relative density variation. To overcome those problems of the AKPD, the SVM is introduced for channel clustering.

SVM was proposed by Vapnik [33], which is called maximum margin classifier. The SVM is a linear classifier that learns by minimizing the generalized error based on classifier with kernel function, which is lowest in classification complexity of learning system [34]. The purpose of the SVM is to find an optimal hyperplane by maximizing the margin between the separating hyperplane and the data. As shown in the Fig. 3, the two kinds of samples delegated by the yellow points and blue points are divided by line H. Two lines H_1 and H_2

are parallel to H, and pass through the two kinds of samples that are nearest to H. The distance between them is called the classification margin. The optimal classification line is the one that can divide the two types of samples correctly, and also makes the margin to be the largest. H in Fig. 2 is the optimal classification line. In a high-dimensional space, the optimal classification line will become the optimal hyperplane.

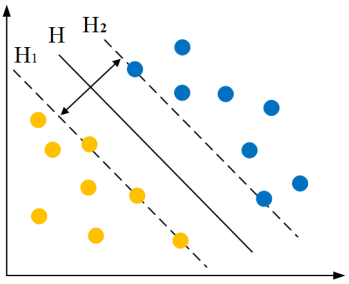


Fig. 2. The basic idea of SVM classification.

A cluster is a convex set composed of MPCs with similar parameters, and any two clusters can be separated linearly. For a linearly separable convex set, there must be support vector and hyperplane to separate different clusters to feature space. Thus, the SVM is feasible for channel clustering. However, one discrepancy is that the SVM is a kind of supervised learning algorithm, while the MPCs are the data without label. Therefore, in the proposed SVM-AKPD algorithm, the cluster centroids calculated by the partial AKPD are used to train the SVM model, which is regarded as labeled data. Compared with the traditional algorithm, the SVM-AKPD does not need any a-prior information. Furthermore, the SVM-AKPD can better adapt to the irregular MPC distribution because of its idea of full partition of feature space. Firstly we search the initial cluster centroids as labeled key MPCs using the AKPD algorithm, and secondly, we train the labeled data using one-versus-rest (OvR) SVM algorithm where OVR is a decomposition methods of multi-class SVM by reconstructing a multi-class classifier from binary SVM-based classifiers. After the above two stages the feature space is divided by hyperplane, then MPCs will be clustered into different feature space.

The SVM-AKPD algorithm is implemented by the following steps.

1) Searching Initial Cluster Centroids: Calculate the key MPCs using the steps 1-4 of the AKPD.

2) Clustering by the OvR SVM: Train the initial cluster centroids to the obtain SVM classifiers, namely the classification model, and classify all MPCs using classification model to implement clustering. Consider the key MPCs set C with N_{key} data samples and M_{key} features. The training data is then represented as

$$C = \{(x_i, y_i) | x_i \in \hat{\Phi}, y_i \in L_{\text{key}}\}, \ i = 1, 2, \ldots, N_{\text{key}} \quad (17)$$

where x_i is small scale parameters vector of the key MPCs, y_i and L_{key} are the label of *i*-th key MPC i.e., *i*-th cluster and the set of total label set, respectively. The clustering of MPCs can be considered as multi-classification with N_{key} cluster categories essentially. The OvR firstly decomposes multi-classification into multiple binary SVM classifications by dividing the multi-class data into N_{key} data sets, which is a basic strategy to solve multi-classification problems by using binary classifier. For *j*-th binary SVM classification, it considers the key MPC with *j*-th cluster label as positive class and the rest of the other data samples as negative class in dataset $C_j, 1 \leq j \leq N_{\text{key}}$. The clustering result of new MPC samples is determined by combining the cluster labels which is predicted from all the SVM classifiers. Suppose the multiple binary SVM classifier are $f_1, f_2, \ldots, f_{N_{\text{key}}}$, the final clustering result of a MPC sample x is decided by

$$c(x) = \arg\max_i f_i(x). \quad (18)$$

A schematic diagram of the OvR SVM is shown in Fig. 3, the blue and yellow line represent the training and predicting process of the OvR SVM, respectively.

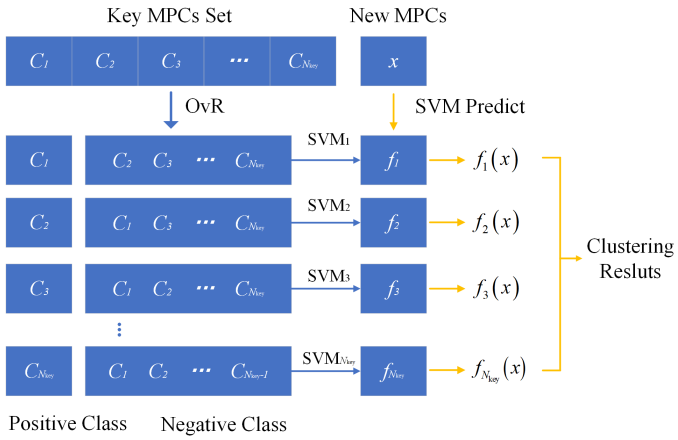


Fig. 3. The schematic diagram of the OvR SVM.

In the process of training the SVM, the set of box constraint (i.e., penalty parameter) and kernel function are crucial. For the box constraint, the training algorithm must allow some error classification in the training set while the data cannot be completely separated. In this case, the higher the box constraint is, the higher the cost of error classification points will be. Thus, it will make the data classification stricter. In this paper, we set box constraint to a smaller value as the cluster is linearly separable essentially. For the kernel function, different kernel functions will greatly affect the classification effect, and the commonly used kernel functions are Logistic Regression (LR), Linear and Gaussian Kernel function. Generally, when the feature number is large and close to the sample number, LR is applied or the SVM is without kernel function (Linear Kernel function). The Gaussian Kernel function is used when the feature number is small and the sample number is normal. When the feature number is small and the sample number is large, the extra features should be added manually to apply the LR or Linear Kernel function. In the channel cluster, the number of features (delay, AOA, EOA) is smaller than the number of samples (key MPCs), thus the Gaussian Kernel function is used to train the SVM to achieve the best clustering performance.

The pseudo-code of the SVM-AKPD algorithm is presented in Algorithm 3.

Algorithm 3: The SVM-AKPD algorithm

Input: $\Phi = \{x_1, \ldots\}, \mu$
Calculate $adptive - K$ according to distance-based method
for $i \in [1, L]$ **do**
    calculate ρ_{x_i} according to (4)
end for
for $i \in [1, L]$ **do**
    calculate $\rho^*_{x_i}$ according to (5)
end for
$\hat{\Phi} := \{x_{C1}, x_{C2}, \ldots, x_{CN_{\text{key}}} = \{x_i | x_i \in \Phi, \rho^*_{x_i} = 1\}\}$
for $i \in [1, N_{\text{key}}]$ **do**
    $label = zeros(1, N_{\text{key}})$
    $label\{i\} = 1$
    $SVMmodel\{i\} = fitcsvm(\hat{\Phi}, label)$
end for
for $i \in [1, L]$ **do**
    for $j \in [1, N_{\text{key}}]$ **do**
        $score_{i,j} = predict(SVMmodel\{j\}, x_j)$
    end for
    for $j \in [1, N_{\text{key}}]$ **do**
        if $score_{i,j} == \max(score\{i, :\})$ **then**
            $c_i = j$
        end
    end for
    $\Phi_{c_i} = \Phi_{c_i} \cup x_i$
end for
Output: $\Phi_1, \ldots, \Phi_{N_{\text{key}}}$

C. Clustering algorithm performance evaluation

In this section, the F measure [35] and Silhouette coefficient *S* [36] are used to evaluate the clustering performance. The Silhouette coefficient *S* is used to evaluate the performance of different clustering algorithms based on the measured data, it is a matric of the tightness of all MPCs in the clusters. As the ground-truth is not available in [37], the Silhouette coefficient *S* can be used to compare the performance of different clustering algorithms, as it only depends on the actual distribution of clusters. The definition of the *S* is as follows:

$$S = E\left[\frac{b(x) - a(x)}{\max\{b(x), a(x)\}}\right]_{x \in \Phi} \tag{19}$$

where $a(x)$ denotes the mean distance between x and MPCs belonging to the same cluster, $b(x)$ denotes the minimum distance between x and MPCs belonging to the different cluster. The detailed definition of $a(x)$ and $b(x)$ can be found in [36]. *S* lies within $[-1, 1]$, and a larger value means a better clustering performance.

The F measure is a robust quality measure that can be used to balance the precision and recall [38]. The ground-truth partition needs to be available in the F measure. The ground-truth partition is often available in the simulated channel data, which is not possible with the measured channel data. The definition of the F measure is given as follows:

$$F = \frac{2 \cdot PB \cdot RB}{PB + RB}. \tag{20}$$

where PB and RB denote the average of BCubed precision and recall, and the detailed definition of PB and RB can be found in [39]. As the Silhouette coefficien, F measure lies within $[-1, 1]$, and a larger value means a better clustering performance.

IV. Algorithm Analysis

In this section, the clustering results of the proposed AKPD and SVM-AKPD algorithms are analyzed based on the measured channels. Firstly, the measurement system and campaigns are introduced, and the measured MPCs are exacted by the SAGE algorithm in delay and angular domains. Then, the measurement data in the NLoS scenario is used as an example to demonstrate the performance of the proposed AKPD and SVM-AKPD clustering algorithms compared with the conventional algorithm.

A. Measurement System and Campaigns

The channel measurement was carried out in the large waiting hall at Qingdao high-speed railway station at 28 GHz with 500 MHz bandwidth. The measurement environment and layout are shown in Figs. 4 (a) and (b), respectively. The transmitter (Tx) and receiver (Rx) are set about the same height of 2 m, a vertically polarized omni-directional antenna was used in the Tx, and a vertically polarized uniform linear array (ULA) with 8 array elements was applied in the Rx. The eight ULA antennas are connected with eight individual RF channels at Keysight receiver. The detailed information of the Keysight time domain channel sounder and the calibration method are described in [17] [40]. It should be noted that the channel sounding with restricted/limited technical setup samples only measures the partial wireless channel and subsequent data analysis would lead to limited even erroneous results. The measured channel data quality is determined by the channel sounder system parameters, e.g., code length, system bandwidth, system dynamic range, antenna array configuration, and so on. By moving the ULA eight times in the horizontal plane, a large scale 1×64 virtual ULA can be formed with inter-element spacing set to 0.5λ at 28 GHz. There were in total 21 and 12 measurement positions (indicted by numbers with yellow circles) in the line-of-sight (LoS) and non-line-of-sight (NLoS) scenarios as shown in Figs. 4 (c) and (d), respectively. The measurements were performed at midnight to avoid moving people. The hall itself and the shops around are equipped with glass windows, which implies rich scattering propagation environment. The measurement system parameters and detailed measurement information can be found in the Table I in [41].

B. Clustering Results Analysis

In the AKPD and SVM-AKPD algorithms, the initial value of μ is required to be determined where μ factor is used to adjust relative distance between the estimated MPC and all MPCs to achieve the better clustering performance. Inappropriate values of μ will cause the unexpected leaping separation distance. For simplicity, we set in the region of $[0.1, 0, 3]$ in

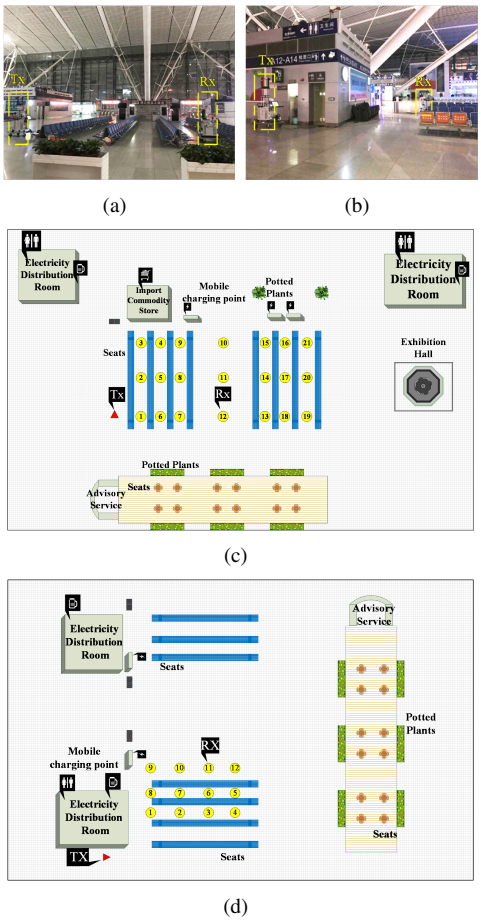


Fig. 4. Measurement environments and layout of the large waiting hall. (a) and (c) are for the LoS scenario, (b) and (d) are for the NLoS scenario, respectively.

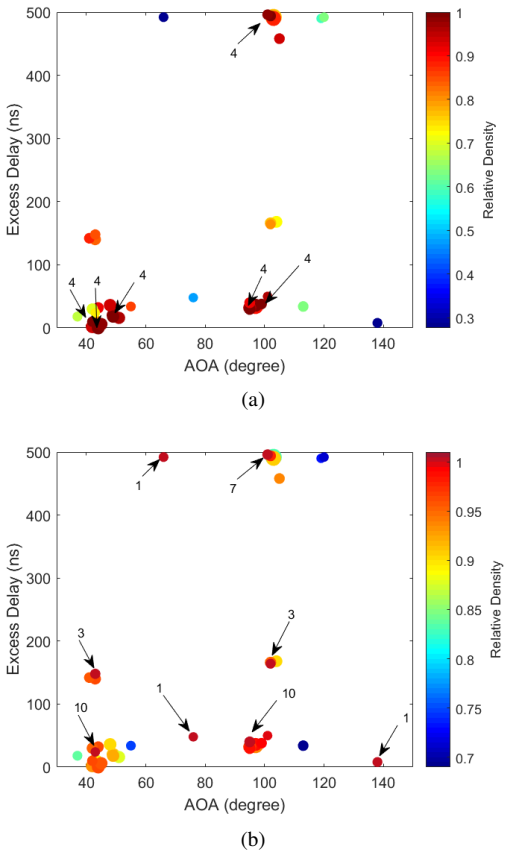


Fig. 5. Relative densities of MPCs calculated with measurement data at the 8-th position in NLoS scenario. (a) KPD, (b) AKPD.

the work, which is found to be a reasonable choice. A heuristic argument for $\mu \in [0.1, 0.3]$ is that a small value of μ ensures that the clusters are separated from each other.

The performance of the proposed AKPD and conventional KPD algorithms are analyzed with the measurement data at the 8-th position in the NLoS scenario. Figs. 5 (a) and (b) show the relative density of the KPD and AKPD algorithms, respectively, the color bar and maker size indicate the value of estimated relative density ρ^* of MPCs, the circles pointed by the arrow are the key MPCs with $\rho^* = 1$ and the value is the *adaptive-K* value of MPCs. In the AKPD, the K of MPCs in different cluster is adaptive, as we can see in Fig. 5 (b), the *adaptive-K* is equal to the number of MPCs inner one cluster. In contrast, the fixed K of all MPCs in the KPD algorithm will lead to inaccurate calculation of the density and relative density in the case of the clusters with small MPCs, in the results of that cluster centroids will be misestimated. As we can see that there are 6 key MPCs calculated by KPD algorithm with 4 nearest MPCs in Fig. 5 (a), while 8 key MPCs are distinguished with *adaptive-K* calculated by the distance-based algorithm in Fig. 5 (b).

In the conventional KPD algorithm, there are usually too many key MPCs according to the fixed K when a large number of MPCs belong to one cluster, and it is necessary to merge those clusters that are fairly close to each other. Fig. 6 shows the clustering results of the KPD and AKPD algorithms. We can see from Fig. 6 (a) that the clusters circled by ellipse with dash line are merged in Fig.6 (b). In addition, in the KPD algorithm, the MPCs circled by ellipses with solid line are clustered into the surrounding clusters incorrectly in Fig. 6 (b), as there are no expected key MPCs among those MPCs as we can see from Fig. 5 (a). Meanwhile, the MPCs circled by ellipses with solid line can be clustered accurately in Fig. 6 (c), which is more in line with the actual cluster distributions. This also verifies the clustering results which are shown in Fig. 5 and intuitively shows the superiority of the AKPD algorithm.

Fig. 7 shows the clustering results of the AKPD and SVM-AKPD with measurement data at the 3-th position in the NLoS scenario. As we explained, the MPCs are clustered by finding the high-density-neighboring MPC in the AKPD, which will lead to the mis-clustering for the MPCs lying at a large distance from the cluster centroids. In the Fig.7 (a), arrow line refers to the process of MPCs clustering, where the black and red arrow lines indicate the right and wrong direction of aggregation, respectively. As we can see in Fig. 7 (b), the MPCs circled by ellipse with solid line are clustered to the cluster circled by ellipse with dash line erroneously. In Fig. 7 (c), scatters with different color are the key MPCs calculated with the steps 1-4 of the AKPD algorithm, and different color regions correspond to different SVM classifiers. Fig. 7 (d)

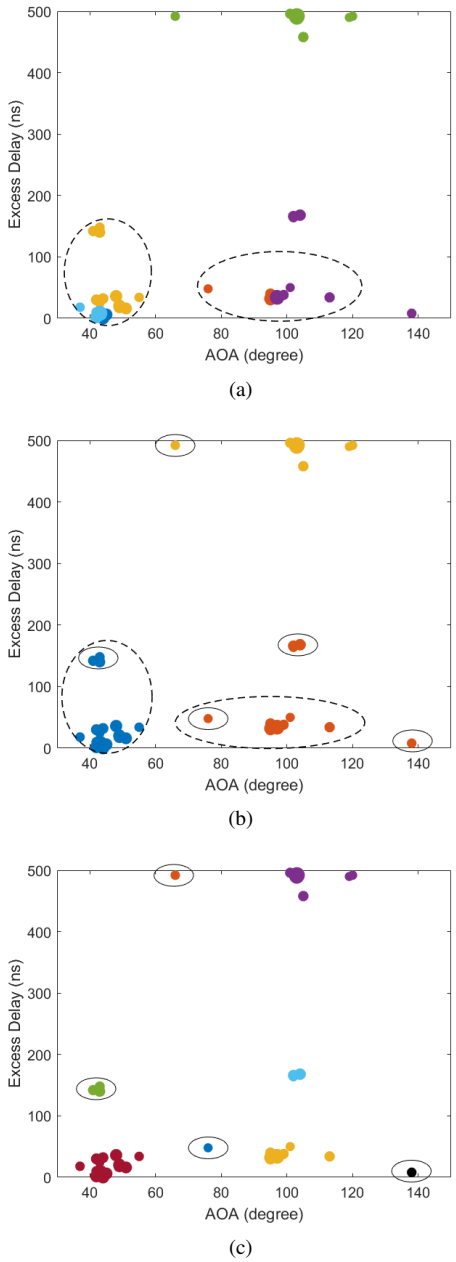


Fig. 6. Clustering results of measurement data at the 8-th position in NLoS scenario. (a) KPD without merging, (b) KPD with merging, (c) AKPD.

is the clustering results by different SVM classifiers where clusters are divided into different regions of feature space. Compared with the AKPD algorithm, SVM-AKPD algorithm has better clustering results.

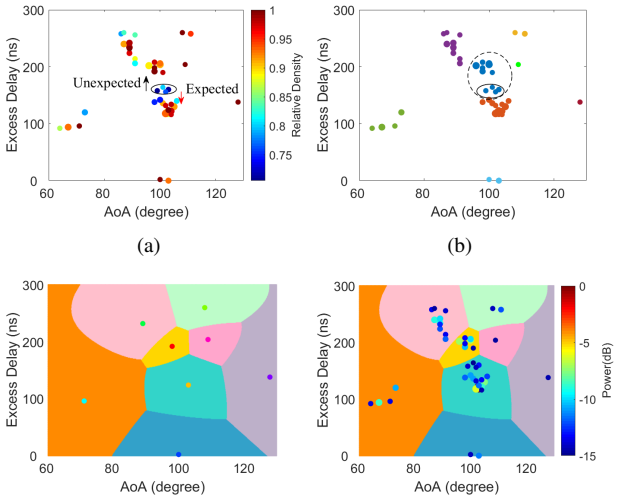


Fig. 7. Clustering results of measurement data at the 3-th position in NLoS scenario. (a) relative density of MPCs, (b) clustering results of AKPD, (c) training results of SVM classifiers, (d) clustering results of SVM-AKPD.

V. Performance Evaluation

In this section, the performance of the proposed algorithms is evaluated based on the measured and simulated channels with the AKPD, SVM-AKPD, KPD, KPM and MCD algorithms, respectively. The simulated channels are generated by 5G channel simulator QuaDRiGa and the synthetic MPCs are extracted. The QuaDRiGa model is an enhancement of the Wireless World Initiative for New Radio (WINNER) model [42] and it uses random statistical distribution to generate scatterers. The QuaDRiGa channel model follows a geometry-based stochastic channel modeling approach, which allows the creation of an arbitrary double directional radio channel. The channel parameters are determined stochastically, based on statistical distributions extracted from channel measurements. The distributions are defined for, e.g., delay spread, angle spread, shadow fading and cross-polarization ratio. Specific channel realizations are generated by summing contributions of rays with specific channel parameters, e.g., delay, power, angle-of-arrival and angle-of-departure. Furthermore, the F measure and Silhouette coefficient S are used to validate the performance of the different clustering algorithms, which evaluates the compactness and separation of the cluster. For the simulated channels, the ground-truth partitions are available [37], the number and spread of clusters are synthesized, which can be easily adjusted and therefore can allow us to test algorithm performance under different conditions. Note that the ground truth is not available and the

TABLE I
THE STATISTICAL PARAMETERS CALCULATED BY DIFFERENT ALGORITHMS BASED ON THE SIMULATED CHANNEL DATA

Algorithms	Number of clusters	Cluster DS [ns]		Cluster ASA [deg]		Cluster ESA [deg]	
		μ	δ	μ	δ	μ	δ
Ground-Truth	12	4.3	0.6	10.6	1.7	5.5	1.1
SVM AKPD	12	4.3	0.6	10.6	1.7	5.5	1.1
AKPD	12	4.3	0.6	10.8	1.9	5.6	1.2
KPD	14	4.3	0.7	10.6	1.9	5.5	1.3
KPM	18	4.1	0.7	9.1	3.3	5.3	1.2
MCD	15	4.4	0.6	10.0	2.3	5.5	3.2

data are more affected by noise for the measured channels. We use two measures, i.e., Silhouette coefficient *S* and F measure, for different cases to better validate the AKPD and SVM-AKPD algorithms.

The number of clusters, cluster DS, cluster-wise root-mean-square azimuth spread of arrival angles (cluster ASA) and cluster-wise root-mean-square elevation spread of arrival angles (cluster ESA) are calculated to reveal the impact of the clustering results calculated by different algorithms on the channel modeling. Table I shows the statistical parameters calculated by different algorithms based on the simulated channel data, as the ground-truth is available. The "Ground-Truth" indicates the results calculated with the real cluster label. As shown in the Table I, the number of clusters of the SVM AKPD and AKPD algorithms is 12, which is same as the Ground-Truth, while the KPD, KPM and MCD algorithms present a larger number of clusters. Especially, the number of clusters is 14 for the KPD algorithm. This is because the fixed K will introduce too many calculated initial cluster centroids, in which the cluster merging of the KPD algorithm cannot guarantee that all clusters can be correctly merged. We reach a similar conclusion that the results of the proposed algorithms are closer to the ground truth for the cluster DS, ASA and ESA. It should be noted that the cluster ASA which is set to simulate the channel is small to ensure the available ground truth. When the ASA in the cluster is too large, the number of clusters will be changed (if the MPCs in one cluster are too scattered, these MPCs may no longer be considered as belonging to the same cluster). In this case, the misclustering is introduced by the splitting of cluster, it will not change the value of cluster DS, ASA and ESA too much, but it will indeed change the number of clusters. Thus, the difference of the cluster DS, ASA and ESA is small by using different algorithms.

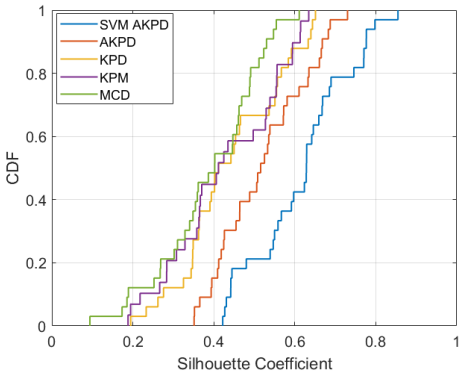


Fig. 8. CDF plots of the estimated Silhouette coefficients with different clustering algorithms.

Fig. 8 shows the cumulative distribution function (CDF) plots of the estimated Silhouette coefficients based on the LoS and the NLoS measured date with different clustering algorithms, namely SVM-AKPD, AKPD, KPD, KPM and MCD. It can be seen from Fig. 8 that the SVM-AKPD and AKPD have better performance than the KPD, KPM and MCD, whereas the MCD performs the worst and the SVM-AKPD performs the best. This is because that the method of determining the cluster centroid in the MCD only considers the power and the MCD ignores the small-scale fading characteristics of the channel. The KPM shows the better performance than the MCD, as the cluster centroids are updated with iteration in the KPM while the MCD always choose the MPCs with the largest power as the reference MPC (i.e., cluster centroid). As for the KPD, the fixed *K* makes the algorithm unable to identify clusters with small MPCs. In contrast, the *adaptive-K* enables the AKPD and SVM AKPD distinguish clusters accurately with similar cluster densities and small MPCs within a cluster. Furthermore, the SVM-AKPD utilize the optimal hyperplane to maximizing the margin between different clusters, which is insensitive to the distribution variation of intra-cluster MPCs.

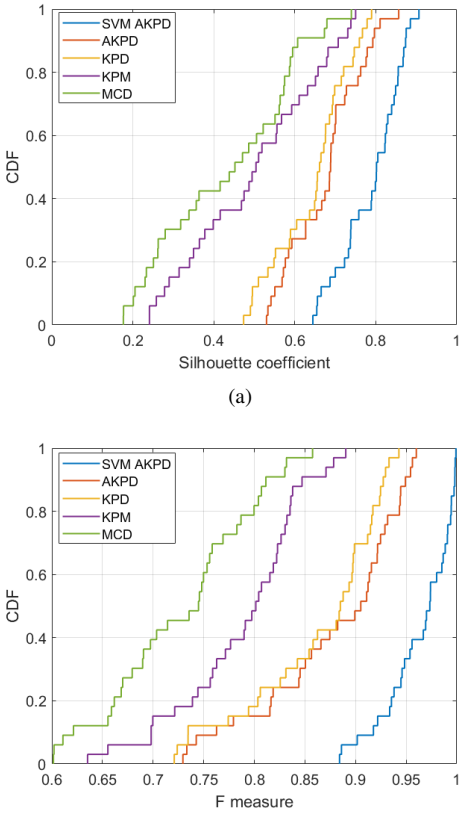
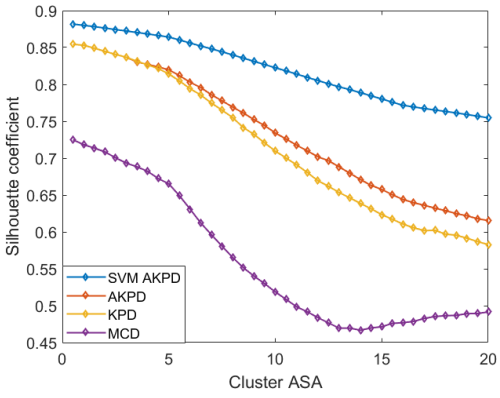


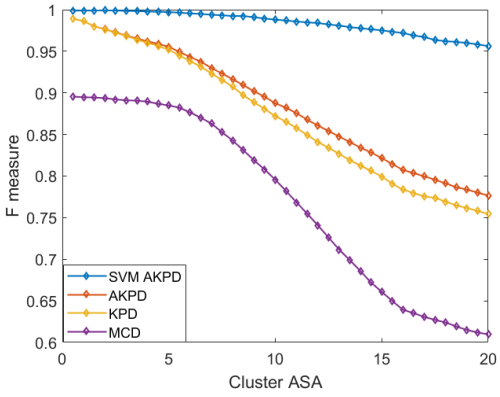
Fig. 9. CDF plots with different cluster algorithms, based on simulated channels. (a) Silhouette coefficient, (b) F measure.

Figs. 9 (a) and (b) show the CDF comparison of Silhouette coefficients and F measure with different clustering algorithms based on simulated channels. For the simulated channels, the large scale parameters calculated with the experimental LoS measurements [1] are used in the QuaDRiGa simulator. By taking a fixed number of clusters, 40 random simulated channels with different cluster ASA [43] are generated for each position. Thus 840 simulated channels are generated with 12 clusters for each channel and 20 sub-MPCs in each cluster. It can be seen that the SVM-AKPD offers the best performance in Fig. 9. It is noteworthy that the value of Silhouette coefficient is difficult to reach 1. In (19), if $a(x)$ is equal to 0, then S is equal to 1, which means that there is only one MPC in each cluster, and it is almost non-realistic

in reality. Thus, the improvement of cluster performance is relatively small in Fig. 9 (a) between the AKPD and KPD algorithms compared with the AKPD and MCD algorithms, as the value of S is fairly large for the AKPD and KPD. However, the improvement of performance is obvious for the SVM-AKPD in Fig. 9 (b), therefore the F measure is a robust quality measure to validate the quality of clustering results if the ground-truth partition of data is available.



(a)



(b)

Fig. 10. The impact of cluster ASA based on simulated channels. (a) Silhouette coefficient, (b) F measure.

Then we analyze the impact of cluster angular spread on the clustering accuracy. In the simulated channels, we set different cluster ASA in the range of $\{0.5°, 1°, \ldots, 20°\}$ to the simulated channels. The value of Silhouette coefficient and F measure is calculated by the means of 21 channels for each cluster ASA. As we can see in Figs. 10 (a) and (b), with the increase of cluster ASA, Silhouette coefficient and F measure have a downward trend of the different algorithms, and the cluster performance of the SVM-AKPD is obviously better than that of others. Moreover, for the different cluster ASA, the performance of the SVM-AKPD algorithm is better and the SVM is insensitive to the distribution variation of intra-cluster MPCs, which benefits from the fact that the cluster of the SVM-AKPD is based on the full partition of feature space. The partition of feature space is stable with different cluster ASA as long as the cluster center (key MPCs) is accurately found, so the SVM-AKPD algorithm can adapt to different "cluster conditions". In addition, the performance gap between the AKPD and KPD algorithms is little when the cluster ASA is small, in the case of that the MPCs is more concentrated, which is very rare in the real channel.

VI. Conclusion

The accurate and effective clustering algorithm is challenging, due to the increasing number and parameter dimensions of the resolvable MPCs in the massive MIMO channels. In this paper, a novel AKPD and SVM-AKPD algorithms are proposed for MPC automatic clustering. The key features are: 1) a distanced-based metric is used to calculated *adaptive-K* for each MPC, and *adaptive-K* nearest MPCs are considered to estimated density and relative density, which enables better identifying of the local density variations of MPCs, especially when the number of MPC inside a cluster is smaller than K; 2) the SVM is innovatively applied in the channel clustering, and the SVM-AKPD can find the hyperplane to make the feature space of different clusters best separable, which makes the algorithm more robust and insensitive to the distribution variation of intra-cluster MPCs. 3) the performance of algorithms is validated based measured channels in 5G millimeter wave band, in which the proposed algorithms outperform the other algorithms in a real-word environment. 4) the proposed algorithms have good performance in the different "cluster conditions" with the simulated channels where different cluster ASA is set to generate synthetic MPCs. Both numerical simulations and experimental measurements demonstrated the effectiveness and robustness of the proposed algorithms for practical channel measurement campaigns, and the results in this paper are useful for the cluster based channel modeling of 5G and beyond 5G communications.

References

[1] X. Zhao, et al., "Playback of 5G and Beyond Measured MIMO Channels by an ANN-Based Modeling and Simulation Framework," in *IEEE J. Sel. Areas Commun.*, vol. 38, no. 9, pp. 1945-1954, Sept. 2020.

[2] R. He, et al., "Wireless Channel Sparsity: Measurement, Analysis, and Exploitation in Estimation," in *IEEE Wireless Commun.*, doi: 10.1109/MWC.001.2000378.

[3] W. Fan, et al., "Over-the-Air Radiated Testing of Millimeter-Wave Beam-Steerable Devices in a Cost-Effective Measurement Setup," in *IEEE Commun. Mag.*, vol. 56, no. 7, pp. 64–71, Jul. 2018.

[4] C. Gustafson, et al., "On mm-Wave Multipath Clustering and Channel Modeling," in *IEEE Trans. Antennas Propag.*, vol. 62, no. 3, pp. 1445-1455, Mar. 2014.

[5] A. F. Molisch and F. Tufvesson, "Propagation channel models for next generation wireless communications systems," in *IEICE Trans. Commun.*, vol. E97-B, no. 10, pp. 2022–2034, 2014.

[6] K. H. Li, M. A. Ingram and A. Van Nguyen, "Impact of clustering in statistical indoor propagation models on link capacity," in *IEEE Trans. Commun.*, vol. 50, no. 4, pp. 521-523, Apr. 2002.

[7] M. Bengtsson and B. Volcker, "On the estimation of azimuth distributions and azimuth spectra," in *Proc. IEEE VTC*, Oct. 2001, pp. 1612–1615.

[8] R. He et al., "Clustering Enabled Wireless Channel Modeling Using Big Data Algorithms," in *IEEE Commun Mag.* , vol. 56, no. 5, pp. 177-183, May 2018.

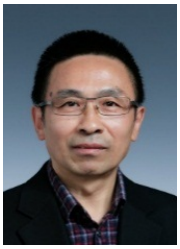
[9] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. BSMSP*, 1967, pp. 281–297.

[10] N. Czink et al., "A framework for automatic clustering of parametric MIMO channel data including path powers," in *Proc. IEEE VTC*, pp. 1–5, Sep. 2006.

[11] C. Schneider, et al., "Clustering of MIMO channel parameters — performance comparison," in *Proc. IEEE Veh. Technol. Conf.*, Anchorage, AK, USA, pp. 1–5, Sep. 2009.

[12] S. Mota, F. Perez-Fontan and A. Rocha, "Estimation of the Number of Clusters in Multipath Radio Channel Data Sets," in *IEEE Trans. Antennas Propag.*, vol. 61, no. 5, pp. 2879-2883, May 2013.
[13] C. Schneider, et al., "On the reliability of multipath cluster estimation in realistic channel data sets," in *Proc. EuCAP*, pp. 449-453, 2014.
[14] C. Schneider, et al., "Estimation and characterization of multipath clusters in urban scenarios," in *Proc. EuCAP*, pp. 1-5, 2018.
[15] M. Gan, et al., "Cluster life-time characterization for vehicular communication channels," in *Proc. EuCAP*, pp. 1–5, 2015.
[16] M. Ester et al., "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. ACM KDD*, pp. 226–231, 1996.
[17] X. Zhao, et al., "Channel Measurements, Modeling, Simulation and Validation at 32 GHz in outdoor Microcells for 5G Radio Systems," in *IEEE Access*, vol. 5, pp. 1062-1072, 2017.
[18] R. He, et al., "A Kernel-Power-Density-Based Algorithm for Channel Multipath Components Clustering," in *IEEE Trans. Wireless Commun.*, vol. 16, no. 11, pp. 7138-7151, Nov. 2017.
[19] F. Du, et al., "An Improved KPD Algorithm of Multipath Components Clustering for 5G Millimeter Wave Radio Channels," in *2018 12th ISAPE*, Hangzhou, China, 2018, pp. 1-4.
[20] S. Jaeckel, et al., "QuaDRiGa: A 3-D multi-cell channel model with time evolution for enabling virtual field trials," in *IEEE Trans. Antennas Propag.*, vol. 62, no. 6, pp. 3242–3256, Jun. 2014.
[21] M. Steinbauer, A. F. Molisch, and E. Bonek, "The double-directional radio channel," in *IEEE Antennas Propag. Mag.*, vol. 43, no. 4, pp. 51–63, Aug. 2001.
[22] B. H. Fleury, et al., "Channel parameter estimation in mobile radio environments using the SAGE algorithm," in *IEEE J. Sel. Areas Commun.*, vol. 17, no. 3, pp. 434–450, Mar. 1999.
[23] A. Richter, "Estimation of radio channel parameters: models and algorithms," Ph.D. dissertation, Dept. Elect. Eng. Inf. Technol., TU Iimenau, Dec. 2005.
[24] F. Zhang and W. Fan, "Near-Field Ultra-Wideband mmWave Channel Characterization Using Successive Cancellation Beamspace UCA Algorithm," in *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 7248-7259, Aug. 2019.
[25] Y. Ji, W. Fan and G. F. Pedersen, "Channel Characterization for Wideband Large-Scale Antenna Systems Based on a Low-Complexity Maximum Likelihood Estimator," in *IEEE Trans. Wireless Commun.*, vol. 17, no. 9, pp. 6018-6028, Sept. 2018.
[26] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," in Science, vol. 344, no. 6191, pp. 1492–1496, Jun. 2014.
[27] J. Bondy and U. Murty, Graph Theory (Graduate Texts in Mathematics). New York, NY, USA: Springer, 2008.
[28] M. Steinbauer et al., "How to quantify multipath separation," in *IEICE Trans. Electron.*, vol. E85-C, no. 3, pp. 552–557, Mar. 2002.
[29] X. Yin, C. Ling, and M.-D. Kim, "Experimental multipath-cluster characteristics of 28-GHz propagation channel," in *IEEE Access*, vol. 3, pp. 3138–3150, 2015.
[30] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc*. BSMSP, 1967, pp. 281–297.
[31] W. Fan, et al., "On Angular Sampling Intervals for Reconstructing Wideband Channel Spatial Profiles in Directional Scanning Measurements," in *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 13910-13915, Nov. 2020.
[32] W. Fan, et al., (2016). Measured Wideband Characteristics of Indoor Channels at Centimetric and Millimetric Bands. in *EURASIP J. Wirel. Comm.*, 2016(58).
[33] V. Vapnik, The Natural of Statistical Learning Theory. Springer Verlag, Heidelberg, DE. 1995.
[34] J. Lee and J. Lee, "K-means clustering based SVM ensemble methods for imbalanced data problem," SCIS&ISIS, pp. 614-617, Dec. 2014.
[35] E. Amigó et al., "A comparison of extrinsic clustering evaluation metrics based on formal constraints," in Inf. Retr., vol. 12, no. 4, pp. 461–486, 2009.
[36] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," in *J. Comput. Appl. Math.*, vol. 20, no. 1, pp. 53–65, 1987.
[37] L. Kaufman and P. J. Rousseeuw, Finding Groups in Data: An Introduction to Cluster Analysis. Hoboken, NJ, USA: Wiley, 2009
[38] M. Steinbach et al., "A comparison of document clustering techniques," in *Proc. KDD Workshop Text Mining*, 2000, pp. 525–526.
[39] J. Han, M. Kamber, and J. Pei, Data Mining: Concepts and Techniques. Amsterdam, The Netherlands: Elsevier, 2011.
[40] C. Hou, et al., "Researching on the Deterministic Channel Models for Urban Microcells Considering Diffraction Effects." in *Energies* vol. 14, no. 8, pp. 2143, 2021.
[41] X. Zhao et al., "Channel Sounding, Modelling, and Characterization in a Large Waiting Hall of a High-Speed Railway Station at 28 GHz," in *IET Microw., Antennas Propag.*, vol. 13, no. 15, pp. 2619-2624,Dec. 2019.
[42] P. Kyösti et al., "IST-4-027756 WINNER II D1.1.2 v.1.1: WINNER II Channel Models," Tech. Rep., 2007 [Online]. Available: http://www.ist-winner.org.
[43] Study on channel model for frequencies from 0.5 to 100 GHz, 3GPP TR 38.901, V15.0.0, Jun, 2018.

Fei Du received the B.Sc. degree with electronic information engineering from North China Electric Power University (NCEPU), Beijing, China, in 2016, and currently he is working in PhD in Communications Engineering. His recent research interests include 3D-MIMO channel modeling, mmWave channel modeling, high-resolution parameter extraction algorithm, time-varying channels. Email: dufei@ncepu.edu.cn



Xiongwen Zhao (Senior Member, IEEE) received the Ph.D. degree (Hons.) from the Helsinki University of Technology (TKK), Finland, in 2002. From 1992 to 1998, he was with the Laboratory of Communications System Engineering, China Research Institute of Radiowave Propagation, where he was a Director and a Senior Engineer. From 1999 to 2004, he was with the Radio Laboratory, TKK, as a Senior Researcher and the Project Manager in the areas of MIMO channel modeling and measurements at 2, 5, and 60 GHz as well as UWB. From 2004 to 2011, he was with Elektrobit Corporation, Espoo, Finland, as a Senior Specialist at EB Wireless Solutions. From 2004 to 2007, he worked in the European WINNER Project as a Senior Researcher in MIMO channel modeling for 4G radio systems. From 2006 to 2008, he also worked in the fifield of wireless network technologies such as WiMAX and wireless mesh networks. From 2008 to 2009, he worked in mobile satellite communications for GMR-1 3G, DVB-SH RF link budget, and antenna performance evaluations.

He is currently a Chair Professor of information and communications engineering with North China Electric Power University, Beijing, China, and responsible for the projects supported by the National Science Foundation of China, the Double First Class construction project by Ministry of Education, the Key Program of Beijing Municipal Natural Science Foundation, Beijing Municipal Science and Technology Commission, and the State Key Laboratories and Industries on 5G wireless channel and wireless power communications research. He has more than 190 publications, in which 75 articles are indexed by SCI. He is a Fellow of Chinese Institute of Electronics (CIE). He was a recipient of the IEEE Vehicular Technology Society Neal Shepherd Memorial Best Propagation Paper Award, IEEE ISAPE and IWCMC best paper awards in 2014, 2018, and 2019, respectively. He served as a TPC Chair, a Session Chair, and a Keynote Speaker for numerous international and national conferences. He is an Associate Editor of IET Communications. Email: zhaoxw@ncepu.edu.cn

Yu Zhang received her B.Sc. (2016) degree at North china electric Power University and currently she is working in PhD. in Communications Engineering. Her main field of interest is 5G networks, millimeter wave communications, MIMO systems and hybrid precoding. Email: zhyu16@163.com

Yang Wen received the B.Sc. degree from North China Electric Power University in 2017, where she is currently pursuing the M.Sc. degree in communications engineering. Her main research interests are mmWave channel modeling, 5G networks, and grey theory algorithms.

Zihao Fu received his B.Sc. (2018) degree at North China electric Power University. and currently he is working toward his MSc. in Communications Engineering. His research interests include 3D-MIMO channel modeling, mmWave time-varying channel modeling, and high-resolution parameter extraction algorithms. Email: 18810555907@163.com

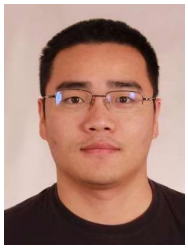
Suiyan Geng received the MSc (Tech.) and PhD in 2003 and 2011, respectively, from the Helsinki University of Technology (TKK), Espoo, Finland. From 1992 to 1998, she was a Research Engineer with the China Research Institute of Radiowave Propagation, Xinxiang, China. From 2001 to 2011, she was a Research Engineer with the Radio Laboratory (Department of Radio Science and Engineering since the beginning of 2008), TKK. She is now an Associate Professor at North China Electric Power University. He was a recipient of IEEE Vehicular Technology Society (VTS) Neal Shepherd Memorial Best Propagation Paper Award in 2014. Her research topics include millimeter-wave and ultra-wideband radio wave propagation and stochastic channel modeling for future-generation radio systems and technologies. Email: gsuiyan@ncepu.edu.cn

Peng Qin received the B.S. and Ph.D. degree from Huazhong University of Science and Technology, Wuhan, China, in 2009 and 2014, respectively. From 2012 to 2013, he was a Visiting Scholar with the University of Victoria, Victoria, BC, Canada. He is currently an Associate Professor with the Department of Electrical and Electronic Engineering, North China Electric Power University.His research interests include spacial terrestrial integrated network, wireless network, SDN networking and network optimization. Email: qinpeng@ncepu.edu.cn

Zhenyu Zhou (M'11–SM'17) received the M.E. and Ph.D degrees from Waseda University, Tokyo, Japan, in 2008 and 2011, respectively. From April 2012 to March 2013, he was the Chief Researcher with the Department of Technology, KDDI, Tokyo, Japan. From March 2013 to now, he is a Professor with the School of Electrical and Electronic Engineering, North China Electric Power University, Beijing, China. His research interests include green communications, vehicular communications, and smart grid communications.Dr. Zhou served as an Associate Editor for IEEE Access, EURASIP Journal on Wireless Communications and Networking, and a Guest Editor for IEEE Communications Magazine and Transactions on Emerging Telecommuni- cations Technologies. He also served as workshop Co-Chair for IEEE Globecom 2018, IEEE ISADS 2015, and TPC member for IEEE Globecom, IEEE ICC, IEEE VTC etc. He is a voting member of IEEE Standard Association P1932.1 Working Group. He was the recipient of the IEEE Vehicular Technology Society "Young Researcher Encouragement Award" in 2009, the "Beijing Outstanding Young Talent" Award in 2016, the IET Premium Award in 2017, and the IEEE ComSoc Green Communications and Computing Technical Committee Best Paper Award in 2017 and 2018. Email: zhenyu_zhou@ncepu.edu.cn

Chen Xu (S'12-M'15) received the B.S. degree from Beijing University of Posts and Telecommunications in 2010, and the Ph.D. degree from Peking University, Beijing, in 2015. She is now a lecturer in School of Electrical and Electronic Engineering, North China Electric Power University, China. Her research interests mainly include wireless resource management, heterogeneous networks, and cooperative communication and computing. She served as a TPC member for IEEE Globecom 2016, IEEE ICC 2016, etc. She received the best paper award at the 2012 International Conference on Wireless Communications and Signal Processing (WCSP), IEEE Leonard G. Abraham Prize in 2016, WCSP 10-year Anniversary Excellent Paper Award in 2019, and the first prize of Natural Science of Chinas Ministry of Education in 2017. She is one of 2019 Beijing Nova of Science and Technology.

Yongsheng Liu is the engineer of China Research Institute of Radiowave Propagation, His research interests include wireless channel measurement and channel modeling. Email: liuys_22s@qq.com



Wei Fan received his Bachelor of Engineering degree from Harbin Institute of technology, China in 2009, Master's double degree with highest honours from Politecnico di Torino, Italy and Grenoble Institute of Technology, France in 2011, and Ph.D. degree from Aalborg University, Denmark in 2014. From February 2011 to August 2011, he was with Intel Mobile Communications, Denmark as a research intern. He conducted a three-month internship at Anite telecoms oy, Finland in 2014. He is currently an associate professor with Aalborg University Denmark and heads the "wireless channel and over-the-air testing" research group. His main areas of research are over the air testing of multiple antenna systems, radio channel sounding, modeling and emulation.