

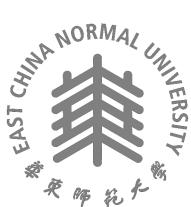
2022 届硕士学位研究生学位论文

分类号: _____

学校代码: 10269

密 级: _____

学 号: 51194501126



東華大學

East China Normal University

硕士专业学位论文

MASTER'S DISSERTATION (Professional)

**论文题目：基于联邦学习的隐私保护的技术
研究**

院 系: 信息学部软件工程学院

专业学位类别: 工程硕士

专业学位领域: 软件工程

论文指导教师: 曹珍富 教授

论文作 者: 何慧娴

2020 年 11 月 20 日

Thesis (Professional) for Master's Degree in 2021

School Code: 10269

Student Number:51184501139

EAST CHINA NORMAL UNIVERSITY

TITLE: VERIFICATION AND ANALYSIS OF ROBUSTNESS OF MACHINE LEARNING TREE MODEL BASED ON SMT TECHNOLOGY

Department:	Software Engineering Institute of Information Department
Major:	Software Engineering
Research Direction:	Trustworthy Artificial Intelligence
Supervisor:	Associate Professor Jianqi Shi
Candidate:	Chaoqun Nie

Nov 9, 2020

摘 要

近年来，机器学习由于其卓越的性能已被越来越多的应用于各个领域，如自动驾驶，人脸识别，个人信用评估等。但机器学习模型一般都为黑盒，由于其不可见性与不可解释性，使得人们对它的安全性有了很大的担忧。因此，越来越多的研究人员投身到机器学习模型的安全性验证的方法和工具的研究中。

与神经网络模型一样，树模型也容易受到对抗性样本的攻击。树模型的“脆弱性”在使其在某些安全性要求较高的应用中造成了隐患。有些情况下，甚至可能导致灾难性的后果。为此，我们研究了树模型的鲁棒性验证问题。本文主要的工作和贡献如下：

1. 我们提出了一个基于 SMT 技术的树模型鲁棒性验证框架，它可以有效的验证树模型的两个重要组成部分：随机森林和 GBDT 模型的鲁棒性，并且支持规模较大的模型的验证。该框架的核心思想是将树模型的鲁棒性验证问题转化为 SMT 公式的约束求解问题。
2. 在验证的基础上，我们进一步对树模型鲁棒性的可解释性问题进行了研究，提出了鲁棒特征集合和局部鲁棒特征重要度的概念来描述模型鲁棒性与样本特征的内在联系，从而为对抗性样本攻击提供了新的思路。
3. 我们基于三个基准测试集评估了框架的可行性和有效性，并且在实验中讨论了模型训练超参数与其鲁棒性的关系，从而为训练阶段提高模型的鲁棒性提供了重要参考。

关键词： 鲁棒性验证， 可解释性， 随机森林， GBDT， SMT

ABSTRACT

In recent years, machine learning has been increasingly applied in various fields, such as autonomous driving, face recognition, personal credit assessment and so on, due to its excellent performance. However, the machine learning model is generally a black box, because of its invisibility and ineluctability, people have great concerns about its security. Therefore, more and more researchers are involved in the research of methods and tools for security verification of machine learning models. Like the neural network model, the tree model is also vulnerable to the attack of the antagonistic sample. The "fragility" of the tree model makes it potentially dangerous and, in some cases, potentially disastrous in some applications where security is high. Therefore, we study the robustness verification of tree models. The main work and contributions of this paper are as follows:

1. We propose a tree model robustness verification framework based on SMT technology, which can effectively verify the robustness of two important components of tree models: random forest and GBDT, and support the verification of larger tree models. The core idea of this framework is to transform the robustness verification problem of tree model into the constraint solving problem of SMT formula.
2. On the basis of verification, we further study the problem of interpretability of tree model robustness, and propose the concepts of robust feature set and local robustness feature importance to describe the internal relationship between model robustness and sample characteristics, so as to provide a new idea for resisting sample attack.

3. We evaluated the feasibility and effectiveness of the framework based on three benchmark test sets, and discussed the relationship between model training hyperparameters and its robustness in the experiment, thus providing an important reference for improving the robustness of the model in the training stage.

Keywords: *Robustness verification, Interpretability, Random forest, GBDT, SMT*

目录

第一章 绪 论	1
1.1 研究背景及意义	1
1.2 问题和挑战	3
1.2.1 数据异构	3
1.2.2 高昂的通信代价	4
1.2.3 安全性和隐私威胁	4
1.2.4 针对联邦学习的隐私保护	5
1.3 本文工作与主要贡献	7
1.4 本文组织结构	8
第二章 相关知识介绍	9
2.1 差分隐私	9
2.1.1 基本定义	9
2.1.2 相关概念	10
2.1.3 差分隐私的实现机制	11
2.2 联邦学习	12
2.2.1 基本定义	13
2.2.2 联邦学习的分类	13
2.2.3 联邦学习的训练步骤	14
2.2.4 联邦学习中的安全和隐私威胁	15
2.3 神经网络	17
2.3.1 SMT 技术	18

2.4	决策树	19
2.5	树模型的基本概念	20
2.5.1	GBDT	21
2.5.2	Bagging 与随机森林	22
2.6	机器学习模型鲁棒性	23
2.7	本章小结	25
第三章	联邦学习	26
3.1	鲁棒性验证框架的整体介绍	26
3.2	树模型解析模块	27
3.3	样本预处理模块	28
3.4	SMT 编码模块	28
3.4.1	决策树的 SMT 编码	28
3.4.2	随机森林模型的 SMT 编码	30
3.4.3	GBDT 的 SMT 编码	32
3.4.4	模型鲁棒性 SMT 编码	32
3.5	SMT 求解器	35
3.6	验证结果分析模块	36
3.7	对抗性样本生成模块	36
3.8	本章小结	36
第四章	联邦学习的自适应加噪机制	37
4.1	模型概览	37
4.1.1	威胁模型	37
4.2	鲁棒特征集合	39
4.3	局部鲁棒特征重要度	41
4.4	本章小结	43
第五章	实验与评估	44
5.1	基准数据集介绍	44
5.2	实验环境与配置	45

5.3	实验结果与分析	45
5.3.1	随机森林模型的鲁棒性验证与分析	45
5.3.2	GBDT 模型鲁棒性的验证与分析	47
5.3.3	树模型鲁棒性可解释性的实验与分析	49
5.3.4	不同类别鲁棒性的验证与分析	51
5.3.5	树鲁棒性超参数与鲁棒性关系的验证与分析	53
5.3.6	验证时间的结果与分析	54
5.4	本章小结	55
第六章	总结与展望	57
6.1	总结	57
6.2	展望	58

插图

2.1 打篮球问题的决策树	20
2.2 GBDT 原理图	21
2.3 随机森林原理图	22
3.1 验证框架图	26
3.2 分类树	29
3.3 回归树	30
5.1 随机森林回归模型验证结果	46
5.2 对抗性样本图	46
5.3 GBDT 回归模型验证结果	48
5.4 GBDT 验证反例	48
5.5 随机森林分类模型鲁棒特征集合	49
5.6 GBDT 分类模型鲁棒特征集合	50
5.7 随机森林分类的局部鲁棒性特征重要度	51
5.8 MNIST 中不同类别鲁棒性的验证结果	52
5.9 FASHION-MNIST 中不同类别鲁棒性的验证结果	52
5.10 单样本验证时间图	55

表格

5.1 基于 MNIST 数据集模型在不同超参数下的鲁棒性验证结果. 54

第一章 緒論

1.1 研究背景及意义

随着机器学习的不断发展和壮大，我们一方面惊叹于它的成就，比如 Alpha GO 击败了围棋世界冠军柯洁，或者面部识别技术帮助我们抓住了躲藏多年的逃犯，而大型工业企业也大力推动机器学习技术的应用。另一方面，我们也必须认识到，它的巨大潜力还有待实现，例如：构建基于大量病例的医疗救助诊断系统，运行基于大量商业行为数据的信用风险控制模型，帮助高价值企业融资，并基于整个产业链的数据提供个性化的产品分配和营销策略。我们真正见证了人工智能（AI）的巨大潜力，以及已经开始期待在许多应用中使用更复杂、更尖端的人工智能技术，包括无人驾驶、医疗、金融等今天，人工智能技术几乎在各方面都大显身手每个行业和各行各业。但是传统的机器学习方法依赖于集中管理的训练数据集，建立在大量数据上，从数据中学习特征，从而完成复杂的任务，甚至是人类也难以完成的操作。

然而，这些数据的采集可能涉及到用户的隐私，随着人们的隐私意识的普遍提高，相关的隐私法律法规的不断完善，中国出台的《网络安全与数据合规》白皮书中明确要求加强用户个人信息保护。2018 年欧洲联盟出台《通用数据保护条例》中强调保护用户的个人隐私和数据安全用户可以删除或撤回其个人数据。近年来，也有越来越多的涉及数据泄漏和隐私侵权的事情，用户们也越来越关注自己的隐私信息是否在未经个人许可，或者出于商业和政治目的被他人或机构利用。随着个人意识和国家政策的关注，在大数据和人工智能领域数据采集和使用的过程中，保护用户隐私和数据的机密显得越来越重要。

大多数训练数据是由不同组织的个人或部门产生的，一个 AI 项目可能涉及多个领域，需要融合各个公司、各个部门的数据。（比如研究居民线上消费问题，需要各个消费平台的数据，可能还需要银行数据等等），但在现实中想要将分散在各地、各个机构的数据进行整合几乎是不可能的。传统的机器学习是通过收集数据并将其发送到一个能看到并控制所有数据的中央服务器来完成的。因此，这个中心位置不仅要有强大的计算机集群来训练和创建机器学习模型，还要处理敏感数据并防止数据被用于其他目的。此外，敏感数据的处理方式必须不损害用户的隐私。然而，这用户完全信任服务器的假设已不再适用。在这种情况下，数据拥有者倾向于将数据掌握在自己手中，这就导致了孤立的数据孤岛，数据孤岛使所有利益相关者无法获得更多的数据。例如，每家医院的居民医疗记录的样本量完全不够，导致模型有偏差。在信贷领域，银行只能使用中央银行的信贷报告来建立风险控制模型。

人工智能的力量是基于大数据的，但我们被更多的小数据包围在孤岛中。大数据的基础就没有了，人工智能的基础也没有了。大数据的基础已经消失，人工智能的未来也岌岌可危。要解决大数据的困境，仅仅靠传统的方法已经出现瓶颈。两个公司简单的交换数据在很多法规包括《通用数据保护条例》是不允许的。用户是原始数据的拥有者，在用户没有批准的情况下，公司间不能交换数据。传统的机器学习和深度学习的方法本身已经成为解决大数据困境的绊脚石。简单地在两家公司之间交换数据，无论是《通用数据保护条例》还是 GDPR 都是不允许的：用户是原始数据的所有者，未经其同意，数据不能在公司之间交换。

那如何创建一个机器学习框架，使人工智能系统能够更有效和准确地集体使用数据，同时满足隐私、安全和监管要求，并解决数据孤岛的问题。如何才能做到这一点呢？

为了解决这个问题，google 在 2016 年率先提出了联邦学习的概念，它提供了一个具有隐私保护功能的分布式机器学习框架，并且能够以分布式方式与成千上万的参与者协作，迭代训练一个特定的机器学习模型。由于训练数据在联合过程中

保持在参与者的本地，这种机制允许参与者之间共享训练数据，同时确保每个参与者的隐私 [15]。联合学习的基本工作流程如下：(1) 初始化：所有用户在他们的设备上都有一个预先分配的神经网络模型，并且可以自愿加入联邦学习协议，指定相同的机器学习和模型训练目标。(2) 本地训练：在一个给定的通信回合中，联邦参与者首先从中央服务器下载全局模型参数，然后使用他们的私人训练模式训练模型，创建本地模型更新（即模型参数），并将这些更新发送到中央服务器。(3) 模型平均化。下一轮的全局模型是通过汇总所有通过训练不同的训练模式获得的模型更新并取其平均值来确定的。(4) 迭代地执行上述步骤以达到优化当前全局模型的目的，整个迭代过程将在全局模型参数满足收敛条件时停止。

联合学习在隐私敏感的场景（包括金融、工业和许多其他与数据相关的场景）中显示出巨大的前景，这是因为它具有独特的优势，能够从多个参与者的本地数据中训练出一个统一的机器学习模型，同时保护数据隐私 [16\17]。联合学习解决了数据聚合的问题，并允许一些机器学习模型和算法在各机构和部门之间进行设计和训练。在一些移动设备上的机器学习模型应用中，联邦学习显示出良好的性能和稳健性。此外，对于一些没有足够的私人数据来开发准确的本地模型的用户（客户）来说，机器学习模型和算法的性能可以通过联合学习得到显著改善。

1.2 问题和挑战

1.2.1 数据异构

由于联邦学习的重点是通过以分布式方式从所有参与的客户端设备中学习本地数据来获得高质量的全局模型，所以它无法捕捉每个设备的个人信息，导致推理或分类性能下降。此外，传统的联邦学习要求所有参与的设备同意使用一个共同的模型来共同训练，这在复杂的现实世界物联网应用中是不现实的。研究人员对学习在实际应用中面临的问题总结如下 [2]。

(1) 设备的异质性：由于客户端设备的硬件条件 (CPU、内存)、网络连接 (3G、4G、5G、WiFi) 和电源 (电池) 的变化，联邦学习网络上每个设备的存储、计算和

通信能力都可能不同。由于网络和设备的限制，在任何时候都只有某些设备可以活动。此外，设备可能会受到意外事件的影响，如断电或断网，这可能会导致暂时的断网。这种异质性的系统结构影响了联邦模型的整体学习战略。

(2) 统计的异质性：在整个网络中，设备通常以不同的方式产生和收集数据，而且不同设备的数据量、特征等会有很大的不同，所以联合学习网络中的数据不是独立和相同的分布（非 IID）。目前，目前的机器学习算法主要是基于对 IID 数据的假想假设。因此，非 IID 数据的异质属性给建模、分析和评估带来了重大挑战。[\[19\]](#) 提出了 Federated Averageing (FedAvg) 方法来解决非均匀同分布数据的问题，但是当数据分布偏态很严重的时候 FedAvg 的性能退化严重，一方面其性能比中心化的方法差好多，另一方面它只能学习到 IoT 设备粗粒度的特征而无法学习到细粒度的特征。

(3) 模型的异质性：每个客户根据其应用场景要求定制不同模型。

1.2.2 高昂的通信代价

在联邦学习过程中，根据存储在几十甚至几百万个远程客户端设备上的数据来学习一个全局模型。在训练期间，客户设备必须定期与中央服务器进行通信原始数据被储存在本地的远程客户端设备上，这些设备必须不断地与中央服务器互动，以完成全局模型的构建。通常情况下，整个联盟学习网络可能涉及大量的设备，而网络通信可能比本地计算慢几个数量级，因此高通信成本成为联邦学习的关键瓶颈。

1.2.3 安全性和隐私威胁

(1) 由于联合学习系统的云端服务器无法访问参与者的本地数据和他们的训练过程，恶意参与者可以发送无效的模型更新来达到并破坏全局模型。例如，内部攻击者可以通过在修改后的训练数据上引起有毒的模型更新来有效地损害全局模型的准确性。内部攻击可以由联邦学习服务器发起，也可以由联邦学习参与方发起。外部攻击（包括偷听者）通过参与方与服务器之间的通信通道发起。外部攻击的发

起者大部分为恶意的参与方，例如敌对的客户、敌对的分析者、破坏学习模型的敌对设备或者其组合。在联邦学习中，恶意设备可以通过白盒或者黑盒的方式访问最终模型，因此在防范来自系统外部的攻击时，需要考虑模型迭代过程中的参数是否存在泄露原始数据的风险，这对严格的隐私保护提出了新的挑战。

(2) 由于局部模型更新和全局模型参数的结合提供了关于训练数据的隐藏知识，用户的个人信息有可能泄露给不受信任的服务器或其他恶意用户。例如，即使是由其他用户的训练数据生成的样本原型也会被恶意用户隐蔽地窃取。在训练过程中，攻击方可以试图学习、影响或者破坏联邦学习模型。在联邦训练的过程中，攻击方可以通过数据中毒攻击的方式改变训练数据集合收集的完整性，或者通过模型中毒攻击改变学习过程的完整性。攻击方可以攻击一个参与方的参数更新过程，也可以攻击所有参与方的参数更新过程。若联邦学习的参与方想利用各方的数据集合训练一个模型，但是又不想让自己的数据集泄露给服务器，就需要约定联邦建模的模型算法(例如神经网络)和参数更新的机制(例如随机梯度下降(stochastic gradient descent, SGD))。那么在训练前，攻击方就可以获取联邦学习参数更新的机制，从而指定对应的推断攻击策略。

(3) 在不信任的云服务器和恶意参与者的勾结下，任何个人的确切私人信息都会被泄露。

1.2.4 针对联邦学习的隐私保护

在联邦学习中，存在着无数与隐私有关的挑战学习中的隐私问题。除了保证隐私之外，重要的是要保证确保通信成本的低廉和高效。有许多关于联合学习的隐私定义[8][2][19]。我们可以把它们分为两类：局部隐私和全局隐私。在本地隐私中，每个客户端发送一个不同的隐私值，该值是安全的加密的到服务器。在全局模型中，服务器在最终输出中添加不同的隐私噪音。安全多方计算、同态加密和差分隐私是最常见的技术来保证联盟环境中的安全和隐私。

安全多方计算模型涉及多方，并在一个定义明确的模拟框架中提供安全证明，以保证完全的零知识，即每一方除了其输入和输出外一无所知。零知识是非常理

想的，但这种理想的属性通常需要复杂的计算协议，而且可能无法有效实现。在某些情况下，如果提供安全保证，部分知识的披露可能被认为是可以接受的。有可能在较低的安全要求下建立一个具有 SMC 的安全模型，以换取效率 [16]。最近，一项研究 [46] 将 SMC 框架用于训练具有两个服务器和半诚实假设的机器学习模型。在 [33] 中，MPC 协议被用于模型训练和验证，而用户不会泄露敏感数据。最先进的 SMC 框架之一是 Sharemind[8]。[44] 的作者提出了一个具有诚实多数的 3PC 模型 [5,21,45]，并考虑了半诚实和恶意假设的安全性。这些作品要求参与者的数据在非共存的服务器之间秘密共享。

同态加密是一种加密形式，它允许人们对密文进行特定形式的代数运算得到仍然是加密的结果，将其解密所得到的结果与对明文进行同样的运算结果一样同态加密 [53]，明文通过同态加密方法得到密文后，可实现密文间的计算（密文计算后解密的结果等价于明文计算的结果）。如果对密文进行加法（或乘法）运算后解密，与明文进行加法（或乘法）运算，结果相等，则称这种加密算法为加法（乘法）同态。如果同时满足加法和乘法同态，则称为全同态加密。在联邦学习中，因为只需要对中间结果或模型进行聚合，一般使用的同态加密算法为 PHE（多见为加法同态加密算法），通过加密机制下的参数交换来保护用户数据隐私 [24, 26, 48]，例如在 FATE 中使用的 Paillier 即为加法同态加密算法。

差分隐私方法涉及向数据添加噪音，或使用概括方法来掩盖某些敏感属性，直到第三方无法区分个人，从而使数据无法被还原以保护用户的隐私。利用差分隐私，可以在本地模型训练及全局模型整个过程中对相关参数进行扰动，从而令敌手无法获取真是模型参数，但是与密码学技术相比，差分隐私无法保证参数传递过程中的机密性，从而增加了模型遭受隐私攻击的可能性。例如刘俊旭等人 [10] 针对联邦学习下差分隐私中存在的攻击方法进行了详细的调研。在 [23] 中，作者为联合学习引入了一种差异化的隐私方法，以便通过在训练期间隐藏客户端的贡献来增加对客户端数据的保护。在深度学习中，差分隐私可以作为一种局部隐私保护方案来保护用户梯度的隐私，Abadi 等人 [43] 提出了一种隐私保护的深度学习方法，主

要通过使用噪声来扰乱少量步骤后的局部梯度，将差分隐私机制与 SGD 算法相结合。令人担忧的是，隐私保护预算的成本和联合学习的有效性之间的权衡是困难的，因为较高的隐私保护预算可能对一些大规模的攻击（如基于 GAN 的攻击）不是很有用 [50]，而较低的隐私保护预算可能阻碍模型的局部收敛。

总的来说，安全多方计算基于复杂的计算协议，同态加密的运算成本非常高，而差分隐私破坏了数据的可用性，很难在模型性能和隐私成本上达到平衡，当前的研究方向主要集中在对数据集和神经网络中的参数的加密和隐私保护机制上，较少关注到模型整体框架等过程。目前的联邦学习中的隐私保护方法还有许多不足，不能在隐私性和模型可用性上都达到一个相对满意的效果，此外，大部分方法是基于统一的、固定的参数设置，会导致模型迭代过程中累积大量隐私损失，使模型性能大幅下降。因此，在联邦学习场景下，保护用户隐私的同时保持模型准确性仍需大量的研究，

1.3 本文工作与主要贡献

针对联邦学习中隐私性和模型精度的双重指标，本文提出了参数匿名上传框架和自适应差分隐私算法，主要的工作和贡献包含以下三个方面：

- (1) 本文提出了一个新的参数聚合框架，该框架支持在参数上传过程中，对于每一个本地模型，通过两个重要实现：拆分和混洗，扰乱模型中各个参数的隐私关联和各个模型之间的隐私关联，实现客户端匿名。
- (2) 本文提出了一个自适应扰动方案，对联邦学习过程中双方所交互的梯度进行分析，在所交互的梯度上添加扰动，并基于梯度自适应加噪，进一步减少隐私预算。
- (3) 本文针对模型训练和上传过程中的隐私安全问题，将改进的参数聚合框架和自适应扰动方案引入联邦学习框架，实现混合隐私保护的联邦学习系统，每个用户在本地训练数据时添加自适应扰动，并在向中心服务器上传时实现客

户端匿名，实现了客户端的数据隐私。

- (3) 本文通过实验，展示了自适应假造方案和参数聚合框架的结合，使得联邦学习的模型的精度和隐私预算达到平衡。

1.4 本文组织结构

本文一共六章，主要内容的组织安排如下：

第一章对本文研究内容：联邦学习的研究背景和实际意义进行了阐述，介绍了目前联邦学习中的隐私保护的研究现状和发展方向。

第二章详细介绍本文研究内容所涉及的一些理论基础与背景知识，包含了联邦学习的相关概念，差分隐私的基础知识。

第三章描述了本文所提出的参数聚合框架的设计和实现。我们首先对框架的整体进行了介绍，之后给出了各个模块的设计和实现细节。

第四章描述了本文所提出的自适应加噪方案，讨论了隐私预算与的关系，并且详细描述了相关概念和算法。

第五章为实验部分，基于本文提出的隐私保护框架，我们在三个基准数据集的进行了实验和讨论。

第六章是对本文的一个内容总结和展望，首先对本文的研究内容进行了概括，并对现有的不足进行总结，对未来的研究和改进方向进行了展望。

第二章 相关知识介绍

我们在本章节中介绍了本文研究所需要的一些基本知识，有助于更好的理解之后章节的内容。

2.1 差分隐私

差异化隐私作为一种隐私保护方法是为一个用户服务的，因为根据隐私的定义，隐私泄露只是与特定用户有关的信息泄露，而一组用户的统计特征不包括在隐私信息中。如果一个对象在数据库中的存在或不存在，或其价值的变化不会对搜索结果产生重大影响，那么该对象的隐私信息就会受到保护，这就是差异性隐私（DP）概念的起源。差异隐私首先被应用于数据查询，为了更好地说明数据集之间的差异，定义了相邻数据集的概念：两个数据集只差一个信息或只差一个数值不同的记录。因此，查询数据库相关信息的攻击者将无法以任何概率确定 \square 是否存在于数据集中，而成员 \square 被认为是相对安全的。

2.1.1 基本定义

对于一个有限域 $Z, z \in Z$ 为 Z 中的元素，从 Z 中抽样所得 z 的集合组成数据集 D ，其样本量为 n ，属性的个数为维度 d 。对数据集 D 的各种映射函数被定义为查询 (Query)，用 $F = \{f_1, f_2, \dots\}$ 来表示一组查询，算法 M 对查询 F 的结果进行处理，使之满足隐私保护的条件，此过程称为隐私保护机制。设数据集 D 和 D' ，具有相同的属性结构，两者的对称差记作 $D \Delta D'$ ， $|D \Delta D'|$ 表示 $D \Delta D'$ 中记录的数量。若 $|D \Delta D'| = 1$ ，则称 D 和 D' 为邻近数据集 (Adjacent Dataset)。

定义 2.1.1 (成立条件). 若随机算法 $M : D \rightarrow R$ 满足 $(\varepsilon, \delta) - DP$, 当且仅当相邻数据集 d, d' 对于算法 M 的所有可能输出子集 $S \in R$ 满足不等式^[40] :

$$\Pr[M(d) \in S] \leq e^\varepsilon \Pr[M(d') \in S] + \delta$$

其中, ε 表示隐私预算参数, ε 越小意味着隐私预算越低, 信息泄露越少, 隐私保护的强度越高。添加项 δ 代表允许以概率 δ 打破 $\varepsilon - DP$ 的可能性, 其值通常选择为小于 $1/|D|$. 当 $\delta = 0$ 时, 定义转化为 $\varepsilon - DP$, 这时机制提供了更加严格的隐私保护。隐私预算参数决定着隐私保护强度, 针对传统数据库保护, 当 $\varepsilon \in (0, 1)$ 时认为隐私保护强度是有效的, 但应用在深度学习领域, $\varepsilon \in (0, 10)$ 都认为是可以被接受的合理范围。如图 1 所示, 算法 M 通过对输出结果的随机化来提供隐私保护, 同时通过参数 ε 来保证在数据集中删除任一记录时, 算法输出同一结果的概率不发生显著变化.

2.1.2 相关概念

差分隐私保护可以通过在查询函数的返回值中加入适量的干扰噪声来实现. 加入噪声过多会影响结果的可用性, 过少则无法提供足够的安全保障. 敏感度是决定加入噪声量大小的关键参数, 它指删除数据集中任一记录对查询结果造成的变化. 在差分隐私保护方法中定义了两种敏感度, 即全局敏感度 (Global Sensitivity) 和局部敏感度 (Local Sensitivity).

定义 2.1.2 (全局敏感度). 设有函数 $f : D \rightarrow R^d$, 输入为一数据集, 输出为一 d 维实数向量. 对于任意的邻近数据集 D 和 D' ,

$$GS_f = \max_{D, D'} \|f(D) - f(D')\|_1$$

称为函数 f 的全局敏感度。函数的全局敏感度由函数本身决定, 不同的函数会有不同的全局敏感度. 一些函数具有较小的全局敏感度 (例如计数函数, 其全局敏感度为 1), 因此只需加入少量噪声即可掩盖因一个记录被删除对查询结果所产生的影响, 实现差分隐私保护。

定义 2.1.3 (局部敏感度). 对于一个查询函数 $f: D \rightarrow R^d$, 其中 D 为一个数据集, R^d 为 d 维实数向量, 是查询的返回结果。对于给定的数据集 D 和它的任意邻近数据集 D' , 有 f 在 D 上的局部敏感度为: $LS_f(D) = \max_{D'} \|f(D) - f(D')\|_1$

局部敏感度由函数和及给定数据集中的具体数据共同决定。由于利用了数据集的数据分布特征, 局部敏感度通常要比全局敏感度小得多。敏感度代表了查询函数针对相邻数据集的输出的最大不同, 或者说量化评估了最坏情况下单个样本对整体数据带来的不确定性大小。敏感度函数仅与查询函数的类型有关, 为扰动的添加提供了依据。但是, 由于局部敏感度在一定程度上体现了数据集的数据分布特征, 如果直接应用局部敏感度来计算噪声量则会泄露数据集中的敏感信息。

全局差分隐私技术旨在实现这样一个目标: 如果替换数据集中的任意样本的效果足够小, 则查询结果不能被用来探索数据集中任何样本的更多信息 [43]。作为一种优势, 这种技术比局部差分隐私技术更准确, 因为它不需要向数据集添加大量的噪声。局部差分隐私技术被引入以去除全局差分隐私中所要求的受信任的中央机构 [34,102]。与全局差分隐私技术相比, 局部差分隐私技术不需要可信的第三方 [146]。其缺点是, 噪声总量比全局差分隐私技术大得多。

定义 2.1.4 (组合定理). 在差分隐私部署过程中常常不仅仅在一处添加噪声, 也不仅仅针对数据集进隐私预算的分配有序列组合性 [41] 和并行组合性 [42] 两种组合特性: (a) 序列组合: 给定 n 个随机算法 $M_i (1 \leq i \leq n)$ 满足 $\varepsilon_i - DP$, 那么针对一个数据库 D 而言, 在 D 上的算法序列组合可以提供 $\varepsilon - DP$, 其中 $\sum_{i=1}^n \varepsilon_i = \varepsilon$ 。

(b) 并行组合: 对于数据库 D , 当其被划分成 n 个不相交的子集 $\{D_1, D_2, \dots, D_n\}$, 在每个子集上应用算法 M_i , 每个算法提供 $\varepsilon_i - DP$, 则在序列 $\{D_1, D_2, \dots, D_n\}$ 上整体满足 $(\max \{\varepsilon_1, \dots, \varepsilon_n\}) - DP$

2.1.3 差分隐私的实现机制

在实践中为了使一个算法满足差分隐私保护的要求, 对不同的问题有不同的实现方法, 这些实现方法称为“机制”。拉普拉斯机制 (Laplace Mechanism)、指数

机制 [22](ExponentialMechanism) 与高斯机制是三种最基础的差分隐私保护实现机制。其中，Laplace 机制和高斯适用于对数值型结果的保护，指数机制则适用于非数值型结果。

在中心化差分隐私中，最为常用的扰动机制是拉普拉斯 (Laplace) 机制，该机制可以后期处理聚合查询（例如，计数、总和和均值）的结果以使它们差分私有。Laplace 分布是统计学中的概念，是一种连续的概率分布。

定义 2.1.5 (拉普拉斯机制). 如果随机变量的概率密度函数分布为：

$$f(x | \mu, b) = \frac{1}{2b} \exp\left(-\frac{|x-\mu|}{b}\right) = \frac{1}{2b} \begin{cases} \exp\left(-\frac{\mu-x}{b}\right) & x < \mu \\ \exp\left(-\frac{x-\mu}{b}\right) & x \geq \mu \end{cases} \quad \text{其中, } D \text{ 表示数据集,}$$

$f(D)$ 表示的是查询函数, Y 表示的是 Laplace 随机噪声, $M(D)$ 表示的是最后的返回结果。 $M(D) = f(D) + Y$ 如果噪声 $Y \sim L(0, \frac{\Delta f}{2})$ 满足 $(\epsilon, 0)-$ ，则表示服从拉普拉斯分布的随机噪声。因此，当隐私预算 ϵ 确定时，敏感度越大，引入的噪声量越大。

对于非数值型的查询结果或数据，通常使用指数机制来随机选择离散的输出结果来满足差分隐私。指数机制整体的思想就是，当接收到一个查询之后，不是确定性的输出一个 R_i 结果，而是以一定的概率值返回结果，从而实现差分隐私。而这个概率值则是由打分函数确定，得分高的输出概率高，得分低的输出概率低。

定义 2.1.6 (指数机制). 指数机制满足差分隐私，如果：

$$M(D) = (\text{return } \varphi \propto \exp\left(\frac{\epsilon q(D, \varphi)}{2\Delta q}\right))$$

评分函数 $q(D, \varphi)$ ，用于评估输出 φ 的质量. Δq 代表了输出的敏感度，

2.2 联邦学习

传统的集中式深度学习需要将训练数据放在一起，交给一个数据中心。模型是以集中的方式进行训练的。而联合学习允许数据所有者持有一个私人学习网络，用本地数据集进行训练。之后，每个参与者将本地模型的梯度上传到云服务器。通

过更新云服务器上收集的全局梯度，可以避免本地模型的过度拟合。此外，它还可以保护本地数据不被其他参与者或云服务器直接知道。

2.2.1 基本定义

2.2.2 联邦学习的分类

联合学习通常分为水平联邦学习、纵向联邦学习和联合迁移学习，这是由 Yang Q 等人提出的 [8]。这种分类是基于用户维度和特征维度的重合。

- **水平联邦学习**: 当两个数据集的用户属性重叠较多而用户重叠较少的情况下，我们对数据集进行横向切割（即按用户维度切割），删除两边用户属性相同但用户不完全相同的那部分数据，用于训练。这种方法被称为横向联合学习。例如，两家银行位于不同的地区，有来自各自地区的用户群，而且它们之间的联系非常少。然而，他们的业务活动非常相似，因此他们的用户特征也是一样的。在这个阶段，我们可以使用跨部门的联合学习来建立一个联合模型。2016 年，谷歌提出了一个在安卓手机上更新模型的联合数据建模系统：模型参数在本地不断更新，并在各个用户使用安卓手机时上传到安卓云端，使拥有数据的每一方都能建立一个具有相同特征维度的联合模型。
- **纵向联邦学习** 在两个数据集与用户重叠较多而与用户属性重叠较少的情况下，我们将数据集纵向切开（即按特征维度），选择数据集中两边用户相同但用户属性不完全相同的部分进行训练。这种方法被称为纵向的联合学习。例如，有两个不同的组织，一个是在一个地方的银行，另一个是在同一个地方的电子商务公司。他们的用户群很可能包括该地的大部分人口，所以有很大的用户交集。然而，由于银行储存的是用户的收入和支出以及信用评分的数据，而电子商务公司储存的是用户的浏览和购买历史的数据，他们的用户档案并没有那么紧密的联系。长期的联合学习是在一个加密的空间里将这些不同的功能结合起来，以提高模型的性能。渐渐地，人们发现可以在这个联合系统之上建立若干机器学习模型，如逻辑回归、树状结构和神经网络模型。

- **联合迁移学习**联合迁移学习是通过使用迁移学习景观来弥补数据或标签的差距，而不是对数据进行切分，两个数据集中的用户和用户属性几乎没有重叠。这种方法被称为混合式学习迁移。作为一个例子，考虑两个不同的组织，一个是中国的银行，另一个是美国的电子商务公司。由于地理上的限制，这两个机构的用户群重叠的地方很少。由于它们是不同类型的组织，数据的特点也没有太多的重叠。在这种情况下，为了保证有效的联邦学习，有必要引入反式学习，以克服单变量数据量小和标注样本小的问题，提高模型的效率。

2.2.3 联邦学习的训练步骤

在很多横向联邦学习应用场景中，参与训练的参与方数据具有类似的数据结构(特征空间)，但是每个参与方拥有的用户是不相同的。有时参与方比较少，例如，银行系统在不同地区的两个分行需要实现联邦学习的联合模型训练；有时参与方会非常多，例如，做一个基于手机模型的智能系统，每一个手机的拥有者将会是一个独立的参与方。针对这类联合建模需求，可以通过一种基于服务器客户端的架构来满足很多横向联邦学习的需求。将每一个参与方看作一个客户端，然后引入一个大家信任的服务器来帮助完成联邦学习的联合建模需求。在联合训练的过程中，被训练的数据将会被保存在每一个客户端本地，同时，所有的客户端可以一起参与训练一个共享的全局模型，最终所有的客户端可以一起享用联合训练完成的全局模型。

- 步骤 1: 中心服务器初始化联合训练模型，并且将初始参数传递给每一个客户端。
- 步骤 2: 客户端用本地数据和收到的初始化模型参数进行模型训练。具体步骤包括：计算训练梯度，使用加密、差分隐私等加密技术掩饰所选梯度，并将加密后的结果发送到服务器。
- 步骤 3: 服务器执行安全聚合。服务器只收到加密的模型参数，不会了解任何

客户端的数据信息，实现隐私保护。服务器将安全聚合后的结果发送给客户端。

- 步骤 4: 参与方用解密的梯度信息更新各自的本地模型，具体方法重复步骤 2。

2.2.4 联邦学习中的安全和隐私威胁

尽管联邦学习提供了隐私保护的机制，还是有各种类型的攻击方式可以攻击联邦学习系统，从而破坏联邦学习系统安全和参与方的隐私。本节将讨论关于联邦学习的攻击问题。从参与方的类型来看，可以将联邦学习的威胁模型细分为半诚实模型 (semi-honest model) 和恶意模型。对于联邦学习系统的攻击，本文按照不同的维度进行不同层次的分类。从攻击方向角度来看，可以将联邦学习的攻击分为从内部发起和从外部发起两个方面。从攻击者的角色角度来看，可以将攻击分为参与方发起的攻击、中心服务器发起的攻击和第三方发起的攻击。从发动攻击的方式角度来看，可以将攻击分为中毒攻击和拜占庭攻击。从攻击发起的阶段角度来看，可以将攻击分为模型训练过程的攻击和模型推断过程的攻击。在密码学领域，基于模型安全的假设通常可以被分为半诚实但好奇 (onest but curious) 的攻击方假设以及恶意攻击方假设。

- 半诚实但好奇的攻击方半诚实但好奇的攻击方假设也被称为被动攻击方假设。被动攻击方会在遵守联邦学习的密码安全协议的基础上，试图从协议执行过程中产生的中间结果推断或者提取出其他参与方的隐私数据。半诚实但好奇的供给方通常是客户端的角色，它们可以检测从服务器接收的所有消息，但是不能私自修改训练的过程。在一些情况下，安全包围或者可信执行环境 (trusted execution environment, TEE) 等安全计算技术的引入，可以在一定程度上限制此类攻击者的影响或者信息的可见性。半诚实但好奇的参与方将很难从服务器传输回来的参数中推断出其他参与方的隐私信息，从而威胁程度被削弱。

- **恶意攻击方**恶意攻击方也被称为主动攻击方。由于恶意攻击方不会遵守任何协议，为了达到获取隐私数据的目的，可以采取任何攻击手段，例如破坏协议的公平性、阻止协议的正常执行、拒绝参与协议、不按照协议恶意替换自己的输入、提前终止协议等方式，这些都会严重影响整个联邦学习协议的设计以及训练的完成情况。恶意的参与方可以是客户端，也可以是服务器，还可以是恶意的分析师或者恶意的模型工程师。恶意客户端可以获取联邦建模过程中所有参与方通信传输的模型参数，并且进行任意修改攻击。恶意服务器可以检测每次从客户端发送过来的更新模型参数，不按照协议，随意修改训练过程，从而发动攻击。恶意的分析师或者恶意的模型工程师可以访问联邦学习系统的输入和输出，并且进行各种恶意攻击。
- **投毒攻击**在联邦学习框架中，攻击者可能试图修改、删除或插入恶意信息到训练数据中，以破坏原始数据分布，改变学习算法的逻辑两种常见的中毒攻击的例子包括标签翻转攻击 [34] 和后门攻击 [19]。标签反转攻击是指恶意用户反转样本标签，并在训练数据中加入预定义的攻击点，导致训练后的模型偏离预测的界限。与标签反转攻击不同，后门要求攻击者用精心设计的训练数据，利用特定的隐藏模式来训练目标的深度神经网络（DNN）模型。这些模型被称为“反馈回路”，可以干扰学习模型，并在预测阶段产生与真实情况截然不同的结果。
- **成员推理攻击**如上文所述，联邦学习机制要求所有参与者通过在本地数据集上训练全局模型来更新梯度。在这种情况下，如果联邦学习系统有一个不被信任的服务器，其知识不能被信任，那么用户的私人信息就不能得到保证。这个不受信任的服务器可以获得关于每个参与者的本地训练模型的大量额外信息（例如，模型结构、用户身份和梯度），并且能够充分损害用户的隐私信息。具体实现如下：攻击者首先在平均化后获得模型的全局参数，并在本地存储这些快照。然后，通过计算以下快照与进一步删除添加的更新，以获得其他用户的模型的汇总更新。通过这种方式，攻击者可以利用数据集的协助，

得出所有其他参与者共同合作的数据样本。

- **GAN 攻击** Hitaj 等人 [21] 发现，一个联邦学习框架非常容易受到系统内参与者发起的主动攻击。他们首先提出了一个由系统内的恶意用户发起的基于 GAN 的重建攻击。在训练阶段，攻击者可以冒充无害的用户，训练 GAN 来模拟由其他用户的训练数据产生的原型样本。通过不断添加假的训练样本，攻击可以逐渐影响整个学习过程，使受害者暴露出更多关于攻击者的目标类的敏感信息。除了客户端发起的 GAN 攻击，服务器也能通过 GAN 攻击。恶意服务器最初假装是一个为用户提供联邦学习服务的正常服务器，但其主要目标是重建被攻击用户的训练样本。

2.3 神经网络

定义 2.3.1 (布尔公式).

- 布尔公式可以是单独的布尔变元，也称为原子公式。
- 如果 α 是布尔公式，那么 $(\neg\alpha)$ 也是布尔公式。
- 如果 α 和 β 是布尔公式，那么 $(\alpha \wedge \beta)$ 和 $(\alpha \vee \beta)$ 也是布尔公式。
- 按照以上 3 条规则生成的表达式也是公式。

在定义 2.3.1 中，带符号的布尔变元称为布尔文字。对于一个布尔变元 x , x 是它的公式正文字, $\neg x$ 是它的公式负文字。对于一个公式文字 l , l_x 是它相对应的布尔变元。若布尔变元 x 在赋值 a 的真值赋值 False，则公式文字 a 的真值赋值也为 True，反之，公式文字 a 的真值赋值 False。根据布尔公式文字析取而构成的子公式我们称之为子句。子句可以通过合取的形式构成布尔公式。并且单独的布尔公式文字也能构造成单独的子句。同理，单独的子句也可以看做是一个布尔公式。

定义 2.3.2 (布尔公式的赋值). 布尔公式的赋值 a 是一个将公式中的布尔变元映射到 $\{True, False\}$ 集合上的函数。对于一个赋值 a , 将 $a(x)$ 称为变元 x 在赋值 a 上的真值取值。

定义 2.3.3 (布尔公式的可行解). 如果称赋值 v 为布尔公式 α 的可行解, 则意味着在赋值为 v 的情况下, 公式 α 真值取值为 *True*。可表示为: $v \models \alpha$ 。

给定一个布尔公式 α 和其赋值 v , 将公式 α 中的所有的布尔变元映射为相应的逻辑真值之后, 我们可以计算得出 α 对应的逻辑真值。如果至少存在一个可行解 v , 使得布尔公式 α 的逻辑真值为 *True*, 那么公式 α 就是可满足的。我们用 $V(\alpha)$ 表示公式 α 的所有可行解集合。对于布尔公式 α 来说, 如果 $V(\alpha)$ 中的所有赋值都是 α 的可行解, 那么公式 α 就是永真的; 与之相对的是, 如果所有的赋值都不是 α 的可行解, 那么该公式 α 就是不可满足的。

2.3.1 SMT 技术

在过去几十年中, SAT 求解技术取得了巨大的进步。但根据上一小节所描述, 我们知道 SAT 面向的是命题逻辑公式, 因此表达能力的角度来看, SAT 有着一定的局限性, 而且抽象层次不高。因此在解决实际问题的时候, SAT 会丢失很多高层次的信息。为了弥补 SAT 技术的不足, 人们将 SAT 扩展为可满足性模理论 (Satisfiability Modulo Theories, SMT) [? ?]。SMT 是基于 SAT 技术加入模块理论背景后拓展而来的。与 SAT 不同的是, 在命题逻辑的基础上, SMT 中增加了量词和项的概念, 基础的研究对象是一阶逻辑公式, 使得 SMT 有着更强的描述问题能力。除此之外, SMT 还可以融合不同的背景理论, 使得公式中的命题变量可以是理论公式, 所以它可以直接描述问题中的抽象层次比较高的部分。与 SAT 公式类似, 可以通过 SMT 约束的合取构成 SMT 公式, 而子句可以通过 SMT 谓词析取组成, 其中的 SMT 谓词则由 SMT 变元和相应的运算符构成; 与 SAT 公式不同的是, 在不同的模块背景理论中, SMT 公式中的变元和运算符的类型也会不同。

定义 2.3.4 (SMT 赋值). 给定一个 SMT 公式 F , 公式 F 的赋值 a 为一个函数 f , f 会基于公式 F 的模块背景理论 T 将所有的变元 x 映射为在 T 中相对应的取值, 则有 $f : \rightarrow R(T)$, 其中 $R(T)$ 是 T 中变量取值范围。

定义 2.3.5 (SMT 可行解). 如果称赋值 a 为 SMT 公式 F 的可行解，则意味着在赋值为 a 的情况下，公式 F 真值取值为 *True*。

定义 2.3.6 (SMT 公式可满足性). 给定一个 SMT 公式 F ，如果至少存在一个可行解 v 使得公式 F 的真值取值为 *True*，那么 F 就是可满足的，可表示为： $v \models F$ ；反之，如果 F 是不可满足的，则意味着 F 的任意的一个赋值 a 都有 $a \not\models F$ 。

根据定义 2.3.6，我们知道判断一个 SMT 公式是否可满足的关键点是需要判断该公式的是否存在可行解，如果存在一个可行解使得该公式取值为 *True*，则是可满足的。否则，该公式就是不可满足的。

目前有很多研究利用 SMT 技术去验证其他机器学习的模型 [? ?]，在本文中，我们同样选择基于 SMT 技术去完成树模型鲁棒性的验证问题，主要用到的背景理论为算数逻辑，在原来的运算符的基础上，增加了加法 (+)，乘法 (\times) 和比较操作符 ($>$, $<$) 等。常用的变量类型包括了整数，浮点数和布尔类型等。

2.4 决策树

决策树是一种比较常用的机器学习算法，用于监督学习中，它既可以解决分类问题也可以解决回归问题，一般情况下，单棵决策树由根结点，内部结点和叶子结点构成。根节点一般会包括所有的样本，从根结点到每个叶子结点的路径对应着一条的判断路径，预测结果即为叶子结点所绑定的值。决策树学习的目的是为了产生一颗泛化能力强的决策树。通常，我们可以通过自上而下、分而治之的策略递归的构建一颗决策树。一棵用于判断是否可以打篮球的决策树如图 2.1 所示：

决策树学习中的关键点是怎样确定最优的特征去划分样本。通常，随着划分过程的不断进行，内部结点中的样本应该尽可能的属于同一类别，这样才能使得划分后的各个不同类别样本的“纯度”(purity) 更高，不确定性更小。一般来说，特征选择的算法有 ID3、C4.5、CART 等，分别对应不同的规则去选择当前最优的特征进行决策树的构造。

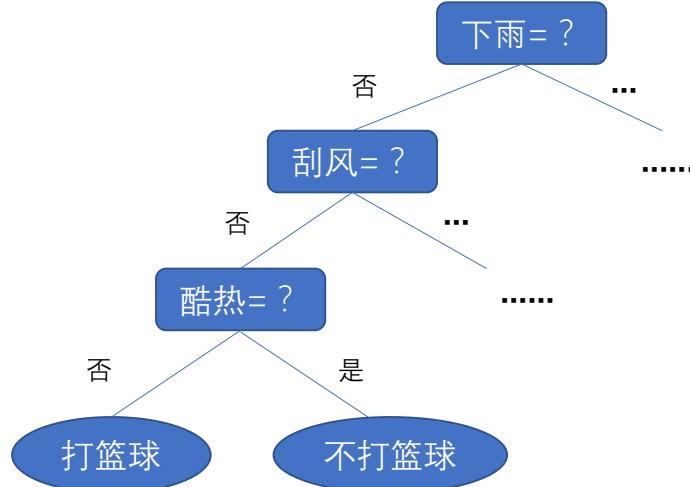


图 2.1: 打篮球问题的决策树

在机器学习的过程中，经常会出现过拟合的问题，即由于模型在训练样本集上过度学习而导致在预测新的样本的时候效果不好。决策树同样也存在过拟合的问题，当它出现过拟合的时候，通常可以进行剪枝操作来预防。剪枝操作可以提高决策树的泛化能力。具体来讲，剪枝操作就是去掉树结构中的一些分支，所以可能会导致树的深度或宽度降低。决策树的剪枝主要分为预剪枝和后剪枝。预剪枝指的是决策树在构造的过程中，将一些不必要的分枝去除；后剪枝则是在整个决策树构造完成后，从叶子节点开始从下往上进行遍历，尝试将该包含节点子树替换为叶子节点，如果替换后能够提高决策树的泛化能力，就将该节点设置为叶子节点。

2.5 树模型的基本概念

树模型简单来说就是以决策树为个体学习器，再通过某种策略将它们结合起来的模型。通常，利用投票的方式来产生模型预测结果。在机器学习中，这种模型结构通常被称为“集成学习”，但由于本文主要研究的是以决策树为基学习器的集成学习模型，所以我们将此模型描述为树模型更为贴切。一般在集成学习模型中，需要尽可能的要求基学习器好而不同。换句话说，我们希望每个个体学习器既要具有准确性又要有多样性，从而保证该模型有着更强的泛化能力，因此基学习器的生成方式也就尤为重要。当前根据基学习器（本文中为决策树）的生成方式，可

以将其分为两大类：1. 每个个体学习器之间存在强依赖关系，必须串行生成的类型。2. 每个个体学习器之间没有关联，因此可以并行生成的类型。第一种的代表是 Boosting，第二种的代表是 Bagging。本文提出的树模型的鲁棒性验证框架，覆盖了以上两种类型中的典型代表：GBDT 模型和随机森林模型。

2.5.1 GBDT

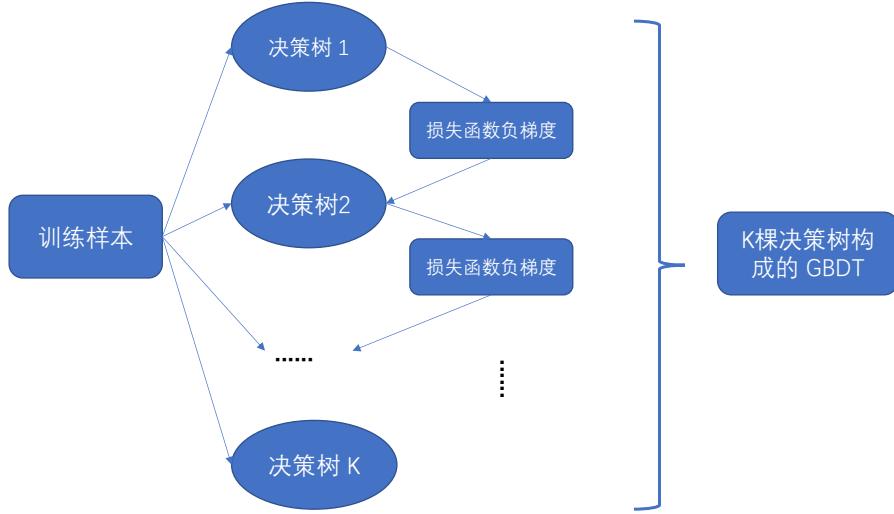


图 2.2: GBDT 原理图

对于 Boosting 类别来说，我们在本文中主要研究其中具有的代表性 GBDT 模型。GBDT 已经在多个领域得到了很好的应用 [??]?。GBDT 是 Jerome Friedman 提出的一种基于迭代思想的决策树算法。GBDT 的全称是 Gradient Boosted Decision Tree，其基本原理如图2.2。GBDT 和其他的 Boosting 算法一样，通常将表现一般的数个决策树结合在一起构成一个预测效果更好的模型。它的核心思想是在每次迭代中构建新的 CART 回归树时候，通过拟合当前模型损失函数的负梯度，来最小化损失函数。GBDT 用于分类任务和回归任务时都使用 CART 回归树作为基学习器，分类时使用指数损失或对数损失，回归时使用平方误差损失函数，绝对值损失函数等。在本文的研究中，我们需要根据不同的预测任务和训练集来选择相应的损失函数来保证模型有好的预测效果。

2.5.2 Bagging 与随机森林

在并行类型的集成学习算法中，Bagging 是其中最有代表性的。它利用了自助采样法（bootstrap sampling）的方法，采取有放回采样策略，通过重复多次采样操作得到预设数目的采样集。由于每次都是随机采样，所以会使有的样本会在采样集里多次出现，有的样本则从来没有出现。之后基于每个的采样集训练出一个基学习器，再这些基学习器进行结合。在对基学习器的预测结果进行结合时，可以采用不同的投票方式来确定最终的预测结果。

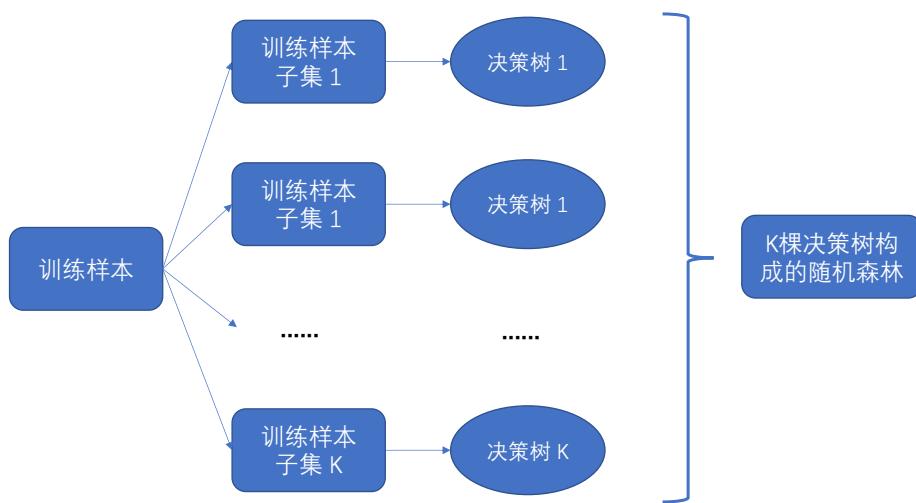


图 2.3: 随机森林原理图

随机森林（Random Forest, 简称 RF）是基于 Bagging 算法的一个扩展，其基本原理如图2.3。它由 Breiman 和 Cutler Adele [?] 所提出，可以用于分类和回归任务。随机森林是在以决策树为基学习器结合 Bagging 思想的基础上，在决策树的训练过程中加入了随机特性，从而使得整个模型获得更好的泛化能力。随机森林的随机特征主要包括两个方面：训练样本的随机抽样和特征的随机选择。由于这些随机特性，使得随机森林不容易发生过拟合。但并不代表因为加入了随机就不会发生过拟合，比如森林中树的数目过多，抑或树的深度过深都会发生过拟合。

随机森林的预测结果是综合多棵决策树的结果，如果处理的是分类任务，输出的结果是多棵决策树投票的结果，如果处理的是回归任务，则是多棵决策树的

平均值。

无论是 GBDT 模型还是随机森林模型，模型中树的棵树和树的深度都对模型的泛化能力和过拟合问题都有着很大的影响。在训练模型的过程中，我们也要通过调节这些超参数，来获得预测效果好的模型。

2.6 机器学习模型鲁棒性

本小节会介绍目前对于机器学习模型鲁棒性的研究现状，并且最后给出树模型鲁棒性的形式化定义。在之后的鲁棒性验证环节中，会根据这些定义去验证树模型的鲁棒性。

对于机器学习模型来说，鲁棒性反映了模型对输入特征扰动的抗干扰能力，鲁棒性越强说明输入特征微小扰动对模型的预测结果影响越小，相反对预测结果影响越大，甚至会导致预测结果完全错误的情况。目前，线性模型和神经网络的对抗性样本和模型鲁棒性已经得到了广泛的研究 [??]，但是对于树模型的鲁棒性的研究仍然有限。与神经网络模型不同，树模型是基于树的模型是非光滑的，不可微的，有时是可解释的，这可能会让人相信它们比 DNNs 更健壮。但通过我们的研究发现，与神经网络模型一样，树模型也容易遭到对抗性样本的攻击。所以验证树模型的鲁棒性就很有必要。

与神经网络模型一样，树模型的鲁棒性也保证了模型在受到对抗性样本的干扰下预测结果依然正确。在之前的关于树模型的介绍中，我们知道树模型可以用于分类任务和回归任务。所以对于不同的预测任务来说，其模型的鲁棒性的定义也是不同的。具体来讲，对于分类任务，模型的鲁棒性需要保证分类结果在输入特征扰动的情况下，预测结果会保持不变；对于回归任务来说，则需要保证模型的预测结果的变化是有界的。

参考神经网络模型鲁棒性的定义，结合树模型的特性之后，我们可以得到树模型的鲁棒性的定义。如果树模型用于分类任务，则我们称该树模型为分类模型，用符号 C 表示；如果树模型用于回归任务，则我们称该树模型为回归模型，用符

号 R 表示；我们用符号 x 表示测试集中的一个输入样本， x' 表示基于 x 的对抗性样本，符号 ϵ 表示输入样本的扰动范围，符号 δ 表示回归任务中预测结果的变化范围，则可以给出树模型鲁棒性的形式化定义如下：

定义 2.6.1 (回归模型的单样本鲁棒性). 给定回归模型 R , x 为测试样本, $R(x)$ 为模型 R 基于样本 x 的预测结果。如果 R 对样本 x 的所有对抗性样本 x' 的预测结果的变化都是有界的，则该回归模型 R 满足单样本鲁棒性，否则不满足。其中 $\|x - x'\|_p < \epsilon$, $|R(x) - R(x')| \leq \delta$. 其中 p 表示对抗性样本距离。

定义 2.6.2 (分类模型的单样本鲁棒性). 给定为分类模型 C , x 为测试样本 $C(x)$ 为模型 C 基于样本 x 的预测结果。如果 C 对样本 x 的所有对抗性样本 x' 的预测结果都不发生变化，则该分类模型 C 满足单样本鲁棒性，否则不满足。其中 $\|x - x'\|_p < \epsilon$, $C(x) = C(x')$. 其中 p 表示对抗性样本距离。

在定义2.6.1和定义2.6.2中我们用 p 表示对抗性样本距离，通常在机器学习鲁棒性验证中，可以用不同的范数来表示输入特征值的扰动距离，在本文的研究中，我们考虑范数 $p = \infty$ 的情况，它限制了每个特征在 x 和 x' 之间的最大扰动。例如，可以将两幅图像之间的距离计算为对应像素对之间的最大差，则此时的特征距离用范数 $p = \infty$ 表示，或这些差的总和用范数 $p = 1$ 表示。在 [??] 中也用了与本文相同的范数作为样本距离的描述。需要注意的是，我们的后文提出鲁棒性验证框架支持任意一种范数形式。

定义 2.6.3 (回归模型的全局鲁棒性). 给定回归模型 R , N 为测试样本集合。如果 N 至少存在 $\rho \cdot |N|$ 个样本，使得 R 满足回归模型单样本鲁棒性，则 R 满足回归模型的全局鲁棒性。

定义 2.6.4 (分类模型的全局鲁棒性). 给定分类模型 C , N 为测试样本集合。如果 N 至少存在 $\rho \cdot |N|$ 个样本，使得 C 满足分类模型的单样本鲁棒性，则 C 满足分类模型的全局鲁棒性。

定义2.6.3和定义2.6.4描述了回归模型和分类模型的全局鲁棒性, 其中参数 ρ 约束了在测试集中保持鲁棒性的样本的比例的大小, 用来反映该模型的全局鲁棒性。

2.7 本章小结

本章主要介绍了本文研究中涉及的基础知识, 从而为后续章节的展开打下基础: (1) 介绍了关于 SAT 和 SMT 技术的基础知识 (2) 树模型相关的基本知识, 包括随机森林和 GBDT 的概念与算法。(3) 对树模型鲁棒性基本定义的描述。

第三章 联邦学习

在本章节中我们提出了一个面向树模型的鲁棒性验证框架，它以可满足性模理论技术为基础。目的是验证树模型的鲁棒性。我们将会在本章节详细的描述该框架中各个模块的设计和实现过程。

3.1 鲁棒性验证框架的整体介绍

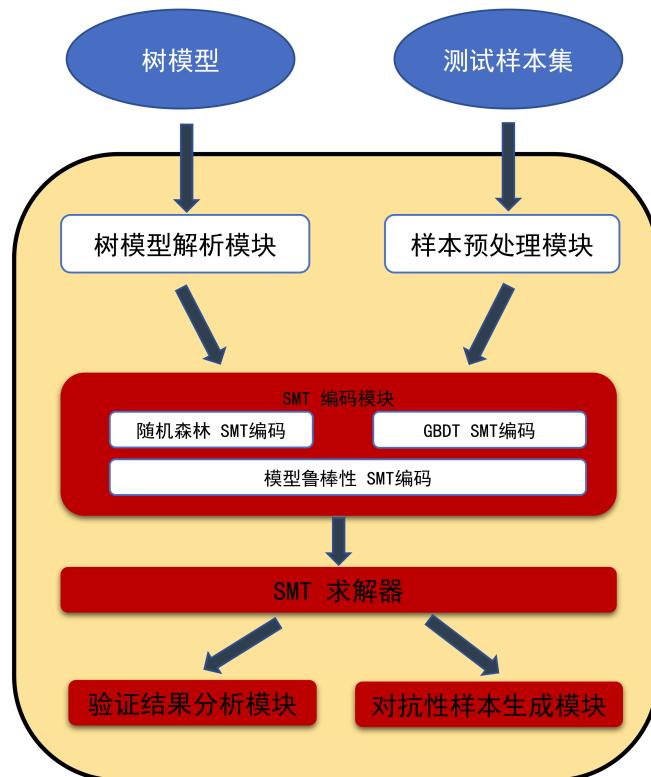


图 3.1: 验证框架图

图3.1展示了树模型鲁棒性验证的框架图。框架的输入包括树模型的源文件和测试样本集。首先，需要对树模型进行结构解析，主要是提取树模型的结点信息。

同时样本预处理模块需要对测试样本集进行数据预处理。接下来的 SMT 编码模块是我们的工作的核心。我们的验证框架需要支持树模型的两个重要实现：随机森林模型和 GBDT 模型。因此，我们实现了这两种模型的编码器。SMT 编码模块主要的任务是把树模型和待验证的样本鲁棒性质编码成 SMT 公式，以便可以利用 SMT 求解器进行可满足性求解。当 SMT 求解器返回求解结果之后，验证结果分析模块会对结果进行统计分析。同时对于鲁棒性不满足的情况，在对抗性样本生成模块中会生成该样本的对抗性样本。我们将在以下小节详细介绍各个模块的内容和实现原理。

3.2 树模型解析模块

树模型解析模块主要是用于解析树模型的内部结构。我们首先需要对待验证的树模型进行结构解析，从而获取到该模型的一些内部信息为之后的 SMT 编码模块做准备。

我们知道，无论是随机森林还是 GBDT 模型都是决策树的集合。所以在解析模块中，我们会对集合中所有的单棵决策树进行解析，并保存成相应的数据结构。一般来说，决策树的结构为二叉树。决策树 T 可以被定义为一个公式 $t : X^d \rightarrow Y^m$ ，对于一个输入样本向量 $x \in X^d$ 且 $x = \langle a_1, \dots, a_d \rangle$ ，决策树 T 可返回一个预测结果 $y \in Y^m$ 。其中 X^d 表示输入样本的特征空间，特征数量为 d ， Y^m 表示输出空间， m 表示预测类别个数。我们可以将决策树定义成一个 3 元组 $T = \langle N, L, V \rangle$ 。

定义 3.2.1 (决策树). 一颗决策树 T 是一个 3 元组 $\langle N, L, V \rangle$ ，其中：

1. $N = \{n_0, n_1, \dots, n_k\}$ 是内部结点的集合，其中 n_0 表示根结点。对于每一个内部结点 $n \in N$ 都会与一个特征阈值表达式 $s_n = (a_i, n_i)$ 相对应，如果 $s_n = a_i \leq n_i$ 为 *True*，则将当前结点的孩子结点左孩子结点 $n_l \in N$ 。相反，如果 s_n 为 *False* 则为右孩子结点 $n_r \in N$ 。
2. $L = \{l_0, \dots, l_j\}$ 是叶子结点的集合。

3. $V = \{v_{l_0}, \dots, v_{l_j}\}$ 是叶子结点的值集合, 其中 $v_{l_0} \in V$ 代表的是叶子结点 $l_0 \in L$ 的叶子结点的值。如果树为分类决策树, 则 $v_l = \{p_i \mid 0 \leq i \leq m, \sum_{i=0}^m p_i = 1\}$, 其中 p_i 代表的是类别 i 的预测概率, 取值不能大于 1; 如果树为回归决策树, 则 $v_l = q_i$, 其中 q_i 为回归值。

总的来说, 树模型解析模块会将树模型集合中的各个子树按照定义3.2.1保存成对应的数据结构, 为 SMT 编码模块提供模型的结构信息。

3.3 样本预处理模块

有些情况下, 我们得到的样本数据集会存在缺失, 格式不统一等问题, 所以在使用之前, 应该进行预处理。样本预处理模块提供了一些基本的数据处理功能, 如缺失值插补, 去重等。我们利用 Pandas, Numpy 等第三方库实现了该模块。

3.4 SMT 编码模块

为了能够使用 SMT 理论来验证树模型的鲁棒性, 我们需要将树模型编码为 SMT 公式。SMT 编码模块是整个验证框架的核心模块。我们将在本节详细阐述随机森林和 GBDT 模型的 SMT 公式的编码过程。

3.4.1 决策树的 SMT 编码

在树模型解析模块, 我们已经将模型集合中的决策树按照定义3.2.1解析完成, 在本小节我们详细描述了如何根据决策树构建其对应布尔逻辑表达式的过程, 为之后整个树模型的 SMT 公式的编码做准备。

首先我们从编码决策树中的一条路径开始, 如果有一棵决策树 $T = \langle N, L, V \rangle$, $l \in L$ 表示其叶子结点, 则叶子结点的决策路径公式 $w(l)$ 如下:

$$\omega(l) : \bigwedge_{n \in N_1} (s_n) \wedge (o = v_l), s_n = \begin{cases} a_i \leq \eta_i & n_c = n_l \\ a_i > \eta_i & n_c = n_r \end{cases} \quad (3.1)$$

其中 N_l 表示从根结点 n_0 到叶子结点 l 的内部结点集合, o 为叶子值 v_l 的约束变量, n_c 表示结点 n 的一个孩子结点。 s_n 为结点 n 所对应的特征阈值条件式。如果 n_c 为左孩子结点, 则 $s_n = (a_i \leq n_i)$ 。相反, 如果 n_c 为右孩子结点, 则 $s_n = (a_i > n_i)$ 。

在公式3.1中, 我们已经将单个叶子结点的决策路径的 SMT 公式编码完成。进一步, 根据决策树 T 的决策原理, 树 T 的 SMT 编码为所有叶子结点的路径公式的析取范式, 即树 T 的 SMT 公式 $\Pi(T)$ 为:

$$\Pi(T) : \bigvee_{l \in L} \omega(l) \quad (3.2)$$

公式 (3.2) 中的每一个子句, 都代表唯一的叶子结点的路径公式。

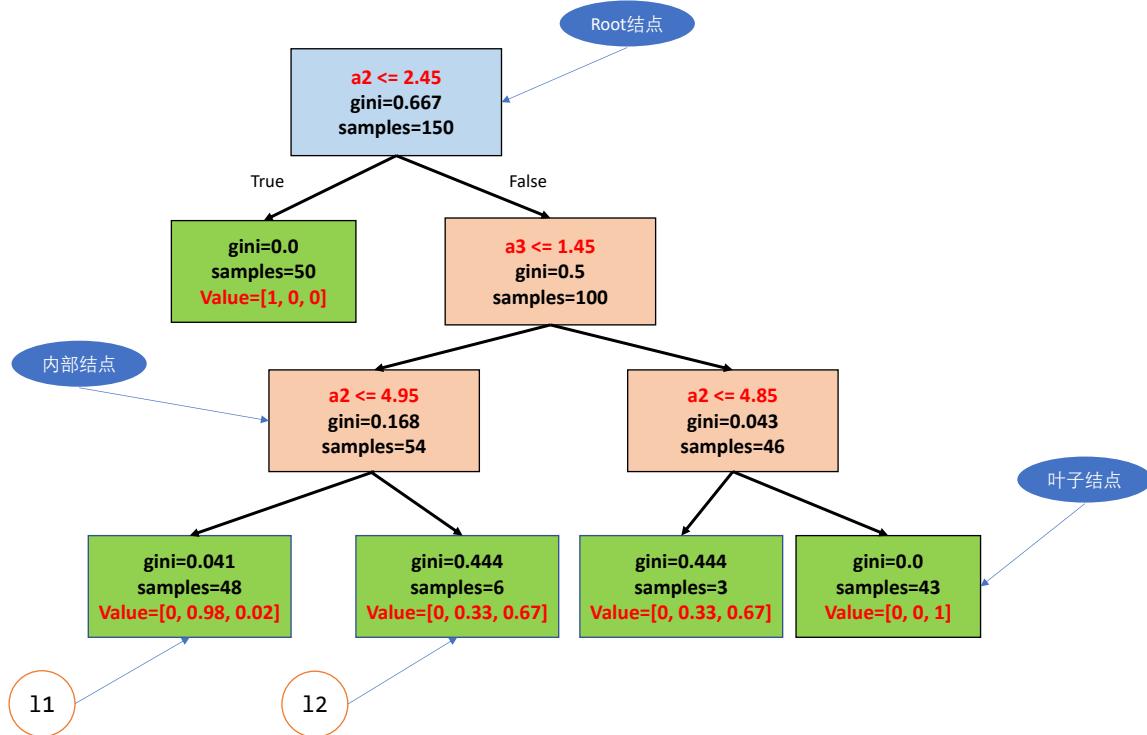


图 3.2: 分类树

例 3.4.1 (分类决策树的 SMT 公式). 图3.2为用于莺尾花分类任务的决策树, 其中叶子结点 l_1 的路径公式 $w_{l_1} = (a_2 \leq 2.45) \wedge (a_3 \leq 1.45) \wedge (a_2 \leq 4.95) \wedge (o = v_{l_1})$, 其中 o 为一个约束变量的集合且 $o = \{o_0 = p_0, o_1 = p_1, o_2 = p_2\}$, v_{l_1} 为预测类别的概率集合且 $v_{l_1} = \{p_0 = 0, p_1 = 0.98, p_2 = 0.02\}$, 其中叶子结点 l_2 的路径公式

$w_{l_2} = (a_2 \leq 2.45) \wedge (a_3 > 1.45) \wedge (a_2 > 4.95) \wedge (o = v_{l_2})$, 且 $o = \{o_0 = p_0, o_1 = p_1, o_2 = p_2\}$, $v_{l_2} = \{p_0 = 0, p_1 = 0.33, p_2 = 0.67\}$ 。相应的该分类树的 SMT 公式为:
 $w_{l_1} \vee w_{l_2} \vee \dots \vee w_{l_j}$ 。

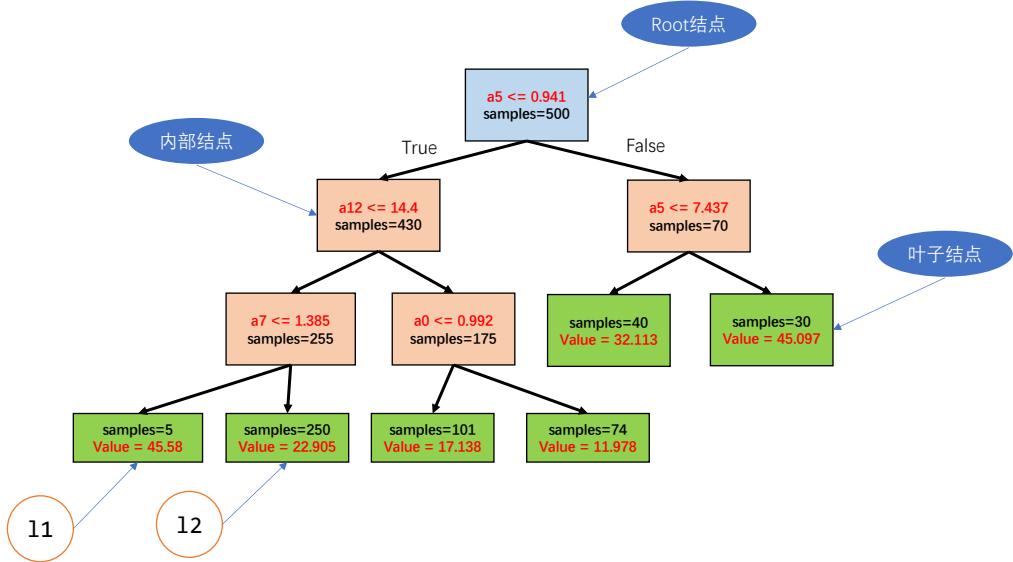


图 3.3: 回归树

例 3.4.2 (回归决策树的 SMT 公式). 图 3.3 为用于预测波士顿房价任务的回归决策树, 其中叶子结点 l_1 的路径公式 $w_{l_1} = (a_5 \leq 0.941) \wedge (a_{12} \leq 14.4) \wedge (a_7 \leq 1.385) \wedge (o = v_{l_1})$, 其中 o 为一个约束变量且 $o = v_{l_1}$, v_{l_1} 为预测的房价回归值且 $v_{l_1} = 45.58$, 叶子结点 l_2 的路径公式 $w_{l_2} = (a_5 \leq 0.941) \wedge (a_{12} \leq 14.4) \wedge (a_7 > 1.385 \wedge (o = v_{l_2}))$, $o = v_{l_2}$ 且 $v_{l_2} = 22.905$ 。相应的该回归树的 SMT 公式为: $w_{l_1} \vee w_{l_2} \vee \dots \vee w_{l_j}$ 。

3.4.2 随机森林模型的 SMT 编码

根据在第二章节关于随机森林基础知识的介绍, 我们已经知道它是决策树的组合, 之后通过投票机制来得出最终的预测结果。在本节中我们会详细描述如何将随机森林模型编码为 SMT 公式。

(1) 随机森林分类模型编码

随机森林分类模型 RFC 是 k 棵分类树的集合, 即 $RFC = \langle T_1, \dots, T_k \rangle$ 。对于一个输入样本 x 来说, 其预测类别为在模型 RFC 中所有决策树给出的投票结果。

在本文中，我们考虑的软投票的情况。因为有研究表明，基于类概率的投票往往性能会更好。我们知道每棵分类树的输出都是一组对应于每个类的概率，然后分类模型 RFC 会将在所有分类树中概率估计平均值最高的预测类别作为预测结果输出。此过程可表示为如下公式：

$$RFC(x) = \operatorname{argmax}_j \frac{1}{k} \sum_{i=1}^k t_i^j(x) \quad (3.3)$$

其中 $t_i(x)$ 表示分类树 T_i 对 x 的预测类别， $t_i^j(x)$ 表示树 T_i 对类别 j 的预测概率，且 $\sum_{i=1}^m t_i^j(x) = 1$ ， m 表示预测类别的个数。根据随机森林的定义，随机森林的编码是其集合中所有分类树的 SMT 公式的合取公式。即：

$$RFC : \bigwedge_{i=1}^k \Pi(T_i) \wedge \left(\text{out} = \operatorname{argmax} \frac{1}{k} \sum_{i=1}^k o_i^j \right) \quad (3.4)$$

其中变量 o_i 用来约束树 T_i 的输出 $t_i(x)$ ，相应的 o_i^j 用来约束预测类别 j 的概率 $t_i^j(x)$ ， out 为模型 RFC 的输出约束变量。

(2) 随机森林回归模型编码

随机森林回归模型 RFR 是 k 棵回归树的集合，即 $RFR = \langle T_1, \dots, T_k \rangle$ 。对于一个输入样本 x 来说，其预测结果为在模型 RFR 中所有回归树给出的平均值。我们知道每棵回归树的输出都是一个预测回归值，回归模型 RFR 会将在所有分类树中预测结果的平均值作为预测结果输出。此过程可表示为如下公式：

$$RFR(x) = \frac{1}{k} \sum_{i=1}^k t_i(x) \quad (3.5)$$

其中 $t_i(x)$ 表示分类树 T_i 对 x 的预测值。随机森林的编码是其集合中所有分类树的 SMT 的合取公式。即：

$$RFR : \bigwedge_{i=1}^k \Pi(T_i) \wedge \left(\text{out} = \frac{1}{k} \sum_{i=1}^k o_i \right) \quad (3.6)$$

其中变量 o_i 用来约束树 T_i 的输出 $t_i(x)$ ， out 为模型 RFR 的输出约束变量。

3.4.3 GBDT 的 SMT 编码

GBDT 模型的 SMT 公式的编码过程与随机森林模型类似。但对于回归任务和分类任务来说，GBDT 都是以回归树作为基学习器去完成预测任务，所以最终构建的 SMT 公式也会有所区别。

(1) GBDT 分类模型的 SMT 编码

对于分类任务来说，GBDT 模型与随机森林模型不同，GBDT 是基于回归树去完成分类预测。对于每个预测类别都会训练对应数目的回归树。例如：GBDT 模型用于一个三分类的任务，指定其基学习器的数目为 5，则每个类别都需要训练 5 棵树，总计需要训练 15 棵树来完成预测任务。具体来说，GBDT 分类模型 GBC 是由多个回归树组成的集合，即 $GRC = \langle R_1, \dots, R_c \rangle$ 且 $R_j = \langle T_1^j, \dots, T_r^j \rangle$ ，其中 c 表示的是预测类别的个数， r 表示回归树的棵树， R_j 表示类别 j 的回归树集合。对于一个输入样本 x 来说，分类模型 GBC 的预测结果为概率最大的类别，即 $GBC(x) = \arg \max_j (R_j(x))$ 。则该 SMT 公式可定义为：

$$GBC : \bigvee_{j=1}^c \left(\arg = j \leftrightarrow \bigwedge_{k=1}^c \text{out}_j > \text{out}_k \right) \quad (3.7)$$

其中变量 o_i 用来约束树 T_i 的输出 $t_i(x)$, out 为模型 GRC 的输出约束变量。

(2) GBDT 回归模型的 SMT 编码

GBDT 的回归模型 GBR 是由多个回归树组成的集合，即 $GBR = \langle T_1, \dots, T_r \rangle$ 。对于一个输入样本 x 来说，回归模型 GBR 的预测结果为集合中回归树预测值的总和，即 $GBR(x) = \sum_{i=1}^r T_i(x)$ 。则该 SMT 公式可定义为：

$$GBR : \left(\bigwedge_{i=1}^r \Pi(T_i) \right) \wedge \left(\text{out} = \sum_{i=1}^r o_i \right) \quad (3.8)$$

其中变量 o_i 用来约束树 T_i 的输出 $t_i(x)$, out 为模型 GBR 的输出约束变量。

3.4.4 模型鲁棒性 SMT 编码

在之前的小节中，我们已将树模型编码为 SMT 公式，接下来，我们会基于测试样本去构造该模型鲁棒性的布尔公式。

(1) 单样本的鲁棒性的编码

定义 3.4.3 (特征扰动约束公式). 若输入样本 $x = \langle a_1, \dots, a_d \rangle$, 最大扰动为 ϵ , 若其对应的对抗性样本为 x' , 则扰动约束公式为 $\Delta(x, x', \epsilon)$:

$$\Delta(x, x', \epsilon) : \bigwedge_{i=1}^d |a_i - a'_i| \leq \epsilon \quad (a_i \in x, a'_i \in x') \quad (3.9)$$

定义 3.4.4 (回归模型单样本鲁棒性公式). 根据定义 2.6.1, 若模型 R 为回归模型, 输入样本 $x = \langle a_1, \dots, a_d \rangle$, 最大扰动为 ϵ , 对抗性样本为 x' , 则回归模型的单样本鲁棒性公式 Φ_R 可定义为:

$$\Phi_R : R(x') \wedge \Delta(x, x', \epsilon) \wedge (|R(x) - \text{out}| \geq \delta) \quad (3.10)$$

在公式 (3.10) 中的, 回归模型 R 代表在随机森林的分类模型 RFR 的 GBDT 的回归模型 GBR。至此, 我们可以将回归模型的鲁棒性验证问题转换为查找是否存在使得公式 Φ_R 的可满足的赋值 x' 的问题。如果该赋值 x' 存在, 则说明该模型在样本 x 受到扰动的情况下 (对抗性样本为 x'), 会预测错误, 即不满足鲁棒性; 相反, 如果赋值 x' 不存在, 则说明该模型满足鲁棒性。我们将在下边给出形式化证明。

定义 3.4.5 (分类模型单样本鲁棒性公式). 根据定义 2.6.2, 若模型 C 为分类模型, 输入样本 $x = \langle a_1, \dots, a_d \rangle$, 最大扰动为 ϵ , 对抗性样本为 x' , 则分类模型的单样本鲁棒性公式 Φ_C 可定义为:

$$\Phi_C : C(x') \wedge \Delta(x, x', \epsilon) \wedge (\text{out} \neq C(x)) \quad (3.11)$$

在公式 (3.11) 中的, 分类模型 C 代表在随机森林的分类模型 RFC 的 GBDT 的分类模型 GBC。同样的, 我们可以将分类模型的鲁棒性验证问题转换为查找是否存在使得公式 Φ_C 的可满足的赋值 x' 的问题。如果该赋值 x' 存在, 该分类模型 C 不满足鲁棒性; 相反, 如果赋值 x' 不存在, 则说明该模型满足鲁棒性。

定理 3.4.6. 给定一个模型 $C = \langle T_1, \dots, T_k \rangle$, 输入样本 $x \in X^d$, 其对应的鲁棒性公式为 Φ_C 。如果 Φ_C 是可满足的, 则其真赋值 $x' \in X^d$ 是该模型的对抗性样本, 该模型 C 不满足鲁棒性; 反之, 如果 Φ_C 不满足的, 则模型 Φ_C 相对于 x 满足鲁棒性。

证明. 我们以随机森林分类模型为例, 给出如下的证明。

假设 Φ_C 是可满足的, 则存在真赋值 x' , 并且 $O = \{o_i | 1 \leq i \leq k\}$ 。根据公式 (3.4) 和公式 (3.9), 我们可以得出: ($out \neq RFC(x)$) 成立, 并且在模型 RFC 中的每个决策树公式 $\Pi(T_i)$ 都成立。在不失一般性的前提下, 让我们考虑树 T_1 的公式 $\Pi(T_i)$ 为 True 的情况, 则必然存在一个真赋值 $o_1 \subset O$ 。我们首先需要证明树 T_1 对于 x' 的预测结果就是真赋值 o_1 , 即 $t_1(x') = o_1$ 。根据公式 (3.2), 我们可以得出至少有一个叶子路径公式的赋值为 True。参考决策树的定义, 除了根结点 n_0 之外, 每个内部结点只有一个前驱结点, 因此我们可以确保只有一个叶子路径公式的赋值为 True。假设叶子公式为 $w(l_0)$, 叶子 l_0 的值为 v_{l_0} 。然后, 考虑在树 T_1 预测输入样本 x' 的决策过程。当 x' 到达内部结点 n 时, 如果特征 $a' \in x'$ 满足特征阈值公式 $(s_n = (a' \leq \eta)) \in w(l_0)$, 则 x' 将被传递到给左孩子结点, 如果不满足 s_n 的话, 将会被传到右孩子结点。从根结点递归应用该决策规则, 最终 x' 将会落到在叶子 l_0 , 最终的预测结果就是该叶子的值, 即 $t_1(x') = v_{l_0}$ 。根据公式 (3.4), 则 $o_1 = v_{l_0}$, 那么我们可以得出 $t_1(x') = o_1$ 的结论。类似地, 我们可以得出 $t_2(x') = o_1, \dots, t_k(x') = o_k$ 。

为了便于描述, 我们可以将公式 (3.3) 简化为 $RFC(x') = argmax(T(x'))$ 。通过公式 (3.4), 我们得出 $out = argmax(O)$, 则 $RFC(x') = out$ 。我们已经知道 ($out \neq RFC(x)$) 为 True, 因此赋值 x' 是模型 RFC 的对抗性样本。类似地, 如果 Φ_C 不满足的话, 则表明不存在这样的对抗性样本 x' , 即该模型相对于 x 满足鲁棒性。即证。 \square

(2) 全局鲁棒性的编码

定义 3.4.7 (回归模型全局鲁棒性公式). 根据定义 2.6.3, 设 R 为回归模型, N 为测试样本集合。如果 N 至少存在 $\rho \cdot |N|$ 个样本, 使得 R 满足回归模型单样本鲁棒性,

则 R 满足回归模型的全局鲁棒性。则回归模型的全局鲁棒性公式 Φ_{GR} 可定义为：

$$\bigwedge_{i=1}^{|N|} (\Phi_i \Leftrightarrow q_i) \wedge \sum_{i=1}^{|N|} q_i \geq \rho \cdot |N| \quad (3.12)$$

在公式 (3.12) 中的，回归模型 R 代表在随机森林的分类模型 RFR 的 GBDT 的回归模型 GBR。

定义 3.4.8 (分类模型全局鲁棒性公式). 根据定义 2.6.4, 设 C 为分类模型, N 为测试样本集合。如果 N 至少存在 $\rho \cdot |N|$ 个样本, 使得 C 满足分类模型的单样本鲁棒性, 则 C 满足分类模型的全局鲁棒性。则分类模型的全局鲁棒性公式 Φ_{GC} 可定义为：

$$\bigwedge_{i=1}^{|N|} (\Phi_i \Leftrightarrow q_i) \wedge \sum_{i=1}^{|N|} q_i \geq \rho \cdot |N| \quad (3.13)$$

在公式 (3.13) 中的，分类模型 C 代表在随机森林的分类模型 RFC 的 GBDT 的分类模型 GBC。

3.5 SMT 求解器

随着对于 SMT 研究和技术的进步，现在已经有了一些功能强大和复杂的 SMT 求解器，例如 Alt-Ergo [?]、Beaver [?], Boolector[?]、MathSAT5[?]、openSMT[?]、SMTInterpol[?]、SONOLAR[?]、STP[?]、veriT[?]、Yices[?]、Z3[?] 等可以快速扩展的应用程序集。目前的应用领域包括处理器验证，等价性检查，有界和无界模型检查，谓词抽象，静态分析，自动测试用例生成，类型检查、计划、调度和优化等。在很多神经网络的验证工具 [? ? ? ?] 中，选择 Z3 求解器作为其底层的求解器。在本文的鲁棒性验证框架中，我们同样将 Z3 求解器作为 SMT 求解器，但需要注意的是，框架也支持其他主流的 SMT 求解器。

Z3 有着高效而且全面的的求解能力，它是开源的，并且支持当前所有的 SMT 理论，并且提供了多种 API 接口。在本框架中，因为我们框架的实现是基于 Python 语言的，所以调用其 Python API。我们会将 SMT 编码模块生成的公式转化为 Z3 求解器可接受的公式形式，之后利用 Z3 求解器去对该公式进行求解。

根据对 SMT 公式的求解结果，Z3 会返回“UNSAT”和“SAT”两种结果。“UNSAT”表示无法找到该公式的可行解，根据我们给出的模型鲁棒性的定义，此结果表示该模型满足鲁棒性。“SAT”结果存在该公式的可行解，则表明该模型不满足鲁棒性，返回的可行解即为该模型的一个对抗性样本。

3.6 验证结果分析模块

验证结果分析模块的基本功能是统计和分析验证结果，如待验证树模型的鲁棒性值，满足鲁棒性的样本个数和所占的百分比，验证失败或验证超时的样本个数或所占的百分比等。除此之外，在下一章节关于树模型可解释研究中提出的算法也在该模块实现。

3.7 对抗性样本生成模块

对抗性样本生成模块是为了生成树模型对抗性样本。当模型的鲁棒性验证结果为不满足的情况时（即求解器返回的结果为“SAT”），此时求解器还会返回一个可行解 x' ， x' 为该模型的一个对抗性样本。但 x' 并不能直接作为对抗性样本输入到模型中，必须要按照原始样本的数据格式进行二次处理才能成为模型可接受的输入形式。该模块提供了将求解器生成的反例转换为树模型可接受的输入样本格式即对抗性样本的功能。

3.8 本章小结

在本章节中，我们介绍了树模型鲁棒性验证框架的设计与实现部分。我们详细描述了该框架的各个模块的设计和实现的细节，包括了树模型解析模块，样本预处理模块，SMT 编码模块等。我们的核心工作是对树模型及其鲁棒性的 SMT 公式编码的设计，展示了从决策树到整体树模型的编码过程，对底层所用的 Z3 求解器也做了介绍。

第四章 联邦学习的自适应加噪机制

最近研究表明深度神经网络容易受到对抗样本的攻击。为了解决这个问题，一些工作通过向图像中添加高斯噪声来训练网络，从而提高网络防御对抗样本的能力，但是该方法在添加噪声时并没有考虑到神经网络对图像中不同区域的敏感性是不同的。针对这一问题，提出了梯度指导噪声添加的对抗训练算法。该算法在训练网络时，根据图像中不同区域的敏感性向其添加自适应的噪声，在敏感性较大的区域上添加较大的噪声，抑制网络对图像变化的敏感程度，在敏感性较小的区域上添加较小的噪声，提高其分类精度。

4.1 模型概览

如图一所示，在我们的系统模型中，有两方，即云服务器和用户。云服务器事先与用户协商一个网络框架。然后，服务器通过公共数据训练一个初始模型，然后将初始模型的参数广播给用户。用户在本地训练各自的模型后，云服务器收集用户发送的模型梯度，并更新全球模型。用户下载由云服务器初始化的模型参数。然后，每个用户在本地数据集上训练私人模型。最后，用户将本地模型的扰动梯度发送到云服务器。

4.1.1 威胁模型

我们认为云服务器是一个“诚实但好奇”的实体。也就是说，服务器将遵循与所有用户的协议。然而，通过利用完全访问用户梯度的便利，它也试图在训练过程中获得额外的信息。出于这个原因，我们的 ANFL 的目标是保护发送到服务器的本地梯度不被推断出任何关于用户的额外信息。

我们用层间相关性传播 (LRP) 算法将输出分解到每一层。关于 LRP 算法的更多细节，我们将在以下部分进行介绍。每个用户都在本地对原始数据进行训练前馈操作，这可以获得一个新的数据操作，从而获得本地模型的输出。根据相邻层之间的线性关系，在 $k - th$ 层的神经元的贡献 $C_{a_i}^{l_k}(x_i)$ 等于连接到神经元 a_i 的相邻层的贡献之和：

$$C_{a_i}^{l_k}(x_i) = \sum_{a_j \in l_{k+1}} C_{a_i \leftarrow a_j}^{l_k \leftarrow l_{k+1}}(x_i)$$

例如，如图 2 所示，我们有：

$$C_{a_7}^{l_2}(x_i) = \sum_{a_j \in l_3} C_{a_7 \leftarrow a_j}^{l_2 \leftarrow l_3}(x_i) = C_{a_7 \leftarrow a_8}^{l_2 \leftarrow l_3}(x_i) + C_{a_7 \leftarrow a_9}^{l_2 \leftarrow l_3}(x_i)$$

其中，“ \leftarrow ”表示两部分之间的连接关系。 l_{23} ”是指深度神经网络 (DNNs) 中 $2 - th$ 层和第 3 层之间相邻层的连接关系。当 $k - th$ 层为输出层时，我们有：

$$C_{a_i}^{l_k}(x_i) = f(x_i, \omega_i^r)$$

随着机器学习应用领域的不断拓展，与机器学习模型的验证一样，模型的可解释性也变得愈发的重要。从用户的角度出发，机器学习模型不仅需要向用户反馈正确的预测结果，还需要向用户解释预测的原因。这样可以增加用户对模型的信任，让用户可以更放心的使用该模型；从开发人员的角度来说，可解释性对模型训练具有重要的意义。例如，根据解释信息可以帮助开发者确定更优的模型训练参数。在预测结果出现误差的时候，也可以了解导致误差产生的模型内部的原因，从而帮助开发人员改进机器学习模型的缺陷。总的来说，模型的可解释性已经成为机器模型应用的必备条件。

目前已经有很多研究着眼于机器学习模型的可解释性问题。[?] 的作者提出了一种基于自动推理的方法，可以从机器学习模型中提取有价值的信息，使用户可以了解树模型决策背后的原因。一些研究者 [?] 则试图用更简单的模型来近似复杂模型来提供更好的解释。局部解释的方法 [?] 了解的是模型的输出如何在局部的

输入扰动上的分布变化的问题，它可以根据结果输出值推断出输入参数的重要性。此外，还有一些研究提出了基于实例的解释方法 [? ?]，即通过选择数据集的特定实例来解释机器学习模型的行为或解释底层的数据分布。相比于其他类型的机器学习模型来说，树模型以决策树的基础，对于一个输入样本来说，我们可以获得预测的决策路径，根据决策路径可以获取一些决策信息，因此树模型通常被认为是一种易于解释模型。但很少有人研究树模型鲁棒性的可解释性问题。

在本小节中，我们主要关注的是模型的样本特征与模型鲁棒性之间的关系。首先，我们提出了鲁棒特征集合（Robust Feature Set, RFS）的概念，鲁棒特征集合可用于解释单个样本的鲁棒性。在鲁棒特征集合的基础上，我们进一步提出了一种计算局部鲁棒特征重要度（Local Robustness Feature Importance, LRFI）的算法，局部鲁棒特征重要度可用于解释树模型的样本特征与预测类别的鲁棒性的关系。

4.2 鲁棒特征集合

目前对于树模型的鲁棒性验证的研究中，在鲁棒性验证失败情况下，验证器通常将返回对抗性样本，但是当验证成功的情况下，通常都不会给出任何解释。针对这种情况，我们想进一步去探索为什么有些样本在特征被扰动的情况下，仍然能被识别正确？换句话说，我们是否可以确定哪些特征会真正的影响该样本的鲁棒性？不同的样本特征对模型的鲁棒性的影响大小是否也是不同的？

类似于 Z3 的 SMT 求解器在判断 SMT 公式不满足的情况下，会产生该公式集合的最小不满足核（Minimal Unsatisfiable Core, MUC），MUC 是原始公式集合的子集。我们首先给出 MUC 的定义。

定义 4.2.1（最小不满足核）。如果 F 是一个 CNF 公式， F_C 代表 F 的公式集合。如果 $S \subseteq F_C$ 同时符合以下条件，则 S 是 F 的最小不满足核：

1. F 是不可满足的。
2. S 是不可满足的。

3. 不存在任何 $S'' \subseteq S$ 是不可满足的。

定义 4.2.2(鲁棒特征集合). 给定一个树模型的分类模型 C , 样本 $x = \langle a_1, a_2, \dots, a_d \rangle$, 最大扰动距离为 ϵ , Φ 是模型的鲁棒性公式, Φ_C 表示 Φ 中的公式集合, $\Delta(x, x', \epsilon) \subset \Phi$ 是特征扰动约束公式并且 Δ 表示 $\Delta(x, x', \epsilon)$ 中的公式集合, 则鲁棒特征集合定义如下:

1. Φ 是不可满足的, $S \subseteq \Phi_C$ 是 Φ 的最小不满足核。
2. $RFS = \{a_i \mid a_i \text{ 是出现在公式集合 } \Delta_s \text{ 中的特征}, \Delta_s \subseteq \Delta \text{ 是 } S \text{ 的子集}\}$

定理 4.2.3. 给定一个树模型 C , 样本 $x = \langle a_1, a_2, \dots, a_d \rangle$, 最大扰动距离为 ϵ 。保持存在于鲁棒特征集合中的特征的值不变的情况下, 任意扰动其他特征的值, 都不会改变该模型 C 对 x 的预测结果。

证明. 根据公式 3.11, 我们知道 out 是由 $C(x')$ 和 $\Delta(x, x', \epsilon)$ 所决定的。在不失一般性的前提下, 我们可以将公式 Φ 转换成 $\Phi' := C(x') \wedge \Delta(x, x', \epsilon) \Rightarrow (out \neq C(x))$ 的形式表示。而公式 $C(x')$ 的构建依赖于模型 C 的结构, 所以当模型 C 给定的情况下, 公式 $C(x')$ 也是确定的。在这种情况下, 公式 Φ 的可满足性就仅仅与 $\Delta(x, x', \epsilon)$ 相关。所以在此定理的证明中, 我们不需要考虑 $C(x')$ 的可满足性。

我们用 Φ_c 表示 Φ' 的公式集合。则 Φ_c 可以被定义为: $\Phi_c = R_C \cup \Delta \Rightarrow \{o\}$ 。
 R_C 表示 $C(x')$ 的公式集合, Δ 表示 $\Delta(x, x', \epsilon)$ 的公式集合, 其中的 $\Delta = \{\delta_i \mid 0 \leq i \leq d, \delta_i = |a_i - a'_i| \leq \epsilon\}$, $o = (out \neq C(x))$ 。假设 Φ_c 是不可满足的, 并且 $S \subseteq \Phi_c$ 表示最小不满足核。首先, 我们可以得出 $\Phi_c \setminus \{o\}$ 是可满足的。因为至少有一个真赋值 $x' = x$ 使其满足。所以 o 必然存在于 S 中, 可表示为 $o \in S$ 。我们用 $R_s \subseteq R_C$ 和 $\Delta_s \subseteq \Delta$ 表示存在于 S 中的公式子集, 则我们可以得到 $S = R_s \cup \Delta_s \cup \{o\}$ 并且有 $\Phi_c \setminus S = (\Delta \setminus \Delta_s) \cup (R \setminus R_s)$.

根据定义 4.2.1 和定义 4.2.2, 我们知道鲁棒特征集合中的特征是出现在 Δ_s 中的特征。并且每个特征 $a_s \in RFS$ 都对应着 Δ_s 中的一个子句。类似的, 每个特征 $a_o \in X^d \setminus RFS$ 也对应着 $\Delta \setminus \Delta_s$ 中的一个子句。公式 (3.7) 表示样本 x 中特征的扰动

约束公式, 如果 $\delta \in \Delta$ 为 True, 则特征 a 的扰动距离必然不可能超过 ϵ 。反之, 如果 δ 为 False, 则扰动距离必然超过了 ϵ 。所以该子句的真假其实代表的是该特征的扰动距离。根据最小不满足核的性质, 我们可以得到每个特征 $a_o \in X^d \setminus RFS$ 对应的子句都不会影响 Φ 的可满足性。因为 $S = R_s \cup \Delta_s \cup \{o\}$ 是不可满足的, 我们可以得出每个 $\delta_s \in \Delta_s$ 都是可满足的, 也就是说, 其中的每个特征的扰动距离都没有超过 ϵ 。在此证明中我们只考虑每个特征的值保持不变的情况, 所以每个特征对应的子句 $\delta_s = |a_s - a'_s| = 0$ 都为 True。因为 $\Phi_c = R_C \cup \Delta \Rightarrow \{o\}$ 是不可满足的, 所以 $\Phi'_c = R_C \cup \Delta \Rightarrow \neg o$ 是有效的, 即 $\neg o = (out = C(x))$, 也就是说该模型对 x 的预测结果保持不变。即证。 \square

根据第三章的树模型的鲁棒性验证框架的介绍, 我们可知当验证样本 x 的鲁棒性的时候, SMT 编码模块会将其编码成对应的 SMT 公式 Φ , 之后利用 SMT 求解器判断 Φ 的可满足性。如果求解器返回的结果为 UNSAT, 则说明 Φ 是不可满足的。同时, 求解器会返回 Φ 的最小不满足核。根据定义4.2.2, 我们可以得到样本 x 在树模型 C 上的鲁棒特征集合。需要注意的是, 获取鲁棒性特征集合的前提条件是模型基于样本 x 是满足鲁棒性的。

根据定理4.2.3, 我们可以得出以下结论: 在保持存在于鲁棒特征集合中的特征的值不变的情况下, 任意改变其他特征的值, 都不会影响树模型对样本 x 的预测结果。换句话说, 在最大扰动距离为 ϵ 的情况下, 相较于其他特征来说, 鲁棒特征集合中的特征对鲁棒性有着更大的影响。

4.3 局部鲁棒特征重要度

在上一小节中, 我们提出了鲁棒特征集合的概念。为了进一步了解样本特征与模型预测类别鲁棒性的关系, 我们提出了局部鲁棒特征重要度 (Local Robustness Feature Importance, LRFI) 来描述这种关系。

在算法1中, 输入为一个树模型 C , N 表示的是标记类别为 y 的测试样本集合其大小为 $|N|$, 最大扰动距离 ϵ 和特征集合 X^d , 其输出为类别 y 的局部鲁邦特征重

Algorithm 1 局部鲁棒特征重要度算法

```

1: Input: 树模型  $C$ , 测试样本集合  $N = \{x_i | 0 \leq i \leq |N|, C(x_i) = y\}$ , 最大扰动距离  $\epsilon$ , 特征集合
    $X^d = \{a_i | 0 \leq i \leq d\}$ .
2: Output: 预测类别为  $y$  的局部鲁棒特征重要性  $LRFI$ 
3: 过程: 函数 LocalRobustFeatureImportance( $C, N, \epsilon, X^d$ )
4: // 初始化集合  $S$  和  $V$  为空集
5:  $S \leftarrow \emptyset$ 
6:  $V \leftarrow \emptyset$ 
7: for  $x \in N$  do
8:    $\Phi_x \leftarrow R(x') \wedge \Delta(x, x', \epsilon) \wedge (out = y)$ 
9:    $UNSAT \leftarrow SMT_{solver}(\Phi_x)$ 
10:   $RFS_x \leftarrow$  根据定义4.2.2得到  $x$  的鲁棒特征集合
11:  将  $RFS_x$  加入到集合  $S$  中
12: end for
13: for  $a \in X^d$  do
14:   //  $n_a$  表示特征  $a$  在集合  $S$  中的出现次数
15:    $n_a \leftarrow 0$ 
16:   for  $RFS_x \in S$  do
17:     if  $a \in RFS_x$  then
18:        $n_a \leftarrow n_a + 1$ 
19:     end if
20:   end for
21:   将  $(a, n_a)$  加入到集合  $V$  中
22: end for
23: // 计算特征出现的最多次数与最少次数值
24:  $min_n = MIN(V)$ 
25:  $max_n = MAX(V)$ 
26: for  $(a, n_a) \in V$  do
27:    $n'_a \leftarrow (n_a - min_n) / (max_n - min_n)$ 
28:   将  $(a, n'_a)$  加入到  $LRFI$  中
29: end for
30: return  $LRFI$ 

```

要度 $LRFI$ 。在第 5 行和第 6 行, 初始化中间变量 S 和 V 为空集。集合 S 用来保存所有测试样本的鲁棒特征集合, V 用来保存所有特征在鲁棒特征集合中出现的次数。在第 7 行至第 12 行, 首先构建 N 中每个样本 x 的单样本鲁棒性公式 Φ_x , 之后利用 SMT 求解器对 Φ_x 进行可满足性判断。如果求解器返回结果为 $UNSAT$, 则根据定义4.2.2计算出样本 x 的鲁棒特征集合存入到 RFS_x 中, 最后将 RFS_x 存入到集合 S

中。在第 13 行至第 22 行，计算每个特征在集合 S 中的出现次数。若特征 a 存在于样本的 x 的 RFS_x 中，则其出现次数 n_a 加 1，所以 n_a 的取值范围为 $0 \leq n_a \leq |N|$ ，并将 (a, n_a) 存入集合 V 中。在第 23 行至第 29 行，对 V 中的值进行数据归一化处理。在此算法中，我们利用的 min-max 标准化（Min-max normalization）操作。最后，将经过标准化处理后的值作为类别 y 的局部鲁棒特征重要度返回。直观的来说，某个特征对鲁棒性的重要度是以该特征在测试样本集合的鲁棒特征集合中出现的频率来确定的，该特征出现的频率越高，说明对鲁棒性的影响就越大，其重要度值也就越高，反之，影响就越小，重要度值就越低。

4.4 本章小结

本章节主要讨论了树模型鲁棒性与样本特征的关系，我们提出了鲁棒特征集合和局部鲁棒特征重要度的概念，并且给出了相应的形式化定义和证明。我们将在下一章节的实验中，进一步证明我们结论。

第五章 实验与评估

之前的章节中，我们描述了树模型鲁棒性验证框架的设计和实现过程。在本节的内容中，我们选取了一些基准的数据集在该验证框架上进行实验评估。

5.1 基准数据集介绍

我们选用了以下三个数据集评估了我们的树模型鲁棒性验证框架：

- (1) 波士顿房价数据集 (Boston House Price Dataset) 收集了在 20 世纪 70 年代中期位于波士顿郊区的房屋价格的中位数，它是用于回归任务的经典数据集。该数据集有 506 个样本数据，每个样本数据包含了城镇人均犯罪率，高速公路便利指数，住宅的房间数等 13 个特征及其房屋价格的中位数。
- (2) 手写体数字识别数据集 (MNIST) 是用于分类任务的经典数据集，来源于美国国家标准与技术研究所。总共包含了 70000 个手写数字图像，每个图像的尺寸为 28×28 像素，每个像素点用灰度值表示，灰度值范围为 0 到 255，图像分为 10 类别，分别代表 0-9。
- (3) FASHION-MNIST 数据集包含了 70000 个不同商品的正面灰度图像，与 MNIST 数据集一样，每个图像的尺寸为 28×28 像素，灰度值范围同样为 0 到 255。所有的图像分为 10 种类别，如：T 恤，牛仔裤，裙子等。虽然数据集格式与 MNIST 相同，但由于图像内容的差别，使得有些模型或者算法在 MNIST 和 FASHION-MNIST 的表现会有很大不同。因此对于分类任务，我们在这两个数据集上都进行了实验作为对比。

5.2 实验环境与配置

本文中的所有的实验均在一台装有 64 位 Ubuntu 操作系统的主机上进行，所使用机器的 CPU 型号为 Intel Core i7-5960X，主频为 4.00GHz，运行内存大小为 32GB 和 1T 存储硬盘大小。我们利用 sklearn(scikit-learn) 来训练实验中所需要的树模型：随机森林模型和 GBDT 模型。sklearn 是一种开源的，基于 Python 编程语言的机器学习框架。需要注意的是，本文提出的树模型鲁棒性验证框架，同样适用于其他机器学习框架下树模型的验证（如：Silas[?]，H2O，Ranger[?] 等）。在对样本数据预处理的部分，我们使用了 Pandas，Numpy 等第三方库。

5.3 实验结果与分析

5.3.1 随机森林模型的鲁棒性验证与分析

回归模型的验证

我们在波士顿房价数据集上展开了对随机森林回归模型的实验。在训练阶段，将数据集随机打乱，按照 4:1 的比例划分训练样本集和测试样本集。随后利用 sklearn 训练出拥有不同超参数的随机森林回归模型，如：模型学习率为 {0.1, 0.2, 0.3}，模型中树的深度为 {5, 8, 10}，模型中树的棵树为 {5, 8, 10}。训练出来的模型的准确率都在 93% 至 98% 之间。我们直接选取测试样本集中的样本来进行鲁棒性的验证。按照回归模型的单样本鲁棒性的定义，我们在此数据集下，对所有数据类型为数值型的特征，我们设置其对应的 ϵ 的值为 3, ρ 值为 5 代表 5000 美元，即在扰动房屋相关特征的情况下，模型对房屋价格的预测结果误差不能超过 5000 美元。

折线图 5.1 为模型学习率为 0.3 下随机森林回归模型的鲁棒性验证结果。从图中我们可以看出：随着树的棵树的增加，模型的全局鲁棒性在降低而且树的深度越小，模型的鲁棒性越高。

分类模型的验证

对于随机森林的分类模型来说，我们分别在 MNIST 和 FASHION-MNIST 两个

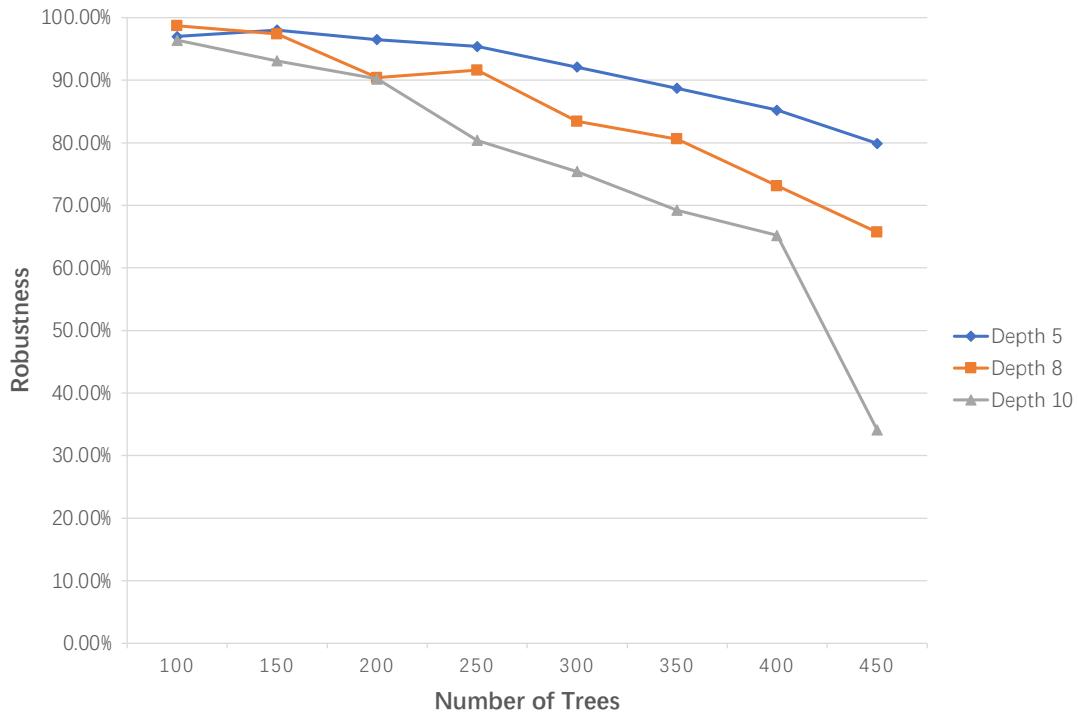


图 5.1: 随机森林回归模型验证结果

数据集上进行了实验验证。对于每个数据集，首先将数据集随机打乱，将其划分为两个子集：80% 训练样本集和 20% 测试样本集。然后，我们从测试样本集中随机抽取了 10 个类别的各 100 个图像，即每个鲁棒性测试样本集的大小为 1000。随后利用 sklearn 训练出随机森林的分类模型，用于验证。

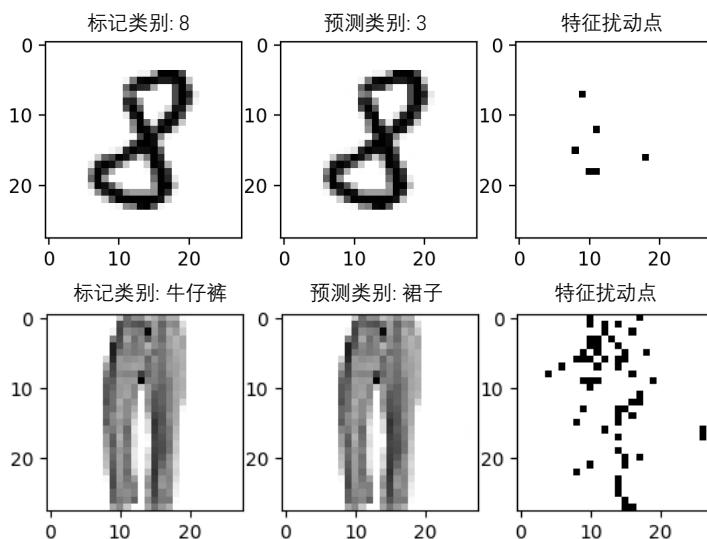


图 5.2: 对抗性样本图

图5.2展示了两个数据集中不满足单样本鲁棒性的测试样本的例子。根据分类模型单样本鲁棒性的定义，我们设置特征扰动范围值 $\epsilon = 1$ ，代表了一个灰度值。图中的第一列图像为原始的样本，第二列为第一列相对应的对抗性样本图像，我们在第三列的图中标记出了受到扰动的特征点。第一个示例来源于 MNIST 数据集，我们可以看出在受到扰动之后，数字“8”被模型错误的预测为数字“3”。第二个示例来源于 FASHION-MNIST 数据集，标记类别为“牛仔裤”的商品图像被错误地分类为“裙子”。在以上示例中，如果我们直接去对比第一列的原始图像和第二列的对抗性样本图像，凭借我们的肉眼，根本无法去找出这两个图像直接的差别（在此结果中，最多只有一个灰度值的差别）。这也反映出我们树模型鲁棒性验证框架的必要性。

5.3.2 GBDT 模型鲁棒性的验证与分析

回归模型的验证

与随机森林的回归模型的验证实验一样，我们同样在波士顿房价数据集上进行了实验。对数据集的划分方式，训练参数的设置都与随机森林的回归模型保持一致。唯一不同的是，在 GBDT 模型中，我们需要设置损失函数，在此实验中，我们选择均方损失函数。同样，设置特征扰动范围值 ϵ 为 3， ρ 值为 5，代表 5000 美元，即在房屋特征扰动的情况下，此模型对房屋价格的预测结果误差不能超过 5000 美元。

图5.3为模型学习率为 0.3 下 GBDT 回归模型的鲁棒性验证结果。从图中我们可以看出，与随机森林回归模型一致，随着树的深度和树的棵树的增加，模型的全局鲁棒性在降低。但在增加同样棵树的决策树情况下，GBDT 的回归模型的鲁棒性要比随机森林模型下降的更快。换句话说，随机森林模型鲁棒性的下降趋势较为“平缓”，而 GBDT 模型鲁棒性的下降趋势则比较“陡峭”。

分类模型的验证

在 GBDT 分类模型的鲁棒性验证实验中，我们同样基于 MNIST 和 FASHION-MNIST 两个数据集上进行了实验验证。数据集的划分方式为：80% 训练样本集和

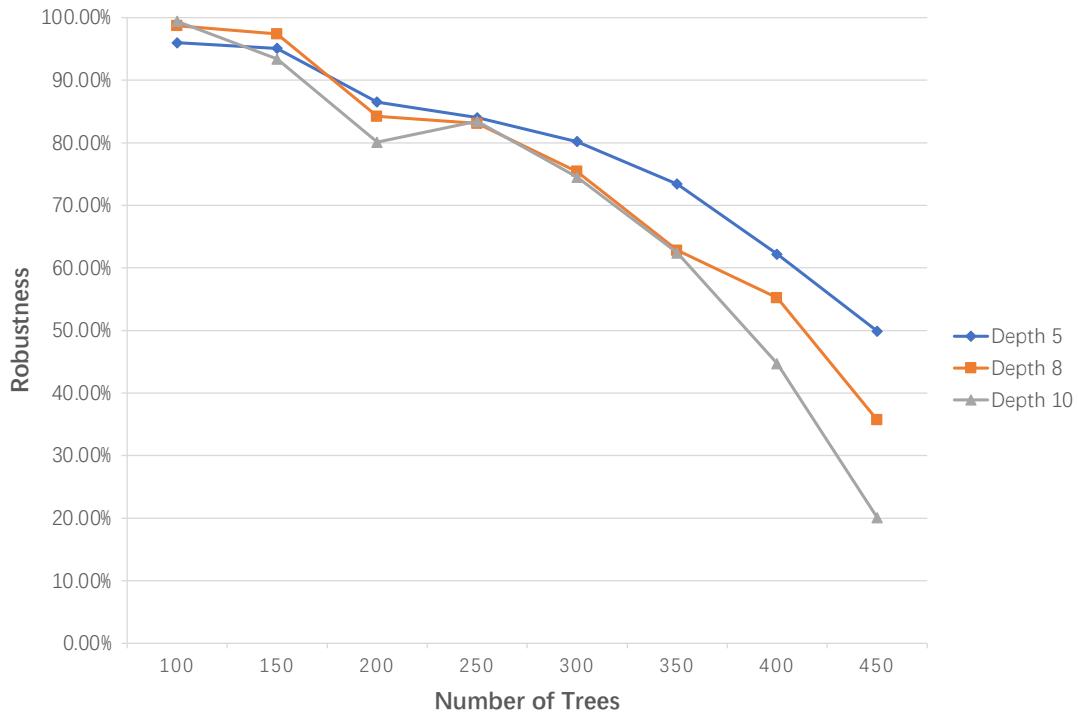


图 5.3: GBDT 回归模型验证结果

20% 测试样本集。之后，从测试样本集中随机抽取了 10 个类别的各 100 个图像，总的鲁棒性测试样本集的大小为 1000。

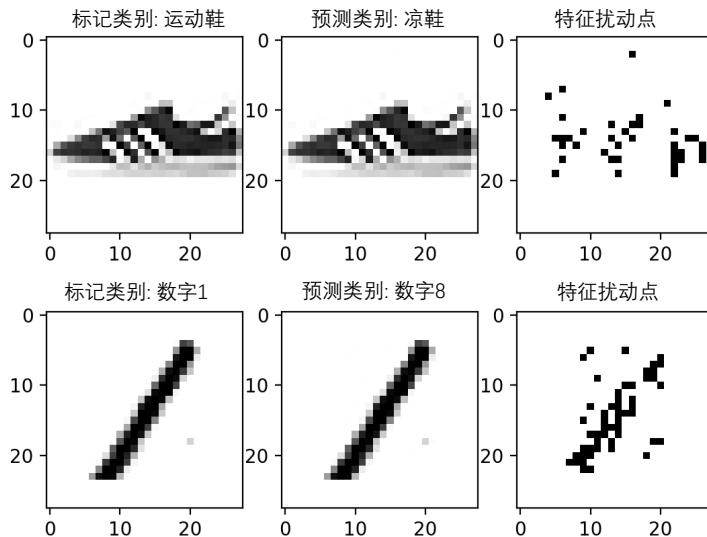


图 5.4: GBDT 验证反例

图5.4显示了 GBDT 分类模型下不满足单样本鲁棒性的测试样本。其特征扰动

范围值 $\epsilon = 3$, 即最多扰动 3 个灰度值。图中的第一列图像为原始的样本, 第二列为第一列相对应的对抗性样本图像, 第三列的图像标记出了受到扰动的特征点。我们可以看出在图像受到扰动之后, 在第一个示例中标记为类别“运动鞋”的商品图像被错误地预测为“凉鞋”。第二个示例中数字“1”被模型错误的预测为数字“8”。在扰动范围设置为 3 个灰度值的情况下, 我们依然无法通过肉眼看出原始图像和对抗性样本图像之间的区别。

5.3.3 树模型鲁棒性可解释性的实验与分析

鲁棒特征集合

根据我们给出的鲁棒特征集合定义, 我们在随机森林分类模型和 GBDT 分类模型中, 进行了相关的实验与分析。根据定义可知, 在测试样本满足单样本鲁棒性的情况下, 我们可以获取其鲁棒特征集合。因此, 我们可以直接在之前分类模型的实验中, 获取满足鲁棒性的样本的鲁棒特征集合。

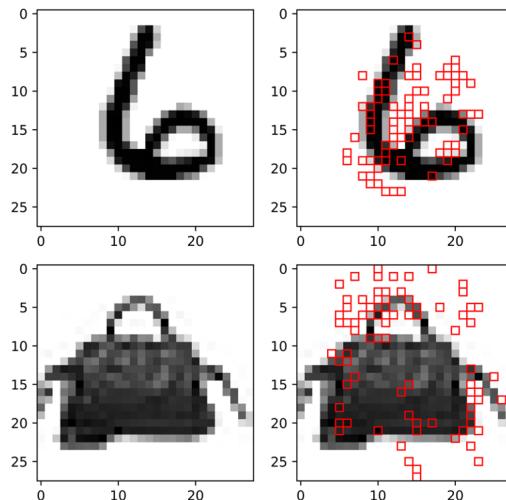


图 5.5: 随机森林分类模型鲁棒特征集合

图5.5显示了在随机森林分类模型下的两个数据集中满足单样本鲁棒性的测试样本的示例。特征扰动范围 ϵ 设置为 3, 代表了 3 个灰度值。在受到扰动的情况下, 这些样本仍然被模型正确识别。图中的第一列显示了原始的样本, 第二列为对应

图像的鲁棒特征集合图。我们用红色矩形标记处了存在于该样本鲁棒特征集合中的特征点。根据鲁棒特征集合的性质，我们知道保持红色矩形标记的特征点的像素灰度值不变，在特征扰动距离最大为 3 个灰度值的情况下任意改变其他特征点的像素灰度值都不能改变模型对该样本的识别结果。为了验证此结论，我们随机的改变不包含于鲁棒特征集合中的特征点的像素灰度值，而保持红色标记点像素灰度值不变，然后让模型去识别将改变后的测试样本之后，去检查预测结果是否发生变化。经过大量的随机测试，以上结论正确。

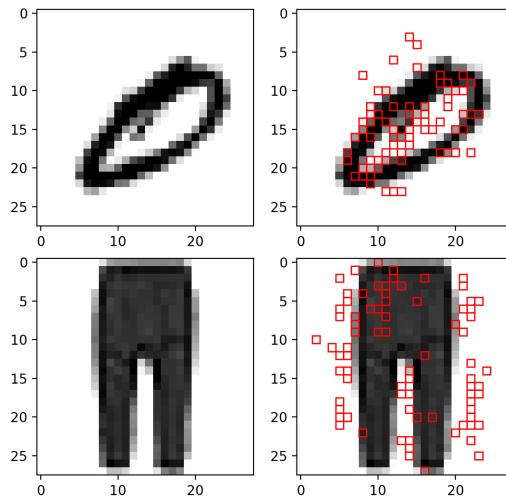


图 5.6: GBDT 分类模型鲁棒特征集合

图5.6显示了在 GBDT 分类模型下的两个数据集中满足单样本鲁棒性的测试样本的示例。特征扰动范围 ϵ 设置为 1。实验过程与随机森林实验保持一致。第一行显示的为数字“0”样本的鲁棒特征集合，第二行显示的为商品“裤子”样本的结果。

局部鲁棒特征重要度

根据我们对局部鲁棒特征重要度的定义，我们首先收集了的在随机森林分类模型测试样本集中所有满足单样本鲁棒性的测试样本，根据算法1，我们可以求得模型不同类别的鲁棒特征重要度。

我们在分别在 MNIST 和 FAHSION-MNIST 数据集中，各自选择了一个类别来计算局部鲁棒特征重要度。图5.7展示了基于随机森林分类模型的实验结果。左侧的图片为 MNIST 中数字“0”的结果，右侧显示了 FASHION-MNIST 中的商品“运动

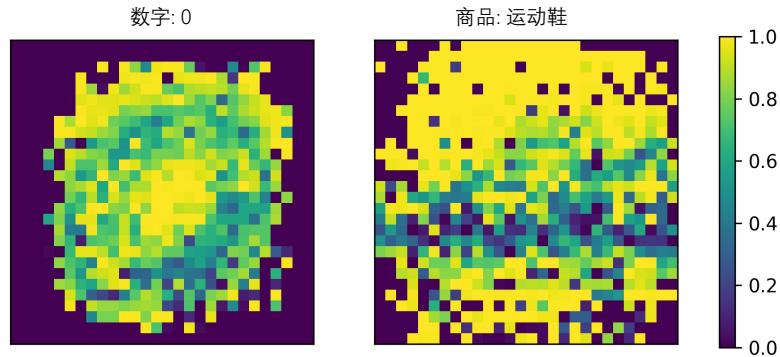


图 5.7: 随机森林分类的局部鲁棒性特征重要度

鞋”的结果。特征点的鲁棒重要度的值越大，则颜色越黄。如果其鲁棒重要度的值为 0，则颜色为紫色。我们可以观察到由于特征点的重要度值的不同，显示出了该类别的基本形状。重要度大的值基本分布在该类别的基本形状周围，而分布在该类别的基本形状之上的特征点的鲁棒重要度值都比较低。这为对抗性样本的攻击提供了新的思路：在进行对抗性样本攻击的时候，应该优先选择这些鲁棒重要度值比较高的点，去产生对抗性样本，这样可以提高攻击的成功率与效率。如果从我们实验得出的特征重要度分布的规律来看，应该优先选择分布在该类别基本形状周围的特征点去进行攻击。需要注意的是，除了我们给出的以上两个类别的结果之外，其他类别的鲁棒特征重要度也有类似的分布规律。

5.3.4 不同类别鲁棒性的验证与分析

在其他关于树模型的鲁棒性的验证研究中，对于分类模型的鲁棒性验证都是针对于该模型的整体而言的。但是同一模型的不同的类别的鲁棒性可能会不同。在此种情况下，整体的模型鲁棒性的验证结果，并不能提供具体类别的鲁棒性信息。所以我们设计了实验去研究同一模型下不同类别的鲁棒性的是否会出现差异的问题。与之前的实验设置保持一致，数据集的划分方式为：80% 训练样本集和 20% 测试样本集，从测试样本集中随机抽取了 10 个类别的各 100 个图像。训练出的树模型的识别率都在 95% 至 98% 之间，并且各个类别的识别率也基本相同。之后我们分别对不同类别的 100 个样本进行鲁棒性验证。

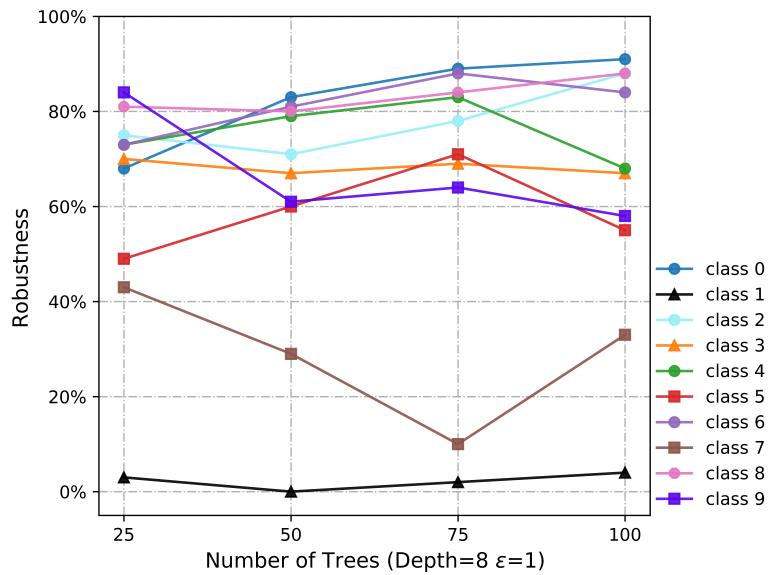


图 5.8: MNIST 中不同类别鲁棒性的验证结果

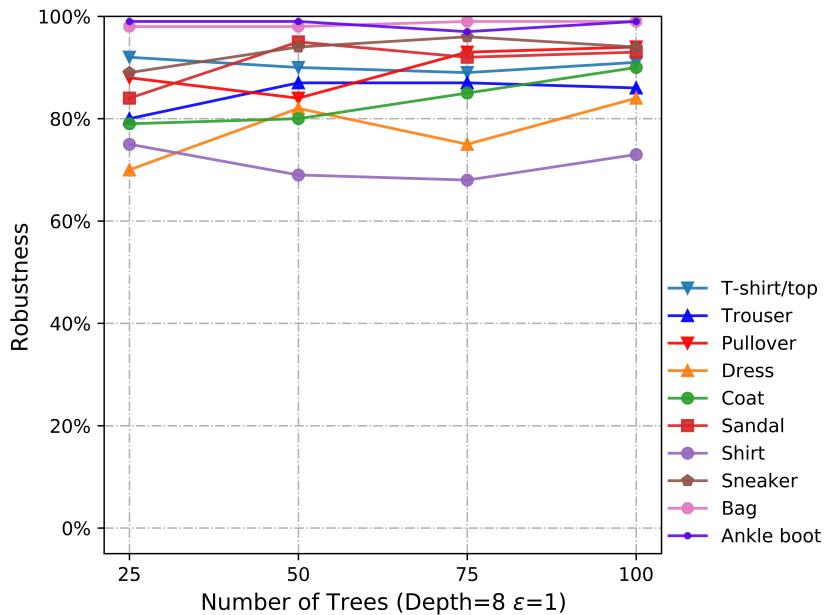


图 5.9: FASHION-MNIST 中不同类别鲁棒性的验证结果

折线图5.8和5.9展示了两个数据集中的不同类别的鲁棒性验证结果。图5.8显示了MNIST数据集的结果。我们可以观察到，存在几个类别（例如，数字“0”，数字“2”，数字“6”，数字“8”）的鲁棒性随着树的棵树的增加而略有提高。而数字“4”，数字“5”，数字“7”，数字“9”类中的鲁棒性值有很明显的波动。除此以外，数字“1”的鲁

棒性始终保持在非常低的值。尽管模型对数字“1”的识别率与其他数字的识别率基本一样，但它们的鲁棒性值却存在着显着差异，差值大约在 40% 至 80% 之间。相比之下，FASHION-MNIST（图5.9）中各个类别的鲁棒性总体上保持稳定，并没有随着树的棵树的增加而出现明显的波动。但是，商品类别为“衬衫”的鲁棒性值相比于其他类别要低一些。在该数据集中，没有类似于数字“1”这种的鲁棒性非常低的类别去影响该模型的总体的鲁棒性值。根据此次的实验结果，我们可知在验证树模型鲁棒性的时候，将注意力集中在整个模型的鲁棒性上是不准确的，对于不同的数据集来说，模型对于不同类别样本的鲁棒性的表现可能会有很大的差别。这给了我们的一个启示，在验证分类模型鲁棒性的时候，验证结果应该细化到不同的类别。这些信息对于模型的使用者来说是非常有用的，可以让他们更加详细的了解到该模型的优缺点，从而增加了模型的可信度。

5.3.5 树鲁棒性超参数与鲁棒性关系的验证与分析

在本文提出的鲁棒性验证框架下，我们进一步研究了树模型中两个重要的训练超参数：树的棵树和树的深度与树模型鲁棒性的关系。我们基于 MNIST 数据集去进行这部分的实验。通过在不同深度，不同树的棵树参数下去训练模型，通过对比其鲁棒性结果来进行研究和分析。

表5.1为随机森林分类模型在不同训练参数下的鲁棒性结果，其中 Trees 和 Depth 分别表示模型中树的棵树和树的深度，Accuracy 表示的是模型的识别率，Verified(ρ) 表示模型的全局鲁棒性，Timeout 表示验证超时，Failed 表示验证失败即不满足单样本鲁棒性所占的百分比。我们可以观察到，在特征扰动值为 $\epsilon = 1$ 的情况下，保持相同树的深度，树的棵树并不会对其鲁棒性造成很大的影响，但在之前的对回归模型的实验中，在保持树的深度相同的情况下，树的棵树的增加，会导致其模型鲁棒性的降低。此外，在保持树的棵树相同的情况下，增加树的深度参数的值，会使模型的鲁棒性少量的增加。与之相反的是，对于其回归模型来说，树的深度的增加，会导致其模型鲁棒性的降低。根据我们的实验结果，在保证模型准确率的情况下，模型的开发人员可以通过调整训练参数来增加其模型的鲁棒性。与

表 5.1: 基于 MNIST 数据集模型在不同超参数下的鲁棒性验证结果.

Trees	Depth	Accuracy	$\epsilon = 1$			$\epsilon = 3$		
			Verified(ρ)	Timeout	Failed	Verified(ρ)	Timeout	Failed
25	5	84%	45.71%	0%	54.29%	9.64%	0.12%	90.24%
50	5	85%	54.68%	0%	45.32%	13.58%	2.69%	83.72%
75	5	86%	48.77%	5.61%	45.61%	9.24%	13.68%	77.08%
100	5	86%	54.07%	14.53%	31.40%	13.02%	21.74%	65.23%
25	8	91%	61.47%	0%	38.53%	9.88%	0.11%	90.01%
50	8	93%	61.02%	0%	38.98%	14.36%	3.02%	82.61%
75	8	92%	63.63%	5.32%	31.05%	12.81%	17.05%	70.14%
100	8	93%	63.48%	15.04%	21.48%	16.86%	24.81%	58.32%
25	10	93%	64.34%	0%	35.66%	14.18%	0%	85.82%
50	10	95%	63.21%	0%	36.79%	17.34%	0.53%	82.14%
75	10	94%	75.32%	5.32%	19.36%	13.83%	4.57%	81.60%
100	10	95%	66.84%	8.74%	24.42%	15.16%	14.21%	70.63%

$\epsilon = 1$ 时做对比, $\epsilon = 3$ 的情况下, 该模型的鲁棒性有了明显的降低, 这是显而易见的, 因为在特征扰动范围为 3 个灰度值的情况下, 会产生更多的对抗性样本使得原始样本的鲁棒性不满足。

还有一点值得我们注意, 随着树的棵树和树的深度的值的增加, 验证超时的比例也在增加, 这揭露了我们验证框架的不足。因为从本质上来说, 验证框架的验证能力一定程度取决于 Z3 求解器的求解能力, 当树模型规模变大的时候, 我们编码形成的 SMT 公式数目也是剧增的, 这就导致状态爆炸的问题, 从而使得求解器无法求解, 导致验证超时, 无法确定该样本的鲁棒性是否满足。

5.3.6 验证时间的结果与分析

框架的验证时间同样也是我们需要关注的部分, 我们需要保证在一定时间内返回正确的验证结果。于是, 我们基于 MNIST 数据集, 通过验证不同规模大小的树模型来统计该框架的验证时间。

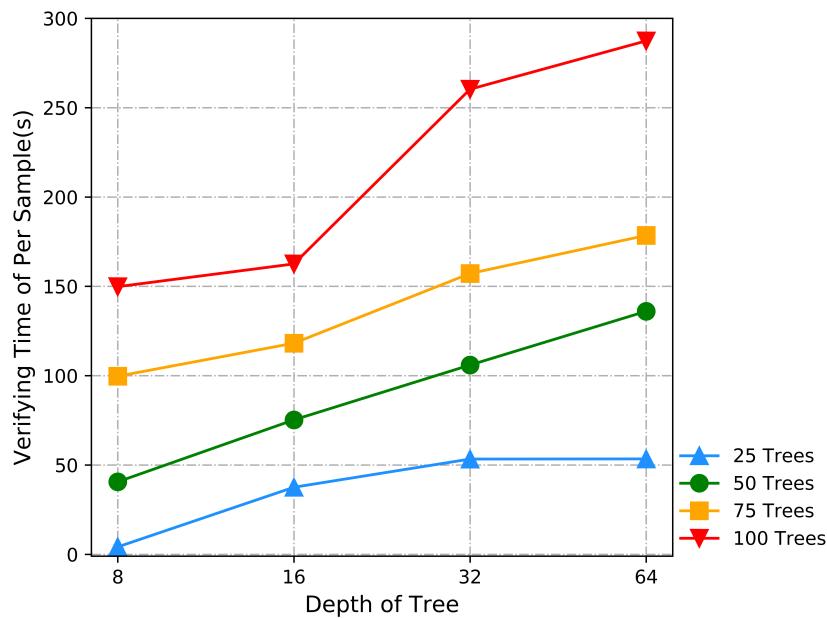


图 5.10: 单样本验证时间图

图5.10为在不同规模树模型下验证单个样本所需要的平均时间统计图。我们在测试样本集中随机选择了 100 个样本来进行验证时间的结果统计，结果值为平均值。我们的验证覆盖了规模很小到大规模的树模型，最小规模为深度为 8，棵树为 25 的树，单个样本的验证时间为 4s。而对于树的棵树为 100，深度为 64 的大规模的模型，我们框架的验证时间为 287s。随着树模型规模的增加，验证时间也在逐渐增加。总体来说，我们框架可以去验证大规模的树模型的鲁棒性，而且验证时间也是可接受的时间范围。但如表5.1所显示的，在进行大规模模型的验证的时候，有些样本可能会出现验证超时的情况，这也是我们在未来工作中，需要解决的问题。

5.4 本章小结

在本章中，我们选取了三个基准数据集对本文提出的鲁棒性验证框架进行了一系列的实验来测试其可行性，并且对树模型鲁棒性的可解释性和树模型训练参数与鲁棒性的关系也进行实验和研究。实验结果表明，我们的验证框架可以有效验证随机森林和 GBDT 这两个树模型的重要组成部分。但也存在不足，就是虽然

可以验证大规模的树模型，可是某些样本还是会出现验证超时的情况，这将是我们未来工作中的重点。

第六章 总结与展望

6.1 总结

随着机器学习在各个领域的大规模应用，尤其是在安全领域也占有了重要的地位。机器学习模型的安全性和可解释性也引起各国政府和研究学者的极大关注。鲁棒性是模型安全性的重要体现之一，所以验证模型鲁棒性的方法和工具也变得尤为迫切。

树模型以其高效，方便，泛化能力强的特点，在各个领域都有广泛的应用。但与神经网络模型一样，树模型也易收到对抗性样本的影响。本文基于 SMT 技术对机器学习树模型的鲁棒性进行了研究和分析。本文的主要工作和贡献如下：

- (1) 本文提出了一个基于 SMT 技术树模型的鲁棒性验证框架，该框架支持树模型中两个重要实现：随机森林与 GBDT 模型的鲁棒性的验证。该框架能够有效验证大规模的树模型的鲁棒性。
- (2) 本文提出了鲁棒特征集和局部鲁棒特征重要度的概念，从模型可解释性的角度，进一步研究了树模型的鲁棒性和样本特征之间的关系。为对抗性样本反例的生成和对抗性攻击提供新的思路，也可以帮助模型开发人员进一步优化模型提高其鲁棒性。
- (3) 本文通过实验讨论了树模型超参数与模型鲁棒性的关系，从而为训练阶段提高模型鲁棒性的研究提供了重要参考。

综上所述，本文的研究充分证明了所提出框架的有效性，可以极大的增加树模型的可靠性，同时也对鲁棒性的可解释性做了研究，从而进一步推进了树模型

在安全领域的应用和发展。

6.2 展望

我们的研究还留存一些待解决的问题，可以考虑从下面的几个方面展开研究：

- (1) 本文所提出的鲁棒性验证框架的验证能力很大程度上受限于 SMT 求解器本身的求解能力，在未来应该考虑针对树模型编码成 SMT 公式的特点，对 SMT 的底层求解算法进行优化，从而提高验证的效率。
- (2) 虽然在现有的验证框架下，可以支持一些大规模树模型的验证，但是对于高维度，更大规模的模型还是会发生状态爆炸的问题，导致验证结果无效。在未来可以先对树模型本身先进行模型缩减的操作，之后再进行验证，以便可以验证超大规模的模型。
- (3) 我们的验证框架在对树模型的鲁棒性验证问题上已经取得了一定的成效，接下来可以将其扩展到复杂度更高的机器学习模型上去。目前，已有研究者考虑将神经网络转换为决策树，使其模型的复杂度降低。我们可以沿着这个思路，将我们的方法扩展到神经网络模型上去，进一步加强框架的验证能力。