**2023年全国大学生信息安全竞赛**

**作品报告**

**作品名称：基于同态加密的非交互式DBSCAN聚类隐私保护方案**

**电子邮箱： 2535210516@qq.com**

**提交日期： 2023年6月14日**

填写说明

1. 所有参赛项目必须为一个基本完整的设计。作品报告书旨在能够清晰准确地阐述（或图示）该参赛队的参赛项目（或方案）。

2. 作品报告采用A4纸撰写。除标题外，所有内容必需为宋体、小四号字、1.5倍行距。

3. 作品报告中各项目说明文字部分仅供参考，作品报告书撰写完毕后，请删除所有说明文字。(本页不删除)

4. 作品报告模板里已经列的内容仅供参考，作者可以在此基础上增加内容或对文档结构进行微调。

5. 为保证网评的公平、公正，作品报告中应避免出现作者所在学校、院系和指导教师等泄露身份的信息。

**目 录**

[摘要 5](#_Toc137678688)

[关键词 6](#_Toc137678689)

[第一章 作品概述 7](#_Toc137678690)

[1.1背景分析 7](#_Toc137678691)

[1.2相关技术 10](#_Toc137678692)

[1.2.1符号介绍 10](#_Toc137678693)

[1.2.2聚类 11](#_Toc137678694)

[1.2.3 LWE和RLWE 14](#_Toc137678695)

[1.2.4 LUT算法 16](#_Toc137678696)

[1.2.5 CKKS算法 18](#_Toc137678697)

[1.3研究现状 20](#_Toc137678698)

[1.3.1全同态加密方案 20](#_Toc137678699)

[1.3.2基于同态加密的聚类算法 21](#_Toc137678700)

[1.4本项目的工作 22](#_Toc137678701)

[1.5应用前景 24](#_Toc137678702)

[第二章 作品设计与实现 26](#_Toc137678703)

[2.1 系统框架 26](#_Toc137678704)

[2.2 设计动机 27](#_Toc137678705)

[2.2.1 全同态加密的应用 27](#_Toc137678706)

[2.2.2 DBSCAN聚类算法 28](#_Toc137678707)

[2.3方案设计 28](#_Toc137678708)

[2.3.1 距离测度的选取 28](#_Toc137678709)

[2.3.2 数据预处理加密上传 31](#_Toc137678710)

[2.3.3 密文运算 31](#_Toc137678711)

[2.3.4 客户端聚类 35](#_Toc137678712)

[2.4安全模型 36](#_Toc137678713)

[2.5正确性分析 37](#_Toc137678714)

[2.6安全性分析 37](#_Toc137678715)

[2.6.1 CKKS密文的安全性分析 37](#_Toc137678716)

[2.6.2 获取模型参数倒推明文的安全性分析 38](#_Toc137678717)

[第三章 作品测试与分析 39](#_Toc137678718)

[3.1测试方案 39](#_Toc137678719)

[3.2测试环境搭建与测试设备 39](#_Toc137678720)

[3.3测试数据与结果分析 39](#_Toc137678721)

[3.3.1 与现有方案的对比 39](#_Toc137678722)

[3.3.2不同数据规模下的聚类结果 40](#_Toc137678723)

[3.3.3不同数据规模的多维度对比 42](#_Toc137678724)

[第四章 创新性说明 44](#_Toc137678725)

[4.1基于PEGASUS框架的隐私保护方案 44](#_Toc137678726)

[4.2无协议交互 44](#_Toc137678727)

[4.3聚类速度显著提升 45](#_Toc137678728)

[4.4 基于python实现的跨平台应用程序 46](#_Toc137678729)

[4.5 更加适合同态加密的DBSCAN流程 49](#_Toc137678730)

[第五章 总结 50](#_Toc137678731)

[参考文献 52](#_Toc137678732)

# 摘要

在当下，云计算快速发展，用户将数据外包至云服务器并进行计算，计算完成后云服务器再将结果返回给用户。随着需要处理的数据量不断增大，单靠在本地端完成计算越来越困难，云计算已经逐渐发展成为主流。但云计算的安全问题也随之而来，用户希望云服务器能帮自己处理数据，但同时又不希望自己的隐私被他人获取，那么如何对发送到服务器端的数据进行隐私保护就成为了一大难题。目前隐私保护方案主要分为两个方向，一种是在通过可信信道传输，另一种是对数据进行加密再传输。第一种方案需要有可信的第三方，对第三方依赖较大，而且云服务器也有可能去窃取隐私数据。第二种方案现在有多种实现方案，其中一种较为主流的就是同态加密。

同态加密就是对数据进行加密，加密后的数据具有一种特性就是：在密文上做的运算等价于在明文上做相同的运算。而由于明文转化为密文后不具有有序性，所以大部分的同态加密方案在遇到如何解决在密文上无法直接进行比较这个问题时，他们都选择使用协议交互，先将密文发送给客户端，客户端解密后进行比较，最后再将比较结果发送给服务器端。而本方案创新性地提出了一种不需要协议直接在服务器端进行比较的新方案。

我们提出了一种在PEGASUS框架下的同态加密的DBSCAN (Density-Based Spatial Clustering of Applications with Noise) 聚类隐私保护方案。基于PEGASUS框架，我们的DBSCAN聚类隐私保护方案可以做到不需要协议交互，运用 LUT (Look Up Table) 算法，直接在服务器端以密文的形式进行大小比较，云服务器将客户端传输来的CKKS (Cheon-Kim-Kim-Song)密文(基于RLWE (Ring Learnig with Error)) 转化为FHEW密文(基于LWE (Learnig with Error))，然后通过计算查找表的方式完成密文的大小比较功能，比较完成后再转化为CKKS形式的密文，这样服务器端就可以使用CKKS算法快速并行地计算距离，并且通过密文转换，使用LUT进行密文比较计算，实现在服务器端以密文的形式进行比较的操作了。同时，由于DBSCAN聚类算法本身不适用于同态加密，我们修改了DBSCAN算法的部分设定，增加了距离矩阵和加入判断是否在邻域内的标签等，使其更加适用于同态加密算法。并以此为思路，尽可能去减少真正需要去计算的数据点，降低时间复杂度。通过计算数据点所属的网格单元，减少用户需要去解密的数据点的个数，从而减少在客户端的解密耗时。

经过理论分析和实验证明，本文方案在能够在有效地保护用户隐私安全的基础上，实现较高的聚类准确性，不再需要协议交互，能够实现在客户端离线状态下，服务器端也可以进行外包聚类计算，很大程度上降低了通讯开销。

# 关键词

隐私保护，同态加密，DBSCAN聚类，密文转换，密文比较，网格单元

# 第一章 作品概述

## 1.1背景分析

随着云计算的迅速发展，用户可以采用多样化的方式将数据集上传至云服务器，由服务器进行相关操作，从而降低用户所需的计算资源。然而，由于用户数量众多，他们所提供的数据类型也涉及多种领域，例如电子邮件信息、后台用户数据库信息、个人健康状况等，这些数据包含着个人隐私或商业机密，因此在进行外包处理时，泄露敏感信息的风险不容忽视。因此，如何对这些敏感数据中的隐私进行有效的保护，已成为当前研究领域的热点问题。

**（1）云计算的快速发展与广泛应用**

在过去十多年里，云计算已经从一种备受质疑的技术概念，发展成为新一代IT标准，并深刻影响了整个ICT产业的商业模式。2006年，IBM和谷歌共同推出了云计算概念，而2007-2009年间，Salesforce推出了PaaS (Platform-as-a-Service)服务Force.com，谷歌则推出了Google App Engine，并逐渐涌现出各种类型的云服务。在2009-2016年间，云计算功能不断完善，类型也愈加多样化，传统企业开始通过自身能力扩展或收购等方式积极投入云计算服务市场。2016-2022年，随着市场的激烈竞争，主流平台产品和标准化产品的功能愈加健全，市场格局相对稳定，云计算进入了成熟阶段。未来，云计算有望在更广泛的领域发挥作用，并创造出更多形式的服务和丰富的应用场景。（如图1.1所示）

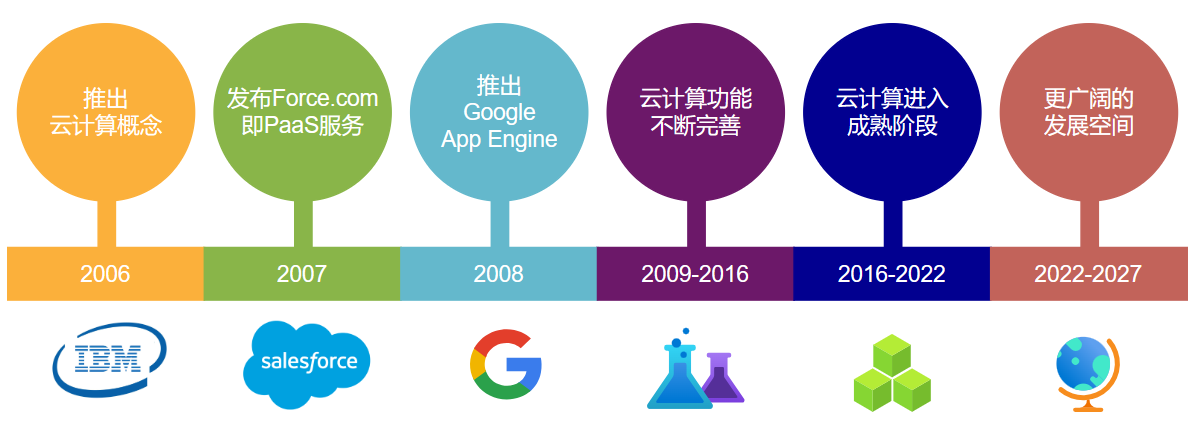


图1.1 云计算的发展历程

**（2）云计算面临的安全问题**

随着云计算的普及，其面临的安全问题也日益突出。云计算中大量的敏感数据可能会因为各种因素而被泄露，如未经授权的访问、数据存储在不安全的地方等。在云计算中，身份验证的安全性至关重要，如果未经授权的人可以访问云中的数据，将会给用户和组织带来巨大的风险。加密在保护云中的数据方面起着至关重要的作用。如果云中的数据没有加密，攻击者可以轻松访问敏感信息。云服务提供商的服务可用性和可靠性对用户非常重要。如果云计算中的服务不可用，会影响用户的生产力和业务运作。

**（3）云计算隐私保护策略**

通常有两种常见的隐私保护方式：第一种方式是利用可信信道，如硬件或可信第三方，来确保在数据传输过程中数据不被窃取。第二种方式则是在数据传输前对数据进行加密，即使攻击者获得密文，也无法解密数据，从而确保数据的隐私安全。虽然第一种方式可以保证数据传输过程中的安全性，但它对于第三方的依赖较大。而且，由于它采用明文传输，一旦数据集传输到云服务器端，仍然存在被窃取的风险。相比之下，第二种方式通过数据加密来确保数据的安全传输，不论在数据传输过程中还是在服务器端的计算过程中，都能够保护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（Homomorphic Encryption, HE）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（Fully Homomorphic Encryption, FHE）算法、支持无限次数和有限种类的部分同态加密（Partially Homomorphic Encryption, PHE）算法、支持有限次数和多种运算的浅同态加密（Somewhat Homomorphic Encryption, SWHE）算法[1]-[4][2][3] **错误!未找到引用源。错误!未找到引用源。**。

聚类算法是数据挖掘和机器学习领域中的重要算法之一，广泛应用于各种领域，如医疗、金融、社交网络等。但是，传统聚类算法通常需要直接访问原始数据，存在数据隐私泄露的风险。同态加密能同时保护数据的保密性和允许对密文进行计算，因此在隐私保护领域具有广泛的应用。

同态加密基于的外包计算隐私保护模型（如图1.2所示）：用户对需要外包的数据集进行加密并上传至服务器（①），由服务器执行较高计算复杂度的数据处理，然后将结果返回给用户（②）。

图示

描述已自动生成

图1.2 基于同态加密的外包计算的隐私保护模型

**（4）机器学习外包计算的隐私保护**

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。机器学习算法通常可分为有监督和无监督两种类型。有监督型机器学习算法通过提取训练数据集的特征和真实标签来建立模型，对测试数据进行分类和预测，常见的算法包括贝叶斯分类器、超平面分类器和决策树分类器等。无监督型机器学习算法则不需要预先训练模型，常用的算法有K-means (K-means clustering algorithm)聚类算法和 DBSCAN (Density-Based Spatial Clustering of Applications with Noise)基于密度的聚类算法等。

DBSCAN算法可以在有噪声点的情况下找到任意形状的聚类簇，相较于另一种常见的聚类算法K-means，其应用范围更广，例如在推荐系统等高级系统中的实现。此外，DBSCAN还可与加速范围访问（如树）相结合来设计一种数据库结构。

因此，基于同态加密的DBSCAN聚类隐私保护方案是一种非常有前景的算法，它可以在保护数据隐私的同时实现聚类分析，具有较高的实用价值。但是，同态加密技术在计算效率和密钥管理等方面存在一定的挑战和限制，需要在实际应用中进行仔细考虑和权衡。

**（5）云计算的隐私问题深受国家重视**

我国高度重视数据安全流通技术的发展应用，过去一年内多个部门密集出台了一系列战略、规划和政策，强调数据要素流通的重要性，提出数据安全流通的建设方案。2022年1月国务院办公厅印发的《要素市场化配置综合改革试点总体方案》中提出，探索“原始数据不出域、数据可用不可见”的交易范式。12月19日，中共中央、国务院发布的《关于构建数据基础制度更好发挥数据要素作用的意见》提出要建立数据产权分置的产权运行机制，建立合规高效的数据要素流通和交易制度，从数据产权、流通交易、收益分配、安全治理等四项制度提出二十条政策举措，初步形成我国数据基础制度的“四梁八柱”。

隐私计算技术作为保障数据安全流通的有效方式，在政策驱动和市场需求同时作用下，乘时乘势高速发展，已逐渐成为促进数据要素跨域流通和应用的核心技术，广泛应用于金融、政务、医疗、能源、制造等诸多领域。

基于同态加密的云计算方案符合“原始数据不出域、数据可用不可见”的范式，为使用云计算的用户同时提供了便利与数据隐私保障，同时能为数据安全流通的建设助一份力。

## 1.2相关技术

### **1.2.1符号介绍**

表示：一个模的系数为整数的多项式环。表示：一个模的系数为整数模的多项式环。表示中的环元素，表示的第系数。表示环元素的乘法。粗体小写字母表示向量，表示的第分量。表示向量分量的left-hand-side rotation。表示向量的内积，表示向量的Hadamard积。粗体大写字母表示矩阵，表示的值。表示集合，其中。

表示：在秘密下，消息的可能LWE加密的集合。，其中，并从误差分布中选择误差。对于LWE密文的解密公式是：。同样地，表示：在密钥下，消息的可能RLWE加密的集合。，其中，误差是从误差分布中独立抽样选择其系数。为了简化符号，我们将RLWE的密钥标识为向量，其中。

的环结构允许将实向量编码为的环元素。我们使用来表示具有缩放因子的编码，并使用表示具有缩放因子的解码。

### **1.2.2聚类**

聚类(Clustering)是按照某个特定标准(如距离)把一个数据集分割成不同的类或簇，使得同一个簇内的数据对象的相似性尽可能大，同时不在同一个簇中的数据对象的差异性也尽可能的大。也即聚类后同一类的数据尽可能聚集到一起，不同类数据尽量分离。

数据聚类方法主要可以分为划分式聚类方法(Partition-based Methods)、基于密度的聚类方法(Density-based methods)、层次化聚类方法(Hierarchical Methods)等。

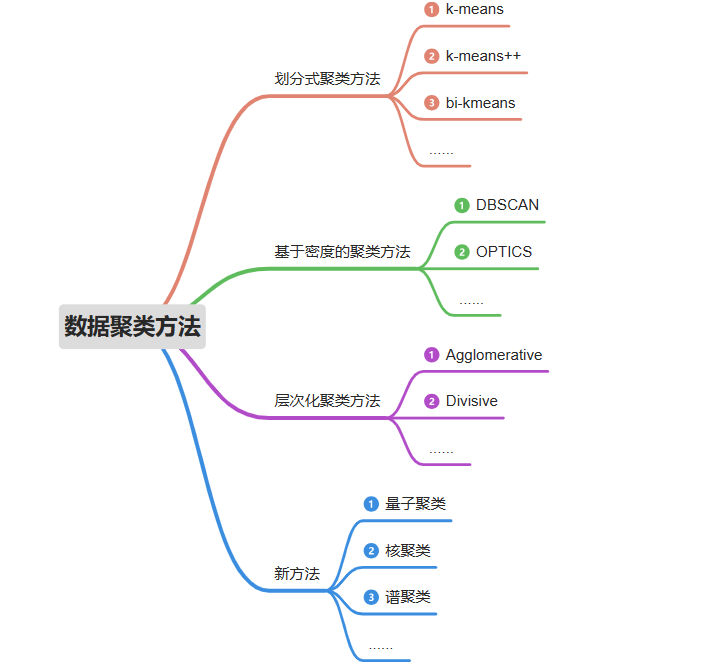


图1.3 数据聚类方法

DBSCAN聚类算法是一种很典型的具有噪声的基于密度的聚类方法。和K-Means这种一般只适用于凸样本集的聚类相比，DBSCAN既适用于凸样本集，又适用于非凸样本集。它将簇定义为密度相连的点的最大集合，能够把具有足够高密度的区域划分为簇，并可在噪声的空间数据库中发现任意形状的聚类。其中PPDBSCAN (Privacy Preserving DBSCAN)在不同类型的数据集上表现良好，运行时间较短。DBSCAN是基于一组邻域来描述样本集的紧密程度的，参数用来描述邻域的样本分布紧密程度。其中，描述了某一样本的邻域距离阈值，MinPths描述了某一样本的距离为的邻域中样本个数的阈值。假设样本集是，则DBSCAN具体的密度描述定义如下：

（1）邻域：对于，其中邻域包含样本集中与的距离不大于的子样本集，即，这个子样本集的个数记为

（2）中心点：对于任意样本，如果其邻域对应的至少包含MinPths个样本，即如果，则是中心点。

（3）密度直达：如果位于的邻域中，且是中心点，则称由密度直达。反之不一定成立，即不能说由密度直达，除非也是中心点。

（4）密度可达：对于和，如果存在样本序列，满足,且由密度直达，则称由密度可达。也就是说，密度可达满足传递性。此时序列中的传递样本均为中心点，因为只有核心对象才能使其他样本密度直达。

（5）密度相连：对于和，如果存在核心样本，使和均由密度可达，则称和密度相连。

|  |
| --- |
| **算法1：DBSCAN算法** |
| **输入**：样本集，参数，距离度量方式 |
| **输出：**聚簇的划分 |
| 1: 初始化核心对象集合，初始化聚簇数，初始化未访问样本集合，聚簇划分 |
| 2: |
| 3: 通过距离度量方式，找样本的邻域子样本集 |
| 4: 若子样本集个数满足，将样本加入到核心对象集合 |
| 5: 如果核心对象集合，则算法结束，否则进入步骤6 |
| 6: 在核心对象集合中，随机选择一个核心对象，初始化当前簇核心队列，初始化类别序号，初始化当前簇样本集合，更新未访问的样本集合 |
| 7: 如果当前簇核心对象队列，则当前聚类簇生成完毕，更新簇划分,更新核心对象集合转入步骤5。否则，更新核心对象集合 |
| 8: 在当前簇核心对象队列，中选取一个核心对象，通过邻域距离阈值找出所有的邻域子样本集，令，更新当前簇样本集合，更新未访问的样本集合，更新，转入步骤7 |
| 9: 输出结果：簇划分 |

如图1.4所示，假定，红色的点都是核心对象，因为其邻域至少有5个样本。黑色的样本是非核心对象。所有核心对象密度直达的样本在以红色核心对象为中心的超球体内，如果不在超球体内，则不能密度直达。图中用绿色箭头连起来的核心对象组成了密度可达的样本序列。在这些密度可达的样本序列的邻域内所有的样本相互都是密度相连的。

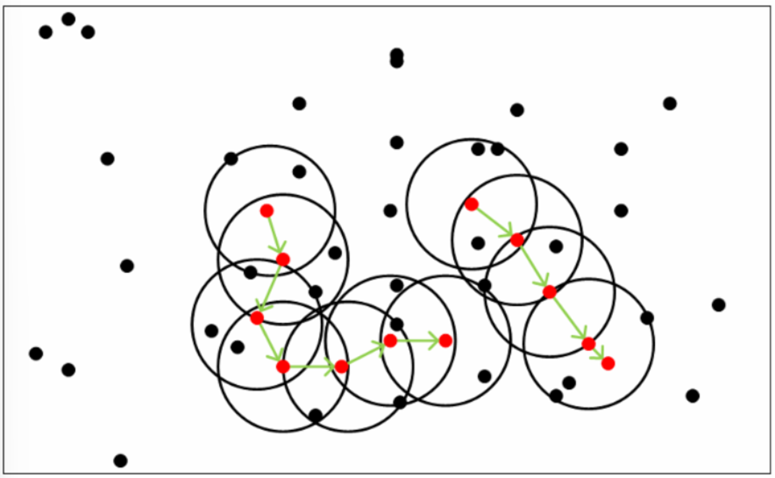


图1.4 密度直达、可达、相连

### **1.2.3** LWE和RLWE

（1） LWE (Learing with Error)问题

一旦量子计算机投入应用，现有的绝大多数公钥密码算法（这些公钥密码体制的安全性均依赖于数学难题的困难性，如大整数分解和离散对数求解）都能被计算量大且稳定的量子计算机攻破，1997年出现的基于格的密码体制由于在安全性、公私钥的尺寸、计算速度上达到了很好的平衡，被认为是最有前景的后量子密码算法之一。带误差学习LWE就是格密码中的一个难题。

对于上面的的矩阵求解问题，给定矩阵和向量，我们可以通过高斯消元法和初等变换的方式求解出未知数，但是当我们给原来的公式添加一个固定数值范围内随机采样的一个噪声向量时，再用刚才的求解方法对每一行进行消元的时候就会把噪声也带进去，无法计算出未知数，这就是一种简单的带误差学习问题。我们需要找到一组系数（未知数）使得一组基向量（矩阵）的线性组合尽可能逼近我们想要的目标向量，而噪声向量就可以定义我们需要距离目标向量多近。

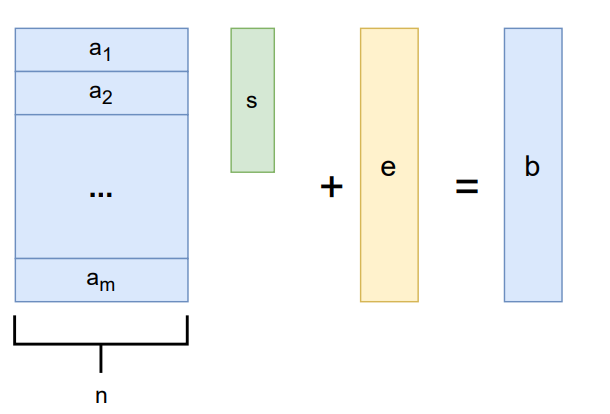


图1.5 LWE问题

如图1.5所示，在 LWE 问题中，蓝色的部分是均匀随机的（或与均匀随机不可区分），是私钥，是不公开的部分。简单来说，就是一个均匀随机的矩阵，乘上服从某个分布的私钥，再加上一个服从某个分布的错误扰动项，计算得到的与均匀随机不可区分。这种通过给定矩阵以及带有误差乘积,，来还原出未知向量的过程就是搜索LWE问题。

其中为素数有限域，矩阵是一个的随机模矩阵，是一个尽量大的素数，是个随机模元素组成的向量，则是由上限为的个元素组成的向量。

被称为整个LWE问题的安全参数（Security Parameter）。如果一个系统中的未知变量越多（即越大），那么整个LWE问题会越难。如果可以用的方程组越多（即越大），那么代表整个LWE问题会越简单。误差上限需要比小很多，<<。误差越小找到正确的解相对来说越简单。

而在密码学中要证明一个困难问题安全的时候一般会用到LWE决策版本。

（2）RLWE (Ring Learning with Error)问题

RLWE问题是LWE问题在环上的版本，不同的是和的选取是在多项式环上。多项式环，表示每次计算后都要对多项式系数模，对多项式模。它其中的每个元素都是一个多项式，每一次操作都相当于对多个元素进行操作，也就是能够一次加密多个比特的明文，对比LWE每次仅能对一个比特操作来说能够大大提高效率满足实际需要。

同样，RLWE也有两个版本，即search和decision version。先定义一些参数：安全参数，模，其中是2的幂次，，是上的错误概率分布。随机选取,,噪声，令，其中。

搜索性RLWE为：给定个，求解出秘密元素。

判定性RLWE为：区分个随机均匀选取自的和个，其中,噪声。

### **1.2.4 LUT算法**

查找表（Lookup Table，LUT）算法的核心思想是通过构造查找表，来构造从输入值到输出值的映射，这个映射是一个离散函数，从而化简每次都需要进行复杂的计算步骤，提高转换的效率。LUT算法提出后被作为滤镜广泛应用到图像处理领域。

LUT应用到LWE问题上，传统的Ducas等方法适用于较小的模数（一般只取到），而涉及到基于同态加密的隐私保护机器学习来说对模数的要求很大（）小模数的LUT算法不再适用，因此PEGASUS对之前的LUT算法进行来改进，使其满足较大模数的情况。PEGASUS改进的LUT算法的核心思路是通过引入可以接受的近似误差牺牲一定的精度来实现对较大明文下较大模数的支持。

算法的形式化表达如下所示：

|  |
| --- |
| **算法2：PEGASUS.LUT()算法** |
| **输入：**，，EK .  其中，是在放缩比例下对明文的在LWE下的加密的结果，对放缩后满足；是查找表函数，即；EK是RGSW在密钥下的密文集合。其中，，. . |
| **输出：**  其中，即为放缩的带近似误差的LWE加密后的集合. |
| 1: 取，其中. 定义多项式：  2: 取，.  3: 初始化.  4: **for**  **do**  5:    6: **end for** |
| 7: 输出 |

是将输入的对明文放缩的结果在较低维度的密钥上可能的LWE加密后的集合和LUT的映射表，输出的结果是对明文经过LUT后放缩的带近似误差的LWE加密后的集合。

具体来说，对而言，由于RLWE方案支持只对的模运算，而大模数，首先需要将模数放缩到，其中，得到，如取。这时就可以到的近似解密公式：

（1.1）

同样对查找表也需要进行放缩到，这时输入的范围变为. 基于此，原有的LUT在放缩后较宽的界上满足同态。其中引入的近似误差被确定在的范围内，其中是用来描述函数光滑性的参数（如可取为Lipschitz连续条件），已经有研究表明当的光滑性满足一定的阈值这样引入的误差带来的影响很小[5]。之后在密文上递推做同态乘法后和初始矩阵做同态加法，最终输出结果，即对明文经过LUT后放缩的带近似误差的LWE加密后的集合。

### **1.2.5 CKKS算法**

2017年Cheon等提出了一种支持近似计算的同态加密方法[6]CKKS (Cheon-Kim-Kim-Song)，其具体构造基于BGV方案，但也可以依赖于其他现有的同态方案。不同于以往同态加密算法中所追求的解密结果和明文完全一致，CKKS算法的目标是做近似计算。这并不偏离需求，因为现实生活中大部分运算，面对的是实数（复数），而实数（复数）的运算，往往只需要保留一部分有效数字即可。此外，允许误差，放宽准确性的限制，使得CKKS对比于其他基于LWE/RLWE难题的同态方案，细节有了较大的简化，计算效率也有了很大提升。

对于一般基于RLWE难题的同态算法，明文空间即多项式商环，CKKS的明文空间则是复数向量空间。因此我们需要找到后者和前者的映射关系，使得一个复数向量能被一个多项式来代表，进一步可以作为加密的输入；反之解密之后的结果也能还原出复数向量。以作为明文空间的好处，首先是从密码学原语的层次支持了浮点数；其次有个维度都可以放置数值，能方便地实现单指令多数据 (Single Instruction Multiple Data, SIMD)。实际应用中有相当多的基于向量的运算，例如求和、点积等，此时只需要对密文进行一次操作即可，不必分成各个分量单独运算。

CKKS作者提出了利用环同构的明文编解码方法。首先看编码过程：将多项式商环记作，记作。定义如下映射：规范嵌入映射：,,这里代表了多项式的N个本原根。的子环。自然投影映射。

首先取出元素，进行扩展，得到。未来保证后续精度，乘以缩放因子，放大整数部分。使用，得到。CKKS采用了一种坐标随机舍入的方法，将多项式系数按照和相邻整数距离的远近分配概率，然后按这个概率决定向上取整还是向下取整。

综上，编码的过程可以表示为。

而解码过程首先用缩放因子的倒数乘，然后计算，最后计算。

下面介绍CKKS算法的主体：给定安全参数，选择两个整数的幂，分别在上为密钥、错误学习、加密设置分布, *,* 。对于一个基础整数和级别数量，以及一个基础模数，设置密文的模数。对于环，此处令，之后随机生成参数，输出：。

（1）

随机生成，并设置私钥。

随机生成和错误，设置公钥：。

随机生成和，设置评估密钥：。

（2）

随机生成以及。输出密文，且:

（1.2）

（3）

对于同一级别的密文，解密得到明文结果：

（1.3）

（4）

对于密文，其密文相加结果为：

（1.4）

（5）

对于密文：

（1.5）

（1.6）

其密文相乘结果为：

（1.7）

由于CKKS加密算法的同态性，使得系统模型中云服务器对密文的计算等同于在明文上的计算，一定程度上保证了用户的隐私安全。

## 1.3研究现状

云计算安全研究是近年来备受关注的热门领域之一。随着云计算中数据的存储和处理越来越多，数据泄露和隐私保护成为云计算安全面临的重要挑战。针对这一问题，研究者提出了很多数据隐私保护方案，如差分隐私、同态加密、多方安全计算等；同态加密方案受到很多研究人员的重视，现有方案在不断地被完善。

### **1.3.1全同态加密方案**

全同态加密是同态加密领域中的一个前沿研究方向。虽然该技术已经发展了多年，但学界尚未就全同态加密形成统一的定义。现有的FHE方案主要通过电路来构造，该方案的核心就在于其密文计算算法可以用电路的思想来理解。其中是一个电路集合，可以是一个函数，公钥可以用于计算密文，解密算法为：

（1.8）

将密文进行运算后解密的结果需要与明文做相同运算的结果的相同。

全同态加密由于支持任意深度的同态加、乘运算而被广泛运用。自从2009年Gentry首次构造出第一个全同态加密方案以来，全同态加密技术已经经历了三代的发展。绝大多数的全同态方案可以分为两类：bit-wise（如BFV[7]、BGV[8]和CKKS[9]）和word-wise（如FHEW[10]和TFHE[11]）。在word-wise方案中，将多个明文打包成一个同态密文[12]，可以进行大量的SIMD操作，进行密文多项式上的同态加、乘法操作，提高同态运算的速度和效率。然而，在实际应用中进行机器学习的隐私保护时，一些非多项式运算（如比较最大、最小值、sigmoid、RELU等激活函数以及除法）无法在同态密文上进行。因此，通常需要使用低次多项式来近似这些非多项式运算[13]-[14]。但是这样的近似通常会降低模型的准确度。

2020年6月5日，IBM发布了基于“HElib”同态加密库的工具包，该算法库支持BGV（Brakerski-Gentry-Vaikuntanathan）算法，BGV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。CKKS算法是一种近似计算同态加密算法，其具体构造基于BGV方案，同样受HElib库的支持。

### **1.3.2基于同态加密的聚类算法**

同态加密技术被广泛应用于机器学习隐私保护，是其中重要的一种应用场景。研究者们针对不同的应用场景、数据类型和机器学习算法，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[15]中实现了高效且安全的k近邻（kNN，k-nearest neighbor）算法。文献[16]-[17]实现了加密数据集上的同态K-means 聚类算法，但是该算法有较大的时间开销。文献[18]-[19][20]实现了同态加密下照片的特征匹配算法，该算法在将照片同态加密后，提取其密文中的特征，使用特殊的匹配算法在只有照片密文的情况下完成了照片的匹配。文献[21]实现了密态数据集上的统计学习算法。文献[22]-[23]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[24]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。文献[25]实现了上传加密数据并在云端完成DBSCAN聚类的方案，但该方案需要客户端与服务器端的持续交互。

DBSCAN算法可以在有噪声点的情况下找到任意形状的聚类簇，相较于另一种常见的聚类算法K-means，其应用范围更广，例如在推荐系统等高级系统中的实现。此外，DBSCAN还可与加速范围访问（如树）相结合来设计一种数据库结构。但目前关于DBSCAN算法同态加密工作较少，缺乏更多完善的加密方案。

## 1.4本项目的工作

随着机器学习和云计算的快速发展，越来越多的企业和组织将自己的数据上传至云服务器进行计算，计算完成后云服务器再将结果返回给用户。但这同时也带来了数据安全和隐私保护的问题，用户希望云服务器帮自己处理数据，而又不希望隐私被他人获取。基于同态加密的机器学习隐私保护方案应运而生，但现有方案中仍然存在问题，如同时计算多项式函数和非多项式函数的方法仍然缺乏。尽管借助安全多方计算来构造适合基于同态加密的聚类算法的协议能够解决该问题，但是随之而来的是方案的安全性问题和性能问题。敌手可以通过延展性攻击使得聚类计算结果无意义或者使得客户端无法分辨收到的结果是正确的还是敌手的干扰，而且繁琐的协议设计也会影响云服务器的处理效率，且双方必须遵守协议才能保证正确性和安全性。也可以在同态加密技术所不支持的计算上进行近似计算来解决该问题，但是有的运算可能不存在近似算法，且近似算法带来的误差也会导致结果的正确性受到一定的影响。此外，目前全同态加密效率比较低，使用全同态加密进行云计算远远没有达到应用的级别。而机器学习需要的数据一般较多、计算量很大，其效率是一个比较难解决的问题。

因此，基于现有的存在的问题，本项目构建了一个在PEGASUS框架下基于同态加密的DBSCAN聚类算法隐私保护方案，能够在提供数据安全保护的同时，有效解决上述问题。本项目基于PEGASUS的密文转换框架，对DBSCAN聚类的流程进行了修改，实现了不需要通过客户端和服务器的交互就能完成加密数据同态聚类的同时，确保了聚类分析结果和明文上聚类结果的一致性，以及用户上传数据的安全性和方案的高效性。

通过PEGASUS框架，服务器可以在CKKS密文和 FHEW 密文之间高效地来回切换，而无需解密，从而解决了在单一同态加密方案中不能同时进行多项式函数和非多项式函数计算的问题。该方案不仅为同态加密技术提速，还将密钥的体积减少2个数量级，大大减少了公钥传输成本和计算成本。

客户端采用目前对机器学习训练支持比较好的CKKS同态加密方案，CKKS支持SIMD操作，在机器学习的训练中，可以对数据进行打包、加密，使得运算效率得到提升。

客户端首先对明文进行加密，转化为CKKS密文，传输给云服务器，云服务器在 CKKS 密文端计算数据点之间的距离，再将客户端传输来的CKKS密文转化为FHEW密文，然后通过LUT算法，在服务器端以密文的形式进行大小比较，比较完成后再转化为CKKS形式的密文。通过这种方式，服务器端可以使用CKKS算法快速并行地计算距离，并且通过密文转换，使用LUT进行密文比较计算。

同时，由于DBSCAN聚类算法本身不适用于同态加密，我们修改了DBSCAN算法的部分设定，增加了距离矩阵和加入判断是否在邻域内的标签等，使其更加适用于同态加密算法。并以此为思路，尽可能去减少真正需要去计算的数据点。本方案通过计算数据点所属的网格单元，减少用户需要去解密的数据点的个数，从而减少在客户端的解密耗时，将时间复杂度缩小为。级别的时间复杂度意味着可以做大型数据计算，此优化使得本方案更具有实用性和创新性。

当比较两点之间距离的平方与预定值的大小时，不同于传统的协议交互，将密文传回给客户端，客户端解密完成后，再将比较结果发送给服务器端，本方案可以直接实现在服务器端不需要解密，直接进行大小的比较，且误差很小。我们的方案解决了用户和服务器需要设计协议交互才能完成DBSCAN聚类分析任务的缺陷，避免了用户和服务器交互过程中被攻击的风险，有效地保护了聚类分析过程中的数据隐私不被泄露。同时，这种方案无需数据所有者在线交互、参与计算，从而减少了本地计算资源使用量，更加便捷且节省通信开销。

在支持复杂计算上，本方案将欧氏距离中不被同态加密算法支持的开方运算使用变形欧氏距离进行替换。相较于传统的欧氏距离，使用变形欧氏距离能够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。同时变形欧氏距离的计算仅需要两次乘法，带来的时间开销也比较低。

在进行密文的比较运算时，我们通过使用在PEGASUS框架下的密文转换方案，在支持多项式运算和支持非多项式运算的全同态加密技术之间搭建一座桥梁，使得在计算距离这样的多项式运算时能够在CKKS密文侧高效完成，计算距离比较这样的非多项式运算时能够在RLWE密文侧高效进行，从而解决了现有方案中同时计算多项式函数和非多项式函数的方法仍然缺乏的问题。

经过测试，本方案的DBSCAN同态加密算法在聚类结果上和在明文上直接进行DBSCAN聚类算法的准确率和聚类结果相近。我们还使用Python内置的tkinter库实现了简单的图形化操作界面，对方案的结果进行可视化展示。

## 1.5应用前景

本项目在PEGASUS框架下设计的基于同态加密技术的DBSCAN聚类隐私保护方案，能在保证方案安全性的前提下，使得在加密数据上的聚类计算的准确度和速度在可接受范围内，从而为实际应用中的机器学习中隐私保护提供了一份可行的方案。

项目的主要应用场景为个人、企业或组织需要进行大量数据的聚类分析服务以及企业为用户提供的借助聚类分析用户的数据的服务等。由于数据量众多，通常选择将数据外包到云服务器以减少本地计算资源使用量，但不可靠的云服务器可能存在隐私泄露问题，因此，用户希望将加密后的隐私数据上传至云服务器，对加密数据进行操作后再将结果返回给客户端，客户端本地解密，从而完成隐私保护。通过借助本项目完成的在PEGASUS框架下基于同态加密的DBSCAN聚类算法隐私保护方案，可以围绕解决数据外包中产生的隐私安全问题来开展更多的云计算服务。

图示

描述已自动生成

图1.7 医疗方面的应用

例如于图1.7所示的医疗数据的隐私问题，医疗数据的庞大繁杂使得医院不得不通过将数据外包到云服务器上进行相应的聚类计算任务，如果云服务器提供商可以对加密数据进行操作，就会减少因为使用第三方服务而产生的隐私安全问题，从而提高云服务器提供商在行业中的竞争力。另外，借助此方案，业界也可能创造出新的盈利模式和机会，可以围绕以前无法实现的数据启用新的业务模型。例如，它解决了机器学习中最大的问题之一，即访问大型、多样的数据集以建立模型。因此，它可以将拥有数据的人与需要数据的人聚集在一起。而且所有这些操作都可以保护数据的安全性和私密性。

同时，本项目受众面广泛，对于云服务器用户、云服务器提供商、科研工作者等均具有一定参考价值：

（1）云服务器用户

对于云服务器用户来讲，敏感信息（如个人资产信息、医疗信息）的保护是重中之重。本项目设计的方案将加密后的数据而非明文上传至云服务器，且用户端和服务器端不必时时交互，从而避免了交互过程中被攻击的风险，有效地保护了聚类分析过程中的数据隐私不被泄露，保证了用户隐私数据的安全性。

（2）云服务器提供商

本项目将密文数据上传至云服务器端进行处理，通过同态加密直接在密文数据上进行运算，从而达到所得到的密文结果解密后等同于在明文上进行相应运算的效果。在实际过程中除了保证数据安全，还兼顾了商业化应用开发的时间开销、存储开销以及普及可行性。对未来接受密文数据的云服务器系统的构建具有一定的参考价值。

（3）科研工作者

隐私保护机器学习一直是信息安全领域的热门研究问题。我们设计的在PEGASUS框架下基于同态加密的DBSCAN聚类算法隐私保护方案对现有方案提出了优化改进措施，可以对相关领域的研究者提供一个新的思路的参考方案。

# 第二章 作品设计与实现

## 2.1 系统框架

系统由客户端、云服务器双方组成：

·云服务器为用户完成密文的DBSCAN聚类运算；

·客户端进行数据的预处理、加解密、上传和下载以及聚类结果展示。

系统架构如图2.1所示：

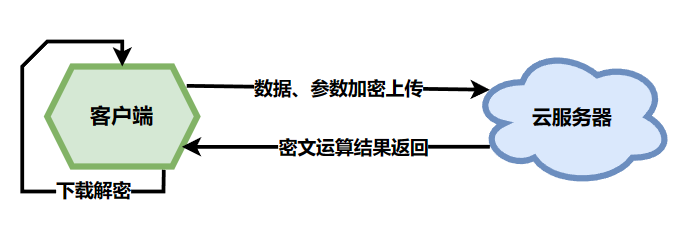


图2.1 系统架构

本系统实现了聚类数据预处理、加密上传、同态密文聚类运算、解密聚类展示的功能。程序执行的流程如图2.2：

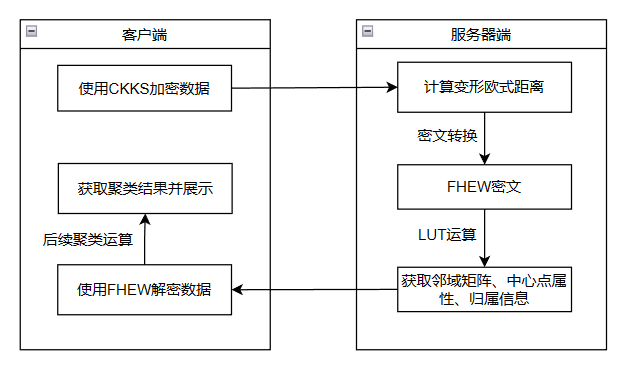


图2.2 程序流程

当客户段上传 DBSCAN 聚类数据时，会根据数据的大小对数据进行预处理，根据数据的最大差值对数据进行等比例缩放，该操作并不会对聚类结果产生影响，但能够提升服务器端的聚类精度与聚类数据的大小限制。同时用户在客户端指定聚类的参数()，将聚类参数与预处理后的数据一并使用PEGASUS框架加密为RLWE 密文，上传至服务器端。

服务器端接收到RLWE密文后首先进行变形欧式距离的计算，使用RLWE密文减法、乘法完成操作；随后通过计算获取变形欧式距离与的差值。之后通过PEGASUS框架的与方法将编码向量的 RLWE 密文转换为向量元素的一组 LWE 密文；最终通过一系列并行后的 LUT 方法进行比较运算、LWE 密文加减运算获取密文数据邻域关系、中心点属性等所需密文信息，返回到客户端。

客户端根据服务器返回的 LWE 密文可使用时间复杂度为级别的算法完成后续的聚类操作，该过程最终无需解密全部 LWE 密文信息，所需解密的 LWE 密文信息的量级为。

在本项目中提出的基于同态加密的隐私保护方案中，在服务器实现聚类操作的过程中，无需用户持续在线，也没有因协议交互而产生的额外通信开销。

## 2.2 设计动机

在本节中我们首先介绍了选用PEGASUS框架的原因与优势，随后介绍了 DBSCAN聚类算法在同态密文下实现的主要难点与克服方法。

### **2.2.1 全同态加密的应用**

绝大多数的全同态方案可以分为 bit-wise和 word-wise两类。bits-wise方案将明文中的每比特进行加密从而得到相应的密文，可以在密文上用布尔电路来进行任意的函数运算。已有研究表明bit-wise的同态加密方案在实际的多项式加、乘法运算中表现欠佳，并且bit-wise密文在运算后的密文膨胀率远远大于 word-wise 的，会造成较大的通信开销。相反，word-wise方案支持高效的单指令多数据形式的同态运算（即加法和乘法），同时可将多个明文打包成一个密文[12]，然而word-wise密文上计算非多项式函数将变得困难。

由此Pegasus[26]提出了可以在无需解密的情况下在两类方案密文之间高效转换的方案，Pegasus可以在 packed 的 CKKS 密文和 FHEW 密文之间高效地来回切换，而无需解密，从而允许我们在 CKKS 端高效地计算算术函数，并在 FHEW 密文上计算查找表。同时Pegasus支持大输入数域的LUT评估，该方案放宽了LUT方法[27]**错误!未找到引用源。**中的约束，以引入一些近似误差为代价，使其能够接收明显更大的输入（如位），该输入大小足以支持隐私机器学习等应用的需求。

### **2.2.2 DBSCAN聚类算法**

虽然DBSACN聚类算法相比于K-Means有不受形状、异常点影响的优点；但其算法实现更加复杂，聚类收敛时间较长。DBSCAN的时间复杂度是, 是点的个数，为找出领域中的点所需要的时间，最坏情况下时间复杂度是。在低维空间数据中,有一些数据结构如KD树，使得可以有效的检索特定点给定距离内的所有点，时间复杂度可以降低到。受其算法思想的限制，在纯密文运算的情况下（在聚类过程中不提供任何的明文信息），完全实现该算法的时间复杂度高达，对于计算代价远大于明文的密文运算，该时间复杂度是不能被接受的。

在此之前已有方案[25]实现了基于同态加密的DBSCAN聚类隐私保护方案，该方案通过引入交互协议实现了低服务器代价的DBSCAN密文聚类方案，但是该交互协议要求客户端在聚类过程中持续交互。

我们采用了一种折中的方式实现了无需交互协议的隐私保护云端DBSCAN聚类方案。通过使客户端增加可接受的运算代价，大幅度降低了服务器所需支付的运算代价；此时客户的运算时间复杂度仍为级别，但服务器端的运算复杂度降至。我们参考了一种改进的DBSCAN聚类方案[28]，通过引入聚类空间中的网格单元，使客户端根据数据点所属的网格单元完成 级别时间复杂度的聚类。

## 2.3方案设计

在本节中介绍了方案的设计与操作流程，核心设计方案主要分为距离测度选择、数据预处理加密上传、密文运算、客户端聚类四部分。

### **2.3.1 距离测度的选取**

在机器学习算法中，距离计算可以采用欧氏距离、曼哈顿距离、变形欧式距离等。对于已知的两点，曼哈顿距离为：，变形欧式距离为：。

同态加密算法不支持欧氏距离中的开方运算，需要对开方运算进一步处理，因此难以采用欧氏距离。为了测试采用不同距离算法对聚类准确度的影响，以选取最合适的距离测度方式，我们进行了如下测试。由于聚类算法中常用的距离计算方式是欧式距离，所以我们在同样的环境下将欧式距离替换为变形欧氏距离和曼哈顿距离后观察聚类的结果是否与使用欧氏距离时的聚类结果相近。选取同样的参数，使用这三种距离算法在明文上进行聚类，其结果如图所示。

图表, 散点图

描述已自动生成

图2.3 欧式聚类聚类结果

图表, 散点图

描述已自动生成

图2.4 变形欧式聚类聚类结果

图表, 散点图

描述已自动生成

图2.5 曼哈顿距离聚类结果

其中横纵坐标为数据点的值，输入的数据集有3000个点，相同颜色的为同一聚簇，不同颜色的为不同聚簇，cluster -1的点为噪声点。观察图2.3到图2.5中的三幅图，分别表示选择欧式距离、变形欧氏距离、曼哈顿距离的聚类结果。通过计算，欧式距离、变形欧氏距离和曼哈顿距离下的聚类算法的准确率依次为0.87，0.885，0.749。再结合上图所示，可以看出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。

同时曼哈顿距离的计算中涉及取模运算，该非多项式运算不受CKKS的支持，需要支付额外的密文转换代价。在文献[9]中使用牛顿迭代法实现了同态开方运算，运算过程如下：

（2.1）

其中为正实数，收敛于，经过有限次多项式运算即可获取足够精确的开方结果。虽然使用牛顿迭代法实现了同态开方运算，但该算法运算开销较大，同时存在一定误差，会对距离结果产生影响。

相比于曼哈顿距离、欧式距离，每次变形欧式距离的运算过程仅涉及两次减法、一次加法以及两次平方运算，这三种运算均受CKKS的支持。同时，变形欧式距离可以提供更高的计算精度，避免由开方运算带来的误差。因此我们采用了变形欧式距离作为距离测度。

### **2.3.2 数据预处理加密上传**

由于PEGASUS框架中支持的FHWE密文的大小受限，同时采用变形欧式距离平方后的密文数据可能进一步变大，需要对距离数据进行缩放预处理，算法如下：

|  |
| --- |
| **算法3：DataScale ()** |
| **输入：**  ：包含 n 个元素的数据集明文  ：邻域半径  ：邻域密度阈值  **输出：**  数据集RLWE密文、邻域半径密文、邻域密度阈值密文 |
| 在数据集载入的过程中获取最大、最小的坐标值  获取坐标的最大差值  根据的值获取缩放比例  根据对中的数据进行缩放获取  分别用CKKS加密获取其RLWE密文 |

### **2.3.3 密文运算**

服务器端的密文运算分为RLWE密文计算、RLWE与LWE密文转换、LWE密文计算三部分。使用RLWE密文计算变形欧式距离以及每个数据点所属的网格单元，使用LWE密文获取邻域矩阵，同时获取每个数据点其他属性。

**（1）RLWE密文计算**

第一步，计算每个数据点所属的网格单元。

**定义 1** 网格单元。设置邻域半径为划分参数，将每一个维度的数据空间划分为相同长度的区域，从而将整个二维数据空间划分若干个边长为的网格单元。图2.6中橙色的部分代表了一个网格单元。

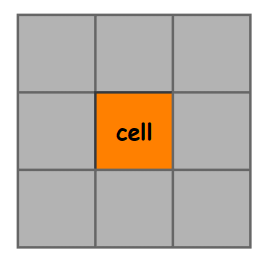


图2.6 网格单元

通过计算获取每个数据点的网格单元，由组成网格单元的位置集。

**定义2** 邻接网格集。cell是一个网格单元格。一个矩形区域可以通过在cell的上下左右四个方向来扩展，邻接网格集定义为该矩形区域所包含的网格集，用Ncell表示，如图2.7区域所示，表示了定义1中cell的邻接网格集。

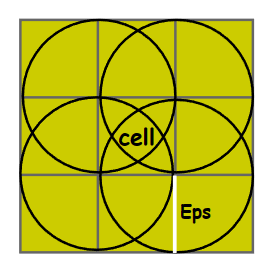


图2.7 邻接网格集

若数据点网格单元为，易知其半径内的点包含在邻接网格集中。

第二步，计算变形欧式距离，算法实现如下：

|  |
| --- |
| **算法4：GetDistance (, )** |
| **输入：** ：数据集的 RLWE 密文  ：邻域半径平方的RLWE 密文  **输出：** RLWE密文邻域矩阵 |
| **FOR** each **DO**  **FOR** each **DO**  计算  **END FOR**  **END FOR** |

此处循环的计算过程中RLWE密文为格式的向量，得到的RLWE密文邻域矩阵为个格式的向量。

**（2）RLWE与LWE密文转换**

|  |
| --- |
| **算法5：Conversion)** |
| **输入：**RLWE密文邻域矩阵  **输出：** LWE形式的密文邻域矩阵 |
| **FOR** each **DO**    **FOR** each **DO**    **END FOR**  **END FOR**  构成 LWE形式的密文邻域矩阵 |

给定加密的RLWE密文，操作将产生一个RLWE密文，操作对解码函数进行同态评估。给定密文和整数，计算得到，该运算的结果等同于在同一密钥下对的第个系数进行LWE加密。

每个 RLWE密文可以提取出 𝑛 个LWE密文，最终由生成一个的LWE密文距离矩阵，其中元素表示为。通过相同的方式对进行运算可得到邻域密度阈值LWE密文。

**（3）LWE密文计算**

通过在 LWE密文上使用 LUT 函数 可以在密文上进行非多项式函数运算，在本方案中使用了LUT函数来比较LWE密文与特定数值的大小，经过LUT运算后可以将比较结果以0或1的LWE密文返回。

例如对于特定的比较运算函数例如IsNegative，即，LUT操作如下：

（2.2）

第一步，计算密文邻域矩阵。

对于 ()，计算，其中T(x)的取值如下：

（2.3）

并令 。

第二步，计算每个数据点是否为中心点，算法如下：

|  |
| --- |
| **算法6：IsCore ()** |
| **输入：** ：密文邻域矩阵  ：邻域密度阈值LWE密文  **输出：** 中心点属性值LWE密文向量 |
| **FOR** each **DO**    **END FOR**  **FOR** each **DO**    **END FOR**  **FOR** each **DO**    **END FOR**  由构成中心点属性值LWE密文向量 |

为格式的LWE密文向量，其中元素解密值（0/1）代表该数据点（不是中心点/是中心点）。

第三步，计算每个数据点归属的中心点，算法如下：

|  |
| --- |
| **算法7：Belong ()** |
| **输入：** ：密文邻域矩阵  ：中心点属性值LWE密文向量  **输出：** 归属值LWE密文向量 |
| **FOR** each **DO**      **END FOR**  **FOR** each **DO**  **FOR** each **DO**        **END FOR**  **END FOR** |

对于，的解密值是之间的整数。

### **2.3.4 客户端聚类**

客户端接收在2.2.3中的LWE密文以及RLWE密文，可解密得到对应信息，后续聚类操作使用解密信息完成（部分解密）。后续聚类算法的实现如下：

|  |
| --- |
| **算法8：𝐸𝑥𝑝𝑎𝑛𝑑𝐶𝑙𝑢𝑠𝑡𝑒 ()** |
| **输入：**邻域矩阵，中心点属性，归属值，位置集  **输出：**聚簇结果集 |
| **FOR** each **DO**  **IF**  **DO**  将入栈离群点栈，结束本轮循环  **END IF**  **IF** && **DO**  删除中的第行、第列，将值入栈临界点栈  **END IF**  **END FOR**  **FOR** each **DO**  将值入栈中心点栈  将{插入HashMap中  **END FOR**  **FOR** each **DO**  如果为已访问，跳过本轮循环。  将标记为已访问，为分配一个新簇  获取HashMap中邻接网格集中的所有  **FOR** each **DO**  获取中第行第列的值，若为1则将聚入当前簇  对重复与相同的操作，但不分配新簇  **END FOR**  **END FOR**  将临界点栈中的点根据归属值聚簇，聚簇结束 |

## 2.4安全模型

我们假定，服务器与客户端之前的通信时的信道是不安全的，敌手可能会窃取到双方通信的消息，从而发起攻击。所以敌手是可能从信道中获取到双方传输的CKKS密文的，同时敌手也可以伪造密文去访问服务器，让服务器去对传输的数据进行聚类，之后再将密文发送给敌手。那么敌手就可以获得CKKS密文和聚类后的CKKS密文，并对其进行频率分析。我们假定服务器是“诚实且好奇”的，即服务器是半可信的，服务器会根据方案执行所有过程并提供相应的服务，但会尽可能的收集数据信息包括文件与关键词信息。

## 2.5正确性分析

CKKS算法的目标是做近似计算。这并不偏离需求，因为现实生活中大部分运算，面对的是实数（复数），而实数（复数）的运算，往往只需要保留一部分有效数字即可。

CKKS在加密阶段使用了多项式加密，即将明文转化为多项式形式，并使用加密算法来加密这些多项式。由于多项式加法和乘法是可逆的运算，因此在解密阶段也可以通过执行相应的多项式运算来恢复明文。

同时CKKS在加密过程中引入了一定量的噪声，以保护密文数据的安全性。在同态计算过程中，噪声会逐渐增加，使得结果不再与明文完全匹配。CKKS通过选择合适的参数和优化噪声消除算法来控制噪声增长并保证同态计算的正确性。

CKKS使用误差估计技术来判断同态计算结果的正确性。在同态计算过程中，CKKS会根据当前噪声水平和计算过程中引入的误差来估计结果的精度，并在必要时进行修正。通过合理的误差估计和修正，CKKS可以保证同态计算结果的正确性。

综上所述，CKKS同态加密方案的正确性可以通过使用多项式加密，噪声控制技术和使用误差估计技术来判断结果精度并进行修正，来保证加密的正确性。

## 2.6安全性分析

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

### **2.6.1 CKKS密文的安全性分析**

服务器或具备窃听能力的敌手能够获得的数据为 和参数、MinPts 。从数据的组成来看，除了 MinPts 之外，其他都是通过 CKKS同态加密得到的密文。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 CKKS加密算法的概率。由同态加密算法的语义安全可知，攻破此加密算法的概率是可忽略的，因为我们已知RLWE问题是计算性困难问题，如果攻破CKKS加密方案的概率是不可忽略的，那么破解RLWE问题的概率也是不可忽略的，这与已知不符。因此，服务器和敌手成功获取用户数据的概率也是可忽略的。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，在已知MinPts的前提下，攻破CKKS加密的概率是不变的。服务器只能进行聚类的操作，无法获取具体加密的数据明文。

### 2.6.2 获取模型参数倒推明文的安全性分析

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，属于无监督学习方案。本方案保留且运用了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，服务器端不需要知道加密的数据具体是什么，就可以对其进行聚类。所以本方案也就不存在模型。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方式无法应用于本方案。另外，本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。要想得到明文，本质上还是需要攻破加密算法，通过解密来获取明文。因此，本方案的数据安全性立足于加密算法的安全性。综上，攻击者在获得同态 DBSCAN 算法后，能够成功获取用户原始数据的概率等同于攻破 CKKS同态加密算法的概率，且我们已知这个概率是可忽略的。所以敌手通过获取模型参数来倒推用户原始数据的这种攻击方式的攻破概率是可忽略的。

# 第三章 作品测试与分析

## **3.1测试**方案

本章在2.3节设计的方案流程的基础上，我们使用scikit-learn库来生成模拟的数据集对于现有的方案（文献[33]）进行纵向对比，对不同数据规模下的明文、密文的聚类结果和效率、准确率等多维度的评估参数进行横向的对比，从而多视角多维度对我们的设计方案进行尽可能全面的评估。上述提到的密文是使用第三方开源的SEAL库中的CKKS加密方案对明文加密得到的结果。

## **3.2测试**环境搭建与测试设备

测试环境与测试设备的信息如表3.1所示：

表3.1 测试环境与测试设备参数信息

|  |  |  |
| --- | --- | --- |
| 项目 | | 型号参数 |
| 服务器 | CPU | 2.6GHz@Intel Xeon Gold 6132 |
| 操作系统 | Ubuntu18.04.5 |
| 内存 | 4G |
| 客户端 | CPU | 2.5GHz@ Intel Core i7- 7200U |
| 操作系统 | Ubuntu20.04 |
| 内存 | 4G |
| 语言环境 | | C++、Python |
| 第三方依赖 | | SEAL库 |

## **3.3测试**数据与结果分析

### **3.3.1 与现有方案的对比**

我们将设计的方案与文献[33]提出的算法进行了对比，为了控制变量，我们使用相同的数据集进行，我们提出的设计方案的结果将详细在3.3.2节和3.3.3节有详细的横向多维度的对比，这里不再赘述。主要体现与文献[33]提出的算法的聚类速度比较分析。

文献[33]在给出的在10个数据项的基础上对精确（Extract）、稳定（Stabilized）、近似（Approximate）三种状态从单轮运行时间、运行总轮数、总运行时间三个维度对提出的聚类方案进行了分析。

表3.2 文献[33]K-means聚类方案的结果分析

|  |  |  |  |
| --- | --- | --- | --- |
|  | Extract | Stabilized | Approximate |
| 单轮运行时间 | 873.46小时 | 15.56小时 | 15.47小时 |
| 运行总轮数 | 15轮 | 40轮 | 40轮 |
| 总运行时间 | 545.91天 | 25.93天 | 25.79天 |

因为Extract方案相较于Stabilized和Approximate方案，尽管轮数很少，但是不可避免使用分数的编码，这就导致了即便在这样小的数据规模下时间消耗仍然是不可接受的。另外，Stabilized和Approximate方案二者由于没有复杂分数编码带来的时间消耗，时间上较好，但由于引入更多的轮数，总的运行时间尽管有了明显的缩减，但在实际应用中还需要多核并行加速等技术才能真正落地。

相比之下，我们提出的设计方案就要好很多，具体见3.3.2节和3.3.3节进行详细的横向分析和阐述。

### **3.3.2不同数据规模下的聚类结果**

为了较好反映我们方案对数据的结果，我们调用scikit-learn库中提供的生成模拟的聚类数据集的接口随机生成16、32、64、128、256规模的数据，并对其数据的明文和密文分别聚类进行比较，结果如图3.1所示。另外，对不同规模的数据的解密准确率也已经给出，如表3.3所示。

表3.3 本55方案不同数据规模下密文聚类结果准确率分析

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 数据规模 | 16 | 32 | 64 | 128 | 256 |
| 聚类结果准确率（%） | 93.75 | 100 | 98.44 | 99.22 | 99.61 |

可以看到，当数据规模较小时（如16、32、64等），对明文的聚类结果和对密文的聚类结果基本没有差异，这说明我们的方案在规模较低的情况下能很好地满足对密文数据的聚类的准确性。当数据规模较大时（如128、256等）除了极少数的边缘点被判断为离群点后，整体的聚类结果也满足明文的要求。

在不同的数据规模下，可以看到对密文聚类结果的准确率除数据规模为16外，基本在99%，这是满足对密文聚类的基本要求的。另外，当数据规模为16时，准确率为93.75%，这是由于数据量过小，密文聚类后的邻域半径影响较大导致的，不具有代表性，这在实际应用的过程中是极大概率不存在的。

地图, 散点图

描述已自动生成

图3.1 不同数据规模下明文和密文聚类结果对比

### **3.3.3不同数据规模的多维度对比**

为了反映我们方案的效率问题分别选取了16、32、64、128、256的数据规模，分别从加密效率（Enc）、距离计算效率（Dis）、服务器到客户端的传输效率（S2C）、特定数据信息提取效率（Extract）、查找表算法效率（LUT）、邻域半径（，Epsilon）、准确率（Accuracy）、邻域密度阈值（MinPts）等多个维度对不同数据规模情况进行对比，结果如图3.2所示。

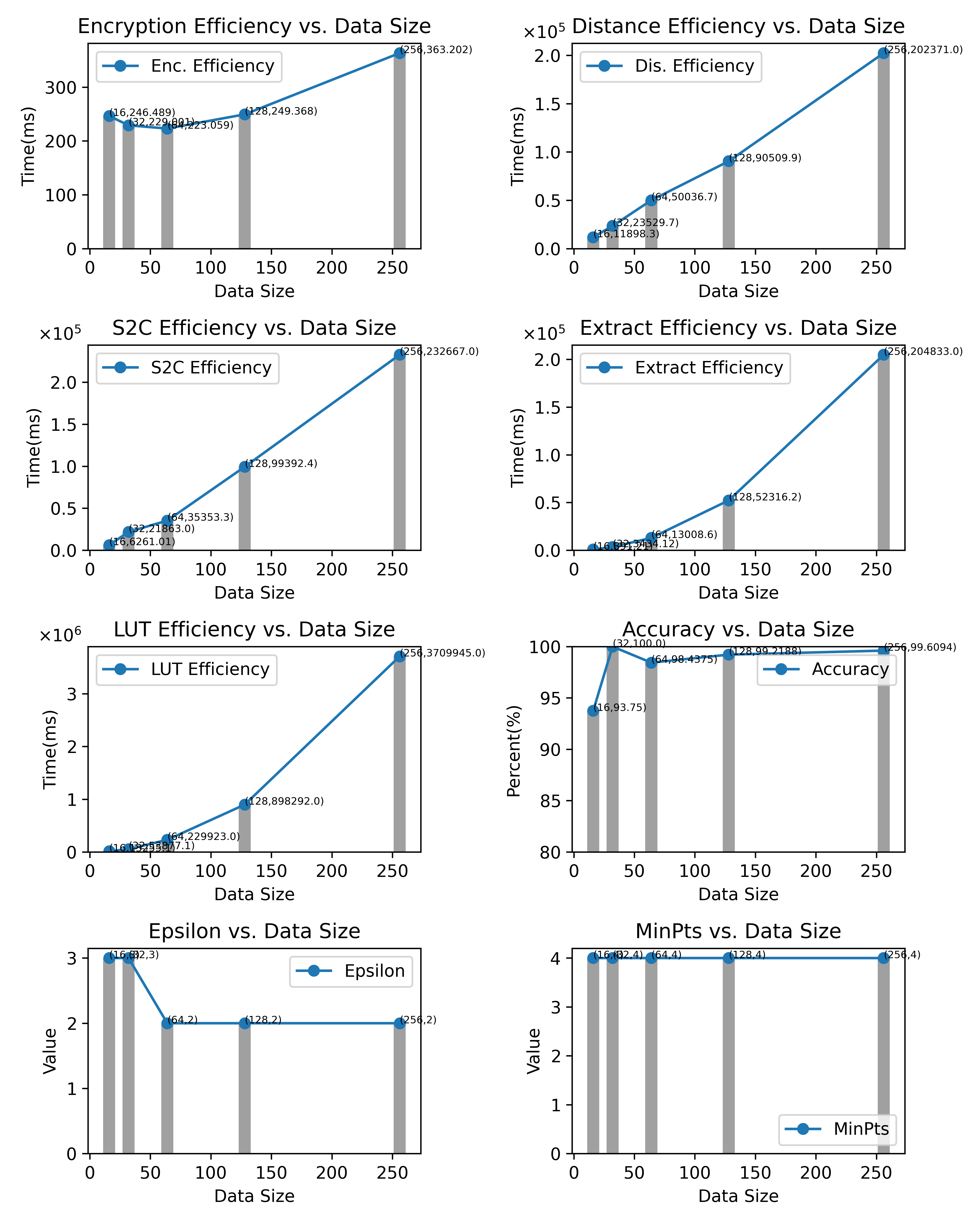


图3.2不同数据规模下各个维度的参数对比

可以看到，当数据规模从16、32、64、128、256逐步不断增大时，加密效率（Enc）、距离计算效率（Dis）、服务器到客户端的传输效率（S2C）基本与数据规模线性递增，这是由于这些计算的复杂度为。其中加密效率当数据规模较小时有比较明显的异常情况，这是因为当数据规模小时，系统的其他内置的参数的影响较大，这是符合客观事实的。

另外，特定数据信息提取效率（Extract）、查找表算法效率（LUT）关于数据规模呈现出非线性，对比具体数值可以发现呈平方相关。这是由于这些计算的复杂度为。首先需要肯定的是我们的方案已经把的问题降到了，其次需要进一步明确的是这些操作在Server-Client模式下对客户端来说仍然是不小的代价，在实现的过程中需要依赖于多核并行、GPU算力支持等技术的支持才能拥有更加普适的应用前景。

# 第四章 创新性说明

## **4.1基于PEGASUS框架的**隐私保护方案

文献[26]中提出了一个实用框架PEGASUS，实现了在打包的CKKS密文和FHEW密文之间有效地来回切换，而无需解密。在CKKS密文中能够高效计算多项式函数，而在FHEW密文中可以评估查找表（也就是非多项式函数），PEGASUS框架通过两种密文的转化，在多项式计算和非多项式计算之间搭建了桥梁，从而可以自适应的加速计算。同时PEGASUS的密钥大小比CHIMERA框架[29]要小（从80千兆字节减少到12兆字节），很大程度上节省了内存成本与计算成本，而且其采用的优化计算方法使得PEGASUS的运算速度也要优于CHIMERA使用的Torus对应的运算（计算复杂度从线性提高到亚线性）。

我们在本项目中使用了PEGASUS方案，并对DBSCAN聚类的流程进行了修改，从而将现有的同态加密聚类方案进行了优化，使其适用于DBSCAN聚类。服务器可以在 CKKS 密文端高效地计算数据点之间的距离，并将CKKS 密文转换为FHEW密文，通过计算查找表LUT的方式完成密文的大小比较功能，从而充分利用了CKKS和FHEW密文的特点，实现了计算过程的高效性和灵活性。同时，通过使用这种密文转换，服务端可以完成DBSCAN聚类所有运算，从而减少了客户端计算资源的使用量。此外，服务端可以在保持数据加密状态下，有效地执行聚类运算，而无需暴露敏感的原始数据。

## **4.2无协议交互**

现有的DBSCAN 隐私保护方案会根据同态加密算法所支持的运算方式，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，必须通过设计协议，增加客户端与服务器端交互的方式来进行处理。其中，文献[30]中设计了一种使用additive blinding和同态加密的DBSCAN聚类隐私保护的协议。除了开销很大的加密操作外，他们的协议还要求各方都能明文访问数据记录，并且实时在线交流，很不适用于数据外包的隐私保护场景。基于安全多方计算[31]-[32]采用多方之间的协议以联合计算函数而不暴露其输入的任何信息。MPC比同态加密速度更快，但是需要数据所有者在线，需要更明显的带宽开销。

总之，现有的基于协议的聚类隐私保护方法要求参与方可以访问明文输入数据，且需要数据所有者在线。而基于同态加密的聚类隐私保护方案可以在加密后完全离线操作，无需数据所有者参与计算，从而更加便捷和节省带宽。

本项目提出一个在PEGASUS框架下的同态加密的DBSCAN聚类隐私保护方案。基于PEGASUS这个密文转换框架，服务器可以在packed的CKKS密文（基于RLWE）和FHEW 密文（基于LWE）之间高效地来回切换，在 CKKS 密文端高效地计算数据点之间的距离，在FHEW密文上通过计算查找表的方式完成密文的大小比较功能。本方案不需要和客户端进行额外通信，在无需协议交互的情况下实现了云服务器对于加密数据的 DBSCAN聚类操作。

优化后的方案无需用户端与服务器端频繁交互，大大降低了用户和服务器交互过程中被攻击的风险，有效地保护了聚类分析过程中的数据隐私不被泄露。此外，此方案无需客户端一直处于运行的状态，优化了用户的使用体验。同时，这种方案无需数据所有者在线交互、参与计算，从而减少了本地计算资源使用量，更加便捷且节省通信开销。

## **4.3聚类速度显著提升**

由于同态加密是在密文端进行运算，所以无论是加解密，还是在密文下进行加减乘除等运算，都是耗时巨大的。且本方案使用的是DBSCAN算法，其算法实现相较于K-Means算法实现更为复杂，聚类收敛时间较长。DBSCAN最坏情况下时间复杂度为，在纯密文计算的情况下，完全实现该算法的时间复杂度为，这种时间复杂度是很难被接受的。

因此，本方案通过使客户端增加可接受的计算代价，大幅度降低了服务器所需支付的计算代价，此时客户端的计算时间复杂度仍为级别，但服务端的计算复杂度减至。我们具体的操作是：在服务端的密文运算中，首先使用RLWE密文计算变形欧氏距离以及每个数据点所属的网格单元，再将RLWE密文转化为LWE密文，通过LWE密文获取邻域矩阵以及每个数据点的其他属性。而通过引进上述聚类空间中的网格单元，明确数据点所属的网格单元，客户端可以减少用户需要解密的数据点的个数，从而完成级别时间复杂度的聚类。

我们在3.3测试数据与结果分析一节中使用相同的数据集将设计的方案与现有方案（文献[33]）进行对比。文献[33]是基于同态加密的KMeans聚类算法，KMeans算法本身的时间复杂度是级别，正常来说，进行聚类的速度远快于时间复杂度为的DBSCAN聚类算法。在第三章的表3.2中给出了该文献的测试数据。而我们方案的效率可详见3.3.3节，我们从加密效率（Enc）、距离计算效率（Dis）、服务器到客户端的传输效率（S2C）、特定数据信息提取效率（Extract）、查找表算法效率（LUT）、邻域半径（，Epsilon）、准确率（Accuracy）、邻域密度阈值（MinPts）等多个维度对不同数据规模情况进行了对比。由测试结果可知，本方案的聚类速度可以做到优于现有的方案[33]。

## **4.4 基于python实现的跨平台应用程序**

为了简化客户端与服务端交互的过程，提高用户的体验，另外考虑到对不同系统平台的兼容性，使用Python内置的tkinter库实现了简单的图形化操作界面，以便对方案的结果进行可视化展示。

我们在核心程序的基础上实现了加密去重的密钥分发系统，进而以GUI的形式作为客户端的功能。主要包括进行客户身份认证的登录界面和进行系统功能的主界面。整体的设计上满足“0知识操作”原则，用户体验良好。

首先是进行客户身份认证的登录界面，如图4.2所示，为简化操作默认用户已经进行注册，后续可进一步提供完整的注册功能。用户的身份认证信息采用目前主流的非对称的方式，即服务器端存储用户名、口令加盐后的慢Hash值，以保证足够的安全性。认证时，需要与服务器的数据库进行一轮交互。另外，为了简化实现，需要将服务器的IP显式给出，从而进行连接认证。

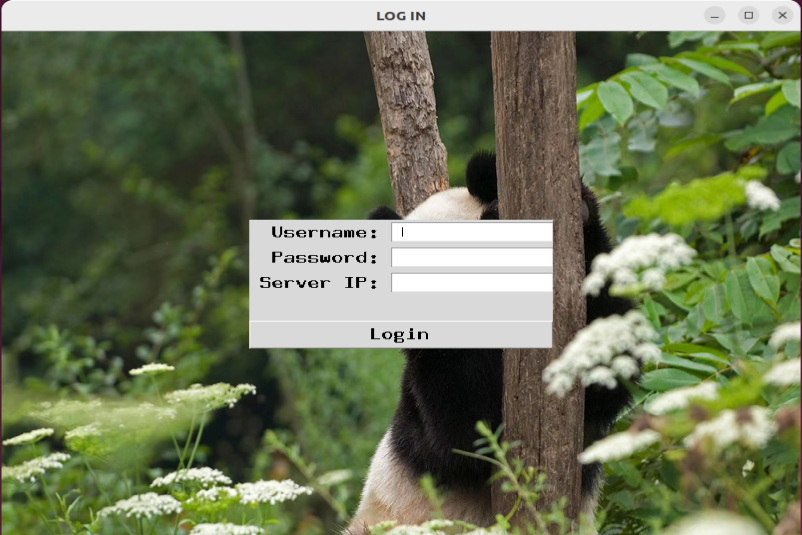


图4.2 跨平台应用程序登录界面

在用户身份认证完成后，进入主界面（如图4.3所示）。主界面主要由日志显示区和功能操作按钮两部分组成。其中，日志显示区显示系统的状态等信息，从而便于与用户进行交互。功能操作按钮包括支持文件上传、文件下载、聚簇结果的可视化显示和退出，点击即可实现相应的功能。

图形用户界面, 应用程序, 表格

描述已自动生成

图4.3 跨平台应用程序主界面

其中，上传和下载文件功能通过全双工实现，点击上传按钮即可调用系统自带的文件资源管理器选择对应的文件，从而进行传输。文件上传、下载页面如图4.4所示。同时，如果收到服务器端返回的聚簇结果会在日志区显示提醒，这时可以通过下载按钮下载文件，进而可以调用显示结果按钮对服务器返回的聚簇结果进行可视化的展示。

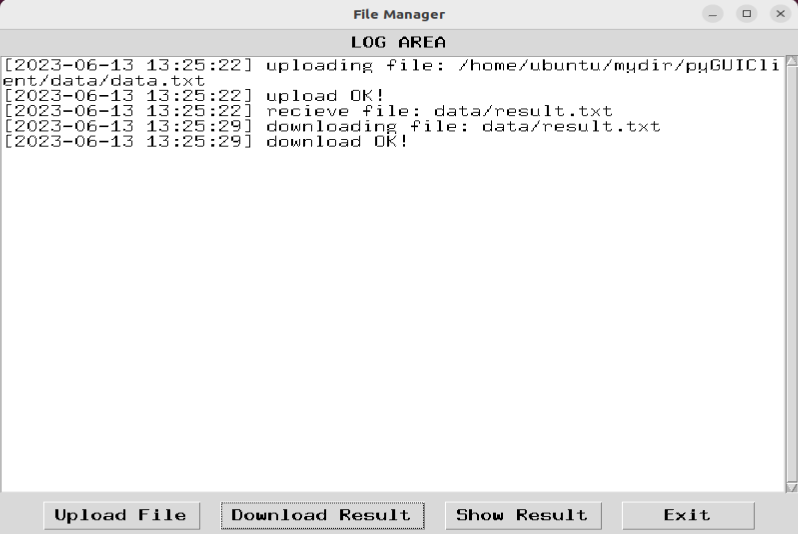


图4.4 跨平台应用程序文件上传、下载

对样例的测试数据进行文件上传，并下载服务器处理后的聚簇结果后，进行可视化的 ，如图4.5所示。

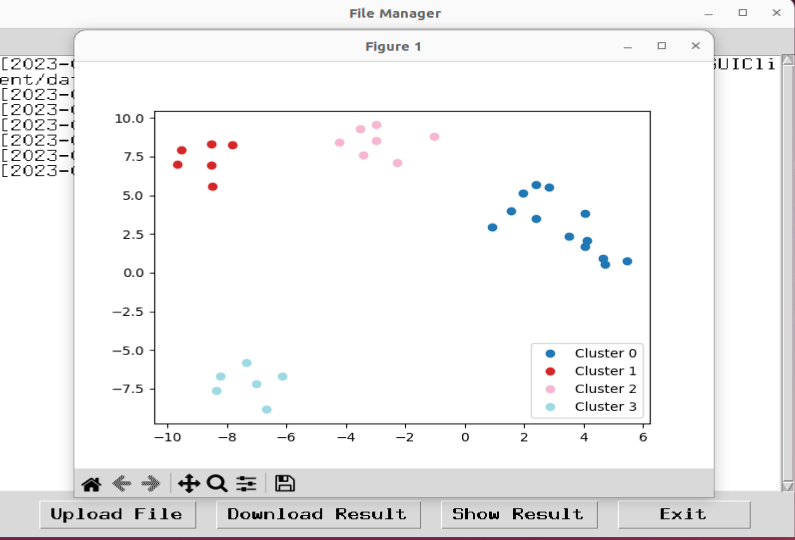


图4.5 跨平台应用程序样例数据聚簇结果可视化展示

从上述结果可以看到聚簇的结果比较满意，此基于Python实现的跨平台的应用程序很好地屏蔽掉了平台的差异，为用户提供了比较便捷的可视化的结果展示，进而有效展示了我们的工作。另外，本程序已经开源。

## **4.5 更加适合同态加密的DBSCAN流程**

由于加密后的明文数据不具有有序性，所以将DBSCAN聚类算法直接用于同态加密是相对困难且不现实的。本方案选择将DBSCAN聚类算法作以改进使其更加适用于同态加密，DBSCAN聚类之所以不适合同态加密，主要问题在于在密文上计算出两点距离之后，无法判断其中一个点是否在另一个点邻域内，进而无法确定哪个点是中心点，哪个点是噪声点。

基于此，本方案提出了先计算两点之间的距离后，判断其是否小于预定的值，由于在密文上，每个密文代表什么数据，服务器端是未知的，所以我们增加了一个的0/1密文矩阵，1代表两点距离小于；0代表两点距离大于，之后将每行的个值相加，与MinPts比较，如果相加之和大于MinPts，则代表该点是中心点，反之，则不是。判断完成后给矩阵的每行都加一个0/1的密文标签，1代表该点是中心点，0代表该点不是中心点，之后判断每个点归属于哪个中心点，最后返回给客户端，客户端收到传输的内容后，判断每个点属于哪个簇中，最后实现选择性解密，不解密全部数据，减少了用户端的解密开销。

通过上述对于DBSCAN聚类算法的修改，可以让服务器端在不知道具体数据的情况下，也能对数据进行聚类并得出结果，同时该修改也降低了客户端在解密时的数据点的个数，减少了解密的运行开销，虽然在客户端进行一系列操作来减少需要解密的数据点是有一定耗时的，但该操作所消耗的时间和在解密全部数据点所消耗的时间相比还是要小很多的。

# 第五章 总结

随着云计算的不断发展，隐私保护越来越受到人们的关注，如何将计算量巨大的工程外包给服务器去进行运算，并且能够保证用户数据的隐私不被泄露，这逐渐成为了人们所研究的方向。本文提出了一种在PEGASUS框架下基于同态加密的无需协议交互的DBSCAN聚类隐私保护方案，该方案不仅在聚类结果上具有较高的准确率，且与之前大部分的同态加密方案有所不同，本方案不再需要协议交互，直接在服务器端以密文形式进行大小的比较，大大降低了通信开销，这也使得不再要求客户端一直处于在线状态，为用户提供了便捷。

在导师和我们团队共同的努力下，我们实现了基于同态加密的DBSCAN聚类隐私保护方案以及可视化界面，当比较两点之间距离的平方与预定值的大小时，不同于传统的协议交互，将密文传回给客户端，客户端解密完成后，再将比较结果发送给服务器端，本方案可以直接实现在服务器端不需要解密，直接进行大小的比较，且误差很小。我们先将CKKS密文(基于RLWE)转换为FHEW密文(基于LWE)，FHEW密文可以通过LUT算法以计算查找表的方式完成密文的大小比较功能，比较完成后，用0/1的密文形式表示大小，并再将FHEW密文转化为CKKS密文，同时，为了进一步加快聚类的速度，我们在我们最初方案时间复杂度为的基础上，通过比较得出邻接网格集和客户端通过筛选来部分解密数据等方式，将时间复杂度缩小为，级别的时间复杂度意味着是可以做大型数据计算的，此优化使得本方案更具有实用性和创新性。

但同时本方案目前聚类速度相对于协议交互来说仍会慢一些，这主要是由于密文转换和密文比较耗时较大，密文比较算法优化起来相对困难，这是由于从数学原理角度上来讲，密文上的比较算法是不具备优化的空间的，其比较的本质就是以查表的方式来得出比较结果。在未来我们还想在客户端加密过程中进行两次加密，首先将数据加密为CKKS密文，再对CKKS密文进行一次SM4加密，由于SM4是对称加密算法，密文传输到服务器端后，在服务器端先运行一次SM4算法，就可以得到CKKS密文。通过使用国产SM4加密算法，使得在传输的过程中更加安全，实现传输过程的自主可控。我们将继续去研究如何将CKKS密文转换为SM4形式的密文。

目前我们已经通过理论分析和实验证明：本方案的DBSCAN同态加密算法，在聚类结果上和在明文上直接进行DBSCAN聚类算法的准确率和聚类结果相近，并且实现了不再需要客户端和服务器端的协议交互，有效减少了通讯开销，通过在服务器端减少实际需要运算的数据点的个数这种方式，可以有效降低时间复杂度，并在客户端减少了实际需要解密的数据点的个数，降低了在客户端解密的时间。

综上，我们提出了一种在能够有效保护用户隐私的基础上高效且实用的实现同态加密的DBSCAN聚类隐私保护方案，在今后的过程中，我们将继续完善我们的同态加密方案，增加更多的功能。

# 参考文献

1. Acar A, Aksu H, Uluagac S, Conti M. 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. 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. Boura, C., Gama, N., Georgieva, M., & Jetchev, D. (2019, May). Simulating homomorphic evaluation of deep learning predictions. In Cyber Security Cryptography and Machine Learning: Third International Symposium, CSCML 2019, Beer-Sheva, Israel, June 27–28, 2019, Proceedings (pp. 212-230). Cham: Springer International Publishing.
6. Cheon, J.H., Kim, A., Kim, M., Song, Y :Homomorphic Encryption for Arithmetic of Approximate Numbers. In: Takagi, T., Peyrin, T. (eds) Advances in Cryptology – ASIACRYPT 2017. ASIACRYPT 2017. Lecture Notes in Computer Science(), vol 10624. Springer, Cham.
7. Fan J , Vercauteren F :Somewhat Practical Fully Homomorphic Encryption IACR Cryptol. ePrint Arch., vol. 2012, p. 144, 2012.
8. Brakerski Z, Gentry C, Vaikuntanathan V: (Leveled) Fully Homomorphic Encryption without Bootstrapping. Innovations in Theoretical Computer Science 2012, Cambridge, USA, January 8-10, 2012, pp. 309–325.
9. 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.
10. Ducas Land Micciancio D, FHEW: Bootstrapping Homomorphic Encryption in Less Than a Second. EUROCRYPT 2015, Sofifia, Bulgaria, April 26-30, 2015, pp. 617–640.
11. Chillotti I, Gama N, Georgieva M, Izabachene M : TFHE: Fast Fully Homomorphic Encryption over the Torus J. Cryptology,vol. 33, no. 1, pp. 34–91, 2020.
12. Smart N P, Vercauteren F:Fully homomorphic SIMD operations Des. Codes Cryptogr., vol. 71, no. 1, pp. 57–81, 2014.
13. Bachrach R G, Dowlin N, Laine K, Lauter K E, Naehrig M, Wernsing J : CryptoNets: Applying Neural Networks to Encrypted Data with High Throughput and Accuracy ICML 2016, New York City, USA, June 19-24, 2016, pp. 201–210.
14. Kim M, Song Y, Wang S, Xia Y, Jiang X:Secure logistic regression based on homomorphic encryption IACR Cryptol. ePrint Arch., vol. 2018, p. 74, 2018.
15. 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
16. 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.
17. Angela J, Armknecht F: Unsupervised Machine Learning on Encrypted Data[C]//International Conference on Selected Areas in Cryptography. Berlin: Springer, 2018: 453-478.
18. 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.
19. 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.
20. Jiang L Z, Xu C X, Wang X F, et al. Secure outsourcing SIFT: Efficient and Privacy-Preserving Image Feature Extraction in the Encrypted Domain [J]. IEEE Transactions on Dependable and Secure Computing, 2017, PP(99): 1.
21. 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.
22. 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].
23. Jiang L Q, Cao Y, Yuan C S, Sun X: An effective comparison protocol over encrypted data in cloud computing[J]. Journal of Information Security and Applications, 2019, 48: 102367
24. 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.
25. 贾春福, 李瑞琪, 王雅飞. 基于同态加密的DBSCAN聚类隐私保护方案[J]. 通信学报, 2021, 42(2): 1-11.
26. Lu W J, Huang Z, Hong C, Ma Y ,Qu H, PEGASUS: Bridging Polynomial and Non-polynomial Evaluations in Homomorphic Encryption . 2021 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 2021, pp. 1057-1073, doi: 10.1109/SP40001.2021.00043.
27. Micciancio D ,Polyakov Y :Bootstrapping in FHEWlike cryptosystems. IACR Cryptol. ePrint Arch., vol. 2020, p. 86, 2020.
28. Ao L M, Dongxue M, Songyuan G ,Shufen L: Research and Improvement of DBSCAN Cluster Algorithm *2015 7th International Conference on Information Technology in Medicine and Education (ITME)*, Huangshan, China, 2015, pp. 537-540, doi: 10.1109/ITME.2015.100.
29. Boura C, Gama N, Georgieva M , Jetchev D. CHIMERA: Combining Ring-LWE-based Fully Homomorphic Encryption Schemes .matical Cryptology, vol. 14, no. 1, pp. 316-338, 2020.
30. Anikin I V , Gazimov R M :Privacy preserving DBSCAN clustering algorithm for vertically partitioned data in distributed systems. 2017 International Siberian Conference on Control and Communications (SIBCON), Astana, Kazakhstan, 2017, pp. 1-4, doi: 10.1109/SIBCON.2017.7998473.
31. Bunn P ,Ostrovsky P :Secure two-party k-means clustering . *Proc. 14th ACM Conf. Comput. Commun. Secur.*, pp. 486-497, Oct. 2007.
32. Doganay, M C , Pedersen T B,Saygin Y,Savas E:Distributed privacy preserving k-means clustering with additive secret sharing.  *International Conference on Pattern Analysis and Intelligent Systems* (2008).
33. Jäschke A, Armknecht F: Unsupervised Machine Learning on Encrypted Data. Cid, C., Jacobson Jr., M. (eds) Selected Areas in Cryptography – SAC 2018. SAC 2018. Lecture Notes in Computer Science(), vol 11349. Springer, Cham. https://doi.org/10.1007/978-3-030-10970-7\_21