基于多分类 LSTM 的浏览器指纹识别方法

李建伏　宋国平
（中国民航大学计算机科学与技术学院　天津 300300）

摘　要　现有基于机器学习的方法将浏览器指纹的用户识别处理成二分类问题，但该处理方式信息损失较多且识别效率低下。为解决上述问题，提出基于多分类长短期记忆网络（Long Short-Term Memory，LSTM）的浏览器指纹识别方法。其基本思路是将同一用户的浏览器指纹数据处理成时间序列，利用多分类 LSTM 模型对其进行分类，从而实现用户识别。实验结果表明，该方法比基于二分类的指纹识别方法有更高的准确率和更快的识别速度。

关键词　浏览器指纹　LSTM　用户识别

中图分类号　TP3　　**文献标志码**　A　　DOI：10.3969/j.issn.1000-386x.2024.04.026

BROWSER FINGERPRINT RECOGNITION METHOD BASED ON MULTI-CLASS LSTM NETWORK

Li Jianfu　Song Guoping
（*College of Computer Science and Technology*, *Civil Aviation University of China*, *Tianjin 300300*, *China*）

Abstract　The existing methods based on machine learning process the user recognition of browser fingerprint into a binary classification problem, but they have more information loss and low recognition efficiency. In order to solve the above problems, this paper proposes a browser fingerprinting recognition method based on multi-class LSTM. The basic idea of this method was to process the same user's browser fingerprint data into time series, and it used the multi-class LSTM model to classify them, so as to achieve user recognition. The experimental results show that the proposed method has higher accuracy and faster recognition speed than the fingerprinting recognition method based on binary classification.

Keywords　Browser fingerprint　LSTM　User recognition

0　引　言

当今社会，互联网给人们的生活带来了诸多便利，使得人们的工作效率大大提高。同时，网络的匿名性也给网络安全和网络服务带来了新的挑战，准确地识别在线用户已经成为互联网应用领域的重要问题。

当前在线用户识别方法主要有三类：基于 IP 地址的在线用户识别方法、基于 Cookies 的在线用户识别方法以及基于浏览器指纹的用户识别方法。基于 IP 地址的在线用户识别方法通过 IP 地址确定用户身份，但是当用户使用动态 IP 地址或者校园网等公共网络环境时，由于 IP 地址的多变性或者重复性，该方法将无法正常识别在线用户。基于 Cookies 的在线用户识别方法通过存储在用户客户端的数据识别用户。但是随着人们隐私意识的提高，越来越多的人在访问网站时选择隐私模式或者在结束访问时主动删除 Cookies[1]，所以基于 Cookies 的在线用户识别方法受到很大的限制。由于浏览器指纹信息的获取无须用户的参与且不需要在客户端存储信息，基于浏览器指纹的用户识别方法具有更好的应用前景。

基于浏览器指纹的用户识别方法的基本思路是利用浏览器所收集的当前访问设备的软硬件配置信息的组合来区分用户，如浏览器类型版本、屏幕分辨率、时

收稿日期：2021－01－28。国家重点研发计划项目（2020YFB1600101）；天津市教委科研计划项目（2020KJ024）。**李建伏**，副教授，主研领域：机器学习，数据挖掘。**宋国平**，硕士生。

区、浏览器插件和系统字体[2]等。基于浏览器指纹的用户识别技术的研究最早出现于2010年[3],该研究表明仅仅由8个设备信息组成的浏览器指纹就可以识别PANOPTICLICK网站的83.6%的访问者,从而验证了基于浏览器指纹的用户识别技术的有效性。2020年Pugliese等[4]再次验证了基于浏览器指纹的用户识别技术的有效性。

早期对于浏览器指纹的研究工作主要集中于发掘出更多的设备特征参数以区分不同的用户[5]。例如,文献[6-7]提出使用canvas API和canvas信息作为浏览器指纹,文献[8-9]研究使用浏览器扩展项信息作为浏览器指纹,文献[10]提出使用Audio API作为浏览器指纹。

实际上,不同于人类指纹的静态不变性,浏览器指纹会随着计算机系统组件、环境等的变化而不断地发生改变,如浏览器版本升级、浏览器插件升级或更新、关闭Cookies和安装新的字体等。除此之外,浏览器指纹防御系统[11]也会改变浏览器指纹的属性特征,例如,用户使用FP-Block[12]工具微调浏览器指纹、使用洋葱浏览器删除插件修改指纹属性值返回相同属性值使自己的浏览器指纹趋于标准化[11]等。为了利用不断变化的指纹信息识别用户,需要对浏览器指纹变化规律进行建模。现有的基于浏览器指纹变化规律的浏览器指纹识别方法大致可以分为基于统计分析的方法和基于机器学习的方法。

(1)基于统计分析的方法。基于统计分析的方法的基本思路是通过对历史数据进行统计分析得到浏览器指纹特征信息的变化规律。Vastel等[11]通过统计分析得到了浏览器指纹特征信息随时间变化的7条规则,通过该7条静态规则识别用户。张良峰等[13]提出了采用统计和侧信道攻击的方法对浏览器安装的插件和字体属性进行多次采样,观察所得的浏览器指纹关键属性的随机值,找到被修改属性值的变化范围,根据变化属性的中间值还原出浏览器指纹中特征属性的真实值,从而达到区分和跟踪用户的目的。Liu等[14]通过引入次要属性根据指纹各属性的重要性、变更难度和频率对其各属性进行评估并赋予权重,然后通过计算两枚指纹的相似度来识别用户。Antonio等[15]通过计算属性的熵来评估其变化程度。

(2)基于机器学习方法。现有的基于机器学习的浏览器指纹识别技术通常将浏览器指纹的用户识别问题看作一个二分类问题。即采用成对分类的思路构造训练集,即一一对比两枚指纹信息,如果两枚指纹对应特征的取值相同,则该训练样本的该特征取值为0,否则为1。如果这两枚指纹是同一个用户的,则将该训练样本的标签设置为1,否则设置为0。然后,利用某种监督学习模型根据训练集建模指纹特征随时间的变化规律。基于以上思路,Vastel等[11]于2018年提出了一种基于随机森林的浏览器指纹识别方法。为了更好地显示样本随时间的变化规律,Li等[16]在文献[11]的基础上提出了基于长短期记忆网络(LSTM)的浏览器指纹识别技术,即通过对比相邻时间段的指纹特征信息构造指纹变化序列,然后利用LSTM模型建模指纹信息随时间变化规律。刘奇旭等[17]提出了一种基于双向循环神经网络(Bi-directional Recurrent Neural Network,Bi-RNN)的安卓浏览器识别方法。

现有的基于二分类的浏览器指纹识别方法有以下两个缺点:

(1)在按照成对分类的方式根据原始数据构建训练集时,对于两枚指纹,只知道某些特征取值是否相等,而丢失了原始数据的特征信息。

(2)当测试一枚新指纹时,需要将新指纹与已有的指纹逐个进行对比、构造测试样本、利用训练好的模型对测试样本进行预测,直至得到了预测结果为"1"的测试样本,即找到了目标用户。可见,基于二分类模式的浏览器指纹识别方法的识别时间与指纹库中指纹的数量成正比。因此,基于二分类模式的浏览器指纹识别方法的识别效率低下,尤其是当指纹库规模比较大时。

为了解决以上两个问题,本文提出一种基于多分类LSTM的浏览器指纹识别方法,其基本思想是将指纹识别问题看成多分类问题,将同一用户的浏览器指纹看作一类。理论上,新的算法直接对浏览器指纹数据进行处理,而无须进行分类对比转换,从而避免了信息损失;并且测试新样本时,通过模型一次计算就能得到结果,避免了传统的二分类指纹识别中多次比对、计算的问题,从而提高了识别效率。

需要指出的是,本文提出的基于多分类LSTM的浏览器指纹识别方法不仅限于基本的LSTM模型,其他改进的LSTM模型也同样适用。本文以基本的LSTM模型为例,说明这种以多分类的方式处理基于指纹浏览器的识别方法是有效的。

1 LSTM模型

循环神经网络(Recurrent Neural Network,RNN)是一种前馈型的神经网络,适用于序列数据建模,在自然语言处理、时间序列预测等领域取得了瞩目的效果,成

为深度学习领域中一个重要的模型。在 RNN 中,神经元在某时刻的输出又作为下一个时刻神经元的部分输入,这样的递归结构可以保持数据之间的依赖关系。然而,在训练过程中,多次递归以后会出现梯度消失或者梯度爆炸问题,导致 RNN 无法捕获序列中的长期依赖关系。

为了解决梯度消失或梯度爆炸问题,目前已经针对 RNN 提出了多种改进算法,其中 LSTM 应用最广泛。LSTM 和 RNN 具有相同的链式结构,不同之处在于,LSTM 在 RNN 的基本循环单元中增加了输入门、遗忘门和记忆单元。图 1 给出了 LSTM 存储器单元的内部结构。LSTM 中各状态更新公式如式(1) – 式(6)所示。

$$\boldsymbol{i}_t=\delta(\boldsymbol{W}_i\cdot[\boldsymbol{h}_{t-1},\boldsymbol{x}_t]+\boldsymbol{b}_i) \tag{1}$$

$$\boldsymbol{f}_t=\delta(\boldsymbol{W}_f\cdot[\boldsymbol{h}_{t-1},\boldsymbol{x}_t]+\boldsymbol{b}_f) \tag{2}$$

$$\boldsymbol{o}_t=\delta(\boldsymbol{W}_o\cdot[\boldsymbol{h}_{t-1},\boldsymbol{x}_t]+\boldsymbol{b}_o) \tag{3}$$

$$\overline{\boldsymbol{c}}_t=\tanh(\boldsymbol{W}_c\cdot[\boldsymbol{h}_{t-1},\boldsymbol{x}_t]+\boldsymbol{b}_c) \tag{4}$$

$$\boldsymbol{c}_t=\boldsymbol{f}_t\odot\boldsymbol{c}_{t-1}+\boldsymbol{i}_t\odot\overline{\boldsymbol{c}}_t \tag{5}$$

$$\boldsymbol{h}_t=\boldsymbol{o}_t\odot\tanh(\boldsymbol{c}_t) \tag{6}$$

式中:$\boldsymbol{i}_t$、$\boldsymbol{f}_t$ 和 $\boldsymbol{o}_t$ 分别表示输入门、遗忘门和输出门 t 时刻的值;$\overline{\boldsymbol{c}}_t$ 和 $\boldsymbol{c}_t$ 分别表示 t 时刻记忆单元的中间量和状态量;$\boldsymbol{W}_i$、$\boldsymbol{W}_f$、$\boldsymbol{W}_o$ 和 $\boldsymbol{W}_c$ 分别表示不同状态单元对应的权重矩阵;$\boldsymbol{b}_i$、$\boldsymbol{b}_f$、$\boldsymbol{b}_o$ 和 $\boldsymbol{b}_c$ 分别表示不同状态单元对应的偏置矩阵;$\boldsymbol{h}_t$ 表示 t 时刻隐藏层的输出;⊙表示对应元素相乘;δ 表示 sigmoid 激励函数。

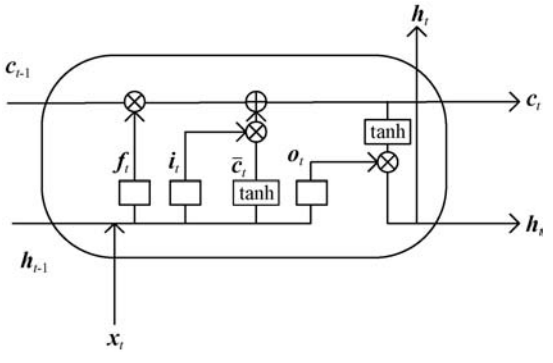

图 1　LSTM 存储器单元结构

一般认为,最后一个循环单元的隐藏层的输出 $\boldsymbol{h}_t$ 包含了整个序列中的信息。

2　基于多分类 LSTM 的浏览器指纹识别

本文提出的基于多分类 LSTM 的浏览器指纹识别方法的基本思路是将浏览器指纹识别问题看成一个多分类问题,将同一用户的浏览器指纹看作一类,其基本流程是把现有的指纹数据按照时间顺序排序得到指纹序列,然后利用这些指纹序列训练多分类 LSTM 网络模型,最后通过训练好的多分类 LSTM 网络模型对浏览器指纹进行分类、识别。

浏览器指纹识别流程如图 2 所示。

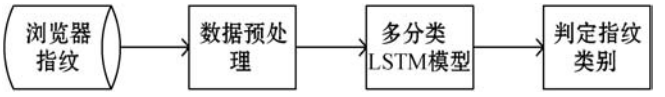

图 2　浏览器指纹识别流程

2.1　数据预处理

(1) 数据编码。为了能够将浏览器指纹数据输入到多分类 LSTM 模型,需要将浏览器指纹信息数值化。浏览器指纹的设备特征取值主要分为三类,分别是数值型、布尔类型、字符型。根据不同类型的数据特点,分别对其进行了不同形式的编码操作。

数值型的数据不需要编码,仅进行归一化处理;布尔类型的数据只有“Yes”和“No”两种取值,因此通过将“Yes”转换成 1、“No”转换成 0 而把布尔类型的数据转换成二进制数据;对于字符型的数据,采样 one-hot 编码的方式将其转换成数值型数据。

(2) 构建指纹序列。为了获得浏览器指纹序列,将现有的属于同一个浏览器的指纹数据按时间进行排序,按照预定的序列长度 k 将数据处理成一个个长度为 k 的指纹序列,并以当前指纹序列所属的浏览器编号作为其类别标签,这样就形成了多分类 LSTM 模型的一个训练样本。

2.2　多分类 LSTM 模型

LSTM 模型结构总共分为三层,分别是 LSTM 层、全连接层、输出层,如图 3 所示。

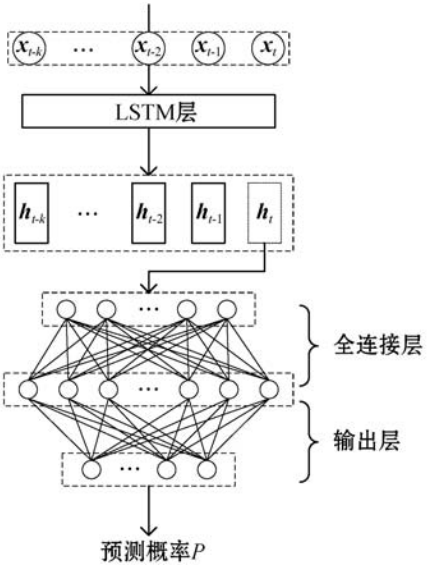

图 3　多分类 LSTM 网络结构

LSTM 层中循环单元的个数取决于所要处理的序列的长度,每个循环单元的输入层中神经元的个数取决于每个浏览器指纹编码的维度。在 LSTM 中,最后

一个循环单元的输出保留了整个浏览器指纹的变化过程,因此,将最后一个隐藏单元的输出 $\boldsymbol{h}_t$ 作为 LSTM 层的输出特征向量。

全连接层以特征向量 $\boldsymbol{h}_t$ 作为该层的输入向量,该层神经元的数目决定了全连接层的输出维度,本文实验中该层神经元的激活函数为 ReLU。

输出层神经元的个数等于类别的数目 n,即样本中不同用户的数目,激活函数为 Softmax。该层输出一个长度为 n 的向量 $\boldsymbol{P}=\{p_1,p_2,\cdots,p_n\}$,其中每个 p_i 表示当前样本属于类别 i 的概率,且有 $\sum_{i=1}^{n}p_i=1$。

在模型训练过程中使用交叉熵函数作为损失函数,如式(7)所示。

$$L=-\frac{1}{m}\sum_{i=1}^{m}y_i\log P_{(i)} \tag{7}$$

式中:m 表示数据集样本数量;y_i 表示第 i 个样本真实标签的概率分布;$P_{(i)}$ 表示 LSTM 模型预测得到的第 i 个样本所属类别的概率分布。该损失函数用于衡量网络模型输出的概率分布和真实分布两个概率分布之间的距离。通过训练网络模型最小化这两个分布的距离,可使输出结果尽可能接近真实标签。同时使用随机梯度下降法对参数进行更新。在多分类 LSTM 网络中采用 Dropout 来防止过度拟合。

2.3 指纹的判定

对于一个待识别的指纹序列,将其输入到训练好的多分类 LSTM 模型后,得到输出向量 $\boldsymbol{P}=\{p_1,p_2,\cdots,p_n\}$,然后根据 $\boldsymbol{P}$ 向量的 n 个概率的分布情况来判断当前该指纹序列是属于某个已有用户的还是属于新用户的。

一般认为,$\boldsymbol{P}$ 向量的 n 个概率的分布越平均,则待识别指纹序列所属类别的指向性就越不明显;反之,如果 n 个概率的分布越不平均,则待识别指纹序列所属类别的指向性就越明显。基于上述观察,从 $\boldsymbol{P}$ 向量的 n 个概率中选择最高者 $p_{\max}$,如果当预测值 $p_{\max}$ 小于某一预先指定的阈值 θ,则将该指纹序列判定为新类指纹序列,并将新类指纹序列加入到指纹数据集中;否则,判定该指纹序列为已有用户的浏览器指纹序列,并返回其标签作为该指纹序列的所属类别。

3 实 验

为了验证本文所提出的基于多分类 LSTM 的浏览器指纹识别方法的有效性,将其与基于二分类 LSTM 的浏览器指纹识别方法[16]和基于二分类 Bi-RNN 的浏览器指纹识别方法[17]进行了实验对比。

3.1 实验环境

所有实验均采用戴尔 Inspiron3437 作为实验机器,CPU 为 Intel® Core™ i5-4200U@1.60 GHz 4 核,GPU 为 NVIDIA GeForce 720M,机器内存为 4 GB。实验中所有网络模型都是基于 Keras 深度学习平台,后端为 TensorFlow 框架。

在本文所提出的多分类 LSTM 模型中,将指纹判定阈值 θ 设定在 0.92。

3.2 数据集

本文以文献[11]中给出的实验数据为基础测试数据。但是,由于该数据集中浏览器指纹的变化规律都符合其给出的 7 条静态规则,而浏览器指纹防御系统可以通过修改浏览器指纹某些原始特征来对抗浏览器指纹追踪。为了模拟浏览器指纹的动态变化以及浏览器指纹防御技术对浏览器指纹修改的情况,依据其给出的 7 条浏览器指纹的变化规则对文献[11]中给出的测试数据进行了数据增强。

数据增强过程如下:获取指纹库每个属性的属性值,将各属性值存入集合,使用集合中的属性值随机替代原有的属性值,通过更改部分属性的属性值,实现数据的增强。通过数据集增强将原始数据增加至 18 200 条。

实验选取了 14 个浏览器指纹属性参与实验,分别是 accept、browserFamily、canvas、dnt、encoding、fonts、language、os、platform、plugins、WebGL、resolution、timezone、userAgent。从数据集中选取长度超过 7 个浏览器指纹的数据进行实验。

指纹序列的构建过程:将来自同一个浏览器的指纹,按时间顺序进行排序,然后将指纹处理成时间步长为 5 的指纹序列,例如对于浏览器 A,处理成长度为 5 的指纹序列 $\{f_{A(t0)},f_{A(t1)},f_{A(t2)},f_{A(t3)},f_{A(t4)}\}$,$\{f_{A(t1)},f_{A(t2)},f_{A(t3)},f_{A(t4)},f_{A(t5)}\}$,$\cdots$,$\{f_{A(tn-4)},f_{A(tn-3)},f_{A(tn-2)},f_{A(tn-1)},f_{A(tn)}\}$。

为了验证模型对于新类指纹与演变指纹的识别,取 70% 的指纹作训练集,30% 的指纹作测试集,模拟真实的浏览器指纹识别过程。

3.3 评价指标

本文从识别速度与准确度两个方面对三种浏览器指纹识别方法进行对比。采用了准确率(Accuracy)和加权平均 F1 值(weighted avg-F1)两个评价指标,其中:准确率是分类正确的样本数占样本总数的比例;加权平均 F1 值是对每一类别的 F1 值进行加权平均,权

重为各类别样本在总样本中所占比例。准确率和加权平均 F1 值越大,表明模型性能越好。准确率和加权平均 F1 值公式如下:

$$A_{\text{ccuracy}} = \frac{t_{\text{p}} + t_{\text{n}}}{t_{\text{p}} + f_{\text{p}} + t_{\text{n}} + f_{\text{n}}} \tag{8}$$

$$w_{\text{eighted avg-F1}} = \sum_{i=1}^{m} w_i F_i \tag{9}$$

式中:t_{p} 表示正样本被分类为正样本的数量;f_{p} 表示负样本被分类为正样本的数量;t_{n} 表示正样本被分类为负样本的数量;f_{n} 表示负样本被分类成负样本的数量;m 表示数据集中样本总类别数量;w_i 表示类别 i 样本数量在总样本中占比;F_i 类别 i 表示的 F1 值[18]。

3.4　实验结果及分析

3.4.1　速度比较

三种浏览器指纹识别方法时间开销如图 4 所示,其中:横轴表示测试样本的数量,纵轴表示测试需要的时间开销。

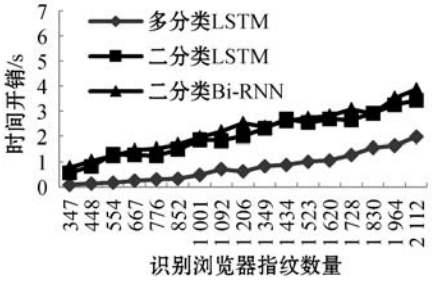

图 4　多分类 LSTM、二分类 LSTM 和二分类 Bi-RNN 模型时间开销

通过图 4 可以得到:

(1) 随着识别指纹数量的增多,三种方法需要的时间都越来越长。

(2) 基于二分类 Bi-RNN 和基于二分类 LSTM 的浏览器指纹识别方法的时间开销相当。这是因为基于二分类模型的浏览器识别方法的时间开销主要取决于指纹库浏览器指纹的数量,在识别过程中指纹库中指纹的数量保持不变,所以导致基于二分类 Bi-RNN 与基于二分类 LSTM 的浏览器指纹识别方法时间开销相当。

(3) 本文提出的基于多分类 LSTM 浏览器指纹识别方法要比基于二分类 Bi-RNN 和基于二分类 LSTM 的浏览器指纹识别方法快得多。这主要是由于两种基于二分类的方法在测试时待识别指纹需要与指纹库中的所有指纹进行一一对比,而使用多分类 LSTM 模型对浏览器指纹进行识别无须对比指纹库的指纹,只需使用训练好的多分类 LSTM 对指纹进行识别即可,导致这两种方法消耗的时间成本比使用基于多分类 LSTM 模型浏览器指纹识别方法消耗的时间成本高。

3.4.2　准确度比较

三种浏览器指纹识别方法准确度如图 5 和图 6 所示。图 5 表示三种模型识别结果的准确率,其中:横轴为训练批次,纵轴为准确率。图 6 表示三种模型识别结果的加权平均 F1 值,其中:横轴为训练批次,纵轴为加权平均 F1 值。

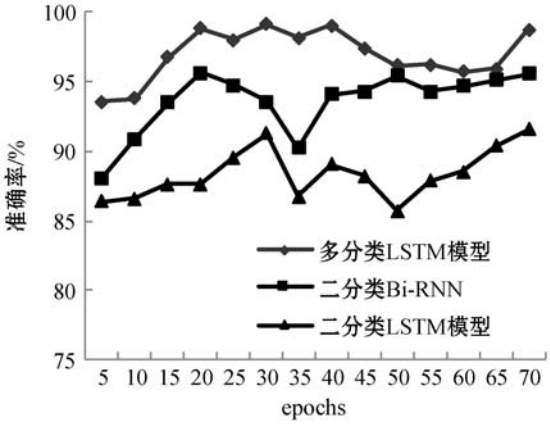

图 5　多分类 LSTM、二分类 LSTM 和 Bi-RNN 模型准确率

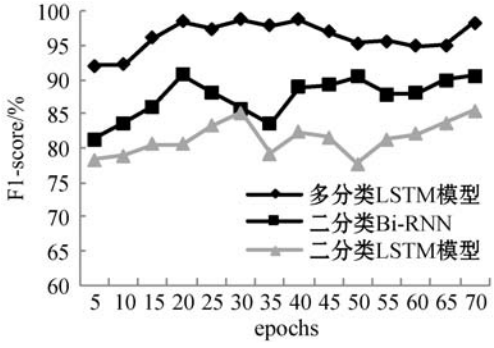

图 6　多分类 LSTM、二分类 LSTM 和 Bi-RNN 模型 F1-score

通过图 5 可以得到以下 2 点:

(1) 使用基于多分类 LSTM 模型的准确率优于其他两个模型,在不同的训练批次上基于多分类 LSTM 模型的准确率均能达到 93.58% 以上,最高甚至可以达到 99.08%。

(2) 从三个模型每个批次的准确率均值来看,多分类 LSTM 模型的准确率均值为 96.95%,二分类 LSTM 模型的准确率均值为 93.57%,二分类的 Bi-RNN 模型的准确率均值为 88.36%,可以得出多分类 LSTM 模型比二分类 LSTM 模型的准确率高 3.38 百分点,比二分类 Bi-RNN 模型的准确率高 8.59 百分点,说明多分类 LSTM 模型比二分类 LSTM 模型和二分类 Bi-RNN 模型更能将浏览器指纹正确分类,达到有效识别用户的目的。

通过图 6 可以得到以下 2 点:

(1) 在不同训练批次中,基于多分类 LSTM 模型的加权平均 F1 值高于二分类 LSTM 模型与基于二分

类 Bi-RNN 模型的加权平均 F1 值,可以说明在浏览器指纹识别方向上,从原始数据集中提取指纹特征作为多分类 LSTM 模型的输入比从指纹对的比较结果获得模型的输入更能保持数据信息的完整性,从而表现出来更加明显的分类优势。

(2) 从三个模型每个批次的加权平均 F1 值的均值来看,基于多分类 LSTM 模型的加权平均 F1 值为 96.27%,而基于二分类 LSTM 模型的加权平均 F1 值为 87.45%,基于二分类 Bi-RNN 模型的加权平均 F1 值 81.4%,可以得出基于多分类 LSTM 模型的加权平均 F1 值比基于二分类 LSTM 模型的加权平均 F1 值高出了 8.82 百分点,比基于二分类 Bi-RNN 模型的加权平均 F1 值高出了 14.87 百分点,说明当以加权平均 F1 值作为评价标准时,基于多分类 LSTM 模型在不同类别指纹上的分类效果同样优于二分类 LSTM 模型与基于二分类 Bi-RNN 模型。

4 结 语

本文针对基于二分类模型的浏览器指纹识别方法中出现的识别效率低、速度慢的问题,将多分类 LSTM 模型应用于浏览器指纹识别,将同一用户使用的浏览器看作一类,使用多分类 LSTM 模型学习浏览器指纹的变化规律,然后将待识别指纹序列输入到训练好的多分类 LSTM 模型中,得到模型的预测概率值,最后使用相似度阈值判断待识别指纹序列所属的类别。实验将多分类 LSTM 模型与二分类 LSTM 模型和 Bi-RNN 模型进行对比,结果表明多分类 LSTM 模型的浏览器指纹识别速度更快且识别效率更高,能够快速准确地识别浏览器指纹,从而实现用户的有效识别。

参 考 文 献

[1] Mayer J R, Mitchell J C. Third-party web tracking: Policy and technology[C]//IEEE Symposium on Security and Privacy,2012:413 - 427.

[2] Saito T, Takahashi K, Yasuda K, et al. OS and application identification by installed fonts[C]//30th International Conference on Advanced Information Networking and Applications,2016:684 - 689.

[3] Eckersley P. How unique is your web browser? [C]//International Symposium on Privacy Enhancing Technologies Symposium,2010:1 - 18.

[4] Pugliese G, Riess C, Gassmann F, et al. Long-term observation on browser fingerprinting: Users' trackability and perspective[J]. Proceedings on Privacy Enhancing Technologies,2020,2020(2):558 - 577.

[5] Laperdrix P, Bielova N, Baudry B, et al. Browser fingerprinting: A survey[J]. ACM Transactions on the Web, 2020,14(2):1 - 33.

[6] Mowery K, Shacham H. Pixel perfect: Fingerprinting canvas in HTML5[C]//Conference on Web 2.0 Security and Privacy,2012:1 - 12.

[7] Laperdrix P, Avoine G, Baudry B, et al. Morellian analysis for browsers: Making web authentication stronger with canvas fingerprinting[C]//International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment, 2019:43 - 66.

[8] Sjosten A, Acker S, Sabelfeld A. Discovering browser extensions via web accessible resources[C]//7th ACM on Conference on Data and Application Security and Privacy, 2017:329 - 336.

[9] Starov O, Laperdrix P, Kapravelos A, et al. Unnecessarily identifiable: Quantifying the finger-printability of browser extensions due to bloat[C]//The World Wide Web Conference,2019:3244 - 3250.

[10] Queiroz J, Feitosa E L. A web browser fingerprinting method based on the web audio API[J]. The Computer Journal, 2019,62(8):1258 - 1269.

[11] Vastel A, Laperdrix P, Rudametkin W, et al. FP-STALKER: Tracking browser fingerprint evolutions[C]//IEEE Symposium on Security and Privacy,2018:728 - 741.

[12] Torres C F, Jonker H, Mauw S. FP-Block: Usable web privacy by controlling browser fingerprinting[C]//20th European Symposium on Research in Computer Security, 2015: 3 - 19.

[13] 张良峰,汪毅,吴源燚,等. 基于统计的浏览器指纹采集技术[J]. 信息网络安全,2019(11):49 - 55.

[14] Liu X F, Liu Q X, Wang X, et al. Fingerprinting web browser for tracing anonymous web attackers[C]//1st International Conference on Data Science in Cyberspace,2016:222 - 229.

[15] Antonio E, Fajardo A, Medina R. Tracking browser fingerprint using rule based algorithm[C]//16th IEEE International Colloquium on Signal Processing & Its Applications, 2020:225 - 229.

[16] Li X Y, Cui X, Shi L M, et al. Constructing browser fingerprint tracking chain based on LSTM Model[C]//3rd International Conference on Data Science in Cyberspace, 2018: 213 - 218.

[17] 刘奇旭,刘心宇,罗成,等. 基于双向循环神经网络的安卓浏览器指纹识别方法[J]. 计算机研究与发展,2020,57(11):2294 - 2311.

[18] 周志华. 机器学习[M]. 北京:清华大学出版社,2016.