计算机科学
COMPUTER SCIENCE

基于同态加密的隐私保护数据分类协议

陆星缘, 陈经纬, 冯勇, 吴文渊

引用本文

陆星缘, 陈经纬, 冯勇, 吴文渊.基于同态加密的隐私保护数据分类协议[J]. 计算机科学, 2023, 50(8): 321-332.

LU Xingyuan, CHEN Jingwei, FENG Yong, WU Wenyuan. Privacy-preserving Data Classification Protocol Based on Homomorphic Encryption [J]. Computer Science, 2023, 50(8): 321-332.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.220700130

基于同态加密的隐私保护数据分类协议

陆星缘[1,2]　陈经纬[3]　冯　勇[3]　吴文渊[3]
1 公共大数据国家重点实验室　贵阳 550025
2 贵州大学计算机科学与技术学院　贵阳 550025
3 中国科学院重庆绿色智能技术研究院　重庆 400714
(568901982@qq.com)

摘　要　随着大数据、云计算技术的发展,用户对于云计算服务的需求也与日俱增。在用户申请云计算服务时,其隐私数据需要在云平台进行存储与计算,而这也带来了隐私数据泄露的问题。同态加密允许在不解密的情况下对密文进行直接运算,得到的新密文解密后即为运算结果,因此可以用于保障用户的隐私数据安全。在半诚实模型下考虑如下两方面的计算框架:用户端按照指定方式将隐私数据加密为密文后发送到服务器端,服务器端根据同态加密方案允许明文与密文间进行运算的性质,使用训练得到的明文模型对用户端发送来的加密数据进行分类,最后将加密的分类结果发送回用户端,由用户端自行解密获得隐私数据的分类结果。在这个框架下,基于同态加密方案BGV设计了超平面分类器、决策树以及KNN这3种机器学习分类算法。根据每种分类器的特性,结合SIMD技术设计不同的密文数据打包策略与分类计算流程,使得用户端与服务器端之间的通信开销大幅降低。特别地,在预测阶段,超平面分类器与决策树实现了无交互的分类,KNN仅需1次交互即可完成分类,并基于HElib同态加密库,采用C++语言实现了这3种分类器。在UCI公开数据集上,超平面分类器能够在几十毫秒到几百毫秒内完成对1个待预测样本的分类,决策树最慢能够在几十毫秒内完成,两种分类器对密文数据的预测准确率均能超过90%,两方仅需要承担用户端发送给服务器端的加密隐私数据与服务器端发送回用户端的加密分类标签的通信开销;KNN分类器平均4s左右完成对1个待预测样本的分类,对密文数据的预测准确率在90%以上,两方除了隐私数据与分类标签的通信开销外,只需要额外负担一轮服务器端与用户端的中间计算结果即可完成分类。与基于同态加密的同类协议相比,在通信轮数、预测准确率、运行效率等方面均有不同程度的改进。

关键词:同态加密;安全多方计算;隐私保护;机器学习;HElib
中图法分类号　TP309.2

Privacy-preserving Data Classification Protocol Based on Homomorphic Encryption

LU Xingyuan[1,2],CHEN Jingwei[3],FENG Yong[3] and WU Wenyuan[3]
1 State Key Laboratory of Public Big Data,Guizhou University,Guiyang 550025,China
2 College of Computer Science and Technology,Guizhou University,Guiyang 550025,China
3 Chongqing Institute of Green and Intelligent Technology,Chinese Academy of Sciences,Chongqing 400714,China

Abstract　With the development of big data and cloud computing,the demand for cloud computing services is growing dramatically. When users apply for cloud computing services,their privacy data needs to be stored and computed on cloud platforms,which may cause leakage of private data. Homomorphic encryption allows direct computation on ciphertexts,and the decryption of the resulting ciphertext is the same as computing on plaintexts,so homomorphic encryption can protect users' private data. Here a framework for two parties in the semi-honest model is considered. The client encrypts the privacy data into ciphertext according to a homomorphic encryption scheme and sends it to the server,and the server uses the plain machine learning model to classify the encrypted data from the client. Finally,the server sends the encrypted classification result back to the client,and the client decrypts the classification result by itself. With the framework above,three machine learning classifiers,the hyperplane,decision tree,and *k*-nearest neighbor classifier,based on the Brakerski-Gentry-Vaikuntanathan(BGV) homomorphic encryption scheme are investigated. According to the characteristics of each classifier,different ciphertext data packaging strategies and calculation

到稿日期:2022-07-12　返修日期:2022-12-14
基金项目:贵州省科技计划项目([2020]4Y056);科技部重点研发计划项目(2020YFA0712303);重庆市科技项目(cstc2021jcyj-msxmX0821,cstc2020yszx-jcyjX0005,cstc2021yszx-jcyjX0004,2022YSZX-JCX0011CSTB,2021000263)
This work was supported by the Guizhou Science and Technology Program([2020]4Y056),Key Research and Development Program of the Ministry of Science and Technology(2020YFA0712303) and Chongqing Science and Technology Program(cstc2021jcyj-msxmX0821, cstc2020yszx-jcyjX0005,cstc2021yszx-jcyjX0004,2022YSZX-JCX0011CSTB,2021000263).
通信作者:陈经纬(chenjingwei@cigit.ac.cn)

processes are designed with single-instruction-multiple-data (SIMD) technology, which significantly reduces the communication overhead between the client and the server. In the prediction phase, the hyperplane and decision tree classifiers achieve interaction-free, and the KNN classifier only needs one interaction. Moreover, the three classifiers are implemented with a homomorphic encryption library HElib. For several UCI public datasets, the hyperplane classifier can complete the privacy-preserving classification within tens of milliseconds to hundreds of milliseconds for a single sample, and the decision tree can complete it within tens of milliseconds. The prediction accuracy of the first two classifiers for ciphertext data exceeds 90%, and the two parties only need the communication cost of the client sending the encrypted private data to the server, and the server returns the encrypted classification label to the client. The k-nearest-neighbor classifier completes one sample's classification in about 4 seconds on average, and the prediction accuracy of ciphertext data is also more than 90%. In addition to the communication overhead of privacy data and classification labels, the two parties also need an additional round of intermediate calculation results between the server and the client to complete the classification. Compared with similar protocols based on homomorphic encryption, the proposed protocols have advantages in the number of communication rounds, prediction accuracy, and computational efficiency.

Keywords Homomorphic encryption, Secure multi-party computation, Privacy protection, Machine learning, HElib

1 引言

1.1 研究背景

随着大数据、云计算技术的不断发展，大量公司、机构得以搭建自己的云计算平台，并能够向用户提供海量数据的存储、复杂计算、预测等服务。用户端将自己的数据发送到服务器端，服务器端根据相应的算法或者模型来完成对数据的处理、计算、预测等工作，并将相应的结果发送回用户端[1]。如果用户端向服务器端申请服务时提供了隐私数据，那么服务器端可能试图收集这些数据；另一方面，用户端也可能从服务器端返回的计算结果中学习服务器端的模型、训练集等信息。因此，双方的隐私数据安全均受到了威胁。

为了解决上述问题，需要建立安全两方计算方案来保障两方的隐私数据安全。在服务器端和用户端都是半诚实的参与者的情况下，可以选择混淆电路[2]、秘密分享[3-4]、不经意传输[5]以及同态加密[6]等技术来保证两方隐私数据的安全。混淆电路技术使得参与方在互相不知晓对方数据的情况下，同时通过对运算电路进行加密来实现隐私保护，但混淆电路的开销随着电路规模的增大而增大，因此其适合实现的运算规模较小。秘密分享技术可以将数据拥有方需要计算的隐私数据切分后分发给多个计算方分别进行计算，最后这些计算方将计算结果合在一起后才能解密计算结果，并以此达到隐私保护的目的，但该技术的通信与计算开销随着参与方数量的增加而增加，且在解密时需要足够多的参与方才能够完成解密。不经意传输技术能够在数据发送方发送消息后，使得数据接收方仅得到某一条消息，且数据接收方不知道另一方发送的其他消息，数据发送方也不知道另一方接收的是哪一条信息。但不经意传输技术需要一套较复杂的算法来实现，且消息的传输效率不高。同态加密技术允许在不解密的情况下直接对密文进行运算，得到的新密文解密后即为运算结果，且能够显著降低各参与方进行交互的轮次。因此，同态加密的特性使得其在隐私保护方面有着独特的优势。

同态加密方案是一种特殊的公钥加密方案，其允许加密后的数据直接进行加法或者乘法运算。如果某个同态加密方案同时支持任意深度的密文加法与乘法运算，则称其为全同态加密(Fully Homomorphic Encryption, FHE)方案。文献[7]提出了bootstrapping技术，并以该技术实现了首个FHE方案。该方案支持任意深度的密文加法与乘法运算。然而，bootstrapping技术的效率较低，不能很好地适应实际需求。现阶段的实际需求更多采用效率更高的BGV[8]，BFV[9]，CKKS[10]等层次型同态方案(Leveled Homomorphic Encryption, LHE)，这些方案可以实现有限深度的密文运算，相对于传统的加密方案来说，还能够抵御量子攻击[11]，并且可以借助bootstrapping技术来实现任意深度的密文运算。除此之外，上述的LHE方案还支持明文与密文之间的混合加法与乘法运算，这个特性使得各方能够专注于隐私数据的加密保护，作为计算方的服务器端在进行分类计算时不必对自己持有的模型进行加密。

1.2 相关工作

到目前为止，已经有较多将同态加密与机器学习结合的研究，其中有与Paillier方案[12]结合的文献[13-15]，也有选择与BGV/BFV等方案进行结合的文献[16-19]。Paillier方案由Paillier于1999年提出，是一种仅能够实现密文加法同态运算的方案，因此与机器学习进行结合时，需要使用一定的方法来使得模型能够在Paillier方案的环境下实现密文间的乘法、比较等其他运算。

Yasumura等[13]将BGV方案与朴素贝叶斯结合，通过使用比特0与1来编码样本，以避免朴素贝叶斯产生的浮点数运算，并以此实现加密的隐私数据的分类。但他们提出的方法中，对于Breast Cancer数据集，一个密文仅包含1～2个样本，对于明文槽空间的利用率不高，这也会加重通信的负担。Cai等[14]将k-means聚类算法与Liu等提出的同态加密方案进行结合，提出了一种安全三方计算方案，该方案能够在只有单个云服务器端的情况下完成整个预测计算过程。该方案保证了申请计算的两方A与B各自持有的隐私数据不泄露给对方与服务器端，但在聚类阶段迭代后需要A与B进行交互才能够计算出新簇心的位置。Park等[15]提出将公平机器学习(Fair Machine Learning)与CKKS方案进行结合，以减轻在训练过程中对种族、性别等敏感属性的歧视或不公平，同时还能保证这些敏感数据的安全性，最终得到一种具有公平性的安全超平面分类器EFSVMHE，该分类器相对于不采用公平机器学习的同种分类器LSSVM来说，尽管显著地提高了

模型预测的公平性，但预测准确率出现了一定的下降。

文献[16-17，19]的工作则与本文工作较为接近。Jia等[16]选用Paillier同态加密方案来对用户端的数据进行加密，在一定程度上保障了用户端的数据计算安全。但他们的方法需要在预测前事先对原始数据进行一定的处理，且对密文数据的分类准确率不如对明文数据的高。Dey等[17]将决策树与超平面分类器结合，并选用Paillier方案实现了安全的分类器DT-TSVM。他们设想由区块链平台提供训练集，使用Paillier方案加密多个数据供应方的隐私数据，相比与其类似的MBSVM分类器的效果有所提升，其对UCI[18]数据集的预测准确率在90%左右，但他们的方法耗时较久。Xu等[19]基于Paillier方案、QR方案以及一种FHE方案的结合，设计了支持隐私保护的安全KNN分类器PP-KNN，该分类器服务于半诚实的服务器端与客户端之间的两方安全计算。这使得该分类器能够同时处理密文的同态加法与同态乘法，但在进行分类时需要进行协议转换，协议转换不仅使得分类效率降低，并且需要频繁地在服务器端与客户端两方之间进行数据交换，加大了通信的开销。

1.3 本文的贡献

本文基于BGV加密方案，在半诚实模型下实现了超平面分类器、决策树以及KNN 3种机器学习分类方案；根据每种分类器的特性，结合SIMD技术设计不同的密文数据打包策略与分类计算流程，使得用户端与服务器端之间的通信开销大幅降低。特别地，在预测阶段，超平面分类器与决策树实现了无交互的分类，KNN仅需1次交互即可完成分类工作。我们基于HElib[20]同态加密库，采用C++语言实现了上述3种分类器。在UCI公开数据集上，超平面分类器能够在几十毫秒到几百毫秒内完成对1个待预测样本的分类，决策树最慢能够在几十毫秒内完成，这两种分类器对密文数据的预测准确率均能超过90%；KNN分类器平均在4s左右能够完成对1个待预测样本的分类，对密文数据的预测准确率在90%以上，两方只需要额外负担一轮服务器端与用户端的中间计算结果的开销即可完成分类。

在与本文相近的成果中，Jia等提出的基于Pallier同态加密方案实现的超平面分类器与安全贝叶斯分类器在预测准确率上较低，仅为60%～71%；Dey等的基于Paillier同态加密方案实现的分类器DT-TSVM在UCI公开数据集上的表现较好，预测准确率在90%左右，但平均需要数秒到十秒以上才能完成对对1个待预测样本的分类；Xu等的PP-KNN分类器则需要上千轮的交互次数才能完成几十个待预测样本的分类。

2 预备知识

2.1 BGV同态加密方案

2.1.1 方案概述

BGV方案由Brakerski，Gentry与Vaikuntanathan于2011年提出，且荣获了2022年的哥德尔奖(Gödel Prize)。本文使用的BGV方案均按照RLWE来进行介绍。

其明文空间为$R_p=\mathbb{Z}_p[X]/\langle\Phi_m(X)\rangle$，密文空间为$R_q=\mathbb{Z}_q[X]/\langle\Phi_m(X)\rangle$。其中$\Phi_m(X)$为$m$次分圆多项式，$p$为素数，$q$为一大整数。BGV方案实现的操作如下：

(1)$BGV.Setup(1^\lambda)$：输入安全参数λ，输出方案参数$params$。

(2)$BGV.KeyGen(params)$：输入方案参数$params$，输出公钥pk以及私钥sk。

(3)$BGV.Enc_{pk}(msg)$：输入待加密的信息$msg\in R_p$，输出加密后的信息$c\in R_q$。此过程需要公钥pk才能进行。

(4)$BGV.Dec_{sk}(c)$：输入待解密的密文$c\in R_q$，输出解密后的信息$msg\in R_p$。此过程需要私钥sk才能进行。

(5)$BGV.Refresh_{pk}(c_e)$：输入一个经过密文运算(密文乘法或者密文加法)后的结果密文c_e，对c_e执行密钥转换(key switching)与模转换(modulus switching)操作后，输出密文c_r。该操作主要用于降低经过密文运算后产生的噪声。

(6)$BGV.Add_{pk}(c_1,c_2)$：输入用同一公钥pk加密的密文$c_1,c_2\in R_q$，输出$c_3=c_1+c_2$。之后可使用$BGV.Refresh_{pk}$操作降低运算时产生的噪声。

(7)$BGV.Mult_{pk}(c_1,c_2)$：输入用同一公钥pk加密的密文$c_1,c_2\in R_q$，输出$c_3=c_1\otimes c_2$，$\otimes$为张量积运算。之后可使用$BGV.Refresh_{pk}$操作降低运算时产生的噪声。

BGV方案的密文在进行运算时将产生一定的噪声，如果噪声超过上限，那么该密文将不能被正确地解密。在密文加法与密文乘法中，乘法产生的噪声要远多于加法。假设两个密文各自噪声的上界为e，那么做了1次密文加法后，结果密文的噪声的上界为$2e$，而做1次密文乘法后，结果密文的噪声的上界增加到了e^2。因此，做了密文乘法后更需要$BGV.Refresh_{pk}$操作来降低其噪声。

BGV方案具有语义安全(semantically secure)，根据语义安全，任何BGV密文间都是不可区分的，同时BGV密文可以通过一些技术(如bootstrapping技术等)将自己的噪声刷新，使得密文间无法通过噪声大小进行区分，从而实现电路安全(circuit privacy)[21-22]。

BGV方案实现了SIMD(Single-Instruction Multiple-Data)技术，这使得1个BGV密文内可填充多个数据，每1个数据被填装在1个明文槽(slot)内。BGV方案也支持一些针对明文槽的操作，如编码(Encode)、旋转(Rotate)、比较(Comp)、求和(Totalsums)等。

2.1.2 密文基本运算

本小节将介绍第3节中协议设计时会用到的部分重要的操作：$Totalsums$，$Comp$，$Rotate$。

$Totalsums(a)$操作可以求出密文a的全部明文槽内数据的总和sum，并将sum赋值到a的所有明文槽内。密文求和操作的例子如图1所示。

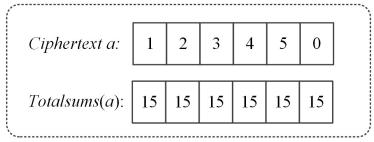


图1 密文求和

Fig. 1 Ciphertext summation

密文比较操作记为$Comp_{pk}(a,b)$，比较后返回一个新

密文 c_n，若 a 的某明文槽 $S_a[x]$ 的值比 b 的同位明文槽 $S_b[x]$ 大，那么 $S_n[x]$ 的值为 0，反之则为 1。关于密文比较的具体原理与实现可以参考文献[23]，密文比较的例子如图 2 所示。

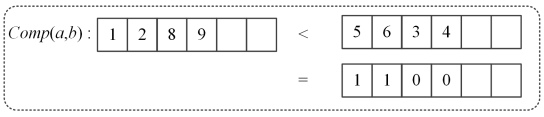


图 2 密文比较

Fig. 2 Ciphertexts comparison

Rotate(*a*,*length*)操作可以将密文 a 的整个明文槽向左或者向右整体移动 *length* 个槽位，当 *length* 为正时向右，为负时向左，密文末端的槽在移动时循环到另一个末端继续移动，密文旋转的例子如图 3 所示。

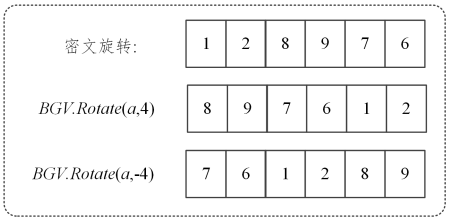


图 3 密文旋转

Fig. 3 Ciphertexts rotation

2.1.3 两种数据打包方式

根据服务器端打包密文时填装数据的方式不同，数据打包方式可分为列方式打包加密与行方式打包加密。采用列方式打包加密时，会将所有样本看做一个整体，然后将样本的第 1 列数据按顺序填装到所有密文的第 1 个明文槽中，之后的数据同理，直到所有列都被填装完毕，此时密文总数等同于样本的特征数。采用行方式打包加密时，则将第 1 个样本的全部特征数据按顺序填装到第 1 个密文中，后续的样本以此类推，此时密文总数等同于样本数。这两种密文打包方式的例子如图 4 所示。

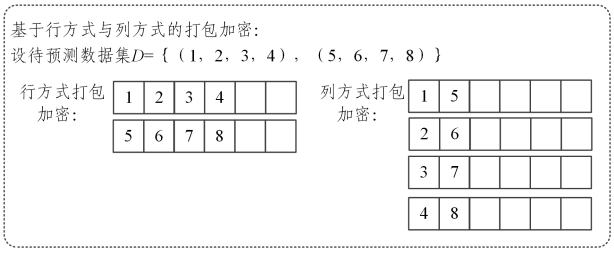


图 4 密文打包加密

Fig. 4 Ciphertexts packaging and encryption

2.2 超平面分类器、决策树与 KNN 分类器

2.2.1 超平面分类器

超平面分类器是一种经典的机器学习分类器，可由 SVM 算法[24]构建。SVM 算法将待预测样本记为 x，将 y 记为这些待预测样本的分类。SVM 算法通过对大量数据的学习来为待预测样本划分出一个超平面，以将这些待预测样本划分为两类。最终学习出的模型可记为 $y=\boldsymbol{w}^{\mathrm{T}}x+b$，其中 $\boldsymbol{w}^{\mathrm{T}}$ 为超平面的“斜率”，是一个 n 维的向量，b 为一个常数，每当 w^{T} 或者 b 有变化时，就产生了一个新的超平面。当进行分类时，将待预测样本 x 代入模型中进行计算，若 y 小于 0 则样本被划分为负类，若 y 大于 0 则样本被划分为正类。由于在 y 等于 0 时超平面分类器不好划分，因此在选定超平面时应当避免有样本使得 y 等于 0。

2.2.2 决策树

决策树则是根据待预测样本的每种分类应该具有的特性来决策每种分类应该具有什么属性，然后将这样的决策过程训练为一颗根节点在上的树。该树的根节点一般代表划分能力最好的属性，待预测样本的分类过程也从此开始；每个非叶子节点代表一个条件，该条件一般约束待预测样本的某个属性，待预测样本将沿着其满足的分支向下继续决策；每个叶子节点则代表一个分类，当待预测样本决策到某个叶子节点时即代表整个分类过程结束，该叶子节点代表的分类即为该待预测样本的分类。

2.2.3 KNN 分类器

k 近邻(k-Nearest Neighbour，简称为 KNN)分类器的思想较为简单，即将待预测样本与训练集样本放在同一空间，然后求解待预测样本与所有训练集样本之间的距离，最后取出与待预测样本前 k 近的训练集样本，根据这 k 个样本的多数分类来决定待预测样本的分类。该分类器没有专门的训练阶段，当有待预测样本出现时就可以直接根据上述思想对待预测样本进行分类。在 KNN 分类器中，常用曼哈顿距离、欧氏距离或者闵氏距离等来度量待预测样本与训练集样本之间的距离。

2.3 敌手模型

为了证明后文提出的协议的安全性，在此定义一些在证明中将会用到的符号以及公式[25]。本文设计的所有协议的参与方均仅有 Client 与 Server 两方，且安全模型均为半诚实模型。

在半诚实模型下，Client 与 Server 都会遵循协议并完成每一个计算步骤，但这两方均带有好奇心，也就是说，Client 会试图从中间计算过程推理出 Server 持有的预测模型，Server 也会试图从整个协议的运行中推理出 Client 持有的隐私数据。

在证明过程中，我们将 Client 简称为 C，将 Server 简称为 S。我们将 $f=(f_{\mathrm{C}},f_{\mathrm{S}})$ 记为某个计算函数。考虑在“现实世界”中完成 f 的计算，就需要一个合理且安全的协议来实现，该协议记为 Π。在某个两方计算协议中，C 与 S 需要共同完成计算 $f(a,b)$，其中 a 是 C 提供的输入，b 是 S 提供的输入，在计算过程中需要使用协议 Π，并且 Π 的安全参数为 λ。在协议的执行过程中，C 与 S 各自能够得知经手于自己这边的数据，按照上述的协议例子 Π，在 C 的视角下能够得知的数据记为 $V_{\mathrm{C}}(\lambda,a,b)$，同理 S 的视角记为 $V_{\mathrm{S}}(\lambda,a,b)$。

考虑在“理想世界”中完成 f 的计算，可以假设存在可信第三方 T，然后由 T 计算 $f(a,b)$后将计算结果发送回去。为了证明协议 Π 的安全性，那么就需要在理想环境下为 C 与 S 各自建立一个与 Π 等效的模拟器，即 S_{C} 与 S_{S}，这两个模拟器仅能够使用 V_{C} 和 V_{S} 的所有消息作为输入，如果 S_{C} 与 S_{S} 这两个模拟器与 Π 在计算上是无法区分的，那么就能证明 Π 是安全的，计算式如下：

$$\{(V_{\mathrm{C}}(\lambda,a,b),\Pi(\lambda,a,b))\}\equiv_c\{S_{\mathrm{C}}(\lambda,a,f_A(a,b),f(a,b))\} \tag{1}$$

$$\{(V_S(\lambda,a,b),\Pi(\lambda,a,b))\}\equiv_c\{S_S(\lambda,f_B(a,b),b,f(a,b))\} \quad (2)$$

其中，$\equiv_c$符号表示左右两个式子在计算上是不可区分的。为了在之后的证明中简洁记号与证明过程，将式(1)、式(2)简化为：

$$S_C(a,f_A(a,b))\equiv_c V_C(a,b) \quad (3)$$

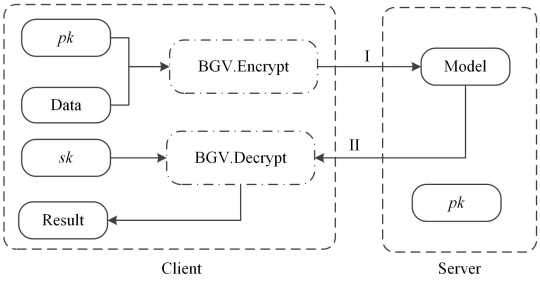
$$S_S(f_B(a,b),b)\equiv_c V_S(a,b) \quad (4)$$

3 基于 BGV 方案的安全两方分类计算协议设计

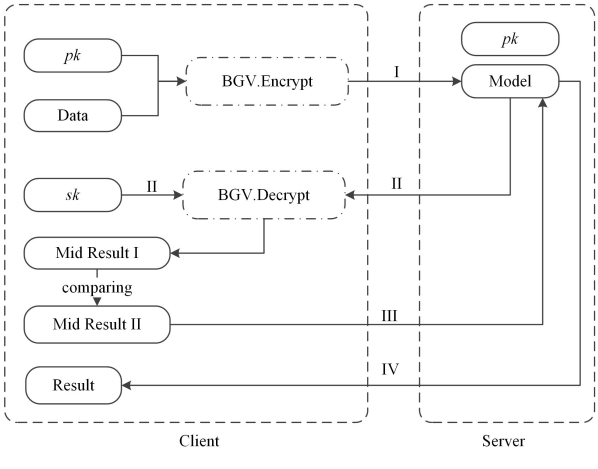
3.1 应用场景概述

在本文设计的协议中，仅有两个半诚实的参与者，即提供分类计算服务的服务器端 Server 以及申请服务的用户端 Client。其中 Client 一般持有待预测的隐私数据 Data、用于加密数据的公钥 *pk* 以及用于解密分类结果的私钥 *sk*，Server 一般持有相应的分类器模型 Model 以及 *pk*。

我们将依次给出采用超平面分类器、决策树以及 KNN 分类器的协议设计以及安全性证明。图 5 给出了这 3 种分类器所使用的协议的框架，其中超平面分类器与决策树的框架如图 5(a)所示，KNN 分类器的框架如图 5(b)所示。



(a)超平面分类器与决策树的框架



(b)KNN 分类器的框架

图 5 安全两方分类计算协议的框架

Fig. 5 Framework of secure two-party classification computing protocols

3.2 超平面分类器的安全两方分类计算协议

3.2.1 协议设计

协议 1 实现了服务器端采用超平面分类器时的安全两方分类计算协议。通过 BGV 方案的 SIMD 技术，我们实现了待预测样本的批量分类，提高了分类效率。

协议 1 超平面分类器的安全两方分类计算协议

Client 的输入：$X=(X_1,\cdots,X_n)$，其中 $X_j=(x_{j1},x_{j2},\cdots,x_{ju})$，n 为待预测样本数，u 为样本特征数

Server 的输入：明文超平面分类模型(w,b)，若 $y<0$ 则标签 $L(x)=0$，若 $y\geqslant 0$ 则标签 $L(x)=1$，其中 $y=\mathbf{w}^Tx+b$

Client 的输出：$(X_1,\cdots,X_n)$的分类标签 $Label=(L_1,\cdots,L_n)$

Client：

1. $Data\leftarrow[X_i]=[(x_{ij})]$，$i=1,2,\cdots,n$；$j=1,2,\cdots,u$；
2. for $i\leftarrow 1$ to u do
3. $\quad Ct[i]\leftarrow BGV.Enc_{pk}(x_{1i},x_{2i},\cdots,x_{ni})$；
4. end //使用列方式打包加密 Data
5. 将 Ct 发送给 Server；

Server：

6. 准备一个空密文 C_L，与仅填充 p/2 的密文 C_{Hp}；
7. 输入明文模型 $y=\mathbf{w}^Tx+b$；
8. for $i\leftarrow 1$ to u do
9. $\quad Ct[i]\leftarrow BGV.Mult_{pk}(\mathbf{w}^T,Ct[i])$；
10. $\quad C_L\leftarrow BGV.Add_{pk}(C_L,Ct[i])$；
11. end //计算 $\mathbf{w}^Tx$
12. $C_L\leftarrow BGV.Add_{pk}(C_L,b)$；
13. $C_L\leftarrow Comp_{pk}(C_L,C_{Hp})$；
14. 将 C_L发送给 Client；

Client：

15. $Label=(L_1,L_2,\cdots,L_n)\leftarrow BGV.Dec_{sk}(C_L)$；
16. output Label。

协议 1 中的 Client 使用列方式打包加密数据，Server 也根据列方式的逻辑对加密数据进行分类计算。由 Server 执行的部分为整个协议的分类阶段，在分类阶段中，Server 不需要向 Client 额外申请其他数据即可完成分类，因此协议 1 可以做到无交互的分类。

协议 1 中 Server 进行分类计算的过程可以抽象为如下的类矩阵运算过程，该矩阵的每一行为一个密文，BGV. Enc 操作在运算过程中简称为 Enc，其他操作同理。

首先，Client 将其持有的隐私数据 Data 使用列方式打包加密后得到密文 *Ct*，*Ct* 的第 *i* 行在协议 1 中就是 *Ct*[*i*]，*Ct* 的具体形式如下：

$$Ct=\begin{bmatrix}Enc(x_{11},x_{21},\cdots,x_{n1})\\Enc(x_{12},x_{22},\cdots,x_{n2})\\\vdots\\\vdots\\Enc(x_{1u},x_{2u},\cdots,x_{mu})\end{bmatrix}$$

然后，Server 将 *Ct* 与模型中的 $\boldsymbol{w}^T$ 按行相乘后，*Ct* 变为如下形式：

$$Ct=\begin{bmatrix}w_1\cdot Enc(x_{11},x_{21},\cdots,x_{n1})\\w_2\cdot Enc(x_{12},x_{22},\cdots,x_{n2})\\\vdots\\\vdots\\w_u\cdot Enc(x_{1u},x_{2u},\cdots,x_{mu})\end{bmatrix}$$

接着，Server 先将 *Ct* 的每一行全部相加，再与明文 *b* 进行相加后得到 C_L。由于 BGV 方案支持明文与密文的混合运算，且运算的结果为密文，因此将明文 *b* 加到密文 C_L 后，C_L 此时加密的是每个待预测样本的 *y* 值。

$$\begin{aligned}C_L&=Enc(\sum_{i=1}^{u}w_ix_{1i},\sum_{i=1}^{u}w_ix_{2i},\cdots,\sum_{i=1}^{u}w_ix_{m})+(b,\cdots,b)\\&=Enc(y_1,y_2,y_3,\cdots,y_n)\end{aligned}$$

计算完毕后，C_L 中加密的 y 值有正有负。由于 BGV 方案的明文空间是非负的，具体来说，$R_p \in [0, p-1]$，因此计算出的负值会被转换为 R_p 内对应的非负数。为了得到每个样本的分类，就需要将 y 与 $p/2$ 做一次密文比较，这样才能够区分出计算结果的正负，进而得到每个待预测样本的分类。采用密文比较的优势在于：1）能够有助于实现无交互的分类计算；2）对密文做完密文比较后，结果密文内仅存储 0 与 1，这样能够防止某些恶意的第三方对分类结果进行篡改。

$$C_L = Comp(C_L, C_{Hp})$$
$$= Comp\left(Enc(y_1, y_2, \cdots, y_n), Enc\left(\frac{p}{2}, \frac{p}{2}, \cdots, \frac{p}{2}\right)\right)$$
$$= Enc(L_1, L_2, \cdots, L_n), L_i = 0 \text{ 或 } 1$$

综上所述，协议 1 的正确性得证，同时由证明过程也可以看出，通过执行协议 1，可以同时完成 n 个样本的预测。

3.2.2 协议 1 的安全性分析

协议 1 的安全性证明如下：

（1）在执行协议 1 时，C 与 S 的视角分别是：

$V_C = (pk, sk, Data; C_L; Label)$

$V_S = (pk, y; Ct; C_L)$

（2）向模拟器 S_C 输入 $(pk, sk, Data; Label)$ 后，使用 pk 将 $Label$ 加密后得到 C_L'。因此，模拟器 S_C 的输出如下：

$S_C = (pk, sk, Data; C_L'; Label')$

由 BGV 方案的语义安全与电路安全可得，C_L' 与 C_L 之间是不可区分的，因此 $V_C \equiv_c S_C$。

（3）向模拟器 S_S 输入 (pk, y) 后，使用 pk 加密 u 个不同的随机数，得到 u 个密文，并按照先后顺序存入同一个密文组后得到 Ct'。

因此，模拟器 S_S 的输出为：

$S_S = (pk, y; Ct')$

与前面同理，Ct' 与 Ct 之间不可区分，因此 $V_S \equiv_c S_S$。

综上所述，协议 1 的安全性证明完毕，协议 1 在半诚实模型下是安全的。

3.3 决策树的安全两方分类计算协议

3.3.1 协议设计

由于决策树的特殊性，需要将决策树模型转换为一个等效多项式方可用于分类计算。将加密样本代入该等效多项式进行计算得到的结果等效于该决策树的分类结果[26]。

将二叉决策树转换为等效多项式的方法如下：图 6 中的示例决策树中的每个非叶子节点代表一个分类条件，即 R_1，R_2，R_3，每个叶子节点代表一个分类结果，即标签 0，1，2；然后对该决策树进行深度优先遍历，得到 3 条完整路径，分别为 $R_1 - R_2 - 0$，$R_1 - R_3 - 1$ 以及 $R_1 - R_3 - 2$。

在该二叉决策树中，如果待预测样本满足条件 R_1，那么在等效多项式中，R_1 的值将取 1，反之则取 0，其他的条件同理。这样，3 条路径就对应如下的 3 个多项式：

$P_1 = 0 \cdot R_1 \cdot R_2 \leftarrow R_1 - R_2 - 0$

$P_2 = 1 \cdot (1 - R_1) \cdot R_3 \leftarrow R_1 - R_3 - 1$

$P_3 = 2 \cdot (1 - R_1) \cdot (1 - R_3) \leftarrow R_1 - R_3 - 2$

以 P_2 为例，若某个待预测样本的分类标签为 1，则其不满足条件 R_1 但满足条件 R_3，因此 R_1 的值取 0 且 R_3 的值取 1。为保证该情况下 P_2 运算出的值为 1，P_2 的第二项为 $(1 - R_1)$，第三项则为 R_3，同时将 R_1 与 R_3 代入到另外两个多项式的计算结果都为 0。

最后，将这 3 个多项式相加，即可得到该决策树的等效多项式 $P = P_1 + P_2 + P_3$。当预测样本时，将样本与所有条件进行比较，得到 R_1，R_2，R_3 的值后代入 P 中进行计算，计算结果即为该样本对应的分类。

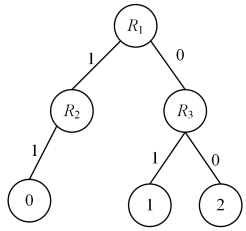


图 6 决策树转换为等效多项式

Fig. 6 Transform decision tree to polynomial

得到了决策树对应的等效多项式后，还应该考虑该多项式的乘法次数与最大深度，因为密文间的乘法次数与最大深度均会影响分类计算的效率，甚至决定分类结果能否正确解密。一般来说，P 的乘法次数越多，其乘法最大深度就越低，随着 P 不断地进行合并同类项，其乘法次数将变得越来越少，但其乘法最大深度会变得越来越大。因此，P 不做任何处理的状态下其乘法次数最大，P 被合并为最简形式的状态下其乘法深度最大。P 无论处于哪个极端，其分类效率都不会是最优的，因此需要均衡 P 的乘法次数与最大深度以达到最优的分类效率并能够解密。

除此之外，决策树的非叶子节点越多，其转换的等效多项式将会越复杂，而每明确一个分类条件 R 的值就需要在加密状态下进行一次比较，但密文间的比较操作的效率较低。为了提高多次密文比较的效率，我们采用并行的方式来执行密文比较。并行相对于串行能够提高的效率程度与执行协议时开启的线程数等条件有关，并行的实际效果与分析见第 4.2 节。

在理想的情况下，设完成 1 次密文比较需要的时间为 T。如图 7 所示，以串行完成 3 次密文比较需要的时间为 $3T$，当开启 3 个线程用于并行时，并行完成 3 次密文比较仅需要 $1T$ 的时间。

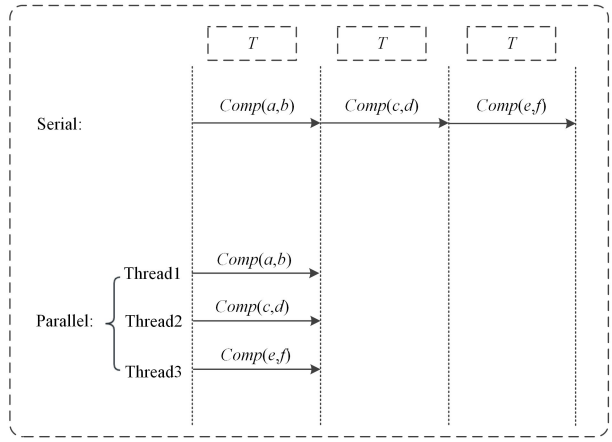


图 7 串行与并行

Fig. 7 Serial and parallel

协议 2 的完整流程如下。

协议 2 决策树的安全两方分类计算协议

Client 的输入：$X=(X_1,\cdots,X_n)$，其中 $X_j=(x_{j1},x_{j2},\cdots,x_{ju})$，n 为待预测样本数，u 为样本特征数

Server 的输入：明文等效多项式 $P(R_1,R_2,\cdots,R_t)$，其中 $R_j=0$ 或 1，t 为决策树的分类条件数

Client 的输出：$(X_1,\cdots,X_n)$的分类标签 $Label=(L_1,\cdots,L_n)$

Client：

1. $Data\leftarrow[X_i]=[(x_{ij})]$，i=1,2,…,n；j=1,2,…,u；
2. for i←1 to u do
3. $\quad Ct[i]\leftarrow BGV.Enc_{pk}(x_{1i},x_{2i},\cdots,x_{ni})$；
4. end //使用列方式打包加密 Data
5. 将 Ct 发送给 Server；

Server：

6. 准备一个空密文 C_L，以及空密文组 C_v；
7. 输入决策树转换的等效多项式 P；
8. t←P 含有的决策条件 R 的数量；
9. Rc=(C_i)，i=1,2,…,t，Rc 存储了 R_i要求比较的特征列号；
10. Vc=[$Enc(V_i)$]，i=1,2,…,t；//V_i填装了 R_i的条件值，如 R_i：$x_j\leqslant 5$，那么 $V_i=6$
11. parallel for i←1 to t do
12. $\quad C_v[i]\leftarrow Comp_{pk}(Ct[Rc[i]],Vc[i])$；
13. end//并行完成密文比较，其提升效率与线程数等相关
14. $C_L\leftarrow P(C_v)$；//将所有 R 的值代入 P 进行计算
15. 将 C_L发送给 Client；

Client：

16. $Label=(L_1,L_2,\cdots,L_n)\leftarrow BGV.Dec_{sk}(C_L)$；
17. output Label。

协议 2 与协议 1 的整体流程是类似的。在协议 2 开始时，Client 与协议 1 一样按照列方式打包加密 Data 并发送给 Server。

Server 在协议开始前事先准备好通过决策树模型转换的等效多项式 P，并根据 P 建立好相应的信息用于分类计算，即通过 P 得到 t，Vc，Rc 等信息。协议 2 中的 Client 同样将 Data 按照列方式打包加密后得到 Ct，这与协议 1 中打包加密 Data 的方式同理，此处不再赘述。

然后 Client 将 Ct 发送给 Server，Server 根据 Rc 取出 Ct 的相应行与 Vc 给出的相应条件值进行比较，直到 Rc 存储的所有特征列都比较完毕后，就求解出了每个待预测样本的所有 R 的值，这些值按照样本的顺序存储在密文组 C_v 中。通过比较得到每个待预测样本所有 R 的值的过程如下：

$$Rc=(C_1,C_2,\cdots,C_t)$$
$$Comp(Ct[C_1],Vc[1])\rightarrow C_v[1]\rightarrow R_1$$
$$Comp(Ct[C_2],Vc[2])\rightarrow C_v[2]\rightarrow R_2$$
$$\vdots$$
$$\vdots$$
$$Comp(Ct[C_t],Vc[t])\rightarrow C_v[t]\rightarrow R_t$$
$$R_i=Enc(a_1,a_2,\cdots,a_n),a_j=0\text{ 或 }1$$

接下来再将 C_v 中的值代入等效多项式 P 进行计算，即可得到每个待预测样本的加密分类 C_L，最后 Server 将其发送给 Client，Client 使用私钥 sk 解密 C_L 即可得到每个待预测样本的分类。计算结果 C_L 如下：

$$C_L=P(R_1,R_2,\cdots,R_t)=Enc(L_1,L_2,\cdots,L_n)$$

综上所述，协议 2 的正确性得证。与协议 1 同理，通过执行协议 2，也可以同时完成 n 个样本的预测。

3.3.2 协议 2 的安全性分析

协议 2 的安全性证明过程如下：

(1)在执行协议 2 时，C 与 S 的视角为：

$V_C=(pk,sk,Data;C_L;Label)$

$V_S=(pk,P;Ct;C_L)$

(2)向模拟器 S_C 中输入 $(pk,sk,Data;Label)$，然后使用 pk 将 $Label$ 加密得到 C_L'。

因此，S_C的输出为：

$S_C=(pk,sk,Data;C_L';Label)$

由于 BGV 方案具有语义安全与电路安全，因此 C_L与 C_L'之间不可区分，因此 $V_C\equiv_c S_C$。

(3)向模拟器 S_S输入 (pk,P)后，使用 pk 加密 u 个不同的随机数，得到 u 个密文，并按照先后顺序存入同一个密文组后得到 Ct'。

因此，S_S的输出为：

$S_S=(pk,P;Ct')$

与前面同理，Ct 与 Ct'之间不可区分，因此 $V_S\equiv_c S_S$。

综上所述，协议 2 的安全性证明完毕，协议 2 在半诚实模型下是安全的。

3.4 KNN 的安全两方分类计算协议

3.4.1 协议设计

在 Server 使用 KNN 分类器进行分类时，需要进行大量的数据比较操作，而密文比较的开销是较大的，因此设计将 KNN 的比较操作交给 Client 在明文中进行。我们采用新的打包加密方式与计算方法来使得该中间计算结果能够屏蔽训练集样本信息，同时不影响后续的计算。

我们选择了两种定义待预测样本与训练集样本点间距离 $dist$ 的方法。假设某个待预测样本点 $\vec{a}=(x_1,x_2,\cdots,x_u)$，某个训练集样本点 $\vec{b}=(y_1,y_2,\cdots,y_u)$，那么这两种距离的计算方式如下：

(1)欧氏距离的平方：

$$dist=(x_1-y_1)^2+(x_2-y_2)^2+\cdots+(x_u-y_u)^2 \tag{5}$$

(2)夹角转内积距离：

$$dist=\vec{a}\cdot\vec{b}=x_1\cdot y_1+x_2\cdot y_2+\cdots+x_u\cdot y_u \tag{6}$$

使用夹角转内积距离来定义两点间的距离时，需要先将待预测样本与训练集样本进行单位化后才能使用式(6)进行距离计算，此时求出的 $dist$ 为两个样本间的夹角的余弦值，$dist$ 越大，代表这两个样本间的距离越近。由于 BGV 方案只支持整数的数据加密，因此两方还需要各自对单位化后的样本进行同等倍数的放大与取整操作，这些操作不影响 $dist$ 间的大小关系。

协议 3 全程由 Client 执行，用于进行安全的明文比较，并且在执行过程中，Client 无法获取到 Server 的训练集样本信息。在协议 3 中有关 $dist$ 的大小比较是由比较矩阵完成的，比较矩阵能够明确任意数量的数据之间的大小关系。设待比较数据集 $L=[2,8,5,4,3]$，则其比较矩阵 $\boldsymbol{CM}$ 为：

$$\boldsymbol{CM}=\begin{bmatrix}1&0&0&0&0\\1&1&1&1&1\\1&0&1&1&1\\1&0&0&1&1\\1&0&0&0&1\end{bmatrix}$$

CM 的第 i 行是由 L 的第 i 个元素与 L 中的所有元素(包括第 i 个元素自身)进行一个大于等于的比较得来的,若其与 L 的第 j 个元素的大于等于关系为真,那么 ***CM*** 的第 i 行第 j 列的元素为 1,否则为 0。

将 ***CM*** 的每一行相加后,得到集合 $L'=[1,5,4,3,2]$。显然,某个数在 L' 中对应的值越大,该数在 L 中就越大。

协议 3 的整体流程如下。

协议 3 安全的明文比较协议 secure_comp(DS)

Server 的输入:DS(DS 为 Server 计算出的距离差集,其具体形式见协议 4)

Client 的输出:每个待预测样本的 k 个最近邻样本的索引值 k_idx,k_idx=(idx_{11},…,idx_{1k},idx_{21},…,idx_{2k},…,idx_{nk}),其中 n 为待预测样本数

```
Client:
1. for i←1 to m do
2.    Pt_S[i]←BGV.Dec_sk(DS[i]);
3. end //解密 DS,DS 的形式与推导见协议 4.
4. 准备 n 个规模为 m * m 的空矩阵,这个矩阵组记为 M;
5. for b←1 to n do
6.    for i←1 to m do
7.       for j←1 to m do
8.          row←(m-i+j)%m;
9.          col←b+i * n;
10.         if Pt_S[row][col]>p/2 do
11.            M[b][i][j]←0;
12.         else do
13.            M[b][i][j]←1;
14.         end
15.      end
16.   end
17. end
18. for b←1 to n do
19.   取出 M[b],计算其每一行的和,并将行和中前 k 大的行对应的行编号记录下来,并将这些行编号按顺序存入 k_idx;//求 k_idx
20. end
21. output k_idx。
```

距离差集 DS 由 Server 计算得出,DS 的具体计算过程见协议 4。距离差 $dist_{ab}-dist_{cd}$ 能够代表待预测样本 a 和训练集样本 b 之间的距离值,与待预测样本 c 和训练集样本 d 之间的距离的这两个距离值之间的大小关系,同时 $dist_{ab}-dist_{cd}$ 的值不包含任何直接的隐私数据,因此 Server 交付给 Client 进行比较不会泄露 Server 的训练集数据。DS 的具体形式如下所示,DS 的每一行为一个密文,其中的 D_j' 均在协议 4 中由 Server 计算得出。

$$DS=\begin{bmatrix} Enc(D_1'-D_1') \\ Enc(D_1'-D_2') \\ \vdots \\ \vdots \\ Enc(D_1'-D_m') \end{bmatrix}=\begin{bmatrix} Enc(D_1-D_1,D_2-D_2,\cdots,D_m-D_m) \\ Enc(D_1-D_2,D_2-D_3,\cdots,D_m-D_1) \\ \vdots \\ \vdots \\ Enc(D_1-D_m,D_2-D_1,\cdots,D_m-D_{m-1}) \end{bmatrix}$$

其中 $D_j=(dist_{1j},dist_{2j},\cdots,dist_{nj})$,$j=1,2,\cdots,m$。

$D_1'=(D_1,D_2,\cdots,D_{m-1},D_m)$

$D_2'=(D_2,D_3,\cdots,D_m,D_1)$

$\vdots$

$\vdots$

$D_m'=(D_m,D_1,\cdots,D_{m-2},D_{m-1})$

当 Client 接收到 DS 之后需要先将 DS 解密得到明文 Pt_S。Pt_S 的形式如下:

$$Pt\_S=\begin{bmatrix} D_1-D_1,D_2-D_2,\cdots,D_m-D_m \\ D_1-D_2,D_2-D_3,\cdots,D_m-D_1 \\ \vdots \\ \vdots \\ D_1-D_m,D_2-D_1,\cdots,D_m-D_{m-1} \end{bmatrix}$$

为了方便理解,首先观察 Pt_S 的第一列,即从 D_1-D_1 到 D_1-D_m 的这 m 个式子,这些式子相当于将 D_1 分别与 D_1,D_2,…,D_m 内的距离值进行了一次比较。由于距离差在计算后有可能是负数,与前面的协议情况相同,负数的计算结果在 Pt_S 中被转换为了相应的非负数,因此在协议 3 中计算比较矩阵的值时,同样需要与 $p/2$ 进行 1 次比较才能明确距离差的正负,进而判断比较矩阵的该元素是 1 还是 0。

$$\begin{bmatrix} D_1-D_1 \\ D_1-D_2 \\ \vdots \\ \vdots \\ D_1-D_m \end{bmatrix}=\begin{bmatrix} 0 \\ dist_{11}-dist_{12},\cdots,dist_{n1}-dist_{n2} \\ \vdots \\ \vdots \\ dist_{11}-dist_{1m},\cdots,dist_{n1}-dist_{mn} \end{bmatrix}$$

在完成上述减法后,第 1 列求解出来的实际上是 M 中的每一个比较矩阵的第 1 行,同理 Pt_S 的第 2 列的 D_2-D_1 到 D_2-D_m 求解出来的就是 M 中的每一个比较矩阵的第 2 行,以此类推。然而 Pt_S 的第 2 行以后的元素显然不是像第 1 列这样的顺序排列,因此协议 3 中的 row 与 col 就是为了找出第 i 列的头部 D_i-D_1,并将第 i 列按照第 1 列这样的顺序统计下去才能正确地得出每个比较矩阵的每一行。

最后,统计 M 中的每一个比较矩阵中前 k 行的和以及行号,从而得到每一个待预测样本的 k 个最近邻样本在训练集中的索引 k_idx,然后 Client 将其发送给 $Server$,k_idx 对应的训练集样本仅 $Server$ 能够得知。

综上所述,协议 3 的正确性得证,执行协议 3 将能够处理由 n 个样本产生的距离差集。

协议 4 将用密文旋转操作进行计算,且为了旋转正确,需要令所有密文的明文槽数刚好等于训练集样本数 m 与待预测样本数 n 的积。在这个前提下,两方只要按照要求将数据打包直到填满明文槽即可。

接下来给出协议 4 的整体流程,为了表达方便省略了夹角转内积距离的数据预处理过程。

协议 4 KNN 的安全两方分类计算协议

Client 的输入:X=(X_1,…,X_n),其中 X_j=(x_{j1},x_{j2},…,x_{ju}),n 为待预测样本数,u 为样本特征数

Server 的输入:明文训练集样本 TDATA=(Y_1,…,Y_m),其中 Y_j=(y_{j1},y_{j2},…,y_{ju}),m 为训练集样本数,u 同上

Client 的输出:(X_1,…,X_n)的分类标签 Label=(L_1,…,L_n)

```
Client:
1. Data←[X_i]=[X_ij],i=1,2,…,n;j=1,2,…,u;
```

2. for i←1 to u do

3. A[i]←BGV.Enc$_{pk}$($\overbrace{x_{1i},x_{2i},\cdots,x_{ni}}^{\text{重复填装m次}}$)；//打包加密 Data //从 x_{1i} 到 x_{ni} 为一组，共重复填装其 m 次

4. end

5. 将 A 发送给 Server；

Server：

6. 输入 TDATA←[y_i]=[(y_{ij})]，i=1，2，…，m；j=1，2，…，u；

7. for i←1 to u do

8. T[i]←($\overbrace{y_{1i},\cdots,y_{1i}}^{n个}$，$\overbrace{y_{2i}}^{n个}$，…，$\overbrace{y_{mi}}^{n个}$)；//打包 TDATA 但不加密 // y_{1i} 重复填装 n 次，然后递进到 y_{2i} 重复前面的操作，以此类推直到 y_{mi}

9. end

10. for i←1 to u do

11. D_1'←BGV.Add$_{pk}$(A[i]，−T[i])；

D_1'←BGV.Mult$_{pk}$(D_1'，D_1')；(式(5))

或

$2D_1'$←BGV.Mult$_{pk}$(A[i]，T[i])；(式(6))

12. end //计算 D_1'

13. for i←1 to m do

14. D_i'←BGV.Rotate(D_1'，−n·(i−1))；

15. DS[i]←BGV.Add$_{pk}$(D_1'，−D_i')；

16. end //计算距离差集 DS

17. 将 DS 发送给 Client；

Client：

18. k_idx←secure_comp(DS)；//Client 执行协议 3

19. 将 k_idx 发送给 Server；

Server：

20. 分别获取每个待预测样本的 k 个最近邻及其分类，这个操作记为 vote(candidate)，其中 candidate 为该待预测样本的 k 个最近邻所在的索引值

21. for b←1 to n do

22. Label[b]←vote(k_idx[k·b−k+1]，…，k_idx[k·b])；

23. end

24. 将 Label 发送给 Client；

Client：

25. output Label。

首先，Client 打包加密自己持有的隐私数据 Data，加密后的密文组记为 $\boldsymbol{A}$，与前面的协议一样，类矩阵的一行为一个密文。其中，$\boldsymbol{A}$ 的第 i 行的第 1 组元素为所有待预测样本的第 i 个特征，然后每一行将其第 1 组元素重复填充到明文槽共 m 次后结束填充，此时密文的明文槽恰好被填满。

$$\boldsymbol{A}=\begin{bmatrix} Enc(x_{11},x_{21},\cdots,x_{n1},x_{11},\cdots,x_{n1},\cdots,x_{n1}) \\ Enc(x_{12},x_{22},\cdots,x_{n2},x_{12},\cdots,x_{n2},\cdots,x_{n2}) \\ \vdots \\ \vdots \\ Enc(x_{1u},x_{2u},\cdots,x_{nu},x_{1u},\cdots,x_{nu},\cdots,x_{nu}) \end{bmatrix}$$

然后，Server 按照如下形式打包自己持有的训练集数据 $TDATA$，$TDATA$ 只打包，不加密，打包后的数据记为 T，同理 T 的一行为一个明文。其中，T 的第 i 行第 1 组元素是将所有训练集样本的第 1 列第 i 个元素重复填充 n 次后得到，第 i 行第 2 组元素是将所有训练集样本的第 2 列第 i 个元素重复填充 n 次后得到……以此类推，直到 m 个训练集样本的相应元素全部被重复填充 n 次，此时 T 的明文槽也正好被填满。

$$T=\begin{bmatrix} y_{11},\cdots,y_{11},y_{21},\cdots,y_{21},\cdots,y_{m1},\cdots,y_{m1} \\ y_{12},\cdots,y_{12},y_{22},\cdots,y_{22},\cdots,y_{m2},\cdots,y_{m2} \\ \vdots \\ \vdots \\ y_{1u},\cdots,y_{1u},y_{2u},\cdots,y_{2u},\cdots,y_{mu},\cdots,y_{mu} \end{bmatrix}$$

接下来，为了得到距离差集 DS，先计算 D_1'。D_1' 是将 A 与 T 使用距离公式(5)或(6)进行计算，之后再将每一行相加后得出的。D_1' 内元素的计算结果记为 $dist$，使用符号“·”来代表距离运算。其计算过程如下：

$$A\cdot T=$$

$$\begin{bmatrix} Enc(x_{11}\cdot y_{11},\cdots,x_{n1}\cdot y_{11},\cdots,x_{11}\cdot y_{m1},\cdots,x_{n1}\cdot y_{m1}) \\ Enc(x_{12}\cdot y_{12},\cdots,x_{n2}\cdot y_{12},\cdots,x_{12}\cdot y_{m1},\cdots,x_{n2}\cdot y_{m2}) \\ \vdots \\ \vdots \\ Enc(x_{1u}\cdot y_{1u},\cdots,x_{nu}\cdot y_{1u},\cdots,x_{1u}\cdot y_{mu},\cdots,x_{nu}\cdot y_{mu}) \end{bmatrix}$$

$$D_j=(dist_{1j},dist_{2j},\cdots,dist_{nj}),j=1,2,\cdots,m$$

所以 $D_1'=(dist_{11},\cdots,dist_{n1},dist_{12},\cdots,dist_{n2},\cdots,dist_{nm})=(D_1,D_2,\cdots,D_m)$。

之后，Server 对 D_1' 进行密文旋转操作，得到 D_2'，D_3'，…，D_m'。D_2' 由 D_1' 向左旋转 n 个槽得到，D_3' 由 D_2' 继续向左旋转 n 个槽得到，以此类推。由于 D_j 的长度正好为 n，因此在旋转后 D_i' 看上去仅移动了一位。

$$D_2'=Rotate(D_1',-n)=Enc(D_2,D_3,\cdots,D_m,D_1)$$

$$D_3'=Rotate(D_2',-n)=Enc(D_3,D_4,\cdots,D_1,D_2)$$

$$\vdots$$

$$\vdots$$

$$D_m'=Rotate(D_{m-1}',-n)=Enc(D_m,D_1,\cdots,D_{m-2},D_{m-1})$$

最后，将 D_1' 分别与 D_1'，D_2'，…，D_m' 相减，得到的密文组就是 DS。

接下来，Server 将 DS 发送给 Client，Client 以 DS 为输入执行协议 3 后会将每个待预测样本的 k 个最近邻在 $TDATA$ 中的索引列表 k_idx 发送给 Server。

Server 在接收 k_idx 之后就将其元素以 k 个为一组，取出这 k 个索引对应的训练集样本，然后统计这 k 个训练集样本中出现最多的标签 L_a，并将 L_a 作为待预测样本 a 的预测结果，然后重复执行上述操作直到所有待预测样本预测完毕。

最终，Server 将所有预测标签 Label 发送给 Client，协议结束。

综上所述，协议 4 的正确性得证，Server 与 Client 在分类阶段额外交互一轮来运行协议 3 进行距离值的比较操作，这样方可完成整个分类阶段。运行协议 4 能够同时分类 n 个待预测样本。

3.4.2 协议 3 与协议 4 的安全性分析

由于协议 4 较为复杂，且在执行协议 4 时还需要执行

协议 3，因此先分析协议 3 的安全性。

协议 3 的安全性证明如下：

(1)尽管执行协议 3 的主体为 C，但输入由 S 提供，且输出发送给 S。在执行协议 3 时，C 与 S 的视角为：

$V_C=(sk,pk,Pt\_S;DS;k\_idx)$

$V_S=(pk,DS;k\_idx;)$

(2)向模拟器 S_C 输入(sk,pk,Pt_S)后，使用 pk 将 Pt_S 加密后得到 DS'。

因此，S_C 的输出为：

$S_C=(sk,pk,Pt\_S;DS')$

由于 BGV 方案具有语义安全与电路安全，因此 DS 与 DS'之间是不可区分的，因此 $V_C\equiv_c S_C$。

(3)向模拟器 S_S 输入(pk,DS)之后，由于 S_S 收到的消息 k_idx 是不含有隐私数据的明文信息，因此易得 $V_S\equiv_c S_S$。

综上所述，协议 3 在半诚实模型下是安全的。

接下来给出协议 4 的安全性证明。

(1)在执行协议 4 时，C 与 S 的视角为：

$V_C=(pk,sk,Data,Pt\_S;DS,Label;k\_idx,Label)$

$V_S=(pk,TDATA;A,k\_idx;DS,Label)$

(2)(向模拟器 S_C 输入$(pk,sk,Data,Pt_S)$后，使用 pk 加密 Pt_S 后得到 DS'。因此，S_C 的输出为：

$S_C=(pk,sk,Data,Pt\_S;DS')$

由于 BGV 方案具有语义安全与电路安全，因此 DS 与 DS'之间是不可区分的，因此 $V_C\equiv_c S_C$。

(3)向模拟器 S_S 输入$(pk,TDATA)$后，使用 pk 按照行方式打包加密 $TDATA$ 后得到 DS'，然后加密 u 个不同的随机数，得到 u 个密文，并按照先后顺序存入同一个密文组后得到 A'。因此，S_S 的输出为：

$S_S=(pk,TDATA;A';DS')$

与前面同理，A 与 A'之间、DS 与 DS'之间均不可区分，因此 $V_S\equiv_c S_S$。

综上所述，协议 4 在半诚实的安全模型下是安全的。

4 实验结果与分析

实验使用的计算机配置为：CPU 为 Intel(R) Core(TM) i7-10710U CPU @ 1.10 GHz，6 核 12 线程；内存为 16 GB；操作系统为 Windows10 与基于 WSL(Windows Subsystem for Linux)的 Ubuntu 20.04.2 LTS。所有实验使用的数据集均取自 UCI 公开数据集，且所有实验的安全参数 λ 均大于 120。

4.1 超平面分类器

我们选用了 breast cancer，iris 以及 balance-scale 数据集用于该实验。其中 breast cancer 与 balance-scale 数据集已经去除了带有“?”的数据行，iris 数据集则将全部数据乘以 10 以化整，并且去掉了后 50 条数据用于做二分类的预测。这 3 个数据集均按照 8∶2 的比例随机划分为训练集与测试集。在 python 平台实现了这 3 个数据集的模型训练，并将模型参数提取到 C++平台上以便使用。实验采用的方案参数如表 1 所列，实验结果如表 2 所列。

表 1 超平面分类器实验采用的方案参数

Table 1 Parameters in hyperplane experiment

data set	p	$\log q$	$Nslots$	N
balance-scale	163	300	3000	24000
breast-cancer	1249	400	2500	25000
iris	491	300	2800	28000

表 2 超平面分类器的实验结果

Table 2 Experiment results by hyperplane classifier

data set	balance-scale	breast-cancer	iris
Num of predict	116	136	20
precision rate/%	93.10	97.79	100.0
ciphertext evaluate time/s	0.117	0.225	0.117
ciphertextcompare time/s	5.101	21.955	11.369
total time/s	5.22	22.18	11.49
real amortized time/ms	44.98	163.08	574.30
ideal amortized time/ms	1.74	8.87	4.10

表 2 以及后续几个表中的 real amortized time 为实际分摊计算时间，其计算式如式(7)所示，ideal amortized time 为理想分摊计算时间，其计算式如式(8)所示。实际分摊计算时间使用待预测样本数(num of predicted)作为分母，用于评估分类一个样本在实际情况下所需的平均时间。理想分摊计算时间使用明文槽数(Nslots)作为分母，用于评估在明文槽全部被填满的假想(理想)情况下分类一个样本所需的平均时间。

$$real\ amortized\ time=\frac{total\ time}{num\ of\ predicted} \tag{7}$$

$$ideal\ amortized\ time=\frac{total\ time}{Nslots} \tag{8}$$

实验结果表明，平均每预测一个样本，需要花费几十毫秒到几百毫秒不等，且由于 iris 数据集的待预测样本较少，模型不容易预测出错，因此其准确率为 100%。因此，维持现有的方案参数不变，在待预测样本数不超过明文槽数 Nslots 的情况下，待预测样本数越多，我们的协议分类效率就会越高。

4.2 决策树

在决策树的实验中选用了 car，nursery 以及 breast cancer 数据集。其中 nursery 数据集事先去除了占比很低的两类数据，即 recommend 以及 very_recom 分类；breast cancer 数据集与超平面分类器使用的是同一个数据集。这 3 个数据集同样按照 8∶2 的比例随机划分为训练集与测试集，并将训练出来的决策树模型转换为了等效多项式。

我们引入了 OpenMP[27] 以实现并行的密文比较，并行的加速比由 CPU 的性能以及开启的线程数决定，当 CPU 的使用率达到 100%之后，开启更多的线程也不会明显提高加速比。表 3 列出了 nursery 数据集的并行密文比较耗时的实验结果，该实验结果呈现了总密文比较时间与线程数的变化。当线程数超过 6 之后，总比较时间并没有明显下降，此时 CPU 使用率显然已经接近或者达到了 100%。此外，其加速比之所以没有达到理想状态，除了 CPU 性能的影响之外，还包括线程之间的调度、任务划分以及各个线程的负载不均衡等原因。

表 3 nursery 数据集的并行密文比较总时间

Table 3 Total time of parallel comparison on nursery data set

threads	compare time/s
1	40.11
2	25.95
4	19.93
6	16.82
8	16.97
10	14.52
12	14.51

决策树实验采用的方案参数如表 4 所列,每个数据集分别拥有 1～3 个仅密文乘法次数不同但分类结果完全相同的等效多项式,代表多项式的罗马数字越大,该多项式的密文乘法次数就越少,但其最大乘法深度越大。

表 4 决策树实验采用的方案参数

Table 4 Parameters in decision tree experiment

data set	poly(multiplication times)	p	log$q$	$Nslots$	N
car	I(23)	13	180	3120	18720
	II(19)		235		
	III(15)		240		
nursery	I(21)	13	240	3120	18720
	II(17)		240		
breast cancer	I(4)	29	240	2970	17820

决策树的实验结果如表 5 所列,从实验结果可以看出,等效多项式的乘法次数与最大乘法深度会共同影响整个分类计算的效率,同时,它也和超平面分类器一样,在尽可能将待预测样本填满明文槽的情况下能够更高效地进行分类。

表 5 决策树的实验结果

Table 5 Experiment results by decision tree

data set	poly	Num of predict	precision rate/%	total time/s	real amortized time/ms	ideal amortized time/ms
car	I	346	93.93	8.45	24.42	2.72
	II			8.39	24.26	2.69
	III			9.27	26.80	2.97
nursery	I	2526	95.17	10.20	4.04	3.27
	II			9.82	3.89	3.15
breast cancer	I	136	97.08	4.07	29.89	1.37

4.3 KNN 分类器

在 KNN 分类器的实验中,仅选用了 Autism-Adolescent 数据集用于实验,该数据集事先去除了少部分带"?"的数据行。KNN 实验采用的方案参数与实验结果如表 6 所列,表 6 分别统计了使用式(5)与式(6)来定义样本间距离的实验结果。实验中大部分的时间用于 Client 解密从 Server 接收到的中间计算结果以及密文旋转的操作。此外,由于数据集较小,因此会出现多个相等的距离值,相等的距离值过多会影响最后投票分类时的结果,这两种距离定义计算出的距离值的重复程度不同,因此两种距离定义方式的实验结果中的预测准确率会有不同。

表 6 KNN 分类器的实验结果

Table 6 Experiment results by KNN classifier

distance definition	formula(5)	formula(6)
data set	Autism-Adolescent	
Num of predict	20	
parameters(p,logq,Nslots,N)	(293,340,1560,18720)	
precision rate/%	100.00	90.00
evaluate D_1' time/s	3.63	0.17
$D_i' \leftarrow$Rotate(D_1') time/s	25.56	19.97
client decrypt middle result time/s	57.64	57.46
total time by both sides/s	87.61	78.35
amortized time/s	4.38	3.92

4.4 与其他文献的方案比较

表 7—表 9 列出了本文的 3 种协议分别与文献[16]、文献[17]和文献[19]对同一数据集进行分类的实验结果。实验的安全参数均大于 120。

在表 7 所列的对比结果中,我们事先对具有 30 个特征的 breast cancer(original)数据集做了主成分分析(PCA)降维后再对其进行训练,然后将训练后的模型应用到加密样本的分类计算。协议 1 与文献[16]相比在整体的运行时间上偏长,主要原因在于密文比较的效率较低。协议 1 为了密文比较有效,需要超平面分类器计算出的 y 值的绝对值必须小于 $p/2$,而 p 值越大则密文比较的效率会越低,但协议 1 的分类准确率是远高于文献[16]的。

在表 8 所列的对比结果中,协议 2 在分类效率上远高于文献[17],在分类准确率上相差不大。文献[17]与协议 2 类似的地方在于,待预测样本不超过上限的情况下,待预测样本越多,分摊计算效率就越高。

在表 9 所列的对比结果中,文献[19]的方案需要在两方之间进行大量的交互才能完成,而协议 4 仅需要两方额外交互一轮即可完成所有待预测样本的分类,但在分类效率上两者的差别不大。

表 7 超平面分类器的方案比较

Table 7 Schemes comparison of hyperplane classifier

scheme	data set	Num of predict	classification time/s	precision rate/%
paper[16]	breastcancer (original)	114	1.65	63.2～70.5
Ours			20.57	92.98

表 8 决策树的方案比较

Table 8 Schemes comparison of decision tree

scheme	data set	Num of predict	total time/s	average time/s	precision rate/%
paper[17]	Seismic-bumps	517	5200	10.06	89.01
ours			90.72	27.78	0.05

表 9 KNN 分类器的方案比较

Table 9 Schemes comparison of KNN classifier

scheme	data set	Num of predict	classification time/s	average time/s	both-side interactions
paper[19]	zoo	36	97.42	2.70	3219
Ours			95.34	2.65	4

结束语 本文基于 BGV 同态加密方案,分别结合 3 种经典的机器学习算法实现了半诚实安全的两方分类计算方案。本文方案能够很好地保护用户端的隐私数据,实现了高准确

率和高效率，且通过引入密文比较实现了较少交互的分类计算，基本能够满足所给的应用场景的需求。但本文方案对于密文比较的效率、密文解密的效率以及进行明文槽旋转操作的效率仍有待提高。

在未来的工作中，我们将探索同态加密在更多机器学习算法隐私保护中的应用，例如其他分类算法、聚类算法和高效密文训练等，并能够在实际的工程场景中应用。

参 考 文 献

[1] YANG Y P, ZHAO Y, ZHANG J M, et al. Recent Development of Theory and Application on Homomorphic Encryption [J]. Chinese Journal of Electronics & Information Technology, 2021, 43(2): 13.

[2] YAO C C. How to generate and exchange secrets[C]// 27th Annual Symposium on Foundations of Computer Science. 1986.

[3] SHAMIR A. How to share a secret[J]. Communications of the ACM, 1979, 22(11): 612-613.

[4] BLAKLEY G R. Safeguarding cryptographic keys[C]// Afips. IEEE Computer Society, 1979.

[5] RABIN M O. Transaction protection by beacons[J]. Journal of Computer and System Sciences, 1981, 27(2): 256-267.

[6] RIVEST R L, ADLEMAN L M, DERTOUZOS M L. On Data Banks and Privacy Homomorphisms[J]. Foundations of Secure Compuation, 1978, 4(11): 169-180.

[7] GENTRY C. A fully homomorphic encryption scheme[M]. Stanford University, 2009.

[8] BRAKERSKI Z, GENTRY C, VAIKUNTANATHAN V. Fully Homomorphic Encryption without Bootstrapping [J]. ACM Transactions on Computation Theory (TOCT) [J]. Special issue on Innovations in Theoretical Computer Science 2012—Part II, 2014, 6(3): 1-36.

[9] FAN J, VERCAUTEREN F. Somewhat practical fully homomorphic encryption[J]. IACR Cryptology Eprint Archive, 2012, 2012: 144.

[10] CHEON J H, KIM A, KIM M, et al. Homomorphic Encryption for Arithmetic of Approximate Numbers [C] // International Conference on the Theory and Application of Cryptology and Information Security. Cham: Springer, 2017.

[11] WANG C, YAO H N, WANG B N, et al. Progress in Quantum Computing Cryptography Attacks [J]. Journal of Computers, 2020, 43(9): 1691-1707.

[12] PAILLIER P. Public-Key Cryptosystems Based on Composite Degree Residuosity Classes[C]// Advances in Cryptology—EUROCRYPT '99, International Conference on the Theory and Application of Cryptographic Techniques. Berlin: Springer, 1999.

[13] YASUMURA Y, ISHIMAKI Y, YAMANA H. Secure Naive Bayes Classification Protocol over Encrypted Data Using Fully Homomorphic Encryption[C]// iiWAS2019: The 21st International Conference on Information Integration and Web-based Applications & Services. 2019.

[14] CAI Y L, TANG C M. Privacy of Outsourced Two-Party K-Means Clustering[J]. Concurrency and Computation-Practice & Experience, 2021, 33(8): 1-12.

[15] PARK S, BYUN J, LEE J. Privacy-Preserving Fair Learning of Support Vector Machine with Homomorphic Encryption[C]// The Web Conference. 2022: 3572-3583.

[16] JIA C F, WANG F Y, CHEN Y, et al. Machine Le-arning Algorithm for a Homomorphic Encrypted D-ata Set [J]. Journal of Tsinghua University (Science and Technology), 2020, 60(6): 456-463.

[17] DEY P, CHAULYA S K, KUMAR S. Secure decision tree twin support vector machine training and classification process for encrypted IoT data via blockchain platform [J]. Concurrency and Computation Practice and Experience, 2021, 33(16).

[18] UCI[OL]. https://archive.ics.uci.edu.

[19] XU J, WANG A D, BI M, et al. Privacy-preserving k-Nearest Neighbor Classifier [J]. Journal of Software, 2019, 30(11): 3503-3517.

[20] HElib[OL]. https://github.com/shaih/HElib.

[21] GENTRY C. Fully homomorphic encryption using ideal lattices [C]// ACM. ACM, 2009: 169-178.

[22] DUCAS L, STEHL É D, FISCHLIN M, et al. Sanitization of FHE Ciphertexts[C]// International Conferenceon Advances in Cryptology-eurocrypt. Berlin: Springer, 2016.

[23] ILIASHENKO I, ZUCCA V. Faster homomorphic comparison operations for BGV and BFV [J]. Privacy Enhancing Technologies, 2021, 3(2021): 246-264.

[24] CORTES C. Support-Vector Networks[J]. Machine Learning, 1995, 20(1995): 273-297.

[25] KHEDR A, GULAK G, VAIKUNTANATHAN V. SHIELD: Scalable Homomorphic Implementation of Encrypted Data-Classifiers [J]. IEEE Transactions on Computers, 2016, 65(9): 2848-2858.

[26] CHEN J Y, FENG Y, LIU Y, et al. Non-interactive Privacy-Preserving Nave Bayes Classifier Using Homomorphic Encryption [C]// International Conference on Security and Privacy in New Computing Environments. Cham: Springer, 2022.

[27] OpenMP[OL]. https://www.openmp.org/.

LU Xingyuan, born in 1997, postgraduate. His main research interests include homomorphic encryption and privacy protection.

CHEN Jingwei, born in 1984, Ph.D, associate researcher. His main research interests include symbolic computation and lattice based cryptography.

（责任编辑：喻藜）