计算机科学
COMPUTER SCIENCE

基于改进Self-paced Ensemble算法的浏览器指纹识别

张德升，陈博，张建辉，卜佑军，孙重鑫，孙嘉

引用本文

张德升，陈博，张建辉，卜佑军，孙重鑫，孙嘉基于改进Self-paced Ensemble算法的浏览器指纹识别[J]. 计算机科学, 2023, 50(7): 317-324.

ZHANG Desheng, CHEN Bo, ZHANG Jianhui, BU Youjun, SUN Chongxin, SUN Jia. Browser Fingerprint Recognition Based on Improved Self-paced Ensemble Algorithm [J]. Computer Science, 2023, 50(7): 317-324.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.220600068

基于改进 Self-paced Ensemble 算法的浏览器指纹识别

张德升[1]　陈　博[2]　张建辉[2]　卜佑军[2]　孙重鑫[2]　孙　嘉[1]
1 郑州大学网络空间安全学院　郑州 450000
2 中国人民解放军战略支援部队信息工程大学信息技术研究所　郑州 450000
(835225140@qq.com)

摘　要　浏览器指纹技术凭借其无状态、跨域一致等优点,已经被许多网站应用到用户追踪、广告投放和安全验证等方面。浏览器指纹识别的过程是典型的不平衡数据的分类过程。针对当前浏览器指纹长期追踪过程中存在数据样本类不平衡导致指纹识别准确度低、长期追踪易失效等问题,提出了改进的 Self-paced Ensemble(Improved SPE,ISPE)方法应用于浏览器指纹识别。对浏览器指纹样本欠采样过程和集成学习单个分类器的训练过程进行了改进,重点针对难以识别的浏览器指纹,添加类注意力机制并优化自协调因子,使分类器在训练和识别浏览器指纹的过程中更加注重边界样本的分类效果,从而提升总体的浏览器指纹识别准确度。在所收集的 3483 条指纹和开源数据集中的 15000 条指纹上进行了实验,结果表明,ISPE 算法在浏览器指纹匹配识别的 F1-score 达到 95.6%,相比 Bi-RNN 算法提高了 16.8%。

关键词:浏览器指纹;用户追踪;Self-paced Ensemble;欠采样;集成学习
中图法分类号　TP393

Browser Fingerprint Recognition Based on Improved Self-paced Ensemble Algorithm

ZHANG Desheng[1],CHEN Bo[2],ZHANG Jianhui[2],BU Youjun[2],SUN Chongxin[2] and SUN Jia[1]
1 School of Cyber Science and Engineering,Zhengzhou University,Zhengzhou,450000,China
2 Information Technology Institute,PLA Strategic Support Force Information Engineering University,Zhengzhou 450000,China

Abstract　Browser fingerprinting technology has been used by many websites for user tracking,advertising delivery and security verification due to its stateless,cross-domain consistency and other advantages. The task of browser fingerprint recognition is a typical classification task of imbalanced data. The data imbalance exists in browser fingerprint long-term tracking task,which will lead to low accuracy of fingerprint recognition and failure of long-term tracking. An improved Self-paced Ensemble(ISPE) method is proposed to identify browser fingerprints. And the undersampling process of browser fingerprint sample and the training process of single classifier in ensemble learning are improved. Focusing on the browser fingerprint which is difficult to identify,added attention-like mechanism and self-paced factor are optimized to make the classifier pay more attention to the boundary samples which are difficult to classify in the training process,to improve the overall accuracy of browser fingerprint recognition. The results show that the F1-score of ISPE algorithm for browser fingerprint recognition reaches 95.6%,which is 16.8% higher than that of Bi-RNN algorithm. It proves that the method has excellent performance for long-term browser fingerprint tracking.

Keywords　Browser fingerprinting,User tracking,Self-paced Ensemble,Undersampling,Ensemble learning

1　引言

网站出于多种原因需要追踪用户身份,有的是出于服务功能的考虑,有的是为了推送广告和信息,有的则是出于对安全性考虑[1]。传统的用户身份追踪方法是基于 cookie 的,但是近年来很多网站对 cookie 的滥用,导致人们越来越不信任 cookie[2],很多人会装有清理 cookie 的插件或者直接使用隐私模式。在 cookie 技术越来越低效[3]的同时,浏览器指纹技术逐渐发展,成为新的用户追踪主流技术。浏览器指纹是由 Eckersley[4]于 2010 年提出的概念,它由一系列可通过常规 API 和 HTTP 请求头获取的浏览器特征信息组成,如请求时的 user-agent、accept-language、屏幕分辨率等。表 1 列出了一个浏览器指纹的样本。

尽管浏览器指纹可作为用户浏览器身份的唯一标识符[5-7],但是浏览器指纹可能因为用户更改配置、安装插件、升级版本等行为而产生变化,要实现准确的用户追踪,必须要

到稿日期:2022-06-07　返修日期:2022-10-14
基金项目:国家自然科学基金(62176264)
This work was supported by the National Natural Science Foundation of China(62176264).
通信作者:张建辉(ndsczjh@163.com)

处理变化的浏览器指纹，对浏览器指纹进行识别。具体地，在识别过程中需要对新来的未知指纹和已知的指纹进行一一匹配，理论上正样本数应为 0 或 1，即新访问浏览器是初次访问或者是老用户的新指纹。这一过程为在不均衡样本上的二分类问题，不均衡的数据样本会导致分类器学习到训练集中样本比例这种先验性信息[8]，以致于实际预测时就会更倾向于将其预测为样本数量较多的多数类。以浏览器指纹分类为例，其会更倾向于判断两个指纹为不同浏览器产生的。传统的不均衡问题中，如安全检测中的常规记录和异常警报，由于错误分类的代价不同，可以通过提升阈值来提高查全率，进而提升系统总体性能。但是在浏览器指纹识别问题上，正负样本分类错误的代价是相同的，都会导致识别失败，这对浏览器指纹识别算法提出了更高的要求。

表 1 浏览器指纹样本

Table 1 Browser fingerprint sample

特征	值
user-agent	Mozilla/5.0(Windows NT 10.0;Win64;x64) AppleWebKit/537.36(KHTML,like Gecko) Chrome/92.0.4495.0 Safari/537.36
accept-language:	zh-CN,en-US;q=0.5
IP	104.193.88.123
platform	Windows
local time	Mon Jul 05 2021 09:57:32 GMT+0800(China Standard Time)
Resolution	1 920×1 080
webgl renderer	ANGLE(Intel(R) HD Graphics 4600 Direct3D11 vs_5_0 ps_5_0)
...	...

针对浏览器指纹识别面临的上述问题，本文对浏览器指纹不均衡数据的影响进行分析，针对性地提出了基于 Self-paced Ensemble(SPE)[9]的改进算法，称为 ISPE(Improved Self-paced Ensemble)。该算法首先采用欠采样和集成学习相结合的方法，能够在训练阶段综合考虑分类器的分类性能和数据特征。其次，针对浏览器指纹数据，优化自协调因子 α，使得算法更加聚焦于难以分类的边界数据。然后添加了类注意力机制，在实现类似注意力权重分配的同时，省去了训练过程，并且该机制具有很强的泛用性。最后在本文自主收集的数据集和开源数据集上进行了实验，验证了该方法可缓解将浏览器指纹用于 web 用户长期追踪面临的类不平衡问题，提升了长期追踪的准确率，进而实现了网站对用户持续追踪的优异性能。

2 相关工作

最早在文献[4]中提出浏览器指纹概念时，Eckersley 提出了基于规则的识别算法，针对变化的浏览器指纹进行识别，文中选取了 8 个重要的特征，如果只有一个特征改变，便仍视其为同一浏览器。该方法中特征的选取和改变的数量是基于 Eckersley 的实践经验，并没有理论支撑。文献[10]提出了基于聚类思想的指纹识别算法，以衡量未知指纹与浏览器指纹集群的相似度，判断该指纹是否属于某个浏览器。识别过程中充分利用数据集特征，实现相同浏览器的自动聚类识别。文献[11]提出了基于 Levenshtein 距离的识别方法，处理了浏览器插件列表这种字符串型特征。在此基础上，Vastel 等[12]在 S&P 发表了一篇比较有建设性的文献，提出了基于规则和基于随机森林的两种方法，针对变化的浏览器指纹进行识别和持续跟踪，达到了 26% 的浏览器指纹在 100 天后依旧能被准确识别的效果。文献[13-14]采用循环神经网络来追踪变化的浏览器指纹，并结合注意力机制来合理地重点关注影响指纹分类效果的特征，提高了模型的鲁棒性和准确性。

之前的研究重点在于选取合适的特征[15-17]和算法来提升匹配的准确度，但从未探讨过类不平衡对匹配准确率的影响，文献[14]的研究虽然指出了类不平衡对算法准确度的影响，但并未进行进一步的探讨，只是先采用样本增强技术增加了总体样本数量，然后采用简单随机抽取的欠采样技术来平衡类数量，其目标也是增强训练效果的稳定性而并非提升训练准确度。

3 浏览器指纹的处理和分析

3.1 基于 ISPE 的浏览器指纹识别模型

浏览器指纹用于身份识别一般需要经过以下几个流程：1)客户发送 web 访问请求；2)服务器返回客户请求，同时包括浏览器指纹的 js 脚本；3)浏览器客户端执行 js 脚本生成浏览器指纹特征；4)回传浏览器指纹；5)传输进入数据库；6)浏览器指纹送入匹配算法；7)新的浏览器指纹和已知指纹进行匹配，关联结果写回数据库。工作流程如图 1 所示。

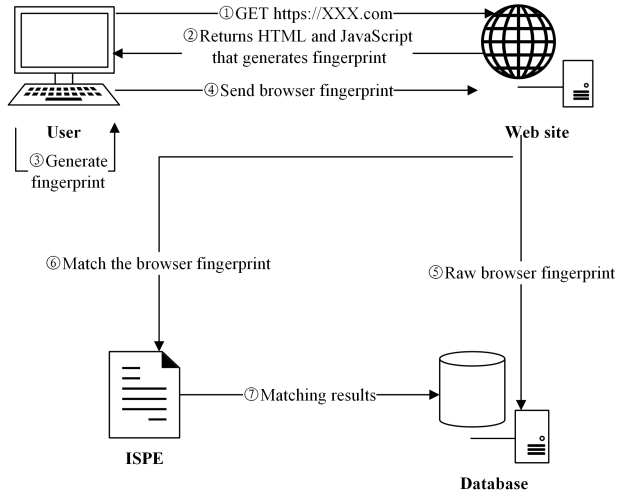

图 1 ISPE 工作流程

Fig. 1 ISPE workflow

本文搭建了一个基于 web 的浏览器指纹收集服务器，具体的实现如下：对于步骤①和步骤②，通常使用的是标准的 http 协议。步骤③浏览器客户端执行的脚本是在开源项目 fingerprinttJS[18]的基础上进行了更改，增加了采取多种特征的代码，丰富了浏览器指纹的特征种类和数量。为了尽量保存原始信息，以供研究利用，步骤④把所有的特征原始值封装成 JSON 数据，以 POST 的方式回传到指纹服务器。在实际应用的工作环境中，可以采取将 hash 部分信息进行传输，若 cavans 特征原数据太大，hash 之后可以优化传输速度。步骤⑤

中先把数据写入数据库进行存储，然后再送入指纹识别算法进行识别，指纹匹配流程如图 2 所示。ISPE 算法中会把所有浏览器的最新指纹放在内存中，未知指纹进入之后依次和内存中所有浏览器的最新指纹进行匹配，如果判定结果为正，则把未知指纹视为该浏览器的最新指纹并进行替换，再把指纹链接关系写入数据库。如果匹配完所有已知指纹，依旧没匹配到可以视为同一浏览器产生的指纹记录，则将其视为新指纹，并添加进算法中的浏览器指纹列表，视为一个新的浏览器实例。

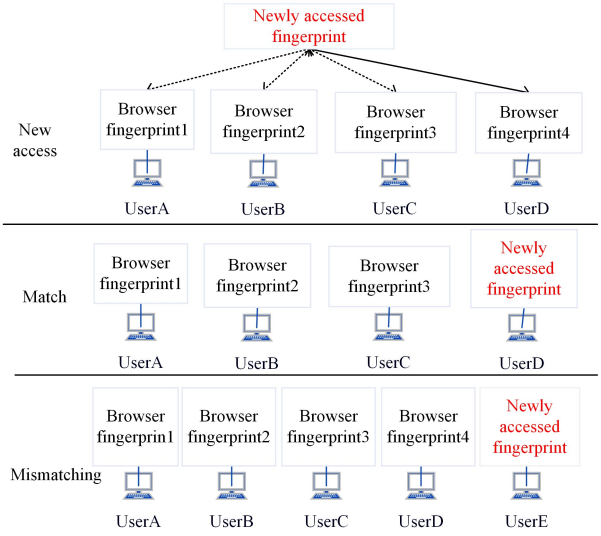

图 2 指纹匹配流程

Fig. 2 Process of fingerprint matching

3.2 特征处理

浏览器指纹的特征种类繁多，具体的种类和数量与浏览器指纹收集方法有关。为了使指纹数据更具通用性，以便用于机器学习算法，需要对浏览器指纹进行预处理，使其具有统一的格式。本文算法每次将一对指纹作为最基本的输入特征，即待匹配的指纹和已知的浏览器指纹转化为一组特征向量 $\boldsymbol{X}=[x_1,x_2,\cdots,x_n]^{\mathrm{T}}$ 输入集成学习模型，输出为 $F(X)=y,y\in\{0,1\}$。其中的特征向量 $\boldsymbol{X}=Gener(row\,X_1,row\,X_2)$，其中，$row\,X_n$ 表示某个原始的指纹特征，$Gener()$ 函数则是指纹对生成算法。

通常，在浏览器指纹的数据集上，$Gener()$ 函数主要需要处理以下 4 类特征：数值型、0-1 型、离散型、字符串型。

(1)数值型。其主要包括浏览器指纹特征中的 sin，cos 等值，本文主要计算的是其相对差值，具体方法如式(1)所示：

$$Gener_{dig}\{row\,X_1[dig],row\,X_2[dig]\}=row\,X_1[dig]-row\,X_2[dig] \tag{1}$$

值得注意的是，本文将其中的日期型特征和创建日期等视为数值型，并把其基本单位转化为秒，计算其差值。

(2)0-1 型特征。主要包括是否启动 cookie、是否启用 DNT 标志等。为了保留尽可能多的原始信息，两个特征都予以保留，并进行简单连接，具体方法如式(2)所示：

$$Gene\,r_{01}\{row\,X_1[01],row\,X_2[01]\}=[row\,X_1[01],row\,X_2[01]] \tag{2}$$

(3)离散型。主要指浏览器指纹的特征只可能取有限个或至多可列个值。代表性的特征有屏幕分辨率、时区等。本文的处理方式是判断其是否相等，转化为 01 值。具体方法如式(3)所示：

$$Gener_{sca}\{row\,X_1[sca],row\,X_2[sca]\}=row\,X_1[sca]==row\,X_2[sca] \tag{3}$$

另外对于 canvas 属性和 webgl 属性，虽然其本质是图片信息，但具体使用时本文只取其 hash 值作为唯一标志，因此它们也被视为离散型特征进行处理。

(4)字符串型。主要有 userAgent，acceptHttp 和 vendor-WebGLJS 等。对于这些字符串的处理方式，本文主要采取的是 Levenshtein 距离，即编辑距离来衡量两个指纹字符串属性的相似度，具体方法如式(4)所示：

$$Gener_{str}\{row\,X_1[str],row\,X_2[str]\}=diff(row\,X_1[sca],row\,X_2[sca]).radio() \tag{4}$$

其中，$radio()$ 函数把 laven 距离转化为 0～1 区间内的值。languageHttp，pluginsJS 等属性本身是由多个离散的特征值组成的，虽然将它们切割成多个离散值进行处理是可以的，但是由于其长度变化较大，可能要截断或者预留多个空白位置，不如将其视为字符串比对，从而能更好地刻画两个字符串的差异性。

3.3 类不平衡分析

我们把本文的项目发布在校园 BBS 上邀请同学们访问，以帮助我们进行浏览器指纹的收集。要求他们使用各自设备上的浏览器，在不同时间，分多次访问我们的浏览器指纹收集服务器。在 2021 年 10 月 4 日到 2022 年 1 月 4 日期间，共收集到浏览器指纹记录 3 483 条。浏览器指纹收集服务器把 cookie 存在时长设为 time()+10 * 365 * 24 * 3 600，以 cookie 作为浏览器身份的真实标志，共有浏览器实例 1 313 个，其中 14.6%的浏览器展现了多个浏览器指纹，浏览器的指纹数量分布如图 3 所示。

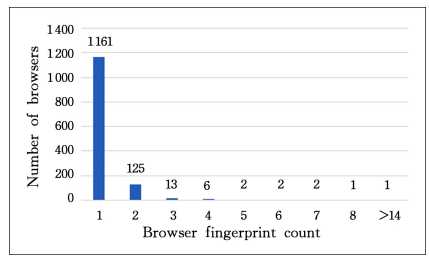

图 3 浏览器指纹统计

Fig. 3 Browser fingerprint statistics

从图 3 可以看到，许多用户的浏览器实例都存在同一浏览器多个指纹的情况。即便在当前数量的数据情况下，当一个浏览器访问服务器产生了一个新的浏览器指纹，如果不做基于操作系统和浏览器的提前分类，只是单纯地依次比配，新的浏览器指纹需要分别和 1 313 个已知浏览器指纹进行匹配，而其中的正样本理论上应该只有 1 个或者 0 个。即便是采用了一些特征进行预分类，类不平衡的问题依旧存在，这是将浏览器指纹运用到用户身份识别无法回避的一个问题。

3.4 类不平衡的影响

本文针对 3.3 节的数据集，在不对正负样本数量做任何处理的前提下，训练了 3 种识别算法，分别是决策树、支持向量机和神经网络算法。其中决策树采用的是 CART 算法，最大深度为 10；支持向量机的核函数为通用性交强的 rbf(径向基函数)；神经网络为 3 层 64 节点隐藏层的全连接神经网络，优化器为 Adam，学习率为 0.001，dropout 为 0.5。数据集样本总数 $N=1313$ 个浏览器实体，在指纹识别算法的准确度为 p 的前提下，新来的 M 个指纹能够正确识别的数学期望 $E(M)$ 如式(5)所示：

$$E(M)=M\sum_{i=1}^{n}\frac{1}{N}p^{i} \tag{5}$$

为了简化分析过程，在这个假设中，每次指纹识别独立同分布，在实际流程中错误的分类会覆盖掉正确的指纹，并且会对后到的指纹识别产生影响，进而对错误分类更加敏感。

本文采取最后 100 条指纹作为新来的指纹进行测试，测试结果如表 2 所列。

表 2 类不均衡下的识别效果

Table 2 Recognition results under class imbalance

Algorithm	Accuracy	Theoretical expectation	Number of identifications
Decision tree	0.998	35.33	32
SVM	0.994	12.69	18
FNN	0.972	2.72	6

从表 2 可以看到，在不做任何处理的前提下，类不均衡的数据样本会使得分类器即使在训练集上表现良好，但是在实际应用中因需要进行大量匹配而最终的效果并不优秀，且最终的识别结果对识别算法的识别精度极其敏感，如决策树和支持向量机的准确率差距仅为 0.4%，最终使得识别结果数量相差近一倍。

4 ISPE 的改进策略

4.1 SPE 算法

SPE 是基于分类难度的衡量指标的欠采样＋集成学习的方法，分类难度的含义为样本对于分类器来说正确分类的难易程度，具体形式为某个样本 (x,y) 对于分类器 $F(x)$ 的分类难度为 $H(x,y,F)$。其中 F 为任意标准分类器，H 可以是任意误差函数，但它必须具有单个误差之和等于总体误差的性质，如绝对误差、平方误差和交叉熵，当 H 为绝对误差时，$H(x,y,F)=|F(x)-y|$。

其训练流程如下：1)首先根据难度把多数类样本分进 K 个桶，K 的大小为超参数；2)其次根据自协调因子 α 分别对每个桶进行欠采样，α 用来调节每个桶采样时样本的数量占总样本的比例；3)然后用新的多数类的欠采样样本和少数类样本训练一个新的基础分类器；4)再根据新训练出来的基础分类器更新多数类的难度值。重复步骤 1－步骤 4 n 次，n 为基础分类器的个数，直到所有的基础分类器都被训练完成，其具体流程如图 4 所示。

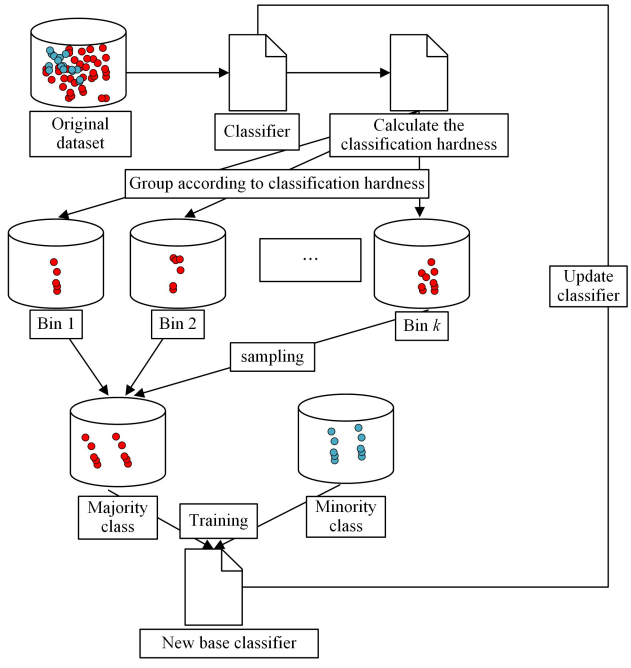

图 4 SPE 训练流程

Fig. 4 SPE workflow

4.2 自协调因子 α

自协调因子 α 的目标是协调各个桶中样本的难度贡献和数量比例。当 $\alpha\to 0$ 时，更注重平均每个桶采样的样本对欠采样的样本的总体难度贡献。当 $\alpha\to\infty$ 时，更注重平均每个桶采样的样本数量占欠采样的样本的总数量的比例。

原始的 SPE 算法中 α 的更新策略为 $\alpha=\tan\left(\frac{i\pi}{2n}\right)$，其中 i 为更新迭代的次数。α 存在后期无法聚焦边界样本的问题，原始的更新策略实质在于前期选取倾向于难度较低的样本，后期倾向于平均不同难度样本数量。而在浏览器指纹识别中，判断完全一样的指纹和差距非常大的指纹并不困难，我们更关注一两个特征不一样的指纹对，即所谓的边界数据，原方法无法对边界数据提供更多的关注。我们的改进思路为，更改后期 α 因子的作用和更新方式，使其不再趋于各个桶中采样数量上的平均，而是更多地关注难度较高的样本，进而提升系统整体的分类准确度。

具体方法为：前 $\frac{n}{2}$ 个循环中，还是采用原先的采样更新策略，第 k 个桶中的采样样本占采样本总体数量的比例为 $\frac{P_j}{\sum_{j=1}^{k}p_j}$，其中 $p_i=\frac{1}{h_i+\alpha}$，h_i 为第 i 个桶中的平均难度，后 $\frac{n}{2}$ 次循环中，$p_i=h_i+\alpha$，由于是二分类，因此 h_i 不会大于 1，其中 $\alpha=\lg_{(n-i+1)}$。

4.3 类注意力机制

注意力机制(Attention Mechanism)源自认知科学，由于信息处理存在瓶颈，人类会选择性地关注所有信息的一部分，同时忽略其他可见的信息。Attention 实质上是一种分配机制，其核心思想是基于原有的数据找到其间的关联性，然后突出其某些重要特征。

由于 SPE 算法每次在训练新的基础分类器之前都要进行

欠采样，其必然会在一定程度上影响数据的分布特征，为了使分类算法都能更准确地聚焦重要权重，本文在数据样本输入前添加了注意力层。主流的注意力层，大多采用的是文献[19]的自注意力机制，但是这种注意力机制需要大量数据进行学习，时间成本高，且欠采样之后的数据样本量较少，易造成欠拟合。

本文采用了注意力机制的思想，提出了类似于注意力机制的算法，将其应用于将原始数据输入训练器之前。对于欠采样过后的数据集，计算所有特征之间的相关系数 ρ_{ij}，如式(6)所示：

$$\rho_{ij}=\rho(x_i,x_j)=\frac{Cov(x_i,x_j)}{\sqrt{Var|x_i|Var|x_j|}} \tag{6}$$

然后把相关系数使用 Softmax 函数进行归一化处理。如式(7)所示：

$$\alpha_{ij}=\mathrm{Softmax}(\rho_{ij})=\frac{e^{\rho_{ij}}}{\sum_{k=1}^{n}e^{\rho_{ik}}} \tag{7}$$

最后把所有特征与 α 加权和之后作为输出，也就是基础分类器的输入，如式(8)所示。从 x_i 到 b_i 的处理流程如图 5 所示。最终结果为 $A(X)=[b_1,b_2,\cdots,b_n]^{\mathrm{T}}$，其中函数 A 为类注意力层。

$$b_i=\sum_{k=1}^{n}\alpha_{ik}\cdot x_i \tag{8}$$

本文方法采用相关系数，在功能上与自注意力机制中 $\boldsymbol{q}$，$\boldsymbol{v}$ 矩阵类似。为了计算输入特征之间的相关度，直接采用特征值与相关度相乘，相当于把自注意力机制中的 $\boldsymbol{k}$ 矩阵取为单位矩阵。这样做的目的是在实现注意力机制对权重的关注的同时，省略训练过程，从而节省大量时间，并且能够做到与基础分类器低耦合。

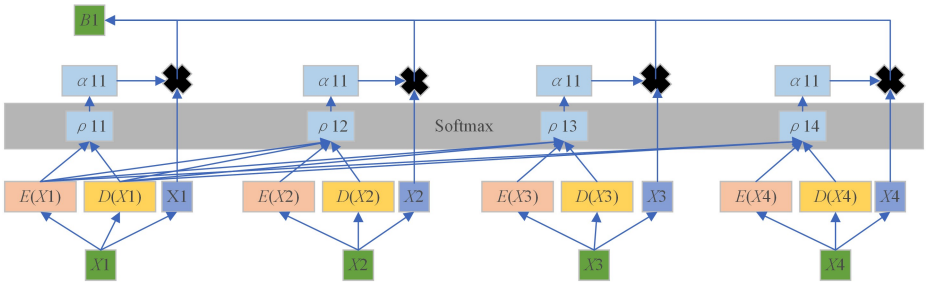

图 5 类注意力机制

Fig. 5 Attention-like mechanism

4.4 分类器权重

集成学习是通过训练多个模型，把多个模型组合在一起来提升总体的分类能力。其单个分类器的出错概率为 ε，在多个单个分类器独立且有效的情况下，有 N 个基础分类器的集成分类器的错误率的表达式如式(9)所示：

$$\sum_{i=\frac{n}{2}+1}^{N}\binom{N}{i}\varepsilon^{i}(1-\varepsilon)^{N-i} \tag{9}$$

根据以上原理，多个弱分类器也能组合成一个具有强鲁棒性的强分类器，集成学习的方法可以把不均衡数据类型分类方法和基础分类算法组合在一起。

原方法训练过后的基础分类器是平权的，即最终分类器 F 表示为 $F(x)=\frac{1}{n}\sum_{m=1}^{n}f_m(x)$，其中 $f(x)$ 为基础分类器。但是对于浏览器指纹匹配，存在非常容易判断的平凡样本和难以判断的边界样本。为了提升综合考虑不同基础分类器的分类能力，不可以对所有的分类器都分配相同的权重。前期训练的样本会更多地处理平凡样本，应该具有相对更高的权重，而进行之后的第 $\frac{n}{2}$ 次循环，训练集会更多地处理边界样本，新训练出的分类器的正确率会降低，因此应该适当降低训练器的权重。

具体的做法是：为每一个基础分类器配置一个权重 w，最终分类器的分类结果为 $F(x)=\frac{1}{n}\sum_{m=1}^{n}w_m f_m(x)$。通过调整分类器权重 w 来影响最终的分类结果，提升分类精度。

5 实验与分析

5.1 实验环境

本实验采用的是 LENOVO 台式机，操作系统为 Windows 11 Pro 64-bit，CPU 为 Intel(R) Core(TM) i7-9700，系统内存为 32GB，软件方面为 python 3.7.0，相关的机器学习算法使用的工具来源于 Sklearn 1.0.1，imbalanced-ensemble 0.1.6。

本文采用了两个数据集进行实验分析，一个是本文收集的包含 3483 条浏览器指纹的数据集，其中每条包括 screenResolution，userAgent，IP，fonts 等 73 种特征，本文仅提取 cookie 和只有唯一值的特征。Cookie 作为认证访问浏览器的凭证，如果某个特征所有指纹都相同，则它在指纹识别中没有意义，因此本文剔除唯一特征不用。另一个是文献[12]公开的部分数据，其数据来源于 2015 年 7 月至 2017 年 8 月，利用 AmIUnique 浏览器插件收集的 15000 条指纹，其中每条指纹包括 ID，IP，creationDate 等 40 种特征。考虑到 Flash 技术已经被现代浏览器所淘汰，因此文本剔除了 Flash 相关的特征和所有指纹都相同的特征。

在浏览器指纹识别的类不平衡问题上，准确率并不能准确地反映样分类器的性能。因为负面样本太多，即便是默认所有的样本全是负样本，也能达到 99%以上的正确率。本文选取了 F1-score，G-mean，MCC 等指标作为衡量分类效果的标准，相应的计算公式如式(10)－式(12)所示。这些衡量标准更能综合考虑分类器对不同类的分类效果，更能体现分类器在类不均衡样本上的真实性能。其中衡量标准 F1-score 是基于查准率和查全率的调和平均，它表示在指纹识别中，正负样本错误分类的代价是相同的。而 G-mean 表示分类器查准率和查全率的几何平均。和 G-mean 相比，F1-score 更重视较小值。MCC 为马修斯相关系数，是一个比较均衡的指标，本质上描述了预测结果和实际结果之间的相关系数，当一个分类的 F1-score 与 MCC 差距较大时，意味着单一指标无法

衡量分类器的所有优缺点。

$$F1\text{-}score=2\cdot\frac{Recall\times Precision}{Recall+Precision} \tag{10}$$

$$G\text{-}mean=\sqrt{Recall\times Precision} \tag{11}$$

$$MCC=\frac{TP\times TN-FP\times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \tag{12}$$

5.2 改进效果

本小节实验测试 ISPE 的改进效果。使用所收集的浏览器指纹样本，进行两两配对，源自相同浏览器的指纹对为正样本，源自不同浏览器的指纹对为负样本，生成一个{0:427326，1:2431}的样本集，然后采用原始 SPE 算法，在确定基础分类器权重时，同样在原始 SPE 算法中为基础分类器增加权重。令前$\frac{n}{2}$个基础分类器的权重为 1，调整后$\frac{n}{2}$个分类器的权重来改变其相对权重。以下所有实验的基础分类器均为 CART 决策树算法，最大深度为 10，集成学习的基础分类器个数为 20。最终显示后$\frac{n}{2}$个分类器的权重为 0.5 时效果最佳，如图 6 所示。

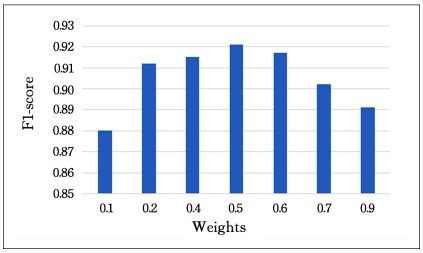

图 6 不同权重对训练效果的影响

Fig. 6 Effect of different weights on training effect

最终，对原本的 SPE 算法、改进了 α 因子的算法、只添加了类注意力机制和只改进了基础分类器权重的方法，以及全部改进的 SPE 方法进行了比较，结果如表 3 所列。

表 3 不同改进对分类效果的影响

Table 3 Impact of different improvements on classification effects

Standard	Original-SPE	α	attention-like mechanism	Weights	ISPE
F1-score	0.876	0.917	0.912	0.911	0.927
G-mean	0.876	0.918	0.911	0.915	0.927
MCC	0.876	0.917	0.912	0.911	0.927

实验结果表明，本文的改进针对浏览器指纹匹配的应用，使用多种综合性衡量标准，均取得了比原始 SPE 算法更好的成绩。相较于原始方法，本文的改进针对浏览器指纹识别的训练应用过程，能更好地反应样本特征，提高识别效果。

5.3 分类性能比较

5.3.1 不平衡算法的比较

为了对比 ISPE 算法与其他处理类不平衡算法的性能，本文对多种算法、方法进行了比较，包括：1）随机森林算法[20]，经常用于类不平衡的分类任务；2）基于 AdaBoost[21] 的集成学习加权方法，主要通过迭代更新权重，使得结果更加符合数据特征权重，提升少数类识别的能力；3）SMOTEBoost[22] 算法采用了数据生成算法 SMOTE 结合 Boost，从少数类创建数据，间接改变权重，提升对少数类分类的能力。

（1）最优序列。本文以 cookie 作为身份认证，按照访问的顺序，依次和已经存储的浏览器指纹信息的 cookie 进行比对，直到相同或者比对完所有记录，其过程类似于 3.1 节。把比对的两组指纹信息按照之前的特征方式生成指纹对信息，如果相同为正样本并更新指纹，反之为负样本。如果比对结束没有相同的，则插入新的指纹信息。如此生成的浏览器指纹对信息序列，为理想状态下每次匹配都判断正确的浏览器指纹匹配过程，被称作最优序列。

本文把其中的前 60%作为训练集，后 40%作为测试集，获得的匹配结果如表 4 所列。

表 4 收集数据集的最优序列结果

Table 4 Optimal sequence results of collected data sets

Standard	Random-Forest	Decision-Tree	Ada-Boost	SMOTE-Boost	ISPE
F1-score	0.709	0.648	0.615	0.355	0.921
G-mean	0.710	0.649	0.630	0.381	0.922
MCC	0.709	0.648	0.630	0.380	0.922

可以看到，在最优序列下，本文采用的方法对浏览器指纹对的 F1-score，G-Mean 和 MCC 等衡量标准均超过了 92%。相较于前几种方法，本文方法更好地缓解了数据不平衡带来的负面影响，高度拟合了指纹数据，提升了综合识别性能。

（2）真实序列。为了接近真实的浏览器指纹处理方式，本文采用所收集的指纹的前 60%两两组成指纹对作为训练集，把剩下的指纹序列依次和指纹记录生成指纹对进行比较。同样类似于 3.1 节，如果相同则更新为新的指纹信息，如果匹配所有的都不同则插入新指纹信息，这种方式更能模拟算法在真实环境下的性能。

本文所收集指纹的真实序列结果如表 5 所列。

表 5 收集数据集的真实序列结果

Table 5 Real sequence results of collected data sets

Standard	Random-Forest	Decision-Tree	Ada-Boost	SMOTE-Boost	ISPE
F1-score	0.631	0.544	0.585	0.306	0.815
G-mean	0.633	0.550	0.602	0.333	0.816
MCC	0.632	0.546	0.594	0.467	0.815

在真实序列的情况下，分类器性能普遍下降，主要原因是前期的错误匹配可能会生成新的浏览器指纹序列，这增加了匹配的次数，导致准确率有所下降。但本文方法与对比方法相比依旧具有一定的优越性。

在公开数据集上，按照上述方法生成最佳序列和真实序列的匹配结果分别如表 6、表 7 所列。

表 6 公开数据集上的最优序列结果

Table 6 Optimal sequence results on public data sets

Standard	Random-Forest	Decision-Tree	Ada-Boost	SMOTE-Boost	ISPE
F1-score	0.847	0.761	0.792	0.845	0.956
G-mean	0.847	0.790	0.793	0.845	0.956
MCC	0.847	0.761	0.793	0.845	0.956

表 7 公开数据集上的真实序列结果

Table 7 Real sequence results on public data sets

Standard	Random-Forest	Decision-Tree	Ada-Boost	SMOTE-Boost	ISPE
F1-score	0.797	0.678	0.780	0.806	0.855
G-mean	0.797	0.679	0.780	0.807	0.855
MCC	0.786	0.678	0.780	0.806	0.854

无论是在公开数据集还是本文自主收集的数据集上，本文方法相比其他方法均取得了更高的识别效果，体现了其对浏览器指纹的优越性。

相比本文自主收集的浏览器指纹数据，本文方法在公开数据集上的表现更好，原因在于公开数据集中的指纹记录质量更高，有大量的浏览器实体存在超过 6 条记录。而本文的数据受限于数量的限制，存在很多只有一两条指纹的浏览器实体记录。

5.3.2 前人研究成果比较

致力于采用浏览器指纹对用户进行长期追踪的研究相对较少，本文选取效果相对较好的 FP-STALKER[12] 和 Bi-RNN[14] 方法进行比较。其中 FP-STALKER 采取的是规则加决策树的混合方法，而 Bi-RNN 则是双向循环神经网络。本文在更具代表性的公开数据集上分别测试了 3 种方法，结果如表 8、表 9 所列。

表 8 公开数据集 STALKER，Bi-RNN 和 SPE 上的最优序列结果

Table 8 Optimal sequence results on public data sets STALKER, Bi-RNN and ISPE

衡量标准	STALKER	Bi-RNN	ISPE
F1-score	0.777	0.788	0.956
G-mean	0.791	0.800	0.956
MCC	0.777	0.788	0.956

表 9 公开数据集 STALKER，Bi-RNN 和 ISPE 上的真实序列结果

Table 9 Real sequence results on public data sets STALKER, Bi-RNN and ISPE

衡量标准	STALKER	Bi-RNN	ISPE
F1-score	0.695	0.722	0.855
G-mean	0.694	0.735	0.855
MCC	0.695	0.723	0.854

相较于 FP-STALKER 和 Bi-RNN 两种方法，本文方法对浏览器指纹的识别效果更好，在最优序列上 F1-score 分别提升了 17.9%和 16.8%，在真实序列上分别提升了 16.0%和 13.3%。证明了本文方法用于浏览器指纹长期追踪能够有效提升指纹识别准确度，延长用户长期追踪时间。

结束语 尽管浏览器指纹的应用在许多领域已经初具规模，但是其在长期追踪方面仍然有更多的潜力可以开发。本文在前人研究的基础上，针对浏览器指纹长期追踪进行识别匹配所面临的在训练和实际应用过程的正负样本不平衡问题，提出了采用 ISPE 的方法对数据结合算法进行采样处理，采用决策树＋集成学习，提升了浏览器指纹对的匹配准确度，增强了浏览器指纹对用户的长期追踪能力。利用所收集的浏览器指纹数据验证了本文方法的有效性，并同时在公开数据集上和多种方法进行了对比，证明了其相较于已有方法，在处理浏览器指纹在类不平衡情况下的识别问题上，综合识别性能有显著提升。

实验结果表明本文方法对指纹识别具有较高的准确性，但是用于匹配的指纹数据中指纹的特征种类和特性都比较有限。后续计划采用更多、更新、更高熵的浏览器指纹特征作为收集数据的一部分，并且增加不同网站分布式收集的浏览器指纹信息来提高对唯一访问实体的追踪效果。

参 考 文 献

[1] Cookie Policy - Intellias[EB/OL]. [2021-12-28]. https://intellias.com/cookie-policy/.

[2] Cookies: An overview of associated privacy and security risks-Infosec Resources[EB/OL]. [2021-12-28]. https://resources.infosecinstitute. com/topic/cookies-an-overview-of-associated-privacy-and-security-risks/.

[3] YEN T F, XIE Y, YU F, et al. Host Fingerprinting and Tracking on the Web: Privacy and Security Implications[C]// 19th Annual Network and Distributed System Security Symposium, NDSS 2012. San Diego, California, USA, 2012.

[4] ECKERSLEY P. How Unique Is Your Web Browser? [C]// Proceedings of the 10th International Conference on Privacy Enhancing Technologie. Berlin, Germany, 2010: 1-18.

[5] TRICKEL E, STAROV O, KAPRAVELOS A, et al. Everyone is Different: Client-side Diversification for Defending Against Extension Fingerprinting[C]// 28th USENIX Security Symposium (USENIX Security 19). Santa Clara, CA: USENIX Association, 2019: 1679-1696.

[6] WU S, LI S, CAO Y, et al. Rendered Private: Making GLSL Execution Uniform to Prevent WebGL-based Browser Fingerprin-ting[C]// 28th USENIX Security Symposium (USENIX Security 19). Santa Clara, CA: USENIX Association, 2019: 1645-1660.

[7] CAO Y, LI S, WIJMANS E. (Cross-)Browser Fingerprinting via OS and Hardware Level Features[C]// 24th Annual Network and Distributed System Security Symposium, NDSS 2017. San Diego, California, USA, 2017.

[8] TAO X M, HAO S Y, ZHANG D X, et al. A Review of Imbalanced Data Classification Algorithms[J]. Journal of Chongqing University of Posts and Telecommunications: Natural Science Edition, 2013, 25: 1-11.

[9] LIU Z, CAO W, GAO Z, et al. Self-paced Ensemble for Highly Imbalanced Massive Data Classification[C]// 36th IEEE International Conference on Data Engineering (ICDE 2020). Dallas, TX, USA: IEEE, 2020: 841-852.

[10] MUFIOZ-GARCIA Ó, MONTERRUBIO-MARTIN J, GARCIA-AUBERT D. Detecting browser fingerprint evolution for identifying unique users[J]. International Journal of Electronic Business, 2012, 10(2): 120-141.

[11] YAMADA T, SAITO T, TAKASU K, et al. Robust Identification of Browser Fingerprint Comparison Using Edit Distance [C]// 10th International Conference on Broadband and Wireless Computing, Communication and Applications, BWCCA 2015. Krakow, Poland: IEEE Computer Society, 2015: 107-113.

[12] VASTEL A, LAPERDRIX P, RUDAMETKIN W, et al. FP-STALKER: Tracking Browser Fingerprint Evolutions[C]// 2018 IEEE Symposium on Security and Privacy. San Francisco, California, USA: IEEE Computer Society, 2018: 728-741.

[13] LI X, CUI X, SHI L, et al. Constructing Browser Fingerprint Tracking Chain Based on LSTM Model[C]// Third IEEE International Conference on Data Science in Cyberspace(DSC 2018). Guangzhou, China: IEEE, 2018: 213-218.

[14] LIU Q X, LIU X Y, LUO C, et al. Android Browser Fingerprinting Method Based on Bidirectional Recurrent Neural Network [J]. Journal of Computer Research and Development, 2020, 57: 2294.

[15] NAKIBLY G, SHELEF G, YUDILEVICH S. Hardware Fingerprinting Using HTML5[J]. arXiv: 1503.01408, 2015.

[16] MOWERY K, SHACHAM H. Pixel perfect: Fingerprinting canvas in HTML5[C]// Proceedings of W2SP. 2012: 1-12.

[17] LAPERDRIX P, RUDAMETKIN W, BAUDRY B. Beauty and the Beast: Diverting Modern Web Browsers to Build Unique Browser Fingerprints[C]// 2016 IEEE Symposium on Security and Privacy(SP). 2016: 878-894.

[18] GitHub-fingerprintjs/fingerprintjs: Browser fingerprinting library with the highest accuracy and stability[EB/OL]. [2021-12-29]. https://github.com/fingerprintjs/fingerprintjs.

[19] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is All You Need[J/OL]. Advances in Neural Information Processing Systems, 2017, 2017: 5999-6009. https://arxiv.org/abs/1706.03762v5.

[20] BREIMAN L. Random Forests [J]. Machine Learning, 2001, 45(1): 5-32.

[21] KARAKOULAS G, SHAWE-TAYLOR J. Optimizing classifers for imbalanced training sets[C]// Advances in Neural Information Processing Systems. 1998.

[22] CHAWLA N V, LAZAREVIC A, HALL L O, et al. SMOTEBoost: Improving Prediction of the Minority Class in Boosting [C]// Knowledge Discovery in Databases: PKDD 2003, 7th European Conference on Principles and Practice of Knowledge Discovery in Databases. Cavtat-Dubrovnik, Croatia: Springer, 2003: 107-119.

ZHANG Desheng, born in 1997, postgraduate. His main research interests include cyberspace security and so on.

ZHANG Jianhui, born in 1977, Ph.D, associate researcher, master supervisor. His main research interests include new network architecture, network routing technology, network data analysis and security control.

（责任编辑：杨雪敏）