文章编号:1671-4229(2022)01-0001-09

基于 TextCNN 的加密恶意流量检测方法

杨彦召[1],朱程威[2],仇 晶[2],童咏昕[3*]
(1. 中汽智创科技有限公司,江苏 南京 211100;2. 广州大学 网络空间先进技术研究院,广东 广州 510006;
3. 北京航空航天大学 计算机学院,北京 100191)

摘 要:随着互联网技术的飞速发展,95% 的流量使用 SSL/TLS 协议进行加密,其中隐藏着大量的恶意流量。由于网络流量体量大、加密数据的不可见性,使得如何在不解密的前提下,检测加密恶意流量的研究成为一个重要课题。现有的基于模式匹配的方法,无法处理加密流量。基于统计特征和时序特征的方法,依赖专家经验,需要花费大量的时间,人工提取特征。文章将深度学习算法与加密恶意流量检测领域相结合,首先,对原始的网络流量进行切分、清洗、转换和修剪,变为统一长度的一维序列;然后,自定义 TextCNN 网络结构,通过多组一维卷积自动地从原始流量中提取上下文特征,并利用这些特征对流量进行分类。为了证明该方法的有效性,使用真实的网络流量样本进行实验,并与 CNN、LSTM 和 GRU 等网络模型进行对比。实验数据显示,文章提出的方法,在未知数据上具有较强的泛化能力,检测精度高,且误报率低。

关键词:SSL/TLS;恶意软件;加密恶意流量检测;深度学习;TextCNN
中图分类号:TP 183 **文献标志码**:A

Encrypted malicious traffic detection method based on TextCNN

YANG Yan-zhao[1], *ZHU Cheng-wei*[2], *QIU Jing*[2], *TONG Yong-xin*[3*]
(1. China Automotive Innovation Corporation, Nanjing 211100, China;
2. Cyberspace Institute of Advanced Technology, Guangzhou University, Guangzhou 510006, China;
3. School of Computer Science and Engineering, Beijing University of Aeronautics and Astronautics, Beijing 100191, China)

Abstract: With the rapid development of Internet technology, 95% of traffic is encrypted using SSL/TLS protocol, which hides a large amount of malicious traffic. Because of the large volume of network traffic and the invisibility of encrypted data, how to detect encrypted malicious traffic without decryption becomes an important topic. Existing methods based on pattern matching cannot handle encrypted traffic. Methods based on statistical features and temporal features rely on expert experience and require a lot of time to extract features manually. In this paper, the deep learning algorithm is combined with the field of encrypted malicious traffic detection. First, the original network traffic is segmented, cleaned, converted and pruned into a one-dimensional sequence of uniform length. Then, the TextCNN network structure is customized, and the context features are automatically extracted from the original traffic through multiple groups of one-dimensional convolution, and these features are used to classify the traffic. In order to prove the effectiveness of this method, real network traffic samples were used for experiments and compared with network models such as CNN, LSTM and GRU. Experimental data show that the method proposed in this paper has strong generalization ability on unknown data, high detection accuracy and a low false positive rate.

Key words: SSL/TLS; malware; encrypt malicious traffic detection; deep learning; TextCNN

基金项目:国家自然科学基金资助项目(U20B2046,U1636215,61871140,U1803263);国家重点研发计划资助项目(2018YFB1800702);广东省重点研发计划资助项目(2019B010136003)

作者简介:杨彦召(1985—),男,工程师. E-mail: yangyanzhao@t3caic.com
* 通信作者. E-mail: yxtong@buaa.edu.cn

引文格式:杨彦召,朱程威,仇晶,等. 基于 TextCNN 的加密恶意流量检测方法[J]. 广州大学学报(自然科学版),2022,21(1):1-9.

随着互联网技术的飞速发展,互联网成为生活中不可或缺的一部分。与此同时,网络数据包的安全性和完整性的研究,成为了工业界和学术界共同关注的重要课题。在这样的背景下,SSL(Secure Socket Layer 安全套接层)[1] 和 TLS(Transport Layer Security 传输层安全)[2-3]等网络安全协议运营而生。SSL/TLS 协议,工作在 TCP/IP 的传输层之上,具有模糊应用数据、验证终端身份和数据完整性检验等功能,从而为应用数据提供隐私和完整性保障。

合法用户使用 SSL/TLS 协议对数据加密,保护数据的隐私,防止中间人窃听。恶意软件的作者也使用 SSL/TLS 协议,对恶意软件在通信过程中产生的流量进行加密,产生加密的恶意流量。该流量能够躲避安全设备的检查,对合法用户实施攻击[4]。

为了在大规模的加密流量中,检测恶意流量,目前的研究主要集中在使用传统机器学习和基于深度学习的异常检测方法上。基于传统机器学习的检测方法[5-11]需要人工提取特征,使用 KNN、逻辑回归、SVM、决策树和随机森林等传统机器算法训练恶意流量分类器。随着加密算法不断升级,例如,TLSv1.3 在 TLSv1.2 的基础上加强了加密的力度,在发送了 Server Hello 报文后,后续的数据包都被加密[10],这将导致可提取的特征越来越少。此外,专家经验的片面性,将会极大地限制模型的通用性。

基于深度学习的异常检测方法[12-17],通常把网络流量中报文头部或者指定数量的数据包,转换为灰度图片的形式,输入到 CNN 深度神经网络中,进行异常检测分类器的训练。将原始流量处理成灰度图片的方法,数据预处理方式复杂,同时,无法很好地对原始流量进行表达,只携带部分的报文特征。

本文将深度学习算法与加密恶意流量检测领域相结合,提出了基于 TextCNN 的恶意流量检测方法,该方法的优势总结如下:

(1)将数据样本转换为一维序列,作为模型的输入,既保留了原始流量的表现形式和报文特征,又简化数据预处理的过程;

(2)根据实际数据,自定义 TextCNN 深度神经网络结构,该模型能够自动地进行特征提取,从而避免专家经验的不足和片面性;

(3)模型泛化能力强,在未知的数据集中,能有较高的分类精度和较低的误报率。

本文内容如下:第一部分,相关工作,介绍现有的恶意流量检测方法;第二部分,详细介绍基于 TextCNN 的加密恶意流量检测方法和设计思路;第三部分,进行实验评估;第四部分为总结和展望。

1 相关工作

据 Google 透明度报告[18],于 2021 年 8 月,所有 Google 产品和服务启用加密的已占 95%。而隐藏在加密流量中恶意流量的比例也在逐年增加[6,19-20],目前的加密恶意流量检测方法主要如下:

基于端口的恶意流量检测[21],该方法利用传输层端口号与应用层协议互相对应的原理,根据传输层端口号的取值,对应用数据进行分类。例如,80 端口对应 HTTP 协议,443 端口对应 SSL/TLS 协议等。该方法实现原理简单,但是容易被攻击者绕过。因为,恶意软件的作者可以通过将恶意软件产生的流量的端口号进行修改,修改为合法应用的端口号,从而绕过检测。

深度包检测技术[22],该方法根据网络工作者预先指定的匹配规则,例如,固定格式和关键字等,对数据包内容进行检测。该方法识别精度高,但是只能检测未加密的数据。如果想要用该方法检测加密流量,需要先对流量进行解密,然后再根据预先指定好的规则对流量内容进行检测。然而,对流量进行解密,违背了 SSL/TLS 协议设计的初衷,侵犯用户数据的隐私,同时,对 TLS 协议加密后的数据进行解密,将会消耗大量的计算资源,极大地增加了网络设备的负担和数据包转发延迟时间。

最近几年,研究者的目光不再局限于基于模式匹配的流量检测方法。开始将机器学习算法应用于加密恶意流量检测领域中,即基于统计和基于行为的检测方法。

2016 年,Anderson 等[6] 从 TLS 报文、DNS 报文和 HTTP 报文中,提取密码套件、扩展、服务器证书、DNS 响应报文和 HTTP 报头中的上下文信息作为特征,选择 L1 正则化的逻辑回归作为分类模型。2017 年,Anderson 等[7]从恶意流量和良性流量中提取 2 个不同的特征集,对不同的传统机器算法进行性能测试,验证特征对模型性能的影响。

2018 年,Stergiopoulos 等[8]从恶意软件、web 攻击和僵尸网络等多种攻击产生的网络流量中提取 TCP 侧信道特征,通过该特征对恶意流量和非恶意流量进行分类。2019 年,Shekhawat 等[9]通过开源网络数据包分析工具 zeek,从网络流量中提取 38 个特征分别对 SVM、随机森林和 XGBoost 进行训练,模型训练好之后,使用上述 3 个模型对特征重要性进行分析。

2020 年,Rong 等[23]从原始数据包中提取统计特征

和时序特征。其中,统计特征包括:源端口、目的端口、流的持续时间、发送的数据包数和返回的数据包数等。时序特征包括:数据包长度和达到时间的转移序列、分位数序列和字节分布等。利用 stacking 思想,将随机森林、XGBoost 和 LightGBM 作为基分类,逻辑回归作为元分类器,构造一个集成学习模型。

上述方法,利用未加密的明文特征,或者与负载无关的统计特征和序列特征,对加密恶意流量进行识别。这些方法都需要人工提取特征,依赖专家经验,若专家经验不足或有片面性,将极大地影响模型的分类效果,限制模型的通有性。随着加密算法的不断更新换代,以及网络流量日趋复杂,数据包中可提取的明文特征将会越来越少,这将极大影响模型的性能。

深度学习算法能够在训练过程中,自动进行特征提取,避免了人工提取特征的缺陷。此外,与传统机器学习相比,深度学习具有更强的学习能力。结合这 2 点优势,深度学习成为了恶意流量检测的理想方法。

2017 年,Wang 等[12]将 PCAP 格式的网络数据包转换为 IDX3 和 IDX1 的数据格式,通过卷积神经网络,从数据包中自动提取特征,并对恶意流量和良性流量进行分类。2018 年,Millar 等[13]使用数据包前 50 字节的有效负载,将网络数据包转换为图片和提取元数据特征 2 种方式,分别对数据进行处理,输入到 DNN 和 CNN 网络中进行训练和分类。

2019 年,Hwang 等[14]将包报头中的一个字段视为一个单词,并将包修剪为 $n=54$ 字节的固定长度作为 LSTM 网络的输入。2020 年,Hwang 等[16]从数据流中选取前 n 个数据包,将 n 数据包的长度修剪为固定长度 l,构成 $n*l$ 的灰度图像,使用的网络模型由一维卷积和 Autoencoder 组成。

上述方法存在以下几个问题:①使用的数据集中,多数为未加密的数据;②数据预处理方式复杂,数据处理之后,携带的信息较少;③网络结构简单,泛化能力差。针对上述问题,本文提出基于 TextCNN 的加密恶意流量检测方法,根据网络数据包的特性对数据进行处理,数据预处理方式简单,能够携带更多的网络流量特征。此外,在加密的未知数据上,具有较高精度和较低的误报率。

2　方法和思路

在这个章节中,将从以下 3 个部分介绍本文提出的方法和具体思路。首先,对本文中使用的数据集的来源和保存形式进行介绍。然后,介绍输入模型的数据的具体表现形式。最后,介绍 TextCNN 网络模型。

2.1　数据集介绍

本文中使用的数据集,为 2020 年 datacon 大数据安全分析竞赛中加密恶意流量赛道的公开数据集。该数据源于 2020 年 2 月至 6 月,在沙箱中运行恶意软件和正常软件,收集软件运行过程中产生的数据流量[24]。其中,恶意软件运行产生的流量,标记为恶意流量,正常软件运行产生的流量,标记为良性流量。样本均以 PCAP 文件格式保存,如图 1 所示。流量的内容经过 SSL/TLS 协议加密,如图 2 所示。

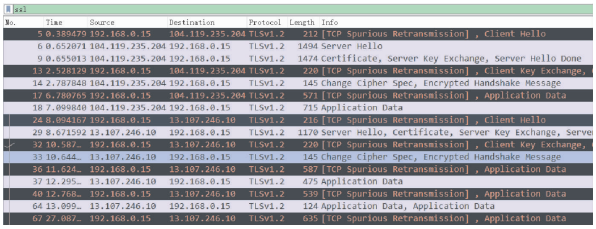

图 1　PCAP 文件

Fig. 1　PCAP files

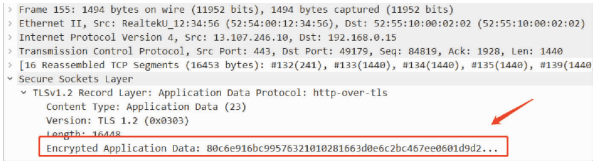

图 2　TLS 加密流量

Fig. 2　TLS encrypted traffic

2.2　数据的表征

将深度学习算法与加密恶意流量检测领域相结合时,存在 4 个问题需要进一步解决:①数据集中,不同的 PCAP 文件具有不同的通信时长和报文信息,需要将 PCAP 文件切分为具有相同粒度的数据单元;②部分的 PCAP 文件中,缺失了区分恶意流量和良性流量样本的关键信息,该类数据将会极大地影响模型的检测精度;③网络数据包样本为 PCAP 文件格式,该数据形式无法输入到模型中进行训练;④不同的客户端和服务器之间,一次通信发送的数据包数量和长度存在差异,但是神经网络的结构需要固定的输入。在本节中,将针对上述问题提出解决方法和思路。

2.2.1　数据切分粒度

为了将 PCAP 文件转换为相同粒度的数据单元,本文使用 PCAP 文件处理工具 SplitCap[25],对 PCAP 文件进行分割,在对 PCAP 文件进行切分时,存在以下切分依据:IP 地址、MAC 地址、流(flow)、会话(session)和数据包数量等。其中,具有相同的源 IP 地址、目的 IP 地

址、源端口、目的端口和协议号的数据包组成流,因此,流是单向的。而会话由双向的流组成。由于一个会话代表着客户端和服务器的一次完整通信,携带的特征数量比一个流要多。因此,将会话作为切分 PCAP 文件的依据。

在分割 PCAP 文件时,可以选择从数据包中提取所有层的信息,或者仅提取应用层的信息。由于所有层的信息中,包含传输层字段信息、TLS 报文头部信息和应用层有效负载,因此,比仅提取应用层信息得到的数据包特征更加丰富。当对数据包进行处理切分时,应当提取数据包中所有层的信息。

根据上述要求,对原始 PCAP 文件进行切分之后,会得到多个小的 PCAP 文件,每个小的 PCAP 文件对应一个会话,即每个小的 PCAP 文件中包含某一客户端和某一服务器之间的某一次会话的所有通信数据。

2.2.2 数据清洗

经过上述的处理流程,不同的 PCAP 文件转换成了相同粒度的数据单元。但是,部分 PCAP 文件中可能丢失了对异常流量和正常流量进行区分的关键信息。而这一部分的数据,如果输入到模型中进行训练,将会极大地影响模型的分类性能。因此,需要从数据集中清除缺失关键信息的数据。

为了解决这个问题,首先需要了解什么样的通信是完整的。假设,一个客户端和一个服务器之间进行通信,在通信过程中使用 TLS 协议对数据进行加密。以目前常用的 TLSv1.2 为例,该会话将会包括以下 3 个阶段:①TCP 3 次握手,建立 TCP 连接通道;②TLS 握手,该阶段会交互 TLS 协议报文,包括 client hello、server hello、certificate、server key exchange 和 client key exchange 报文等,对服务器身份进行验证,对密钥信息进行协商;③交互应用数据,该数据使用 TLS 协议进行加密。

综上所述,一个完整的 TLS 通信会话的标志是完成了 TLS 握手,互相进行了应用数据的交互。因此,将没有完成 TLS 握手,或者没有交互应用数据的会话进行丢弃,从而实现数据的清洗。

2.2.3 数据转换

为了让 TextCNN 深度神经网络从报文中学习上下文信息,需要将 PCAP 文件转换为能够输入模型的数据格式。而在数据转换过程中,保留数据原始结构和足够多的报文特征至关重要。

网络数据包在传递过程中,即在物理层中,以 0 和 1 组成的二进制比特流的形式进行传递。每 8 bit 的二进制数值对应数据包中的某一个字段信息。数据包的信息严格按照 TCP/IP 协议栈的形式进行封装,例如,一个 web 访问中,服务器发送的数据包的结构分为物理层、数据链路层、网络层、传输层和应用层[26-27]。如图 3 所示,网络数据包每个层次、每个字段之间有着严格的顺序关系。

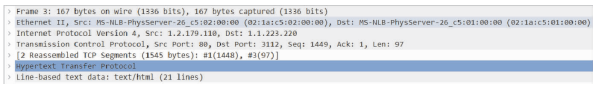

图 3 网络数据包

Fig. 3 Network data package

为了还原和保留数据包在网络中的原始状态,将 PCAP 文件转换为一维序列的形式,作为模型的输入。操作流程如图 4 所示,恶意流量和良性流量的 PCAP 文件经过切分之后,变成多个小的 PCAP 文件。将小的 PCAP 文件以二进制的形式进行读取,以 8 bit 为一组进行分组。然后,将 8bit 的二进制数转换为对应的十进制数值,得到元素为十进制数值的一维数组。例如,二进制的 0000 0000,对应十进制的 0。二进制的 1111 1111,对应十进制的 255。因此,该十进制一维数组中每个元素都在[0,255]之间。而每个数据的标签,由原来数据集中 PCAP 文件的类型决定。

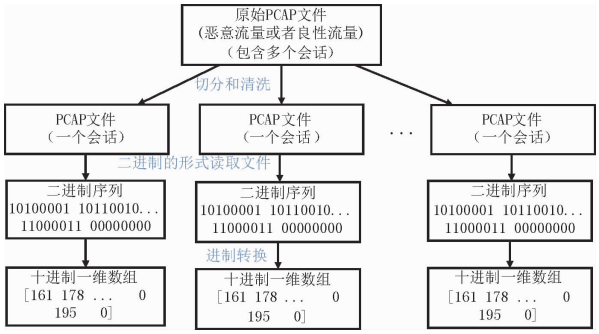

图 4 数据处理流程

Fig. 4 Data processing process

2.2.4 数据裁剪

不同的通信会话中,传递的网络数据包数量和大小存在差异,使得转换后得到的十进制一维数组的长度存在差异。由于网络结构是固定的,输入网络的数据的形状也是固定的。因此,需要将所有的十进制一维数组转换为相同长度。

本文中,固定长度的取值 L,通过对切分后的数据包的长度进行统计获得,如果十进制一维数组的长度小于或等于 L 的数量 N,占所有样本数量 M 的比例为 71%。同时,在 L 之上,长度增加 200,增加 5 次,每次增加数据包长度时,占比的增加均小于 1%,则将所有数据的长度都转换为 L。数据包长度和占比如图 5 所示。

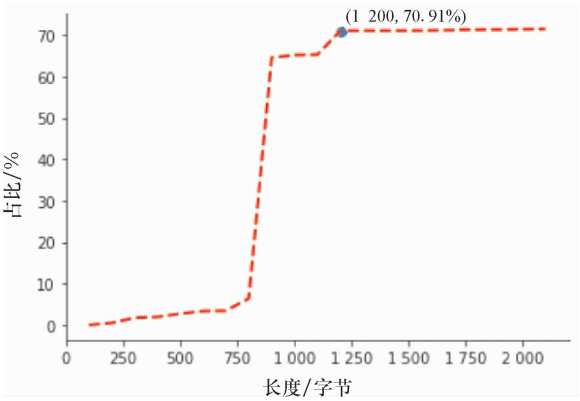

图 5　数据包长度和占比

Fig. 5　Package length and proportion

如图 5 所示,横坐标为数据包长度,纵坐标为某长度的数据包数量占所有数据样本的比例。其中,数据包长度小于等于 1 200 的数据包所有样本比例约 71%,在 1 200 之上,长度增加 200,增加 5 次,占比变化均小于 1%。因此,本文中使用的固定长度为 1 200,当数组长度大于 1 200 时,只保留前 1 200 个数值。当数组长度小于 1 200 时,用 0 填充至 1 200。

上述方法的好处有:①保存了尽可能多的数据信息;②确保较短的数据,不会被大量的 0 填充;③避免输入数据长度过大,导致模型训练速度慢的问题。

2.3　TextCNN 模型

本文中设计的 TextCNN 网络结构,如表 1 所示。网络层次包括:输入层、词嵌入层、SpatialDropout1D、5 对并行的一维卷积和一维全局最大池化层、Dropout、批标准化、全连接层、Dropout、批标准化、全连接层、Dropout、批标准化与输出层等。

该网络结构中词嵌入层,从多个维度对单词进行描述,将一个单词转换为对应的多维向量。多个不同种类的单词组合在一起,就形成了词嵌入表。该词嵌入表能够很好地表示不同单词之间的关系,在文本分类任务中,能够很好地表示词与词之间的上下文关系。在加密恶意流量检测领域中,利用词嵌入层的优点,对数据包各个层次、各个字段之间的上下文关系进行描述。由于十进制一维数组的每个元素取值范围都在 0 到 255 之间,一共 256 种取值。因此,建立了一张词嵌入表,其单词种类数为 256,每个单词的维度为 256。

表 1　TextCNN 网络结构

Table 1　TextCNN network structure

层数	名称	输入	过滤器	输出
1	Embedding	1 200	–	1 200 * 256
2	SpatialDropout1D	1 200 * 256	–	1 200 * 256
3	Conv1d_1	1 200 * 256	2 * 64	1 199 * 64
	Conv1d_2	1 200 * 256	3 * 64	1 198 * 64
	Conv1d_3	1 200 * 256	4 * 64	1 197 * 64
	Conv1d_4	1 200 * 256	5 * 64	1 196 * 64
	Conv1d_5	1 200 * 256	6 * 64	1 195 * 64
4	GlobalMaxPoling1D_1	1 199 * 64	–	64
	GlobalMaxPoling1D_2	1 198 * 64	–	64
	GlobalMaxPoling1D_3	1 197 * 64	–	64
	GlobalMaxPoling1D_4	1 196 * 64	–	64
	GlobalMaxPoling1D_5	1 195 * 64	–	64
5	Concatenate	5 * 64	–	320
6	Dropout_1	320	–	320
7	BatchNormalization_1	320	–	320
8	Dense_1	320	–	256
9	Dropout_2	256	–	256
10	BatchNormalization_2	256	–	256
11	Dense_2	256	–	128
12	Dropout_3	128	–	128
13	BatchNormalization_3	128	–	128
14	Dense_3 + SoftMax	128	–	2

TextCNN 通过不同大小的卷积核提取词嵌入表中的关键信息[28]。本文中利用5个不同大小的一维卷积核,对词嵌入表进行上下文信息提取。在对词嵌入表进行特征提取时,卷积核从上至下移动进行卷积运算(对应位置相乘,然后相加,得到的值作为卷积运算的输出)。

由于一个单词的词嵌入维度是固定的,因此,卷积核的宽度必须是固定的,宽度大小等于词嵌入的维度。不同卷积核的高度不同,不同的高度从词嵌入表中提取到不同特征值。假设卷积核的高度为 h,词嵌入的高度为 H,则卷积运算后得到的特征图为 $(H-h+1)*1$。全局最大池化层对特征进行降维,从特征图中选取最大值作为输出。

5对并行的一维卷积层和全局最大池化层分别得到的输出整合在一起,构成 TextCNN 网络的核心部分。将上述 TextCNN 的工作原理运用于恶意流量检测中,能够充分提取恶意样本和良性样本的字段和有效负载中存在明显差异的上下文信息。

在网络中加入 Dropout 和批标准化。Dropout,让模型在训练时,随机选择部分的神经元不参与训练,有助于防止过拟合。批标准化将数据进行归一化处理,使其平均值接近于0,标准偏差将接近于1,使得数据分布更有规律,提升模型的训练速度。

全连接层是标准的神经网络,主要的参数是神经元的个数。将卷积和池化运算后得到的特征图,输入到全连接层中,进行进一步的特征提取。本文将神经元个数为2的全连接层作为输出层,选择 SoftMax 作为激活函数。最终,得到输入样本属于恶意流量概率值 P1,以及良性流量的概率值 P2、P1 和 P2 之和为1,取值较大者将作为输出结果。

3 实验评估

在本章节中,将对实验环境、步骤、结果和不同网络模型之间的效果对比进行详细描述。基于公开数据集,证明本文提出的基于 TextCNN 的加密恶意流量检测方法的有效性。

3.1 实验环境

实验在操作系统为 Ubuntu 20.04.2 的服务器上进行,该服务器的 CPU 型号为 Intel(R) Xeon(R) Gold 5218R CPU @ 2.10 GHz,其核数为20个。GPU 型号为 GeForce RTX 2080 Ti,显存的大小为 10 G。内存空间为 96 G。

在该服务器上,安装 Anaconda,一个集成的 python 开发环境。通过深度学习框架 Tensorflow 搭建 TextCNN 网络结构,该网络中 SpatialDropout1D 的随机丢弃概率为0.15,Dropout 的随机丢弃概率为0.3。模型优化器使用基于随机梯度下架的 Adam,该方法计算效率高,同时内存消耗低,其学习率采用默认值0.001。由于数据的标签为0或者1的整数,因此,选择稀疏类别交叉熵作为损失函数。

3.2 实验步骤

如图6实验步骤所示,实验分5个部分:①数据切分、清洗、转换和裁剪;②划分训练集和测试集;③模型的训练和验证;④模型测试;⑤预测结果评估。

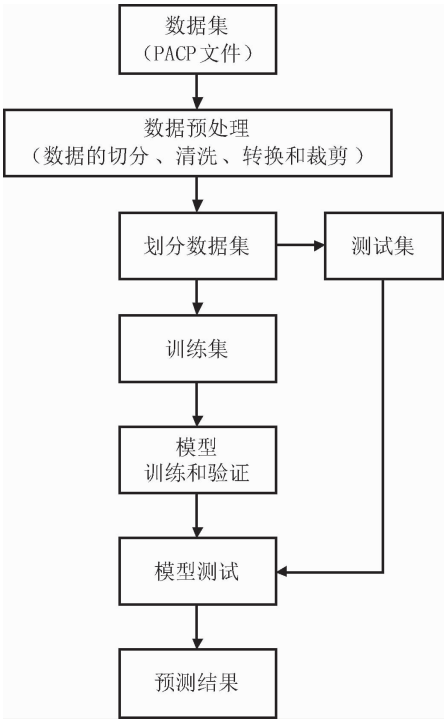

图6 实验步骤

Fig. 6 Experimental Procedure

使用的数据集和数据预处理的过程,在3.1和3.2节中已有详细介绍。经过数据预处理之后,恶意流量和良性流量的会话数量,如表2所示。此时,恶意流量和良性流量数据,均以十进制一维数组的形式保存。

表2 会话统计

Table 2 Session statistics

流量类型	会话数量
Malware	94 256
Benign	31 498
All flows	125 754

在流量划分阶段,为了更好地验证模型的泛化能

力,将恶意流量和良性流量会话随机划分为 2 个部分。其中,训练集占所有流量的比例为 60%,测试集的占比为 40%。

在模型训练和验证阶段,采用 5 折交叉验证的方式进行,将训练集随机划分为 5 个部分,每一个部分轮流做一次验证样例,其余 4 个部分作为训练样例[29]。这样就得到 5 个模型和 5 个验证结果,对 5 个验证结果求和取平均,得到模型的验证评分。

在 5 折交叉验证的过程中,每一折训练的 epoch 为 100,batch 为 64。通过 Early Stopping,监视每次训练的 Validation Loss,如果 Validation loss 在 3 次验证中都没有提升,则提前终止训练过程。

3.3　模型评估指标

在模型测试阶段,通过测试集对模型的泛化能力进行验证。而模型的预测效果的评估指标为:精度(Accuracy)、查准率(Precision)、查全率(Recall)、F1 值(F1-score)、真正例率(True Positive Rate TPR)和假正例率(False Positive Rate FPR)[30]。

计算上述指标之前,需要先了解 *TP*、*FP*、*TN* 和 *FN*。*TP*(True Positive 真正例),表示预测为恶意流量,实际也为恶意流量的样本数量。*FP*(False Positive 假正例),表示预测为恶意流量,实际为良性流量的样本数量。*TN*(True Negative 真负例),表示预测为良性流量,实际也为良性流量的样本数量。*FN*(False Negative 假负例),表示预测为良性流量,实际为恶意流量的样本数量。

精度:

$$Acc = \frac{TP + TN}{TP + FP + TN + FN};$$

查准率:

$$Pre = \frac{TP}{TP + FP};$$

查全率:

$$Rec = \frac{TP}{TP + FN};$$

F1 值:

$$F1\ Score = \frac{2 \times Pre \times Rec}{Pre + Rec};$$

假正例率:

$$FPR = \frac{FP}{TN + FP}。$$

3.4　实验结果

在实验过程中,在不同的网络模型上进行训练和测试,进行对比实验。涉及的网络模型包括本文提出的 TextCNN 网络模型、CNN 网络模型的变体、LeNet-5[31] 和 ResNet-12[32],以及在文本分类中常用的 LSTM、双向 LSTM、GRU 和双向 GRU。

上述网络模型中,在进行 LeNet-5 和 ResNet-12 实验时,将十进制一维数组,转换为 32 * 32 * 1 的灰度图片作为模型的输入。LeNet-5 的网络结构如表 3 所示,ResNet-12 的网络结构如表 4 所示。

表 3　LeNet-5 网络结构

Table 3　LeNet-5 network structure

层数	名称	输入	过滤器	输出
1	Conv2D	32 * 32 * 1	5 * 5	32 * 32 * 32
2	MaxPool	32 * 32 * 32	2 * 2	16 * 16 * 32
3	Conv2D	16 * 16 * 32	5 * 5	16 * 16 * 64
4	MaxPool	16 * 16 * 64	2 * 2	8 * 8 * 64
5	Dense	8 * 8 * 64	–	1 024
6	Dense	1 024	–	2

表 4　ResNet-12 网络结构

Table 4　ResNet-12 network structure

层数	名称	输入	过滤器	输出
1	Conv2D	32 * 32 * 1	3 * 3	30 * 30 * 32
2	Conv2D	30 * 30 * 32	3 * 3	28 * 28 * 64
3	MaxPool	28 * 28 * 64	3 * 3	9 * 9 * 64
4	Conv2D	9 * 9 * 64	3 * 3	9 * 9 * 64
5	Conv2D	9 * 9 * 64	3 * 3	9 * 9 * 64
6	add	9 * 9 * 64	–	9 * 9 * 64
7	Conv2D	9 * 9 * 64	3 * 3	9 * 9 * 64
8	Conv2D	9 * 9 * 64	3 * 3	9 * 9 * 64
9	add	9 * 9 * 64	–	9 * 9 * 64
10	Conv2D	9 * 9 * 64	3 * 3	7 * 7 * 64
11	Global Avg Pool2D	7 * 7 * 64	–	64
12	Dense	64	–	256
13	Dense	256	–	2

对比实验中的 LSTM、Bi-LSTM、GRU 和 Bi-GRU 网络结构,是在本文提出的 TextCNN 网络结构的基础上进行修改得到的。将 TextCNN 的核心部分,5 对并行的一维卷积层和最大池化层,更换为一层的 LSTM、bi-LSTM、GRU 和 bi-GRU 中之一,神经元个数设置为 256。

使用 3.2 节中提到的训练方法对上述模型进行训练,然后在测试集上进行测试。在训练时间方面,LeNet-5 和 ResNet-12 训练时间最短,TextCNN 的训练时间适中,而 LSTM、Bi_LSTM、GRU 和 Bi-GRU 的训练时间最长。各个网络模型的测试结果如表 5 所示。

表5 实验结果与对比

Table 5 Experimental results and comparison %

模型	精确度	查准率	召回率	F1值	假正例率
LeNet-5	94.55	95.17	97.70	96.42	14.92
ResNet-12	95.58	96.30	97.85	97.07	11.18
LSTM	96.15	96.57	98.36	97.46	10.50
Bi_LSTM	96.90	97.18	98.73	97.95	8.61
GRU	97.19	97.75	98.71	98.14	7.40
Bi-GRU	98.08	98.00	99.47	98.73	6.10
TextCNN	98.76	98.92	99.43	99.17	3.27

从表5中能够发现,本文提出的TextCNN网络结构在加密恶意流量检测中,各项评估指标都取得了较高的分数。模型的预测精度为98.72%,查全率为98.91%,F1值为99.15%,误报率为3.28%,在所有模型中表现最优。而在查准率,即真正例率上,TextCNN和双向GRU的得分非常接近。实验数据说明,基于TextCNN的加密恶意流量检测方法,具有较强的泛化能力,在未知数据集上,也能表现出不错的效果,模型检测精度高,同时误报率低。

4 总结和展望

在不解密的前提下,从加密流量中检测恶意流量,是一个非常热门的课题。现在常用的方法中,基于传统机器学习的恶意流量检查方法,需要花费大量时间,人工提取特征。基于深度学习的恶意流量检测方法,数据预处理方式复杂,模型泛化能力差。对此,本文提出了一种基于TextCNN的加密恶意流量检测方法。

该方法将恶意流量和良性流量的分类任务,转换为文本分类任务进行处理。结合网络数据包按照TCP/IP协议栈的格式进行封装的领域知识,将网络流量根据会话切分,然后进行清洗、转换和裁剪,处理为等长的十进制一维数组。通过词嵌入层,充分描述网络流量中的上下文关系,通过多组一维卷积,从原始的流量中自动学习上下文特征。将多组一维卷积中学习到的特征结合起来,对流量进行分类。

在实际的网络流量数据上,进行实验验证,证明了基于TextCNN的加密恶意流量检测方法的有效性。实验结果显示,该方法具有良好的性能,优于CNN、LSTM和GRU等网络模型。此外,该方法在实际的网络流量数据上训练后,能够在未知数据集上表现出较高的检测精度和较低的误报率,模型泛化能力强。

在未来的研究中,将会从以下3个方面进行尝试:

(1)将最新的网络模型和技术,运用于加密恶意流量检测中。

(2)将多任务共享参数的设想加入到加密恶意流量检测中,对良性流量和恶意流量进行二分类任务,同时对恶意流量进行更细致的多分类,区分是哪种攻击产生的流量。

(3)本文中采用的是机器学习中有监督学习的分类模型,流量样本都经过人工进行标签的标注。但是,在实际网络中,标注流量是比较困难的。因此,研究如何从少量带标签样本和大量无标签样本中进行半监督学习,将会是加密流量检测中的一个重要研究课题。

参考文献:

[1] Wagner D, Schneier B. Analysis of the SSL 3.0 protocol[J]. The Second USENIX Workshop on Electronic Commerce Proceedings, 1996, 1(1): 29-40.

[2] Dierks T, Rescorla E. RFC 5246-The transport layer security (TLS) protocol Version 1.2[EB/OL]. (2008-08-01)[2021-04-09]. https://www.hjp.at/doc/rfc/rfc5246.html.

[3] Rescorla E, Dierks T. RFC 8446-The transport layer security (TLS) protocol version 1.3[EB/OL]. (2018-08-01)[2021-04-09]. https://www.hjp.at/doc/rfc/rfc8446.html.

[4] Qiu J, Tian Z, Du C, et al. A survey on access control in the age of internet of things[J]. IEEE Internet of Things Journal, 2020, 7(6):4682-4696.

[5] Anderson B, Paul S, McGrew D. Deciphering malware's use of TLS (without decryption)[J]. Journal of Computer Virology and Hacking Techniques, 2018, 14(3): 195-211.

[6] Anderson B, McGrew D. Identifying encrypted malware traffic with contextual flow data[C]//Proceedings of the 2016 ACM workshop on artificial intelligence and security. New York: ACM, 2016: 35-46.

[7] Anderson B, McGrew D. Machine learning for encrypted malware traffic classification: Accounting for noisy labels and non-stationarity[C]//Proceedings of the 23rd ACM SIGKDD International Conference on knowledge discovery and data mining. New York: ACM, 2017: 1723-1732.

[8] Stergiopoulos G, Talavari A, Bitsikas E, et al. Automatic detection of various malicious traffic using side channel features

on TCP packets[C]//European Symposium on Research in Computer Security. Berlin: Springer, 2018: 346-362.
[9] Shekhawat A S, Di T F, Stamp M. Feature analysis of encrypted malicious traffic[J]. Expert Systems with Applications, 2019, 125: 130-141.
[10] Roques O. Detecting malware in TLS traffic[D]. London: Imperial College London, 2019.
[11] Luo C, Tan Z, Min G, et al. A novel web attack detection system for internet of things via ensemble classification[J]. IEEE Transactions on Industrial Informatics, 2020, 17(8): 5810-5818.
[12] Wang W, Zhu M, Zeng X, et al. Malware traffic classification using convolutional neural network for representation learning[C]//2017 International Conference on Information Networking (ICOIN). Piscataway: IEEE, 2017: 712-717.
[13] Millar K, Cheng A, Chew H G, et al. Deep learning for classifying malicious network traffic[C]//Pacific-Asia Conference on Knowledge Discovery and Data Mining. Berlin: Springer, 2018: 156-161.
[14] Hwang R H, Peng M C, Nguyen V L, et al. An LSTM-based deep learning approach for classifying malicious traffic at the packet level[J]. Applied Sciences, 2019, 9(16): 1-14.
[15] Yang J, Liang G, Li B, et al. A deep-learning and reinforcement-learning-based system for encrypted network malicious traffic detection[J]. Electronics Letters, 2021, 57(9): 363-365.
[16] Hwang R H, Peng M C, Huang C W, et al. An unsupervised deep learning model for early network traffic anomaly detection[J]. IEEE Access, 2020(8): 30387-30399.
[17] Tian Z, Luo C, Qiu J, et al. A distributed deep learning system for web attack detection on edge devices[J]. IEEE Transactions on Industrial Informatics, 2019, 16(3): 1963-1971.
[18] Google. HTTPS encryption on the web[EB/OL]. (2021-09-01)[2021-09-12]. https://transparencyreport.google.com/https/overview.
[19] Malwarebytes. Analysis of malware trends for small and medium businesses[EB/OL]. (2017-03-31)[2021-09-01]. https://www.malwarebytes.com/pdf/white-papers/MalwareTrendsForSMBQ12017.pdf.
[20] Radware. Global application and network security report[EB/OL]. (2017-12-01)[2021-09-01]. https://www.datacom.cz/userfiles/radware_ert_report_2017_2018_final.pdf.
[21] Zheng W, Gou C, Yan L, et al. Learning to classify: A flow-based relation network for encrypted traffic classification[C]//Proceedings of The Web Conference 2020. New York: ACM, 2020: 13-22.
[22] Sen S, Spatscheck O, Wang D. Accurate, scalable in-network identification of p2p traffic using application signatures[C]//Proceedings of the 13th international conference on World Wide Web. New York: ACM, 2004: 512-521.
[23] Rong C, Gou G, Cui M, et al. MalFinder: An ensemble learning-based framework for malicious traffic detection[C]//2020 IEEE Symposium on Computers and Communications (ISCC). Piscataway: IEEE, 2020: 7-7.
[24] 奇安信集团,清华大学. 加密恶意流量[EB/OL]. (2020-07-13)[2021-09-01]. https://datacon.qianxin.com/opendata/maliciousstream.
[25] NETRESEC. SplitCap[EB/OL]. (2010-2021)[2021-09-01]. https://www.netresec.com/index.ashx?page=SplitCap.
[26] Lu H, Jin C, Helu X, et al. DeepAutoD: Research on distributed machine learning oriented scalable mobile communication security unpacking system[J]. IEEE Transactions on Network Science and Engineering, 2021. doi: 10.1109/TNSE.2021.3100750.
[27] Richardstevens W. TCP/IP 详解. 卷 1,协议[M]. 北京:机械工业出版社, 2000.
[28] Kim Y. Convolutional neural networks for sentence classification[EB/OL]. (2014-08-23)[2021-08-30]. https://arxiv.org/abs/1408.5882.
[29] 天池平台. 阿里云天池大赛赛题解析[M]. 北京:电子工业出版社, 2020.
[30] 周志华. 机器学习[M]. 北京:清华大学出版社, 2016.
[31] Wang W, Zhu M, Wang J, et al. End-to-end encrypted traffic classification with one-dimensional convolution neural networks[C]//2017 IEEE International Conference on Intelligence and Security Informatics (ISI). Piscataway: IEEE, 2017: 43-48.
[32] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE, 2016: 770-778.

【责任编辑: 周　全】