IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 25, 2020, accepted January 19, 2021, date of publication January 22, 2021, date of current version January 29, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3053631

Regularized Sparse Modelling for Microarray Missing Value Estimation

AIGUO WANG[1], JING YANG[2], AND NING AN[2], (Senior Member, IEEE)
[1]School of Electronic Information Engineering, Foshan University, Foshan 528225, China
[2]School of Computer Science and Information Engineering, Hefei University of Technology, Hefei 230009, China

Corresponding author: Ning An (ning.an@hfut.edu.cn)

This work was supported by the Guangdong Basic and Applied Basic Research Foundation under Grant 2020A1515011499.

ABSTRACT The existence of missing values in microarray data inevitably hinders downstream biological analyses that expect complete data as input, therefore how to effectively explore the underlying structure of data to accurately estimate missing entries remains crucial and meaningful. In this study, we formalize the problem under a regularized sparse framework and accordingly propose local learning-based imputation models to capture the relationships that are hidden in gene expression profiles towards better imputation. Specifically, in view of the simultaneous variable selection and grouping effect of the elastic net penalty, we present an elastic net regularized local least squares-based imputation method to estimate the missing entries of a target gene with its neighbors. Besides, we investigate different similarity filtering metrics to select neighbor genes and develop another four imputation methods under the framework. Furthermore, the proposed methods process the target genes in ascending order of their associated missing rates. Finally, extensive comparative experiments against other eight commonly-used methods are conducted on multiple microarray datasets having varying missing rates. Results indicate the power of sparse regularization techniques and the superiority of elastic net over its competitors in terms of statistical analysis metrics.

INDEX TERMS Microarray data, missing value imputation, local structure, penalty.

I. INTRODUCTION

It has been known to us that DNA microarray technology provides researchers a high-throughput way to efficiently obtain the gene expression levels of a certain disease from different environments, subjects, tissues, and cell cycles and that microarray data analysis greatly facilitates the identification of disease genes and the diagnosis of cancers and tumor subtypes [1], [2]. Accordingly, researchers have utilized a wealth of statistical analysis and machine learning models (e.g., classification, clustering, feature selection, network analysis, and causal inference) to analyze gene expression profiles towards understanding the underlying biological mechanisms [3], [4]. However, both human and non-human factors, including, but not limited to, false positive PCR, inappropriate use of test chips, impurity of chip surface, and insufficient resolution of fluorescent images, can result in gene expression profiles with missing entries [5]. Previous studies indicate that most microarray datasets have different degrees of incompleteness that reach fifty percent and even up to ninety-five percent [6].

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

On the other side, a large number of data analysis models take as input a complete dataset, so the existence of missing entries inevitably impedes downstream biological analyses [7], [8]. Obviously, repeating experiments is a direct way to obtain complete gene expression profiles. However, the complex experimental procedure and no guarantee of returning a sample without any missing value via multiple replicates prevent it from being a priority in a practical setting [8], [9]. Besides, removing the genes with missing entries is a trivial solution, which seriously suffers from substantial loss of information. It is even worse if the discarded genes are potential biomarkers. In addition, we can simply replace the missing entries with zeros, ones, or average of the observed values of a gene or a sample [5]. Despite easy and efficient implementation, these methods tend to return estimations that largely deviate from the true values, as they ignore the valuable structure information latent in the dataset (e.g., the covariance structure and gene co-expression). Therefore, microarray missing value estimation remains a challenging yet rewarding topic deserving further investigation [10].

To facilitate the analysis of incomplete microarray data, researchers have designed a great number of missing value

imputation algorithms [11]. According to the information utilization scheme, we broadly categorize them into biology knowledge-, global learning-, local learning-, and hybrid-based methods. Biology knowledge-based methods take as prior information the biologically validated knowledge to establish the relationships among genes and use it to impute missing values [12], [13]. One major limitation is that they often require specific domain knowledge. Hence, they have poor extensibility to new and under-explored cases where there is a lack of verified biological knowledge. In contrast to the above methods, global learning-based methods adopt a data-driven strategy to estimate missing values under the assumption that a covariance structure exists in the obtained microarray dataset [14]. Such methods generally perform well on gene expression profiles with a large size, but they suffer from performance degradation in the case where the global covariance structure does not exist or local structures dominate [15]. In contrast, local learning-based methods explore the latent local structure information to obtain the relationships between a target gene and its neighbors [5], [16]. They typically work by first identifying neighbor genes of a target gene $\mathbf{g}_t$ and then estimating the missing entries of $\mathbf{g}_t$ with its neighbors [17], [18]. For example, the k-nearest-neighbor imputation method (KNNimptue) selects k nearest neighbors of $\mathbf{g}_t$ according to a distance metric and estimates the missing values by weighting its k neighbors [5]. Local least squares imputation method (LLSimpute) applies the least squares regression model to explicitly establish the relationships between $\mathbf{g}_t$ and its neighbors [18]. From the viewpoint of model fitting, the inappropriate selection of neighbors can lead to degraded accuracy. Hybrid-based methods aim to get improved results by combining multiple well-performing imputation algorithms in a sequential or parallel scheme [19], [20]. In practice, many hybrid-based methods take as the building blocks several global learning and/or local learning-based methods. For example, there are studies that linearly integrate multiple imputation methods and transform it into an optimization problem [19], combine estimators under an ensemble learning framework [20], or take the output of an imputation algorithm as the input of subsequent algorithms [21]. Undoubtedly, this increases the complexity of an imputation algorithm.

With respect to the high-dimension small-sample-size microarray data, since they typically contain a multitude of genes that have similar expression profiles, local learning-based methods generally better utilize the local structure of the data and get better imputation results than that of global learning-based methods. Currently, there are numerous imputation methods available, but most of them suffer from over-fitting and degraded performance. Considering that regularization techniques help mitigate this issue, we herein introduce a regularized sparse framework to establish the relationships between a target gene and its neighbors for missing value estimation. In view of the simultaneous variable selection and grouping effect of elastic net penalty [22], we design an elastic net regularized local least squares imputation method, called RLLSimpute_EN, to capture the hidden data structure information. RLLSimpute_EN ranks the target genes according to the missing rates and handles them sequentially from the minimal to the maximal missing rate to exploit previously estimated values. This enables us to obtain a robust missing value estimator. Besides, we take a further step to introduce a different similarity metric for selecting neighbor genes. Specifically, we use a filtering metric to exclude the genes with large missing rates from the candidates in choosing the neighbors of a target gene, and accordingly we design another four imputation methods with different regularization terms under the framework and also experimentally compare them with RLLSimpute_EN. The main contributions of this study are itemized as follows. (1) We present a regularized sparse framework to impute missing entries of microarray data and propose an elastic net regularized local least squares-based imputation method to capture the relationships between a target gene and its neighbors. This helps utilize the latent local structure of data and reduce the risk of overfitting. (2) We introduce a filtering metric to select neighbor genes and integrate it into the framework. Moreover, we propose another four missing value imputation methods. This shows the flexibility of the framework and presents a meaningful insight in choosing neighbors. (3) Extensive experiments on eight microarray datasets are conducted to evaluate the goodness of the proposed methods and compare them with other eight commonly used imputation algorithms in terms of three metrics. Particularly, three different regularization techniques are evaluated. Experimental results demonstrate the power of sparse regularization and the superiority of the proposed models.

The remainder of this paper is organized as follows. Section II discusses related work on imputation methods. Section III details the regularized sparse framework and introduces the proposed imputation methods. In section IV, experimental setup, evaluation metrics, and results and analysis are presented. Section V analyzes the theoretical time complexity. The last section summarizes the study.

II. RELATED WORK

To maximize the value of gene expression profiles and serve downstream analyses, researchers have proposed a lot of missing value estimation methods. As we discussed in the previous section, we can categorize them into biology knowledge-, global learning-, local learning-, and hybrid-based methods. To be specific, biology knowledge-based methods use the validated biology knowledge (e.g., gene function network, protein-protein interaction networks, and gene ontology) to estimate missing entries [12], [13]. For example, Yang *et al.* [12] proposed a gene ontology-based similarity measure to select neighbor genes and used them to impute missing values. Xiang *et al.* [13] proposed the histone acetylation information aided imputation algorithm (called HAIimpute) with the knowledge of gene regulatory mechanism. They conducted comparisons with KNNimptue and LLSimpute to show the effectiveness of HAIimpute.

Obviously, knowledge-driven methods depend heavily on domain knowledge. Hybrid-based methods are basically built on multiple local learning and global learning-based methods to improve the overall imputation performance. For example, Jornsten *et al.* [19] proposed to combine global and local learning-based methods (LinCmb) to estimate the missing values. Li *et al.* [23] combined the predictions from local least squares-based imputation method and Bayesian principal component analysis imputation method under the ensemble framework and inferred the missing values from the weighted outputs of its components. Meng *et al.* [24] proposed to combine Bayesian principal component analysis and bicluster analysis, where the latter filtered the selection of neighbors and the former was applied on the biclusters to explore the local data structure. To ease the setting of initial parameter values, Shi *et al.* [21] used the output of the local least squares-based imputation algorithm to initialize the parameter values of Bayesian principal component analysis. For global learning-based methods, they generally assume that there exists a covariate structure in the studied dataset. Singular vector decomposition imputation (SVDimpute) and Bayesian principal component analysis imputation method (BPCAimptue) are two representatives [14], [15], where the former employs the singular value decomposition to obtain the most significant eigengenes and the latter utilizes the principal component analysis, probability estimation and variational Bayesian inference to infer the parameter values for missing values estimation. BPCAimpute iteratively performs the principal component regression, the Bayesian estimation, and an Expectation-Maximization-like algorithm until there is no missing entry [14]. Experimental results indicate the superiority of BPCAimpute over SVDimpute. Generally, global learning-based methods better handle the gene expression profiles of large sizes and easily suffer from degraded accuracy if the local structure dominates the data.

Unlike global learning-based methods, local learning-based methods, rather than rely on a covariate structure, utilize the similar genes of a target gene to estimate missing values [5], [18], [25], where how to identify neighbors of the target genes and further establish their relationships largely determines the imputation results of an algorithm. There are studies that divide genes into multiple clusters and use the within-group genes of the target gene to estimate missing values. For example, Ouyang *et al.* [26] gave the Gaussian mixture clustering imputation algorithm (GMCimpute), where the Gaussian mixture model is used to parameterize the gene expression profiles and estimate the missing values within a cluster. Keerin *et al.* [27] developed a cluster-directed framework for neighbor-based imputation method (CFNIimpute), which first chose neighbor genes using the data clustering technique and then estimated the missing values with KNNimpute. To improve the selection of similar genes, Chattopadhyay *et al.* [28] proposed a bicluster-based imputation method (BIimpute) that selected a subset of both samples and genes. BIimpute used the weighted average of similar samples and genes to estimate missing values of the target gene. Another line of research is to first measure the similarity between the target gene and each candidate gene and then choose the neighbor genes. For example, k-nearest-neighbor imputation method (KNNimptue) estimates the missing entries of a target gene with its k nearest neighbors [5]. KNNimpute first selects k nearest neighbors of a target gene according to a distance metric and then imputes the missing values by weighting the values of the k neighbors. According to KNNimpute, Kim *et al.* [29] proposed the sequential version of KNNimptue (SKNNimpute) that estimated missing values in a sequential scheme. Brás and Menezes [30] proposed the iterative k-nearest-neighbor imputation method (IKNNimptue) that works in an iterative scheme. One drawback of nearest neighbors-based methods is that they handle the neighbors independently and ignore the relationship between the neighbors. To mitigate the problem, there are studies that apply regression models to model the relationships between the target gene and its neighbors and use the optimized regression model for imputation. For example, least squares imputation method (LSimpute) [17] and LLSimpute [18] used the least squares regression model. Zhang *et al.* [31] proposed the sequentially local least squares-based imputation method (SLLSimpute). Wang *et al.* [32] proposed the shrinkage regression-based method (ShrinkageLLS) that first selected similar genes by Pearson correlation coefficients and then adjusted the regression coefficients with a shrinkage estimation operator. Compared with KNNimpute and its variations, regression-based methods generally obtain better results, especially when a larger number of neighbor genes are used [33]. However, regression methods easily suffer from severe overfitting. To mitigate this issue, Wang *et al.* [33] trained a local least squares imputation model with L_2 regularization between a target gene and its neighbors. Empirical results show that it outperforms other nine competitors, including BPCAimpute, BIimpute, KNNimpute, and LLSimpute. For regularization techniques, given a group of correlated variables, in the case of L_1 regularization, one of the correlated variables has a larger coefficient and the rest have a coefficient close to zero, while for L_2 regularization, the coefficients of correlated predictors are similar. That is, L_1 regularization works well if there are a small number of significant parameters, and L_2 regularization is preferable if there are a group of correlated features. Unfortunately, the selection of regularization terms depends on the data at hand, and we usually have no idea about the true parameter values in advance. Therefore, it is desirable to consider the robust elastic net regularization that owns the characteristics of both L_1 and L_2 regularization. Accordingly, we present a regularized sparse framework to formalize the problem and propose an elastic net regularized local least squares imputation method.

III. THE PROPOSED METHOD

For illustration purpose, gene expression profiles are often represented as a matrix $\mathbf{D} \in \mathrm{R}^{m \times n}$ ($m \ll n$), as shown in Fig. 1, where each column denoting a gene and each

$$G=[\mathbf{g}_1, \mathbf{g}_2, \mathrm{L}\ , \mathbf{g}_n]=\begin{bmatrix}\mathbf{s}_1^{\mathrm{T}}\\ \mathbf{s}_2^{\mathrm{T}}\\ \mathrm{M}\\ \mathbf{s}_m^{\mathrm{T}}\end{bmatrix}=\begin{bmatrix} g_{1,1} & g_{1,2} & x_{1,3} & \mathrm{L} & g_{1,n}\\ g_{2,1} & g_{2,2} & g_{2,3} & \mathrm{L} & g_{2,n}\\ g_{3,1} & g_{3,2} & g_{3,3} & \mathrm{L} & x_{3,n}\\ \mathrm{M} & \mathrm{M} & \mathrm{M} & \mathrm{O} & \mathrm{M}\\ g_{m,1} & g_{m,2} & g_{m,3} & \mathrm{L} & g_{m,n}\end{bmatrix}$$

FIGURE 1. Logical storage structure of microarray data.

row indicating a sample. We use $g_1, g_2, \ldots, g_n$ to denote n genes and $\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_n (\mathbf{g}_i \in \mathrm{R}^{1\times m}, 1 \le i \le n)$ to indicate the vector forms. $d_1, d_2, \ldots, d_n$are m samples and $\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_m (\mathbf{d}_i \in \mathrm{R}^{n\times 1}, 1 \le i \le m)$ correspond to their vectors. Specifically, Fig. 1 presents a microarray dataset with missing values. The entry $g_{i,j}$ indicates the expression level of the j-th gene of the i-th sample. We use $x_{i,j}$ to represent a missing entry at that position, such as $x_{1,3}$ and $x_{3,n}$. In this study, a gene with at least one missing entry across all samples is called a *target gene* $\mathbf{g}_t$ and all genes excluding $\mathbf{g}_t$ are *candidate genes*. The gene that has a similar expression pattern to $\mathbf{g}_t$ is called a similar gene. The collection of candidate genes consists of candidate neighbors of $\mathbf{g}_t$, and the set of similar genes has the similar genes of $\mathbf{g}_t$. Obviously, for $\mathbf{g}_t$, its similar genes is a subset of corresponding candidate genes. Thus, the task of an imputation algorithm is to accurately infer the missing entries of $\mathbf{g}_t$ with the observed values of its similar genes.

To fully utilize the local structure of data and achieve a better bias-variance tradeoff, we explore the regularization techniques and formalize a regularized sparse framework to capture the relationships between a target gene $\mathbf{g}_t$ and its neighbors. Specifically, the proposed framework mainly consists of the following six components: *A*) identifying $\mathbf{g}_t$ according to a given criterion; *B*) choosing the similar genes of $\mathbf{g}_t$ from its candidate neighbors; *C*) measuring the similarity between $\mathbf{g}_t$ and each of its similar genes based on a certain metric; *D*) training a model on $\mathbf{g}_t$ and its neighbors and applying it to estimate the missing values of $\mathbf{g}_t$; *E*) marking $\mathbf{g}_t$ as a complete gene; *F*) repeating the above steps until all target genes have been processed. In summary, Algorithm 1 presents the proposed missing value imputation framework, where lines 3-7 correspond to the key steps. In next subsections, based on the above discussions, we detail its key components and present the elastic net regularized imputation method. Fig. 2 shows the flow chart. Particularly, the proposed framework is a general one and other specific implementations can be integrated into it flexibly. Besides, in this study, we also introduce a filtering metric for the selection of similar genes into the framework and further present another four missing value imputation methods.

A. IDENTIFYING THE TARGET GENE

Given a microarray dataset, it usually contains more than one genes with missing values and we need to specify the order

Algorithm 1 The Proposed Imputation Framework

Input: Microarray dataset G with missing values
Output: Imputed dataset G
1 // initialization
  identify genes that have missing values and store them in g_s
2 **while** *not_empty*(g_s) **do**
3 // select the target gene
  identify the target gene $\mathbf{g}_t$ according to a certain strategy
4 // filter similar genes
  select the genes similar to $\mathbf{g}_t$ based on (1)
5 // choose nearest neighbors
  5.1) calculate the neighbor distances of $\mathbf{g}_t$ using (2)
  5.2) choose k nearest neighbors of $\mathbf{g}_t$
6 // imputation
  6.1) train a regression model using (4)
  6.2) estimate the missing values of $\mathbf{g}_t$ using (6) and return $\mathbf{g}$
7 // update g_s
  7.1) update G by replacing $\mathbf{g}_t$ with $\mathbf{g}$
  7.2) delete $\mathbf{g}_t$ from g_s
8 **end while**
9 **return** G;

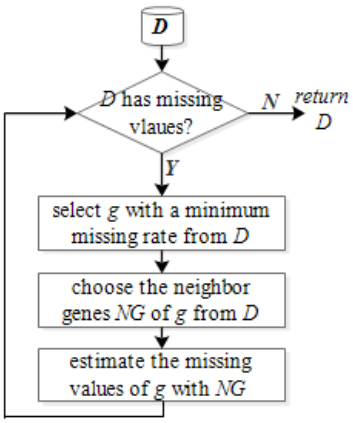


FIGURE 2. The flow chart for missing value estimation.

of handling these target genes. In general, there are multiple strategies that are available for use. For example, we can randomly choose a gene having at least one missing entry as the target gene. This, even though simple, poses a challenge to repeat the experimental results. To reduce the randomness, one common used strategy is to scan the gene expression profile forward or backward and sequentially estimate missing values [5], [18]. In addition, since handling a gene that has a larger missing rate is more challenging than the case of handling a gene with fewer missing entries, we sequentially estimate the missing values. Specifically, we handle the target gene that has a minimal missing rate in each round, which also enables us to reuse those estimated values in processing the genes that have larger missing rates. We here adopt it

to identify the target gene. Formally, for $\mathbf{D}$ that contains n_1 incomplete genes, we use a matrix $\mathbf{D}_1 \in \mathrm{R}^{m\times n_1}$ to store them and use $\mathbf{D}_2 \in \mathrm{R}^{m\times(n-n_1)}$ to record the complete genes. The missing rate r_i of gene $\mathbf{g}_i \in \mathbf{D}_1$ equals the ratio between the number of missing entries of $\mathbf{g}_i$ and the dataset size m. Then, the gene $\mathbf{g}_i$ with the minimum r_i is chosen as $\mathbf{g}_t$.

B. SELECTING SIMILAR GENES

After identifying the target gene $\mathbf{g}_t$, this step involves the selection of similar genes from the candidate set. Though the aim is to impute the missing entry of $\mathbf{g}_t$ with the observed values of its similar genes, not all candidate genes of $\mathbf{g}_t$ can be used and one requirement is that the similar genes should have observed values at the indices where $\mathbf{g}_t$ has missing values. For example, if $\mathbf{g}_t$ has a missing entry in the first example, a candidate gene $\mathbf{g}_c$ that misses the first entry provides no information to estimate the first value of $\mathbf{g}_t$. Suppose $idx(\mathbf{g})$ indicates the indices of missing entries of gene $\mathbf{g}$, the qualified similar gene should meet the criterion: the intersection between $idx(\mathbf{g}_c)$ and $idx(\mathbf{g}_t)$ is not empty.

$$idx(\mathbf{g}_t) \cap idx(\mathbf{g}_c) \neq \emptyset. \tag{1}$$

C. SIMILARITY MEASUREMENT

After determining a $\mathbf{g}_t$ and its similar genes, we measure the similarity between $\mathbf{g}_t$ and each of its similar genes using a certain distance metric. Generally, there are a multitude of distance metrics available for use (e.g., Pearson correlation coefficient, cosine distance, and Euclidean distance). Since most existing distance metrics take as input complete data, they have limited power in directly handling the case with missing entries. We herein only take their common parts for measurement. Specifically, for $\mathbf{g}_t$ and its similar gene $\mathbf{g}_c$, we use (2) to calculate the distance dis_c between $\mathbf{g}_c$ and $\mathbf{g}_t$,

$$dis_c = f(\mathbf{g}_t, \mathbf{g}_c) = sim(\mathbf{g}_t^{\sim idx(\mathbf{g}_t)\cap\sim idx(\mathbf{g}_c)}, \mathbf{g}_c^{\sim idx(\mathbf{g}_t)\cap\sim idx(\mathbf{g}_c)}) \tag{2}$$

where $\sim idx(\mathbf{g}_t)$ denotes the complementary set of $idx(\mathbf{g}_t)$, sim means a certain similarity metric such as Mahalanobis distance, Euclidean distance, cosine distance, and Pearson correlation coefficient, $\sim idx(\mathbf{g}_t)\cap \sim idx(\mathbf{g}_c)$ denotes the set of indices where both $\mathbf{g}_c$ and $\mathbf{g}_t$ have values, and $|\mathbf{A}|$ denotes the cardinality of the set $\mathbf{A}$. Among the above-mentioned distance metrics, Euclidean distance metric is widely used and previous studies have also shown its effectiveness in analyzing microarray data, therefore we use it, shown in (3).

$$dis_c = sim(\mathbf{g}_t, \mathbf{g}_c) = \sqrt{\sum_{i=1}^{|\sim idx(\mathbf{g}_t)\cap\sim idx(\mathbf{g}_c)|} (\mathbf{g}_t^i - \mathbf{g}_c^i)^2}. \tag{3}$$

D. ESTIMATING THE MISSING VALUES

According to the distances between $\mathbf{g}_t$ and its similar genes, we select k similar genes to estimate the missing values of $\mathbf{g}_t$. Particularly, the key is to establish the relationships between $\mathbf{g}_t$ and its neighbors and also avoid overfitting. We introduce a regularized sparse framework that builds a linear regression model on $\mathbf{g}_t$ and its k neighbors to capture the local structure, which seeks to solve the optimization problem (4),

$$\arg\min_{\boldsymbol{\beta}}\{(\mathbf{g}_t^{idx(\mathbf{g}_t)} - \sum_{c=1}^{k} \beta_c \mathbf{g}_c^{idx(\mathbf{g}_t)})^2 + \lambda R(\boldsymbol{\beta})\}, \tag{4}$$

where $\boldsymbol{\beta} = [\beta_1, \beta_2, \cdots, \beta_k]$, β_c is the regression coefficient of $\mathbf{g}_c (1 \leq c \leq k)$, dis_c is the distance between $\mathbf{g}_t$ and$\mathbf{g}_c$, and $R(\boldsymbol{\beta})$ is a regularization term. The parameter λ controls a tradeoff between small coefficients of $\boldsymbol{\beta}$ and data fitting.

In view of the simultaneous variable selection and grouping effect of elastic net penalty, we train an elastic net regularized local least squares-based imputation model (RLLSimpute_EN) to infer the missing values of $\mathbf{g}_t$ with its neighbors. RLLSimpute_EN takes the form of the specific objective function (5),

$$\begin{cases} \arg\min_{\boldsymbol{\beta}}\{(\mathbf{g}_t^{idx(\mathbf{g}_t)} - \sum_{c=1}^{k} \beta_c \mathbf{g}_c^{idx(\mathbf{g}_t)})^2 + \lambda R(\boldsymbol{\beta})\} \\ s.t. R(\boldsymbol{\beta}) = \alpha||\boldsymbol{\beta}||_1 + \dfrac{(1-\alpha)}{2}||\boldsymbol{\beta}||_2^2 \\ \quad \|\boldsymbol{\beta}\|_1 = \sum_{i=1}^{k} |\beta_i|, \|\boldsymbol{\beta}\|_2^2 = \sum_{i=1}^{k} \beta_i^2, \end{cases} \tag{5}$$

where λ controls the overall penalty and α balances the elastic net penalty. Furthermore, for the regularization term,

1) if $\lambda = 0$, RLLSimpute_EN becomes a standard regression model, which easily suffers from overfitting.
2) if $\alpha = 0$, RLLSimpute_EN reduces to the L_2 regularized regression model RLLSimpute_L2 that aims to minimize the sum of the squares of coefficients.
3) if $\alpha = 1$, RLLSimpute_EN equals the L_1 regularized regression model RLLSimpute_L1 that makes many coefficients close to zero.

For the three regularization terms, in comparison with L_2 regularization that keeps or discards a group of highly correlated variables in a model and L_1 regularization that tends to select one variable, the elastic net regularization is a compromise between L_1 and L_2 and enjoys the sparsity of L_1 and the regularization of L_2. This indicates that the elastic net largely contributes to the accurate and robust missing value estimation. After getting $\boldsymbol{\beta}^* = (\beta_1^*, \beta_2^*, \ldots, \beta_k^*)$ of (5), we estimate the missing values of $\mathbf{g}_t$ using (6),

$$\mathbf{g}_t^{\mathbf{miss}} = (\beta_1^*, \beta_2^*, \ldots, \beta_k^*) * [\mathbf{g}_1^{\mathbf{miss}}, \mathbf{g}_2^{\mathbf{miss}}, \cdots, \mathbf{g}_k^{\mathbf{miss}}]^{\mathbf{T}}, \tag{6}$$

where **miss** refers to the indices of samples without missing values for$\mathbf{g}_t$. After $\mathbf{g}_t$ is handled, we move it from $\mathbf{D_1}$ to$\mathbf{D_2}$. We then take the gene with minimal missing rate from $\mathbf{D_1}$ as the target gene. Repeat steps A-D until $\mathbf{D_2}$ is empty.

E. FILTERING SIMILAR GENES

In subsection *B*, we present the basic requirement for the selection of similar genes. In practice, we can involve other metrics to filter similar genes. We here take a further step to introduce a filtering metric into the proposed framework. Since a gene with many missing entries generally contains

less information, we can filter out the candidate genes that have a high missing rate besides the condition specified in (1). That is, we filter out the gene with a missing rate that is larger than the mean missing rate of all genes in $\mathbf{D_1}$, as shown in (7).

$$r_c < \frac{1}{n_1}\sum_{i=1}^{n_1} r_i. \tag{7}$$

Accordingly, we design another four imputation methods based on the filtering metric, including filtering local least squares-based imputation method (fLLSimpute), filtering local least squares-based imputation with L_1 regularization (fLLSimpute_L1), filtering local least squares-based imputation with L_2 regularization (fLLSimpute_L2), and filtering local least squares-based imputation with elastic net (fLLSimpute_EN).

IV. EXPERIMENTAL RESULTS AND ANALYSIS

A. EXPERIMENTAL DATA

To evaluate the proposed methods, comparative experiments are conducted on eight microarray datasets that cover both time series and non-time series data. A brief summary of them is given in Table 1. GDS38, GDS39, and GDS2967 are time series datasets and GDS1761, GDS3835, GDS4831, GSE19119, and GASCH are non-time series datasets [33]. The second column indicates the size of the original dataset, the third column shows the number of complete genes, and the fourth column gives the missing rate of a dataset. We observe that all datasets contain varying degrees of missing entries. For example, GDS1761 only has a missing rate of 0.15% and GDS3835 has a missing rate of 72.25%. The last column provides the associated references for further study.

TABLE 1. Description of the experimental datasets.

Dataset	Original dataset (genes, samples)	Complete dataset (genes, samples)	Missing rate (%)	Ref.
GDS38	7680, 16	5282, 16	6.10	36
GDS39	7680, 14	6942, 14	3.21	15
GDS2967	6159, 33	3587, 33	9.24	37
GDS1761	9706, 64	8849, 64	0.15	38
GDS3835	27648, 48	5070, 48	72.25	39
GDS4831	24526, 22	10523, 22	23.75	40
GSE19119	5299, 34	1617, 34	25.88	16
GASCH	6152, 173	2990, 154	3.01	41

B. EVALUATION METRICS

As for the evaluation metrics, we compare them in term of the root mean square error, Pearson correlation coefficient, and conserved pairs proportion. Particularly, the first two are statistical analysis related indicators (the first is a global metric and the second is a local metric), while the last one is biologically related.

1) ROOT MEAN SQUARE ERROR

Root mean square error (*RMSE*) measures the overall deviations of estimated values from their true values [34], and is calculated using (8),

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{m}\sum_{j=1}^{n}[G_{ori}(i,j) - G_e(i,j)]^2}, \tag{8}$$

where G_{ori} is the complete dataset, the incomplete G is randomly generated from G_{ori}, N equals the number of missing entries of G, and G_e is the output of an imputation method. Obviously, *RMSE* takes a non-negative value and the smaller the *RMSE*s, the better the corresponding method. If an imputation method works perfectly, *RMSE* equals 0.

2) PEARSON CORRELATION COEFFICIENT

Pearson correlation coefficient (*PCC*) measures the power of an imputation method in recovering the original structure of a dataset [35]. It works on the sample level rather than matrix level and it takes the form of (9),

$$correlation\ coefficient = \frac{\mathrm{cov}(\mathbf{s}_{ori}^{\mathbf{T}}, \mathbf{s}_e^{\mathbf{T}})}{std(\mathbf{s}_{ori}^{\mathbf{T}})std(\mathbf{s}_{ori}^{\mathbf{T}})}, \tag{9}$$

where $\mathbf{s}_{ori}^{\mathbf{T}}$ is a sample of G_{ori}, $\mathbf{s}_e^{\mathbf{T}} \in G_e$ is the estimated sample of $\mathbf{s}_{ori}^{\mathbf{T}}$, $\mathrm{cov}(\mathbf{s}_{ori}^{\mathbf{T}}, \mathbf{s}_e^{\mathbf{T}})$ is the covariance between $\mathbf{s}_{ori}^{\mathbf{T}}$ and $\mathbf{s}_e^{\mathbf{T}}$, and $std(\mathbf{s}_{ori}^{\mathbf{T}})$ ($std(\mathbf{s}_e^{\mathbf{T}})$) is the standard deviation of $\mathbf{s}_{ori}^{\mathbf{T}}$ ($\mathbf{s}_e^{\mathbf{T}}$). Pearson correlation coefficient takes a value between -1 and 1 and a larger value indicates better performance of an algorithm.

3) CONSERVED PAIRS PROPORTION

Conserved pairs proportion (*CPP*) is a biological indicator to evaluate the stability of two groups of gene clusters that are obtained on the original complete dataset and on the estimated dataset, respectively [42]. Similar to *CPP*, the average distance between partitions (*ADBP*) also aims to evaluate how well an imputation algorithm preserves the cluster structures. Particularly, *CPP* uses the hierarchical clustering algorithm and *ADBP* uses the k-means to cluster data points. Compared with *CPP*, *ADBP* is sensitive to the choice of the initial cluster centers of k-means. We here use *CPP*, which is also used by previous studies, to assess the structure preservation [10], [33], [42]. Given the original dataset G_{ori}, we use C_k^{ref} and L_k^{ref} to denote the k-*th* cluster and its gene list, respectively. For the estimated G_e of G_{ori}, $C_{k'}^{est}$ and $L_{k'}^{est}$ indicate the k'-*th* cluster and its gene list, respectively. Afterwards, *CPP* is obtained using (10),

$$CPP = \sum_{k=1}^{k=K} N_k/n, \tag{10}$$

where K is the number of clusters, n is the total number of genes, and N_k is obtained using (11),

$$N_k = \max_{k'=1,\cdots,K}\left(\sum_{i\in L_k^{ref}}\sum_{i'\in L_{k'}^{est}} I(i == i')\right), \tag{11}$$

where $I(\ )$ is an indicator function. Obviously, *CPP* has the maximal value 1 if the clustering results are the same.

C. EXPERIMENTAL RESULTS AND ANALYSIS

1) LAMDA-VALUE SELECTION

The parameter λ of regularized local learning methods controls a tradeoff between fitting the training set well and obtaining small weights. Herein, we repeat experiments ten times on each dataset at a representative 5% missing rate to search for the approximately optimal value of λ. According to our preliminary work, we determine the λ value from 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.1, 0.2, 0.5, 0.8, 1, 2, 2.5, 3, 3.5, 4, 5, and 6 by comparing the averaged *RMSE*s. Similar to λ, we can search for the value of alpha in elastic net from a set of candidate values. We here empirically tune the value and use 0.2 in this study. The sparse learning with efficient projections toolbox is used to solve the above optimization problem [43]. Fig. 3 presents the *RMSE*s of RLLSimpute_L1, RLLSimpute_L2, RLLSimpute_EN, fLLSimpute_L1, fLLSimpute_L2, and fLLSimpute_EN. The X-axis presents the candidate values of λ, and the Y-axis gives their *RMSE*s. From Fig. 3, we can observe a general trend: *RMSE*s first decrease and then increase with the increase of λ for all the datasets. We also observe that the elastic net regularization behaves like L_1 and obtains degraded performance at a high value of λ. According to the *RMSE* curve, we choose λ for each gene expression profile.

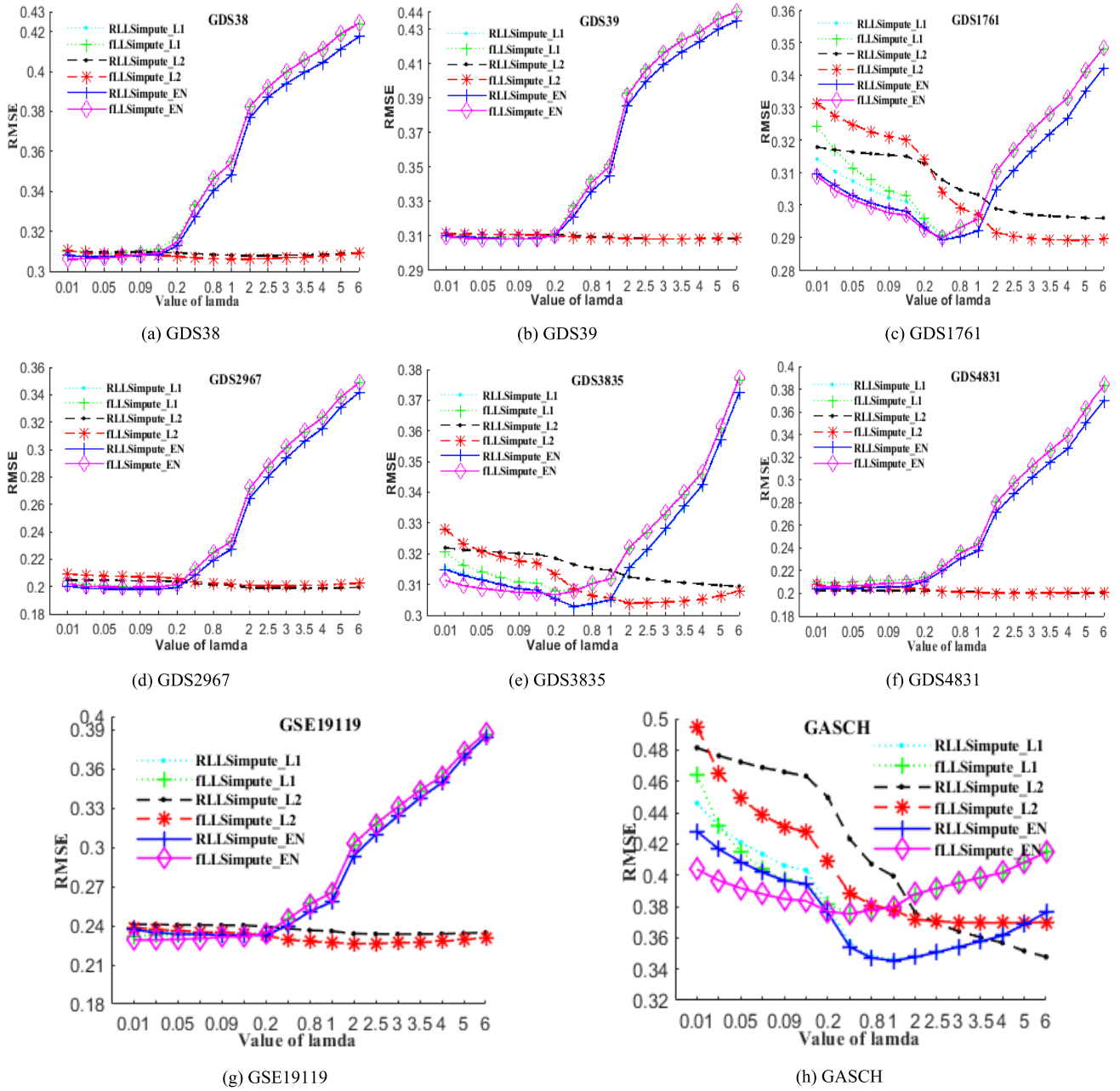
2) K-VALUE SELECTION

The number of neighbor genes largely determines the performance of local learning-based methods and is an important parameter for nearest-neighbor-based imputation methods (KNNimpute, SKNNimpute, and IKNNimpute), least squares-based imputation methods (LLSimpute, SLLSimpute, and fLLSimpute), and regularized local learning-based imputation methods (RLLSimpute_L1, RLLSimpute_L2, RLLSimpute_EN, fLLSimpute_L1, fLLSimpute_L2, and fLLSimpute_EN). We experimentally investigate the *RMSEs* with different number of neighbors. Specifically, we vary the number of neighbors from 1 to 400 and repeat each experiment ten times at a missing rate of 5% and report the averaged root mean square errors. Fig. 4 presents the *RMSE*s of 12 local learning-based methods. The X-axis shows the number of neighbors and the Y-axis gives the *RMSE*s. From Fig. 4, we observe that the *RMSE*s of nearest neighbor-based methods first decrease and then increase with the increase of k. The root of the problem is that these methods do not consider the relevance between the neighbors and thus behave poorly with the increase of k. We also observe that KNNimpute, SKNNimpute as well as IKNNimpute obtain relatively smaller *RSME*s when k ranges from 5 to 13. For least squares-based methods, the *RMSE*s increase quickly when the value of k approaches the sample size of a dataset and then gradually decrease with the increase of k. This is mainly because the solution to Eq. 5 without the regularization term is not fully optimized when the number of neighbors is set to be the number of samples. In contrast to the above methods, regularized local learning methods consistently obtain smaller *RMSE*s than these of the non-regularized methods, since they better handle the overfitting by penalizing the model complexity.

3) ROOT MEAN SQUARE ERROR

Fig. 5 presents the comparative *RMSEs* of 13 imputation algorithms on the microarray datasets. The X-axis denotes five different missing rates and corresponding *RMSE*s are given in the Y-axis. from Fig. 5, we observe that *RMSE*s tend to increase with the increase of missing rates for all the evaluated methods. This is reasonable, since a larger missing rate causes greater loss of information. We observe that the *RMSE*s of least squares-based methods and regularized local learning methods are close to each other at a small missing rate. However, regularized local learning methods generally outperform its competitors with the increase of missing rates. This is possibly because least squares-based methods suffer from over-fitting. In contrast, regularized local learning-based methods achieve a better tradeoff between avoiding overfitting and fitting the training data well. In addition, we observe that least squares-based methods generally perform better than nearest neighbor-based methods that ignore the relevance between neighbors. This indicates the superiority of local squares regression model over the nearest neighbor-based methods. Compared with the regularized local learning methods, BPCAimpute obtains larger *RMSE*s except on GDS3835 and GASCH, where they obtain similar results. This is probably because a covariance structure exists in GDS3835 and GASCH. This also supports the priority of local learning-based methods in imputing missing values.

Tables 2-3 show the results corresponding to Fig. 5 with the missing rates of 5% and 20%, respectively. For each dataset, the best result is shown in bold and the second best is underscored. The numbers in the first row correspond to the 13 algorithms shown in Fig. 5, where "1" refers to KNNimpute, "13" denotes fLLSimpute_EN, and etc. We can also observe the power of regularization techniques and robustness of elastic net across different datasets. For example, in the case of 5%, the elastic net-based methods obtain the best results on six datasets and the second best results on the left two datasets. Besides, to present a better comparison of regularized local learning methods and investigate their behaviors across datasets, Fig. 6 shows the corresponding results, where the X-axis denotes different missing rates and the Y-axis gives *RMSE*s. From Fig. 6, we observe that regularized methods generally outperform the non-regularized methods. The performance of RLLSimpute is inferior to that of RLLSimpute_L1, RLLSimpute_L2, and RLLSimpute_EN, and three methods (fLLSimpute_L1, fLLSimpute_L2 and fLLSimpute_EN) perform better than fLLSimpute. As for using the baseline criterion of selecting similar genes, RLLSimpute_EN beats RLLSimpute_L1 on eight datasets and outperforms RLLSimpute_L2 except on GDS4831. As for the filtered metric,



(a) GDS38 (b) GDS39 (c) GDS1761

(d) GDS2967 (e) GDS3835 (f) GDS4831

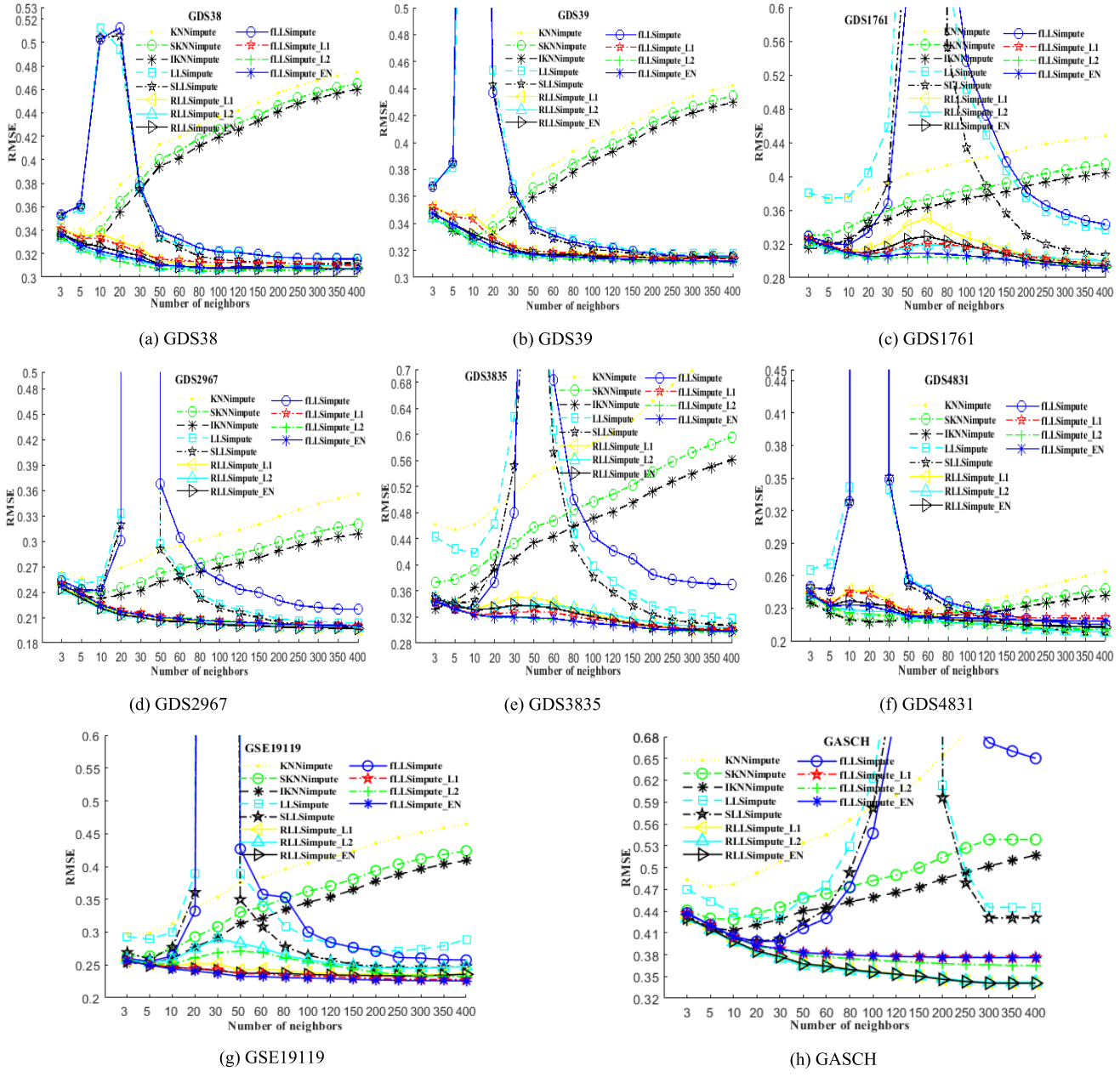
(g) GSE19119 (h) GASCH

FIGURE 3. ***RMSEs*** **of the proposed methods with different λ-values.**

fLLSimpute_L1, fLLSimpute_L2, and fLLSimpute_EN have mixed experimental results, where fLLSimpute_EN outperforms fLLSimpute_L2 on five datasets and fLLSimpute_L1 performs better than fLLSimpute_L2 on five datasets. This is mainly because the filtering criterion discards the less similar genes and the selected genes are highly correlated to each other, which is more suitable for L_1 regularization.

4) PEARSON CORRELATION COEFFICIENT

To measure the power of an estimator in recovering the original structure of a dataset, we conduct experiments on each dataset with a 5% missing rate and record the Pearson correlation coefficient. Fig. 7 shows the results. The X-axis shows the sample index and corresponding results of Pearson correlation coefficient are given in Y-axis. Due to the large number of samples of GASCH, we here only show results of the first 30 samples. From Fig. 7, we observe that RLLSimpute_EN generally achieves better performance, indicating its superiority in recovering the data structure. As for KNNimpute, SKNNimpute, and IKNNimpute, they perform worse than the other methods. For BPCAimpute, it is comparable to that of the least squares-based methods except on GDS38 and GDS39. But for the two datasets, BPCAimpute is inferior to least squares-based methods and regularized local learning methods.



(a) GDS38 (b) GDS39 (c) GDS1761

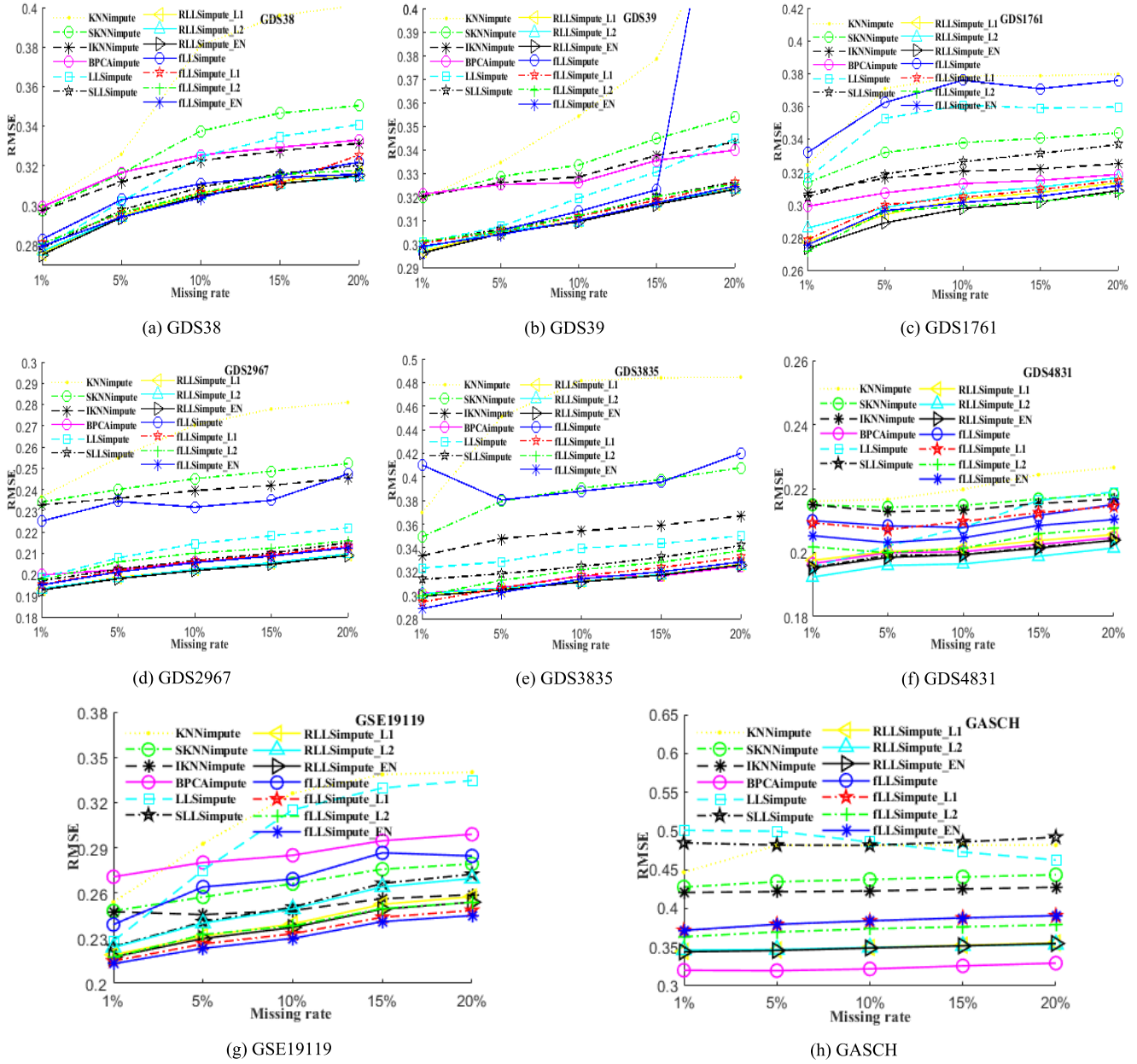
(d) GDS2967 (e) GDS3835 (f) GDS4831

(g) GSE19119 (h) GASCH

FIGURE 4. ***RMSEs*** **vs. the number of neighbors.**

TABLE 2. ***RMSEs*** **of the imputation methods on the datasets with a missing rate of 5%.**

	1	2	3	4	5	6	7	8	9	10	11	12	13
GDS38	0.3258	0.3159	0.3118	0.3164	0.3017	0.2974	0.2946	0.2939	**0.2937**	0.3027	0.2958	0.2962	0.2940
GDS39	0.3345	0.3286	0.3260	0.3252	0.3073	0.3062	0.3049	0.3052	0.3043	0.3057	0.3052	0.3054	**0.3039**
GDS2967	0.2548	0.2400	0.2358	0.2022	0.2077	0.2025	0.1985	0.1983	**0.1980**	0.2344	0.2018	0.2059	0.2011
GDS1761	0.3708	0.3318	0.3162	0.3071	0.3527	0.3185	0.2939	0.2981	**0.2889**	0.3623	0.2998	0.2956	0.2963
GDS3835	0.4511	0.3793	0.3477	0.3061	0.3281	0.3180	0.3046	0.3059	0.3042	0.3805	0.3059	0.3124	**0.3022**
GDS4831	0.2166	0.2142	0.2127	0.1999	0.2021	0.1991	0.2003	**0.1960**	0.1985	0.2084	0.2071	0.1999	0.2031
GSE19119	0.2927	0.2573	0.2456	0.2802	0.2751	0.2408	0.2313	0.2399	0.2297	0.2639	0.2262	0.2324	**0.2232**
GASCH	0.4814	0.4344	0.4215	**0.3197**	0.4996	0.4818	0.3464	0.3470	0.3457	0.7058	0.3794	0.3696	0.3795



(a) GDS38 (b) GDS39 (c) GDS1761

(d) GDS2967 (e) GDS3835 (f) GDS4831

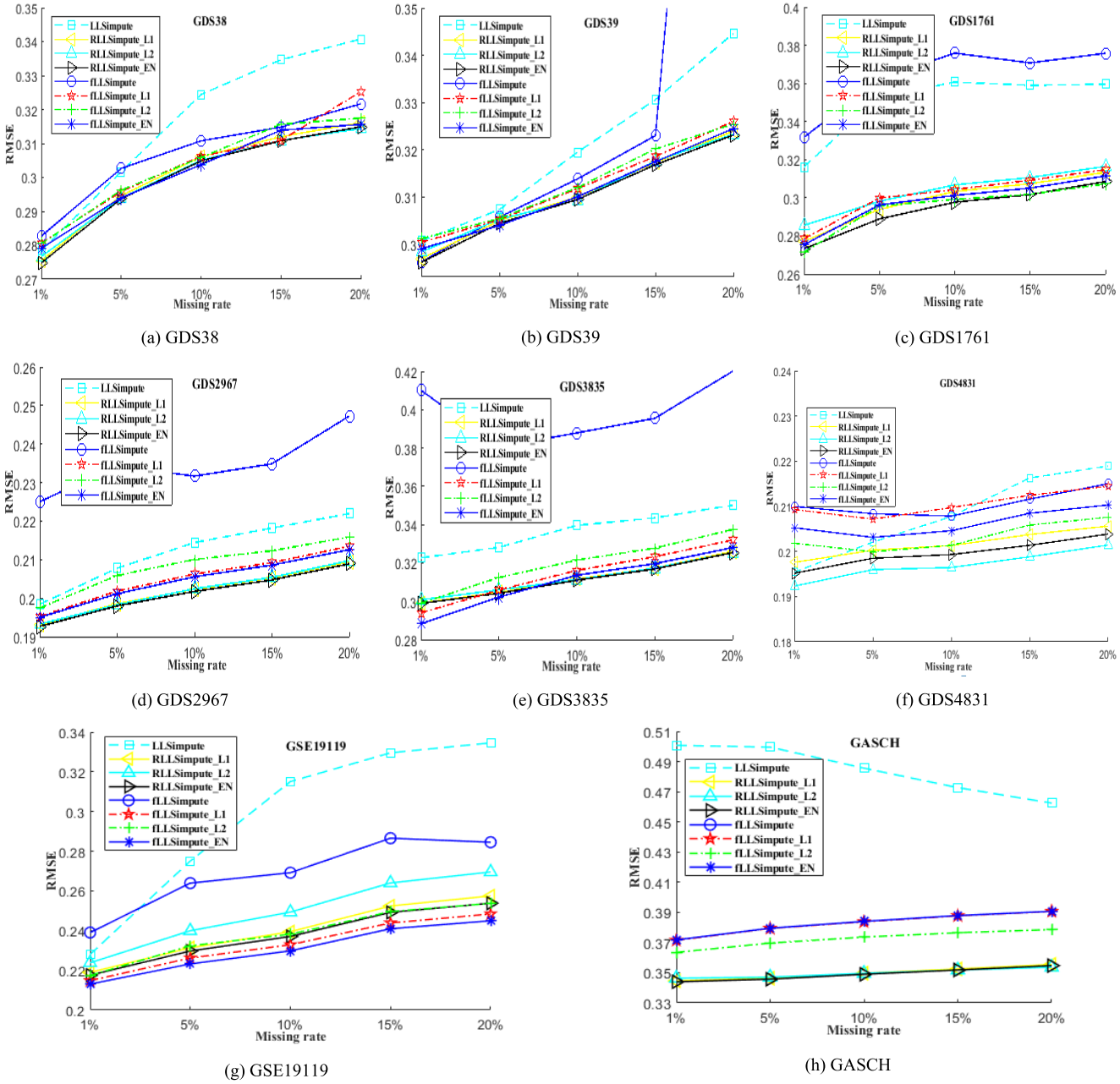
(g) GSE19119 (h) GASCH

FIGURE 5. *RMSEs* **vs. different missing rates.**

TABLE 3. *RMSEs* **of the imputation methods on the datasets with a missing rate of 20%.**

	1	2	3	4	5	6	7	8	9	10	11	12	13
GDS38	0.4011	0.3503	0.3312	0.3329	0.3408	0.3202	0.3162	**0.3144**	0.3149	0.3217	0.3253	0.3175	0.3156
GDS39	0.4440	0.3541	0.3431	0.3398	0.3446	0.3262	0.3239	0.3232	**0.3231**	0.5047	0.3260	0.3254	0.3244
GDS2967	0.2810	0.2521	0.2455	0.2129	0.2219	0.2150	0.2094	0.2097	**0.2089**	0.2473	0.2134	0.2159	0.2126
GDS1761	0.3796	0.3437	0.3249	0.3184	0.3596	0.3366	0.3134	0.3165	0.3085	0.3758	0.3147	**0.3072**	0.3115
GDS3835	0.4846	0.4078	0.3671	0.3245	0.3502	0.3418	0.3260	0.3254	0.3254	0.4200	0.3320	0.3374	0.3280
GDS4831	0.2266	0.2182	0.2167	0.2046	0.2189	0.2038	0.2056	**0.2014**	0.2038	0.2150	0.2145	0.2076	0.2103
GSE19119	0.3401	0.2792	0.2588	0.2989	0.3346	0.2724	0.2575	0.2696	0.2538	0.2845	0.2484	0.2537	**0.2450**
GASCH	0.4817	0.4434	0.4274	**0.3294**	0.4625	0.4919	0.3554	0.3536	0.3547	0.8115	0.3906	0.3786	0.3907

Furthermore, according to Figs. 5 and 7, we observe that methods with a larger *RMSE* can also better recover data structure. For example, SKNNimpute obtains a larger *PCC* than that of LLSimpute on GDS1761, however, it has a larger *RMSE*. This is because *RMSE* reflects the overall imputation performance at the dataset level, while Pearson



(a) GDS38 (b) GDS39 (c) GDS1761

(d) GDS2967 (e) GDS3835 (f) GDS4831

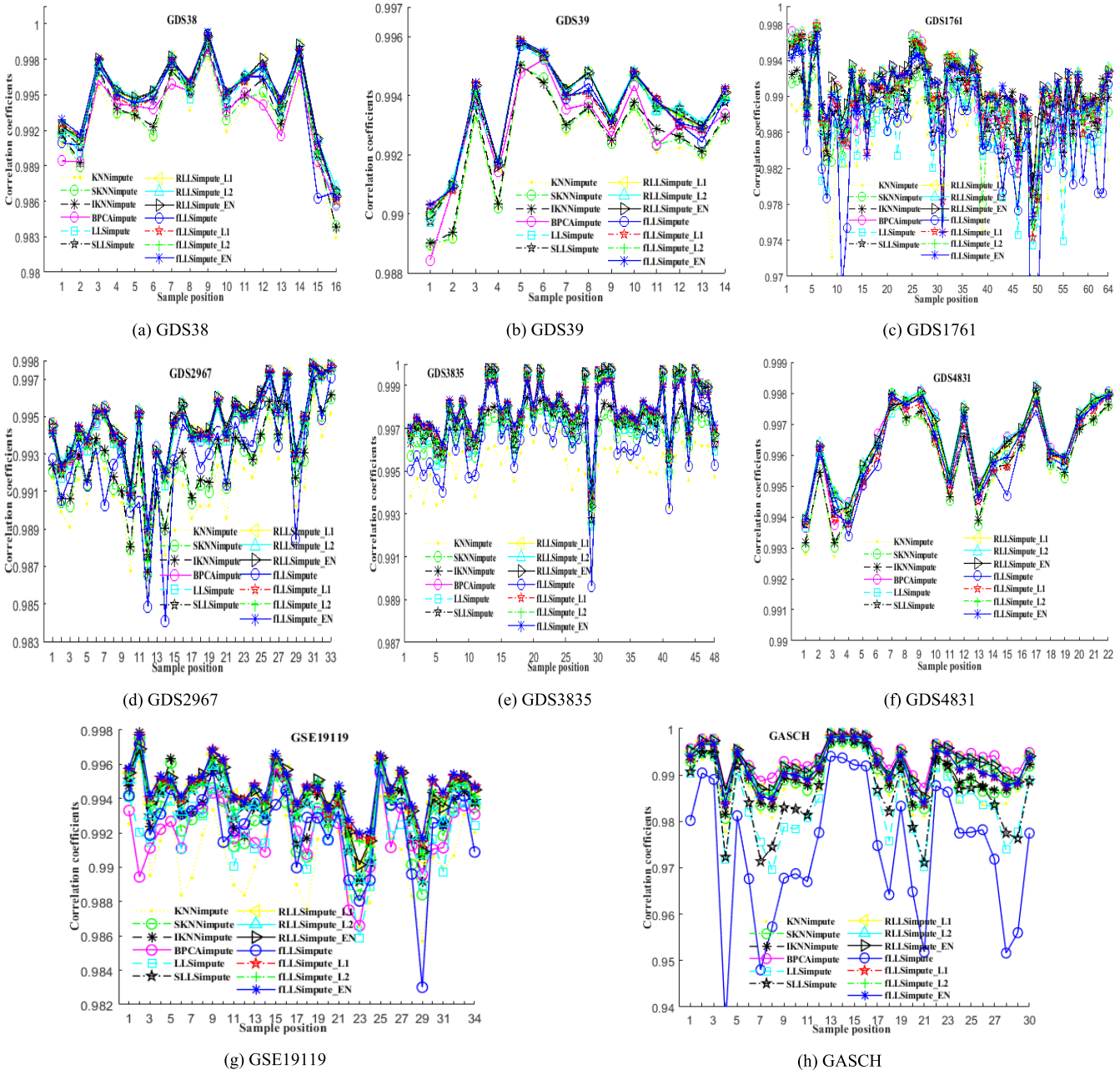
(g) GSE19119 (h) GASCH

FIGURE 6. ***RMSEs*** **of regularized local learning methods vs. different missing rates.**

correlation coefficient works at the sample level. Overall, according to the above results and analyses, we conclude that the proposed regularized local learning-based methods (i.e., RLLSimpute_L1, RLLSimpute_L2, RLLSimpute_EN, fLLSimpute_L1, fLLSimpute_L2, and fLLSimpute_EN) better estimate the missing values than its competitors, including one global learning-based method (BPCAimpute), three nearest neighbor-based methods (KNNimpute, SKNNimpute, and IKNNimpute), and three least squares-based methods (LLSimpute, SLLSimpute, and fLLSimpute) in terms of the two statistical analysis related metrics. Also, the elastic net regularized model is among the first priorities in choosing an imputation algorithm due to its robustness across different experimental datasets.

5) CONSERVED PAIRS PROPORTION

In terms of the number of returned clusters, as suggested in [42], we initially set the number K of clusters to be 500 and test whether the first 10 most important clusters represent 80% of the genes. If yes, we report K and take it as the number of clusters; otherwise, we set $K = K$-1 and repeat the above procedure until it meets the criteria. Afterwards, we apply the hierarchical clustering with Ward's linkage to partition the dataset and then calculate *CPP*. Fig. 8 presents the *CPPs* of



(a) GDS38 (b) GDS39 (c) GDS1761

(d) GDS2967 (e) GDS3835 (f) GDS4831

(g) GSE19119 (h) GASCH

FIGURE 7. Experimental results of Pearson correlation coefficients.

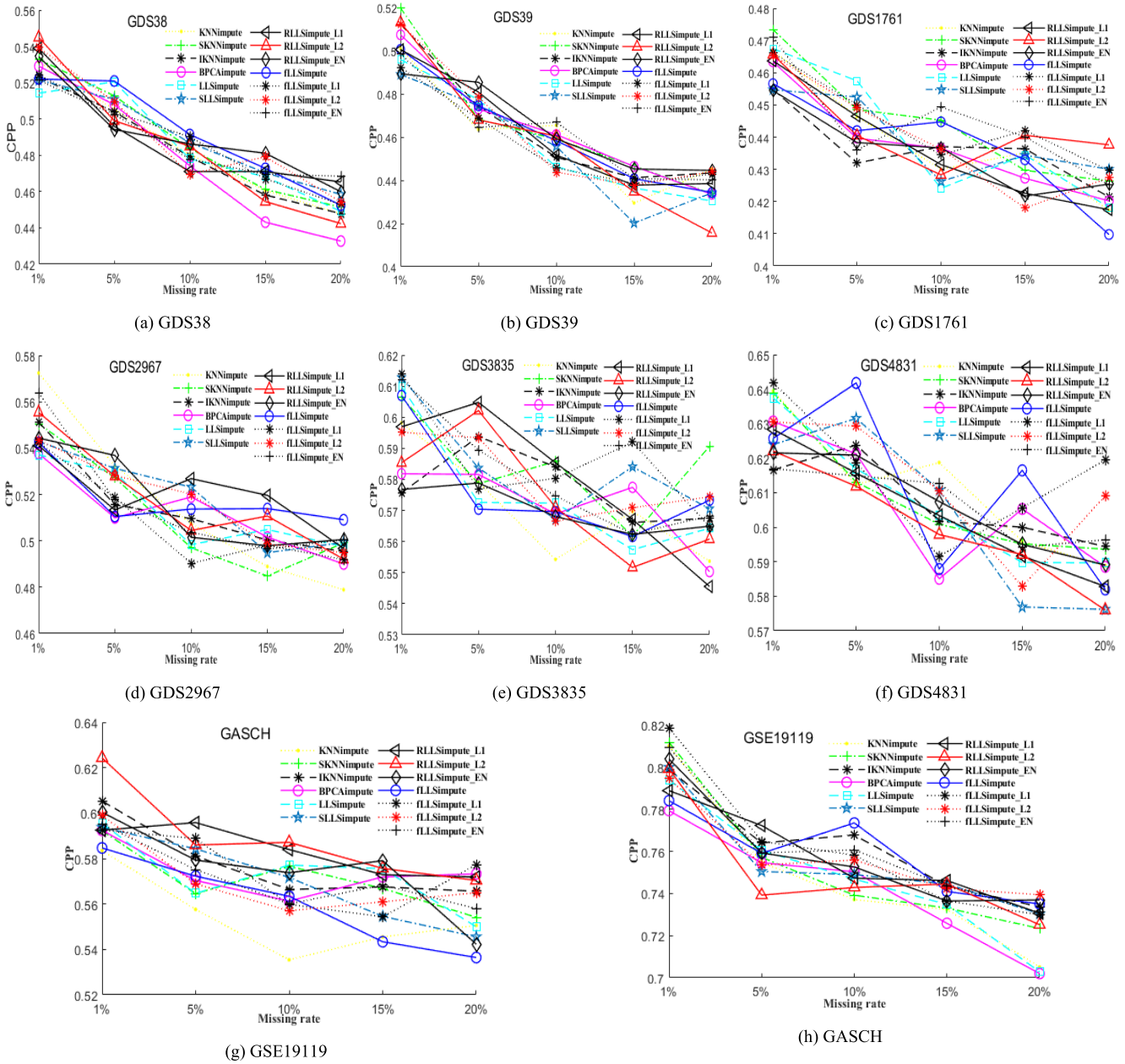
different imputation methods on the datasets under different missing rates.

From Fig. 8, we observe that missing values indeed have much influence on the stability of gene clusters and *CPP* generally decreases with the increase of missing rates. The reason is that a larger missing rate comes with much loss of information and consequently has a bigger impact on the clustering. Second, we observe that the imputation methods have mixed results on the datasets and no one dominates the others, which is consistently with previous research [42]. The possible reason is that the neighborhood relationship is easily disturbed by the estimated values, even if there is a small deviation of the estimations from their true values.

Tables 4-5 show the results corresponding to Fig. 8 with the missing rates of 5% and 20%, respectively. For each dataset, the best result is shown in bold and the second best is underscored. The numbers in the first row correspond to the 13 algorithms shown in Fig. 8, where "1" refers to KNNimpute, "13" denotes fLLSimpute_EN, and etc. From Tables 4-5, we also observe mixed results of the methods.

V. TIME COMPLEXITY ANALYSIS

Given a microarray dataset that has m samples and n genes, if k neighbors are considered, then the time complexity of KNNimpute and SKNNimpute are $O(mn^2)$ and $O(n \log n + mn \log n)$, respectively. The time complexity of IKNNimpute



(a) GDS38 (b) GDS39 (c) GDS1761

(d) GDS2967 (e) GDS3835 (f) GDS4831

(g) GSE19119 (h) GASCH

FIGURE 8. Experimental results of Conversed pairs proportion.

TABLE 4. *CPP* of the imputation methods on the datasets with a missing rate of 5%.

	1	2	3	4	5	6	7	8	9	10	11	12	13
GDS38	0.5053	0.5127	0.504	0.5079	0.5205	0.5108	0.4960	0.4991	**0.4942**	0.5209	0.5039	0.5096	0.5024
GDS39	**0.4635**	0.4683	0.4742	0.4730	0.4773	0.4745	0.4809	0.4679	0.4854	0.4745	0.4686	0.4787	0.4645
GDS2967	0.5323	0.5272	0.5156	**0.5099**	0.5294	0.5313	0.5133	0.5279	0.5369	0.5105	0.5186	0.5281	0.5177
GDS1761	0.4463	0.4486	**0.4320**	0.4397	0.4575	0.4524	0.4465	0.4405	0.4383	0.4420	0.4500	0.4491	0.4360
GDS3835	0.5791	0.5786	0.5938	0.5816	0.5726	0.5836	0.6048	0.6021	0.5788	**0.5704**	0.5769	0.5934	0.5894
GDS4831	0.6124	0.6131	0.6238	0.6212	0.6169	0.6315	0.6157	**0.6119**	0.6209	0.6418	0.6187	0.6293	0.6149
GSE19119	0.7547	0.7571	0.7639	0.7546	0.7605	0.7504	0.7722	**0.7392**	0.7592	0.7591	0.7647	0.7535	0.7592
GASCH	**0.5576**	0.5645	0.5811	0.5703	0.5647	0.5842	0.5958	0.5860	0.5793	0.5722	0.5888	0.5688	0.5751

is $O(mn + imn \log n)$, where i specifies the number of iterations. The time complexity of BPCAimpute is $O(n(m^2n + mn(m-1))) = O(m^2n^2)$. Least squares based methods take $O(k^3)$ to find the inverse of a $k * k$ matrix, so the time complexity of LLSimpute, SLLSimpute, and fLLSimpute are $O(n(mn + k^3))$, $O(n \log n + n(mn + k^3))$, and $O(n(mn + k^3))$,

TABLE 5. *CPP* of the imputation methods on the datasets with a missing rate of 20%.

	1	2	3	4	5	6	7	8	9	10	11	12	13
GDS38	0.4477	0.4514	0.4479	**0.4327**	0.4491	0.4584	0.4653	0.4424	0.4599	0.4525	0.4529	0.4543	0.4685
GDS39	0.4452	0.4338	0.4434	0.4335	0.4304	0.4342	0.4385	**0.4156**	0.4446	0.4344	0.4429	0.4441	0.4402
GDS2967	**0.4789**	0.4991	0.4954	0.4900	0.4975	0.4993	0.4967	0.4921	0.5006	0.5091	0.4919	0.4948	0.5004
GDS1761	0.4173	0.4264	0.4214	0.4202	0.4181	0.4300	0.4174	0.4376	0.4254	**0.4098**	0.4296	0.4274	0.4251
GDS3835	0.5537	0.5907	0.5675	0.5503	0.5643	0.5705	**0.5455**	0.5608	0.5649	0.5733	0.5648	0.5744	0.5683
GDS4831	0.5888	0.5936	0.5944	0.5885	0.5897	0.5762	0.5828	**0.5760**	0.5891	0.5820	0.6195	0.6091	0.5964
GSE19119	0.7051	0.7234	0.7336	**0.7020**	0.7028	0.7303	0.7306	0.7252	0.7369	0.7350	0.7299	0.7394	0.7310
GASCH	0.5505	0.5541	0.5656	0.5732	0.5499	0.5456	0.5718	0.5704	0.5423	**0.5364**	0.5773	0.5649	0.5580

respectively. For regularized local learning-based methods, the time complexity of lasso, ridge regression and elastic net regression is $O(k^3 + k^2 n)$. Hence, RLLSimpute_L1, RLLSimpute_L2, and RLLSimpute_EN have the time complexity of $O(n(k^3 + k^2 m + mn) + n \log n)$, where $O(n \log n)$ is the time complexity for sorting. The time complexity of filtering out similar genes is $O(n)$, so the time complexity of fLLSimpute_L1, fLLSimpute_L2, and fLLSimpute_EN are $O(n(k^3 + k^2 m + mn + n) + n \log n)$. Accordingly, we observe that nearest neighbor-based methods have a lower time complexity than least squares-based methods and that regularized local learning-based methods have slightly higher time costs than least squares-based methods.

VI. CONCLUSION

Accurately estimating the missing values of microarray data plays a crucial role in fully utilizing a collection of gene expression profiles and facilitating downstream analyses, therefore it remains a challenging yet rewarding research topic. In this study, we develop a regularized local learning framework that aims to better utilize the local structure of microarray data. After detailing the key components of the framework, we analyze three different regularization terms. Motivated by the simultaneous variable selection and grouping effect of elastic net penalty, we design an elastic net regularized local least squares-based imputation method, named RLLSimpute_EN, to estimate the missing entries of a target gene with its neighbors. Besides, we integrate a new filtering similarity metric into the framework and accordingly propose another four imputation methods (i.e., fLLSimpute, fLLSimpute_L1, fLLSimpute_L2, and fLLSimpute_EN). To reuse previously estimated values, the proposed methods work in ascending order of the missing rates. Extensive comparative experiments against other eight imputation methods are conducted on eight gene expression profiles. Results indicate the power of sparse regularization techniques in mitigating overfitting and the superiority of elastic net penalty in imputing the missing values. Finally, theoretical time complexity analysis shows its efficiency.

To further advance relevant researches, we plan to work along the following lines. First, the biology knowledge, if available, can provide valuable information of the genes, we will then explore domain knowledge-oriented missing value imputation methods by analyzing global and local structural information, sample relevance, and gene semantic knowledge. Second, the proposed framework and methods provide a way to handle missing values and have potential use in other topics such as clinical and sensor data analysis. Third, accurately evaluating the influence of missing values on downstream tasks such as microarray data clustering and interpretation remains another interesting topic [44].

REFERENCES

[1] J. E. Mirus, Y. Zhang, C. I. Li, A. E. Lokshin, R. L. Prentice, S. R. Hingorani, and P. D. Lampe, "Cross-species antibody microarray interrogation identifies a 3-Protein panel of plasma biomarkers for early diagnosis of pancreas cancer," *Clin. Cancer Res.*, vol. 21, no. 7, pp. 1764–1771, Apr. 2015.

[2] W. Ke, C. Wu, Y. Wu, and N. N. Xiong, "A new filter feature selection based on criteria fusion for gene microarray data," *IEEE Access*, vol. 6, pp. 61065–61076, Oct. 2018.

[3] A. Wang, N. An, G. Chen, L. Liu, and G. Alterovitz, "Subtype dependent biomarker identification and tumor classification from gene expression profiles," *Knowl.-Based Syst.*, vol. 146, pp. 104–117, Apr. 2018.

[4] Z. Bao, Y. Zhu, Q. Ge, W. Gu, X. Dong, and Y. Bai, "GwSPIA: Improved signaling pathway impact analysis with gene weights," *IEEE Access*, vol. 7, pp. 69172–69183, May 2019.

[5] O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R. B. Altman, "Missing value estimation methods for DNA microarrays," *Bioinformatics*, vol. 17, no. 6, pp. 520–525, Jun. 2001.

[6] Y. Chen, A. Wang, H. Ding, X. Que, Y. Li, N. An, and L. Jiang, "A global learning with local preservation method for microarray data imputation," *Comput. Biol. Med.*, vol. 77, pp. 76–89, Oct. 2016.

[7] M. Lenz, F.-J. Müller, M. Zenke, and A. Schuppert, "Principal components analysis and the reported low intrinsic dimensionality of gene expression microarray data," *Sci. Rep.*, vol. 6, no. 1, p. 25696, Jun. 2016.

[8] R. Priya and R. Sivaraj, "Pre-processing of microarray gene expression data for classification using adaptive feature selection and imputation of non-ignorable missing values," *Int. J. Data Mining Bioinf.*, vol. 16, no. 3, pp. 183–204, Dec. 2016.

[9] M. C. de Souto, P. A. Jaskowiak, and I. G. Costa, "Impact of missing data imputation methods on gene expression clustering and classification," *BMC Bioinf.*, vol. 16, no. 1, p. 64, Dec. 2015.

[10] A. W.-C. Liew, N.-F. Law, and H. Yan, "Missing value imputation for gene expression data: Computational techniques to recover missing data from available information," *Briefings Bioinf.*, vol. 12, no. 5, pp. 498–513, Dec. 2010.

[11] K. Moorthy, M. Mohamad, and S. Deris, "A review on missing value imputation algorithms for microarray gene expression data," *Current Bioinf.*, vol. 9, no. 1, pp. 18–22, Jan. 2014.

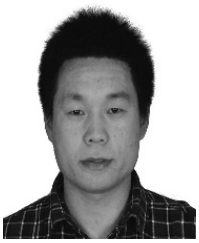
[12] Y. Yang, Z. Xu, and D. Song, "Missing value imputation for microRNA expression data by using a GO-based similarity measure," *BMC Bioinf.*, vol. 17, no. S1, p. S10, Dec. 2016.

[13] Q. Xiang, X. Dai, Y. Deng, C. He, J. Wang, J. Feng, and Z. Dai, "Missing value imputation for microarray gene expression data using histone acetylation information," *BMC Bioinf.*, vol. 9, no. 1, p. 252, Dec. 2008.

[14] S. Oba, M.-A. Sato, I. Takemasa, M. Monden, K.-I. Matsubara, and S. Ishii, "A Bayesian missing value estimation method for gene expression profile data," *Bioinformatics*, vol. 19, no. 16, pp. 2088–2096, Nov. 2003.

[15] O. Alter, P. O. Brown, and D. Botstein, "Singular value decomposition for genome-wide expression data processing and modeling," *Proc. Nat. Acad. Sci. USA*, vol. 97, no. 18, pp. 10101–10106, Aug. 2000.

[16] T. Yu, H. Peng, and W. Sun, "Incorporating nonlinear relationships in microarray missing value imputation," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 8, no. 3, pp. 723–731, May 2011.
[17] T. H. Bo, "LSimpute: Accurate estimation of missing values in microarray data with least squares methods," *Nucleic Acids Res.*, vol. 32, no. 3, p. 34e, Feb. 2004.
[18] H. Kim, G. H. Golub, and H. Park, "Missing value estimation for DNA microarray gene expression data: Local least squares imputation," *Bioinformatics*, vol. 21, no. 2, pp. 187–198, Aug. 2004.
[19] R. Jörnsten, H.-Y. Wang, W. J. Welsh, and M. Ouyang, "DNA microarray data imputation and significance analysis of differential expression," *Bioinformatics*, vol. 21, no. 22, pp. 4155–4161, Aug. 2005.
[20] C. He, H.-H. Li, C. Zhao, G.-Z. Li, and W. Zhang, "Triple imputation for microarray missing value estimation," in *Proc. IEEE Int. Conf. Bioinf. Biomed. (BIBM)*, Washington, DC, USA, Nov. 2015, pp. 208–213.
[21] F. Shi, D. Zhang, J. Chen, and H. R. Karimi, "Missing value estimation for microarray data by Bayesian principal component analysis and iterative local least squares," *Math. Problems Eng.*, vol. 2013, Mar. 2013, Art. no. 162938.
[22] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *J. Roy. Stat. Soc., B (Stat. Methodol.)*, vol. 67, no. 2, pp. 301–320, Apr. 2005.
[23] H. Li, C. Zhao, F. Shao, G.-Z. Li, and X. Wang, "A hybrid imputation approach for microarray missing value estimation," *BMC Genomics*, vol. 16, no. S9, pp. 1–11, Dec. 2015.
[24] F. Meng, C. Cai, and H. Yan, "A bicluster-based Bayesian principal component analysis method for microarray missing value estimation," *IEEE J. Biomed. Health Informat.*, vol. 18, no. 3, pp. 863–871, May 2014.
[25] S. K. Pati and A. K. Das, "Missing value estimation for microarray data through cluster analysis," *Knowl. Inf. Syst.*, vol. 52, no. 3, pp. 709–750, Sep. 2017.
[26] M. Ouyang, W. J. Welsh, and P. Georgopoulos, "Gaussian mixture clustering and imputation of microarray data," *Bioinformatics*, vol. 20, no. 6, pp. 917–923, Jan. 2004.
[27] P. Keerin, W. Kurutach, and T. Boongoen, "A hybrid imputation approach for microarray missing value estimation," *Int. J. Data Min. Bioin.*, vol. 15, no. 2, pp. 165–193, May 2016.
[28] S. Chattopadhyay, C. Das, and S. Bose, "A novel biclustering based missing value prediction method for microarray gene expression data," in *Proc. Int. Conf. Man Mach. Interfacing (MAMI)*, Bhubaneswar, India, Dec. 2015, pp. 1–6.
[29] K. Kim, B. Kim, and G. Yi, "Reuse of imputed data in microarray analysis increases imputation efficiency," *BMC Bioinf.*, vol. 5, no. 1, p. 160, Dec. 2004.
[30] L. P. Brás and J. C. Menezes, "Improving cluster-based missing value estimation of DNA microarray data," *Biomol. Eng.*, vol. 24, no. 2, pp. 273–282, Jun. 2007.
[31] X. Zhang, X. Song, H. Wang, and H. Zhang, "Sequential local least squares imputation estimating missing value of microarray data," *Comput. Biol. Med.*, vol. 38, no. 10, pp. 1112–1120, Oct. 2008.
[32] H. Wang, C.-C. Chiu, Y.-C. Wu, and W.-S. Wu, "Shrinkage regression-based methods for microarray missing value imputation," *BMC Syst. Biol.*, vol. 7, no. 6, p. S11, 2013.
[33] A. Wang, Y. Chen, N. An, J. Yang, L. Li, and L. Jiang, "Microarray missing value imputation: A regularized local learning method," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 16, no. 3, pp. 980–993, May 2019.
[34] T. Chai and R. R. Draxler, "Root mean square error (RMSE) or mean absolute error (MAE)?" *Geosci. Model Develop. Discuss.*, vol. 7, no. 1, pp. 1525–1534, Feb. 2014.
[35] C. Truntzer, C. Mercier, J. Estève, C. Gautier, and P. Roy, "Importance of data structure in comparing two dimension reduction methods for classification of microarray gene expression data," *BMC Bioinf.*, vol. 8, no. 1, p. 90, Dec. 2007.
[36] P. Spellman, G. Sherlock, M. Zhang, V. Iyer, K. Anders, M. Eisen, P. Brown, D. Botstein, and B. Futcher, "Comprehensive identification of cell cycle–regulated genes of the yeast saccharomyces cerevisiae by microarray hybridization," *Mol. Biol. cell*, vol. 9, no. 12, pp. 3273–3297, Dec. 1998.
[37] O. Sarig. *Bar1-Deficient Mating Type a Cells Response to Alpha Mating Factor: Time Course and Dose Response*. Accessed: Jun. 5, 2018. [Online]. Available: https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE8982
[38] D. Ross, U. Scherf, M. Eisen, C. Perou, C. Rees, P. Spellman, V. Iyer, S. S. Jeffrey, M. V. de Rijn, M. Waltham, A. Pergamenschikov, J. Lee, D. Lashkari, D. Shalon, T. G. Myers, J. N. Weinstein, D. Botstein, and P. O. Brown, "Systematic variation in gene expression patterns in human cancer cell lines," *Nature Genet.*, vol. 24, no. 3, pp. 227–235, Mar. 2000.
[39] C. G. Artieri and R. S. Singh, "Molecular evidence for increased regulatory conservation during metamorphosis, and against deleterious cascading effects of hybrid breakdown in drosophila," *BMC Biol.*, vol. 8, no. 1, p. 26, Dec. 2010.
[40] Y. Lee, J. Andersen, H. Song, A. Judge, D. Seo, T. Ishikawa, J. Marquardt, M. Kitade, M. E. Durkin, C. Raggi, H. Woo, E. Conner, I. Avital, I. MacLachlan, V. Factor, and S. Thorgeirsson, "Definition of ubiquitination modulator COP1 as a novel therapeutic target in human hepatocellular carcinoma," *Cancer Res.*, vol. 70, no. 21, pp. 8264–8269, Nov. 2010.
[41] A. P. Gasch, M. Huang, S. Metzner, D. Botstein, S. J. Elledge, and P. O. Brown, "Genomic expression responses to DNA-damaging agents and the regulatory role of the yeast ATR homolog Mec1p," *Mol. Biol. Cell*, vol. 12, no. 10, pp. 2987–3003, Oct. 2001.
[42] A. Brevern, S. Hazout, and A. Malpertuy, "Influence of microarrays experiments missing values on the stability of gene groups by hierarchical clustering," *BMC Bioinf.*, vol. 5, no. 1, p. 114, Aug. 2004.
[43] J. Liu, S. Ji, and J. Ye, *SLEP: Sparse Learning With Efficient Projections*. Tempe, AZ, USA: Arizona State Univ., 2009.
[44] M. Celton, A. Malpertuy, G. Lelandais, and A. G. de Brevern, "Comparative analysis of missing value imputation methods to improve clustering and interpretation of microarray experiments," *BMC Genomics*, vol. 11, no. 1, p. 15, 2010.



AIGUO WANG received the B.S. degree and the Ph.D. degree in computer science from the Hefei University of Technology, China, in 2010 and 2015, respectively. He is currently a Distinguished Research Fellow with the School of Electronic Information Engineering, Foshan University, Foshan, China. His research interests include machine learning, data mining, pervasive computing, and bioinformatics.

JING YANG received the B.S. and Ph.D. degrees from the Hefei University of Technology, China, in 2004 and 2013, respectively. She is currently an Associate Professor with the School of Computer and Information, Hefei University of Technology, China. Her current research interests include bioinformatics, artificial intelligence, data mining, and Bayesian networks.

NING AN (Senior Member, IEEE) received the B.S. degree in computer science from Lanzhou University, China, in 1993, and the Ph.D. degree in computer science and engineering from Pennsylvania State University, in 2002. From 2002 to 2012 , he was a Technical Staff with Oracle USA. Since 2012, he has been a Researcher with the Hefei University of Technology. He is currently a Yellow Mountain Professor with the School of Computer and Information, Hefei University of Technology. His research interests include gerontechnology, data mining, and mobile health technologies.