基于网络流时空序列的加密流量分类

唐博麟[1] 王晨飞[1] 江 帆[1] 张 虎[1] 徐李阳[1] 赵文华[1] 王 蕾[1] 李晓红[2]

[1](国家电网有限公司客户服务中心 天津 300309)

[2](天津大学智能与计算学部 天津 300072)

摘 要 流量分类问题对于网络资源管理和安全非常重要。然而用户流量经常被加密处理,为流量分类问题带来极大的挑战。为此,提出一种新型的时间序列特征提取方法,用于解决加密应用程序流量分类问题。该方法通过分析数据包的空序列,提取加密网络流量的关键行为特征,并结合自注意力机制的长短时记忆网络来训练并对流量进行分类。为了评估方法的有效性,在公开网络数据集 ISCXVPN2016 上进行了详细的实验。结果表明,此方法能够显著提高识别加密应用程序流量的准确性和计算效率。

关键词 深度学习 加密流量识别 神经网络

中图分类号 TP393.08 **文献标志码** A DOI: 10.3969/j.issn.1000-386x.2024.03.045

ENCRYPTED TRAFFIC CLASSIFICATION BASED ON NETWORK FLOW TIME-SPACE SERIES

Tang Bolin[1] Wang Chenfei[1] Jiang Fan[1] Zhang Hu[1] Xu Liyang[1] Zhao Wenhua[1] Wang Lei[1] Li Xiaohong[2]

[1](*Customer Service Center, State Grid Corporation of China, Tianjin 300309, China*)

[2](*College of Intelligence and Computing, Tianjin University, Tianjin 300072, China*)

Abstract Traffic classification is very important for network resource management and security. However, user traffic is often encrypted, which brings great challenges to traffic classification. Therefore, we propose a novel time series feature extraction technique to address the encrypted traffic classification problem. We extracted significant attributes of the encrypted network traffic behavior by analyzing the time series of received packets. We used the LSTM combined with attention mechanism to train and classify traffic. To evaluate the efficiency of the proposed method, we carried out intensive experiments on an open network dataset ISCXVPN2016. The experimental results show that the proposed method can significantly improve the performance in identifying encrypted application traffic in terms of accuracy and computation efficiency.

Keywords Deep learning Encrypted traffic classification Neutral network

0 引 言

近年来,随着加密技术的进步,加密流量在互联网上得到了广泛应用。许多服务和应用程序使用加密算法来保护信息安全。据 Gartner 估计,到 2019 年,超过 80% 的企业网络流量已被加密。以谷歌浏览器为例,截至 2019 年 5 月,其流量中 94% 已被加密。加密技术不仅可以保护网络用户的自由、隐私和匿名性,而且可以绕过防火墙检测和监视系统[3]。然而,一些不法分子也利用加密来获取非法利益。例如,2020 年超过 70% 的恶意软件活动使用加密来隐藏恶意软件传递、命令和数据泄漏。因此,加密流量的识别和分类受到了学术界和工业界的广泛关注[1-3]。加密技术的发展使得数据包在通过加密算法(如对称加密或非对称加密算法)后由明文变为密文,大量信息变得不可见,这

收稿日期:2023-08-12。基于深度学习的网上国网流量及微服务安全防护技术研究(SGTWAZQT2200040);国家自然科学基金面上项目(61872262)。唐博麟,工程师,主研领域:网络安全。王晨飞,硕士。江帆,工程师。张虎,硕士。徐李阳,硕士。赵文华,硕士。王蕾,工程师。李晓红,教授。

为加密流量的分类带来了巨大的挑战。实际场景中，通常需要识别特定的协议或应用程序类型。对于加密的应用程序流量来说，由于应用程序类型较多，而且不同类型之间几乎没有差异，因此流量分类也变得困难。

网络流量的加密处理为现有的网络流量分类带来了许多挑战。第一，加密算法会影响分类准确性。对于有效载荷[4]的方法，加密算法会影响数据包结构，从而降低分类准确性。而传统基于网络流的方法由于通常利用网络流的统计特征[7,12-13]进行分类，因此该方法也会增加计算和存储开销。第二，传统的基于有效载荷和基于流的统计特征技术在加密网络流量分类中的识别准确率和计算资源开销无法达到实际使用需求。虽然已经引入了许多基于机器学习的分类方法来克服传统方法的局限性[5]，但它们的有效性在很大程度上取决于特征提取过程的准确性和有效性。综上所述，由于大多数应用程序产生的网络流量最近都使用加密技术进行了加密[6-8]，导致传统流量识别方法表现不佳，此外针对基于机器学习方法的流量识别，其识别特征严重依赖于手动设计和提取。

为了解决上述问题，本文提出了一种新的基于机器学习的分类方法，能够有效抽取加密流量的特征。首先，利用包的时间依赖性来表现数据流的行为特征。之后，利用 LSTM(Long Short-Term Memory)[22]网络的优势来从已加密的数据流中抽取时空依赖。为了验证本方法的高效性，我们将本方案应用于了公开数据集 ISCXVPN2016。实验证明，本文方法在具备高效性的同时，还具有良好的准确性。

本文主要的贡献如下:

1) 提出了一种新的基于深度学习的分类方法，并利用数据流量的时空序列特征来表示加密流量。

2) 利用 LSTM 网络来对网络应用流量数据的时间序列分析来保留接收流量的时间依赖性。

3) 本文对上述方法进行了实现，并与现有方法进行了对比，验证了本文方法的有效性。

1 相关工作

1.1 基于流的方法

流是一组具有相同互联网协议(Internet Protocal，IP) 地址、目标 IP 地址、源传输层端口、目标传输层端口和传输协议的数据包。先前的工作[9-11]展示了基于流的方法的有效性。入侵检测系统将恶意流量与正常流量进行分类。因此，该方法可以有效地分析数据包集的行为。Gil 等[12]介绍了使用基于时间的数据包属性进行网络流量分析的方法。然而，这些属性需要大量存储空间保存在一段时间内的数据包[13]。因此，在提取特征之前将许多数据包收集到一个流中是非常耗时的。

1.2 基于有效载荷的方法

基于有效载荷的方法一次处理一个数据包，因此具有较高的处理速度。然而，这种方法无法准确有效地描述流量行为。最近，在不对数据包进行解密的情况下分析加密的数据包负载受到了更多的关注[14-15]。然而，它仅对数据包大小和传输时间极为不同的一些特定应用程序有效，例如 HTTP、VoIP、视频流和 P2P[16-17]。此外，深度学习技术也被引入到原始数据包的特征提取中[4,18]。然而，这些方法提取整个原始数据包，将其放入深度学习网络。这种方法可以用于处理未加密的流量，但对于加密的流量来说效率很低。

1.3 基于有效载荷和流的组合方法

使用深度学习方法(如堆栈自动编码器(SAE) 、CNN、LSTM 网络[4,19]) 对网络流量进行分类的工作有限。然而，这些工作只关注深度网络模型的结构，以从大型流的原始数据包中提取大量有效负载字节。因此，深层网络需要大量具有许多神经元的隐藏层来提高其准确性。Zhang 等[20]的工作表明，网络流量的时间序列分析对于网络流量分类问题是有效的。他们提取数据包和流的手工统计特征来表示流量应用，导致准确性严重依赖于人类知识和高计算资源[23]。在本文的工作中，只提取强烈表示加密数据包的数据包特征(根据应用程序) ，然后按时间序列顺序排列接收数据包样本。此外，本文利用 LSTM 网络自动提取时间序列数据包的时间依赖性，不仅可以有效地保留流量的特征，还可以提高分类加密流量和应用程序的准确性并减少延迟。

2 本文方法

2.1 整体框架

本文提出的网络流量分类模型网络结构如图 1 所示，首先，根据每个数据包内的时间戳和五元组信息(源 IP、目的 IP、源端口、目的端口和协议) 将数据包划分到不同网络流中。然后通过抽取一定数量的数据包来提取数据包特征。之后进一步提取目标特征的词向量。接下来，将词向量作为输入，利用长短时记忆网络结合自注意力机制来进行流量分类。

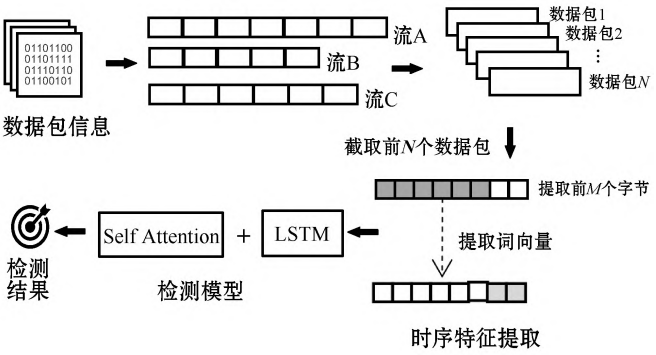

图 1　整体模型框架

2.2　网络流量数据包时空序列特征

网络流量数据包是传输控制协议/互联网协议(TCP/IP)模型中网络层的数据单元。此前大多数工作都将数据包的整个有效负载作为流量特征,例如 1 500 字节[4]和 1 000 字节[19]的流量负载。此外,从图 2 可以看出数据包由 20 个字节的 IP 报头、20 个字节的传输层报头以及应用报头和应用数据的所有剩余字节组成。在加密应用程序中,使用对称加密算法对应用层数据加密,以确保加密和解密的速度,如 AES256 和 RC4[21]。换句话说,加密密钥是在不同会话中更改的伪随机数。因此,尽管不同会话来自于同一类型的应用程序,但会话之间加密的应用程序层数据的值会有不同。因此,应用层数据无法准确表示应用程序类型。然而,应用层的头部描述了对识别网络流量非常重要的内容,如加密算法名称和协议等内容。

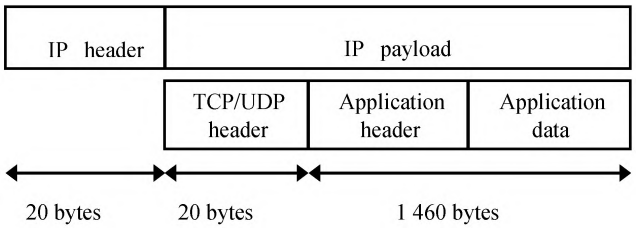

图 2　数据包结构

因此,本文选取前 N 个数据包的前 M 个字节作为流量特征,其中在训练词向量阶段,传统的方式是使用 one-hot 的形式来表示词向量,即用长度为词库词数量大小的向量来表示每一个词,并且向量中只有一个分量为 1,其余皆为 0。其中 1 的位置代表该词在词库中的位置。但是当词库数量很大时,向量维度太长,在深度学习算法中不适用,而且对于近义词来说,对应的词向量关联性很差,所以本文采用 Elmo 编码方式训练词向量,其原理是将所需要训练的语言中的词每个词映射成固定长度的向量,并且每个分量的数字不再固定只能为 0、1。本文之所以采用的 Elmo 对数据包的词向量进行训练,是考虑到相同的字节在不同应用的加密流量的协议中代表的含义差别很大。如果采用传统的词向量训练方式如 word2vec 获取固定的词向量,这在加密流量识别模型中是不符合数据特性的。而 Elmo 双层双向的 LSTM 网络预训练结构,可以得到满足协议复杂性的词向量,并且相同字节在不同的协议环境下对应出不同的词向量。在本文中我们设置 N 为 6,M 为 128。

2.3　基于自注意力机制的时序网络流量分类方法

长-短期记忆(LSTM)网络是一种特定类型的递归神经网络(RNN)[22],通过引入“门”的结构来保持长期依赖关系的记忆。LSTM 与 RNN 的区别在于它们的基本组成单元,其优点之一是它能够通过门结构向单元状态添加或移除信息。每个单元包含三个门,即输入门、遗忘门和输出门。其中,遗忘门决定了哪些信息应该被遗忘,以便在处理长序列时保持记忆的稳定性,遗忘门决定了哪些新的信息应该被加入到单元状态中,而输出门决定了单元状态的哪部分将作为输出。

一个隐藏层具有多个 LSTM 单元,而一个深度学习模型通常是多个隐藏层的组合。通过计算隐藏层的输出、权重和输入值的偏导数,系统可以向后移动以跟踪实际输出值和预测输出值之间的误差,并使用梯度下降算法并行更新权重以减少预测误差。

LSTM 虽然能够捕获长期的序列依赖关系,并相对于普通的 RNN 有更好的训练稳定性,但是在实际使用中随着序列长度的增加,它可能会逐渐丧失早期的信息。为了应对该问题,我们提出基于注意力机制的 LSTM 流量分类方法。该方法通过增加自注意力层,其可以使得模型直接关注到流量特征序列的任何部分。此外,传统的统 LSTM 中解码器在每个时间步计算中都依赖于前一个时间步的输出,因此可能导致错误的累积,而基于自注意力的 LSTM 流量分类算法可以在每个时间步计算中重新考虑整个输入序列,从而避免上述问题。

图 3 展示了特征提取和注意力机制框架,该网络模型包括四个部分,Elmo 特征编码层、长短时网络层、自注意力层和全连接分类层。假设在 Elmo 层输入 n 个数据包,则每个数据包截取前 108 个字节作为 Elmo 阶段的输入,编码后的数据作为自注意力 LSTM 模型的输入。之后,第一层的 LSTM 使用 ReLU 作为激活函数。数据包特征经过学习后,其输出尺寸为 256,这将作为下一层 Batch Normalization 的输入。下一层为自注意力层,通过 Self-Attention 层,模型可以学习加密会话或流中的数据包之间的内部结构和协议特征的相对

依赖关系,这有助于更好地分类网络流量。经过 Self-Attention 层后,输出尺寸仍为 256。接下来,这些特征将再次通过一层 Batch Normalization 并进入全连接层。第一个全连接层包含 64 个神经元并使用 ReLU 作为激活函数,最后使用 Softmax 层进行分类,其中 K 的数值根据具体分类任务进行调整。

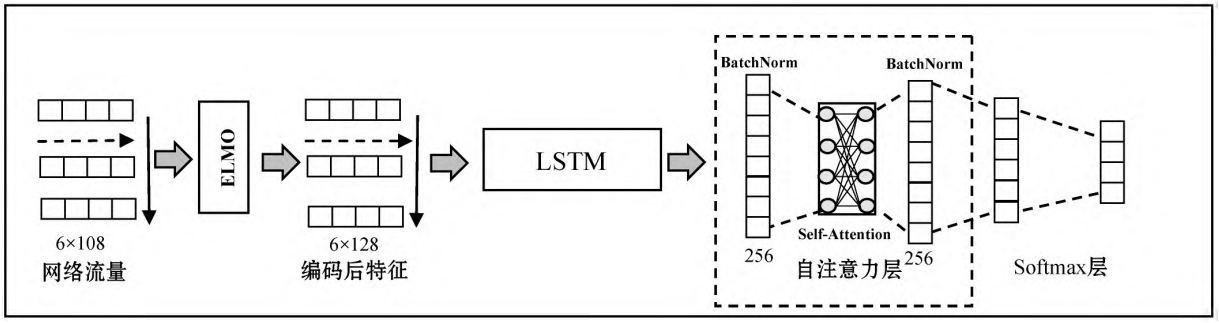

图 3 特征提取和自注意力机制模型

3 实验设置和评估

3.1 实验设置

数据集:为了测试所提出方法的有效性,本文使用著名的加密网络流量数据集,ISCXVPN2016[12] 数据集是一个用于网络流量分析和安全研究的公开数据集,旨在模拟虚拟专用网络(VPN)环境中的网络流量,该数据集包含浏览器流量、社交网络流量、视频网站流量等绝大部分网络中的流量类型,因此是作为网络流量分类评估的优秀数据集。数据集中共 5 个不同的标签,每个标签中都包含三种流量类别,分别是 non_VPN、VPN 和 Tor。其中从标签角度考虑,本文对 voip 和 video 应用进行十分类。从流量类别考虑,对 non_VPN 流量类别进行十分类,对 VPN 流量类别进行四分类,对 Tor 类别的流量进行五分类以及从整体上对三种流量类别的三分类。

此外,我们还选择了 CTU-13[24] 作为数据集进行了对比实验。CTU-13 包含了各类网络场景中的恶意流量,包括僵尸网络、拒绝服务攻击等异常流量。由于流量样本是从实际的网络环境中捕获而来,这使得它能够有效模拟真实世界网络情况。

对比方法:我们选择四种机器学习方法 SVM(支持向量机)、RF(随机森林)、KNN(k-邻近)以及 XGBoost(梯度提升树)作为基准方法进行对比。

实验环境:本文实验在 Ubuntu 20.04 服务器上进行,CPU 为 AMD EPYC 72618@2.6 GHz,内存为 32 GB,GPU 为 RTX3090 24 GB。实现中使用的程序语言是 Python 3.8.7。深度神经网络是用 PyTorch 1.6.0 以及 Numpy 1.16.4 实现的。

3.2 评估指标

实验评估使用四个评估指标:准确度(Accuracy)、精确度(Precision)、召回率(Recall)和 F1 分数(F1-Score)。如式(1)-式(4)所示,T_P 是正确分类为 X 的样本数,T_N 是正确分类为非 X 的样本数,F_P 是错误分类为 X 的样本数,F_N 是错误分类为非 X 的样本数。准确度用于评估分类器的整体性能,这在式(1)中计算。精确度(式(2))、召回率(式(3))和 F1 分数(式(4))通常用于评估识别某类流量的性能。精确度和召回率的优点是非常直观且易于实现,如果仅仅看精确度或者召回率中的一个,有可能会在不知情的情况下走向极端,故不足以度量分类器,尤其对于不平衡的数据集。F1 分数克服了精确度和召回评分的缺点。

$$A_{ccuracy}=\frac{T_P+T_N}{T_P+T_N+F_P+F_N} \tag{1}$$

$$P_{recision}=\frac{T_P}{T_P+F_P} \tag{2}$$

$$R_{ecall}=\frac{T_P}{T_P+F_N} \tag{3}$$

$$F_1=\frac{2\times P_{recision}\times R_{ecall}}{P_{recision}+R_{ecall}} \tag{4}$$

F1 分数是精确度和召回率的调和平均值,其中调和平均值是平均比率的合适方法。因此,它不受各个类别规模差异的影响,F1 分数的计算使用式(4)。F1 分数通常被认为是评估分类算法在多种数据集中性能的可靠指标。因此,后续实验将使用此度量来衡量本文提出的模型的性能,并与其他工作进行比较。另外,精确度和召回率也作为评估标准,以便参考。

3.3 研究问题

RQ1:本方法分类效果如何?与其他方法相比,以上述评估指标作为依据,能否达到更好效果?

RQ2:本文提出的特征工程方法在多分类任务上的效果如何?

3.4　实验评估

3.4.1　本文方法与其他方法进行对比结果(RQ1)

我们在 CTU-13 数据集上对比了不同机器学习方法的分类效果,结果如表 1 所示。我们能够发现,与基准中最好的方法相比,本方法在准确率方面提高了约 1%,在召回率方面提升了约 4%,在精确率和 F1 分数方面均提升了约 2%。

表 1　在不同机器学习方法中的分类对比

方法	Accuracy	Precision	recall	F1-score
SVM	0.751	0.725	0.829	0.718
RF	0.943	0.929	0.931	0.916
KNN	0.834	0.751	0.815	0.889
XGBoost	0.916	0.884	0.912	0.863
our method	0.957	0.951	0.972	0.934

综上所述,与其他方法相比,本文方法在四项指标上均能达到最好的分类效果。

3.4.2　本文方法在多分类任务中的结果(RQ2)

对流量进行十分类的实验结果如表 2 所示,可以发现在不同应用上模型性能有一定的差距,可能与流量的包结构有一定的关系。在每个类别上精确率均高于 94%,召回率都超过了 89.9%,表示本文方法对于正例预测准确性和实际正例样本的识别能力较好。除此之外,在 F1 分数上也表现优异,均高于 92%,说明本文方法在分类任务中能够取得较好的综合性能。

表 2　十分类结果的性能分析

流量类型	Precision	Recall	F1-score
facebook_audio	0.958	0.921	0.939
facebook_video	0.942	0.987	0.964
hangouts_audio	0.963	0.951	0.957
hangouts_video	0.948	0.899	0.923
netflix	0.947	0.927	0.937
skype_audio	0.966	0.945	0.955
skype_video	0.951	0.974	0.962
vimeo	0.957	0.973	0.965
voipbuster	0.974	0.924	0.948
youtube	0.975	0.936	0.955
Accuracy	0.986		

我们进一步对 VPN 类别的流量进行四分类,实验结果如表 3 所示。该实验的目的不是识别应用,而是识别应用所属的类别。从实验结果中可以得出,该模型对 chat 和 video 类别的应用分类准确性较高,在其他两个类别上效果略有下降,总体准确率高达 94.2%。

表 3　VPN 四分类结果的性能分析

流量类型	Precision	Recall	F1-score
chat	0.956	0.892	0.923
filetrans	0.923	0.874	0.898
video	0.967	0.936	0.951
voip	0.898	0.902	0.900
Accuracy	0.942		

接下来我们对 Tor 类别的流量进行了五分类,这个实验的目的是识别应用所属的类别。如表 4 所示。该模型在 Tor 类别五分类任务上表现良好,F1-score 均超过了 91.5%。browsing 类别上表现最好,在 filetrans 类别上表现略差,易将 browsing 类别的应用识别为 browsing 类别,但是却不易将 browsing 类别识别为 file-trans 类别。

表 4　在 Tor 类别的五分类性能分析

流量类型	Precision	Recall	F1-score
browsing	0.983	0.925	0.953
chat	0.923	0.927	0.925
filetrans	0.897	0.934	0.915
voip	0.937	0.927	0.932
Accuracy	0.952		

最后,为了对流量种类进行分类,将数据集划分为三个类别,分别是 non_VPN、VPN 和 Tor,实验结果如表 5 所示,该模型在 tor 和 VPN 两个类别上表现良好,在 non_VPN 类别的流量识别方面最低,但也达到了 89.3%。

表 5　non_VPN、VPN 和 Tor 三分类结果

流量类型	Precision	Recall	F1-score
non_VPN	0.893	0.902	0.897
tor	0.938	0.924	0.931
VPN	0.931	0.932	0.931
Accuracy	0.937		

从以上实验可以得出,本文方法在十分类、五分类,以及三分类方面均表现良好,能够完成现实要求中的多分类任务。

4　结　语

本文提出了一种新的用于网络流量分类的时间序

列表示方法,以及基于注意力机制的 LSTM 网络模型的流量分类方法。首先,利用网络数据包的重要特征,将数据包流表示为时间序列数据,将原始数据包中的数据样本重新配置为能够表示网络流量行为的时间序列样本。此外,本文还利用 LSTM 网络的优势,并结合自注意力机制设计了深层网络模型,能够有效地学习时空序列特征。实验结果表明,采用加密网络流量的时间序列分析来表示加密流量,并结合自注意力机制和 LSTM 网络,可以帮助分类器实现更好的加密网络流量分类性能。本文的工作为利用加密网络流量的时间序列特征和深度学习方法来表示网络流量建立了基础,这是分析网络流量的重要步骤,也为异常检测和流量分类等各种网络流量分析研究提供了方向。

参 考 文 献

[1] Soleymanpour S, Sadr H, Beheshti H. An efficient deep learning method for encrypted traffic classification on the web [C]//6th International Conference on Web Research, 2020: 209 – 216.

[2] Anderson B, McGrew D. Identifying encrypted malware traffic with contextual flow data [C]//ACM Workshop on Artificial Intelligence and Security, 2016: 35 – 46.

[3] Al-Obaidy F, Momtahen S, Hossain M, et al. Encrypted traffic classification based ML for identifying different social media applications [C]//IEEE Canadian Conference of Electrical and Computer Engineering, 2019: 1 – 5.

[4] Lotfollahi M, Siavoshani M, Zade R, et al. Deep packet: A novel approach for encrypted traffic classification using deep learning [J]. Soft Computing, 2020, 24(3): 1999 – 2012.

[5] Moustafa N, Slay J. UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set) [C]//Military Communications and Information Systems Conference, 2015: 1 – 6.

[6] Wang P, Li S H, Ye F, et al. PacketCGAN: Exploratory study of class imbalance for encrypted traffic classification using CGAN [C]//IEEE International Conference on Communications, 2020: 1 – 7.

[7] Xi N, Ma J F, Sun C, et al. Information flow control on encrypted data for service composition among multiple clouds [J]. Distributed and Parallel Databases, 2018, 36(3): 511 – 527.

[8] Naylor D, Finamore A, Leontiadis I, et al. The cost of the "s" in HTTPS [C]//10th ACM International on Conference on emerging Networking Experiments and Technologles, 2014: 133 – 140.

[9] Gharib A, Sharafaldin I, Lashkari A, et al. An evaluation framework for intrusion detection dataset [C]//International Conference on Information Science and Security, 2016: 1 – 6.

[10] Alshammari R, Zincir-Heywood A. Can encrypted traffic be identified without port numbers, IP addresses and payload inspection? [C]//Computer Networks, 2011: 1326 – 1350.

[11] Sharafaldin I, Lashkari A, Ghorbani A. Toward generating a new intrusion detection dataset and intrusion traffic characterization [C]//International Conference on Information Systems Security & Privacy, 2018: 108 – 116.

[12] Draper-Gil G, Lashkari AH, Mamun MS, et al. Characterization of encrypted and VPN traffic using time-related features [C]//International Conference on Information Systems Security & Privacy, 2016: 407 – 414.

[13] Ring M, Wunderlich S, Scheuring D, et al. A survey of network-based intrusion detection data sets [EB]. arXiv: 1903.02460, 2019.

[14] Sherry J, Lan C, Popa R A, et al. BlindBox: Deep packet inspection over encrypted traffic [J]. ACM SIGCOMM Computer Communication Review, 2015, 45(4): 213 – 226.

[15] Leroux S, Bohez S, Maenhaut P J, et al. Fingerprinting encrypted network traffic types using machine learning [C]//IEEE/IFIP Network Operations and Management Symposium, 2018: 1 – 5.

[16] Bhatia M, Sharma V, Singh P, et al. Multi-level P2P traffic classification using heuristic and statistical-based techniques: A hybrid approach [J]. Symmetry, 2020, 12(12): 2117.

[17] Richter C, Finsterbusch M, Rocha E, et al. A survey of payload-based traffic classification approaches [J]. Communications Surveys & Tutorials, 2014, 16(2): 1135 – 1156.

[18] Lipton Z C, Berkowitz J, Elkan C. A critical review of recurrent neural networks for sequence learning [EB]. arXiv: 1506.00019, 2015.

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

[20] Zhang F, He W B, Xue L, et al. Inferring users' online activities through traffic analysis [C]//4th ACM Conference on Wireless Network Security, 2011: 59 – 70.

[21] Singhal N, Raina J. Comparative analysis of AES and RC4 algorithms for better utilization [J]. International Journal of Computer Trends & Technology, 2011, 7(3): 177 – 181.

[22] Bengio Y, Simard P, Frasconi P. Learning long-term dependencies with gradient descent is difficult [J]. IEEE Transactions on Neural Networks, 1994, 5(2): 157 – 166.

[23] Garcia S, Grill M, Stiborek J, et al. An empirical comparison of botnet detection methods [J]. Computers & Security, 2014, 45: 100 – 123.

[24] Hochreiter S, Schmidhuber J. Long short-term memory [J]. Neural Computation, 1997, 9(8): 1735 – 1780.