中国科学: 信息科学 2021 年 第 51 卷 第 1 期: 56–74
SCIENTIA SINICA Informationis
论文

基于改进深度卷积神经网络的网络流量分类方法

张小莉[1,2*], 程光[2], 张慰慈[2]

1. 山西铁道职业技术学院智能控制系, 太原 030013
2. 东南大学网络空间安全学院, 南京 211189
* 通信作者. E-mail: xiaoli2408@163.com

收稿日期: 2019–09–27; 修回日期: 2020–01–14; 接受日期: 2020–03–07; 网络出版日期: 2020–12–25

国家重点研发计划 (批准号: 2018YFB1800600) 资助项目

摘要 机器学习方法对网络流量分类的前提是假设流量具有独立同分布性, 而实际情况下流量特征不断发生变化, 导致该方法在处理海量、不具备独立同分布的流量数据时开销较大, 计算复杂度较高, 精度较低. 针对上述问题, 本文提出一种新的分类模型. 该模型将 PCA 算法与改进的深度卷积神经网络分类模型 (improved deep LeNet-5 convolutional neural networks, LCNN) 相结合进行流量分类. 前者进行降维分析, 发现影响检测精度的关键特征, 后者采用自主特征学习方式提升分类精度. 实验表明, 本文方法的内存开销较之前方法降低了 3.2%, 检测精度提升了 5%∼8%.

关键词 网络流量分类, 深度卷积神经网络, PCA, 多分类器, 特征选择, Tensorflow

1 引言

网络流量分类的技术研究在很多领域都得到了广泛关注. 具体包括网络性能监控、用户行为分析、差分服务质量 (quality of service) 保障、网络带宽资源管理和异常检测等 [1]. 近年来, 传统的基于端口的和基于深度包检测的分类方法已被认为具有较低的检测精度, 这是由于越来越多的应用程序在通信过程中使用相同的端口号进行数据传输. 如文献 [2] 采用动态的分类技术使用非固定的 TCP 和 UDP 端口号克服传统传输架构的性能限制.

基于深度包检测 (deep packet inspection, DPI) 的流量分类方法有效提高了分类精度. 该方法不仅要检测 IP 包头和 TCP/IP 包头, 还会针对数据包的部分和全部负载内容进行检测 [3]. 尽管该方法有效提高了检测精度, 但是仍具有以下缺陷: (1) 难用于加密负载流量; (2) 算法的复杂度较高, 效率较低 [4].

为了应对上述困难, 机器学习技术广泛应用到流量分类中. 具体方法包括浅层学习和深度学习. 在浅层学习中, 研究人员主要利用 artificial neural network [5]、k-nearest neighbour [6]、支持向量机 [7,8] 和决策树 [9] 对网络流量进行二元分类, 即区分出正常的和恶意的网络流量. 另外, 研究人员还利用

引用格式: 张小莉, 程光, 张慰慈. 基于改进深度卷积神经网络的网络流量分类方法. 中国科学: 信息科学, 2021, 51: 56–74, doi: 10.1360/SSI-2019-0213
Zhang X L, Cheng G, Zhang W C. Network traffic classification method based on improved deep convolutional neural network (in Chinese). Sci Sin Inform, 2021, 51: 56–74, doi: 10.1360/SSI-2019-0213

genetic algorithm [10]、多分类的支持向量机等识别不同类型的网络流量, 如: DoS, Probe, U2R 和 R2L 等. 另一种思路就是利用深度学习进行流量分类. 如文献 [4,11] 提出半监督的深度置信网络模型, 构造 P2P 流量合适的特征空间, 进行流量分类. 这不仅证明了深度学习在流量分类方面的可行性, 同时还表明其可通过微调原有最优模型生成新的最优分类模型完成流量分类, 这往往是浅层机器学习方法无法做到的. 本文的研究思路更接近后者, 即: 建立深度学习模型进行不同类型的网络流量识别. 研究发现, 浅层学习方法主要存在以下几个困难: (1) 类不平衡, 即: 一些类别的流数目明显多于其他类别, 而一些机器学习算法的前提是假设样本是均匀分布的, 这就导致有些小类别分类准确度极低. (2) 最优特征子集难以确定, 导致分类算法得不到最大的检测精度. (3) 浅层机器学习方法的过度拟合问题致使算法在特定数据集上得到较好的分类模型, 却不适用于其他数据集. (4) 概念漂移, 即流量的特征或类别随着时间的变化而变化.

基于上述分析, 为了解决浅层机器学习中面临的一些问题, 本文提出一种新的分类模型, 即将 PCA (principal components analysis) 算法与改进的深度卷积神经网络相结合进行流量分类. 模型中的 PCA 算法对初始特征向量进行降维, 发现影响检测精度的主要特征, 有效提高了检测效率. 而改进的 LCNN (improved deep LeNet-5 convolutional neural networks) 分类模型无需假定流量具备独立同分布的特点, 实现了对海量网络流量数据的精准分类. 本文使用公开数据集 CICIDS 2017[1)], KDD 99 及 CERNET 的真实流量对 LCNN 分类性能进行评估, 得出本文算法在计算精度和检测效率方面具有较大的优势.

本论文主要贡献如下:

(1) 提出一种新的分类模型: 将 PCA 算法与卷积神经网络相结合进行网络流量分类. 实验过程中采用两类公开数据集和实测数据集进行评估, 实验表明, 本文算法具有较高的检测精度, 比其他同类算法提高了 8%.

(2) 通过 PCA 分析, 发现了影响检测精度的关键特征集合, 极大提高了算法的计算速度. 同时, 算法运行时的内存开销降低了 3.2%, 在效率方面有很大改进.

(3) 在相同数据集上对比一维、二维卷积神经网络模型和深度循环神经网络模型 (recurrent neural network, RNN) 的分类精度, 发现本文算法的检测精度和效率优于其他两种模型.

本论文包括以下部分: 第 2 节论述相关工作, 第 3 节阐述本文提出的流量分类模型, 第 4 节描述实验数据集、实验过程及结果分析, 第 5 节进行总结和展望.

2 相关工作

流量识别是指通过不同的特征、标准识别和感知 IP 流记录, 实现对其所属类型的分类. 由此可见流量识别是一种分类问题, 即根据流量的特征属性确定流量的归属.

目前, 流量识别问题的研究主要包括多类流量分类方法研究和兴趣流量识别研究 [12~14]. 采用的技术手段主要包括基于端口的分类技术、基于报文负载的分类方法和统计学习方法. 本文的研究思路更贴近使用统计学习方法进行不同类型的网络流量分类, 实质是一个多分类问题. 基于上述研究, 我们如下阐述流量分类的相关工作, 并分析它们存在的不足, 导出本文的研究动机.

基于端口号的流量分类方法是根据 IP 报文中源、宿端口的取值进行应用类别识别. 在互联网发展初期, 该方法具有较高的检测精度, 随着网络的发展, 主机的通信可采用约定、随机端口、动态端口及伪装技术导致该技术几乎失效. 目前仅将该技术作为粗粒度的高带宽网络设备流量均衡的判断

1) Canadian Institute for cybersecurity. Intrusion detection evaluation dataset (CICIDS2017). http://www.unb.ca/cic/datasets/ids-2017.html.

依据 [15, 16].

基于负载的方法除了分析 IP 和 TCP/IP 包头, 还会对报文的负载进行检测. 该技术主要发现匹配网络应用对应的签名 (signature), 如: 在 http 报文中, 通常由命令紧跟 URL 和协议版本组成. 文献 [17] 使用一个会话的第 1 个报文进行 DPI 检测, 实现及早的分类, 但是这种技术除了要预先提取有效的签名外, 还要检测报文负载, 不仅导致计算开销大还会侵犯隐私. 为了解决 DPI 面临的一些问题, Khakpour 等 [18] 使用随机报文探测 (stochastic packet inspection, SPI) 技术 [12], 设计了一种快速算法, 该算法计算报文中第 1 个负载字节数的信息熵进行载荷内容类型识别. 但是该技术仍然可能涉及侵犯客户隐私的问题.

为了应对上述困难, 统计学习技术逐渐引起研究人员的关注, 并广泛应用到流量分类中. 它包括浅层学习技术和深度学习技术. 浅层学习主要统计不同应用流量的统计信息进行分类. 如: 文献 [19] 提出了一种高效的、鲁棒性的特征提取和选择方法进行流量分类. 该方法利用小波技术、主成分分析技术取得冗余和不相关的特征, 并用 SVM (support vector machine) 进行流量分类. de la Hoz 等 [20] 利用 PCA 和 SOM (self-organizing maps) 进行入侵流量的分类, 实验表明该方法具有较高的检测精度, 可实现近实时的检测. 除了上述成果外, artificial neural network [5]、k-nearest neighbour [6]、支持向量机 [7, 8] 和决策树 [9] 及 genetic algorithm [10] 等技术也得到了广泛应用, 并取得到较好的效果.

近年来深度学习在图像识别、语音识别、音频处理和自然语音处理等方面得到广泛应用并取得了不错的效果. 深度学习模型在训练数据发生变化时, 仅需对最优模型进行微调即可实现最优分类模型, 而无需对新数据进行重新特征提取和训练. 如: 文献 [4] 是基于神经网络的分类算法, 使用 WK-EML (wavelet kernel based Extreme learning machine) 结合 GA (genetic algorithm) 技术获取最佳神经网络参数值, 而非使用传统的凭借多次试验和经验的方式获取参数值. 该方法又结合极端学习机优化算法增加分类精度, 使精度达到 96.57%. 文献 [11, 21, 22] 基于 BP 神经网络对流量进行分类, 提高算法的灵活性和高效性. 但是 BP (back propagation) 神经网络的收敛速度较慢, 需要花费大量时间才能得到训练模型且容易陷入局部最优. 其次, BP 神经网络的泛化能力得不到保证, 需要根据经验或多次尝试才能确定网络参数.

由于概率神经网络上述缺陷, 文献 [23~27] 基于 CNN 的分类模型, 对网络流量进行分类, 并取得了较好的分类精度. 如: 文献 [26] 阐述一维 CNN 模型在流量分类识别中的具体应用, 数据集来源于 2017 年 CPCT 会议举办期间获取的流量. 作者主要对比分析了 SGD-Momentum, Adam, SGD, SGD-Nesterov, RMSprop 5 类梯度下降优化算法在一维 CNN 模型上的分类准确度, 指出梯度下降算法 SGD-Momentum 和 Adam 在一维 CNN 中的优化性能最佳. 文献 [27] 在物联网中用 CNN+RNN 的神经网络模型训练来源于 RedIRIS 的真实网络流量, 并进行了分类. 该文中, 作者分析了流量数据包的数目对分类精度的影响, 认为流包数在 5~15 之间就足以使得模型取得优秀的分类性能, 但没有提供 ROC 分类评价曲线.

通过上述分析, 本文分类模型能够消除不同算法对特征提取的不确定性, 极大地提高了网络流量的分类准确率, 精度可达 96.58%, 比同类方法检测精度提高了 5%~8%, 同时运用 PCA 算法又使模型运算复杂度大幅降低, 内存开销降低了 3.2%.

3 网络流量分类模型

本文网络流量分类模型由 3 个模块组成: 预处理模块, 主要功能是对数据进行过滤及简单特征提取; 降维分析模块, 主要寻找影响分类性能的关键特征; 分类模块, 利用深度卷积神经网络对网络流量分类, 如图 1 所示. 下面将主要以 CICIDS 2017 数据集为例详细描述 3 个模块的具体内容.

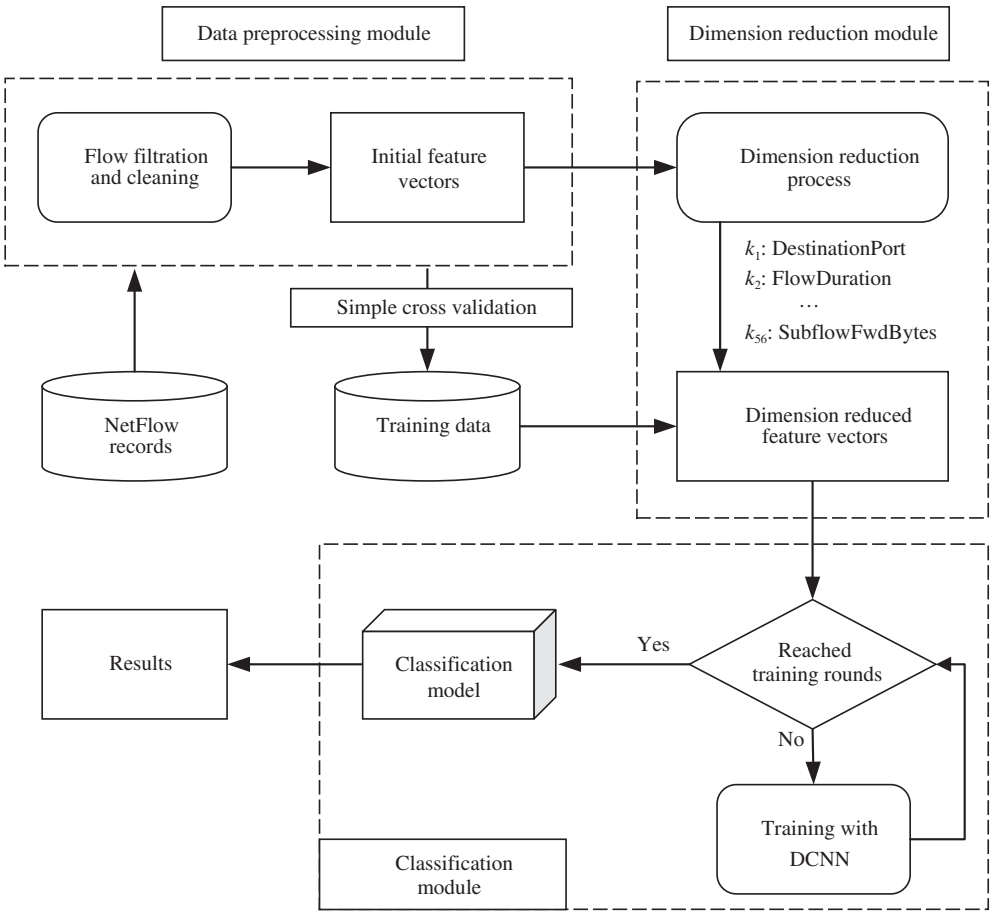

图 1 网络流量分类模型整体架构

Figure 1 The overview of network traffic classification scheme

3.1 数据预处理模块

数据预处理是将含有噪声、不完整、不一致的原始数据经过规约、变换等方式转换成恰当的适合本文分类模型的输入数据, 从而得到比较准确的分类结果.

在数据预处理模块, 经过执行流过滤、流清洗处理 CICIDS 2017 数据集, 每条流记录拥有 70 个特征属性. 把每一条流记录的特征属性映射为相同数目的像素点, 形成一个大小为 1×70 的矩阵块, 作为分类模型的一次输入. 按序对每条流记录编号, 形成与数据集样本数相同的矩阵块, 按编号顺序依次输入分类模型, 获得每个矩阵块 (即每条流记录) 的分类结果. KDD 99 和江苏省网边界实际流量数据集的输入形式类同于 CICIDS 2017.

由于篇幅有限, 本文未给出该数据集的 70 个特征属性描述, 而是给出了部分特征, 如表 1 所示. 为了避免数据集中由于量纲的不同而使某些特征属性对分类结果形成主导作用, 本文将输入数据统一作 MIN_MAX 归一化处理, 即将数据的所有特征属性映射到同一尺度上.

3.2 降维分析模块

本文在上述数据预处理模块的基础上对训练数据的初始特征向量进一步做降维处理, 发现影响检

表 1 CICIDS 2017 部分特征属性

Table 1 Some features of CICIDS 2017

Attribute	Features
Ports	DestinationPort
During time	FlowDuration
Total packets	TotalFwdPackets, TotalBackwardPackets
Total length of packets	TotalLengthofFwdPackets, TotalLengthofBwdPackets
Packet length parameter values	FwdpacketLengthMax, FwdPacketLengthMin FwdPacketLengthMean, FwdPacketLengthStd BwdPacketLengthMax, BwdPacketLengthMin BwdPacketLengthMean, BwdPacketLengthStd BwdHeaderLength, MinPacketLength, FwdHeaderLength MaxPacketLength, PacketLengthMean, PacketLengthStd PacketLengthVariance, FwdHeaderLength.1
Number of Bytes/packets per second	FlowBytes/s, FlowPackets/s, FwdPackets/s BwdPackets/s, FlowIATMean, FlowIATStd, FlowIATMax FlowIATMin, FwdIATTotal, FwdIATMean
Time between two packets	FwdIATStd, FwdIATMax, FwdIATMin, BwdIATTotal BwdIATMean, BwdIATStd, BwdIATMax, BwdIATMin
Packet count of each flag bit	FINFlagCount, SYNFlagCount, RSTFlagCount PSHFlagCount, ACKFlagCount, URGFlagCount CWEFlagCount, ECEFlagCount FwdPSHFlags, FwdURGFlags
Average	AveragePacketSize, AvgFwdSegmentSize AvgBwdSegmentSize
The average number of packets in a sub flow	SubflowFwdPackets, SubflowFwdBytes

测精度的关键特征.

通过比较 PCA, LLE (locally linear embedding) 和 Autoencoder 发现在这些算法中 PCA 是典型的线性降维算法, 适用于图形聚类、分类和回归模型. LLE 是典型的非线性降维算法, 对最近邻样本数的选择较敏感, 不同的最近邻数对最后的降维结果有很大影响. Autoencoder 是一种无监督学习算法, 也是一种神经网络结构, 用于解决多层神经网络的参数初始化问题. 由于 Autoencoder 可以学习出一个跟 PCA 结果非常相似的输入数据的低维数表示, 但是计算量较大, 考虑到整个 PCA 算法模型结构简单、高效的原则, 这两者之间我们选择 PCA 进行降维, 且 PCA 在 56 维度下精确度最高.

分析图 2, 比较相同纬度下使用 PCA 和 LLE 降维算法得出分类性能前者明显优于后者. 另外在时间开销方面, PCA 显著快于 LLE. 基于 PCA 的特征选择算法不仅有较好的选择效果, 更有很好的稳定性, 本文选择 PCA 降维.

PCA 可以从多方面分析主要影响因素, 简化复杂问题. 假设训练数据的特征属性个数为 m, 要降维的特征向量为 $1 \times m$, PCA 降维矩阵为 $C \in \mathbb{R}^{m \times k}$, 将特征向量矩阵与降维矩阵相乘得到 $1 \times k$, 矩阵维度即从 m 降低到 k ($k < m$).

运用 PCA 算法对 CICIDS 2017 数据集进行分析, 确定 56 个主成分能够解释 93.1% 的流量分类性

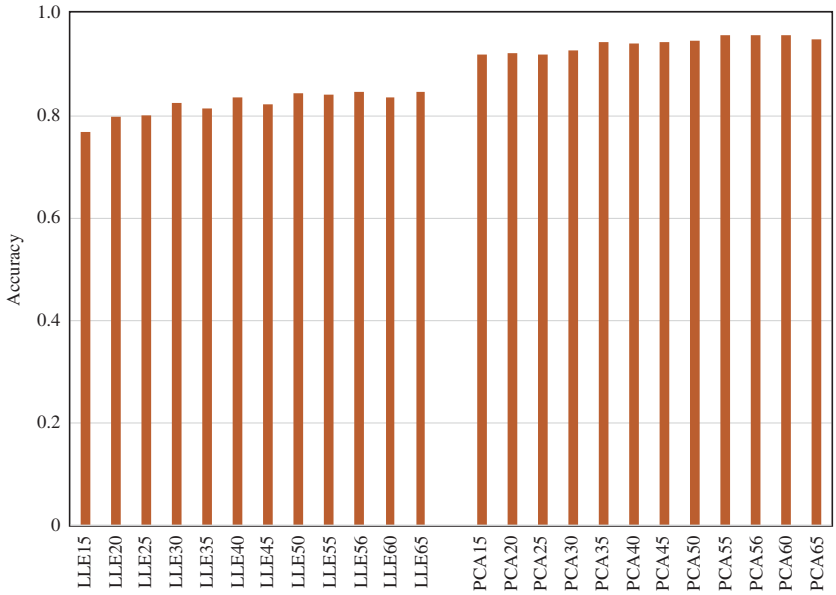

图 2 (网络版彩图) 相同维度下 PCA 和 LLE 准确性比较

Figure 2 (Color online) Accuracy comparisons of PCA and LLE in the same dimension

能差异. 分析这 56 个主成分得出表 1 中的前 56 个特征属性是影响分类的关键属性即数据包头长度、数据包总长度、数据包个数、每秒转发数据包数、两个数据包间隔等, 同时得出平均空闲时间等 14 个属性为次要因素. 又通过 PCA 实验对比及计算累计贡献率的方法最终确定本文 PCA 空间维数 K 为 56, 模型分类性能评估基于 PCA 的 56 维数据进行. 而对于数据集 KDD 99 和实测数据, 通过上述类似降维处理, K 分别为 19 和 67.

虽然降维会失去一部分特征属性, 对分类精度也有一定影响, 但差别仅为 0.0029, 相差甚微, 且降维后计算复杂度大幅降低, 极大地提高了计算速度, 分类模型运行时占用内存比未使用 PCA 时降低了 3.2%. 二者权衡考虑, 使用 PCA 降维数据的分类模型运算优势更明显. 因此本文将预处理后的数据进行 PCA 降维处理, 作为分类模型的输入数据.

为了验证主成分特征的合理性, 本文分析了一个具体的 DDoS [28] 攻击大类, 以及 DosGoldenEye, DoSHulk, DoSSlowhttptest, DoSslowloris, Heartbleed 等几个小类. 由于有些 DDoS 攻击者可以降低攻击速率, 使其流量速率接近正常的流量, 因此本文主成分中的流量持续时间、每秒转发字节/包数, 以及数据包总长度等相对有效. 而有些 DDoS, 在攻击中数据包头信息统计分布与正常情况不同, 因此, 计算数据包头部信息中的各标志位相对高效. 通过分析, 进一步证明了本文主成分特征的合理性.

3.3 分类模块

论文的核心方法是训练一个改进的深度卷积神经网络分类模型 LCNN (用 P 表示) 对输入数据进行训练, 得出分类结果. 假设 X 表示输入矩阵, S 表示输出类别, 本文工作是训练一个模型 $P(S|X)$ [16], 其中, $X=\{x_1,x_2,\ldots,x_n\}$ 代表一条样本的 n 个特征向量, $S=\{s_1,s_2,\ldots,s_m\}$ 代表输出的 m 类结果. 每条数据流量都有 m 种可能性, 因此在输出层用一个 softmax 分类器表示类别是可行的. softmax 分类器接收由模型 P 从 X 中提取的 n 个输入特征. 在模型测试阶段, 用

$$S(s_1,s_2,\ldots,s_m)=\text{argmax}_{a,x_1,x_2,\ldots,x_n}\times P(S|X) \tag{1}$$

来表示预测结果, a 表示模型对特征向量的随机选择参数. 本文检测算法如算法 1 所示.

算法 1 Network traffic classification algorithm

Input: IP netflow data, random initialized weight and the threshold value: w, $\delta \leftarrow$ the integer close to zero, define the accuracy ε.

```
Processing IP flows and computing the feature vector V;
Using PCA for dimension reduction V';
while not reach training steps N do
    for each sample V' do
        Forward compute the output vector;
        Compute the errors γ between output and ideal vector;
        if γ > ε then
            Feedback to the network, computing the errors γ, △w, △δ, and updating w and δ;
        else
            Forward compute and adjust the w and δ;
        end if
    end for
    Save the model and using it to test new data;
end while
return C = (C_1, C_2, ..., C_n);
```

Output: Different types of netflows: $C = (C_1, C_2, \ldots, C_n)$.

LCNN 主要由输入层、卷积层、下采样层 (也称池化层)、全连接层、输出层构成. 通常用一幅图像作为输入, 图像的像素点构成输入矩阵. 本文未对数据做图像转化处理, 直接以 3.1 小节的矩阵块作为输入层数据. 典型的 LeNet-5 是经典的 CNN 结构 [29], 依次由两个卷积层、两个池化层, 以及两个全连接层组成. 卷积层、池化层和全连接层 3 种结构经过各种排列组合就可构建一个完整的卷积神经网络 [17], 通常情况是卷积层和池化层交替叠加出现.

LCNN 模型用激活函数将线性关系转化成非线性关系, 经常使用的激活函数有 tanH, sigmoid 和 Relu. 使用 Relu 的深度卷积神经网络训练速度是 tanH 函数的几倍, 但 tanH 函数在处理数据过拟合问题上收效显著 [23], 对于本文 CICIDS 2017 如此巨大的数据量, 学习和网络收敛速度是一个至关重要的因素, 所以综合多方考虑并选择 Relu 函数作为本文分类模型的激活函数.

LCNN 模型使用 Dropout 函数防止过拟合及降低模型训练复杂度. 该函数在模型训练中随机让一定比例的隐层神经元输出为 0, 相当于把整个大网络拆分成无数个小网络训练, 训练完成以后再把网络进行合并, 输出结果相当于无数个子网络的 “平均” [30~32]. 通常设置 Dropout 值为 0.5, 本文经实验验证取 Dropout 值为 0.7 时模型精确度最高.

4 实验评估

本文实验主要包括 3 块内容, 首先介绍本文的数据集, 其次描述本文的实验模型, 最后分析并评估本文的实验结果. 如下将一一进行介绍.

4.1 数据集选取

本文 LCNN 模型是一种基于监督学习的分类模型, 要求数据集自带标签. 本文使用 3 种带标签数据集对模型实施性能测评. 流量分类目前面临着类不平衡、概念漂移、最优特征子集提取和过拟合

表 2 CICIDS 2017 流量类型分布 (1)

Table 2 The distribution of different traffic types in CICIDS 2017 (1)

Network traffic type	Training set records	Training set percentage (%)	Test set records	Test set percentage (%)
BENIGN	1591082	80.29	682015	80.31
Bot	1358	0.07	608	0.07
DDoS	89710	4.53	38317	4.51
DoSGoldenEye	7197	0.36	3096	0.36
DoSHulk	161867	8.17	69206	8.15
DoSslowloris	4088	0.21	1708	0.20
DoSSlowhttptest	3834	0.19	1665	0.20
FTP-Patator	5511	0.28	2427	0.29
Heartbleed	5	Very small	6	Very small
Infiltration	24	Very small	12	Very small
PortScan	111251	5.61	47679	5.61
SqlInjection	14	Very small	7	Very small
SSH-Patator	4044	0.20	1853	0.22
WebAttack	1532	0.08	627	0.07
Total records	1981517		849226	

表 3 CICIDS 2017 流量类型分布 (2)

Table 3 The distribution of different traffic types in CICIDS 2017 (2)

Network traffic type	Training set records	Training set percentage (%)	Test set records	Test set percentage (%)
BENIGN	1612046	81.35	691001	81.37
Bot	2316	0.12	1037	0.12
DDoS	77003	3.89	32890	3.87
DoSGoldenEye	7187	0.36	3092	0.36
DoSHulk	152003	7.67	64989	7.65
DoSslowloris	4123	0.21	1723	0.20
DoSSlowhttptest	3697	0.19	1606	0.19
FTP-Patator	5431	0.27	2392	0.28
Heartbleed	5	Very small	6	Very small
Infiltration	24	Very small	12	Very small
PortScan	112416	5.67	48137	5.67
SqlInjection	14	Very small	7	Very small
SSH-Patator	3768	0.19	1727	0.2
WebAttack	1484	0.07	607	0.07
Total records	1981517		849226	

方面的问题. 为了应对类的不平衡, 本文对数据重采样, 即通过随机采样样本, 平衡训练样本集中大类和小类之间的样本数量差距. 所以我们对 CICIDS 2017 数据集的每个子文件进行训练, 得到最优的训练模型, 然后合并这些子文件, 随机抽样 280 万条进行模型评估. 为了验证算法的性能, 随机两次提取

表 4　真实采集流量类型分布

Table 4　The distribution of different traffic types in real data

Flow type	Samples
Video streams	300
Audio streams	300
Browse the Webs	300
Pictures or voice files	300
Upload and download large files	300
Text-chat	300
VIDEO_CALLs	300
VOICE_CALLs	300

数据集, 每一次提取的结果如表 2 和 3 所示. 表 2 和 3 对训练集和测试集的划分方法采用了简单交叉验证, 即: 70% 的训练数据和 30% 的测试数据. 在对 KDD 数据集的选择过程中也采用类似的方法.

4.1.1　CICIDS 2017 数据集

CICIDS 2017 数据集来源于公开的 Canadian's Cybersecurity Hub 入侵检测评估数据集 (CICIDS 2017), 包含 3100 万条流量记录, 流量类型包括正常 (BENIGN) 和目前常见的各种新型攻击流量, 且已打标记.

本文仅将 CICIDS 2017 数据集作为流量分类模型测试数据集, 把正常流量和各种攻击流量当作不同流量类型进行分类准确率评估, 并不涉及有关正常流量、异常攻击流量 [33], 以及入侵检测的研究内容.

本文选取该数据集中 280 多万条流量记录进行模型性能评估, 经数据过滤处理并将样本数可忽略不计的部分流量进行整合, 最后得到 14 类流量, 具体参数值见表 2 和 3 两种不同的抽样.

4.1.2　KDD 99 数据集

KDD 99 数据集是从 DARPA1998 中提取 41 种人工设计的特征组成的数据集 [24]. 该数据集来源于一个模拟的美国空军局域网环境下采集的 9 个星期的网络数据, 也是 1999 年举行的 KDD 竞赛所采用的数据集. 本文选择其中 10% 的 corrected 测试样本作为第 2 类数据集, 共 311029 条样本.

4.1.3　真实采集流量数据集

本文所采用的第 3 种数据集是中国教育科研网江苏省网边界环境下用网络管理工具 TCPDUMP 采集的 2400 条网络流记录, 一共 8 类, 流量分布见表 4. 因实际流量采集困难, 又因时间关系用时半年仅得 2400 条可用流记录, 但在后续的扩展工作中会对实测数据持续采集, 建立更大的样本数据, 验证本文算法的正确性和合理性.

为了更好地评估模型, 本文分别对 CICIDS 2017、KDD 99 和中国教育科研网江苏省网边界的流记录数据集进行了十折交叉验证.

4.2　实验模型描述

本文分类模型 LCNN 是对经典 LeNet-5 模型的改进, 具体如下: (1) 使用 3 个卷积层、3 个池化层、2 个全连接层, 共 8 层. (2) 数据集 CICIDS 2017 中输出 14 个类别, 数据集 KDD 99 输出 38 个类

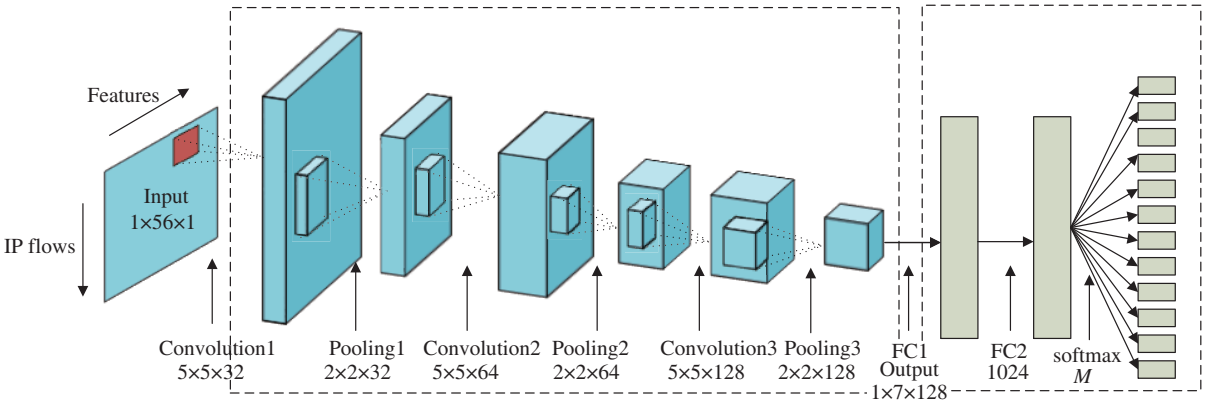

图 3 (网络版彩图) LCNN 模型结构图

Figure 3 (Color online) The structure of LCNN

别, 真实采集流量数据集输出 8 个类别. (3) 最后两个全连接层都使用 Dropout 函数防止数据过拟合, 每个卷积层后紧跟一个最大池化层, 如图 3 所示.

本文采用网络搜索的方法确定所有超参数. 每 1 层卷积层和连接层都用激活函数 Relu, 梯度下降优化算法使用 Adam 函数, 损失函数用交叉熵函数.

第 1 层卷积层的输入没有使用 CNN 惯用的图片格式, 而是将经过 PCA 降维算法处理的一个 1×56 的矩阵作为黑白图片的像素点来识别. 所以第 1 层输入为 1×56×1 , 最后的 1 是通道数, 卷积核数设为 32, 大小为 5×5, 步长 stride (卷积核在特征面上进行卷积运算时每次移动的步数) 为 1.

第 1 层卷积输出为 1×56×32, 32 个卷积核形成 32 个输出特征面. 随后在池化层做最大池化处理, 本文设置每个池化层的过滤器尺寸均为 2×2, 步长均为 2. 按照图像说法, 池化后特征面长宽成为原来的 1/2 (向上取整), 所以第 1 次池化后输出为 1×28×32. 第 2 层卷积接受上层池化层的输出, 设卷积核数为 64, 大小仍为 5×5, 所以输出变为 1×28×64, 卷积操作产生了 64 个特征面. 经随后池化层最大池化处理后输出变为 1×14×64. 第 3 次卷积运算的卷积核大小仍为 5×5, 卷积核数选 128, 输出为 1×14×128. 第 3 次池化后输出为 1×7×128. 然后与全连接层连接将张量数据平铺展开为 1×7×128 大小的向量数据. 再经过一个全连接层并运用 softmax 函数得出最后分类结果作为输出.

4.3 实验结果分析

本文实验模型基于 Google 开源软件框架 Tensorflow [34], 并用 Python 的 scikit-learn 库函数计算性能度量. 实验环境如下: 服务器型号为 Intel(R) Xeon(R), 内存为 16 GB RAM & 120 GB SSD ROM, CPU 是 16 核 E5-2609v4@1.70 GHz, 硬盘容量为 10 TB.

本小节将对模型实验结果进行分析比较, 分析对本文分类模型影响较大的几个超参数的选取, 比如: 模型结构变化、某些函数值对分类精度的影响等. 然后与其他相似文献的不同网络流量分类算法进行对比分析, 证明本文算法的优越性.

为了评估分类模型在分布不均的各类数据集上的分类性能, 本文使用分类模型的几个常用评价指标 TPR (true positive rate)、FPR (false positive rate)、Precision、Accuracy、Recall、AUC (area under ROC curve) 及 $F1$ [35, 36]. 对上述评价指标的定义基于机器学习二分类的 4 个基本指标, (1) true positives (TP_i), 实际为 i 类, 也被模型划分为 i 类的流量样本数. (2) false positives (FP_i), 实际不是 i 类, 但被模型划分为 i 类的流量样本数. (3) false negatives (FN_i), 实际为 i 类, 但被模型划分为其他

类的流量样本数. (4) true negatives (TN_i), 实际不是 i 类, 模型也没有将它划分为 i 类的流量样本数. 基于上述定义得到本文使用的评价指标, 其中 n 表示分类数, $i = 1, 2, \ldots, n$.

$$\mathrm{Accuracy} = \frac{\sum_i^n (\mathrm{TP}_i + \mathrm{TN}_i)}{\sum_i^n (\mathrm{TP}_i + \mathrm{FP}_i + \mathrm{TN}_i + \mathrm{FN}_i)}, \tag{2}$$

$$\mathrm{Precision}_i = \frac{\mathrm{TP}_i}{(\mathrm{TP}_i + \mathrm{FP}_i)}, \tag{3}$$

$$\mathrm{TPR}_i = \frac{\mathrm{TP}_i}{(\mathrm{TP}_i + \mathrm{FN}_i)}, \tag{4}$$

$$\mathrm{FPR}_i = \frac{\mathrm{FP}_i}{(\mathrm{FP}_i + \mathrm{TN}_i)}, \tag{5}$$

$$F1 = 2 \times \frac{(\mathrm{Precision} \times \mathrm{Recall})}{(\mathrm{Precision} + \mathrm{Recall})} \quad (\mathrm{Recall} = \mathrm{TPR}), \tag{6}$$

其中 TPR_i 表示模型识别为 i 类的样本占所有 i 类样本的比例, 越接近 1 分类模型性能越佳; FPR_i 表示模型识别为非 i 类占所有非 i 类的比例, 越接近 0 模型效果越佳. Accuracy 表示模型总体分类正确率, $\mathrm{Precision}_i$ 表示模型中第 i 类的分类准确率. 特别地由 TPR 和 FPR 构成的接受者操作特征曲线 ROC 被认为是分类模型最重要的性能评价指标 [37], 图像越靠近 (0,1) 代表模型分类性能越优秀. AUC 表示 ROC 曲线下方的面积, 其值介于 0.5~1.0 之间, 越靠近 1 模型越完美. $F1$ 分数是模型 Precision 和 Recall 的一种加权平均, 它的最大值是 1, 最小值是 0, 模型的 $F1$ 分数越接近 1, 分类性能越好.

4.3.1 模型结构变化对分类准确率的影响

为寻求本文最优分类模型 LCNN, 针对不同卷积层数的 DCNN, 以及同一卷积层数不同超参数的组合模型进行实验. 然后在 3 类数据集上对本文分类算法的分类准确率进行对比分析.

本文对比了 14 种 DCNN 分类模型, 分别以 Cnn3_A1, Cnn3_A2, Cnn3_B1, Cnn3_B2, Cnn3_C, Cnn3_D, Cnn3_E1, Cnn3_E2, Cnn3_F, Cnn3_G, Cnn4_H1, Cnn4_H2, Cnn5_I, Cnn2_J (LeNet-5) 各模型将卷积层数、池化层数、卷积核数、卷积核尺寸等超参数做不同排列组合. 其中 Cnn3_A1 和 Cnn3_A2 模型的卷积层数均为 3, 设各层卷积核数为 32, 64, 128, 大小分别为 3×3, 5×5, 3×3. 另 Cnn3_A1 模型有 3 个池化方式为 max 的池化层, 而 Cnn3_A2 模型有 2 个. Cnn4_H1 和 Cnn4_H2 模型具有 4 个卷积层, 同样 H2 比 H1 少 1 个池化层; Cnn5_I 具有 5 个卷积层、5 个池化层; Cnn2_J 模型即经典的 LeNet-5 模型.

表 5 表示上述部分模型结构. 表中用 CnnA 代表各模型名称, A 表示模型中的卷积层数. $\mathrm{Conv}(x, y, z, 1, s)$ 表示卷积运算, x 代表卷积核数, y 和 z 代表卷积核尺寸, l 代表卷积运算的步长, s 代表 "SAME" 填充方式. $\mathrm{MaxP}(x, y, n, s)$ 表示池化层采用最大池化方式, x, y 代表池化层过滤器尺寸, n 代表池化运算的步长, s 代表 "SAME" 填充方式. PCA 表示对特征向量进行降维分析, MM 表示对输入数据做 MIN_MAX 标准化处理, 将数据值限定在 0~1 范围内, 有利于提高模型分类精度. 实验结果证明, 模型结构的变化对数据集的分类精度有一定影响.

图 4 表示各模型结构在数据集 CICIDS 2017 上的总体分类正确率, 各类分类准确率、召回率, 以及 $F1$ 值. 观察图 4 结合实验数据分析得知拥有 3 个卷积层的各类模型性能区别微小, 其中最优秀的模型是 Cnn3_B1, 该分类模型具有 3 个卷积层和 3 个池化层, 具体参数见表 5. 对比使用 4 个卷积层的 Cnn4_H2 模型, 其总体分类正确率 0.9626 比 Cnn3_B1 模型的 0.9658 减少 0.0032, 且 Cnn4_H2 模

表 5 部分模型结构组合表

Table 5 Some model structure combination table

Model	Descriptions of concrete model of each layer
Cnn3_A1	PCA-MM-Conv(32, 3, 3, 1, s)-MaxP(3, 3, 3, s)-Conv(64, 5, 5, 1, s)-MaxP(2, 2, 2, s) -Conv(128, 3, 3, 1, s)-MaxP(3, 3, 3, s)
Cnn3_A2	PCA-MM-Conv(32, 3, 3, 1, s)-MaxP(3, 3, 3, s)-Conv(64, 5, 5, 1, s)-MaxP(2, 2, 2, s)-Conv(128, 3, 3, 1, s)
Cnn3_B1 (LCNN)	PCA-MM-Conv(32, 5, 5, 1, s)-MaxP(2, 2, 2, s)-Conv(64, 5, 5, 1, s)-MaxP(2, 2, 2, s) -Conv(128, 5, 5, 1, s)-MaxP(2, 2, 2, s)
Cnn4_H2	PCA-MM-Conv(32, 5, 5, 1, s)-MaxP(2, 2, 2, s)-Conv(64, 5, 5, 1, s)-MaxP(2, 2, 2, s) -Conv(128, 5, 5, 1, s)-MaxP(2, 2, 2, s)-Conv(256, 5, 5, 1, s)
Cnn5_I	PCA-MM-Conv(32, 5, 5, 1, s)-MaxP(2, 2, 2, s)-Conv(64, 5, 5, 1, s)-MaxP(2, 2, 2, s)-Conv(128, 5, 5, 1, s) -MaxP(2, 2, 2, s)-Conv(256, 5, 5, 1, s)-MaxP(2, 2, 2, s)-Conv(128, 5, 5, 1, s)-MaxP(2, 2, 2, s)
Cnn2_J (LeNet-5)	PCA-MM-Conv(32, 5, 5, 1, s)-MaxP(2, 2, 2, s)-Conv(64, 5, 5, 1, s)-MaxP(2, 2, 2, s)

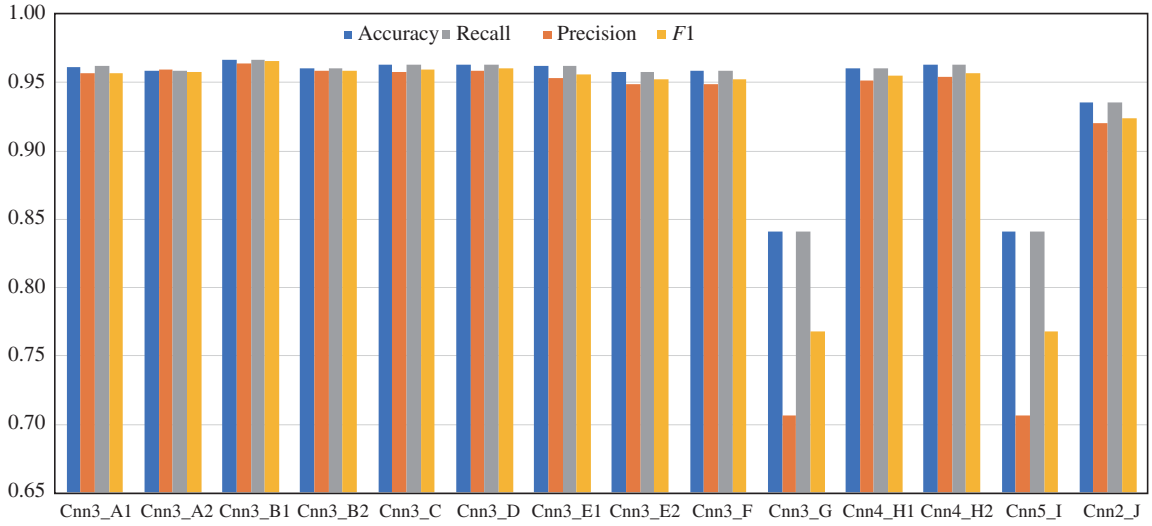

图 4 (网络版彩图) CICIDS 2017 模型结构分类指标评价比较图

Figure 4 (Color online) The evaluation results of different models in CICIDS 2017 dataset

型结构更复杂, 运行时间更长, 总体性能有所下降. 接着对比拥有 5 个卷积层的模型 Cnn5_I, 发现总体分类准确率仅 0.8405, 性能下降厉害, 因此没有继续对比卷积层数更深的模型.

另外, 分析图 4 中 Cnn3_G 模型, 得知当卷积层的卷积核尺寸变大到 7×7, 卷积核数增多到 128, 池化层过滤器尺寸变为 3×3 时, Accuracy 值下降到 0.8405, Recall 及 $F1$ 值也分别下降到 0.8405 和 0.7677, 同时模型运行速度变得极其缓慢.

综上对 14 种模型结构多方面的对比分析, 发现了本文最优分类模型 Cnn3_B1 模型 (具体描述见 4.2 小节), 其总体分类正确率比采用经典模型 LeNet-5 (Cnn2_J) 提高 5%.

实验结果证明: 第一, 卷积核数是影响模型分类性能的关键因素, 并不是特征面越多, 分类结果越优秀, 特征面的多少与数据输入形式关系密切. 对于图片、语音等输入形式, 特征提取越细得到的分类效果越好, 但对于特征数比较少 (如 56) 的矩阵块数据输入形式来说, 较少的卷积核数反而会得到更佳的效果, 如本文 LCNN 模型的第 1 层卷积层只有 32 个卷积核数, 而分类性能却表现良好. 本文算法

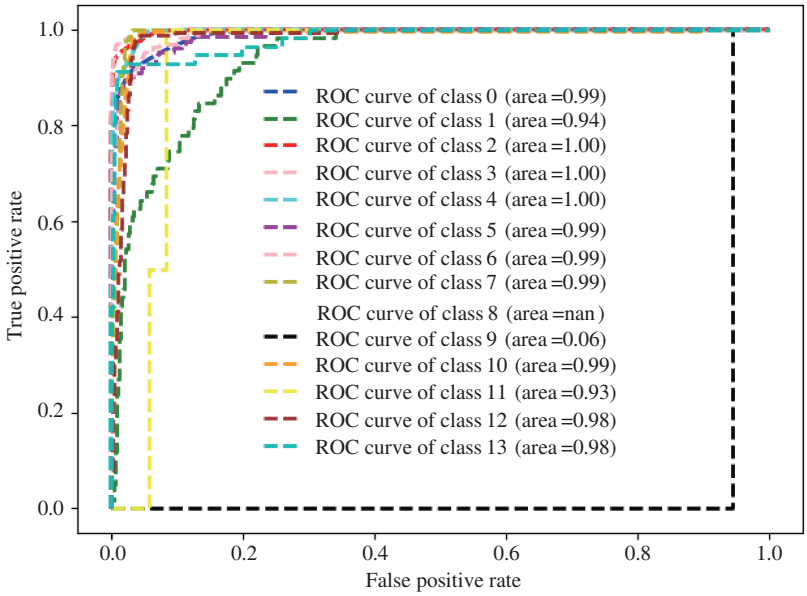

图 5　(网络版彩图) CICIDS 2017 数据集的 ROC 曲线及 AUC

Figure 5　(Color online) The ROC and AUC curves in CICIDS 2017 dataset

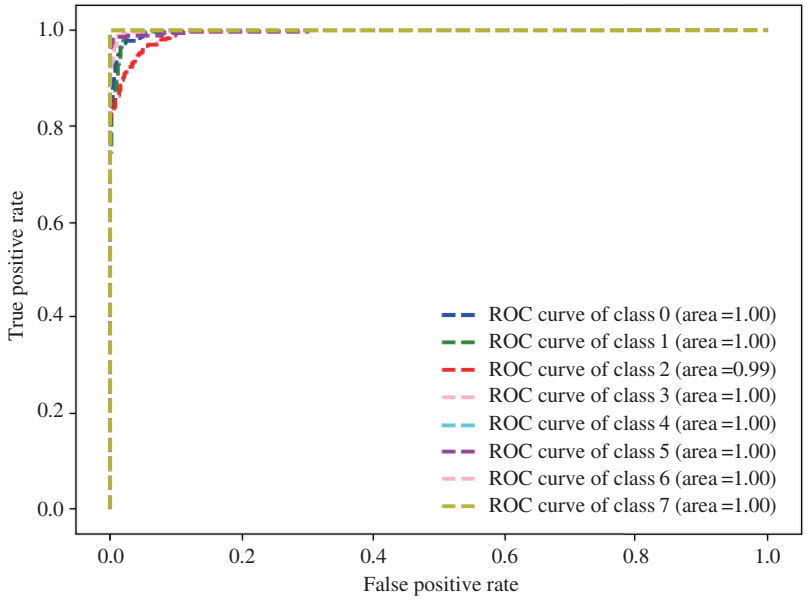

图 6　(网络版彩图) 实测数据集的 ROC 曲线及 AUC

Figure 6　(Color online) The ROC and AUC curves in real dataset

在 CICIDS 2017 和真实采集流量数据集上的 ROC 曲线和 AUC 值, 如图 5 和 6 所示.

第二, 图 4 的 Cnn5_I 模型分类准确率下降迅速, 说明在本文数据集 CICIDS 2017 上, 并不是卷积神经网络模型层数越多即模型深度越深效果越好, 随着模型加深, 运行速度愈来愈慢, 分类准确率也开始下降.

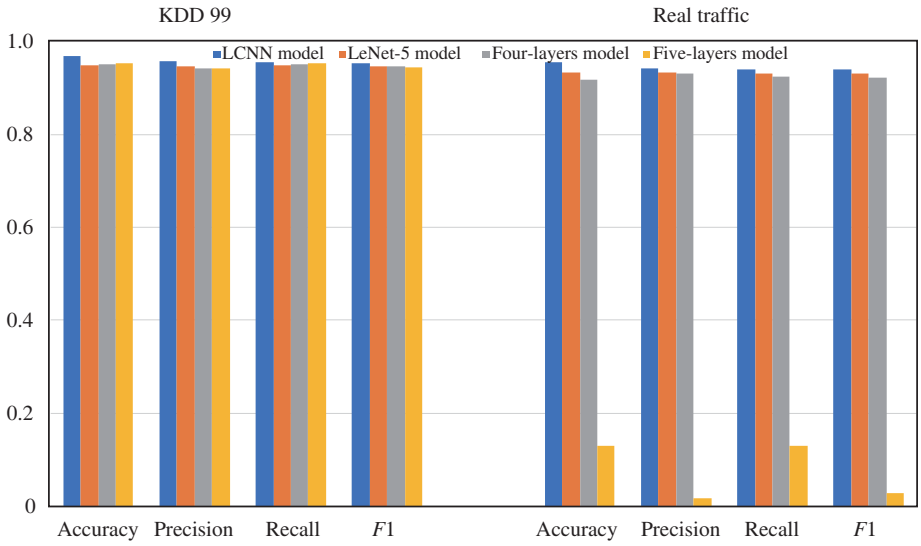

图 7 (网络版彩图) KDD 99 (左) 与真实采集流量 (右) 分类性能对比

Figure 7 (Color online) The performance comparison between KDD 99 (left) and the real dataset (right)

第三, 通过分析模型 Cnn4_H2, 得出池化层并不是一直积极有效的, 在该模型中, 去掉池化层的模型总体分类准确率反而得到一定程度的提升.

分析图 5 可知 14 类流量中绝大多数流量的 ROC 曲线都集中在左上角靠近 (0,1) 区域, 它们的 AUC 值也接近于 1, 说明本文算法的分类准确率较高, 模型性能比较优秀. 注意到第 9 类没有显示结果, 这是因为数据集 CICIDS 2017 中该类型样本数只有 5 条, 相比于 100 多万条流量样本数来说, 这类流量显得微不足道, 流量占比太小, 所以模型并没有识别出此类型流量. 同时第 10 类的 AUC 值也非常小, 接近于 0, 同样是样本数较少的缘故. 但随着当前网络流量以 EB 形式出现, 呈现大数据化趋势, 模型处理的基本都是大数据, 上述小样本数流量出现的概率会越来越小, 可以忽略不计.

由图 6 可知, 本文模型在 CERNET 真实采集流量数据集上的分类准确率也较好, 说明模型同样适用于样本数分布均匀的数据集. 但图 6 的总体分类准确率仅有 95.48%, 这是由于该数据集样本数只有 2400 条, 相对于 CICIDS 2017 数据集来说样本数过少. 说明本文算法对大样本数据集的分类性能更为优秀, 更适合当前的大数据环境.

为了进一步证明本文分类模型 LCNN 的优越性, 本文在 KDD 99 和真实流量数据集上对 LeNet-5、LCNN、4 层模型、5 层模型的分类评价指标也做了详细对比, 如图 7 所示. 由图可知虽然优势不明显, 但也能看出本文 LCNN 模型在其他数据集上同样具有较高的优越性.

此外也简单比较了本文算法 (our algorithm)、一维 CNN 分类模型和 RNN 分类模型在 3 种数据集上的分类性能. 如图 8 所示, 可知本文算法的检测精度也是最佳的.

4.3.2 梯度下降优化算法对分类准确率的影响

Adam 利用梯度的一阶矩估计和二阶矩估计动态调整每个参数的学习率, 进行自适应学习. 它将原始梯度做一个指数加权平均, 在归一化处理后进行梯度值的更新. 因此在偏置校正后, 参数比较平稳且每一次迭代学习率都有个确定范围. 它基本上就是将 Momentum 和 RMSprop 的结合, 据此猜测使用此方法进行优化会得到较好的效果. 为了证明我们的假设, 本小节对比分析了不同的优化算法, 由

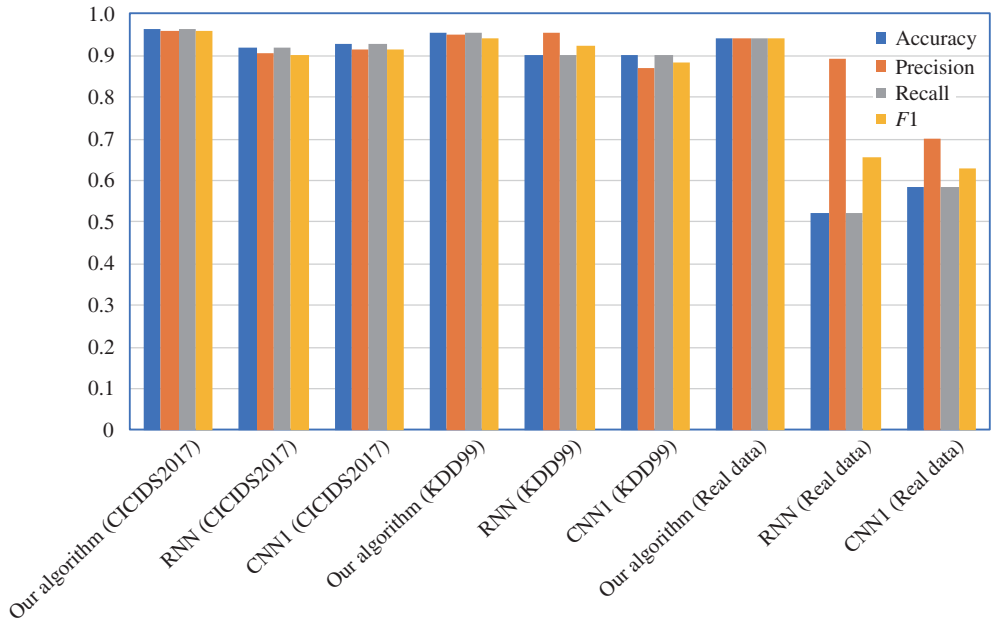

图 8　(网络版彩图) 3 种数据集在不同模型上的比较

Figure 8　(Color online) Comparison of three data sets on different models

表 6　不同梯度下降优化算法对 LCNN 模型精度的影响

Table 6　The evaluation results of different Gradient descent optimization algorithms to LCNN

Algorithm	Accuracy	Precision	Recall	$F1$
Momentum	0.9198	0.8965	0.9203	0.9049
Adagrad	0.9010	0.8752	0.9013	0.8778
Adam	0.9643	0.9589	0.9642	0.9606
RMsprop	0.9553	0.9539	0.9557	0.9525
Gradient descent	0.8839	0.8448	0.8841	0.8533

表 6 可知, 在学习率相同的情况下 (learning rate = 0.001), 使用梯度优化算法 Adam 的模型分类性能最佳.

4.3.3　不同文献网络流量分类算法对比实验

本文分别与文献 [25, 26, 38] 中的模型进行对比分析, 结果如图 9~11 所示. 实验证明本文算法的分类性能最佳, 尽管本文算法不能完全取代其他算法, 但在一定程度上可以弥补其他算法的不足, 实现更好的流量分类.

分析图 9~11 发现文献 [38] 模型分类性能仅次于本文分类模型, 原因是该模型卷积层数为 5, 相比文献 [25] 的 2 层模型结构和文献 [26] 的 1 层结构, 其深度有所增加, 验证了在深度卷积神经网络中深度加深将有助于提高整个神经网络的模型效果. 但本文模型采用 3 层也得到较好的分类性能, 说明模型层数深度加深并非唯一提高模型性能的有效方法 [39]. 此外文献 [38] 第 1 层卷积核数选取 96, 说明较多的卷积核数可以得到类别的显著特征, 有助于类别的区分. 但是该模型的运算时间较长, 不利于实时测量. 在针对复杂图片分类时性能较好, 但是本文特征属性较少, 且在网络环境下需要尽快得

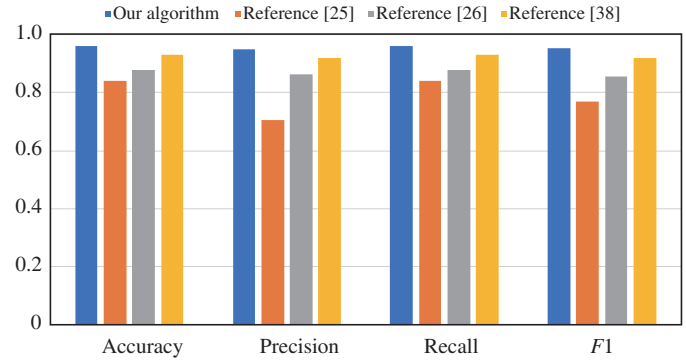

图 9 (网络版彩图) 不同分类方法在 CICIDS 2017 中的性能评估

Figure 9 (Color online) The performance evaluation of different schemes on CICIDS 2017

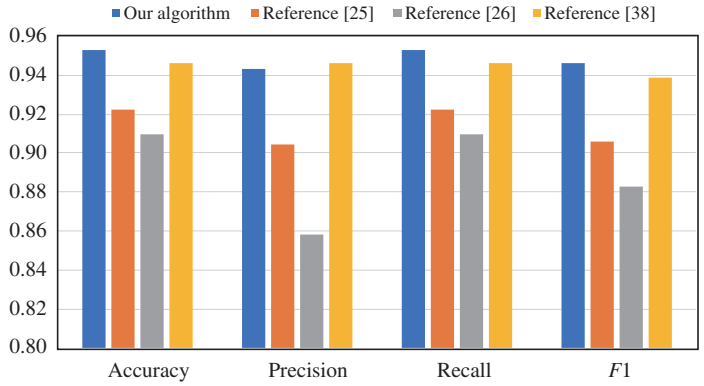

图 10 (网络版彩图) 不同分类方法在 KDD 数据集上的性能评估

Figure 10 (Color online) The performance evaluation of different schemes on KDD

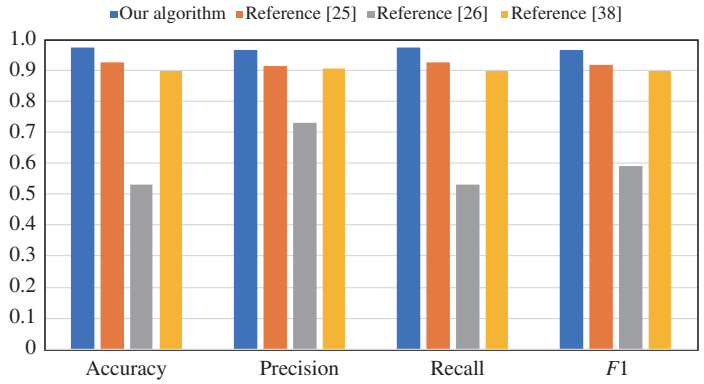

图 11 (网络版彩图) 不同分类方法在 real data 数据集上的性能评估

Figure 11 (Color online) The performance evaluation of different schemes on real data

到分类结果, 因此该模型在本文中不是很适用. 文献 [26] 使用 1D-CNN 评估结果相对较差, 因其只使用了一层卷积层, 深度较浅, 得到的特征粒度较粗.

5 总结

本文使用 PCA 算法结合改进的深度卷积神经网络分类模型 (LCNN) 对网络流量进行分类. 为了评估算法性能, 本文使用 3 种数据集进行验证, 两种为公开数据集一种为实测流量. 实验表明本文算法的检测精度可达 96.58%, 比目前相似工作提升了 5%~8%. 尽管本文算法不能完全替代其他算法, 但可与其他算法相互弥补, 模型泛化能力优秀, 可在实际环境下进行高效的检测.

另外, 本文与一维深度卷积神经网络和深度循环神经网络进行了比较, 也比较了 5 类梯度下降算法对模型分类性能的影响. 从多方位充分证明本文算法能对网络流量进行准确分类, 具有广阔的应用场景.

未来我们还希望研究以下内容: (1) 三维深度卷积神经网络对网络流量的分类性能; (2) 研究批标准化优化算法对本文模型分类准确率的影响, 以期继续提高本模型的分类准确率; (3) 研究 KPCA 非线性特征优化算法对模型分类准确率的影响.

参考文献

1 Hao F, Kodialam M, Lakshman T V, et al. Fast dynamic multiple-set membership testing using combinatorial bloom filters. IEEE/ACM Trans Netw, 2012, 20: 295–304

2 Sen S, Spatscheck O, Wang D M. Accurate, scalable in-network identification of P2P traffic using application signatures. In: Proceedings of the 13th International Conference World Wide Web Conference, Florham Park, 2004. 512–521

3 Moore A W, Papagiannaki K. Toward the accurate identification of network applications. In: Proceedings of the 6th International Workshop on Passive and Active Network Measurement, Cambridge, 2005. 41–54

4 Ertam F, Avci E. A new approach for internet traffic classification: GA-WK-ELM. Measurement, 2017, 95: 135–142

5 Naoum R S, Abid N A, Al-Sultani Z N. An enhanced resilient backpropagation artificial neural network for intrusion detection system. Int J Comput Sci Netw Secur, 2012, 12: 11–16

6 Naoum R S, Al-Sultani Z N. Learning vector quantization (LVQ) and knearest neighbor for intrusion classification. World Comput Sci Inf Technol J, 2012, 2: 105–109

7 Aburomman A A, Reaz M B I. A novel weighted support vector machines multiclass classifier based on differential evolution for intrusion detection systems. Inf Sci, 2017, 414: 225–246

8 Eid H F, Darwish A, Ella H A, et al. Principle components analysis and support vector machine based intrusion detection system. In: Proceedings of the 10th International Conference on Intelligent Systems Design and Applications, Cairo, 2010. 363–367

9 Kuang F, Xu W H, Zhang S Y. A novel hybrid KPCA and SVM with GA model for intrusion detection. Appl Soft Comput, 2014, 18: 178–184

10 Rastegari S, Hingston P, Lam C P. Evolving statistical rulesets for network intrusion detection. Appl Soft Comput, 2015, 33: 348–359

11 He J, Zhao L. Research on P2P traffic classification based on PCA-probabilistic neural network. Comput Dev Appl, 2011, 7: 1–3

12 Valenti S, Rossi D, Dainotti A, et al. Reviewing Traffic Classification. Berlin: Springer, 2013

13 Pan W B, Cheng G, Guo X J, et al. Review and perspective on encrypted traffic identification research. J Commun, 2016, 37: 1–14 [潘吴斌, 程光, 郭晓军, 等. 网络加密流量识别研究综述及展望. 通信学报, 2016, 37: 1–14]

14 Yang Y, Kang C C, Gou G P, et al. TLS/SSL encrypted traffic classification with autoencoder and convolutional neural network. In: Proceedings of the 20th HPCC/16th SMARTCITY/4th DSS, Beijing, 2018. 362–369

15 Xiong G, Zhao Y, Cao Z G. Real-time classification for encrypted P2P traffic based on host behavior association. Chinese High Technol Lett, 2013, 23: 1008–1015 [熊刚, 赵咏, 曹自刚. 基于主机行为关联的加密 P2P 流量实时分类方法. 高技术通讯, 2013, 23: 1008–1015]

16 Karagiannis T, Broido A, Brownlee N, et al. Is P2P dying or just hiding? In: Proceedings of IEEE Global Telecommunications Conference, San Diego, 2004. 1532–1538

17 Aceto G, Dainotti A, de Donato W, et al. PortLoad: taking the best of two worlds in traffic classification. In: Proceedings of INFOCOM IEEE Conference on Computer Communications Workshops, Naples, 2010

18 Khakpour A R, Liu A X. High-speed flow nature identification. In: Proceedings of IEEE International Conference on Distributed Computing Systems, Montreal, 2009. 510–517

19 Shi H T, Li H P, Zhang D, et al. Efficient and robust feature extraction and selection for traffic classification. Comput Netw, 2017, 119: 1–16

20 de la Hoz E, de la Hoz E, Ortiz A, et al. PCA filtering and probabilistic SOM for network intrusion detection. In: Proceedings of the 12th International Work-Conference on Artificial Neural Networks (IWANN), Puerto de la Cruz, 2015. 71–81

21 Agrawal S, Sohi B S. Off-line analysis of internet traffic for accurate identification of P2P applications using neural networks. In: Proceedings of the 1st International Conference on Recent Advances in Information Technology (RAIT), Chandigarh, 2012. 431–435

22 Dong S, Li R X. Traffic identification method based on multiple probabilistic neural network model. Neural Comput Appl, 2019, 31: 473–487

23 Ertam F, Galip A. Data classification with deep learning using tensorflow. In: Proceedings of the 2nd International Conference on Computer Science and Engineering, Elazig, 2017. 755–758

24 Peng L Z, Yang B, Chen Y H. Effective packet number for early stage internet traffic identification. Neurocomputing, 2015, 156: 252–267

25 Wang Y, Zhou H Y, Feng H, et al. Network traffic classification method based on improved capsule neural network. J Commun, 2018, 1: 14–23 [王勇, 周慧怡, 俸皓, 等. 基于深度卷积神经网络的网络流量分类方法. 通信学报, 2018, 1: 14–23]

26 Jain A V. Network traffic identification with convolutional neural networks. In: Proceedings of the 16th DASC/16th PICom/4th DataCom/3rd CyberSciTec, Rochester, 2018. 1001–1007

27 Lopez-Martin M, Carro B, Sanchez-Esguevillas A, et al. Network traffic classifier with convolutional and recurrent neural networks for Internet of things. IEEE Access, 2017, 5: 18042–18050

28 Hoque N, Bhattacharyya D K, Kalita J K. An alert analysis approach to DDoS attack detection. In: Proceedings of International Conference on Accessibility to Digital World, Assam, 2016. 33–38

29 Sze V, Chen Y H, Yang T J, et al. Efficient processing of deep neural networks: a tutorial and survey. Proc IEEE, 2017, 105: 2295–2329

30 Sun G L, Liang L L, Chen T, et al. Network traffic classification based on transfer learning. Comput Electr Eng, 2018, 69: 920–927

31 Deng X G, Tian X M, Chen S, et al. Deep learning based nonlinear principal component analysis for industrial process fault detection. In: Proceedings of International Joint Conference on Neural Networks, Qingdao, 2017. 1237–1243

32 Dias K L, Pongelupe M A, Caminhas W M, et al. An innovative approach for real-time network traffic classification. Comput Netw, 2019, 158: 143–157

33 Radford B J, Richardson B D, Davis S E. Sequence aggregation rules for anomaly detection in computer network traffic. Comput Sci, 2018, 8: 1–5

34 Aurélien G. Hands-on Machine Learning with Scikit-learn & Tensorflow. Beijing: China Machine Press, 2018

35 Aceto G, Ciuonzo D, Montieri A, et al. Multi-classification approaches for classifying mobile App traffic. J Network Comput Appl, 2018, 103: 131–145

36 Kornycky J, Abdul-Hameed O, Kondoz A, et al. Radio frequency traffic classification over WLAN. IEEE/ACM Trans Netw, 2017, 25: 56–68

37 Zhou F Y, Jin L P, Dong J. Review of convolution neural network. Chinese J Comput, 2017, 6: 1229–1251 [周飞燕, 金林鹏, 董军. 卷积神经网络研究综述. 计算机学报, 2017, 6: 1229–1251]

38 Krizhevsky A, Sutskever I, Hinton G E. ImageNet classification with deep convolutional neural networks. Commun ACM, 2017, 60: 84–90

39 Gao Y C, Liu N H, Zhang S. Relative indexed compressed sparse filter encoding format for hardware-oriented acceleration of deep convolutional neural networks. In: Proceedings of the 7th IEEE International Symposium on next-generation Electronics (ISNE), Taipei, 2018. 323–326

Network traffic classification method based on improved deep convolutional neural network

Xiaoli ZHANG[1,2*], Guang CHENG[2] & Weici ZHANG[2]

1. *Department of Intelligent Control, Shanxi Railway Vocational and Technical College, Taiyuan* 030013*, China*;
2. *School of Cyber Science and Engineering, Southeast University, Nanjing* 211189*, China*
* Corresponding author. E-mail: xiaoli2408@163.com

Abstract The prerequisite of traffic classification based on machine learning models is that traffic is independent and has identical distributions. However, traffic changes in the wild increase the memory cost of these models and reduces their accuracy. To tackle these problems, this work proposes a new classification model. The model combines a principal component analysis algorithm and an improved deep convolutional neural network. The former performs dimensionality reduction so that the key features affecting detection accuracy are found. The latter adopts the autonomous feature learning method to improve the classification accuracy. Experiments show that the memory overhead is reduced by 3.2% and that the detection accuracy is improved by 8% relative to other similar works.

Keywords network traffic classification, deep convolutional neural network, PCA, multi-classifier, feature selection, Tensorflow

Xiaoli ZHANG was born in 1982. She received her M.S. degree from the School of Computer Science and Engineering, Taiyuan University of Technology, Taiyuan, in 2012. Currently, she is an associate professor in Shanxi Railway Vocational and Technical College. Her interests include computer network technology and network security.

Guang CHENG was born in 1973. He received his Ph.D. degree in Computer Science from Southeast University, Nanjing, in 2003. From 2006 to 2007, he held a postdoctoral position with the School of Electrical and Computer Engineering, Georgia Institute of Technology. Currently, he is a full professor and doctoral supervisor in the School of Cyber Science and Engineering, Southeast University. His interests include network architecture, active measurement, and traffic sampling in computer networks.

Weici ZHANG was born in 1996. He received his B.S. degree in Computer Science from Hohai University, Nanjing, China, in 2018. He is pursuing his M.S. degree in the School of Computer Science and Engineering at Southeast University, Nanjing, China. His main research interests include cyber security and cyber threat awareness.