Received January 24, 2020, accepted February 6, 2020, date of publication February 10, 2020, date of current version February 24, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2972975

Using Granule to Search Privacy Preserving Voice in Home IoT Systems

WEI LI[1], **YUMIN CHEN**[1], **HUOSHENG HU**[2], **(Senior Member, IEEE), AND CHAO TANG**[3]

[1]School of Computer and Information Engineering, Xiamen University of Technology, Xiamen 361024, China
[2]School of Computer Science and Electronic Engineering, University of Essex, Colchester CO4 3SQ, U.K.
[3]Department of Computer Science and Technology, Hefei University, Hefei 230601, China

Corresponding author: Wei Li (drweili@hotmail.com)

This work was supported in part by the Science and Technology Planning Guidance Project of Xiamen of China under Grant 3502Z20179038, in part by the Scholarship for Studying Abroad Program of Fujian of China, in part by the National Natural Science Foundation of China under Grant 61976183 and Grant 61573297, in part by the Natural Science Foundation of Fujian Province of China under Grant 2019J01850, in part by the Excellent Talents Training Funded Project of Universities of Anhui Province of China under Grant gxfx2017099, and in part by the Key Teaching and Research Project of Heifei University of China under Grant 2018 hfjyxm09.

ABSTRACT The Home IoT Voice System (HIVS) such as Amazon Alexa or Apple Siri can provide voice-based interfaces for people to conduct the search tasks using their voice. However, how to protect privacy is a big challenge. This paper proposes a novel personalized search scheme of encrypting voice with privacy-preserving by the granule computing technique. Firstly, Mel-Frequency Cepstrum Coefficients (MFCC) are used to extract voice features. These features are obfuscated by obfuscation function to protect them from being disclosed the server. Secondly, a series of definitions are presented, including fuzzy granule, fuzzy granule vector, ciphertext granule, operators and metrics. Thirdly, the AES method is used to encrypt voices. A scheme of searchable encrypted voice is designed by creating the fuzzy granule of obfuscation features of voices and the ciphertext granule of the voice. The experiments are conducted on corpus including English, Chinese and Arabic. The results show the feasibility and good performance of the proposed scheme.

INDEX TERMS Fuzzy search, granule computing, k-nearest neighbor, searchable encrypted voice, obfuscation function.

I. INTRODUCTION

Voice activation devices, such as Amazon Alexa, Apple Siri, Google Assistant or Microsoft Cortana were widely used on over 2 billion smartphones in 2018. Moreover, as the demand for smart home devices continues to grow, sound interaction devices such as Amazon Echo, Apple Home-Pod, or Google Home are also widely deployed. When people enjoy using these devices, personal privacy may be revealed if the data is stored in the cloud server with the plaintext. Therefore, data owners tend to encrypt the data and then outsource the ciphertext to the cloud server. However, with the proliferation of data volume and number of users, cloud servers may become the performance bottleneck of cloud services. This results in the long waiting time and seriously affects the user's search experience. Hence, how to quickly obtain the search results in the vast ciphertext is a challenge for using the personalized search technology.

A. CIPHERTEXT SEARCH SCHEMES

The existing ciphertext search schemes can be classified into searchable symmetric encryption (SSE) framework and public key encryption with keyword search (PKEKS) framework. According to technical details and its inherent nature, the SSE scheme is further divided into a sequential scan scheme and a secure index scheme. Song et al. first proposed the SSE scheme based on sequential scanning in [1] by splitting the plaintext into ''words'' and then encrypting them. When the user submits a search request, the ciphertext file containing the keyword is returned by sequentially scanning and comparing the ciphertext word with the keyword to be retrieved. It was able to support searching for any word in a file. However, its efficiency was extremely low as the server had to traverse the entire file during the search. Moreover, the scheme cannot resist the frequency analysis attack on ciphertext.

Goh [2] proposed an improved SSE scheme based on secure forward index. The secure index of each file was matched to keywords by the server and the user's keyword

The associate editor coordinating the review of this manuscript and approving it for publication was Xiaochun Cheng.

search can be supported with a high efficiency. However, the search results were not completely correct due to the positive mis-detection probability of the Bloom Filter in the building index. This may bring some additional overhead of bandwidth and computation to users. The scheme security can reach the indistinguishability against chosen keyword attack. In contrast, the scheme presented by Chang et al. [3] can avoid the positive mis-detection probability of Goh's scheme. Furthermore, by adopting the inverted index construction method, its ability to anti-selective keyword attack was stronger than Goh's scheme. It can resist the adaptive selection keyword attack.

Curtmola et al. [4] further improved and clearly defined the security of SSE scheme. On the one hand, they proposed SSE-1 and SSE-2 solutions to achieve indistinguishable security under adaptive and non-adaptive models. On the other hand, PEKS. Boneh et al. [5] proposed the PEKS scheme and presented several construction schemes based on bilinear pairings. Abdalla et al. [6] further gave the complete definition of the PEKS scheme, and presented the process of constructing PEKS based on the identity anonymity scheme. In [7]–[9], researchers designed PEKS schemes that don't require a secure channel under the random language model and the standard model.

Subsequently, some improved searchable encryption schemes were proposed for various scenarios that promoted the development of searchable encryption technology [10]–[16], [20]. Zhao et al. [17] combined content filtering and collaborative filtering to provide users with personalized search results. Experimental results showed that the method can provide accurate search results and improve the user's search experience. Leung et al. [18] obtained the user's interest preference by mining the user's click data, and introduced the user's location information, and adopted entropy to balance the weight between the user's preference and the location information. This method improved the search accuracy and promoted the user's search experience.

However, it is still a challenging task to achieve a personalized search in a ciphertext environment and improve the user's search experience. Fu et al. [19] constructed user models based on the user's search history and integrated users' interests into the user's query keywords through keyword priority according to the word net. Then they searched the ciphertext stored on the cloud server and got the top K search results with the highest relevance score of the user to achieve personalized search in the ciphertext environment. These searchable encrypted schemes proposed above are from the perspective of numerical calculation.

In summary, the schemes mentioned above did not pay much attention to the hierarchy of data. To improve the performance, we will design a new scheme based on the hierarchy of data from the granular computing viewpoint.

B. GRANULAR COMPUTING

Information granule is an information that is ubiquitous around us. It is a basic concept of human to know the world. Humans tend to put a part of similar things together as a whole in understanding the world to study their nature or characteristics. In fact, this way of dealing with things is information granulation and the study of the ''whole'' is called information granule. In granular computing, the information granule is used as the basic operation unit instead of the sample, and the exact solution is replaced by the approximation solution, which can achieve the purpose of designing high performance algorithm.

As a methodology, granular computing aims to effectively establish an external world-based, user-centric concept that simplifies the understanding of the physical world and the virtual world. In the process of solving the problem, the ''granule'' with the appropriate level of granularity is used as the processing object, so as to improve the efficiency of solving the problem under the premise of ensuring satisfactory solution. Since Zadeh published the first paper on information granularity in 1979, researchers have made in-depth research on granular computing theory and models, and combined them with computational intelligence and machine learning techniques. A lot of research results have been achieved.

The appropriate granularity is often determined by the problem itself and its context, which is important for designing data processing framework based on granular computing. For example, someone asked his or her friend, ''When did you return home in China?''. The time granularity chosen to answer this question is actually determined by how long his or her friend has been back to China. If it was not more than one day, then the answer could be ''Yesterday afternoon''. If it was more than one week, the answer can be ''Last week''. Note that the above answers have different granularities, namely afternoon and week. If you do not use the appropriate granularity but the unified time stamp format to answer, such as: ''at 1:00 am yesterday'', it might make people feel awkward.

As early as 1979, a famous American cybernetic expert, Zadeh [21] firstly presented the problem of fuzzy information granulation. He believed that human cognition can be summarized into three main characteristics: granulation, organization, and causation. In 1985, Hobbs [22] proposed the concept of granularity. In the early 1990s, Zhang et al. [23] pointed out that ''a recognized characteristic of human intelligence is that people can observe and analyze the same problem from very different granularities in their monograph ''Question Theory and Application''. People can not only solve problems in different granular worlds, but also quickly jump from one granular world to another, freely and easily, without difficulty.'' This ability can deal with different granular space and is a powerful manifestation of human problem solving.

Yager and Filev [24] further pointed out that ''people have formed a granular view of the world, in which human observation, measurement, conceptualization and reasoning are carried out.'' These views all believe that granulation, as one of the important characteristics of human cognition, plays an important role in the knowledge discovery of complex data. The concept of granular computing was first proposed in 1997, Zadeh [25] and the principles were identified by Pedrycz [26]. Pedrycz showed how information granules were constructed and subsequently used in describing relationships among data items. Later, many scholars in different fields worldwide began to pay attention to this problem, which gradually formed a new research direction in intelligent information processing.

In addition, granular computing has promoted the development of many concepts, such as diagrams [51], information tables [52], knowledge representations [53] and so on. Granular computing is also widely used in time series forecasting [54], manufacturing [55], mission forecasting [56] and information fusion [58].

C. RESEARCH PROGRESS

1) DATA GRANULATION RESEARCH

Data granulation is the process of decomposing complex data into information granules according to a given granulation strategy. According to different data modeling goals and user needs, a variety of granulation strategies can be adopted. Most of the common granulation strategies relying solely on data can be attributed to a granulation scheme based on data binary relations, which essentially distributes two data samples that satisfy a predefined binary relationship into the same granule. In many granulation strategies, data can be granulated into corresponding binary structure by using equivalence relations, similarity relations, maximal similarity relations, fuzzy equivalence relations, fuzzy similarity relations, neighborhood relations, and dominant relations [27]–[34]. The current data granulation strategies and methods are mostly based on single modal characteristics, setting weight parameters between different modal features or simply integrating results, which can not effectively solve the problem of data co-granulation with multi-modal features.

2) MULTI-GRANULARITY PATTERN DISCOVERY AND FUSION

Multi-granularity pattern discovery and fusion are the inherent logic requirements for solving complex problems under the granular computing framework. The so-called multi-granularity includes multiple data subsets, multiple subspaces representing a space, multiple different modal variable sets, multiple local or intermediate results in a problem solving process. They correspond to multiple problems angle and multiple local or multiple levels. In order to obtain a global solution to the overall data set or problem, it is necessary to fuse multiple patterns found on a single granularity. Although the term multi-granularity has not widely been used, scholars have conducted research on multi-modality in the fields of medical image analysis, network, video semantic analysis, annotation and retrieval, emotion recognition, and mainly consider data from different modalities. In these situations, the features are extracted separately to form a multi-modal feature space to develop the method of pattern discovery with multimodal features. The current research focuses on three aspects: multimodal data classification based on multicore learning [35], multimodal data modeling based on multidictionary collaborative expression [36] and multimodal data fusion based on deep learning [37].

3) GRANULAR COMPUTING REASONING

Reasoning is one of the important abilities in human intelligence. It is a formal logic, a science used to study people's forms of thinking, laws, and logical methods. The role of reasoning is to obtain unknown knowledge from known knowledge. The reasoning of Granular Computing refers to the logical method of deducting using known information granules or granule spaces. In the field of Granular Computing, there have been some studies on reasoning [21], [38]–[43].

4) HIGH PERFORMANCE ALGORITHMS

In recent years, there have been some preliminary explorations on the use of granular computing to solve big data problems. Ye et al. [44] achieved the clustering analysis of large-scale data by granulating the data space and feature space using integrated learning technology. Chang et al. [45] proposed a big data decomposition method using decision trees, and then separately learned the Support-vectors Machine classifier on each decomposed data granule, which greatly improved the learning efficiency of Support-vectors Machine. Gopal et al. [46] employed the hierarchical relationship between data categories and gave a corresponding Bayesian model to increase its generalization performance. Miao et al. [47] proposed a property reduction method that can be computed in parallel by adopting the data decomposition principle in MapReduce. By splitting the original big data set into multiple easy-to-process information granules,

Liang et al. [48] proposed an efficient big data feature selection algorithm by solving and merging the feature selection results on each information granule. Qian et al. [49] employed the information granularity to construct the forward approximation of the rough set and proposed the feature selection accelerator to accelerate a series of feature selection algorithms of forward greedy search. Chen et al. [50] pointed out that different information granules imply different characteristics and patterns, which can be used to design machine learning and data mining algorithms effectively. The challenges were mainly reflected in two aspects: Firstly how to rationalize the information granulation and ensure the effective solution; Secondly how to efficiently obtain an approximate solution by balancing the algorithm efficiency and the solution accuracy.

This paper proposes a novel scheme for searchable symmetric encrypted-voice from the new perspective of granular computing. The rest of this paper is organized as follows. Section II presents the construction of system model for voice retrievals. In Section III, we will discuss how to extract the voice feature, transform the raw data into information

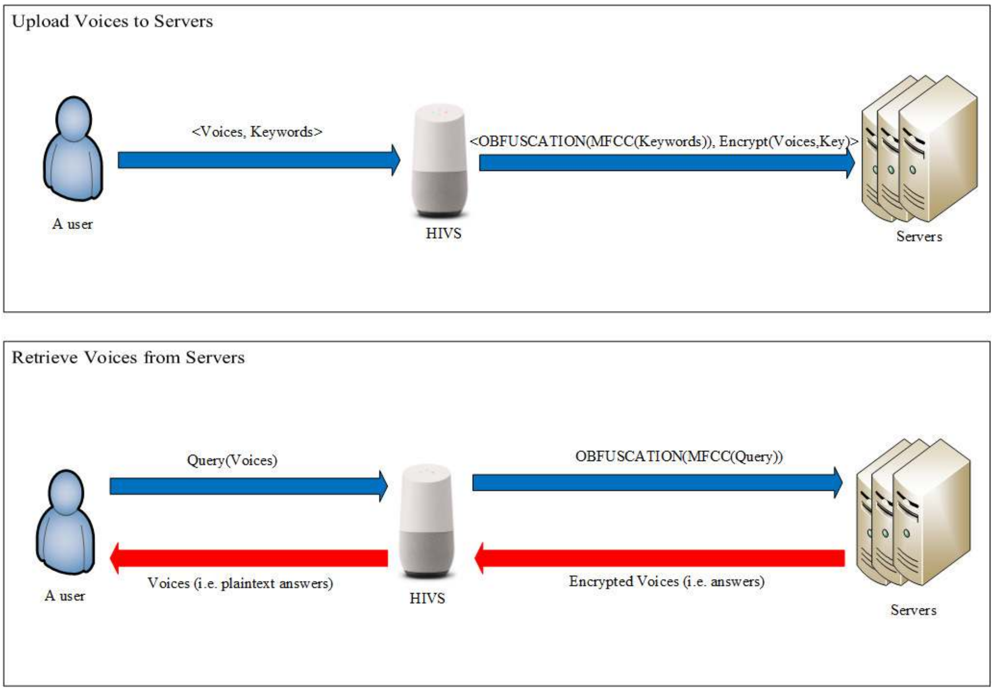

Upload Voices to Servers
<Voices, Keywords>
<OBFUSCATION(MFCC(Keywords)), Encrypt(Voices,Key)>
A user
HIVS
Servers
Retrieve Voices from Servers
Query(Voices)
OBFUSCATION(MFCC(Query))
Voices (i.e. plaintext answers)
Encrypted Voices (i.e. answers)
A user
HIVS
Servers

FIGURE 1. **Overview of system model.**

granule, encrypt data, and search over encrypted data via granule computing. The system evaluation is given in Section IV to demonstrate the feasibility and performance of the proposed approach. Finally, a brief conclusion and future work are described in Section V.

II. SYSTEM MODEL

A. OVERVIEW

The system model has three types of entities: user, HIVS and servers. It is composed of two phases, namely voice uploading phase and voice retrieving phase. During the uploading phase, users upload voice and the keywords to HIVS; the voices are encrypted and the features of voices are extracted and obfuscated by HIVS. Then the features and encrypted voice are submitted to the server for storage. During the retrieving phase, users send the voice query to HIVS; the features of query voice are extracted, obfuscated and uploaded by HIVS to the server; the features matching is done by the server using the scheme proposed in this paper. The answers (encrypted voices) are returned to HIVS. Then, these answers are decrypted by HIVS and sent to the user (See Fig. 1).

B. SCHEME CONSTRUCTION

Our solution consists of two parts: (1) voice pre-processing and uploading server; (2) retrieving data using voice commands (See Fig. 1).

1) UPLOADING VOICE

In the voice uploading phase, the data structure is made up of two parts: objects and keywords, which are uniquely stored on the server. The object is the data that the user wants to store on the server. The keyword represents a category or an attribute of the object. More specifically, the objects are stored as encrypted form on the server. The keywords are saved as the form of features on the server. The relationship between object and keyword can be many-to-many, i.e., one keyword can be associated with multiple objects, or one object can be associated with multiple keywords. For example, ''What holiday is today? holiday, today'', the first element ''What holiday is today? '' is used as an query, and the second and the third element ''holiday, today'' is as a keyword for query. If the server receives another voice for ''What is the holiday today?, New Year'', it will add the new keyword ''New Year'' to the query ''What is the holiday today?''. During the uploading process, the object is encrypted into a ciphertext by AES. The keyword is extracted features by MFCC and then these features are obfuscated. Thus, the obfuscation features and the ciphertext are transmitted to the server for privacy protection.

2) RETRIEVING VOICE

In the voice retrieval stage, when a user sends query command to a server to seek an answer, the query command is firstly sent to HIVS including a keyword. After the feature extraction and obfuscation are performed by HIVS, then the feature is sent to the server. The k-nearest neighbors ciphertext granule search (KNNCGS) algorithm proposed in the paper is adopted. The encrypted answer is returned to HIVS. Then it is decrypted by HIVS through AES algorithm, and the plaintext is sent to the user. In the ciphertext retrieval process, the returned ciphertext may be multiple related answers, and the number of answer can be set by the user to improve performance.

III. SCHEME IMPLEMENTATION

In this section, we will discuss how to extract the voice feature, transform the raw data into information granule, encrypt data, and search over encrypted data via granule computing.

A. EXTRACTING VOICE FEATURE

Mel-Frequency Cipstal Coefficients (MFCC) is a set of key coefficients used to establish the Mel Cepstrum. From the segments in the voice signal, we can get a set of cepstrums that are sufficient to represent this voice signal. The Mel-Frequency Cepstral Coefficient is the cepstrum (namely the spectrum of the spectrum) derived from this cepstrum. Unlike the general cepstrum, the frequency band on the Melt's cepstrum is evenly distributed on the Mel scale. That is, such a frequency band will be closer to the human nonlinear auditory system. MFCC is a distinguishable feature in speech signal processing.

Let v be voice. The m-order of the i^{th} frame can be represented as $\{v_{i1}, v_{i2}, \ldots, v_{im}\}$. After extracting feature, the signal of frames form a matrix below:

$$V = \begin{bmatrix} v_{11} & v_{12} & \cdots & v_{1m} \\ v_{21} & v_{22} & \cdots & v_{2m} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ v_{n1} & v_{n2} & \ldots & v_{nm} \end{bmatrix} \tag{1}$$

To reduce the complexity, we employ a vector to denote the signal, which is expressed by average value of frame of voice at MFCC below.

$$average(V) = (\frac{1}{n}\sum_{i=1}^{n} v_{i1}, \frac{1}{n}\sum_{i=1}^{n} v_{i2}, \ldots, \frac{1}{n}\sum_{i=1}^{n} v_{im}). \tag{2}$$

B. OBFUSCATION OF FEATURES

In this section, we designed an approach based on adding noise into feature of voice to match.That is, a reversible $(m + 3) \times (m + 3)$ confusion matrix A and three random numbers α, β and γ are introduced in order to hide the features and prevent these ones from being revealed to the server. In other words, voice feature is not directly uploaded to a server, but they are done operation with an obfuscation matrix before uploading, and then the result is uploaded to the server. Specifically, $A\vec{g_i}^T$ is uploaded to a server firstly, where

$$\vec{g_i} = (\sum_{j=1}^{m} a_{ij}^2 + \alpha - \beta, a_{i1}, \ldots, a_{im}, 1, \beta). \tag{3}$$

When we are searching, $\sum_{j=1}^{m}(a_{ij} - v_j)^2$ is used to measure the similarity between voice feature $v = (v_1, v_2, \ldots, v_m)$ and $a_i = (a_{i1}, a_{i2}, \ldots, a_{im})$, and this metric can be equivalent to calculate $\vec{f}\vec{g_i}^T$, where $\vec{f} = (1, -2v_1, \ldots, -2v_m, \gamma, 1)$. The proof of the approach is given as the follows.

Lemma 1: Given two voice features v and a_i, a reversible random matrix A and a series of random number α, β, γ, we let $g_i = (\sum_{j=1}^{m} a_{ij}^2 + \alpha - \beta, a_{i1}, \ldots, a_{im}, 1, \beta)$ and $\vec{f} = (1, -2v_1, \ldots, -2v_m, \gamma, 1).\vec{f}\vec{g_i}^T$ can be used as a metric of the similarity between v and a_i.

Proof: $\vec{f}A^{-1}A\vec{g_i}^T = \vec{f}\vec{g_i}^T = (\sum_{j=1}^{m} a_{ij}^2 + \alpha - \beta) - 2(v_1a_{i1}+\ldots+v_1a_{im})+\gamma+\beta = \sum_{j=1}^{m}(a_{ij} - v_j)^2 - \sum_{j=1}^{m} v_j^2 + \gamma + \alpha$ Therefore, we have that $\sum_{j=1}^{m}(a_{ij} - v_j)^2 = \vec{f}\vec{g_i}^T + \sum_{j=1}^{m} v_j^2 - \gamma - \alpha$.

Because $\sum_{j=1}^{m} v_j^2 - \gamma - \alpha$ is a constant, the server can adopt $\vec{f}\vec{g_i}^T$ as a metric of distance between f and a_i.

□

C. FROM RAW DATA TO FUZZY GRANULE

Fuzzy granulation is inspired by human granulation and information processing and is on the basis of mathematics. The promotion mode is divided into fuzzy and granular. Among them, fuzzification is to replace a clear set with a fuzzy set. Granulation is that a collection is divided into granules. Fuzzy granulation is composed of two phases: (1) The fuzzy granulation method is used to transform keyword into fuzzy granule. In this process, fuzzy granule, fuzzy granule vector and operators are defined to represent the feature. (2) The δ-neighborhood of fuzzy granule vector is employed to cluster the encrypted data (namely ciphertext granule). Some concepts such as δ-neighborhood ciphertext granule, ciphertext granule vector and related operators are defined to denote the encrypted data.

1) FUZZY GRANULATION

Definition 1: Let $SS = (P, E, R, V, K, \theta, \vec{f}, \vec{g})$ be a searchable system over ciphertext, where $P = \{p_1, p_2, \ldots, p_n\}$ is a plaintext set, $E = \{e_1, e_2, \ldots, e_n\}$ is a encryption set corresponding to P, $R = \{r_1, r_2, \ldots, r_m\}$ is a attribute set, $V = \cup_{r \in R} V_r$ is a set of the feature value. V_r is the range of feature value and it is satisfied to $V_r \in [0, 1]$. $\theta : P \times Rarrow V$ is an information function, which represents the feature value of each object p in P. $K = \{(key_1, A_1, \alpha_1, \beta_1, \gamma_1), (key_2, A_2, \alpha_2, \beta_2, \gamma_2), \ldots, (key_n, A_n, \alpha_n, \beta_n, \gamma_n)\}$ is a key set. Here, A is a 4×4 reversible confusion matrix. α, key, β and γ are random numbers. $\vec{f} = (1, -2V_r, \gamma, 1)$ and $\vec{g} = (V_r^2 + \alpha + \beta, V_r, 1, \beta)$ represent obfuscation vector on the feature r. The encrypted plaintext can be represented as $Encrypt(p_i, key_i) = e_i$ and the decrypted ciphertext can be expressed as $Decrypt(e_i, key_i) = p_i$.

Definition 2: Let $SS = (P, E, R, V, K, \theta, \vec{f}, \vec{g})$ be a searchable system over ciphertext. For $\forall p_i, p_j \in P$ and $\forall r \in R$, the distance on r between p_i and p_j is defined by:

$$d_r(p_i, p_j) = \vec{f}A^{-1}A\vec{g}^T = \vec{f}\vec{g}^T \tag{4}$$

where $d_r(p_i, p_j) \in [0, 1]$. According to Lemma 1, $d_r(p_i, p_j)$ can be metric between p_i and p_j on the feature r.

Definition 3: Let $SS = (P, E, R, V, K, \theta, \vec{f}, \vec{g})$ be a searchable system over ciphertext. For $\forall p \in P$ and $\forall r \in R$, fuzzy granule of the plaintext p on an atom feature r can be defined by:

$$N_r(p_i) = \{(p_1, d_{i1}), (p_2, d_{i2}), \ldots, (p_n, d_{in})\} \tag{5}$$

The former of the sequence pair is the plaintext, the latter of that is the distance between p_i and p_j on the feature r, in short, that is $d_{ij} = d_r(p_i, p_j)$.

Definition 4: Let $SS = (P, E, R, V, K, \theta, \vec{f}, \vec{g})$ be a searchable system over ciphertext. For $\forall p \in P$ and $\forall r \in R$, the module of fuzzy granule $N_r(p)$ can be defined by:

$$|N_r(p)| = \sum_{q \in P} d_r(p, q) \tag{6}$$

It is easy to get $1 \leq |N_r(p)| \leq |P|$, where $|P|$ denotes the number of elements in P.

Definition 5: Let $SS = (P, E, R, V, K, \theta, \vec{f}, \vec{g})$ be a searchable system over ciphertext. For $\forall p \in P$, any $Q \subseteq R$, and $Q = \{r_1, r_2, \ldots, r_k\}, (k \leq m)$, the fuzzy granule vector of p on feature subset Q can be defined by:

$$\hat{N}_Q(p) = (N_{r_1}(p), N_{r_2}(p), \ldots, N_{r_k}(p)) \tag{7}$$

Definition 6: Let $SS = (P, E, R, V, K, \theta, \vec{f}, \vec{g})$ be a searchable system over ciphertext. For $\forall p \in P$, any $Q \subseteq R$ and $Q = \{r_1, r_2, \ldots, r_k\}, (k \leq m)$, the module of fuzzy granule vector on p of feature subset Q can be defined by:

$$|\hat{N}_Q(p)| = \sum_{r \in Q} |N_r(p)| \tag{8}$$

Definition 7: Let $SS = (P, E, R, V, K, \theta, \vec{f}, \vec{g})$ be two fuzzy granules on the feature r, we define three operators $\cap, \cup, \oplus$ as follows:

$$\begin{aligned} N_r(p) \cap N_r(q) &= \{(p_1, d_{min,1}), (p_2, d_{min,2}), \\ &\quad \ldots, (p_n, d_{min,n})\} \quad (9)\\ N_r(p) \cup N_r(q) &= \{(p_1, d_{max,1}), (p_2, d_{max,2}), \\ &\quad \ldots, (p_n, d_{max,n})\} \quad (10)\\ N_r(p) \oplus N_r(q) &= \{(p_1, d_{max,1} - d_{min,1}), \\ &\quad (p_2, d_{max,2} - d_{min,2}), \ldots, \\ (p_n, d_{max,n} - d_{min,n})\} & \quad (11)\\ d_{min,i} &= min\{1, d_r(p_i, p) + d_r(p_i, q)\}, \\ d_{max,i} &= max\{0, d_r(p_i, p) + d_r(p_i, q) - 1\}, \\ d_{ij} &= d_r(p, p_j) \quad (12) \end{aligned}$$

Definition 8: Let $SS = (P, E, R, V, K, \theta, \vec{f}, \vec{g})$ be a searchable system over ciphertext. Here, $P = \{p_1, p_2, \ldots, p_n\}$ represents plaintext set, and $R = \{r_1, r_2, \ldots, r_m\}$ denotes feature set. For $\forall p, q \in P$, there exists two fuzzy granule vectors $\hat{N}_R(p) = (N_{r_1}(p), N_{r_2}(p), \ldots, N_{r_m}(p))$ and $\hat{N}_R(q) = (N_{r_1}(q), N_{r_2}(q),$
$\ldots, N_{r_m}(q))$ on R, we define three operators $\cup, \cap, \oplus$ as follows:

$$\begin{aligned} \hat{N}_R(p) \cap \hat{N}_R(q) &= (N_{r_1}(p) \cap N_{r_1}(q), N_{r_2}(p) \cap N_{r_2}(q), \\ &\quad \ldots, N_{r_m}(p) \cap N_{r_m}(q)) \quad (13)\\ \hat{N}_R(p) \cup \hat{N}_R(q) &= (N_{r_1}(p) \cup N_{r_1}(q), N_{r_2}(p) \cup N_{r_2}(q), \\ &\quad \ldots, N_{r_m}(p) \cup N_{r_m}(q)) \quad (14)\\ \hat{N}_R(p) \oplus \hat{N}_R(q) &= (N_{r_1}(p) \oplus N_{r_1}(q), N_{r_2}(p) \oplus N_{r_2}(q), \\ &\quad \ldots, N_{r_m}(p) \oplus N_{r_m}(q)) \quad (15) \end{aligned}$$

Definition 9: Let $SS = (P, E, R, V, K, \theta, \vec{f}, \vec{g})$ be a searchable system over ciphertext. Here, $P = \{p_1, p_2, \ldots, p_n\}$ is plaintext set, and $R = \{r_1, r_2, \ldots, r_m\}$ is feature set. For $\forall p, q \in P$, there exists the two fuzzy granule vectors $\hat{N}_R(p) = (N_{r_1}(p), N_{r_2}(p), \ldots, N_{r_m}(p))$ and $\hat{N}_R(q) = (N_{r_1}(q), N_{r_2}(q), \ldots, N_{r_m}(q))$ on R, their distance is defined by:

$$d(\hat{N}_R(p), \hat{N}_R(q)) = \frac{1}{|R| * |P|} \sum_{r \in R} \frac{|N_r(p) \oplus N_r(q)|}{|N_r(p) \cup N_r(q)|} \tag{16}$$

Theorem 1: For $\forall p, q \in P$, the two fuzzy granule vector satisfy:

$$0 \leq d(\hat{N}_R(p), \hat{N}_R(q)) \leq 1 \tag{17}$$

Proof: Assuming that $p = p_i, q = p_j$, according to definition 1-3, we have $N_r(p_i) = \{(p_1, d_{i1}), (p_2, d_{i2}), \ldots, (p_n, d_{in})\}$, $N_r(p_j) = \{(p_1, d_{j1}), (p_2, d_{j2}), \ldots, (p_n, d_{jn})\}$, $d_{ij} = d_r(p_i, q_j) \in [0, 1]$, $|N_r(p)| = \sum_{q \in P} d_r(p, q)$ $\hat{N}_R(p_i) = (N_{r_1}(p_i), N_{r_2}(p_i), \ldots, N_{r_m}(p_i))$, $\hat{N}_R(p_j) = (N_{r_1}(p_j), N_{r_2}(p_j), \ldots, N_{r_m}(p_j))$. According to equation (6)-(13), we also have that $\forall r \in R$, $0 \leq \frac{|N_r(p) \oplus N_r(q)|}{|N_r(p) \cup N_r(q)|} \leq |P|$, $0 \leq \sum_{r \in R} \frac{|N_r(p) \oplus N_r(q)|}{|N_r(p) \cup N_r(q)|} \leq |R| * |P|$, and $0 \leq \frac{1}{|R| * |P|} \sum_{r \in R} \frac{|N_r(p) \oplus N_r(q)|}{|N_r(p) \cup N_r(q)|} \leq 1$. Because of $d(\hat{N}_R(p), \hat{N}_R(q)) = \frac{1}{|R| * |P|} \sum_{r \in R} \frac{|N_r(p) \oplus N_r(q)|}{|N_r(p) \cup N_r(q)|}$, the equation $0 \leq d(\hat{N}_R(p), \hat{N}_R(q)) \leq 1$ is established. □

Theorem 2 (Monotony): Let $SS = (P, E, R, V, K, \theta, \vec{f}, \vec{g})$ be a searchable system over ciphertext. For $\forall p \in P$, feature subset $T \subseteq Q$, $\hat{N}_T(p)$, and $\hat{N}_Q(p)$ are two fuzzy granule vectors on p about T and Q respectively, then $|\hat{N}_T(p)| \leq |\hat{N}_Q(p)|$ is established.

Proof: According to the definition of fuzzy granule vector, we have that $\hat{N}_T(p) = (N_{r_1}(p), N_{r_2}(p), \ldots, N_{r_u}(p))$, and $\hat{N}_Q(p) = (N_{r_1}(p), N_{r_2}(p), \ldots, N_{r_v}(p))$. For $\forall r \in T$, the fuzzy granule on r is $N_r(p)$. Since $T \subseteq Q$, we have $r \in Q$. Hence, the fuzzy granule satisfies $N_r(p) \in \hat{N}_Q(p)$ and $|T| \leq |Q|$. Therefore, $\sum_{r \in T} |N_r(p)| \leq \sum_{r \in Q} |N_r(p)|$ is established. That is, $|\hat{N}_T(p)| \leq |\hat{N}_Q(p)|$ is established. □

2) CIPHERTEXT GRANULATION

We give some definitions of fuzzy granule, fuzzy granule vector, metrics and operators based on fuzzy set in the last section. In this section, on the basis of ciphertext, key and fuzzy granule of plaintext, we define ciphertext granule, ciphertext granule vector, operators and metrics to prepare the presentation of KNNCGS. As shown in the definition 10, we decrypt the ciphertext by the key to get the plaintext. On the basis of fuzzy granule of plaintext, we can define ciphertext granule of δ-neighborhood. After that, ciphertext granule vector, operators and metrics are also defined by ciphertext granule.

Definition 10: Let $SS = (P, E, R, V, K, \theta, \vec{f}, \vec{g})$ be a searchable system over ciphertext. For $\forall e \in E$ and $\forall r \in R$, the ciphertext granule of e on the feature r in δ-neighborhood ($\delta > 0$) can be defined by:

$$\begin{aligned} M_r^\delta(e) = \{u|u \in E, r \in R, p = Decrypt(e, key_e),\\ q = Decrypt(u, key_u), d(N_r(p), N_r(q)) \le \delta\} \end{aligned} \quad (18)$$

Definition 11: Let $SS = (P, E, R, V, K, \theta, \vec{f}, \vec{g})$ be a searchable system over ciphertext. The ciphertext granule vector of e on the feature set R in δ-neighborhood can be defined by:

$$\hat{M}_R^\delta(e) = (M_{r_1}^\delta(e), M_{r_2}^\delta(e), \ldots, M_{r_m}^\delta(e)) \quad (19)$$

Definition 12: Let $SS = (P, E, R, V, K, \theta, \vec{f}, \vec{g})$ be a searchable system over ciphertext. For $\forall e \in E$ and $\forall r \in R$, the cardinal number associated with ciphertext granule in δ-neighborhood $M_r^\delta(e)$ can be defined by $|M_r^\delta(e)|$, which denotes the number of elements. It is easy to get: $1 \le |M_r^\delta(e)| \le |E|$.

Definition 13: Let $SS = (P, E, R, V, K, \theta, \vec{f}, \vec{g})$ be a searchable system over ciphertext. For $\forall e \in E$ and any subset $Q \subseteq R$ (here, $Q = \{r_1, r_2, \ldots, r_k\}, (k \le m)$), the module of ciphertext granule vector of e on feature subset Q can be defined by:

$$|\hat{M}_Q^\delta(e)| = \sum_{r \in Q} |M_Q^\delta(e)| \quad (20)$$

Definition 14: Let $SS = (P, E, R, V, K, \theta, \vec{f}, \vec{g})$ be a searchable system over ciphertext. For $\forall e_i, e_j \in E$, $M_r^\delta(e_i)$ and $M_r^\delta(e_j)$ are ciphertext granules on $r \in R$ in δ-neighborhood. We define four operators, $\cap$, $\cup$, $-$ and $\oplus$ below:

$$M_r^\delta(e_i) \cap M_r^\delta(e_j) = \{e|e \in M_r^\delta(e_i) \; and \; e \in M_r^\delta(e_j)\} \quad (21)$$

$$M_r^\delta(e_i) \cup M_r^\delta(e_j) = \{e|e \in M_r^\delta(e_i) \; or \; e \in M_r^\delta(e_j)\} \quad (22)$$

$$M_r^\delta(e_i) - M_r^\delta(e_j) = \{e|e \in M_r^\delta(e_i) \; and \; e \notin M_r^\delta(e_j)\} \quad (23)$$

$$M_r^\delta(e_i) \oplus M_r^\delta(e_j) = \{M_r^\delta(e_i) - M_r^\delta(e_j)\} \cup \{M_r^\delta(e_j) - M_r^\delta(e_i)\} \quad (24)$$

Definition 15: Let $SS = (P, E, R, V, K, \theta, \vec{f}, \vec{g})$ be a searchable system over ciphertext. For $\forall e_i, e_j \in E$, $\hat{M}_R^\delta(e_i)$ and $\hat{M}_R^\delta(e_j)$ are ciphertext granule vectors on feature set R in δ-neighborhood. We define four operators $\cap$, $\cup$, $-$, and $\oplus$ below:

$$\hat{M}_R^\delta(e_i) \cap \hat{M}_R^\delta(e_j) = \{r \in R|M_r^\delta(e_i) \cap \hat{M}_r^\delta(e_j)\} \quad (25)$$

$$\hat{M}_R^\delta(e_i) \cup \hat{M}_R^\delta(e_j) = \{r \in R|M_r^\delta(e_i) \cup M_r^\delta(e_j)\} \quad (26)$$

$$\hat{M}_R^\delta(e_i) - \hat{M}_R^\delta(e_j) = \{r \in R|M_r^\delta(e_i) - M_r^\delta(e_j)\} \quad (27)$$

$$\hat{M}_R^\delta(e_i) \oplus \hat{M}_R^\delta(e_j) = \{r \in R|M_r^\delta(e_i) \oplus M_r^\delta(e_j)\} \quad (28)$$

Definition 16: Let $SS = (P, E, R, V, K, \theta, \vec{f}, \vec{g})$ be a searchable system over ciphertext, where $E = \{e_1, e_2, \ldots, e_n\}$ is ciphertext set and $R = \{r_1, r_2, \ldots, r_m\}$ is feature set. For $\forall e_i, e_j \in E$, there are two ciphertext granule vectors $\hat{M}_R^\delta(e_i) = (M_{r_1}^\delta(e_i), M_{r_2}^\delta(e_i), \ldots, M_{r_m}^\delta(e_i))$ and $\hat{M}_R^\delta(e_j) = (M_{r_1}^\delta(e_j), M_{r_2}^\delta(e_j), \ldots, M_{r_m}^\delta(e_j))$ on R in δ-neighborhood. The distance between $\hat{M}_R^\delta(e_i)$ and $\hat{M}_R^\delta(e_j)$ is defined by:

$$d(\hat{M}_R^\delta(e_i), \hat{M}_R^\delta(e_j)) = \frac{1}{|R| * |S|} \sum_{r \in R} \frac{|M_r^\delta(e_i) \oplus M_r^\delta(e_j)|}{|M_r^\delta(e_i) \cup M_r^\delta(e_j)|} \quad (29)$$

Definition 17: Let $SS = (P, E, R, V, K, \theta, \vec{f}, \vec{g})$ be a searchable system over ciphertext, where $P = \{p_1, p_2, \ldots, p_n\}$ is a plaintext set, $R = \{r_1, r_2, \ldots, r_m\}$ is a feature set, and $E = \{e_1, e_2, \ldots, e_n\}$ is a ciphertext set. For $\forall p \in P, e_p \in E$, we can define a rule on R as: $lb_R(p) =< \hat{N}_R(p), \hat{M}_R^\delta(e_p), e_p >$. Furthermore, rule library can be defined as: $LB_R = \{lb_R(p)|\forall p \in P\}$. Search over ciphertext can be converted into reasoning and matching in the rule library LB_R.

D. ENCRYPTING DATA

In this paper, we adopt AES for encryption and decryption for the small calculation overhead and a large block of data. The limitation is that a key has to be negotiated between the encryption side and the decryption side in advance, and then transmitted through the secure channel.

E. K-NEAREST NEIGHBORS CIPHERTEXT GRANULE SEARCH

1) k-NEAREST NEIGHBORS FUZZY GRANULE VECTOR

Definition 18: Given a searchable system over ciphertext $SS = (P, E, R, F, K)$, let Z be a fuzzy granule vector group on R, where $k > 0$ and k is an integer. For any fuzzy granule vector $z \in Z$, k-nearest neighbors fuzzy granule vector of z can be defined by:

$$\begin{aligned} KNN(z, Z) = \{T \subseteq Z|\forall t_i \in T, \forall t_j \in Z - T,\\ (|T| = K) \& d(z, t_i) \le d(z, t_j)\} \end{aligned} \quad (30)$$

A fuzzy granule vector group can be viewed as a set. The k-nearest neighbors fuzzy granule vector group is a subset of fuzzy granule vector group. They are the nearest k granule vectors to z in the fuzzy granule vector group. k-nearest Neighbors Ciphertext Granule Search (KNNCGS) is a decision algorithm based on fuzzy set operation, which is divided into granulation, matching and making decision process. The principle of KNNCGS is discussed below, and the algorithm is given.

2) PRINCIPLE OF KNNCGS

The KNNCGS includes granulation, matching, and making decision processes. The granulation process involves data pre-processing, dividing the training set and the test set. In the training set granulation, the feature fuzzy granulation and ciphertext granulation can form a rule library. Fuzzy

TABLE 1. ***K*-nearest neighbors ciphertext granule search algorithm on server.**

```
Input:   Original feature $\{v_{r_1}(t), v_{r_2}(t), ..., v_{r_m}(t)\}$ of test sample $t$, parameter $k$, neighborhood $\delta$, and key set $K$
Output:  Ciphertext and ciphertext granule of test sample $t$
1        Normalization and obfuscation of features:
         $\{\vec{fr_1}(t)A_t^{-1}, \vec{fr_2}(t)A_t^{-1}, ..., \vec{fr_m}(t)A_t^{-1}\} \leftarrow \{v_{r_1}(t), v_{r_2}(t), ..., v_{r_m}(t)\}$
         $\{A_t\vec{gr_1}(t), A_t\vec{gr_2}(t), ..., A_t\vec{gr_m}(t)\} \leftarrow \{v_{r_1}(t), v_{r_2}(t), ..., v_{r_m}(t)\}$
2        FOR $\exists p \in P \cup \{t\}$
             Granulate data on each atom feature $r_i \in R$, get fuzzy granule $N_{r_i}(p)$.
             Form fuzzy granule vector $\hat{N}_R(p) = (N_{r_1}(p), N_{r_2}(p), ..., N_{r_m}(p))$ on plaintext $p$.
             Get the ciphertext $e_p$ corresponding to $p$.
             Granulate data on every atom feature $r_i \in R$ of $\delta$ neighborhood of $e_p$ to get $M^{\delta}_{r_i}(e_p)$.
             Build ciphertext granule vector $\hat{M}^{\delta}_{R}(e_p) = (M^{\delta}_{r_1}(e_p), M^{\delta}_{r_2}(e_p), ..., M^{\delta}_{r_m}(e_p))$
             on $\delta$ neighborhood of $p$.
             For a training sample, a rule $lb_R(p) =< \hat{N}_R(p), \hat{M}^{\delta}_{R}(e_p), e_p >$ can be created
             and inserted in the rule library.
             For test sample $t$, $\hat{M}^{\delta}_{R}(e_t) = null, e_t = null$ (here, $null$ is not certain.)
         END FOR
         //For a test sample $t$, it can be compared with each fuzzy granule vector in the training set.
3        FOR $\exists p \in P$
             Calculate $d(\hat{N}_R(t), \hat{N}_R(p))$ according to definition 9.
             Insert $< \hat{N}_R(p), \hat{M}^{\delta}_{R}(e_p), e_p >$ and $d(\hat{N}_R(t), \hat{N}_R(p))$ into variable $T$.
         END FOR
4        Fuzzy granule vectors are be sorted by ascend distance in $T$, that is, $T \leftarrow SortAscend(T)$
5        Select the top $k$ fuzzy granule vectors in $T$, and insert ciphertext and ciphertext granule
         to object variable $O$. $O \leftarrow Select(T, K)$.
6        Count the votes of variable $O$, and the most ciphertext granules which belong to
         the same category represent the ciphertext granule corresponding to the test sample $t$.
         $< \hat{M}^{\delta}_{R}(e_s), e_s > \leftarrow Vote(O)$
7        Return $e_t$ and the ciphertext granule of $\delta$ neighborhood, $< \hat{M}^{\delta}_{R}(e_s), e_s >$
```

TABLE 2. ***K*-nearest neighbors ciphertext granule search algorithm on client.**

```
Input:   test sample $t$ and its key set $K$
Output:  the plaintext of $t$
1        Extract and obfuscate the feature of $t$: $\{\vec{fr_1}(t), \vec{fr_2}(t), ..., \vec{fr_m}(t)\}$
2        Upload $\{\vec{fr_1}(t)A_t^{-1}, \vec{fr_2}(t)A_t^{-1}, ..., \vec{fr_m}(t)A_t^{-1}\}$ to a server.
3        $< \hat{M}^{\delta}_{R}(e_s), e_s > \leftarrow GetResultsFromServer$
4        $PlainVoice \leftarrow Decryption(< \hat{M}^{\delta}_{R}(e_s), e_s >, key)$
5        Return plaintext.
```

granule vector matching process includes: Calculating the distance between test granule vector and all granule vectors in the rule library; Sorting by the distance; Selecting k-nearest rules. The decision process is to judge category of ciphertext granule according to fuzzy granule vector. The principle is as follows.

- **Step 1.** Pre-processing data - Delete the data with missing values and normalize the data set to the range [0, 1].
- **Step 2.** Divide 80% of data set as the training set and 20% of that as the test set.
- **Step 3.** Granulate data according to atom feature extracted by MFCC and obfuscated by obfuscation function and form fuzzy granule, fuzzy granule vector and ciphertext granule to build a rule library.
- **Step 4.** Searching and matching of fuzzy granule vectors. Take a test fuzzy granule vector, and calculate the distance between the test fuzzy granule vector and that of each rule, then sort the distance by ascend and select the top k fuzzy granule vectors.
- **Step 5.** Decision. The class of having the largest number of ciphertext granules associated with the k fuzzy granule vectors are selected as the final ciphertext granules (i.e., decision ciphertext granules).
- **Step 6.** Go to **Step 4** (Searching and matching of fuzzy granule vectors) and make the next test granule to decide, until all the test granule are finished. Get all decision ciphertext granules corresponding to all test fuzzy granule vectors.
- **Step 7.** Return the ciphertext with the corresponding ciphertext granule.

3) k-NEAREST NEIGHBORS CIPHERTEXT GRANULE SEARCH

After giving the principle above, we design the related algorithm, k-nearest neighbors ciphertext granule search (KNNCGS). The part of the algorithm is performed in a server (see Table 2), and the other part of the algorithm is executed in a client (see Table 3).

TABLE 3. Search performance of dataset.

Dictionary Size	AVE. Search Time of KNN (s)	AVE. Search Time of KNNCGS (s)	AVE. Space Cost of KNN (MB)	AVE. Space Cost of KNNCGS (MB)
10	0.05	0.09	0.11	0.13
15	0.07	0.12	0.13	0.14
20	0.10	0.14	0.14	0.15
25	0.12	0.16	0.16	0.21

IV. EVALUATION

A. SECURITY ANALYSIS

In our scheme, each sound can be encrypted with a unique key using AES which belongs to a symmetric encryption algorithm. Because of the AES security, the voice can not be decrypted by the adversary. Since each voice can be encrypted with a different key, it can be guaranteed that the same content in different voices will be encrypted into different ciphertext. It is a deterministic encryption that is resistant to selective plaintext attacks. When voice is uploaded to server, stored in server, and downloaded from server, the voice exists in the form of obfuscation feature and ciphertext. The features are calculated by MFCC and processed by obfuscation function, which can be hided and almost irreversible. In other words, the obfuscation features are very difficult to be recovered to the original voices. Therefore, the whole process is secure and reliable.

We analyse the security of our concrete scheme. The proposed scheme is adaptively secure (i.e. satisfies definition in [57]).

Proof: What we need to do is to construct a simulator $\mathcal{S} = \{\mathcal{S}_0, \ldots, \mathcal{S}_q\}$ such that for the adversary $\mathcal{A} = (\mathcal{A}_0, \ldots, \mathcal{A}_q)$, the outputs of Real(k) and Sim(k) are computationally indistinguishable. We construct a simulator $S = S_0, \ldots, S_q$ that adaptively produces a vector $v' = (t', E') = (t'_1, \ldots, t'_n, M^{\delta}_{R}(e_1)', \ldots, M^{\delta}_{R}(e_n)')$ where t'_i indicates the trapdoor of $\vec{f_i}$ and $t'_i = \vec{f_i}Q^{-1}$, $(Qarrow_r M_{m+3,m+3}(\mathbb{F})$, where $M_{m+3,m+3}(\mathbb{F})$ is a predetermined finite integral matrix group consisting of invertible $(m+3) \times (m+3)$ matrices over field $\mathbb{F}$.) as the follows:

1. $\mathcal{S}_0(1^k, \tau(F))$: it constructs a simulated $A_i arrow_r M_{m+3,1}(\mathbb{F})$ such that for the matrix $A_{|E|\times(m+3)} = (A^T_1, \ldots, A^T_{|E|})$. For a matrix A, the rank of A is denoted by $Rank(A) = min(|E|, m+3)$. So then includes A in $\mathcal{A}'s$ state st_s and outputs (E', st_s). We now claim that A_i are indistinguishable from $A_{\vec{g_i}}$, where $A_{\vec{g_i}} = Q\vec{g}^T_i$. It is evident that the distributions over A_i and $A_{\vec{g_i}}$ are identical. Furthermore, since the private-key encryption scheme is secure, each $M^{\delta}_{R}(e_i)'$ is indistinguishable from a real cipher granule vector.

2. $\mathcal{S}_1(st_s, \tau(F, \vec{f_1}))$: it solves system of linear equations $A\vec{f} = b_1$, where b_1 indicates the closeness degree between queried word $\vec{f_i}$ and noisy keyword $\vec{g_j}$. Note that it knows b_1 from the trace of $(F, \vec{f_1})$. We denote a solution of $A\vec{f} = b_1$ by t^* (if there exists solution). Let $t'_1 = t^{*T}$ that is indistinguishable from a real trapdoor t_1, since $t_1 \times A_{\vec{g_i}} = t'_1 \times A_i$ holds for $i \in [1, |E|]$. S_1 then includes t'_1 in st_s and outputs (t'_1, st_s).

3. $\mathcal{S}_i(st_s, \tau(F, \vec{f_1}, \ldots, \vec{f_i}))$: $\mathcal{S}_i$ generates a trapdoor t'_i in the same way that $\mathcal{S}_1$ does, i.e. by solving the system of linear equations $A\vec{f} = b_i$. $\mathcal{S}_i$ then includes t'_i in st_s and outputs (t'_i, st_s). It is evident that t'_i is indistinguishable from a real trapdoor t_i. This completes the proof.

B. EXPERIMENTAL RESULTS

To measure how well the KNNCGS performed at encrypted voice, we used 300 words of voice as a corpus to experiment, involving English, Chinese and Arabic. Since the value range of the data set is different, the data set needs to be normalized and obfuscated (see Table 2). The features of voice can be fuzzy granulated and form a fuzzy granule vector. Then, we granulated the ciphertext with δ-neighborhood of the fuzzy granule vector to build ciphertext granules. In order to verify the performance of the scheme, we compared KNN adopted in raw data with KNNCGS used in granule form. And we took the accuracy and recall as metrics of performance.

We fist explain criteria of the performance evaluation. True Positive (TP) is the number of positive samples predicted by model. True Negative (TN) denotes the number of negative samples predicted by model. False Positive (FP) expresses the number of the negative samples that is predicted as positive label by model. Relatively, False Negative (FN) represents the number of positive samples that is predicted as negative label by model. $TP\ Rate = \frac{TP}{TP+FN}$ is called true positive rate. $FP\ Rate = \frac{FP}{FP+TN}$ denotes false positive rate We use accuracy, recall to metric the performance as the follows: $Accuracy = \frac{TP}{TP+FP}$; $Recall = \frac{TP}{TP+FN}$. In the evaluation, we exhibited the relationship between metrics and the parameters of nearest neighbor and δ neighborhood (See Fig. 2-9).

As shown in Fig. 2, when K = 3 (the parameter of nearest neighbor), the accuracy of KNN was 0.93. In constrast, KNNCGS' accuracy reached peak value 0.951 at $\delta = 0.55$. It improved by 2.26%. The accuracy of KNNCGS was almost higher than that of KNN between $\delta = 0.05$ and $\delta = 0.85$. From $\delta = 0.85$ to $\delta = 1$, with δ rising, KNN's accuracy was higher KNNCGS's. In most cases with K = 3, KNNCGS is better than KNN at accuracy.

When K is 5, the results for different δ are exhibited in Fig. 3. Compared by Fig. 1, KNN got 0.933 (improvement by 0.32%). KNNCGS reached 0.95 at $\delta = 0.20$ and $\delta = 0.55$, respectively. Compared with KNN, KNNCGS got improvement by 1.82%. From $\delta = 0.15$ to $\delta = 0.85$,

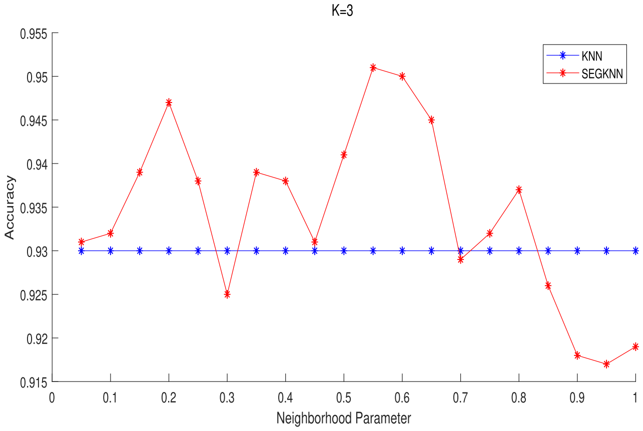


FIGURE 2. **The Accuracy of K = 3 and $\delta \in [0, 1]$.**

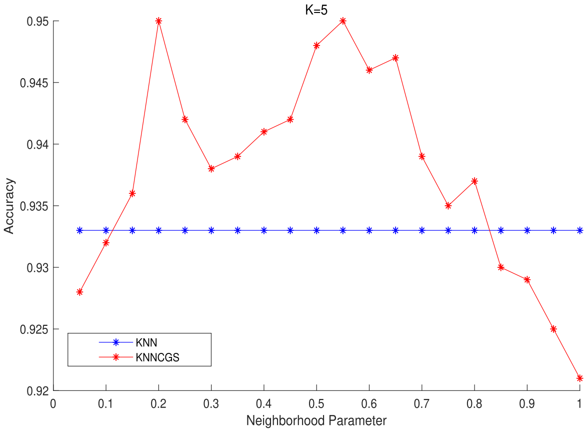


FIGURE 3. **The Accuracy of K = 5 and $\delta \in [0, 1]$.**

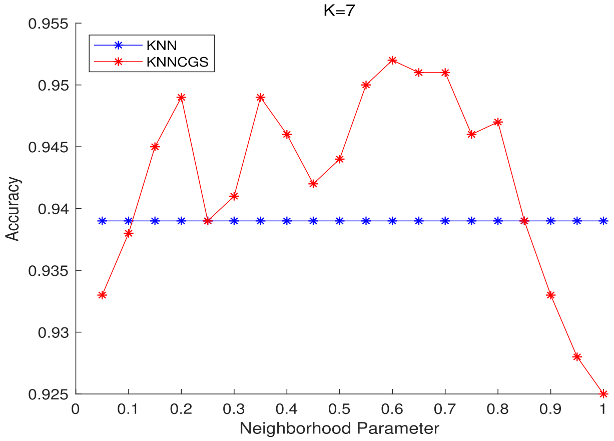


FIGURE 4. **The Accuracy of K = 7 and $\delta \in [0, 1]$.**

the accuracy of KNN was lower than that of KNNCGS. When $\delta = 1.0$, KNNCGS reached the lowest value 0.921 (decreased by 1.2%).

As demonstrated in Fig. 4, when $K = 7$, KNNCGS achieved top value 0.952 at $\delta = 0.6$ and was increased by 1.38% (KNN's accuracy was 0.939). KNN made an improvement by 0.64% and 0.97% compared by itself with $K = 5$ and $K = 3$ respectively. From $\delta = 0.15$ to $\delta = 0.85$,

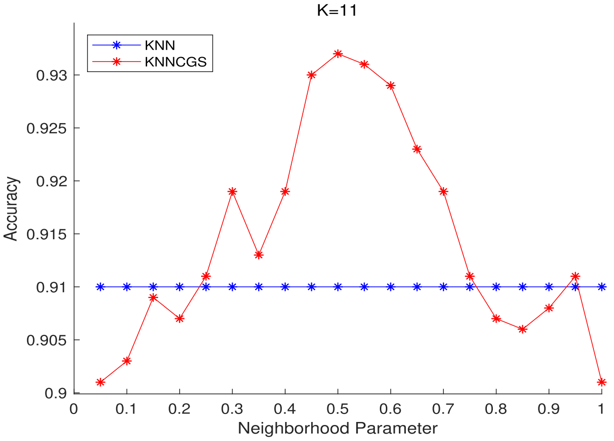


FIGURE 5. **The Accuracy of K = 11 and $\delta \in [0, 1]$.**

KNNCGS' accuracy was always higher than KNN's accuracy. However, when $\delta > 0.85$, the accuracy of KNNCGS was dropped quickly and decreased by 2.84% compared with its top value.

As shown in Fig. 5, when K reached 11, KNN and KNNCGS both decreased at top value. As far as KNN was concerned, it only reached 0.91 and dropped by 4.41% compared with its peak value. In contrast, KNNCGS's accuracy was 0.932 and decreased by 2.10% at its top value. Compared by KNN, the accuracy of KNNCGS is still higher than that of KNN between $\delta = 0.25$ and $\delta = 0.75$.

The recall rate is another important metric of performance. From Fig. 6 to Fig. 9, we compared the recall rate between KNN and KNNCGS. As shown in Fig. 6 (here, $K = 3$), KNNCGS achieved 0.961 at its peak value, but KNN was 0.95. KNNCGS improved by 1.12%. KNN was lower than KNNCGS between $\delta = 0.4$ and $\delta = 0.65$. The valley value of KNNCGS was 0.937 (decreased by 1.37%). At $\delta = 0.1, 0.2, 0.4$ and 0.7, KNN and KNNCGS were almost the same.

Note that $K = 5$ in Fig. 7. When $\delta = 0.5$, KNNCGS achieved a recall rate of 0.962, while KNN got 0.95 (1.26% improvement). From $\delta = 0.4$ until $\delta = 0.65$, the recall rate of KNN was lower than KNNCGS. When $\delta = 0.9$, the recall rate of KNNCGS reached its valley value of 0.937 and was decreased by 1.37%. When $\delta = 0.1$ and $\delta = 0.7$, their recall rate were almost same.

When $K = 7$ in Fig. 8, KNNCGS got a recall rate of 0.958 at $\delta = 0.5$ but KNN reached 0.949 (improvement 0.95%). When $\delta < 0.3$, the recall rate of KNN was higher than that of KNNCGS. The metric of KNNCGS was increased quickly between $\delta = 0.05$ and $\delta = 0.3$. The rate of growth was 19.2%. It reflected that the neighborhood parameter δ is important to the results.

The recall rates of both KNN and KNNCGS were decreased when $K = 11$, as shown in Fig. 9. KNN reached its valley value of 0.906 and had dropped by 4.63% compared with the highest value (when $K = 5$). Similarly, the maximum recall rate of KNNCGS was down to 3.12% from the highest

TABLE 4. Compare of search performance between [59] and KNNCGS.

Algorithm	Security	AVE. Accuracy	AVE. Recall	AVE. Space Cost	AVE. Search Time
Scheme-I in [59]	CPA-secure	0.93	0.93	0.17	0.15
Scheme-II in [59]	CPA-secure	0.94	0.93	0.16	0.14
KNNCGS	Adaptively Secure	0.95	0.93	0.23	0.17

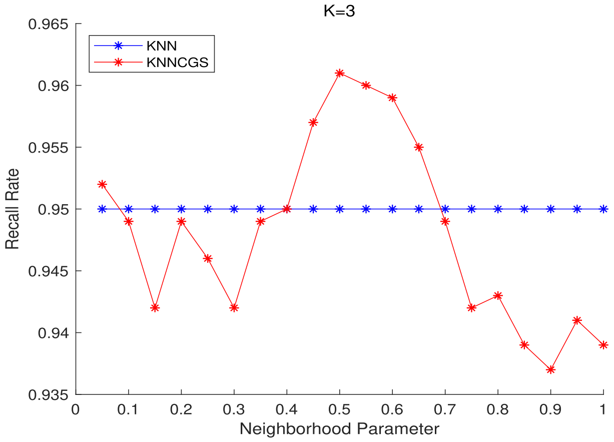


FIGURE 6. The Recall of K = 3 and $\delta \in [0, 1]$.

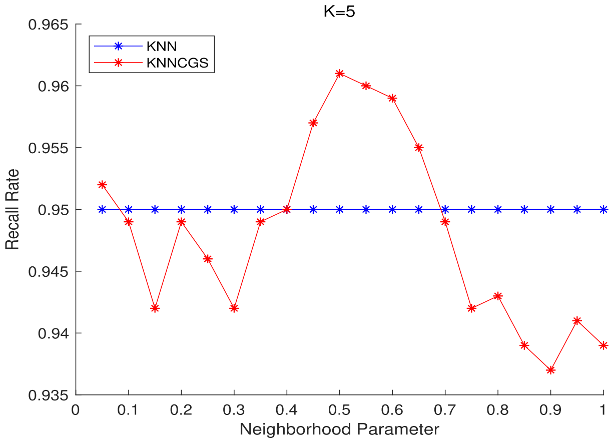


FIGURE 7. The Recall of K = 5 and $\delta \in [0, 1]$.

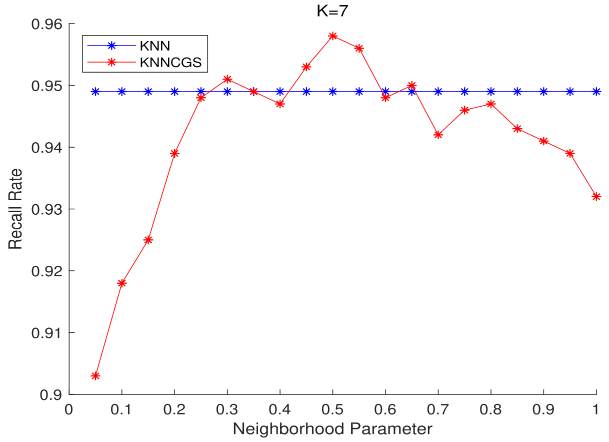


FIGURE 8. The Recall of K = 7 and $\delta \in [0, 1]$.

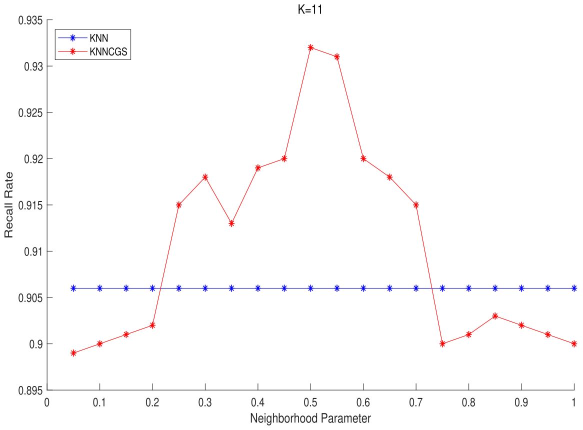


FIGURE 9. The Recall of K = 11 and $\delta \in [0, 1]$.

historical value. KNNCGS made an improvement by 2.87% at $\delta = 0.5$ compared with KNN. From $\delta = 0.25$ to $\delta = 0.7$, KNNCGS was performing slightly better.

The space cost is to measure the space efficiency of the index data structure. The space cost of the index should be practical compared to the original data size. Search time is to evaluate the search speed of answering on search query over the encrypted similarity sample. It includes the times of extracting features, cluster, fuzzy granule, encryption and decryption. As shown in Table 4, when the dictionary size is 10, the average search time of KNNCGS is more than that of KNN, and the average space cost of KNNCGS is a little more than that of KNN. When $K = 25$, the average search time of KNN is 0.12 seconds and that of KNNCGS is 0.16 seconds. The search time and space cost of KNN is superior to those of KNNCGS. The main reason is that KNNCGS involves the granule process compared with KNN.

When the size of dictionary is 30, scheme I and II of [59] were compared with KNNCGS. As demonstrated in Table 5, the security of scheme I and II were both CPA-secure and that of KNNCGS was adaptively secure. KNNCGS achieved the average accuracy of 95%. Scheme I and II were 93% and 94%. It enhanced by 2.05% and 1.06% respectively. The average recall rates of three algorithms were the same and were 93%. The average search time of KNNCGS is 0.17 seconds. It increased by 13.33% and 21.43% respectively. It costed time in granule process. The average space cost of KNNCGS increased by 0.06 MB and 0.07 MB respectively compared with scheme I and II.

Overall, KNNCGS outperforms KNN by adjusting its neighborhood parameter δ. The main reason lies in two aspects. On the one side, fuzzy granulation was considered before searching and it embodied the view of the collective structures of all voices. On the other side, the equivalence class principle was taken into account, which can cluster the encrypted voice according to fuzzy granule vector. The cluster voice can be achieved by KNNCGS. In contrast, KNN only got the optimal solution by calculating raw features.

V. CONCLUSION

This paper has presented the design of a searchable scheme over encrypted voice by using the Granule Computing technique. The voices' features obfuscated and the voices encrypted by AES algorithm were stored in the server. In order to prevent the restoration of voice features, we also use the obfuscated function to further process the features of the voice. The security is improved greatly by binding obfuscated features and encrypted voice. In addition, a series of concept have been defined, such as fuzzy granule, fuzzy granule vector, ciphertext granule, operators and metrics. Based on the defined concepts, both the neighbor fuzzy granule vector and the counting voting strategy were deployed to retrieve the ciphertext. The results were returned as the form of ciphertext granule, i.e. ciphertext equivalence class. Its security was analysed. The experimental results demonstrated that KNNCGS employed in encrypted voice is feasible and secure. Also, its performance is superior to that of KNN given special parameters.

The performance of KNNCGS is very much depended on neighbor parameter δ and the balance of dataset. In the future, we plan to consider the localized granulation rather than the global one, as well as parallel and distributed strategies, in order to improve the performance further and apply the scheme to the research of big data.

REFERENCES

[1] X. D. Song, D. Wagne, and A. Perri, "Practical techniques for searches on encrypted data," in *Proc. IEEE Symp. Secur. Privacy*, Nov. 2000, pp. 44–55.

[2] E. Goh. (2003). *Secure Indexes*. [Online]. Available: http://eprint.iacr.org/20003/216

[3] Y.-C. Chang and M. Mitzenmacher, "Privacy preserving keyword searches on remote encrypted data," in *Applied Cryptography and Network Security*. Berlin, Germany: Springer, 2005, pp. 442–455.

[4] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: Improved definitions and efficient constructions," *J. Comput. Secur.*, vol. 19, no. 5, pp. 895–934, Nov. 2011.

[5] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn.* Berlin, Germany: Springer, 2004, pp. 506–522.

[6] M. Abdalla, M. Bellare, D. Catalano, E. Kiltz, T. Kohno, T. Lange, J. Malone-Lee, G. Neven, P. Paillier, and H. Shi, "Searchable encryption revisited: Consistency properties, relation to anonymous IBE, and extensions," in *Advances in Cryptology*. Berlin, Germany: Springer, 2005, pp. 205–222.

[7] J. Baek, R. Safavi-Naini, and W. Susilo, "Public key encryption with keyword search revisited," in *Computational Science and its Applications*. Berlin, Germany: Springer, 2008, pp. 1249–1259.

[8] H. S. Rhee, J. H. Park, W. Susilo, and D. H. Lee, "Improved searchable public key encryption with designated tester," in *Proc. 4th Int. Symp. Inf., Comput., Commun. Secur. (ASIACCS)*, New York, NY, USA, 2009, pp. 376–379.

[9] L. Fang, W. Susilo, C. P. Ge, and J. D. Wang, "A secure channel free public key encryption with keyword search scheme without random oracle," in *Cryptology and Network Security*. Berlin, Germany: Springer, 2009, pp. 248–258.

[10] H. Li, D. Liu, Y. Dai, and T. H. Luan, "Engineering searchable encryption of mobile cloud networks: When QoE meets QoP," *IEEE Wireless Commun.*, vol. 22, no. 4, pp. 74–80, Aug. 2015.

[11] Z. Fu, X. Sun, Q. Liu, L. Zhou, and J. Shu, "Achieving efficient cloud search services: Multi-keyword ranked search over encrypted cloud data supporting parallel computing," *IEICE Trans. Commun.*, vol. E98.B, no. 1, pp. 190–200, 2015.

[12] J. Li, C. F. Jia, Z. Liu, J. Li, and M. Li, "Survey on the searchable encryption," *J. Softw.*, vol. 26, no. 1, pp. 109–128, 2015.

[13] J. Cui, H. Zhou, H. Zhong, and Y. Xu, "AKSER: Attribute-based keyword search with efficient revocation in cloud computing," *Inf. Sci.*, vol. 423, pp. 343–352, Jan. 2018.

[14] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. J. Lou, "Fuzzy keyword search over encrypted data in cloud computing," in *Proc. IEEE INFOCOM*, San Diego, CA, USA, Mar. 2010, pp. 1–5.

[15] M. Hadian, T. Altuwaiyan, X. Liang, and W. Li, "Privacy-preserving voice-based search over mHealth data," *Smart Health*, vol. 12, pp. 24–34, Apr. 2019.

[16] F. Brasser, T. Frassetto, K. Riedhammer, A.-R. Sadeghi, T. Schneider, and C. Weinert, "VoiceGuard: Secure and private speech processing," in *Proc. Interspeech*, 2018, pp. 1303–1307.

[17] F. Zhao, F. Yan, H. Jin, L. T. Yang, and C. Yu, "Personalized mobile searching approach based on combining content-based filtering and collaborative filtering," *IEEE Syst. J.*, vol. 11, no. 1, pp. 324–332, Mar. 2017.

[18] K. W.-T. Leung, D. L. Lee, and W.-C. Lee, "PMSE: A personalized mobile search engine," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 4, pp. 820–834, Apr. 2013.

[19] Z. Fu, K. Ren, J. Shu, X. Sun, and F. Huang, "Enabling personalized search over encrypted outsourced data with efficiency improvement," *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 9, pp. 2546–2559, Sep. 2016.

[20] X. Zhang, C. Xu, H. Wang, Y. Zhang, and S. Wang, "FS-PEKS: Lattice-based forward secure public-key encryption with keyword search for cloud-assisted industrial Internet of Things," *IEEE Trans. Depend. Sec. Comput.*, to be published, doi: 10.1109/tdsc.2019.2914117.

[21] L. A. Zadeh, "Fuzzy sets and information granularity," in *Advances in Fuzzy Set Theory and Applications*, vol. 2, no. 1, M. Gupta, R. Ragade, and R. Yager, Eds. Amsterdam, The Netherlands: North-Holland Publishing, 1979, pp. 3–18.

[22] J. R. Hobbs, "Granularity," in *Proc. IJCAI*, Los Angeles, CA, USA, 1985, pp. 432–435.

[23] B. Zhang and L. Zhang, *Theory and Applications of Problem Solving*. Amsterdam, The Netherlands: Elsevier, 1992.

[24] R. R. Yager and D. Filev, "Operations for granular computing: Mixing words with numbers," in *Proc. IEEE Int. Conf. Fuzzy Syst.*, Anchorage, AK, USA, May 1998, pp. 123–128.

[25] L. A. Zadeh, "Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic," *Fuzzy Sets Syst.*, vol. 90, no. 2, pp. 111–127, Sep. 1997.

[26] W. Pedrycz, "Granular computing for data analytics: A manifesto of human-centric computing," *IEEE/CAA J. Autom. Sinica*, vol. 5, no. 6, pp. 1025–1034, Nov. 2018.

[27] J. Zalewski, "Rough sets: Theoretical aspects of reasoning about data," *Control Eng. Pract.*, vol. 4, no. 5, pp. 741–742, May 1996.

[28] Q. Hu, Z. Xie, and D. Yu, "Hybrid attribute reduction based on a novel fuzzy-rough model and information granulation," *Pattern Recognit.*, vol. 40, no. 12, pp. 3509–3521, Dec. 2007.

[29] C. C. Aggarwal and C. K. Reddy, "Rough sets," in *Data Classification: Algorithms and Applications*. London, U.K.: Chapman & Hall, 2014.

[30] H.-P. Kriegel, P. Kröger, and A. Zimek, "Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering," *Pattern Recognit.*, vol. 3, no. 1, pp. 3509–3521, 2009.

[31] L. Zhang, C. Chen, J. Bu, Z. Chen, D. Cai, and J. Han, "Locally discriminative coclustering," *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 6, pp. 1025–1035, Jun. 2012.

[32] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, no. 5814, pp. 972–976, Feb. 2007.

[33] A. Ahmad and L. Dey, "A k-mean clustering algorithm for mixed numeric and categorical data," *Data Knowl. Eng.*, vol. 63, no. 2, pp. 503–527, Nov. 2007.

[34] C.-C. Hsu, C.-L. Chen, and Y.-W. Su, "Hierarchical clustering of mixed data based on distance hierarchy," *Inf. Sci.*, vol. 177, no. 20, pp. 4474–4492, Oct. 2007.

[35] S. S. Bucak, R. Jin, and A. K. Jain, "Multiple kernel learning for visual object recognition: A review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 7, pp. 1354–1369, Jul. 2014.

[36] M. Yang, L. Zhang, D. Zhang, and S. L. Wang, "Relaxed collaborative representation for pattern classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, Jun. 2012, pp. 2224–2231.

[37] P. C. Wu, S. C. H. Hoi, H. Xia, P. L. Zhao, D. Wang, and C. Miao, "Online multimo del deep similarity learning with application to image retrieval," in *Proc. 21st ACM Int. Conf. Multimedia (MM)*, New York, NY, USA, 2013, pp. 153–162.

[38] Y. Y. Yao, "A generalized decision logic language for granular computing," in *Proc. IEEE Int. Conf. Fuzzy Syst.*, Honolulu, HI, USA, 2002, vol. 21, no. 5, pp. 12–17.

[39] Q. Liu and Q. Liu, "Granules and applications of granular computing in logical reasoning," *J. Comput. Res. Develop.*, vol. 41, no. 4, pp. 546–551, 2004.

[40] H. Thiele, "On semantic mo dels for investigating computing with words," in *Proc. 2nd Int. Conf. Knowl. Based Intell. Electron. Syst.*, Adelaide, SA, Australia, 1998, pp. 32–98.

[41] K. Daphne and F. Nir, *Probabilistic Graphical Models: Principles and Techniques*. Cambridge, MA, USA: MIT Press, 2009.

[42] N. Friedman, "Inferring cellular networks using probabilistic graphical models," *Science*, vol. 303, no. 5659, pp. 799–805, Feb. 2004.

[43] J. Fan, Y. Gao, H. Luo, and R. Jain, "Mining multilevel image semantics via hierarchical classification," *IEEE Trans. Multimedia*, vol. 10, no. 2, pp. 167–187, Feb. 2008.

[44] Y. M. Ye, Q. Y. Wu, J. Z. X. Huang, M. K. Ng, and X. T. Li, "Stratified sampling for feature subspace selection in random forests for high dimensional data," *Pattern Recognit.*, vol. 46, no. 3, pp. 769–787, 2013.

[45] F. Chang, C.-Y. Guo, X.-R. Lin, and C.-J. Lu, "Tree decomposition for large-scale SVM problems," *J. Mach. Learn. Res.*, vol. 11, pp. 2935–2972, Oct. 2010.

[46] S. Gopal, Y. M. Yang, B. Bai, and A. Niculescu-Mizil, "Bayesian models for large-scale hierarchical classification," in *Proc. Adv. Neural Inf. Process. Syst.*, South Lake Tahoe, CA, USA, 2012, pp. 2420–2428.

[47] J. Qian, D. Miao, Z. Zhang, and X. Yue, "Parallel attribute reduction algorithms using MapReduce," *Inf. Sci.*, vol. 279, pp. 671–690, Sep. 2014.

[48] J. Liang, F. Wang, C. Dang, and Y. Qian, "An efficient rough feature selection algorithm with a multi-granulation view," *Int. J. Approx. Reasoning*, vol. 53, no. 6, pp. 912–926, Sep. 2012.

[49] Y. Qian, J. Liang, W. Pedrycz, and C. Dang, "Positive approximation: An accelerator for attribute reduction in rough set theory," *Artif. Intell.*, vol. 174, nos. 9–10, pp. 597–618, Jun. 2010.

[50] C. P. Chen and C.-Y. Zhang, "Data-intensive applications, challenges, techniques and technologies: A survey on big data," *Inf. Sci.*, vol. 275, pp. 314–347, Aug. 2014.

[51] S. K. Pal and D. B. Chakraborty, "Granular flow graph, adaptive rule generation and tracking," *IEEE Trans. Cybern.*, vol. 47, no. 12, pp. 4096–4107, Dec. 2017.

[52] G. Chiaselotti, T. Gentile, and F. Infusino, "Granular computing on information tables: Families of subsets and operators," *Inf. Sci.*, vols. 442–443, pp. 72–102, May 2018.

[53] G. Chiaselotti, T. Gentile, and F. Infusino, "Knowledge pairing systems in granular computing," *Knowl.-Based Syst.*, vol. 124, pp. 144–163, May 2017.

[54] O. Hryniewicz and K. Kaczmarek, "Bayesian analysis of time series using granular computing approach," *Appl. Soft Comput.*, vol. 47, pp. 644–652, Oct. 2016.

[55] J. Leng, Q. Chen, N. Mao, and P. Jiang, "Combining granular computing technique with deep learning for service planning under social manufacturing contexts," *Knowl.-Based Syst.*, vol. 143, pp. 295–306, Mar. 2018.

[56] Z. Han, J. Zhao, H. Leung, and W. Wang, "Construction of prediction intervals for gas flow systems in steel industry based on granular computing," *Control Eng. Pract.*, vol. 78, pp. 79–88, Sep. 2018.

[57] X. Pang, B. Yang, and Q. Huang, "Privacy-preserving noisy keyword search in cloud computing," in *Proc. Int. Conf. Inf. Commun. Secur.*, in Lecture Notes in Computer Science, vol. 7018, 2012, pp. 154–166.

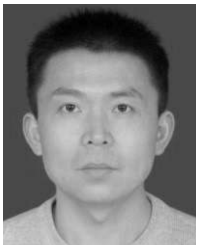
[58] W. Xu and J. Yu, "A novel approach to information fusion in multi-source datasets: A granular computing viewpoint," *Inf. Sci.*, vol. 378, pp. 410–423, Feb. 2017.

[59] Q. Wang, M. He, M. Du, S. S. M. Chow, R. W. F. Lai, and Q. Zou, "Searchable encryption over feature-rich data," *IEEE Trans. Depend. Sec. Comput.*, vol. 15, no. 3, pp. 496–510, May 2018.

WEI LI received the Ph.D. degree from Xiamen University, China, in 2013. He was also a Visiting Scholar with the Department of Computer Science, University of Massachusetts Boston, Boston, MA, USA, from June 2018 to June 2019. He is currently an Associate Professor with the School of Computer and Information Engineering, Xiamen University of Technology, China. His research interests include machine learning, fuzzy modeling and granular computing, big data, and information security. He has published over 20 articles in journals and conferences in these areas. He is a committee member of the China Computer Federation.

YUMIN CHEN received the M.E. degree in computer application technology from Nanchang University, China, in 2005, and the Ph.D. degree in pattern recognition and intelligence system from Tongji University, China, in 2010. From November 2018 to November 2019, he was a Visiting Scholar, supported by the Education Department of Fujian Province, with the Department of Mathematics and Statistics, Mississippi State University, Mississippi, USA. He is currently a Professor with the College of Computer and Information Engineering, Xiamen University of Technology. He has published over 50 articles in refereed journals and conferences. His research interests include machine learning, rough sets, and granular computing.

HUOSHENG HU (Senior Member, IEEE) is currently a Professor with the School of Computer Science and Electronic Engineering, University of Essex, U.K., leading the Robotics Research Group. His research interests include mobile robotics, human-robot interaction, embedded systems, mechatronics, learning algorithms, and cloud computing. He has published over 500 articles in journals, books, and conferences in these areas. He is a Fellow of the IET and InstMC. He currently serves as an Editor-in-Chief for the *International Journal of Automation and Computing*, and the Editor-in-Chief for the *MDPI Robotics Journal*.



CHAO TANG was born in Hefei, Anhui, China. He received the M.S. degree from Shanxi University, Taiyuan, China, in 2009, and the Ph.D. degree in artificial intelligence from Xiamen University, Xiamen, China, in 2014. He is currently an Associate Professor with the Department of Computer Science and Technology, Hefei University, China. His research and project works focus on machine learning, computer vision, and human action recognition.