基于同态加密的 DBSCAN 聚类隐私保护方案

贾春福[1,2]，李瑞琪[1,2]，王雅飞[1,2]

（1. 南开大学网络空间安全学院，天津 300350；2. 天津市网络与数据安全技术重点实验室，天津 300350）

摘 要：为了降低数据外包聚类运算过程中存在的隐私泄露风险，提出了一个基于同态加密的 DBSCAN 聚类隐私保护方案。为了加密实际场景中的浮点型数据，给出了针对不同数据精度的 3 种数据预处理方式，并提出了一种基于数据特点且综合考虑数据精度和计算开销等方面的数据预处理方式的选择策略。由于同态加密不支持密文比较运算，设计了一个用户端与云服务器之间的协议实现密文比较功能。理论分析和实验结果表明，所提方案能够保证数据隐私安全，并且具有较高的聚类准确率和较低的时间开销。

关键词：隐私保护；密度聚类；同态加密；数据预处理；密文比较

中图分类号：TP309.2

文献标识码：A

DOI: 10.11959/j.issn.1000−436x.2021026

Privacy protection scheme of DBSCAN clustering based on homomorphic encryption

JIA Chunfu[1,2], LI Ruiqi[1,2], WANG Yafei[1,2]

1. College of Cyber Science, Nankai University, Tianjin 300350, China
2. Tianjin Key Laboratory of Network and Data Security Technology, Tianjin 300350, China

Abstract: In order to reduce the risk of data privacy leakage in the process of outsourced clustering, a privacy protection scheme of DBSCAN clustering based on homomorphic encryption was proposed. In order to encrypt the float data in the actual scene, three data preprocessing methods for different data accuracy were given, and a policy for choosing a proper data preprocessing method based on data characteristics, accuracy and computational cost was also proposed. For the ciphertext comparison operation that was not supported by homomorphic encryption, a protocol between the client and the cloud server was designed to realize the function of ciphertext comparison. Theoretical analysis and experimental results show that the proposed scheme can ensure the security of data privacy, and has a higher clustering accuracy rate and lower time overhead.

Keywords: privacy protection, density clustering, homomorphic encryption, data pre-processing, comparison on encrypted data

1 引言

云服务的快速发展，为用户提供了更加多元化的数据处理方式。用户可以选择将数据集上传到云服务器，由云服务器执行相关操作，从而减少用户计算资源的使用。由于用户种类繁多，其提供的数据集也将涵盖多种类型，比如电子邮件信息、后台用户数据库信息、个人健康状况等涉及个人隐私或企业商业机密的信息。用户将上述信息进行外包处理时，其中包含的敏感信息存在着泄露的风险。因

收稿日期：2020−09−28；**修回日期**：2020−12−27

基金项目：国家重点研发计划基金资助项目（No.2018YFA0704703）；国家自然科学基金资助项目（No.61972215, No.61702399, No.61972073）；天津市自然科学基金资助项目（No.20JCZDJC00640）

Foundation Items: The National Key Research and Development Program of China (No.2018YFA0704703), The National Natural Science Foundation of China (No.61972215, No.61702399, No.61972073), The Natural Science Foundation of Tianjin (No.20JCZDJC00640)

此，如何对敏感数据所包含的隐私进行保护是当前的研究热点。

隐私保护常见的处理方式主要有 2 种：一种是通过硬件或可信第三方提供的可信信道来保证传输过程中的数据不被窃取；另一种是将数据加密后进行传输，即使攻击者获得密文，没有密钥也无法进行解密。第一种方式虽然可以保证传输过程中的安全性，但其对第三方有较大依赖；并且，由于其使用明文传输，当数据集传输到云服务器端后，仍然存在着被窃取的风险。第二种方式将数据集加密后传输，这样无论在传输过程中还是在服务器端计算过程中都能够保证数据明文不会被泄露。

对数据集进行加密时，如果使用传统的加密算法，服务器无法直接对密态数据进行处理，需要用户向服务器提供密钥或执行解密操作。同态加密（HE, homomorphic encryption）是一种新型密码学工具，其支持在加密信息上进行任意函数运算，并且解密后得到的结果与在明文上执行相应运算的结果一致。同态加密算法依照所支持的运算种类和次数不同，可大致分为以下几类：支持无限次数和多种运算的全同态加密（FHE, fully homomorphic encryption）算法、支持无限次数和有限种类的部分同态加密（PHE, partial homomorphic encryption）算法、支持有限次数和多种运算的浅同态加密（SWHE, somewhat homomorphic encryption）算法[1-4]。大部分应用场景只需要有限次同态加法或乘法操作，并且 SWHE 算法相较于 FHE 算法而言具有更好的效率，因此 SWHE 算法具有更加广泛的应用场景[5]。BGV（Brakerski-Gentry-Vaikuntanathan）方案是一种常用的 SWHE 算法，支持加密多项式或整数，同时也可以利用中国剩余定理实现并行化操作。

同态加密同时满足保密性和密文可操作性的特点使其在隐私保护领域有较广泛的应用。基于同态加密的外包计算的隐私保护模型如图 1 所示。用户对需要外包的数据集进行加密，并上传到服务器端（①），由服务器进行较高计算复杂度的数据处理；然后，服务器将结果返回给用户（②）。

机器学习是一种重要的数据外包计算，其隐私保护问题是同态加密的一类重要应用场景。在过去几年中，研究者针对不同的应用场景、不同的数据类型以及不同的机器学习算法，开展了许多同态加密在机器学习隐私保护中应用的工作。文献[6]实现了高效且安全的 k 近邻（kNN，k-nearest neighbor）算法。文献[7-8]实现了加密数据集上的同态 K-means 聚类算法，但其存在时间开销较大的问题。文献[9-11]的加密对象是图片，其对图片型数据集执行加密操作后，在密态图片上提取特征，并应用不同的图像匹配算法，实现密态图片的匹配。文献[12]实现了密态数据集上的统计学习算法。文献[13-14]通过逐比特运算的方式实现了加密数据集上的比较算法或协议，但其开销较大。文献[15]预先在数字型数据集上提取相关特征，然后将特征加密后上传到云端，在云端对相关特征的密文进行同态运算后，完成分类运算。本文提出的方案与之不同，加密对象为原数据集，而后在云端执行同态聚类操作。

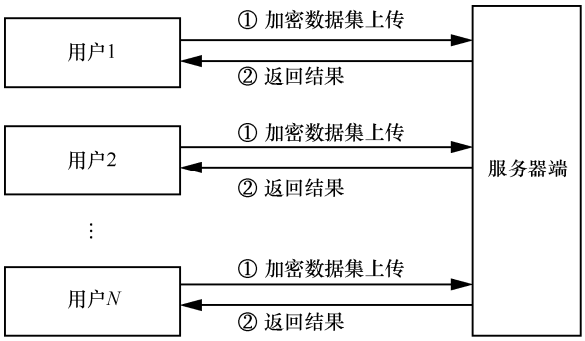


图 1 基于同态加密的外包计算的隐私保护模型

常见的机器学习算法主要分为有监督型和无监督型。有监督型机器学习算法能够提取训练数据集的特征和真实标签以构建模型，并对测试数据进行测试从而得出相应结果，代表算法有贝叶斯分类器、超平面分类器、决策树分类器等；无监督型机器学习算法不需要预先训练模型，代表算法主要有 K-means 聚类算法、DBSCAN（density-based spatial clustering of application with noise）算法等。DBSCAN 是一种基于密度的无监督型机器学习聚类算法，能够在有噪声点的情况下找到任意形状的聚类簇，相较于另一个常见的聚类算法 K-means，其应用范围更加广泛，如可用于推荐系统等高级系统的实现。DBSCAN 还可用于与加速范围访问（如 R*−树）相配合的一种数据库结构的设计。因此，DBSCAN 隐私保护问题的研究具有重要意义。

本文提出了一个基于同态加密的聚类学习的隐私保护方案，实现了在密文数据集上进行同态 DBSCAN 的聚类操作。因为 BGV 同态加密算法计算效率较高、明文空间选择灵活，所以本文选择其对数据进行加密。为了解决 BGV 算法不能直接加

密浮点型数据的问题，本文提出了 3 种不同的浮点型数据预处理方式，根据浮点型数据集自身特点、综合考虑计算开销和数据精度进行选择。对于同态加密不支持的密文比较运算，本文设计了一个云服务器与用户之间的协议实现密文的比较操作，协议只涉及同态密文之间的加减法操作，计算复杂度较低。

2　基础知识

2.1　符号介绍

对于一个实数 z，$\lceil z \rceil$、$\lfloor z \rfloor$和$\lceil z \rfloor$分别表示对 z 向上、向下和就近取整；$[z]_m$ 表示 $z \bmod m$。使用小写粗体字母表示向量；大写字母 $X=\{x_1,x_2,\cdots,x_i\}$ 表示集合，$|X|$表示集合中元素的个数；大写粗体字母表示矩阵（如 $\boldsymbol{A}$），$\boldsymbol{A}^{\mathrm{T}}$ 表示矩阵的转置。符号 $\leftarrow$ 表示随机选取；R 表示环，$R_q=R/qR$。对于明文空间中的元素 $m\in\mathcal{M}$，符号 $[\![m]\!]$ 表示其密文。

2.2　同态加密算法

定义 1 同态加密。同态加密算法主要包含 4 个部分：密钥生成（KeyGen）、加密（Enc）、密文运算（Eval）、解密（Dec）。

① $(\mathrm{pk},\mathrm{sk})\leftarrow \mathrm{KeyGen}(\lambda)$：输入安全参数 λ，输出公钥 pk 和私钥 sk。

② $c\leftarrow \mathrm{Enc}(\mathrm{pk},m)$：输入消息明文 $m\in\mathcal{M}$ 和公钥 pk，输出密文 c。

③ $m\leftarrow \mathrm{Dec}(\mathrm{sk},c)$：输入密文 c 和私钥 sk，输出消息明文 m。

④ $c_{\mathrm{Eval}}\leftarrow \mathrm{Eval}(\mathcal{C},\{c_1,\cdots,c_k\},\mathrm{pk})$：输入一个布尔电路 $\mathcal{C}$、一组密文 $\{c_1,\cdots,c_k\}$ 和公钥 pk，输出结果密文 c_{Eval}。

本文选取文献[2]提出的 BGV 方案，并借助于同态加密算法库 HElib 完成方案的实现。BGV 方案包括以下几个算法。

$\mathrm{Setup}(1^{\lambda},1^{\mu})$。随机选取 μ bit 的模数 q，选取 $n=n(\lambda)$，$N=N(\lambda)$，$\chi=\chi(\lambda)$，$t=t(\lambda)$，$d=d(\lambda)$。令 $R=\mathbb{Z}[x]/f(x)$，其中 $f(x)$ 是 n 次分圆多项式。上述参数的选取应保证方案的困难性能够基于格困难问题 GLWE（general learning with error），并能够抵御现有攻击。

$\mathrm{KeyGen}(\lambda)$。选取 $\boldsymbol{s}\leftarrow\chi^d$，令私钥为 $\mathrm{sk}=(1,\boldsymbol{s})\in R_q^{d+1}$。随机抽取矩阵 $\boldsymbol{A}\leftarrow R_q^{Nd}$，向量 $\boldsymbol{e}\leftarrow\chi^N$，计算 $\boldsymbol{b}\leftarrow\boldsymbol{A}\boldsymbol{s}+t\boldsymbol{e}$，令公钥为 $\boldsymbol{B}=[\boldsymbol{b},-\boldsymbol{A}]$。显然，$\boldsymbol{B}\boldsymbol{s}=t\boldsymbol{e}$。

$\mathrm{Enc}(\mathrm{pk},m)$。将明文 $m\in R_t$ 扩展成 $\boldsymbol{m}\leftarrow(m,0,\cdots,0)\in R_q^{d+1}$，随机选取 $\boldsymbol{r}\leftarrow R_2^N$，输出密文 $\boldsymbol{c}=\boldsymbol{m}+\boldsymbol{B}^{\mathrm{T}}\boldsymbol{r}\in R_q^{d+1}$。

$\mathrm{Dec}(\mathrm{sk},c)$。输出 $m=\left[\left[\langle \boldsymbol{c},\boldsymbol{s}\rangle\right]_q\right]_t$。

$\mathrm{Eval}(c_1,c_2)$。密文加法输出 $c_1\oplus c_2$，密文乘法输出 $c_1\otimes c_2$。

HElib 中除了上述算法之外还包括一些用于实现 FHE 算法的处理程序[16]，例如 key switching、modulus switching 和 bootstrapping。FHE 算法存在密钥过长、密文量级过大、降噪处理耗时较长等弊端，在实际应用中具有一定的局限性；SWHE 算法虽然不支持无限次密文运算，但其密文和密钥的尺寸较小，也不需要 bootstrapping 技术对密文进行刷新，因而运行效率较高。由于本文的场景只需有限次密文运算，因此选择使用 SWHE 算法，以获得较高的计算效率。

2.3　数据预处理算法

大多数实际应用中的数据都不能够直接作为加密算法的明文，需要一定的预处理算法将实际数据映射到明文空间。本文所研究的 DBSCAN 算法需处理的数据是浮点型数据集，所以需要使用预处理算法将其映射至加密算法的明文空间。本节将介绍 3 种数据预处理方式。

方式 1　整数对编码

设原数据为浮点数 c，选取整数 a 和 b 使 $b=\lceil a\times c\rfloor$，则使用数对 (a,b) 表示数 c。例如：1.5 可使用数对 $(2,3)$ 表示；1.2 可使用数对 $(2,2)$、$(5,6)$ 等表示。选取数对时应综合考虑时间开销和精度要求。

方式 2　移位舍入编码

设原数据为 $c.d$，其中 c 为整数部分，d 为小数部分，且 c 与 d 位数基本相同。将原数据 $c.d$ 的小数点后移 v 位，得到新的数据 cd'。

输入　$c.d$

输出　$cd'\leftarrow\lceil c.d\times 10^v\rfloor$

这种处理方式相当于将原有数据扩大了 10^v 倍，然后根据实际要求，对扩大后数据的小数部分进行适当的舍入，从而既能够保证该类型数据的精度，又不需要在加密时选取较大的加密方案参数。

方式 3　多元处理编码

原数据 $c.d$ 如方式 2 描述，且 c 与 d 的位数均小于加密算法所能处理的最大位数。

输入　$c.d$

输出　(c,d)

将原数据的整数部分 c 和小数部分 d 拆分为二元组 (c,d)。

这种处理方式最具有普适性，能够完全保证数据精度不会因预处理操作而受到损失，然而这种处理方式也存在一定的弊端。用户若使用方式 3 将原数据变为二元组，则数据量变为原来的两倍，从而增加加密阶段的开销。除此之外，在执行同态乘法操作时，由原来一对整数相乘变为两对整数相乘，乘法次数增加，计算复杂度增加。

在实际应用过程中，根据所需处理的数据精度，可以选取上述 3 种预处理方式中的一种，也可以选取多种预处理方式结合使用，使效率与精度之间保持平衡。与此同时，针对选取的同态加密算法不同，其明文空间的类型和容错能力也不相同的情况，在选取具体预处理方式时，也应当考虑预处理方式对后续同态加密计算开销等方面的影响。

2.4　DBSCAN 算法

DBSCAN 算法是一种常见的聚类算法，用来在数据集中构建聚类簇并发现噪声数据[17-18]。相较于同样常见的 K-means 聚类算法，DBSCAN 算法不需要预先定义数据簇的个数，并且适用于任何形状的聚类簇的构建，甚至是无连接的环状聚类簇。由于存在最少点数的限制，相较于 K-means 算法，DBSCAN 算法可以避免 single-link 影响，因此对于任意形状的数据分布，DBSCAN 算法都具有较好的聚类效果。

DBSCAN 算法本身也有很多变形，原始的 DBSCAN 算法复杂度为 $O(n^2)$，ρ-近似 DBSCAN 算法复杂度为 $O(n)$，但是其对数据的维度有一些限制。DBSCAN 算法的选取与数据类型相关，根据本文方案数据集的数据类型，选取二维 DBSCAN 算法完成相关实验。

DBSCAN 算法的一些概念定义如下。MinPts 定义了一个聚类簇所需的最少数据点数，ε-邻域表示以某个点为中心点，并以其为圆心、以 ε 为半径的圆所覆盖的范围。中心点，即聚类簇的中心，其 ε-邻域中包含的数据点比 MinPts 多；边缘点，即在聚类簇边缘的节点，其 ε-邻域中包含的数据点比 MinPts 少，并且其在其他中心点的 ε-邻域中；噪声点，即数据集中非中心点和边缘点的其他数据点。节点定义的实例化描述如图 2 所示。除此之外，定义 2 和定义 3 给出了密度可达和密度相连的定义。

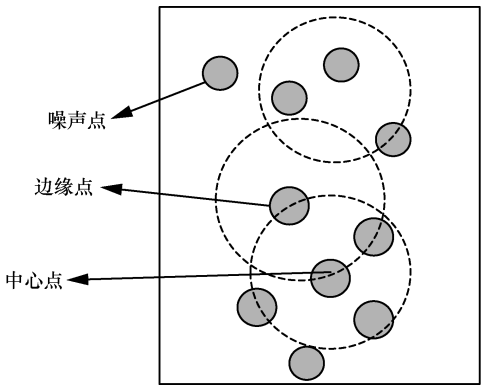


图 2　节点定义的实例化描述

定义 2　密度可达。令 x_i,x_j 为 2 个数据点。设存在样本序列 $p_1,p_2,\cdots,p_n$，其中 $p_1=x_i$，$p_n=x_j$，若对于 $1\leqslant k\leqslant n$，都有 p_{k+1} 在以 p_k 为中心点的 ε-邻域中，则称 x_j 由 x_i 密度可达。

定义 3　密度相连。对于数据点 x_i,x_j，如果存在一个点 x_k，使通过 x_k 后 x_i 与 x_j 是密度可达的，则称 x_j 与 x_i 密度相连。

DBSCAN 算法的流程如下。

步骤 1　输入数据集合。选择一个未被标记过的观察点 x_{I} 作为当前节点，标记第一个聚类簇 1。

步骤 2　找出以 x_{I} 为中心点的 ε-邻域内的所有节点，其均为 x_{I} 的邻居节点。执行下述操作。

①若找到的邻居节点数目少于 MinPts，则 x_{I} 是一个噪声点，执行步骤 4。

②若找到的邻居节点数目不少于 MinPts，则 x_{I} 是一个中心点，执行步骤 3。

步骤 3　分别将所有邻居节点作为中心点，重复步骤 2，直到没有新的邻居节点可以作为中心点。

步骤 4　从数据集 X 中选择下一个没有被标记的点作为当前节点，将聚类簇的数目更新并加 1。

步骤 5　重复步骤 2～步骤 4，直到数据集 X 中的所有点都被标记。

2.5　方案的评估标准

方案的评估标准包括 2 个方面：时间效率和准确率。

时间方面，本文关注的是云端执行同态 DBSCAN 算法的时间。除了时间的评估外，另一个重要的评估标准是聚类的准确率。根据 DBSCAN 算法的特点，准确率的评估标准包含聚类簇数量和噪声点判断。本文将对比直接在明文上进行 DBSCAN 算法所得结果和在密文上同态执行 DBSCAN 算法解密后得到的结果。聚类的准确率定

义为 2 次聚类结果中相同结果的占比，最理想状态为 2 次结果完全相同。

3　DBSCAN 隐私保护方案

本文构建了一个在加密数据集上的同态聚类算法，从而实现在外包计算过程中敏感数据集的隐私保护。根据同态加密算法所支持的运算方式，可以将 DBSCAN 算法中的运算划分成 2 类：同态加密算法支持的运算和同态加密不支持的运算。对于同态加密不支持的运算，可以通过设计协议的方式来进行处理，从而实现加密数据集上的同态聚类功能。同态 DBSCAN 算法如算法 1 所示。

算法 1　同态 DBSCAN 算法

输入　密文集合 $X=\{[\![x_1]\!],[\![x_2]\!],\cdots,[\![x_m]\!]\}$，密文参数 $[\![\varepsilon]\!]$，MinPts

输出　聚类簇 $C_1,C_2,\cdots,C_k$

1)初始化中心点集合 $\Omega=\varnothing$

2)for j=1,2,$\cdots$,m do

3)　获取 $[\![x_j]\!]$ 的 $[\![\varepsilon]\!]$ - 邻域：$N_{[\![\varepsilon]\!]}([\![x_j]\!])=$ findneighbor$([\![x_j]\!],[\![\varepsilon]\!])$

4)　if $\left|N_{[\![\varepsilon]\!]}([\![x_j]\!])\right|\geqslant$ MinPts　then

5)　　将 $[\![x_j]\!]$ 放入集合：$\Omega=\Omega\cup\{[\![x_j]\!]\}$

6)　end if

7)end for

8)初始化聚类集合 $k=0$

9)初始化未访问数据点集合 $\Gamma=X$

10) while　$\Omega\neq\varnothing$ do

11)　记录未访问的数据集 $\Gamma_{\text{old}}=\Gamma$

12)　随机选取一个中心点 $o\in\Omega$，初始化队列 $Q=\langle o\rangle$（$\langle\cdot\rangle$ 表示队列，一种先进先出的数据结构）

13)　$\Gamma=\Gamma\setminus\{o\}$

14)　while　$Q\neq\langle o\rangle$

15)　　从队列 Q 中选取 $[\![q]\!]$

16)　　if $\left|N_{[\![\varepsilon]\!]}([\![q]\!])\right|\geqslant$ MinPts　then

17)　　　令 $\Delta=N_{[\![\varepsilon]\!]}([\![q]\!])\cap\Gamma$

18)　　　将 Δ 中的样本放入 Q

19)　　　$\Gamma=\Gamma\setminus\{\Delta\}$

20)　　end if

21)　end while

22)　$k=k+1$，获得聚类簇 $C_k=\Gamma_{\text{old}}\setminus\Gamma$

23)　$\Omega=\Omega\setminus C_k$

24) end while

常见的同态加密方案仅能支持加法（减法）、乘法运算，所以复杂运算需要进行相应的转换使其能够用加法和乘法表示。算法 1 的步骤 3)中的函数 findneighbor$([\![x_j]\!],[\![\varepsilon]\!])$ 所涉及的运算并不能够完全被同态加密算法所支持。不支持的运算有以下 2 种。

1)　在计算 $[\![\varepsilon]\!]$-邻域时，需要计算点与点之间的距离，最常用的是欧氏距离，但其中所涉及的开方运算并不被同态加密算法支持。

2)　本文所选取的同态加密方案不具备保序加密的性质，加密后的密文之间不会保持原有明文间的大小关系，故需解决密文大小比较问题。

上述 2 个问题的解决方式将分别在 3.2 节和 3.3 节中进行介绍。

3.1　数据集预处理

本节首先给出一种基于数据特点，结合精度和计算开销等条件的选取数据预处理方式的方法，选取流程如图 3 所示。

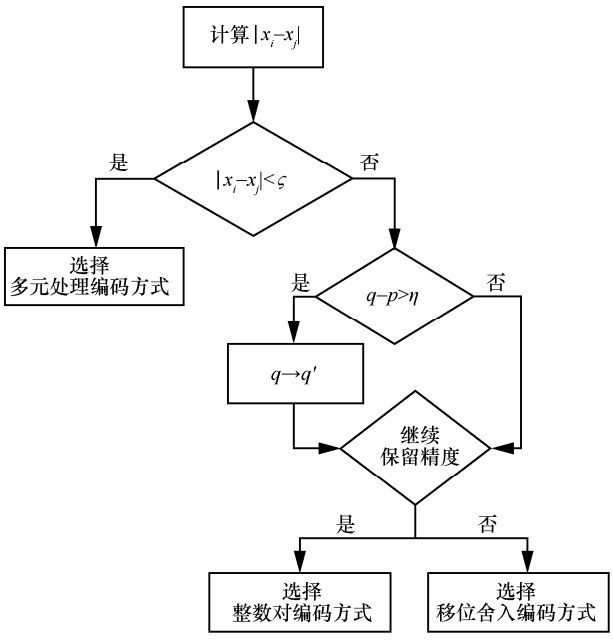


图 3　预处理方式的选取流程

设需处理的数据集为 $X=\{x_1,x_2,\cdots,x_n\}$，数据整数部分的位数为 p，小数部分的位数为 q（有效位数不足时用 0 补齐）。从 X 中选取较大的数 x_i 和较小的数 x_j，计算 $\left|x_i-x_j\right|$ 的值，判断此绝对值是否

满足$\left|x_i - x_j\right| < \varsigma$，其中$\varsigma$为用户根据数据集的特点自行设定的阈值。若满足（即数值相差较小），则直接选用多元处理编码方式以保证数据精度（此种情况说明集合中的数值大小差异主要在小数部分）；若不满足，则进一步判断是否满足$q - p > \eta$，其中η为用户根据情况自主设定的阈值。若$q - p > \eta$，则将小数部分的较低位舍去（大约舍去$q - p$位），即小数部分的位数由q变为q'（使$q' \approx p$）；否则直接跳转到下一步判断。此时，需要综合考虑保留精度和后续计算开销问题来判断是选择整数对编码方式还是移位舍入编码方式：若此时的数据精度的损失对后续计算结果影响较大，为提高计算精度，则选择整数对编码方式；若数据精度的损失对后续计算结果影响较小，则选择移位舍入编码方式。

本文方案中需要保护的敏感信息是每条数据的坐标值，是浮点型数据，因此需要根据数据集的特点以及本文方案所选用的加密方案来选择适合的预处理算法。本文方案选取了2个浮点型数据集A、B，A中数据的小数点后位数较多，但其数值大小的差异较大，且小数位数远大于整数位数，故根据图3所示的选取流程，可以考虑采用整数对编码方式或移位舍入编码方式，并且对小数部分进行舍去，本文方案中分别保留了3、4、5位小数进行后续实验。B中数据间差异较大且小数位数与整数位数相差不大，因此根据上文所述的选取方法，也可以考虑使用整数对编码方式或移位舍入编码方式。使用整数对编码方式处理数据后的加密过程如图4所示。

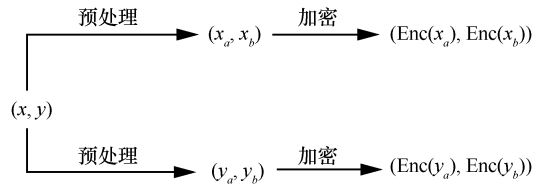


图4 使用整数对编码方式处理数据后的加密过程

从图4中可以看出，一个数据经方式1处理后会转换成一个二元组进行加密，后续的密文操作也会增多，从而计算开销会变大，但整数对编码方式能够尽可能地保留当前数据的精度。移位舍入编码方式则会造成一定程度的数据精度损失，但相较于整数对编码方式计算开销会更低。因此，可以通过实验来观测数据精度的损失对A、B这2个数据集聚类结果的影响，从而决定使用哪种预处理方式是更合适的（实验结果见4.1节）。本文方案选取的加密算法的明文空间是有限比特长度的整数集，因此预处理后的数据可以直接进行加密。

3.2 距离测度选取

在机器学习算法中，常用的距离测度包括欧氏距离、曼哈顿距离、变形欧氏距离等。相较于传统欧氏距离，变形欧氏距离能更进一步保留数据的精度，降低由开方舍入造成的误差影响。

定义4 变形欧氏距离。现有2个点$\boldsymbol{a}(x_1, y_1)$与$\boldsymbol{b}(x_2, y_2)$，变形欧氏距离可表示为

$$\mathrm{dis}(\boldsymbol{a}, \boldsymbol{b}) = \left(x_1 - x_2\right)^2 + \left(y_1 - y_2\right)^2 \tag{1}$$

在进行距离比较时，欧氏距离和变形欧氏距离具有较高的准确性，曼哈顿距离有一定的误差。然而，欧氏距离中的开方运算不被同态加密算法支持，需要对开方运算进行进一步处理。文献[19]中使用了牛顿迭代的方法进行同态开方运算，过程如下。

定理1 牛顿迭代开方。若方程$x^2 - t = 0$，其中t为实数，则在x_0附近存在一个根，使用迭代公式

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \tag{2}$$

依次计算$x_1, x_2, \cdots$，序列将无限逼近方程的根。

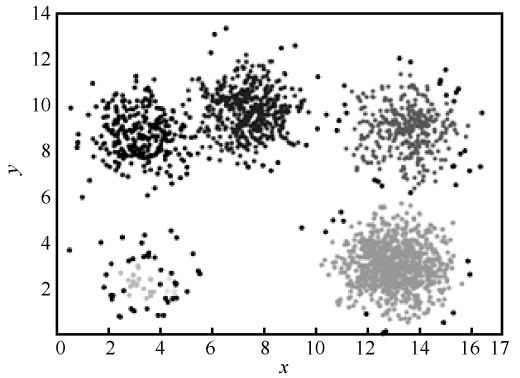
如定理1所示，牛顿迭代法可将开方运算转换为基础运算，从而满足同态加密方案的要求。这种解决方法虽然可以实现开方运算，但式(2)涉及的所有常数均需提前进行加密处理，并且在运算过程中涉及多次密文乘法运算，计算复杂度较高。除此之外，牛顿迭代法所得结果是近似结果，会有一定的误差，此误差可能会给聚类结果造成较大影响。

定理2 令a和b为2个整数，设a的比特长度为l_1，b的比特长度为l_2，那么有ab的比特长度不大于$l_1 + l_2$，$\sqrt{ab}$的比特长度不大于$\left(l_1 + l_2\right)/2$。

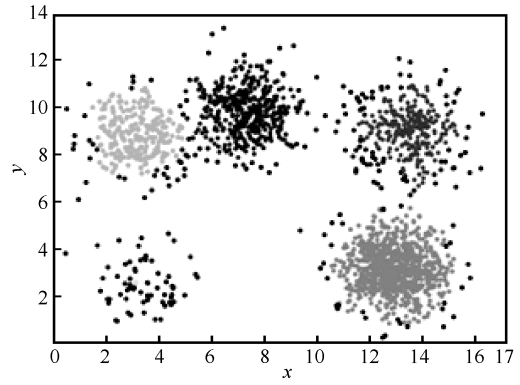
如定理2所示，使用变形欧氏距离所得结果的比特长度为欧氏距离结果的两倍。虽然这会导致在后续计算过程中数据规模变大，从而使计算时间略有上升，但变形欧氏距离的计算过程仅需2次乘法，相较于牛顿迭代法（为了保证开方结果的精度，需要迭代多次，从而需要多次乘法），比特长度增长所带来的额外开销相对较低。

与此同时，本文进行了距离测度对聚类准确度影响的测试。一般情况下，聚类算法中使用的距离测度是欧氏距离，因此测试的主要方式是将欧氏距

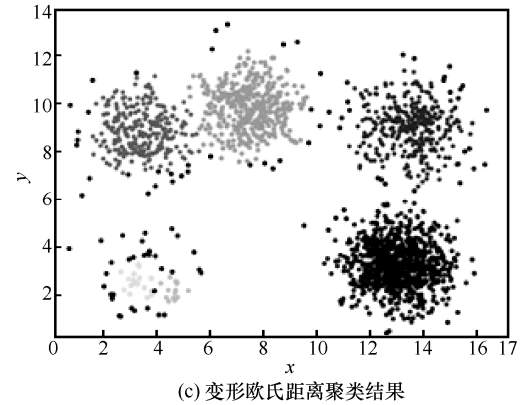
离分别替换成曼哈顿距离和变形欧氏距离后观察聚类结果是否与使用欧氏距离时的聚类结果相近。选取同样的输入参数，使用这 3 种距离测度在明文上进行聚类，得到的结果如图 5 所示。其中，横纵坐标为数据点的值，相同颜色为同一聚类簇，不同颜色为不同聚类簇，边缘深色数据点为噪声点。图 5 中的 3 幅图分别表示选择欧氏距离、曼哈顿距离和变形欧氏距离的聚类结果。由图 5 可知，欧氏距离和变形欧氏距离的聚类结果类似，而曼哈顿距离相较于欧氏距离的聚类结果有一定误差。结合上述分析与实验结果，同时考虑数据集精度和后续同态运算的开销，本文方案采用变形欧氏距离。



(a) 欧氏距离聚类结果



(b) 曼哈顿距离聚类结果



(c) 变形欧氏距离聚类结果

图 5　数据集 A 不同距离测度聚类结果

3.3　密文比较协议

在算法 1 的步骤 3 中，判断某一点是否在 $[\![x_j]\!]$ 的邻域中需要进行比较大小的操作。然而本文所使用的 BGV 同态加密算法不具有保序加密的特点，故使用其加密的数据不能保持原有大小顺序，不能直接进行比较。为解决这一问题，本文方案通过设计协议对密文大小进行比较，具体如协议 1 所示。令 $x=(x_{l-1},x_{l-2},\cdots,x_0)_2$ 和 $y=(y_{l-1},y_{l-2},\cdots,y_0)_2$ 分别为 x和y 的二进制表示。在 x和y 的密文上进行比较大小时，首先计算 $\mathrm{Enc}(x)-\mathrm{Enc}(y)+\mathrm{Enc}(K)$，其中，$K$ 表示一个比特位数大于 l 的整数，然后将所得密文返回给用户。用户解密密文得到 $\mathrm{tmp}=x-y+K$ 并提取出其最高位，若该位为 1，则 $x>y$；若该位为 0，则 $x<y$。由于协议中未使用同态乘法操作，因此不需要引入复杂的降噪技术，从而具有较低的时间复杂度。在一轮协议中，服务器向用户发送密文中间结果 Enc(tmp)，其比特长度为 $O(dn\log q)$（参数 n,d,q 的定义见 2.2 节）。用户收到 Enc(tmp) 后，向服务器端返回 1 bit 的结果（0/1）。因此，一轮比较协议的通信量的理论值为 $O(dn\log q)$。

协议 1　密文比较协议

输入　$\mathrm{Enc}(\mathrm{dis}_1)$，$\mathrm{Enc}(\mathrm{dis}_2)$

输出　0 / 1

1) 服务器计算 $r=\mathrm{Enc}(\mathrm{dis}_1)\text{-}\mathrm{Enc}(\mathrm{dis}_2)+\mathrm{Enc}(K)$
2) 服务器将 r 返回给用户端
3) 用户端解密 r 得到 tmp，并提取最高位 tmp_l
4) if $\mathrm{tmp}_l==1$ then
5) return 1
6) else
7) return 0
8) end if

用户端与服务器端的交互场景如图 6 所示。服务器将需要解密的密文发送给用户，用户解密密文后将解密结果的最高位发回给服务器。在此过程中，用户与服务器可以协商设置一个时间段 t，每隔 t 时间，用户向服务器发起询问，服务器依据运算进程返回需解密的中间结果或同态聚类结果。在这种情形下，用户不需要一直在线等待，只需间隔一段时间发起询问即可。这一过程涉及明文的直接传输，攻击者可能通过窃听的方式获得明文和密文，这一过程的安全问题将在 5.2 节中进行详细分析。

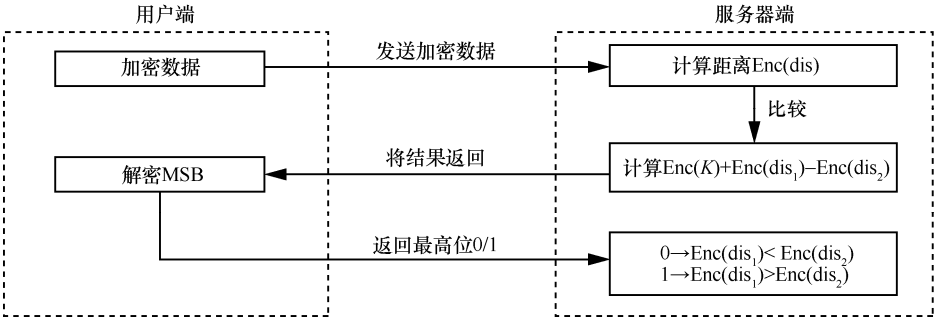


图 6　用户端与服务器端的交互场景

4　方案实现

本文实验使用的配置为 Intel(R) Core(TM) i7-6700 CPU @ 3.40 GHz　3.41 GHz，16 GB 内存，借助 Helib 同态加密库完成对数据集的加密和同态聚类运算。方案选取 2 个常见的聚类数据集。数据集 A 是常见形状数据集，共有 5 个聚类簇，2 000 条数据，每条数据含有 2 个坐标，每个坐标小数点后有 10 位小数。数据集 B 为 Aggregation 数据集，是常用特殊形状聚类数据集，共有 750 条数据，每条数据含有 2 个坐标,每个坐标小数点后有 2 位小数，与整数部分位数基本相同。

本文通过比较明文数据的聚类结果和密文数据上的同态聚类结果来验证方案的准确性。实验中，明文数据集和密文数据集在执行 DBSCAN 算法时，所用参数 ε 和 MinPts 一致，实验中 ε 记作 eps，MinPts 记作 min_Pts。

4.1　数据预处理方式选取

数据集 A 和数据集 B 选取的是常见的聚类算法数据集，分别代表整数与小数部分位数相差较大和较小 2 种情况，以此说明本文方案具有普适性。本节通过实验结果来验证 3.1 节选取方法的正确性，并得出更适合于数据集 A 和数据集 B 的预处理方式。实验使用 3 种预处理方式分别对 2 个数据集编码，然后进行同态聚类操作，得到同态聚类算法的时间开销和同态聚类的准确率，如表 1 所示。从表 1 中可以看出，多元处理编码的准确率是 100%，但是时间开销较大；整数对编码和移位舍入编码的准确率虽然均没有达到 100%，但也是非常准确的，这说明 3.1 节的选取流程中首先将多元处理编码排除，认为数据集 A 和数据集 B 更适合采用整数对编码方式或移位舍入编码方式，并且对数据集 A 中数据的小数部分进行一定程度的舍弃是正确的。此外，整数对编码方式的准确率会略高于移位舍入编码方式，但是移位舍入编码的准确率也非常高，并且移位舍入编码的计算效率远高于整数对编码。实验结果表明，本文实验选取的数据集更适合使用移位舍入编码方式，所以本文实验给出的具体结论都是建立在选择移位舍入编码方式的基础上的。

4.2　数据集明文聚类结果

选取合适的参数，分别对数据集 A 和数据集 B 进行聚类处理，结果如图 7 和图 8 所示。其中数据集 A 选取参数为 eps =0.547（编码后为 5 470），min_Pts=9；数据集 B 选取参数为 eps =1.8（编码后为 180），min_Pts=11。

4.3　数据集密文聚类结果

对数据集 A 和数据集 B 分别进行移位舍入编码处理后再加密，然后对加密后的数据执行同态聚类操作。对数据集 A 中的数据进行移位舍入编码处理，结果分别保留 3、4、5 位小数，而后在密文上执行同态聚类算法，解密后的聚类结果如图 9～图 11 所示。对数据集 B 直接进行移位舍入编码，加密后进行同态聚类，计算结束后解密得到的聚类结果如图 12 所示。

表 1　　**不同编码方式时间开销与准确率**

数据集	整数对编码		移位舍入编码		多元处理编码	
	时间/s	准确率	时间/s	准确率	时间/s	准确率
A	298.45	99.81%	115.48	99.75%	302.45	100%
B	115.45	100%	45.28	99.74%	155.65	100%

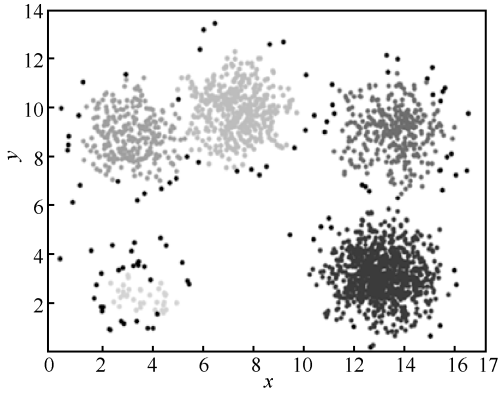


图 7　数据集 A 明文聚类结果

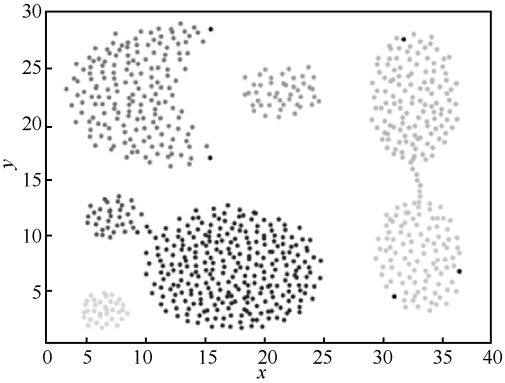


图 8　数据集 B 明文聚类结果

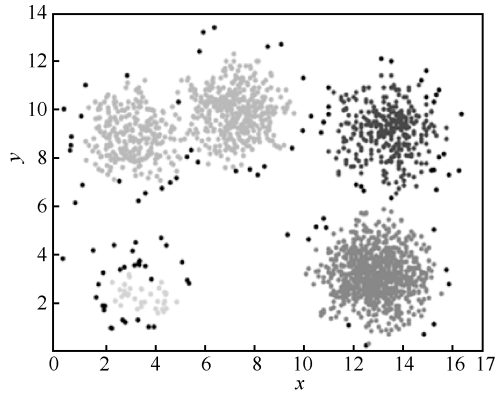


图 9　数据集 A 保留 3 位小数密文聚类结果

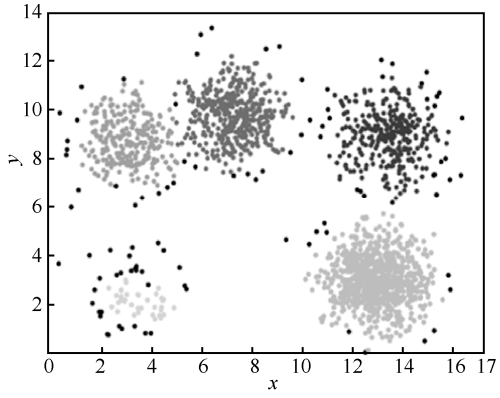


图 10　数据集 A 保留 4 位小数密文聚类结果

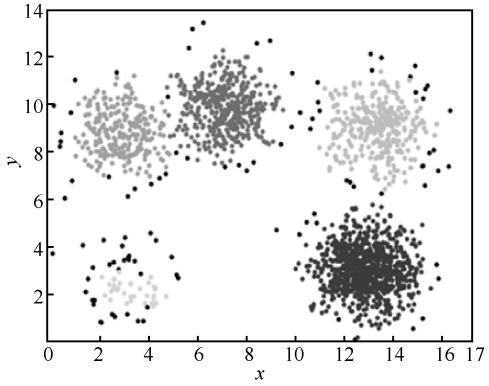


图 11　数据集 A 保留 5 位小数密文聚类结果

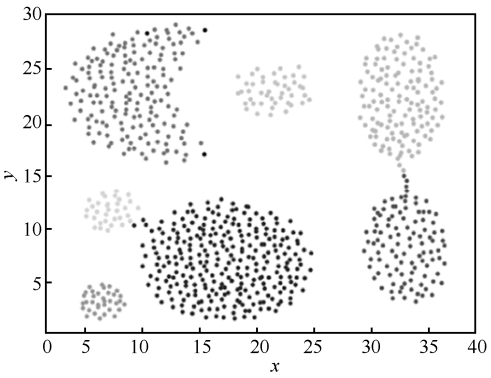


图 12　数据集 B 密文聚类结果

5　方案评估

5.1　实验结果分析

本节分别对比数据集A和数据集B的明文和密文聚类结果，得出方案准确率如表 2 和表 3 所示。

表 2　　数据集 A 明文和密文聚类结果对比

数据集 A	聚类簇数	噪声点数目	准确率
明文数据集	5	93	100%
加密数据集（3 位小数）	4	90	80%
加密数据集（4 位小数）	5	90	99.75%
加密数据集（5 位小数）	5	90	99.75%

表 3　　数据集 B 明文和密文聚类结果对比

数据集 B	聚类簇数	噪声点数目	准确率
明文数据集	7	8	100%
加密数据集	7	6	99.74%

从表 2 可以看出，预处理时保留 3 位小数的加密数据集的聚类簇数与明文有较大差异，准确率仅为 80%；进一步保留精度的加密数据集（4 位小数和 5 位小数）具有更好的聚类准确性，这 2 种情况下的准确率约为 99.8%。表 3 结果显示，在数据集

B 的聚类结果中，明文和密文聚类簇数相同，噪声点数目接近，准确率在 99.7%以上。文献[20]给出了现有其他使用同态加密实现机器学习隐私保护方案的准确率，平均准确率为 95%，最高准确率为 99.8%，因此相较于此前的方案，本文方案实现了较高的准确率。

时间开销方面，方案[8]对数据进行逐比特加密，导致运算时间较长，对 400 条密态二维数据的处理时间长达 15 h；本文方案时间开销较低，在处理数据集 B 时，对 750 条密态二维数据进行同态聚类算法仅需 45.28 s。

聚类结果出现误差的主要原因包括 2 个方面：一方面是实际数据的舍入误差，考虑到后续加密时间开销问题，在数据预处理阶段对数据的精度进行了一定程度的舍弃；另一方面，由于本文方案中的密文比较协议仅能得出两密文之间的大于或小于关系，对于等于关系直接将其归入大于的范围内，这会造成一定误差。针对第二个问题，可以考虑采用数值比较器进行改善。HElib 已有相关函数，但该函数需要多次密文乘法操作，大大影响计算效率。

5.2 安全性分析

本节将对本文方案的安全性进行评估。在本文方案中，将云服务器设定为诚实但是具有好奇心的半诚实模型，选取的 BGV 算法是满足语义安全的同态加密算法。

本文方案希望达到的安全目标为云服务器和具备窃听能力的敌手无法获取用户的数据信息，也无法通过同态 DBSCAN 算法倒推出原始数据。在以上设定的前提下，下文将给出几个定理并进行证明，以说明本文方案的安全性。

定理 3 云服务器或具备窃听能力的敌手根据他们获取的信息成功恢复用户原始数据的概率是可忽略的。

证明 云服务器或具备窃听能力的敌手能够获得的数据为 $X=\{[\![x_1]\!],[\![x_2]\!],\cdots,[\![x_m]\!]\}$ 和参数 $[\![\varepsilon]\!]$、MinPts。从数据的组成来看，除了 MinPts 之外，其他都是通过 BGV 同态加密得到的密文。云服务器和敌手在不知道私钥的情况下成功获得用户原数据的概率等同于攻破 BGV 算法的概率。由同态加密算法的语义安全可知，攻破 BGV 算法的概率是可忽略的，因此云服务器和敌手成功获取用户数据的概率也是可忽略的。云服务器或敌手能够获得的数据中唯一的明文是 MinPts，而此明文只是表示聚类簇中包含的最小数据个数，并不会泄露用户的任何数据。

此外，本文方案中涉及云服务器与用户之间的一个协议。本质上来说，在此协议中云服务器将一个 BGV 密文发送给用户，用户返还给云服务器 0 或 1。云服务器或敌手在不知道私钥的情况下，无法获得该密文的具体内容，并且此时的用户返还的明文（0 或 1）和其对应的云服务器发送的密文也不构成密码学中定义的“明密文对”，因而无法在获取足够多的 0 或 1 明文与其对应密文的情况下推测出密钥。证毕。

定理 4 云服务器或具备窃听能力的敌手获取同态 DBSCAN 算法后，能够成功获得用户原始数据的概率是可忽略的。

证明 云服务器或具备窃听能力的敌手可能通过窃听等手段获得在云端执行的同态 DBSCAN 算法的具体内容。DBSCAN 是一种无监督型机器学习算法，即不需要通过设立训练集预先构建出模型再利用该模型对其他数据进行处理，而是直接在数据上进行聚类操作得到结果。本文方案保留了 DBSCAN 算法的这一特点，直接作用于密文数据来获得同态聚类结果，并不存在模型。与分类器的隐私保护方案不同，利用模型参数倒推用户原始数据这类攻击无法应用于本文方案。另外，本文方案实质上进行的运算是依据密文数据同态计算距离，然后给每个数据点标记其归属的聚类簇或将其标记为噪声点，因而攻击者即使获取了算法过程，也无法通过算法内容来推测出原始数据信息。因此，本文方案的数据安全性立足于加密算法的安全性。

综上，攻击者获得同态 DBSCAN 算法后，能够成功获取用户原始数据的概率等同于攻破 BGV 算法的概率，由定理 4 可知，此概率是可忽略的。证毕。

6 结束语

本文提出了一种在加密数据集上进行同态 DBSCAN 聚类算法的方案，用于解决数据外包计算过程中的隐私保护问题。该方案针对不同数据集精度，提出了多种编码预处理方式，并且给出了一种基于数据集特点、综合考虑数据精度和计算开销等方面的数据预处理方式的选取策略；依据同态加密算法支持的运算类型和实验测试结果，选取变形欧氏距离作为算法中的距离测度；针对同态加密算法

不支持的比较运算，设计了一个交互协议来实现此功能。本文方案具有可靠的数据安全性、良好的聚类效果和计算性能。

参考文献：

[1] ACAR A, AKSU H, ULUAGAC S, et al. A survey on homomorphic encryption schemes: theory and implementation[J]. arXiv Preprint, arXiv:1704.03578, 2017.

[2] BRAKERSKI Z, GENTRY C, VAIKUNTANATHAN V. (Leveled) fully homomorphic encryption without bootstrapping[C]//Innovations in Theoretical Computer Science. New York: ACM Press, 2012: 309-325.

[3] BRAKERSKI Z, VAIKUNTANATHAN V. Fully homomorphic encryption from ring-LWE and security for key dependent messages[C]//Proceedings of the 31st Annual Conference on Advances in Cryptology. Berlin: Springer, 2011: 505-524.

[4] KIM J, KIM S, SEO J H. A new scale-invariant homomorphic encryption scheme[J]. Information Sciences, 2018, 422: 177-187.

[5] 蒋林智, 许春香, 王晓芳, 等. 同态加密在基于密文计算模型中的应用[J]. 密码学报, 2017, 4(6): 596-610.
JIANG L Z, XU C X, WANG X F, et al. Application of homomorphic encryption for encrypted computing models[J]. Journal of Cryptologic Research, 2017, 4(6): 596-610.

[6] YANG H M, HE W C, LI J, et al. Efficient and secure kNN classification over encrypted data using vector homomorphic encryption[C]//IEEE International Conference on Communications. Piscataway: IEEE Press, 2018: 1-7.

[7] CHEON J H, JEONG J, KI D, et al. Privacy protection K-means clustering with multiple data owners[J]. IACR Cryptology ePrint Archive, 2019(2019): 466.

[8] ANGELA J, ARMKNECHT F. Unsupervised machine learning on encrypted data[C]//International Conference on Selected Areas in Cryptography. Berlin: Springer, 2018: 453-478.

[9] HU S S, WANG Q, WANG J J, et al. Securing SIFT: privacy protection outsourcing computation of feature extractions over encrypted image data[J]. IEEE Transactions on Image Processing, 2016, 25(7): 3411-3425.

[10] CHEN G M, CHEN Q, ZHU X Y, et al. Encrypted image feature extraction by privacy protection MFS[C]//2018 International Conference on Digital Home. Piscataway: IEEE Press, 2018: 42-45.

[11] JIANG L Z, XU C X, WANG X F, et al. Secure outsourcing SIFT: efficient and privacy protection image feature extraction in the encrypted domain[J]. IEEE Transactions on Dependable and Secure Computing, 2017, PP(99): 1.

[12] JIANG L Z, XU C X, WANG X F, et al. Statistical learning based fully homomorphic encryption on encrypted data[J]. Soft Computing, 2017, 21(24): 7473-7483.

[13] BACON D F, BENT G A, BERGAMASCHI F A, et al. Performing efficient comparison operations on encrypted data: 14952210[P]. (2015-11-25)[2020-09-28].

[14] JIANG L Q, CAO Y, YUAN C S, et al. An effective comparison protocol over encrypted data in cloud computing[J]. Journal of Information Security and Applications, 2019, 48: 102367.

[15] 贾春福, 王雅飞, 陈阳, 等. 机器学习算法在同态加密数据集上的应用[J]. 清华大学学报(自然科学版), 2020, 60(6): 456-463.
JIA C F, WANG Y F, CHEN Y, et al. Machine learning algorithms on homomorphic encrypted data set[J]. Journal of Tsinghua University (Science and Technology), 2020, 60(6): 456-463.

[16] HALEVI S, SHOUP V. Bootstrapping for HElib[C]//Annual International Conference on the Theory and Applications of Cryptographic Techniques. Berlin: Springer, 2015: 641-670.

[17] KRYSZKIEWICZ M, SKONIECZNY L. Faster clustering with DBSCAN[C]//2005 Intelligent Information Processing and Web Mining. [S.n.:s.l.], 2005: 605-614.

[18] CHEN Y W, TANG S Y, BOUGUILA N, et al. A fast clustering algorithm based on pruning unnecessary distance computations in DBSCAN for high-dimensional data[C]//Pattern Recognition. [S.n.:s.l.], 2018: 375-387.

[19] CHEON J H, KIM D, KIM D, et al. Numerical method for comparison on homomorphically encrypted numbers[C]//25th International Conference on the Theory and Application of Cryptology and Information Security. [S.n.:s.l.], 2019: 415-445.

[20] 谭作文, 张连福. 机器学习隐私保护研究综述[J]. 软件学报, 2020, 31(7): 2127-2156.
TAN Z W, ZHANG L F. Survey on privacy preserving techniques for machine learning[J]. Journal of Software, 2020, 31(7): 2127-2156.

[作者简介]

贾春福（1967− ），男，河北文安人，博士，南开大学教授、博士生导师，主要研究方向为网络与系统安全、可信计算、恶意代码分析、密码技术应用等。

李瑞琪（1993− ），男，黑龙江尚志人，南开大学博士生，主要研究方向为同态加密、格密码学等。

王雅飞（1995− ），女，天津人，南开大学硕士生，主要研究方向为同态加密应用、隐私保护等。