研究与开发

一种基于随机森林和改进卷积神经网络的网络流量分类方法

云本胜，干潇雅，钱亚冠
（浙江科技学院理学院，浙江 杭州 310023）

摘　要：为了提高网络流量分类模型的效率、降低模型复杂度，提出了一种基于随机森林和改进卷积神经网络的分类方法。首先，利用随机森林评估了网络流量各个特征的重要性，并根据重要性排序进行特征选择；其次，采用 AdamW 优化器和三角循环学习率优化了卷积神经网络分类模型；最后，将该模型搭建在 Spark 集群上实现模型训练的并行化。采用循环幅度恒定的三角循环学习率，选择 1 024、400、256 和 100 个最重要的特征作为输入的实验结果表明，模型的准确率分别提高到 97.68%、95.84%、95.03%和 94.22%。选择 256 个最重要的特征，采用不同学习率的实验结果表明，循环幅度减半的三角循环学习率的效果最佳，模型的准确率提高到 95.25%，模型训练时间减少近 1/2。

关键词：网络流量分类；随机森林；卷积神经网络；Spark
中图分类号：TP393
文献标志码：A
doi: 10.11959/j.issn.1000−0801.2023138

A network traffic classification method based on random forest and improved convolutional neural network

YUN Bensheng, GAN Xiaoya, QIAN Yaguan
School of Science, Zhejiang University of Science and Technology, Hangzhou 310023, China

Abstract: In order to improve the efficiency and reduce the complexity of network traffic classification model, a classification method based on random forest and improved convolutional neural network was proposed. Firstly, the random forest was used to evaluate the importance of each feature of network traffic, and the feature was selected according to the importance ranking. Secondly, AdamW optimizer and triangular cyclic learning rate were adopted to optimize the convolutional neural network classification model. Then, the model was built on Spark cluster to realize the parallelization of model training. Adopting triangular cyclic learning rate with constant cycle amplitude, the experimental results of selecting 1 024, 400, 256 and 100 most important features as input show that the model accuracy is improved to 97.68%, 95.84%, 95.03% and 94.22%, respectively. The 256 most important features were selected and the experimental results based on adopting different learning rates show that the learning rate with half the cycle amplitude works best, the accuracy of the model is improved to 95.25%, and training time of the model is reduced by nearly half.

Key words: network traffic classification, random forest, convolutional neural network, Spark

收稿日期：2023−02−08；**修回日期**：2023−07−02
基金项目：国家自然科学基金资助项目（No.61972357）；浙江省自然科学基金资助项目（No.LZ22F020007）
Foundation Items: The National Natural Science Foundation of China (No.61972357), The Natural Science Foundation of Zhejiang Provincial of China (No.LZ22F020007)

0　引言

随着网络数据量和数据复杂性的增加，深度学习在流量识别和分类领域受到越来越广泛的关注。基于深度学习的流量分类方法可以分为两类：基于数据包原始字节特征的深度学习方法和基于流内数据包序列特征的深度学习方法[1]。其中，基于数据包原始字节特征的深度学习方法首先对原始流量进行协议识别，以原始数据作为特征输入，然后利用深度学习模型提取特征[2]。

卷积神经网络（convolutional neural network，CNN）能更准确且高效地提取特征，在流量识别领域具有较好的性能。2017 年，Wang 等[3]率先提出使用网络流量的原始流数据作为 CNN 的输入，对恶意流量分类能达到90%以上的准确率。2020 年，Feng 等[4]提出一种改进的二维卷积神经网络分类模型（PtrCNN），对网络流进行归一化后将其映射成二维矩阵作为 CNN 的输入，不仅可以提高对不同协议的流量的识别精度，还可以减少分类时间。2021 年，Sun 等[5]提出了一种基于 Spark 的具有周期学习率的分布式 CNN，选取数据流的原始字节的信息作为输入特征，对正常流量和恶意流量进行二分类的准确率达到 90.417%。

以上方法直接使用原始字节的信息作为输入，存在特征冗余和信息利用率过低的问题，导致分类准确率不够高。

为了提高分类的准确率，研究人员通过对数据进行预处理或改进 CNN 模型等方法提高模型性能。Tong 等[6]对一些基于 QUIC 协议的服务进行分类，同时采用了统计方法、随机森林、CNN 进行特征预处理，实验表明，使用前 1 400 个特征进行分类效果最好，实现了 F1 的最小平均分数为 99.24%，而采用前 300 和 900 个特征的数据集的性能并不好，在微观和宏观平均 F1 分数方面仅超过 70%。Hu 等[7]对原始流的数据包净荷进行分割和重组，自动提取相关的特征，结合 CNN、长短期记忆（long short-term memory，LSTM）网络和若干个 Dense 层提出 CLD-Net，实验表明，采用 256 个特征时效果最好，在区分是否为虚拟专用网络（virtual private network，VPN）数据时准确度能达到 98%，但是八分类的平均准确率在 92%～94%。于帅等[8]提出了一种基于深度特征融合的流量分类方法，对 8 000 条流量数据生成共计 21 616 个特征，利用前 19 600 个特征进行分类，准确率达到 92%～99.89%。

以上方式虽然使分类准确率得到有效提高，但是仍存在特征选取过多、模型过于复杂等问题。另外，当下网络流量数据量大、特征复杂多样，还存在单机算力不足的问题。

针对以上问题，在现有工作的基础上，本文提出了一种基于随机森林与改进卷积神经网络对网络流量进行分类的方法，并将模型搭建在 Spark 集群上实现分布式运算。主要工作如下。

（1）通过随机森林算法计算基尼重要性，对原始字节信息提取的特征进行重要性排序，然后根据重要性确定特征选择个数并选择特征，简化了特征提取方式，并具有一定的可解释性。

（2）对 CNN 模型进行优化和简化，在保证模型性能的前提下轻量化模型，并且采用了 AdamW 优化器和学习率优化策略。对三角周期循环学习率和指数衰减学习率进行了比较，并且对三角循环学习率的 3 种不同周期循环幅度变化策略做了对比，提升了准确率，减少运算时间。

（3）运用 TensorFlowOnSpark，将模型搭建在 Spark 集群上，实现完全分布式计算，针对大规模数据，能够自动处理失效节点和负载均衡的问题，有效地提高运算效率。

1　改进的分类模型

流量分类模型主要分为预处理模块、特征选择和分类模块。预处理模块包括对原始数据进行预处理并基于统计的方法构建特征，采用随机森林算法评估特征重要性，选择特征，然后划分训练集和测

试集，再通过改进的 CNN 模型对训练集进行分类训练，最后利用测试集测试模型，并测定其性能指标。网络流量分类模型流程如图 1 所示。

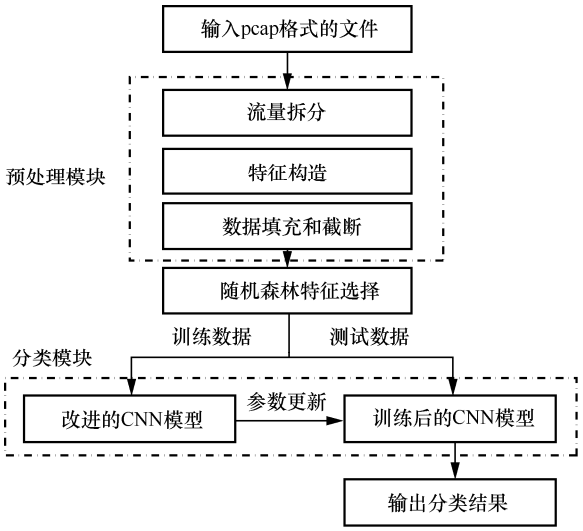

图 1　网络流量分类模型流程

1.1　原始数据预处理

原始网络流量通常以 pcap 格式存储且原始流量的长度不一致，不能直接输入神经网络模型，故需要对输入的原始数据进行预处理，如图 1 预处理模块所示，预处理的主要流程有流量拆分、特征构造和数据填充和截断。

原始网络流量通常以 pcap 文件格式存储，pcap 文件由一个文件头和多个连续的数据包组成。每个数据包由一个包头和包数据组成。pcap 的文件格式如图 2 所示。

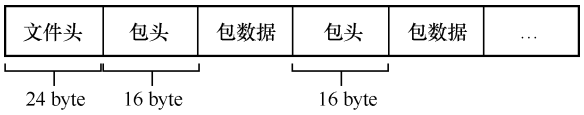

图 2　pcap 的文件格式

对 pcap 文件进行处理以提取所需信息。pcap 文件没有规定区分数据包的字符串，而是根据每个包头的 Caplen 定义数据区的长度得到下一个数据帧的位置，因此需要对原始流量数据进行拆分和提取，使用 Python 的 Scapy 库拆分 pcap 包，提取原始流量信息。原始流量数据中包含的 MAC 地址和 IP 地址不承载可以区分流量类型的特征[9]，但是容易使模型产生偏差，故对 MAC 地址和 IP 地址进行匿名化处理[10]。

将经过上述处理得到的十六进制数据信息，每两位一个字节，对应 0～255 的灰度数值转换为十进制信息，每个字节的十进制数表示一个流特征。然后利用全 0 填充的方式将不同数据包之间的字节长度填充或截断到相同长度，最终生成每条流的特征和类别标签构成的待选特征数据集。

1.2　随机森林特征选择

网络通常包含许多复杂且难以解释的预测变量，这意味着专业人员难以洞察可能导致潜在攻击等异常情况的数据模式。在这种情况下，特征选择可以发挥重要作用，因为它可以确定特征在异常流量检测中的重要性[11]，从而得到一个解释性更好的模型。

随机森林（random forest，RF）算法[12]组合多个相同分布的决策树作为基分类器，并通过投票机制得出最终结果，降低了决策树的不稳定性，使模型整体有较好的泛化性能。

RF 可通过重要性评估进行特征选择，RF 特征重要性选择如图 3 所示，特征重要性的度量方法主要有置换重要性和基尼重要性。

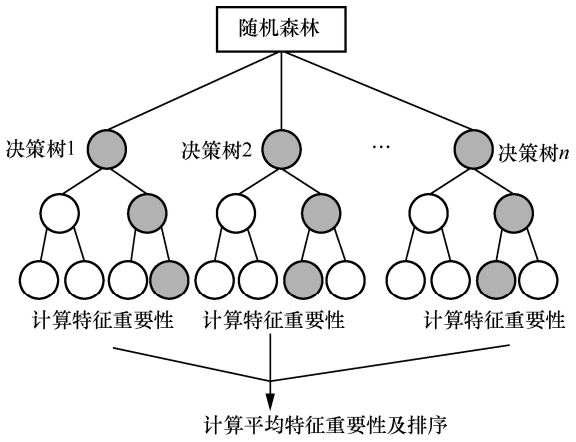

图 3　RF 特征重要性选择

置换重要性对应平均精确度减少量，对袋外数据（OOB）样本集的每个特征对应列的值进行

重排或添加噪声，然后比较 OOB 错误率在置换特征值前后的差值得到每个特征的重要性，对于网络流量这种大规模数据来说，这需要消耗较多的计算资源。

基尼重要性对应平均不纯度减少量。该方法通过判断在 RF 中，某个特征由于节点分裂所有树模型上基尼指数（Gini index）减少的平均值大小，判断特征的基尼重要性。即导致基尼指数减少的平均值越大的特征，基尼重要性越高。利用基尼指数计算特征重要性[13]计算速度快、抗扰动能力较强，故本文采用基尼指数进行重要性评估。

对数据集 F_i，Gini index 的计算式为：

$$\mathrm{Gini}(F_i)=1-\sum_{n=0}^{255}{F_{in}}^2,\ n\in[0,255] \tag{1}$$

其中，特征从十六进制转化为对应的十进制值后的取值范围是 0～255，n 表示该范围内的某一取值，F_{in} 表示所有节点中取值 n 所占的比例。

$\mathrm{GI_l}$ 和 $\mathrm{GI_r}$ 分别表示在节点 m 分支后产生的两个新节点的 Gini index，则第 k 个特征 C_k 在某棵树上的节点 m 分支前后的 Gini 变化量可以表示为：

$$\mathrm{VIM}_{km}^{(\mathrm{Gini})}=\mathrm{GI}_m-\mathrm{GI_l}-\mathrm{GI_r} \tag{2}$$

对于有 p 棵树模型的 RF，特征 C_k 的基尼指数重要性为 VIM_k，通过对其在 RF 中的所有树模型中的分支的重要性求平均求出：

$$\mathrm{VIM}_k=\frac{\sum_{p=1}^{p}\sum_{m\in M}\mathrm{VIM}_{km}^{(\mathrm{Gini})}}{\sum_{i=1}^{c}\mathrm{VIM}_i} \tag{3}$$

从训练集 S_c 中选择 s 个样本作为样本集，构建 RF 模型。从样本集 S_c 中选择 s 个样本作为训练集，构建 RF 模型。根据式（1）～式（3）的计算方法，计算每个特征 C_k 在 s 个样本上的 VIM_k，按重要性对特征进行排序生成数据集 C_{sort}，然后选取前 w 个特征 $C_{\mathrm{id}}=(c_{\mathrm{impl}},c_{\mathrm{imp2}},\cdots,c_{\mathrm{impw}})\leftarrow C_{\mathrm{sort}}$。$C_{\mathrm{impx}}$ 表示重要性排序，C_{id} 表示在原始流序列中特征出现的位置的集合。

1.3　轻量化 AlexNet 的网络流量分类模型

基于 AlexNet[14]改进模型的网络结构。AlexNet 将 ReLU、Dropout 和局部响应归一化（local response normalization，LRN）等策略成功地加入 CNN 中，也使用了 GPU 进行运算加速。AlexNet 分布在两个 GPU 上，每个 GPU 存储一半的神经元的参数，并且 GPU 之间的通信只在网络的某些层进行，控制了通信的性能损耗。但模型本身的网络深度较深和尺寸较大，导致模型参数多、计算复杂、耗时较长。对于深度学习网络存在显著的冗余，对模型进行压缩能达到和原模型相近或更好的分类结果，因此对模型结构进行优化，降低模型存储和运算成本，使模型在保证运算性能的情况下轻量化[15]。

AlexNet 模型对 1 000 类数据进行分类，参数量达 6 000 万个，参数主要集中在卷积层的卷积核个数和全连接层的大量神经元上，故对卷积层 Conv1、Conv2 以及全连接（fully connected，FC）层进行修改，减少模型参数。由于本文使用二维特征作为输入，将 AlexNet 改为单通道模式，修改输入特征大小，将第一层 11×11 的卷积核改为 5×5，并对对应的步长 stride 进行调整。为减少模型计算量并防止模型过拟合，仅保留一个 FC 层，将输出维度从 2 048 修改为 256，再添加一个 Dropout 层。由于本实验进行二分类，将归一化指数函数（softmax）输出层的输出改为 2。改进的 AlexNet 网格结构如图 4 所示。

1.4　模型优化

对网络进行优化提高模型的学习效率和泛化能力。训练神经网络时常用的优化方法有梯度估计修正和调整学习率，本模型使用 AdamW[16]优化器进行梯度修正，使用循环学习率的方法调整学习率。

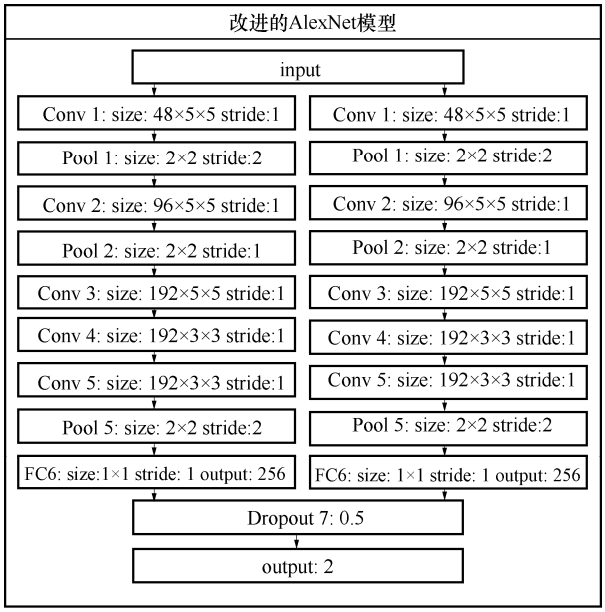

图 4　改进的 AlexNet 网格结构

文献[16]指出，在使用 Adam 优化器时，L2 正则化和权重衰减不完全一致，并提出了 AdamW 优化器。AdamW 实现 Adam 与权重衰减（weight decay）共同使用时的解耦，使权重衰减超参数的选择独立于学习率衰减策略，在使用 Adam 算法的同时，在全局采用一定的学习率衰减方法，可以显著提高 Adam 的性能。

Adam 算法[17]的主要计算过程如下。

M_t 和 G_t 分别表示该算法更新梯度的一阶矩（均值）和二阶原始矩（有偏方差）。参数 β_1 和 β_2 控制两个移动平均的指数衰减率。M_t 和 G_t 的更新式如下。

$$M_t = \beta_1 M_{t-1} + (1-\beta_1) g_t \tag{4}$$

$$G_t = \beta_2 G_{t-1} + (1-\beta_2) g_t^2 \tag{5}$$

然后对其进行偏差修正：

$$\widehat{M}_t = \frac{M_t}{1-\beta_1^t} \tag{6}$$

$$\widehat{G}_t = \frac{G_t}{1-\beta_2^t} \tag{7}$$

α 表示学习率，则 Adam 算法的参数更新式如下。

$$\theta_t = \theta_{t-1} - \frac{\alpha}{\sqrt{\widehat{G}_t + \varepsilon}} \widehat{M}_{t-1} \tag{8}$$

Adam 算法和 L2 正则化一起使用时，是通过在损失项引入 L2 正则项实现权重衰减。而 AdamW 是通过在损失函数更新的过程中额外引入一个权重衰减系数 λ 实现权重衰减，即权重衰减发生在 Adam 参数更新之后，乘以学习率之前。AdamW 的参数更新式如下。

$$\theta_t = \theta_{t-1} - \frac{\alpha}{\sqrt{\widehat{G}_t + \varepsilon}} \widehat{M}_{t-1} - \alpha\lambda\theta_{t-1} \tag{9}$$

循环学习率（cyclic learning rate，CLR）[18]方法使学习率在合理边界值之间循环变化，能帮助模型跳出成本函数的局部最小值，使模型有较快的收敛速度。文献[18]对常见数据集上不同学习率的性能表现进行实验分析和对比，实验结果表明，循环学习率可以在模型训练速度和精度上取得较好的平衡。

本文采用三角循环学习率（cyclic learning rate），并采取不同的周期循环幅度调整策略进行对比。则第 t 次迭代的学习率 α_t 可以表示为：

$$\alpha_t = \alpha_{\min} + (\alpha_{\max} - \alpha_{\min}) \cdot \max(0, (1-x)) \cdot f \tag{10}$$

其中，$\alpha_{\max}$ 和 $\alpha_{\min}$ 分别表示 CLR 的上界和下界，f 为循环学习率函数。3 种循环学习率策略如图 5 所示，TRI、TRI2 和 TRIEXP 分别表示对每周期循环幅度恒定、每周期后将循环幅度减半、每周期后将循环幅度呈指数衰减。

f 表达式如下。

$$f = \begin{cases} 1, & \text{TRI} \\ \dfrac{1}{2^{x-1}}, & \text{TRI2} \\ \alpha^x, & \text{TRIEXP} \end{cases} \tag{11}$$

x 的计算式为：

$$x = \left| \frac{t}{\Delta T} - 2\left\lfloor 1 + \frac{t}{2\Delta T} \right\rfloor + 1 \right|, x \in [0,1] \tag{12}$$

另外，本文采用指数衰减学习率（exponential decay）与上述循环学习率策略进行对比。

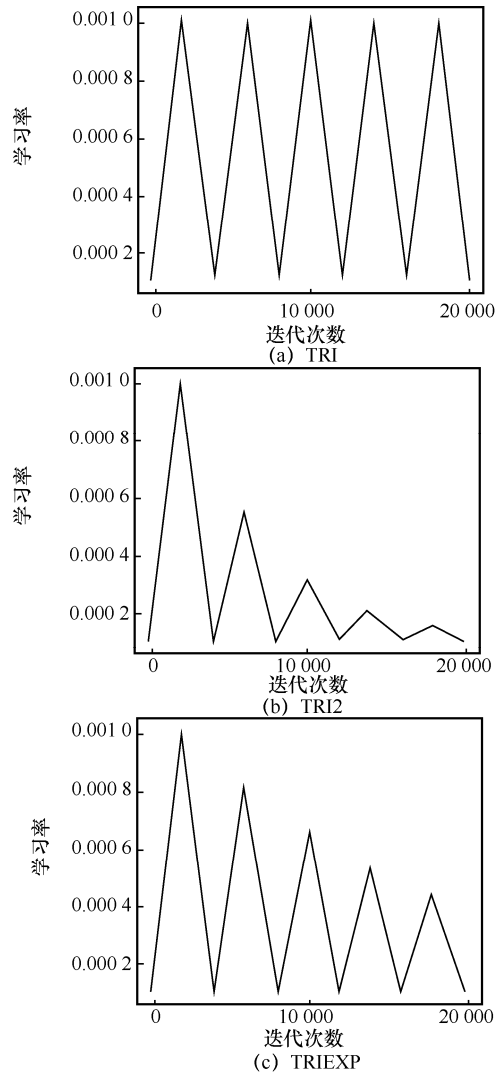

图 5　3 种循环学习率策略

1.5　分布式部署

TensorFlowOnSpark 是雅虎（Yahoo）开发的一个开源框架，对现有的 TensorFlow 框架进行少量修改，就可以实现 TensorFlow 集群在 Spark 平台上的服务部署，并且支持所有的 TensorFlow 功能，包括同步/异步学习、模型并行化、数据并行化等。此外，通过将 TensorFlow 的关键特性与大型数据框架相结合，可以实现 GPU 和 CPU 服务器集群上可拓展的分布式深度学习。

本文选择 TensorFlowOnSpark 进行分布式架构。与 TensorFlow 原生的分布式解决方案相比，TensorFlowOnSpark 充分利用 Spark 的弹性分布式数据集（resilient distributed dataset，RDD）在数据并行化和集群节点映射方面的特性，解决了集群间数据传输的问题，实现了应用进程的自动管理。在对大规模网络流量分类的场景下，这一分布式架构可以有效缓解网络流量、数据容量的增大导致的主存容量不足和特征变量太多导致的搜索空间过大这两个问题。并且，由于分布式架构具有灵活的体系结构、较高的容错性和较好的可拓展性，在实际应用中能够加强对庞大数据流量的管理和提高网络资源的利用率。

TensorFlowOnSpark 运行机制示意图如图 6 所示。首先，通过 Spark 的驱动应用程序（Spark Driver）传递配置参数和访问 Spark 的服务，SparkContext 掌控 Spark 的生命周期，但不参与 Tensor 内部通信和计算，具体任务由每个执行器（Spark Executor）完成。每个 Spark Executor 启动 TensorFlow 应用程序，类似一个独立 TensorFlow 集群，拥有一个参数服务器（parameter server，PS）和多个 Worker 节点，远程过程调用（g remote procedure call，gRPC）或远程直接内存访问（remote direct memory access，RDMA）表示使用的通信协议类别。此外，TensorFlowOnSpark 接受数据的方法有直接从 Hadoop 分布式文件系统（Hadoop distributed file system，HDFS）文件中读取数据和接受 Spark 的 RDD 分配。

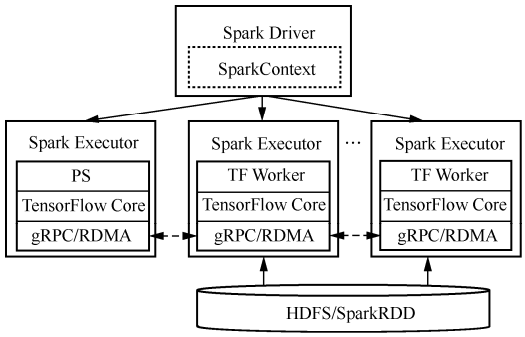

图 6　TensorFlowOnSpark 运行机制示意图

2　实验过程

2.1　实验环境

本实验在 3 台 Ubuntu 系统的计算机上搭建

Spark 完全分布式集群。首先，搭建分布式环境，设置一个主节点和两个从节点，分别命名为 sparkmaster、sparkslaver 1、sparkslaver 2。然后，安装 Hadoop 和 Spark，部署 Hadoop 相关服务包括 HDFS、资源调度系统（yet another resource negotiator，YARN），最后安装 TensorFlow，配置相应的 TensorFlowOnSpark 框架，分布式集群信息见表 1。使用的软件及版本主要有 Ubuntu 18.04.4 LTS（master）、Ubuntu19.04 LTS（slaver）、Spark 2.7.1、Hadoop 3.0.0、Java 1.8.0、Tensor 2.3.0、TensorFlowOnSpark 2.2.1。

2.2 数据预处理

本实验采用的网络流量数据集为 USTC-TFC2016[3]，该数据集以 pcap 文件格式存储，包含 10 种正常流量和 10 种恶意流量。

根据文献[5]，正常流量与异常流量之间字节信息在前 1 500 个特征差异很大，故将前 1 500 个特征作为待选择的特征。通过数据预处理模块，对每种类别的构造特征进行可视化后，部分实例网络流可视化如图 7 所示，前两排为正常流量，后两排为异常流量。可以发现不同类别的流量有较明显的可分性。

然后将异常流量标签设置为“1”，正常流量标签设置为“0”，选取每种流量 6 000 个样本，最终生成 60 000 个正常流量样本，60 000 个异常流量样本，共计 120 000 个流量样本的数据集。将处理后的数据集按 9:1 划分为训练集、测试集。

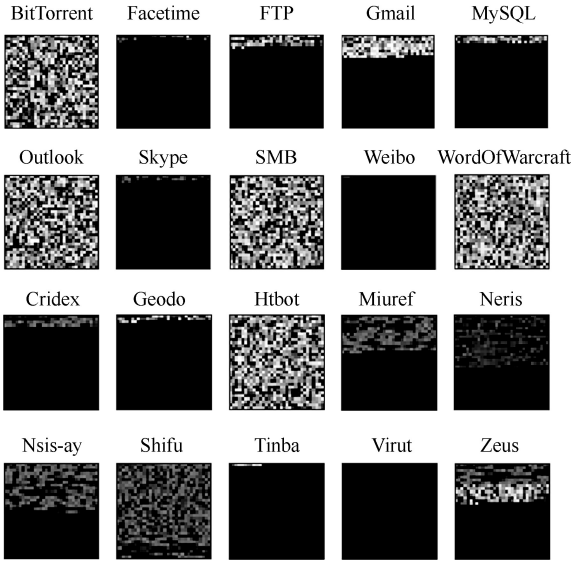

图 7 部分实例网络流可视化

2.3 特征选择

通过第 1.2 节中的方法得到特征重要性排名，列举重要性排名前十的特征，特征重要性排名见表 2。C_imp 表示重要性排序，C_id 表示在原始流序列中特征出现的位置，V_imp 表示特征重要性的取值，如第一行表示重要性排序为 1 的特征在原始流序列中出现位置是 0，其重要性评估值为 0.060 4。

对较重要特征的分布位置进行分析。以 C_id 为横坐标，V_imp 为纵坐标，绘制散点图，重要特征分布情况如图 8 所示。从图 8 可以看出，较为重要的特征分布在不同位置，如果直接选取前 1 024 byte 会丢失部分重要信息，对模型性能造成一定的影响，因此特征选择是有必要的。

表 1 分布式集群信息

对比项	sparkmaster	sparkslaver 1	sparkslaver 2
Spark	Master	Worker	Worker
HDFS	NameNode DataNode	DataNode	SecondaryNameNode DataNode
YARN	NodeManager	ResourceManager NodeManager	NodeManager

注：Master 为 Spark 主节点，Worker 为 Spark 工作节点；NameNode 为 HDFS 管理节点，DataNode 为 HDFS 工作节点，SecondaryNameNode 为辅助管理节点；ResourceManager 为 YARN 管理节点，NodeManager 为 YARN 工作节点。

表 2　特征重要性排名

C_imp	C_id	V_imp
1	0	0.060 4
2	4	0.038 7
3	3	0.027 0
4	5	0.021 8
5	2	0.020 4
6	7	0.018 6
7	1	0.016 1
8	1 455	0.013 9
9	1 448	0.013 8
10	9	0.012 8

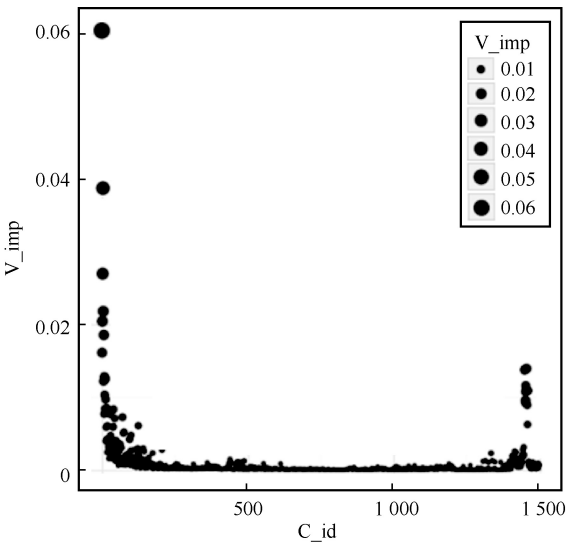

图 8　重要特征分布情况

对较重要特征的选择数量进行分析。对特征按重要性进行排序，以 C_imp 为横坐标，对应 V_imp 为纵坐标，对特征重要性进行可视化，特征重要性及排序如图 9 所示。可以看出，排名 400 以后的重要特征的重要性基本一致，因此本实验在前 1 500 个特征中分别选择最重要的 400、256、100 和 64 个特征作为输入进行比较。另外，本实验再在前 1 500 个特征中选择最重要的 1 024 个特征作为输入，与直接使用原始的前 1 024 个特征[5]形成对比。

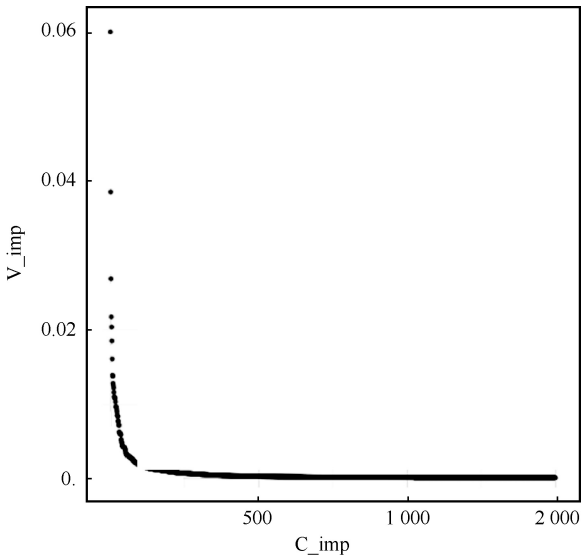

图 9　特征重要性及排序

2.4　评价指标

在本实验中，将恶意流量作为正例，正常流量作为反例。采用准确率 A、精确率 P、召回率 R、F1 值作为评价指标。将预测结果用符号定义为真正例（TP）、假正例（FP）、假反例（FN）、真反例（TN）。其中 TP 表示实际为正例，预测结果也为正例；FN 表示实际为正例，预测结果为反例；FP 表示实际为反例，预测结果为正例；TN 表示实际为反例，预测结果为反例。

（1）准确率

准确率是分类问题最基本的评价指标，指预测结果中分类正确的样例占样本总数的比例，其表达式为：

$$A=\frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FN}} \tag{14}$$

（2）精确率和召回率

精确率和召回率通常可以较有效地评估分类性能。准确率指预测结果中预测为正的样例中所含的真正例的比例。

$$P=\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}} \tag{15}$$

召回率是正确预测为正的占所有正例样本的比例。

$$R=\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} \tag{16}$$

（3）F1 值

F1 值是精确率和召回率的调和平均值，此时精确率和召回率权重是相等的。

$$\mathrm{F1}=\frac{2\times\mathrm{TP}}{2\times\mathrm{TP}+\mathrm{FP}+\mathrm{FN}}=\frac{2\times P\times R}{P+R} \tag{17}$$

2.5 实验结果和分析

对 CNN 模型输入层进行修改。对于随机森林算法选择的 1 024、400、256、100、64 个特征，分别处理为 32×32、20×20、16×16、10×10、8×8 大小的图像作为输入。每批次数据输入量设置为 50，即 batch size=50，训练次数设置为 2，即 epoch=2，对卷积核采用 He 初始化，选用 AdamW 优化器，采用三角循环学习率策略 TRI。通过多次实验，初始学习率同时也是循环学习率的最小值设置为 0.000 1，循环学习率的最大值设置为 0.001 5 时模型有较好的性能。用随机森林算法做特征选择，和直接选用原始字节前 1 024 个特征的原模型比较，不同特征选择方法性能指标及训练时间对比见表 3。

实验结果表明，本文提出的特征选择方法选择 1 024 个特征与使用原始字节前 1 024 个特征的方法相比，准确率提高了 7.26%；当选择特征个数减少为 256 时，准确率提高了 4.61%，模型训练时间减少了约 40%；而当选择特征个数减少为 64 时，准确率降低了 3.22%。

综合考虑计算资源和计算效率，使用随机森林选取 256 个特征作为输入。采用不同学习率循环幅度变化策略对模型进行优化。不同学习率性能指标及训练时间对比见表 4，其中 EXP 表示采用指数衰减学习率。

实验结果表明，采用 TRI2，即三角循环学习率每个周期后将循环幅度减半的迭代方式的效果最佳，因为其准确率和 F1 值最高，而精确率和召回率较为均衡，且模型训练时间比 TRI 策略进一步减少。

3 结束语

本文提出一种随机森林算法结合改进的卷积神经网络分类方法，该方法能有效进行网络流量分类、提高模型性能、减少运行时间。在大规模网络流量的条件下，可以为提高流量分类模型性能提供一定的理论依据，可以在流量检测与识别方面得到应用，具有一定的推广价值。在将来的工作中，可以对组合特征提取方法和特征融合的方法进一步展开研究，以及继续研究提升运算效率和泛化能力的方法，提升算法性能。

表 3 不同特征选择方法性能指标及训练时间对比

特征选取方法	准确率	精确率	召回率	F1 值	训练时间/min
原始字节前 1 024 个特征[5]	90.42%	89.47%	91.62%	90.53%	—
随机森林选择 1 024 个特征	97.68%	97.94%	97.42%	97.68%	97
随机森林选择 400 个特征	95.84%	94.95%	96.83%	95.88%	56
随机森林选择 256 个特征	95.03%	92.35%	98.20%	95.19%	52
随机森林选择 100 个特征	94.22%	94.19%	94.25%	94.22%	48
随机森林选择 64 个特征	87.20%	92.27%	81.20%	86.38%	26

表 4 不同学习率性能指标及训练时间对比

学习率（括号中为取值范围）	准确率	精确率	召回率	F1 值	训练时间/min
EXP（0.001）	93.77%	93.09%	94.55%	93.81%	41
TRI（0.000 1～0.001 5）	95.03%	92.35%	98.20%	95.19%	52
TRI2（0.000 1～0.001 5）	95.25%	95.83%	94.61%	95.21%	44
TRIEXP（0.000 1～0.001 5）	94.24%	94.59%	93.85%	94.22%	44

参考文献：

[1] 顾玥, 李丹, 高凯辉. 基于机器学习和深度学习的网络流量分类研究[J]. 电信科学, 2021, 37(3): 105-113.
GU Y, LI D, GAO K H. Research on network traffic classification based on machine learning and deep learning[J]. Telecommunications Science, 2021, 37(3): 105-113.

[2] 冯文博, 洪征, 吴礼发, 等. 网络协议识别技术综述[J]. 计算机应用, 2019, 39(12): 3604-3614.
FENG W B, HONG Z, WU L F, et al. Review of network protocol recognition techniques[J]. Journal of Computer Applications, 2019, 39(12): 3604-3614.

[3] WANG W, ZHU M, ZENG X W, et al. Malware traffic classification using convolutional neural network for representation learning[C]//Proceedings of 2017 International Conference on Information Networking (ICOIN). Piscataway: IEEE Press, 2017: 712-717.

[4] FENG W B, HONG Z, WU L F, et al. Network protocol recognition based on convolutional neural network[J]. China Communications, 2020, 17(4): 125-139.

[5] SUN Y L, YUN B S, QIAN Y G, et al. A Spark-based method for identifying large-scale network burst traffic[J]. Journal of Computers, 2021, 32(4): 123-136.

[6] TONG V, TRAN H A, SOUIHI S, et al. A novel QUIC traffic classifier based on convolutional neural networks[C]//Proceedings of 2018 IEEE Global Communications Conference (GLOBECOM). Piscataway: IEEE Press, 2019: 1-6.

[7] HU X Y, GU C X, WEI F S. CLD-net: a network combining CNN and LSTM for Internet encrypted traffic classification[J]. Security and Communication Networks, 2021: 1-15.

[8] 于帅, 董育宁, 邱晓晖. 一种基于深度特征融合的网络流量分类方法[J]. 南京邮电大学学报(自然科学版), 2022, 42(3): 82-89.
YU S, DONG Y N, QIU X H. A network traffic classification method based on deep feature fusion[J]. Journal of Nanjing University of Posts and Telecommunications (Natural Science), 2022, 42(3): 82-89.

[9] 薛靖靓, 陈迎春, 李鸥. 未知流量数据的智能特征提取与实时分类识别算法[J]. 信息工程大学学报, 2021, 22(5): 597-605.
XUE J L, CHEN Y C, LI O. Intelligent feature extraction and real-time identification algorithm for unknown traffic data[J]. Journal of Information Engineering University, 2021, 22(5): 597-605.

[10] MARÍN G, CAASAS P, CAPDEHOURAT G. DeepMAL - deep learning models for malware traffic detection and classification[C]//Data Science – Analytics and Applications. Wiesbaden: Springer Vieweg, 2021: 105-112.

[11] REIS B, MAIA E, PRAÇA I. Selection and performance analysis of CICIDS2017 features importance[C]//International Symposium on Foundations and Practice of Security. Cham: Springer, 2020: 56-71.

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

[13] 陈卓, 吕娜. 基于随机森林和XGBoost的网络入侵检测模型[J]. 信号处理, 2020, 36(7): 1055-1064.
CHEN Z, LYU N. Network intrusion detection model based on random forest and XGBoost[J]. Journal of Signal Processing, 2020, 36(7): 1055-1064.

[14] HE K M, ZHANG X Y, REN S Q, et al. Deep residual learning for image recognition[C]//Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE Press, 2016: 770-778.

[15] 甘众远. 基于深度学习的轻量化恶意流量识别及其分布式方法的研究与实现[D]. 南京: 南京邮电大学, 2021.
GAN Z Y. Research and implementation of lightweight malicious traffic identification and its distributed method based on deep learning[D].Nanjing: Nanjing University of Posts and Telecommunications, 2021.

[16] LOSHCHILOV I, HUTTER F. Decoupled weight decay regularization[J]. arXiv preprint, 2017, arXiv: 1711.05101.

[17] KINGMA D P, BA J. Adam: a method for stochastic optimization[J]. arXiv preprint, 2014, arXiv: 1412.6980.

[18] 刘云飞, 张俊然. 深度神经网络学习率策略研究进展[J]. 控制与决策, 2022: 0147.
LIU Y F, ZHANG J R. Research advances in deep neural networks learning rate strategies[J]. Control and Decision, 2022: 0147.

[作者简介]

云本胜（1980− ），男，博士，浙江科技学院理学院副教授，主要研究方向为大数据分析与挖掘和机器学习。

干潇雅（2000− ），女，浙江科技学院理学院在读，主要研究方向为大数据分析。

钱亚冠（1976− ），男，博士，浙江科技学院理学院教授，主要研究方向为深度学习、人工智能安全、大数据处理。