软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2016,27(11):2870−2883 [doi: 10.13328/j.cnki.jos.004885]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

基于集成聚类的流量分类架构*

鲁 刚[1], 余翔湛[2], 张宏莉[2], 郭荣华[1]

[1](中国洛阳电子装备试验中心,河南 洛阳 471003)

[2](哈尔滨工业大学 计算机科学与技术学院,黑龙江 哈尔滨 150001)

通讯作者: 鲁刚, E-mail: lgang198202@126.com

摘 要: 流量分类是优化网络服务质量的基础与关键.机器学习算法利用数据流统计特征分类流量,对于识别加密私有协议流量具有重要意义.然而,特征偏置和类别不平衡是基于机器学习的流量分类研究所面临的两大挑战.特征偏置是指一些数据流统计特征在提高部分应用识别准确率的同时也降低了另外一部分应用识别的准确率.类别不平衡是指机器学习流量分类器对样本数较少的应用识别的准确率较低.为解决上述问题,提出了基于集成聚类的流量分类架构(traffic classification framework based on ensemble clustering,简称 TCFEC).TCFEC 由多个基于不同特征子空间聚类的基分类器和一个最优决策部件构成,能够提高流量分类的准确率.具体而言,与传统的机器学习流量分类器相比,TCFEC 的平均流准确率最高提升 5%,字节准确率最高提升 6%.

关键词: 基于集成聚类的流量分类架构;集成聚类;流量分类;数据流特征;机器学习

中图法分类号: TP393

中文引用格式: 鲁刚,余翔湛,张宏莉,郭荣华.基于集成聚类的流量分类架构.软件学报,2016,27(11):2870−2883. http://www.jos.org.cn/1000-9825/4885.htm

英文引用格式: Lu G, Yu XZ, Zhang HL, Guo RH. Traffic classification framework based on ensemble clustering. Ruan Jian Xue Bao/Journal of Software, 2016,27(11):2870−2883 (in Chinese). http://www.jos.org.cn/1000-9825/4885.htm

Traffic Classification Framework Based on Ensemble Clustering

LU Gang[1], YU Xiang-Zhan[2], ZHANG Hong-Li[2], GUO Rong-Hua[1]

[1](Chinese Luoyang Electronic Equipment Center, Luoyang 471003, China)

[2](School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China)

Abstract: Traffic classification is the basis and key for optimizing network quality of service. Machine learning algorithms apply flow statistics in traffic classification, which are significant for identifying both encrypted and private traffic. However, the discriminator bias problem and the class imbalance problem are two main challenges in traffic classification. The discriminator bias problem denotes that some flow statistics can improve the accuracies for some applications but reduce the accuracies for other applications. The class imbalance problem denotes that machine learning based traffic classifier identifies the minority application with a low accuracy. To address the above two issues, traffic classification framework based on ensemble clustering (TCFEC) is proposed in this paper. TCFEC is composed of several base classifiers trained by clustering in different feature subspaces and an optimal decision component. It is able to improve accuracy in traffic classification. Specifically, compared with the traffic classifier based on traditional machine learning algorithms, TCFEC improves average flow accuracy by 5% as well as average byte accuracy by 6%.

Key words: traffic classification framework based on ensemble clustering (TCFEC); ensemble clustering; traffic classification; flow-based feature; machine learning

流量分类对于优化网络服务质量、提高网络性能发挥着至关重要的作用.目前,端口识别技术已不能准确

* 基金项目: 国家自然科学基金(61303061, 61402485); 高性能计算国家重点实验室开放课题(201513-01)
Foundation item: National Natural Science Foundation of China (61303061, 61402485); Open Fund from HPCL (201513-01)
收稿时间: 2015-03-16; 修改时间: 2015-04-07; 采用时间: 2015-08-04

地分类端口伪装和端口随机分配的应用流量,而负载加密技术又限制了深度数据包检测系统的识别能力.基于机器学习的流量分类技术利用数据流统计特征分类流量,能够较为准确地识别端口伪装且负载加密的应用,因而倍受学术界的重视.然而,流量统计特征的偏置问题和流量分布的类别不平衡问题一直是基于机器学习的流量分类技术面临的两项挑战.流量统计特征的偏置问题是指一些流量统计特征虽提高了部分应用识别准确率,但同时降低了另外一部分应用识别准确率.特征选择算法无法解决此类问题,因为无论该特征是否被选择,分类的整体准确率都将受到影响.流量分布的类别不平衡现象是指一种应用的样本数远远超过了其他应用,该现象造成的问题(类别不平衡问题)是机器学习流量分类器对占流量分布比例较小的应用识别准确率低.相关研究工作对流量分布的类别不平衡问题关注较多[1−3],但缺少解决流量统计特征偏置问题的有效方法.鉴于网络服务提供商需要利用机器学习算法同时分类多种应用流量,解决特征偏置问题具有紧迫性,为此,本文提出了基于集成聚类的流量分类架构TCFEC.TCFEC由多个基分类器和一个决策器构成,每个基分类器采用K均值聚类算法在不同的特征子空间聚类而成.由于采用了聚类算法,基分类器能够将新出现的样本聚集成新簇,以此发现应用行为模式的变化.在基分类器分类结果不一致时,决策器能够以最小的错误率对流的类别进行判定,消除特征偏置问题对分类结果的不利影响.此外,本文充分考虑了训练数据集中类别不平衡现象对聚类算法参数选择的影响,首先采用人工小样本过抽样技术(synthetic minority over-sampling technique,简称SMOTE)来平衡训练数据集,再利用归一化互信息量度量选择不同参数时聚类的质量.与Bernaille等人[4]提出的参数确定方法相比,本文验证了平衡训练数据集对K均值聚类算法参数选择的重要性.

概括来说,本文的创新点在于以下4个方面:

(1) 提出了TCFEC流量分类架构,解决了特征偏置问题,提高了流量分类的准确率.

(2) 充分考虑了训练数据集的不平衡现象对K均值聚类算法参数选择的影响,先应用了SMOTE平衡训练数据集,再利用归一化互信息量度量聚类质量.在国际公开的UNIBS数据集上的实验结果表明,该方法可提高加密私有协议Skype流量的识别准确率.

(3) 设计并实现了两种最优决策器,充分考虑了基分类器分类结果不一致的情况,以最小错误率判定流的类别.

(4) 在国际公开的UNIBS数据集上,本文验证了TCFEC的有效性.与传统的机器学习流量分类器相比,TCFEC的流准确率最高提升5%,字节准确率最高提升6%.

本文首先介绍流量分类的最新研究进展.然后详细阐述基于集成聚类的流量分类架构.接着给出评估方法和实验结果.最后总结全文.

1 相关工作

1.1 基于聚类算法的流量分类

目前,已有很多研究人员将聚类算法应用于流量分类中.McGregor等人[5]最早使用EM算法聚类传输层的流特征,然而他们没有评估分类的准确率,也没有给出产生最优聚类结果的流特征.Zander等人[6]进一步拓展了McGregor等人的工作,他们使用Autoclass算法分类流量并选择出产生最优结果的流特征.文献[5,6]的工作都不能用于在线分类流量,因为其流特征是从完整的TCP流中提取得到的.而且Erman等人[7]指出,EM聚类算法和AutoClass算法的学习时间较长.Bernaille等人[4,8]进一步将聚类算法应用于在线流量分类,在离线训练时,他们用K均值聚类算法聚类一个TCP连接的前若干个数据包大小建立分类模型,然后将该模型应用于在线流量分类.文献[9−11]进一步利用K均值聚类算法识别未知流量.虽然文献[4−11]的方法取得了显著的成效,但都没有关注流量分类器的稳定性,也没有解决流量统计特征的偏置问题.

1.2 多分类器结合模型

目前,主要有两种多分类器结合模型:一种是集成学习模型,该模型建立多个基分类器,分类的最终结果是在多个基分类器间做出选择得到的,例如Bagging算法[12];另一种是多分类器融合模型.

相对于单个分类器,集成学习模型能够获得更加稳定的分类结果[13−15],更适合于实际环境下的网络流量分类.Kuncheva 等人[15]验证了 K 均值集成聚类的稳定性.

分类器融合模型近年来也受到了不少学者的关注,该模型常采用分层分类的方法.Claesen 等人[16]结合使用监督学习和无监督聚类来分类流量,他们先将训练数据集分成若干个簇,在包含多类样本的簇中进一步应用监督学习算法建立分类器.Szabo 等人[17]提出了多层机器学习流量分类系统,该系统的学习方法与文献[16]描述的方法类似.Este 等人[18]将单分类支持向量机与多分类支持向量机结合来分类流量,他们首先用单分类支持向量机独立地描述应用行为,对于难分类的流样本,采用多分类支持向量机进一步分类.然而,网络应用行为是复杂的,例如FTP会产生控制数据包和文件传输数据包,仅通过单分类支持向量机很难描述网络应用行为.而且,文献[19,20]都没有考虑流量分类时,流量统计特征的偏置问题.为此,本文提出了基于集成聚类的流量分类架构TCFEC,并尝试只用数据包大小分类流量.即,使网络应用采用加密隧道技术进行通信,数据包大小特征也一定能够从 IP 头中提取出来,而且该特征是数据包层面所有特征中最稳定的[21].

2 基于集成聚类的流量分类架构

本节主要介绍 TCFEC 分类架构的训练和在线分类过程.TCFEC 由多个基分类器和一个决策器构成.每个基分类器通过在不同的特征子空间中聚类得到.TCFEC 的决策器用于在基分类器分类结果不一致时,以最小的错误概率给出分类结果.图 1 描述了 TCFEC 的整体架构.

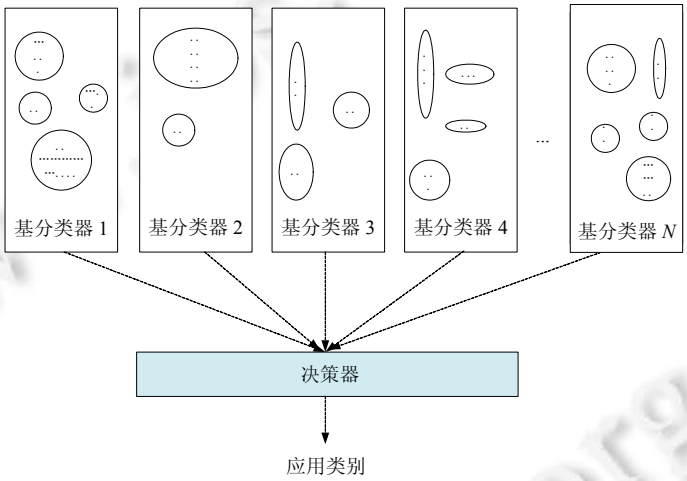


Fig.1 Traffic classification framework based on ensemble clustering

图 1 基于集成聚类的流量分类架构

2.1 建立基分类器

2.1.1 特征子空间下的 K 均值聚类

本节提取每个TCP单向流的前N个数据包大小作为特征.因此,用于分类的特征向量处于N维空间中.然而,网络中某些短流的数据包数目可能少于 N.在这种情况下,本文用 0 来填充特征向量.零填充的合理性在于每个数据包大小是从 IP 报文头中解析得到的,且至少为 40 字节,零特征值表示在一个流中相应位置上没有数据包.

每个子空间是基于原有 N 维空间的一个特定映射得到的.具体映射方法为:选择每个 TCP 数据流的前 P ($1\leqslant P\leqslant N$)个数据包大小作为特征向量,所有这些特征向量处于 P 维空间中,该 P 维空间是原有 N 维空间的子空间,每个基分类器在相应的子空间训练得到的.具体来说,我们使用每个单向 TCP 流的第 1 个数据包大小建立第 1 个基分类器,使用每个流的前两个数据包大小建立第 2 个基分类器,使用每个流的前 3 个数据包大小建立第 3 个基分类器,以此类推.本节只观察每个数据流的前 N 个数据包,因此有 N 个基分类器被建立.

由于 K 均值聚类算法比其他聚类算法,如高斯混合聚类和谱聚类更简单、快速,本节采用 K 均值聚类算法建立每个基分类器.K 均值聚类算法在特征空间中随机选择 k 个数据点作为初始簇中心,通过计算数据点和簇中心的欧式距离,将其他数据点分配到相似的簇中.然后,该算法重新计算每个簇的均值.上述过程不断重复,直到每个簇中心不再改变为止.K 均值算法的参数 k 对聚类结果有一定的影响.在第 2.1.2 节中,本文将设计簇数目 k 的选择方法.

2.1.2 簇数目选择算法

本节提出了一种基于平衡训练数据集和归一化互信息量相结合的最优簇数目选择算法(如图 2 所示).在介绍该算法之前,本节首先介绍标记每个生成簇的方法.

算法 1. 基于平衡训练数据集和归一化互信息量相结合的参数选择算法.
输入:*max_clusnum*. //最大的簇数目
输出:*optimal_clusnum*. //最优簇数目
procedure *choose_clusnum* (*max_clusnum*)
1. **begin**
2. *clusnum*←训练数据集中应用类别的数目;
3. *optimal_NMI*=−1;
4. *optimal_clusnum*=*clusnum*;
5. 应用 SMOTE 平衡训练数据集;
6. **while** (*clusnum*<*max_clusnum*)
7. 使用 *clusnum* 在训练数据集上建立 *K* 均值聚类器;
8. 标记训练集的簇和类别的映射;
9. 根据标记的簇分布和类别分布,计算归一化互信息量 *NMI*;
10. **if** (*NMI*>*optimal_NMI*)
11. *optimal_NMI*=*NMI*;
12. *optimal_clusnum*=*clusnum*;
13. **end if**
14. *clusnum*++;
15. **end while**
16. **return** *optimal_NMI*;

Fig.2 Algorithm for determining the number of clusters

图 2 簇数目选择算法

在理想情况下,生成的簇数目与实际分类的应用类别数目相同,而且每个簇中只含有一个应用类别的样本.然而在实际聚类过程中,一种应用有多个通信行为,如FTP协议流分为控制流和数据流,因此,每个应用可以用多个簇来描述,且一个簇中可以包括多个应用类别的样本.本文把一个簇标记成在该簇中样本数目占主导地位的应用类别.例如,若一个簇中包含 bittorrent 和 Skype 两类应用的样本,且 bittorrent 样本数为 1 000,而 Skype 样本数为 10,则该簇被标记为 bittorrent 类别.值得注意的是,如果训练数据集是不平衡的,比如 bittorrent 的样本数远多于 Skype 的样本数,那么生成的簇更有可能被标记为 bittorrent.为此,必须先平衡训练数据集,再进行聚类.此外,在使用标记后的簇分类流量时,若测试样本被分类到标记的簇中,则该样本以一定概率 P 被指派到相应的类别.概率 P 是由簇中占主导地位的应用样本数除以整个训练集中的样本总数得到的.

由图 2 所示,算法首先应用 SMOTE[22]平衡训练数据集.SMOTE 技术能够为分布比例较小的应用(这里简称小应用)人工地生成样本,它计算每个小应用样本的近邻点,在小应用样本与其近邻点之间人工地生成小应用样本以平衡训练数据集.在平衡训练数据集的基础上,再利用归一化互信息量来度量选择不同参数时的聚类质量.

给定随机变量 X 表示应用类别的分布,随机变量 Y 表示簇标签的分布,X 和 Y 之间的互信息量为

$$MI(X,Y)=\sum_{x,y}P(x,y)\log\frac{P(x,y)}{P(x)P(y)} \tag{1}$$

其中,$P(x,y)$表示应用 x 产生的流被分配到簇 y 的概率,$P(x)$是应用 x 的概率,$P(y)$是簇 y 的概率.为确保 $MI(X,Y)$ 的值在 0 和 1 之间,本节归一化互信息量如下:

$$NMI(X,Y)=\frac{MI(X,Y)}{\min[H(X),H(Y)]} \tag{2}$$

其中,$H(X)$和$H(Y)$分别表示X和Y的熵.归一化互信息量越大,聚类结果越好.

图 2 所示的算法在平衡后的训练数据集上,以最大归一化互信息量来确定最优的簇数目.由于 K 均值算法的分类时间随着簇数目的增多而上升,本文为加快流量分类的速度,将最大的簇数目的上限值 *max_clusnum* 设置为 100.*K*-means 算法循环地尝试不同的簇数目,每一次聚类后都要重新计算生成的簇和应用之间的归一化互信息量.*K*-means 算法的簇数目初始值为训练数据集中应用类别的数目,以步长为 1 开始尝试,直至尝试到簇数目为 100,算法选择归一化互信息量最大的簇数目作为最优簇数目.

2.2 建立决策器

当多个基分类器在分类同一个数据流所获得的结果不一致时,决策器要以最小错误率进一步判别该流的应用类别.

在训练基分类器时,若第 i 个基分类器以概率 P_{iq} 将流分类成应用 q,而第 j 个基分类器以概率 P_{jm} 将同一个流分类成应用 m,且 P_{iq} 和 P_{jm} 是所有基分类器中分类概率最大的前两个.为了进一步确定该流的应用类别,本节首先建立新的模式,其形式为〈P_{iq}/P_{jm},实际的应用类别〉.将所有冲突的结果都以这样的方式进行处理,于是构成了一个新的训练数据集 *new_traindata*.值得注意的是,为了减弱在 *new_traindata* 上各个应用类别的样本数分布的不平衡性对分类结果的影响,TCFEC 决策器在每两个应用之间进行决策.即如果有 n 个应用,TCFEC 将最多建立 $n\times(n-1)/2$ 个决策器.图 3 给出两个基分类器在分类应用 q 与应用 m 冲突时,决策器的建立算法.该算法在训练数据集上建立两种决策器:基于 SVM 的决策器和基于 Hash 的决策器.由文献[2]的研究结果可知,AUC 度量值更适合于在不平衡数据集上衡量分类器的性能.因此,该算法在训练数据集上分别用 SVM 决策器和 Hash 决策器进行分类,并选择获得最高 AUC 度量值的决策器作为最终的决策器.最终的决策器被插入到哈希表 modelmap 中.modelmap 的键值为冲突的应用名,modelmap 的值为相应的决策器.下面将详细介绍 SVM 决策器和 Hash 决策器的建立.

算法 2. 建立决策器.
输入:*newtrain_data*(P_{iq}/P_{jm},实际的应用类别). //P_{iq} 是第 i 个基分类器将样本分类成
//应用 q 的概率;P_{jm} 是第 j 个基分类器
//将样本分类成应用 m 的概率
输出:*modelmap*. //决策器

```
procedure decision_model (newtrain_data)
1.  begin
2.  在 newtrain_data 上建立支持向量机 SVM_qm;
3.  在 newtrain_data 上以 P_iq/P_jm 为关键字建立 HASH 表 H_qm;
4.  AUC1←SVM_qm.classify(newtrain_data);
5.  AUC2←H_qm.classify(newtrain_data);
6.  modelmap.key←(应用 q、应用 m);
7.  if AUC1>AUC2
8.     modelmap.value←SVM_qm;
9.  else
10.    modelmap.value←H_qm;
11. End
12. return modelmap;
```

Fig.3 Algorithm for decision when conflicts occur between base classifiers

图 3 在基分类器分类冲突时用于决策的算法

2.2.1 SVM 决策器

在新建立的训练集上,决策器的目的是为了确定一个阈值Λ.如果 $P_{iq}/P_{jm}>\Lambda$,那么决策器将该流分类成应用 q;否则,分类成应用 m.然而,为了以最小的错误率来确定流的类别,决策器想确定一个最优的阈值Λ并非易事.在图 4 中,横坐标为 P_{iq} 与 P_{jm} 的比值;而纵坐标无物理含义,只是为了便于观察,将模式值的分布在二维平面中显示.如图 4 所示,用于区分 bittorrent 和 http 的阈值并不清晰.造成这种现象的主要原因在于:一是 K 均值算法的聚类结果是局部最优的,这使得一些簇不能很好地描述应用的行为;二是由于流量类别的标注过程中产生误差,这使

得训练数据集中存在一定的噪声.为此,本节对训练数据进行了预处理,应用高斯核函数将每个模式 P_{iq}/P_{jm} 转换为 $y=f(P_{iq}/P_{jm})$,其中转换后的模式 y 可以通过一个超平面来区分.因此,问题转化为如何确定一个最优的超平面以分类模式 y.

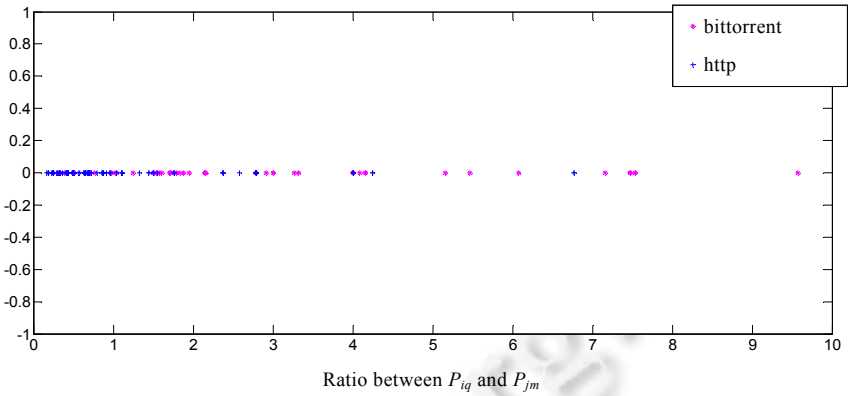


Fig.4 Distribution of pattern values that are used for discriminating between http and bittorrent

图 4 用于区分 http 和 bittorrent 的模式值分布

若超平面表示成 $g(y)=a^t y$,则从模式 y 到超平面的距离是$|g(y)|/\|a\|$.本节的目的是为了找到一个最小的$\|a\|$,使得距离最大.借助于拉格朗日因子,我们构造函数如下:

$$L(a,\alpha)=\frac{1}{2}\|a\|^2-\sum_{k=1}^{n}\alpha_k[z_k a^t y_k-1] \tag{3}$$

其中,z_k=1 或−1 表明模式 y_k 的应用类别,α_k 是拉格朗日待定因子.本文使用二次规划来求解这个等式,得到一个最小的权向量 a.综上所述,建立最优超平面的过程实际上就是支持向量机 SVM 的训练过程.

然而,我们的前期研究结果[23]显示:在不平衡的训练数据集中,SVM 分类器将偏向于大类别的识别,造成识别大类别的误报率较高,而识别小类别的检全率较低.为此,本文又建立了 Hash 决策器,以校正每个基分类器的错误分类结果.

2.2.2 Hash 决策器

哈希决策器实际上是一个哈希表,用于校对基分类器分类结果.训练数据集中的每个样本构成了哈希表中的每一项.具体来说,P_{iq}/P_{jm} 是哈希表的关键字,实际的应用类别是哈希表的值.如果在建立哈希表的过程中,哈希表中的关键字冲突,我们就移除哈希表中相应的项.然而,这种移除操作会使得 TCFEC 产生漏报率.为此,在实际的分类过程中,若哈希表中没有检索到 P_{iq}/P_{jm},我们就应用最大似然比加以决策.即,如果 P_{iq} 大于 P_{jm},TCFEC 就将该流判别为应用 q;否则,判别为应用 m.

2.3 TCFEC在线分类算法

TCFEC 在线分类算法如图 5 所示.该算法首先从每个 TCP 流中提取前 n 个数据包大小作为特征向量.TCFEC 的第 1 个基分类器 *kmeans_base*[1]使用每个 TCP 流的第 1 个数据包大小分类,第 2 个基分类器 *kmeans_base*[2]使用每个 TCP 流的前 2 个数据包大小分类,*kmeans_base*[3]使用每个 TCP 流的前 3 个数据包大小分类,以此类推.然后,每个基分类器以概率 *res*[*i*].*probability* 将流分类成应用 *res*[*i*].*classname*.该算法选择概率最大的前两个分类结果 *res*[*k*]和 *res*[*j*].最后,根据训练时就已确定的应用名顺序来建立新的样本 *prob*(第 12 行和第 15 行),并用基分类器分类的应用名检索哈希表 *modelmap* 以使用相应的决策器,给出流的最终类别(第 13 行和第 16 行).

算法 3. TCFEC 在线分类算法.
输出:应用类别.
procedure ***TCFEC_classifier***()
1. **begin**
2. 跟踪 TCP 流,并提取每个流的前 *n* 个数据包大小形成特征向量 *v*[1],*v*[2],…,*v*[*n*];
3. **for** *i*=1 **to** *n*
4. *res*[*i*]←*kmeans_base*[*i*].*classify*(*v*[*i*]);
5. **end for**
6. 根据 *res*[*i*].*probability* 从大到小排列 *res*;
7. 从 *res* 中选择前两个识别结果 *res*[*k*]和 *res*[*j*];
8. **if** *res*[*k*].*classname*==*res*[*j*].*classname*
9. **return** *res*[*k*].*classname*;
10. **else**
11. **if** *samesequencewithtrain*(*res*[*k*].*classname*,*res*[*j*].*classname*)
12. *prob*=*res*[*k*].*probability*/*res*[*j*].*probability*;
13. **return** *modelmap*(*res*[*k*].*classname*,*res*[*j*].*classname*).*classify*(*prob*);
14. **else**
15. *prob*=*res*[*j*].*probability*/*res*[*k*].*probability*;
16. **return** *modelmap*(*res*[*j*].*classname*,*res*[*k*].*classname*).*classify*(*prob*);
17. **end if**

Fig.5 TCFEC online classification algorithm
图 5 TCFEC 在线分类算法

3 评估方法

3.1 评估度量

3.1.1 流准确率和字节准确率

流的准确率是指被正确识别的流数占网络所有流数的百分比.字节的准确率是指被正确识别的数据包承载的字节数占网络传输的总字节数的百分比.Erman 等人[24]指出:在评价流量识别的准确性时,字节的准确率是非常关键的.他们给出的数据集表明:0.1%的流占了整个流量字节总数的 46%,如果流量识别算法能够识别出除了这 0.1%的流以外所有的流,那么流的准确率可以达到 99.9%,但却损失了 46%的字节准确率.因此,在实际的流量识别效果评估中,在给出流的准确率的同时,也要给出字节的准确率.

3.1.2 检全率和误报率

检全率(真阳性率)是指分类器识别出的应用流数占该应用产生的总流数的百分比.真阳性(true positive)是指属于应用 *C* 的流量而被分类成应用 *C*.漏报(false negative)是指属于应用 *C* 的流量而被分类成非应用 *C*.若真阳性数为 *TP*、漏报数为 *FN*,检全率(true positive rate)的计算如下:

$$true\ positive\ rate = \frac{TP}{TP + FN}.$$

误报(false positive)是指非应用 *C* 的流量被分类成应用 *C*.真阴性(true negative)是指非应用 *C* 的流量被分成非应用 *C*.假定误报数为 *FP*、真阴性数为 *TN*,则误报率(false positive rate)为

$$false\ positive\ rate = \frac{FP}{FP + TN}.$$

3.2 实验数据集

本文采用国际公开的 UNIBS 数据集进行实验.该数据集是由意大利都灵理工大学某科研小组在他们学院的路由器上用 tcpdump 软件捕获得到的.UNINS 数据集共有 3 个,分别命名为 unibs20090930,unibs20091001 和 unibs20091002.每个数据集上的样本标注信息是由 GT 工具准确提供的.本文主要在 UNIBS 数据集分类单向的 TCP 数据流.由于非对称路由的影响,单向 TCP 数据流频繁出现在互联网上.TCP 数据流可以进一步分为从客户端到服务器方向和从服务器到客户端方向.发送 SYN 数据包的一端为客户端,另一端则为服务器端.本文使用 libnids 工具跟踪 TCP 流,并提取 TCP 流中前若干个数据包的大小.表 1 列出了应用类别的流样本数目,其中,

Skype 为典型的加密私有协议[25,26],other 表示除了 bittorrent,Skype,http 和 ssl 之外的其他协议类别.

Table 1 Number of application flows in the UNIBS dataset

表 1 UNIBS 数据集中应用流的数目

	unibs20090930		unibs20091001		unibs20091002	
	s→*c*	*c*→*s*	*s*→*c*	*c*→*s*	*s*→*c*	*c*→*s*
bittorrent	3 278	1 741	867	525	344	201
Skype	154	145	152	154	214	213
http	7 086	4 821	14 096	10 592	7 360	5 059
ssl	1 223	752	1 230	1 013	579	506
other	176	176	270	262	176	173

由表 1 可见,在这 3 个数据集上,各种应用的流样本分布有较大的变化.这符合实际网络环境,因为应用的分布在实际的网络中是动态改变的.本文将分别评估 TCFEC 对客户端到服务器方向和服务器到客户端方向的 TCP 流识别能力.在评估中,本文使用其中一个数据集用于训练,其他数据集用于测试.该过程循环地进行 3 次,取 3 次结果的平均值来评估算法.

4 实验结果

本节首先分析数据包大小对流量识别的贡献度;然后验证平衡训练数据集对 *K* 均值聚类算法参数选择的重要性;接着,确定观察窗口的大小,即,提取单向 TCP 流数据包的数目;最后,通过与常见的典型算法 SVM 和 Bagging 进行比较分析,验证了 TCFEC 的有效性.

4.1 数据包大小的识别能力分析

文献[4,8,21]已经仅利用数据包大小分类流量.以此为依据,本节在 UNIBS 数据集上分析单向 TCP 数据流中不同数据包大小的识别能力.

信息增益可用于度量数据包大小在流量识别中的贡献度,其计算方法为

$$InfoGain(\text{类别},\text{数据包大小})=H(\text{类别})-H(\text{类别}|\text{数据包大小}) \tag{4}$$

其中,H(类别)表示应用类别的熵值;H(类别|数据包大小)表示数据包大小出现的情况下,应用类别的条件熵值.*InfoGain*(类别,数据包大小)表示数据包大小与应用类别间的信息增益,信息增益越大,表明数据包大小对流量识别的贡献度越高.本文分别统计 TCP 客户端到服务器方向(*c*→*s*)和 TCP 服务器到客户端方向(*s*→*c*)的前 10 个数据包,并计算它们的信息增益,如图 6 所示.可见,无论 *c*→*s* 方向还是 *s*→*c* 方向,第 1 个数据包大小的识别能力最强,而且 TCP 单向流的前 10 个数据包大小的识别能力按数据包的编号顺序逐渐下降.进一步发现,TCP 单向流的前 3 个数据包大小的贡献度高于 0.6.因此,数据包大小具有较强的识别能力,且应充分利用 TCP 单向流的前 3 个数据包.

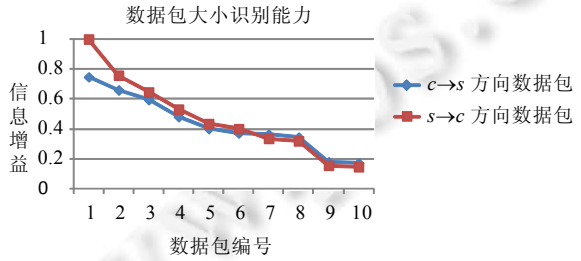


Fig.6 Information gain achieved with different packets

图 6 不同数据包的信息增益

4.2 平衡训练数据集对*K*均值聚类算法参数选择的影响

文献[4]仅用归一化互信息量来确定 *K* 均值聚类算法的参数,未考虑训练数据集的不平衡现象对参数选择

的影响.本节将平衡训练数据集后的参数选择方法与文献[4]的参数选择方法(未平衡训练数据集)进行比较.由图 7 可见,利用平衡训练数据集后选择的聚类参数识别应用时,将获得较高的检全率,尤其是对样本数较少的 Skype 和 other 类的识别.这主要是由于在平衡后的训练数据集上进行聚类,更有可能生成识别样本数较少应用的簇.因此,平衡训练数据集对 K 均值聚类算法的参数选择是至关重要的.

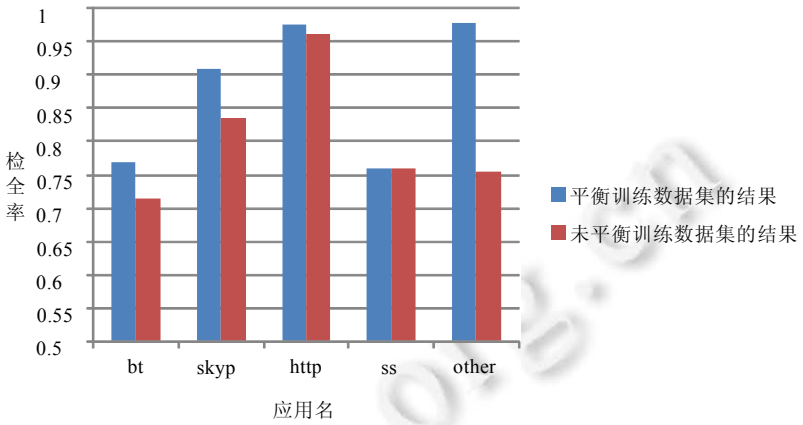


Fig.7 Comparison of TPRs on the balanced training dataset and non-balanced training dataset

图 7 平衡训练数据集与未平衡训练数据集的检全率比较

4.3 TCP单向流观察窗口大小

在确定数据包识别能力和 K 均值聚类算法的参数之后,还需要确定 TCP 观察窗口的大小,以便能够在线识别流量.对于从客户端到服务器方向的流和从服务器到客户端方向的流,本节为 TCFEC 分别确定观察窗口的大小.我们通过改变观察窗口大小使 TCFEC 获得不同的分类准确率,然后选择使分类准确率达到最大的观察窗口大小.

由图 8 和图 9 可见,对于客户端到服务器方向的流,前 3 个数据包可使 TCFEC 达到最高的准确率;而对于服务器到客户端方向的流,前两个数据包获得的准确率最高.当观察窗口变大时,例如从客户端到服务器方向观察 4 个数据包时,分类准确率下降.这很可能是由于第 4 个数据包干扰了 TCFEC 识别流量,导致整体分类准确率下降.因此,对于客户端到服务器方向的流,本节设置观察窗口大小为 3;而对于服务器到客户端方向的流,本节设置观察窗口大小为 2.

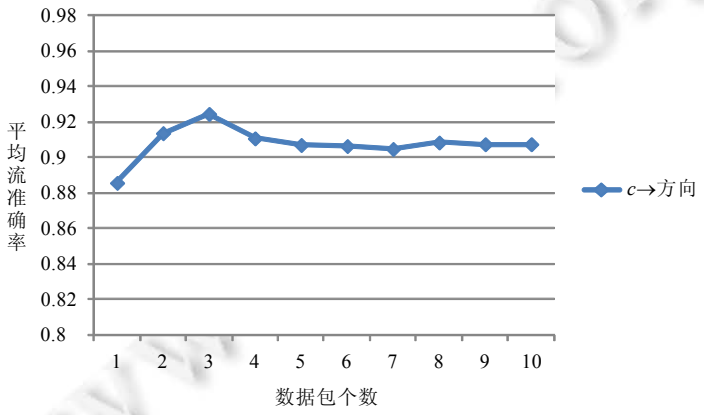


Fig.8 Overall accuracy achieved with the different number of packets in the client-to-server flows

图 8 客户端到服务器方向的流,使用不同的包数目获得的整体准确率

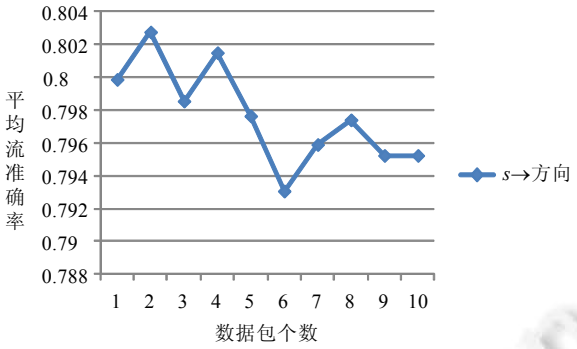


Fig.9 Overall accuracy achieved with the different number of packets in the server-to-client flows

图 9 服务器到客户端方向的流,使用不同的包数目获得的整体准确率

4.4 TCFEC与基于传统*K*均值聚类的流量分类器比较

本节从流准确率、字节准确率以及识别每个应用的检全率这 3 种度量来比较 TCFEC 与基于传统 *K* 均值聚类的流量分类器性能.

4.4.1 流准确率和字节准确率的比较

对于从客户端到服务器方向的流识别而言,由于仅观察流的前 3 个数据包,因此,TCFEC 由 3 个基分类器构成.其中,

- 第 1 个基分类器使用每个流的第 1 个数据包大小训练而成;
- 第 2 个基分类器使用每个流的前两个数据包大小训练而成;
- 第 3 个基分类器使用每个流的前 3 个数据包大小训练而成.

就客户端到服务器方向的流识别而言,由于 TCFEC 使用了前 3 个数据包大小,因此基于传统 *K* 均值聚类的流量分类器也使用前 3 个数据包大小(表 2 中表示为 *k*_means_3).同理,就服务器到客户端方向的流识别而言,TCFEC 使用了前两个数据包大小,因此基于传统 *K* 均值聚类的流量分类器使用前两个数据包大小进行流量分类(表 2 中表示为 *k*_means_2).

Table 2 Comparison of flow accuracies and byte accuracies

表 2 流准确率与字节准确率的比较

	数据流方向	平均流准确率	流准确率的标准差	平均字节准确率	字节准确率标准差
TCFEC	*c*→*s*	0.936 9	0.020 6	0.964 8	0.021 1
	s→*c*	0.860 9	0.042 6	0.892 2	0.010 6
*k*_means_2	*s*→*c*	0.805 6	0.026 9	0.854 1	0.011 0
*k*_means_3	*c*→*s*	0.922 1	0.024 9	0.922 6	0.070 1

表 2 比较了 TCFEC 与基于传统 *K* 均值聚类的流量分类器流准确率和字节准确率.容易看出,就流准确率和字节准确率而言,无论从客户端到服务器方向还是从服务器到客户端方向,TCFEC 明显好于基于传统的 *K* 均值聚类流量分类器.这是因为 TCFEC 的每个基分类器是用 *K* 均值聚类算法训练而成,TCFEC 在多个基分类器间做了最优的决策,以最小的错误率选择某个基分类器的结果作为最终结果.

4.4.2 识别每个应用的检全率比较

对于从客户端到服务器方向的 TCP 流,TCFEC 与每个基分类器识别不同应用的检全率如图 10 所示.可见,TCFEC 识别应用的检全率明显高于 *k*_means_3.这是由于 TCFEC 以最小的错误率在每个基分类器间做出最优的选择,因此,TCFEC 对每种应用识别的检全率获得较高的结果.对于分类服务器到客户端方向的 TCP 流,结果类似(如图 11 所示).

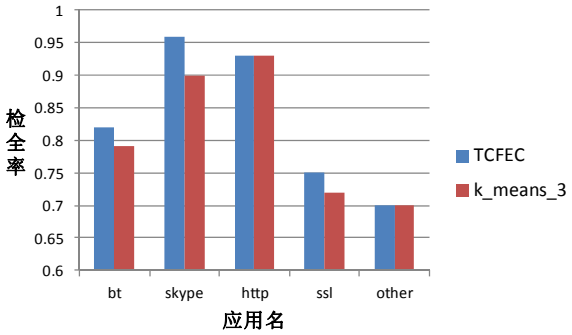


Fig.10 Comparison between TCFEC and *K*-means based classifier, when classifying client-to-server flows

图 10 分类客户端到服务器方向的流时, TCFEC 与 *K* 均值分类器的比较

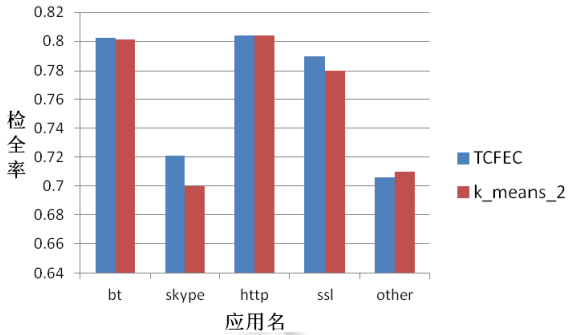


Fig.11 Comparison between TCFEC and *K*-means based classifier, when classifying server-to-client flows

图 11 分类服务器到客户端方向的流时, TCFEC 与 *K* 均值分类器的比较

4.5 TCFEC与典型流量分类算法的比较

我们前期的研究工作[23]使用了 Bagging 算法识别流量,Bagging 算法的基分类器为 C4.5 决策树.本节比较分析 TCFEC,SVM 和 Bagging 这 3 种算法的识别准确率.

4.5.1 流准确率和字节准确率的比较

表 3 对比了 TCFEC,SVM 和 Bagging 这 3 种分类算法的流准确率和字节准确率.可见,集成聚类 TCFEC 的识别结果明显要好于 SVM 和 Bagging 分类算法.更低的字节准确率标准差和流准确率标准差,表明在训练数据集发生变化时,集成聚类 TCFEC 分类更稳定.

Table 3 Comparison of flow accuracies and byte accuracies

表 3 流准确率与字节准确率的比较

	数据流方向	平均流准确率	流准确率的标准差	平均字节准确率	字节准确率标准差
TCFEC	$c\rightarrow s$	0.936 9	0.020 6	0.964 8	0.021 1
	$s\rightarrow c$	0.860 9	0.042 6	0.892 2	0.010 6
SVM	$c\rightarrow s$	0.890 3	0.062	0.954 6	0.054 1
	$s\rightarrow c$	0.805 5	0.060 7	0.835 2	0.046 2
Bagging	$c\rightarrow s$	0.902 5	0.051 1	0.945 3	0.039 6
	$s\rightarrow c$	0.851 6	0.043 9	0.867 7	0.021 0

4.5.2 比较每个应用的识别结果

图 12 和图 14 比较了 TCFEC,SVM,Bagging 分类器识别每一种应用的检全率.就客户端到服务器方向的 TCP 流识别而言(如图 12 所示),TCFEC 识别 bt,skype,ssl 的检全率明显高于 SVM 和 Bagging 分类算法;然而,TCFEC 识别 http 流量的检全率低于 SVM 分类算法.如图 12 所示,SVM 分类算法虽然识别 http 协议的检全率较高,但识别 http 协议的误报率却高达 50%.另一方面,就服务器到客户端方向的 TCP 流识别而言(图 14 所示),TCFEC 识别 bt,Skype 流量的检全率都高于 SVM 和 Bagging 分类算法.同样,如图 15 所示,Bagging 和 SVM 分类算法识别 http 流量的误报率也很高.由于 UNIBS 数据集中类别分布是不平衡的,http 流数占数据集中的大部分,而 SVM 和 Bagging 分类算法在不平衡的数据集上偏向于样本数较多的类别,因此,SVM 和 Bagging 分类算法识别 http 流数的误报率较高.但相对于 SVM 分类算法,Bagging 分类算法在不平衡训练数据集上的识别效果更好.

值得注意的是,由于从训练数据的预处理到流量分类器的设计,TCFEC 的实现充分考虑了流量分布的类别不平衡问题,对样本数较少的加密私有协议 Skype 流量识别的检全率较高且误报率较低.因此,TCFEC 更适合于识别加密的 Skype 流量.

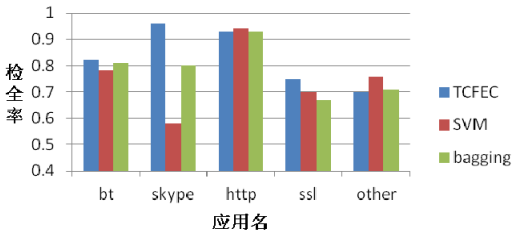


Fig.12 Comparison of TPRs when classifying client-to-server flows

图 12 分类客户端到服务器方向的流时,检全率比较

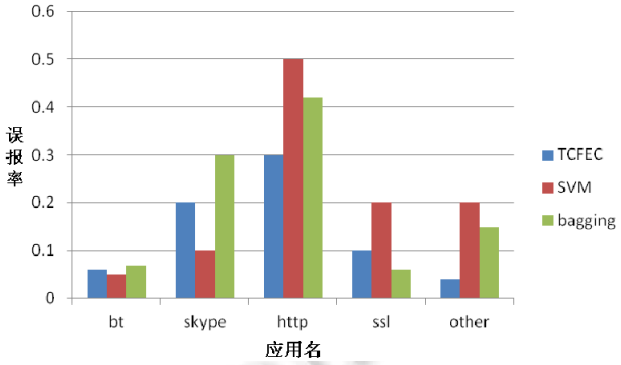


Fig.13 Comparison of FPR between TCFEC and SVM, when classifying client-to-server flows

图 13 分类客户端到服务器方向的流时,TCFEC 与 SVM 的误报率比较

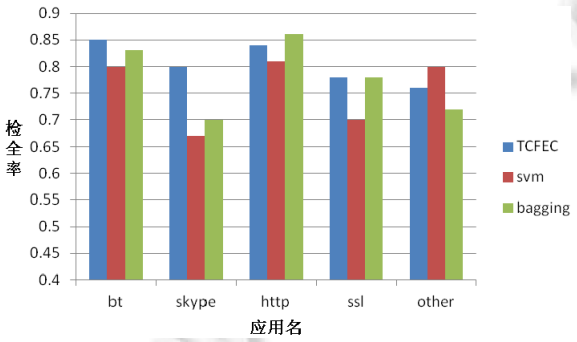


Fig.14 Comparison of TPRs when classifying server-to-client flows

图 14 分类服务器到客户端方向的流时,检全率比较

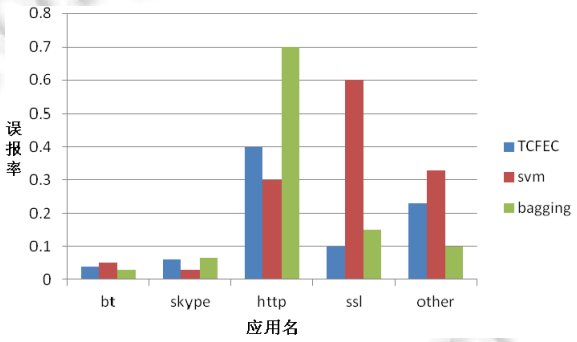


Fig.15 Comparison of FPR when classifying server-to-client flows

图 15 分类服务器到客户端方向的流时,误报率比较

5 总 结

本文提出了基于集成聚类的流量分类架构 TCFEC.TCFEC 仅需提取单向 TCP 流前若干个数据包大小作为特征,适合于在线分类流量.TCFEC 的每个基分类器通过在不同的特征子空间中聚类生成,对于分类不一致的样本,本文设计并实现了 SVM 决策器和 Hash 决策器以进一步决策该数据流的类别.通过与 K 均值聚类算法、SVM 分类算法和 Bagging 分类算法的比较,本文在公开的 UNIBS 数据集上验证了 TCFEC 的准确性和稳定性.

References:

[1] Zhang H, Lu G, Mahmoud TQ, Zhang Y, Yu XZ. Feature selection for optimizing traffic classification. Computer Communications, 2012,35(12):1457−1471. [doi: 10.1016/j.comcom.2012.04.012]

[2] Yang J, Wang Y, Qiao Y, Zhao X, Liu F, Cheng G. On evaluating multi-class network traffic classifiers based on AUC. Wireless Personal Communications, 2015,83(3):1731−1750. [doi: 10.1007/s11277-015-2473-4]

[3] Liu Q, Liu Z. A comparison of improving multi-class imbalance for internet traffic classification. Information Systems Frontiers, 2012,16(3):509−521. [doi: 10.1007/s10796-012-9368-7]

[4] Bernaille L, Teixeira R, Salamatian K. Early application identification. In: Proc. of the 2006 ACM CoNEXT Conf. New York: ACM Press, 2006. 1−12. [doi: 10.1145/1368436.1368445]

[5] McGregor A, Hall M, Lorier P, Brunskill J. Flow clustering using machine learning techniques. In: Proc. of the Passive and Active Network Measurement (PAM). LNCS 3015, Heidelberg: Springer-Verlag, 2004. 205−214. [doi: 10.1007/978-3-540-24668-8_21]

[6] Zander S, Nguyen T, Armitage G. Automated traffic classification and application identification using machine learning. In: Proc. of the IEEE Conf. on Local Computer Networks (LCN 2005). Sydney: IEEE Computer Society Press, 2005. 2257. [doi: 10.1109/LCN.2005.35]

[7] Erman J, Arlitt M, Mahanti A. Traffic classification using clustering algorithms. In: Proc. of the 2006 SIGCOMM Workshop on Mining Network Data (MineNet 2006). New York: ACM Press, 2006. 281−286. [doi: 10.1145/1162678.1162679]

[8] Bernaille L, Teixeira R, Akodkenou I. Traffic classification on the fly. ACM SIGCOMM Computer Communication Review, 2006, 36(2):23−26. [doi: 10.1145/1129582.1129589]

[9] Erman J, Mahanti A, Arlitt M, Cohen I, Williamson C. Offline/Realtime traffic classification using semi-supervised learning. Performance Evaluation, 2007,64(9-12):1194−1213. [doi: 10.1016/j.peva.2007.06.014]

[10] Zhang J, Xiang Y, Zhou W, Wang Y. Unsupervised traffic classification using flow statistical properties and IP packet payload. Journal of Computer and System Sciences, 2013,79(5):573−585. [doi: 10.1016/j.jcss.2012.11.004]

[11] Zhang J, Chen C, Xiang Y, Zhou W, Vasilakos AV. An effective network traffic classification method with unknown flow detection. IEEE Trans. on Network and Service Management, 2013,10(2):133−147. [doi: 10.1109/TNSM.2013.022713.120250]

[12] Breiman L. Bagging predictors. Machine Learning, 1996,24(2):123−140. [doi: 10.1023/A:1018054314350]

[13] Li L, Hu Q, Wu X, Yu D. Exploration of classification in ensemble learning. Pattern Recognition, 2014,47:3120−3131. [doi: 10.1016/j.patcog.2014.03.021]

[14] Wang G, Sun J, Ma J, Xu K, Gu J. Sentiment classification: The contribution of ensemble learning. Decision Support Systems, 2014,57:77−93. [doi: 10.1016/j.dss.2013.08.002]

[15] Kuncheva L, Ludmila I, Dmitry P. Evaluation of stability of *k*-means cluster ensembles with respect to random initialization. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2006,28(11):1798−1808. [doi: 10.1109/TPAMI.2006.226]

[16] Claesen M, Frank S, Suykens J. EnsembleSVM: A library for ensemble learning using support vector machines. Journal of Machine Learning Research, 2014,15:141−145.

[17] Szabo G, Szule J, Turanyi Z, Pongracz G. Multi-Level machine learning traffic classification system. In: Proc. of the 11th Int'l Conf. on Networks (ICN). Saint Gilles: IEEE, 2012. 69−76.

[18] Este A, Gringoli F, Salgarelli L. Support vector machines for TCP traffic classification. Computer Networks, 2009,53(14): 2476−2490. [doi: 10.1016/j.comnet.2009.05.003]

[19] Wright C, Monrose F, Masson G. HMM profiles for network traffic classification. In: Proc. of the 2004 ACM Workshop on Visualization and Data Mining for Computer Security. Washington: ACM Press, 2004. 9−16. [doi: 10.1145/1029208.1029211]

[20] Dainotti A, Donato W, Pescape A, Salvo P. Classification of network traffic via packet-level Hidden Markov models. In: Proc. of the Global Telecommunications Conf. New Orleans: IEEE, 2008. 1−5. [doi: 10.1109/GLOCOM.2008.ECP.412]

[21] Este A, Gringoli F, Salgarelli L. On the stability of the information carried by traffic flow features at the packet level. ACM SIGCOMM Computer Communication Review, 2009,39(3):13−18. [doi: 10.1145/1568613.1568616]

[22] Nitesh V, Kevin W, Lawrence O, Philip K. SMOTE: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research. 2002,16:321−357.

[23] Zhang HL, Lu G. Machine learning algorithms for classifying the imbalanced protocol flows: evaluation and comparison. Ruan Jian Xue Bao/Journal of Software, 2012,23(6):1500−1516 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4074.htm [doi: 10.3724/SP.J.1001.2012.04074]

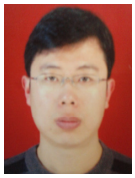
[24] Erman J, Mahanti A, Arlitt M. Byte me: A case for byte accuracy in traffic classification. In: Proc. of the 3rd Annual ACM Workshop on Mining Network Data (MineNet 2007). New York: ACM Press, 2007. 35−37. [doi: 10.1145/1269880.1269890]

[25] Yuan Z, Du C, Chen X, Wang D, Xue Y. SkypeTracer: Towards fine-grained identification for skype traffic via sequence signatures. In: Proc. of the Int'l Conf. on Computing, Networking, and Communications (ICNC). IEEE, 2014. 1−5. [doi: 10.1109/ICCNC.2014.6785294]

[26] Bonfiglio D, Mellia M, Meo M, Rossi D, Tofanelli P. Revealing Skype traffic: When randomness plays with you. ACM SIGCOMM Computer Communication Review, 2007,37(4):37–48. [doi: 10.1145/1282427.1282386]

附中文参考文献:

[23] 张宏莉,鲁刚.分类不平衡协议流的机器学习算法评估与比较.软件学报,2012,23(6):1500−1516. http://www.jos.org.cn/1000-9825/4074.htm [doi: 10.3724/SP.J.1001.2012.04074]



鲁刚(1982−),男,辽宁沈阳人,博士,工程师,主要研究领域为流量分类,网络行为分析与建模.

张宏莉(1973−),女,博士,教授,博士生导师,CCF 专业会员,主要研究领域为网络与信息安全,网络测量.

余翔湛(1973−),男,博士,教授,博士生导师,CCF 专业会员,主要研究领域为网络流量分类,网络行为分析.

郭荣华(1972−),男,博士,副研究员,CCF 专业会员,主要研究领域为网络流量分类,网络行为分析.