计算机科学
COMPUTER SCIENCE

基于多粒度统计特征的僵尸网络流量智能检测方法

张海霞, 黄克振, 连一峰, 赵昌志, 袁云静, 彭媛媛

引用本文

张海霞, 黄克振, 连一峰, 赵昌志, 袁云静, 彭媛媛. 基于多粒度统计特征的僵尸网络流量智能检测方法[J]. 计算机科学, 2025, 52(11): 373-381.

ZHANG Haixia, HUANG Kezhen, LIAN Yifeng, ZHAO Changzhi, YUAN Yunjing, PENG Yuanyuan. Intelligent Botnet Traffic Detection Method Based on Multi-granularity Statistical Features[J]. Computer Science, 2025, 52(11): 373-381.

相似文章推荐（请使用火狐或 IE 浏览器查看文章）

Similar articles recommended (Please use Firefox or IE to view the article)

基于变体注意力的关系与属性感知实体对齐

Relationship and Attribute Aware Entity Alignment Based on Variant-attention

计算机科学, 2025, 52(11): 230-236. https://doi.org/10.11896/jsjkx.240800140

基于联合注意力机制与多阶段特征提取的图像去雨

Image Deraining Based on Union Attention Mechanism and Multi-stage Feature Extraction

计算机科学, 2025, 52(11): 206-212. https://doi.org/10.11896/jsjkx.240900013

基于多分支注意力和深度下采样的医疗图像目标检测方法

Medical Image Target Detection Method Based on Multi-branch Attention and Deep Down- sampling

计算机科学, 2025, 52(11): 196-205. https://doi.org/10.11896/jsjkx.240900088

基于VMD复合神经网络模型的手势动作预测

Gesture Action Prediction Based on VMD Composite Neural Network Model

计算机科学, 2025, 52(11): 166-174. https://doi.org/10.11896/jsjkx.241000115

基于细粒度注意力机制的人与物体交互检测

Human-Object Interaction Detection Based on Fine-grained Attention Mechanism

计算机科学, 2025, 52(11): 141-149. https://doi.org/10.11896/jsjkx.240900113

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.241100019

基于多粒度统计特征的僵尸网络流量智能检测方法

张海霞[1] 黄克振[1] 连一峰[1] 赵昌志[2] 袁云静[1] 彭媛媛[1]
1 中国科学院软件研究所可信计算与信息保障实验室 北京 100190
2 中国科学院信息工程研究所 北京 100085
(haixia@iscas.ac.cn)

摘 要 随着信息技术的飞速发展,僵尸网络攻击已成为具有高危害程度的网络安全威胁,及时有效的僵尸网络检测和处置可以遏制攻击者利用僵尸网络发起其他衍生攻击。当前已有的僵尸网络检测方法存在因特征选取视角单一而易被攻击者绕过、误报率高等局限性。对此,提出一种基于多粒度统计特征的僵尸网络流量智能检测方法,该方法提取待检测网络流量的局部粗粒度统计特征和面向源IP的全局细粒度画像,进而利用具有多头注意力机制的长短期记忆网络挖掘良性网络流量与僵尸网络流量在两类统计特征方面存在的差异性,最终基于这些差异性识别僵尸网络流量。在CTU-13和ISCX僵尸网络数据集上进行了对比实验,该方法在准确率、精确率、召回率和F1分数均可达到99%以上。

关键词:网络攻击;僵尸网络;统计特征;注意力机制;长短期记忆网络
中图分类号 TP311.1

Intelligent Botnet Traffic Detection Method Based on Multi-granularity Statistical Features

ZHANG Haixia[1],HUANG Kezhen[1],LIAN Yifeng[1],ZHAO Changzhi[2],YUAN Yunjing[1] and PENG Yuanyuan[1]
1 Trusted Computing and Information Assurance Laboratory,Institute of Software,The Chinese Academy of Science,Beijing 100190,China
2 Institute of Information Engineering,The Chinese Academy of Science,Beijing 100085,China

Abstract With the rapid development of information technology,botnet attacks have become a higly harmful cyber security threat. Botnet detection and disposal can prevent attackers from launching other derivative attacks based on botnets. The current botnet detection methods have limitations such as single feature selection perspective,easy to be bypassed or high false alarm rate. In response to these limitations,this paper proposes an intelligent botnet traffic detection method based on multi-granularity statistical features. This method extracts local coarse-grained statistical features of the network flows to be detected and global fine-grained profile of the source IP based on historical network flows,and then uses the long-short term memory networks with a multi-head attention mechanism to mine the difference in these features between benign network flows and botnet flows at different times. The botnet is ultimately identified based on these differences. Comparative experiments are conducted on the CTU-13 and ISCX botnet datasets,the proposed method achieves more than 99% in accuracy,precision,recall and F1 score.

Keywords Cyber attack,Botnet,Statistical features,Attention mechanism,Long-short term memory

1 引言

随着互联网、5G、物联网等技术的发展,网络空间内的软硬件数量越来越多,暴露给攻击者的攻击面也日益增多。攻击者利用这些攻击面进行各类攻击,给国家网络空间的安全造成了诸多威胁,典型的威胁如僵尸网络。僵尸网络是受僵尸程序感染的主机的集合,其由僵尸管理程序通过控制协议进行控制,可分布于局域网内部和互联网中。攻击者可利用僵尸网络实施分布式拒绝服务(Distributed Denial-of-Service,DDOS)攻击、垃圾邮件投递、勒索软件传播、敏感信息窃取等,具有巨大的危害性。例如,Qakbot僵尸网络感染超过700000台计算机,在全球造成了数亿美元的损失[1]。及时有效的僵尸网络检测可遏制攻击者基于僵尸网络发起其他衍生攻击,保障网络空间的安全,降低网络攻击对社会和经济的影响。

当前僵尸网络的检测方法可分为基于主机行为的检测方法、基于网络行为的检测方法和基于综合关联行为的方法[2]。基于主机行为的检测方法和基于综合关联行为的方法需采集和汇聚主机端修改系统注册表、创建网络连接等相关的系统调用信息,系统调用信息的采集会损耗主机端计算资源,信息

到稿日期:2024-11-04 返修日期:2025-03-04
基金项目:国家重点研发计划(2023YFB3107203)
This work was supported by the National Key Research and Development Program of China(2023YFB3107203).
通信作者:黄克振(kezhen@iscas.ac.cn)

汇聚会占用大量的网络链路带宽。基于网络行为的检测方法大多基于网络流量开展检测工作，已有的基于网络流量的检测方法又包括基于网络流量负载内容的检测方法和基于网络流统计特征的检测方法[3]。其中，随着网络数据包量的增加和端到端加密的发展，基于网络流量负载内容的检测方法难以及时有效地开展僵尸网络检测工作；基于网络流统计特征的检测方法则是将一段时间内的具有相同属性的网络数据包聚合成网络流（如 Netflow 协议[4]）并基于网络流的统计信息进行检测，在一定程度上缓解了网络数据包量大和端到端加密与僵尸网络检测需求之间的矛盾。

随着机器学习和深度学习的发展，研究人员尝试将支持向量机（SVM）[5]、K-近邻（KNN）[6]、决策树[7]、随机森林[8]、卷积神经网络[9]等机器学习或深度学习算法引入僵尸网络检测过程中，在一定程度上提升了僵尸网络检测方法的扩展性和准确率，但是仍存在因特征选取视角单一而易被攻击者借助调整网络数据包结构等方法绕过[10]或误报率高[11]的问题。

针对当前僵尸网络检测中存在的上述问题，本文提出一种基于多粒度统计特征的僵尸网络流量智能检测方法。该方法首先从待检测的网络流量中进行网络流元数据提取、数据填充及转换等数据预处理动作，以形成标准化的分析源数据；然后从局部和全局两种视角分别提取粗粒度的统计特征和绘制源 IP 网络流细粒度画像，并采用字典映射的方式将统计特征转换为数学向量；最后利用具有多头注意力机制的 LSTM（Long-Short Term Memory）僵尸网络流量检测模型（M-AttLSTMBotModel）挖掘僵尸网络流与良性网络流在上述统计特征中存在的差异，基于这些差异识别僵尸网络流，达到检测僵尸网络的目的。

本文的主要贡献有：

1）在已有研究的局部视角粗粒度统计特征提取的基础上，引入面向源 IP 的全局网络流细粒度的画像，为深度学习类模型挖掘僵尸网络在不同时段内或全生命周期发出的网络流量呈现的特征提供了可能；

2）利用具有多头注意力机制的 LSTM 模型深度挖掘僵尸网络和良性网络流量在不同时序的统计特征的差异性，并基于这些差异性实现僵尸网络流量的检测。

2 基于多粒度统计特征的僵尸网络流量智能检测框架

如图 1 所示，僵尸网络检测框架包括流量数据预处理、局部网络流粗粒度统计特征提取、面向源 IP 的全局网络流细粒度画像、网络流特征向量化、利用 M-AttLSTMBotModel 检测僵尸网络流量 5 个阶段。

1）在流量数据预处理阶段，从网络流量中提取网络流开始时间、持续时长、源 IP、源端口、目的 IP、目的端口、源 IP 发送至目的 IP 的数据包数量及字节数等元数据，并进行缺失值默认填充、异常值转换、数据类型转换等数据处理动作，形成标准化数据，便于后续数据分析。

2）在局部网络流粗粒度统计特征提取阶段，基于 δ 时间段内网络流元数据统计与待检测流属性相同的特征，典型的如与待检测网络流中具有相同源 IP 和目标 IP 的网络流数量、具有相同源 IP 和目标端口的网络流数量等。

3）在面向源 IP 的全局网络流细粒度画像阶段，主要是从与待检测网络流中源 IP 相同的全部历史网络流中统计目标端口访问次数、网络流平均时长、数据包平均数量及平均字节数等。

4）在网络流特征向量化阶段，将提取的局部网络流粗粒度统计特征、面向源 IP 的全局网络流细粒度画像转换为数学向量，方便后续僵尸网络流检测模型使用。

5）在僵尸网络流检测阶段，利用 M-AttLSTMBotModel 分析待检测网络流提取特征的数学向量，以判断待检测网络流是否为僵尸网络相关的网络流。

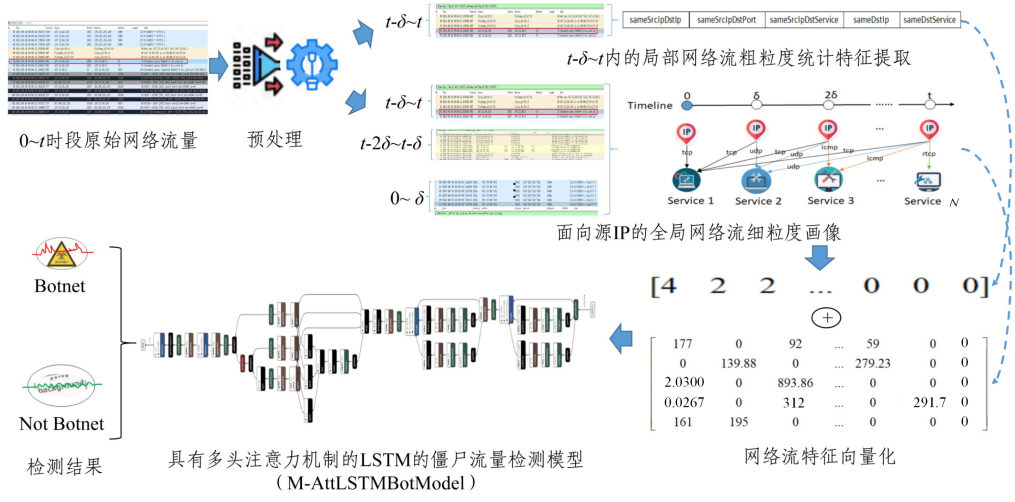


图 1 基于多粒度统计特征的僵尸网络流量智能检测框架

Fig. 1 Intelligent detection framework for Botnet traffic based on multi-granularity statistical features

下文分别对网络流统计特征提取相关算法、基于注意力机制和 LSTM 的僵尸网络流检测算法进行详细阐述。

3 网络流统计特征提取

本文中网络流统计特征主要从局部和全局两个不同的视角开展：在局部视角中进行粗粒度的同属性特征统计，以发现僵尸网络流量中存在的时序性局部行为；在全局视角中进行面向源 IP 的细粒度的画像，以挖掘不同时刻的网络流量中存在时序性全局行为。

3.1 基于时间片的局部网络流粗粒度统计特征提取算法（timeSliceCoarStatFeatExtAlg）

定义 1（网络流，networkFlow，nf） 表示一定时间间隔内通过网络中某个观测点的具有相同属性或特征的一组网络数据包[11-12]，一般包括网络流 id、开始时间 $startTime$、持续时长 dur、协议类型 $proto$、通信的方向 dir、源 IP 地址 sip、源端口 $sport$、目标 IP 地址 dip、目标端口 $dport$、数据流的状态 $state$、网络流中发送数据包总量 $toTPkts$、网络流中发送数据的总字节数 $toTBytes$ 等元数据字段，即 $nf=\langle id, startTime, dur, proto, dir, sip, sport, dip, dport, state, toTPkts, toTBytes\rangle$。

定义 2（时间片，timeSlice） 表示对网络流进行粗粒度统计特征提取的时间间隔长度。例如，图 1 中的时间片为 δ。

已有研究表明，僵尸网络存在时序性局部行为，即在一定时间片内其攻击行为不变或相似[13]。根据 Garcia 等[14]的研究，在粗粒度统计特征提取阶段，统计一定时间片内（即 $t-\delta\sim t$）的网络流序列（即 $nfList=[nf_0, nf_1, \cdots, nf_k]$，其中，$t-\delta\leqslant nf_j[startTime]\leqslant t, j\in\{0,1,\cdots,k\}$）中与待检测网络流 nf_T（$nf_T\in\{nf_0, nf_1, \cdots, nf_k\}$）具有同属性或属性对的网络流数量。具体包括：

1）如式（1）所示，仅统计与请求发起端的属性相同的网络流数量，请求发起端的属性包括源 IP、源端口、源 IP 和源端口。

$$num_{(att,any)}^{(t-\delta,t),nf_T}=\sum_{j=0}^{k} i$$
$$i=\begin{cases}1, & \text{if } nf_j[att]==nf_T[att]\\ 0, & \text{else}\end{cases} \tag{1}$$
$$att\in\{sip, sport, sip:sport\}$$

2）如式（2）所示，统计与请求发起端和接收端的属性均相同的网络流数量，请求发起端的属性包括源 IP、源端口、源 IP 和源端口，请求接收端的属性包括目标 IP、目标端口、目标 IP 和目标端口。

$$num_{(att_1,att_2)}^{(t-\delta,t),nf_T}=\sum_{j=0}^{k} i$$
$$i=\begin{cases}1, & \text{if } nf_j[att_1]==nf_T[att_1],\\ & \quad nf_j[att_2]==nf_T[att_2]\\ 0, & \text{else}\end{cases} \tag{2}$$
$$att_1\in\{sip, sport, sip:sport\}$$
$$att_2\in\{dip, dip:dport, dport\}$$

3）如式（3）所示，仅统计与请求接收端的属性均相同的网络流数量，请求接收端的属性包括目标 IP、目标端口、目标 IP 和目标端口。

$$num_{(any,att)}^{(t-\delta,t),nf_T}=\sum_{j=0}^{k} i$$
$$i=\begin{cases}1, & \text{if } nf_j[att]==nf_T[att]\\ 0, & \text{else}\end{cases} \tag{3}$$
$$att\in\{dip, dport, dip:dport\}$$

基于上述分析，本文设计了算法 1 所示的基于时间片的网络流粗粒度统计特征提取算法（timeSliceCoarStatFeatExtAlg），算法采用空间换时间的方式，将时间片内的统计特征值均存储于 featDict 内。在算法 1 中，第 1－6 行定义函数 featDictUpdateFunc 实现了 featDict 的值的更新。第 7－20 行定义主体函数 timeSliceCoarStatFeatExtAlg 实现了时间片内网络流的粗粒度统计。在主体函数中，第 8－10 行进行变量初始化；第 11 行获取 st 至 $st+\delta$ 内的已预处理完成网络流序列 $nfList$；第 12－19 行对 $nfList$ 内的网络流进行逐一扫描统计，其中第 15－16 行统计请求发起端属性值相同的网络流数量，第 17－18 行统计请求接收端属性值相同的网络流数量，第 19 行结合第 15－18 行的统计过程一起实现请求发起端和接收端属性值均相同的网络流数量的统计。该算法的时间复杂度为 O(|$nfList$|)。

算法 1 基于时间片的网络流粗粒度统计特征提取算法（timeSliceCoarStatFeatExtAlg）

输入：分析起始时间 st，时间片长度 δ

输出：粗粒度统计特征 featDcit

```
1. featDictUpdateFunc(key,featDict,value=1):
2.     if key in featDict:
3.         featDict[key]=featDict[key]+value
4.     else:
5.         featDict[key]=values
6.     return featDict
7. timeSliceCoarStatFeatExtAlg(st,δ):
8.     featDict={}
9.     srcAttrList=["sip","sport","sip:sport"]
10.    dstAttrList=["dip","dport","dip:dport"]
11.    nfList=getNfList(st,st+δ)/*获取时间片内已预处理完成的网络流*/
12.    for nf in nfList:
13.        nf["sip:sport"]=nf["sip"]+str(nf["sport"])
14.        nf["dip:sport"]=nf["dip"]+str(nf["dport"])
15.        for att in srcAttList:
16.            featDictUpdateFunc(nf[att],featDict) /*统计请求发起端同属性网络流数量*/
17.            for att1 in dstAttList:
18.                featDictUpdateFunc(nf[att1],featDict) /*统计请求接收端同属性网络流数量*/
19.                featDictUpdateFunc(nf[att]+"-"+nf[att1],featDict) /*统计请求发起端、接收端属性均相同的网络流数量*/
20.    return featDict
```

3.2 面向源 IP 的全局网络流细粒度画像算法（globalFineFeatProAlg）

基于时间片的网络流粗粒度统计特征提取算法可提取僵尸网络流量在一定时间片内呈现的统计特征，但是难以呈现僵尸主机发出网络流量全生命周期内的累计性统计特征，如

图 2 所示，僵尸网络恶意代码 Rbot 和 Neris 感染的主机访问目标端口 53 的累计性统计结果与良性网络主机访问目标端口 53 的统计结果具有明显的差异。鉴于此，本文引入面向源 IP 的全局网络流的细粒度画像。

统计全部历史网络流序列（$allNfList=[nf_0,nf_1,\cdots,nf_n]$，其中 $nf_0[startTime]<nf_1[startTime]<\cdots<nf_n[startTime]$）中与待检测网络流 nf_Γ（$nf_\Gamma=nf_n$）源 IP 和目标端口相同的网络流数量以及相应的数据包数量、发送的字节平均数、通信的平均时间长度及使用的协议分布情况。

1）如式（4）所示，在全部历史网络流中，统计与网络流 nf_Γ 具有相同源 IP $vsip$（$vsip=nf_\Gamma[sip]$，下同，不再赘述）和目标端口 $vdport$（$vdport=nf_\Gamma[dport]$，下同，不再赘述）的网络流数量。

$$fineNum^{vsip,vdport}_{dport,(sip,dport)}=\sum_{j=0}^{n}i$$
$$i=\begin{cases}1, & \text{if } nf_j[sip]==vsip,\\ & \quad nf_j[dport]==vdport\\ 0, & \text{else}\end{cases}\tag{4}$$

2）如式（5）所示，在全部历史网络流中，统计与网络流 nf_Γ 具有相同源 IP $vsip$ 和目标端口 $vdport$ 的网络流中发送的数据包的总数量。

$$fineNum^{vsip,vdport}_{toTPkts,(sip,dport)}=\sum_{j=0}^{n}i$$
$$i=\begin{cases}nf_j[toTPkts], & \text{if } nf_j[sip]==vsip,\\ & \quad nf_j[dport]==vdport\\ 0, & \text{else}\end{cases}\tag{5}$$

3）如式（6）所示，在全部历史网络流中，计算与网络流 nf_Γ 具有相同源 IP $vsip$ 和目标端口 $vdport$ 的网络流中发送数据包的平均字节数。

$$fineNum^{vsip,vdport}_{ave_toTBytes,(sip,dport)}=\frac{\sum_{j=0}^{n}i}{fineNum^{vsip,vdport}_{toTPkts,(sip,dport)}}$$
$$i=\begin{cases}nf_j[toTBytes], & \text{if } nf_j[sip]==vsip,\\ & \quad nf_j[dport]==vdport\\ 0, & \text{else}\end{cases}\tag{6}$$

4）如式（7）所示，在全部历史网络流中，计算与网络流 nf_Γ 具有相同源 IP $vsip$ 和目标端口 $vdport$ 的网络流中通信的平均时长。

$$fineNum^{vsip,vdport}_{ave_dur,(sip,dport)}=\frac{\sum_{j=0}^{n}i}{fineNum^{vsip,vdport}_{toTPkts,(sip,dport)}}$$
$$i=\begin{cases}nf_j[dur], & \text{if } nf_j[sip]==vsip,\\ & \quad nf_j[dport]==vdport\\ 0, & \text{else}\end{cases}\tag{7}$$

5）如式（8）所示，在全部历史网络流中，分析与网络流 nf_Γ 具有相同源 IP $vsip$ 和通信协议 $vproto$（$vproto=nf_\Gamma[proto]$，下同，不再赘述）的网络流使用的通信协议分布情况。

$$fineNum^{vsip,vproto}_{proto,(sip,any)}=\sum_{j=0}^{n}i$$
$$i=\begin{cases}1, & \text{if } nf_j[sip]==vsip,\\ & \quad nf_j[proto]==vproto\\ 0, & \text{else}\end{cases}\tag{8}$$

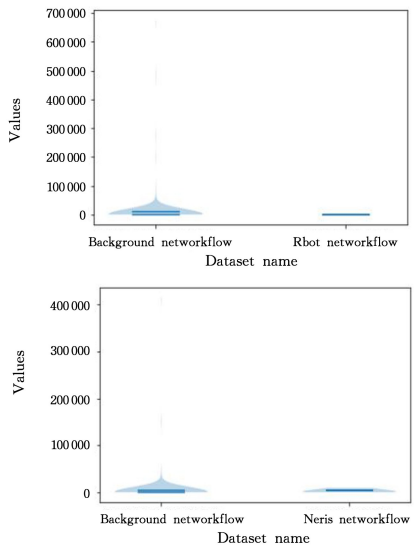


图 2 Rbot 和 Neris 僵尸网络流与良性网络流访问 53 端口的累积性统计结果

Fig. 2 Cumulative statistics of Rbot and Neris botnet network flow and benign network flow accessing port 53

然后，对待分析网络流的源 IP $vsip$ 在历史网络流中访问目标端口、各个目标端口的通信平均时长及发送数据的平均字节数、涉及的通信协议分布情况进行细粒度画像。

1）如式（9）所示，当 $fea=dport$ 时，统计与网络流 nf_Γ 的源 IP $vsip$ 相同的历史网络流中访问目标端口的情况，形成源 IP 在访问目标端口维度上的画像。

$$profile^{vsip}_{fea,(sip,dport)}=\{vport_0:fineNum^{vsip,vport_0}_{fea,(sip,dport)},vport_1:fineNum^{vsip,vport_1}_{fea,(sip,dport)},\cdots,vport_r:fineNum^{vsip,vport_r}_{fea,(sip,dport)}\}$$

其中：

$$vport_0,vport_1,\cdots,vport_r\in\{nf_i[dport]\mid nf_i[sip]==vsip \text{ and } 0\leqslant i\leqslant n\}$$
$$fineNum^{vsip,vport_q}_{fea,(sip,dport)}=\max(\{fineNum^{vsip,vport_q}_{fea,(sip,dport)}\mid nf_j[sip]==vsip \text{ and } nf_j[dport]==port_q \text{ and } 0\leqslant q\leqslant r \text{ and } 0\leqslant j\leqslant n\})$$
$$fea\in\{dport,toTPkts,ave_toTBytes,ave_dur\}\tag{9}$$

2）如式（9）所示，当 $fea=toTPkts$ 时，分析与网络流 nf_Γ 的源 IP $vsip$ 相同的历史网络流中访问各个目标端口过程中发送数据包的数量，形成源 IP 访问不同目标端口时在数据包发送数量维度上的画像。

3）如式（9）所示，当 $fea=ave_toTBytes$ 时，分析与网络流 nf_Γ 的源 IP $vsip$ 相同的历史网络流中访问各个目标端口过程中发送数据的平均字节数，形成源 IP 访问不同目标端口时在发送的数据平均字节数维度上的画像。

4）如式（9）所示，当 $fea=ave_dur$ 时，分析与网络流 nf_Γ 的源 IP $vsip$ 相同的历史网络流中访问各个目标端口过程中通信的平均时长，形成源 IP 访问不同目标端口时在通信平均时长维度上的画像。

5）如式（10）所示，统计与网络流 nf_Γ 的源 IP$vsip$ 相同的历史网络流中通信协议的应用情况，形成源 IP 在通信协议维度上的画像。

$$profile^{vsip}_{proto,(sip,any)}=\{vproto_0:fineNum^{vsip,vproto_0}_{proto,(sip,any)},vpro-$$

$to_1:fineNum_{proto,(sip,any)}^{vsip,vproto_1},\cdots,vproto_p:fineNum_{proto,(sip,any)}^{vsip,vproto_p}\}$

其中：

$$vproto_0,vproto_1,\cdots,vproto_p\in\{nf_i[proto]\mid nf_i[sip]==nf[sip]\text{and }0\leqslant i\leqslant n\}$$

$$fineNum_{proto,(sip,any)}^{vsip,vproto_q}=\max(\{fineNum_{proto,(sip,any)}^{vsip,vproto_q}\mid nf_i[sip]==vsip$$

$$nf_i[proto]==vproto_q\text{ and }0\leqslant q\leqslant n\text{ and }0\leqslant i\leqslant n\}) \tag{10}$$

基于上述分析，本文设计了算法 2 所示的全局网络流细粒度画像算法(globalFineFeatProAlg)，同算法 1 类似，算法采用空间换时间的方式，将历史网络流相关的特征值均存储于 globalFeatDict 内。在算法 2 中，第 2－3 行进行变量定义和初始化；第 4－8 行对首次出现的源 IP 在 globalFeatDict 内进行初始化赋值；第 9－13 行更新 globalFeatDict 内与当前网络流 nf 的源 IP、目标端口相同的网络流数量、通信总时长、发送的数据包总数、发送的数据总字节数；第 14－15 行分析当前网络流 nf 的源 IP、目标端口相同的历史网络流的平均通信时间长度和发送的平均字节数；第 16－17 行提取当前网络流的源 IP 相关细粒度画像。该算法的时间复杂度为 O(1)。

算法 2 面向源 IP 的全局网络流细粒度特征画像算法(globalFineFeatProAlg)

输入：待分析网络流 nf，历史网络流统计结果 globalFeatDict

输出：待分析网络流相关的细粒度统计特征 nfFineFeatDict

```
1. globalFineFeatProAlg(nf,globalFeatDict):
2.    dport=nf["dport"],proto=nf["proto"];
3.    nfFineFeatDict={};
4.    if sip∉globalFeatDict:
5.       for dim∈{"dport","proto"}:
6.          globalFeatDict[sip][dim][dport]=1
7.       for dim∈{"dur","totpkts","totbytes"}:
8.          globalFeatDict[sip][dim][dport]=nf[dim];
9.    else:
10.      for dim∈{"dport","proto"}:
11.         globalFeatDict[sip][dim][dport] +=1
12.      for dim∈{"dur","totpkts","totbytes"}:
13.         globalFeatDict[sip][dim][dport] +=nf[dim]
14.         globalFeatDict[sip]["ave_dur"][dport]=globalFeatDict
            [sip]["dur"][dport]/ globalFeatDict[sip]["totpkts"]
            [dport].
15.         globalFeatDict[sip]["ave_totbytes"][dport]=globalFeat-
            Dict[sip]["totbytes"][dport]/globalFeatDict[sip]["totpk-
            ts"][dport].
16.      for dim∈{"dport","ave_dur","ave_totbytes","proto"}:
17.         nfFineFeatDict[sip][dim]=globalFeatDict[sip][dim];
18.   return nfFineFeatDict.
```

4 基于多头注意力机制的 LSTM 僵尸网络流量检测模型(M-AttLSTMBotModel)

4.1 网络流特征向量生成

为了能够充分利用机器学习或深度学习在挖掘特征之间复杂关系、捕捉数据内在结构等方面的优势，需将上文已提取的各类特征转换为机器学习或深度学习可接受的数学向量。本文将提取的不同粒度的不同维度的特征转换为数学向量后在水平方向进行向量堆叠，以形成最终的数学向量。

对于提取的基于时间片的网络流粗粒度特征，依据特定的顺序构建数学向量，如式(11)所示，按照请求发起端统计特征、两端统计特征和请求接收端统计特征进行排列。

$$\boldsymbol{coarVector}^{(t-\delta,t),nf_T}=[num_{(sip,any)}^{(t-\delta,t),nf_T},\cdots,num_{(att1,att2)}^{(t-\delta,t),nf_T},\cdots,num_{(any,att2)}^{(t-\delta,t),nf_T}] \tag{11}$$

其中，$att_1\in\{sip,sport,sip:sport,any\}$，$att_2\in\{any,dip,dip:dport,dport\}$。

对提取的面向源 IP 的全局网络流的细粒度画像的各个维度分别采用字典映射的方式构建数学向量，如式(12)、式(13)所示。

$$\boldsymbol{fineVector}_{fea,(sip,dport)}^{vsip}=[fineNum_{fea,(sip,dport)}^{vsip,vport_0},fineNum_{fea,(sip,dport)}^{vsip,vport_1},\cdots,fineNum_{fea,(sip,dport)}^{vsip,vport_r}] \tag{12}$$

其中，$fea\in\{dport,toTPkts,ave_toTBytes,ave_dur\}$，$|\boldsymbol{fine\ Vector}_{fea,(sip,dport)}^{vsip}|=r$。

$$\boldsymbol{fineVector}_{proto,(sip,any)}^{vsip}=[fineNum_{proto,(sip,any)}^{vsip,vproto_0},fineNum_{proto,(sip,any)}^{vsip,vproto_1},\cdots,fineNum_{proto,(sip,any)}^{vsip,vproto_p}] \tag{13}$$

其中，$|\boldsymbol{fineVector}_{proto,(sip,any)}^{vsip}|=p$。

最后，将式(11)和式(12)生成的数学向量采用水平堆叠的方式生成网络流特征向量 $\boldsymbol{featureVector}$，其中，$|\boldsymbol{featureVector}|=|\boldsymbol{coarVector}^{(t-\delta,t),nf_T}|+4*r+p$。

4.2 长短期记忆网络 LSTM

僵尸网络产生的网络流具有一定的时间序列特征，而循环神经网络(Recurrent Neural Network，RNN)能够捕捉时间序列中的长期和短期依赖关系，提升时间序列数据处理任务的准确率，但其在长序列数据处理任务上存在梯度消失和梯度保障等问题。Hochreiter 等[15]提出的 LSTM 有效突破了 RNN 存在的局限性，可用于分析具有长期依赖关系或时间序列特征的数据。基于此，本文引入 LSTM 挖掘僵尸网络产生的网络流、良性网络流的粗粒度统计特征和细粒度画像在时间序列上的差异性，进而基于这种差异性识别僵尸网络产生的网络流。

如图 3 所示，LSTM 由输入门 $\boldsymbol{I}_t$、遗忘门 $\boldsymbol{F}_t$、输出门 $\boldsymbol{O}_t$、候选记忆单元 $\tilde{\boldsymbol{C}}_t$、记忆单元 $\boldsymbol{C}_t$ 和隐藏状态 $\boldsymbol{H}_t$ 构成。

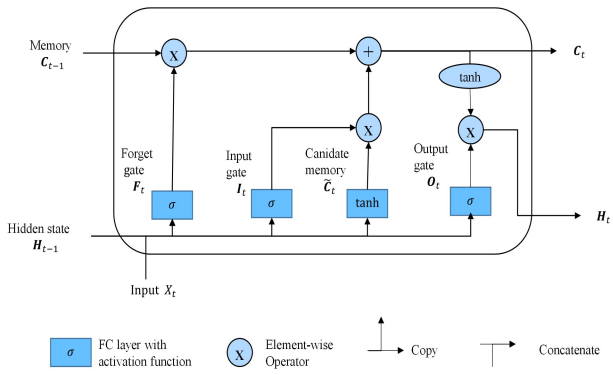


图 3 LSTM 的单元结构[16]

Fig. 3 Structure of the LSTM cell[16]

其中，$\boldsymbol{I}_t$决定$\boldsymbol{X}_t$的哪些信息会更新至$\boldsymbol{C}_t$，$\boldsymbol{F}_t$决定前一个时刻的记忆单元的输出$\boldsymbol{C}_{t-1}$中信息被保留的程度，$\boldsymbol{O}_t$负责控制当前时刻$\boldsymbol{H}_t$中的哪些信息输出到下一层或作为模型的最终输出，$\widetilde{\boldsymbol{C}}_t$是基于对当前时刻输入$\boldsymbol{X}_t$的理解形成的抽象表示值，$\boldsymbol{C}_t$主要是基于$\boldsymbol{F}_t$和$\boldsymbol{I}_t$分别从前一时刻的记忆单元输出$\boldsymbol{C}_{t-1}$、候选记忆单元的输出$\widetilde{\boldsymbol{C}}_t$中筛选留存的信息而形成的值，$\boldsymbol{H}_t$则是基于模型对当前时刻输入$\boldsymbol{X}_t$的理解而形成的值。LSTM的计算过程如式(14)－式(17)所示：

$$\boldsymbol{I}_t=\sigma(\boldsymbol{W}_i\boldsymbol{X}_t+\boldsymbol{U}_i\boldsymbol{H}_{t-1}+\boldsymbol{b}_i) \tag{14}$$

$$\boldsymbol{F}_t=\sigma(\boldsymbol{W}_f\boldsymbol{X}_t+\boldsymbol{U}_f\boldsymbol{H}_{t-1}+\boldsymbol{b}_f) \tag{15}$$

$$\boldsymbol{O}_t=\sigma(\boldsymbol{W}_o\boldsymbol{X}_t+\boldsymbol{U}_o\boldsymbol{H}_{t-1}+\boldsymbol{b}_o) \tag{16}$$

$$\widetilde{\boldsymbol{C}}_t=\tanh(\boldsymbol{W}_c\boldsymbol{X}_t+\boldsymbol{U}_c\boldsymbol{H}_{t-1}+\boldsymbol{b}_c) \tag{17}$$

$$\boldsymbol{C}_t=\boldsymbol{F}_t*\boldsymbol{C}_{t-1}+\boldsymbol{I}_t*\widetilde{\boldsymbol{C}}_t \tag{18}$$

$$\boldsymbol{H}_t=\boldsymbol{O}_t*\tanh(\boldsymbol{C}_t) \tag{19}$$

其中，$\boldsymbol{X}_t\in R^{n\times|featureVector|}$；$\boldsymbol{H}_{t-1},\boldsymbol{I}_t,F_t,\boldsymbol{O}_t,\widetilde{\boldsymbol{C}}_t,\boldsymbol{C}_{t-1},\boldsymbol{C}_t,\boldsymbol{H}_t\in R^{n\times h}$；$\boldsymbol{W}_i,\boldsymbol{W}_f,\boldsymbol{W}_o,\boldsymbol{W}_c\in R^{|featureVector|\times h}$；$\boldsymbol{U}_i,\boldsymbol{U}_f,\boldsymbol{U}_o,\boldsymbol{U}_c\in R^{h\times h}$；$\boldsymbol{b}_i,\boldsymbol{b}_f,\boldsymbol{b}_o,\boldsymbol{b}_c\in R^{1\times h}$。

4.3 基于多头注意力机制的 LSTM

相同类型僵尸网络在传播时的不同杀伤链阶段的行为特征具有相似性[10]，挖掘分析不同杀伤链阶段的行为特征及之间的依赖关系，可以提升僵尸网络识别分析的准确率，这也需要模型能够提取输入网络流序列中某个位置网络流的特征及不同位置网络流之间的依赖关系。传统的LSTM模型对输入的一组网络流进行无差异化的分析和挖掘，难以对不同类型的网络流采取不同的分析策略，这在一定程度上限制了模型对长时间段间网络流间依赖关系的分析和僵尸网络产生网络流中重要特征的挖掘。Vaswani等[17]提出的多头注意力机制可以挖掘输入序列数据中不同位置的关键特征信息，对解决长距离依赖性和全局信息的捕捉具有明显优势。基于此，本文在LSTM模型的基础上增加多头注意力机制以实现动态关注输入网络流序列中的不同位置，更有效地挖掘网络流间存在的依赖关系，提取网络流的上下文信息，进而基于这些信息发现僵尸网络流和良性网络流在统计特征方面存在的差异性。

如图4所示，网络流经过LSTM模型层后输入δ层并行的注意力计算模型$\boldsymbol{head}_t^l$(见式(20))中，每层的注意力计算模型的运算过程如式(21)－式(24)所示：

$$\begin{aligned}\boldsymbol{Y}_t&=MultiHead(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V})\\&=Concat(\boldsymbol{head}_t^1,\boldsymbol{head}_t^2,\boldsymbol{head}_t^3,\boldsymbol{L},\boldsymbol{head}_t^\delta)\boldsymbol{W}^O\end{aligned} \tag{20}$$

$$\boldsymbol{head}_t^l=\mathrm{softmax}\left(\boldsymbol{Q}^l\frac{(\boldsymbol{K}^l)^{\mathrm{T}}}{\sqrt{d_k}}\right)\boldsymbol{V}^l,l=1,2,\cdots,\delta \tag{21}$$

$$\boldsymbol{Q}^l=\boldsymbol{H}\cdot\boldsymbol{W}_Q^l \tag{22}$$

$$\boldsymbol{K}^l=\boldsymbol{H}\cdot\boldsymbol{W}_K^l \tag{23}$$

$$\boldsymbol{V}^l=\boldsymbol{H}\cdot\boldsymbol{W}_V^l \tag{24}$$

$$\boldsymbol{H}=(\boldsymbol{H}_1,\boldsymbol{H}_2,\cdots,\boldsymbol{H}_t) \tag{25}$$

其中，$\boldsymbol{Q}^l$表示当前网络流在注意力计算中的希望提取关注度，$\boldsymbol{K}^l$表示其他位置网络流在注意力计算中的关联性，$\boldsymbol{V}^l$表示当前网络流通过注意力计算获取的实际关注度。$\boldsymbol{W}_Q^l,\boldsymbol{W}_K^l,\boldsymbol{W}_V^l\in R^{h\times d_k}$，$\boldsymbol{W}^O\in R^{\delta d_k\times h}$，$d_k=h*64$。

综上所述，基于多头注意力机制的LSTM的计算过程可描述为式(26)－式(27)，其中各参数定义同式(14)－式(25)。通过综合利用LSTM的序列建模能力和多头注意力的全局建模能力，以提升模型的表现力。

$$\begin{aligned}\boldsymbol{H}_t,\boldsymbol{C}_t&=LSTM(\boldsymbol{X}_t,\boldsymbol{H}_{t-1},\boldsymbol{C}_{t-1})\\&=\sigma(\boldsymbol{W}_o\boldsymbol{X}_t+\boldsymbol{U}_o\boldsymbol{H}_{t-1}+\boldsymbol{b}_o)*\tanh(\sigma(\boldsymbol{W}_f\boldsymbol{X}_t+\\&\quad\boldsymbol{U}_f\boldsymbol{H}_{t-1}+\boldsymbol{b}_f)*C_{t-1}+\sigma(\boldsymbol{W}_i\boldsymbol{X}_t+\boldsymbol{U}_i\boldsymbol{H}_{t-1}+\boldsymbol{b}_i)*\\&\quad\tanh(\boldsymbol{W}_c\boldsymbol{X}_t+\boldsymbol{U}_c\boldsymbol{H}_{t-1}+\boldsymbol{b}_c))\end{aligned} \tag{26}$$

$$\begin{aligned}\boldsymbol{Y}_t&=MultiHead(\boldsymbol{HW}_Q,\boldsymbol{HW}_k,\boldsymbol{HW}_v)\\&=Concat(software\left(\boldsymbol{HW}_Q^1\frac{(\boldsymbol{HW}_K^1)^{\mathrm{T}}}{\sqrt{d_k}}\right)\boldsymbol{HW}_V^1,\\&\quad software\left(\boldsymbol{HW}_Q^2\frac{(\boldsymbol{HW}_K^2)^{\mathrm{T}}}{\sqrt{d_k}}\right)\boldsymbol{HW}_V^2,\\&\quad software\left(\boldsymbol{HW}_Q^3\frac{(\boldsymbol{HW}_K^3)^{\mathrm{T}}}{\sqrt{d_k}}\right)\boldsymbol{HW}_V^3,\cdots,\\&\quad software\left(\boldsymbol{HW}_Q^\delta\frac{(\boldsymbol{HW}_K^\delta)^{\mathrm{T}}}{\sqrt{d_k}}\right)\boldsymbol{HW}_V^\delta\end{aligned} \tag{27}$$

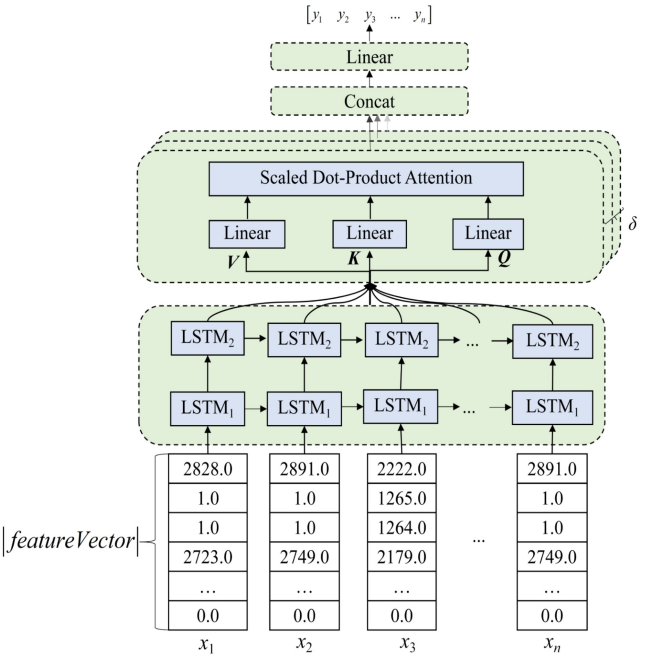


图4 M-AttLSTMBotModel结构

Fig. 4 Structure of M-AttLSTMBotModel

5 实验评估

5.1 实验数据集及环境

本文使用布拉格捷克理工大学创建的CTU-13数据集[18]和加拿大信息安全中心的ISCX僵尸网络数据集[19]验证本文方法的可行性和有效性。

CTU-13数据集包含13个不同场景的网络流量，每个场景的流量混杂着正常网络流量、背景网络流量和特定僵尸网络恶意代码产生的流量，本文统称正常网络流量和背景网络

流量称为良性网络流量。从网络流量中提取网络流的分布情况，如表 1 所列。

表 1 CTU-13 数据集中不同场景不同网络流的分布情况

Table 1 Distribution of different network flows in various scenarios in the CTU-13 dataset

Scen.	Total Flows	Botnet Flows	Normal Flows	Background Flows	Bot Name
1	2824636	40961(1.45%)	30387(1.08%)	2753288(97.47%)	Neris
2	1808122	20941(1.16%)	9120(0.5%)	1778061(98.34%)	Neris
3	4710638	26822(0.57%)	116887(2.48%)	4566929(96.95%)	Rbot
4	1121076	2580(0.23%)	25268(2.25%)	1093228(97.52%)	Rbot
5	129832	901(0.7%)	4679(3.6%)	124252(95.7%)	Virut
6	558919	4630(0.83%)	7494(1.34%)	546795(97.83%)	Menti
7	114077	63(0.06%)	1677(1.47%)	112337(98.47%)	Sogou
8	2954230	6127(0.2%)	72822(2.47%)	2875281(97.33%)	Murlo
9	2087508	184987(8.86%)	29967(1.44%)	1872554(89.7%)	Neris
10	1309791	106352(8.12%)	15847(1.21%)	1187592(90.67%)	Rbot
11	107251	8164(7.6%)	2718(2.5%)	96369(89.9%)	Rbot
12	325471	2168(0.67%)	7628(2.34%)	315675(96.99%)	NSIS. ay
13	1925149	40003(2.08%)	31939(1.66%)	1853207(96.26%)	Virut

为了保障两个数据集具有相同的原始特征，利用 CICFlowmeter 工具对 ISCX 僵尸网络数据集的 pcap 包进行特征提取并按照 CTU-13 的特征进行数据预处理，包括字段名称映射、数据格式转换（如将 CICFlowmeter 提取的协议编码转换为协议名称）、无效数据删除（如删除 CICFlowmeter 提取转换特征时的异常值）、字段值标尺统一转换（如将数据流通信持续时长 Dur 统一转换为秒）等，并对数据集进行部分裁剪，以保障标记数据集的准确性。裁剪结果如表 2 所列。

表 2 ISCX 数据集中不同场景下不同网络流的分布情况

Table 2 Distribution of different network flows in various scenarios in the ISCX dataset

Scen.	Type	Num.
1	Neris	8322(15.57%)
2	RBot	13998(26.2%)
3	Virut	1302(2.44%)
4	Normal Flows	29811(55.79%)

本文模型训练和测试的软硬件配置如下：2 颗 10 核 Intel Xeon Silver 4210 @ 2.20 GHZ CPU、64 GB 内存、Nvidia Quadro RTX 4000 显卡、Centos 7.5 操作系统、PyTorch 神经网络框架。

5.2 实验结果评估及分析

本文将僵尸网络流量的检测问题转换为二分类问题，即输入从网络流量中提取的统计特征，输出是否为僵尸网络流量。使用准确率、精确率、召回率和 F1 值对模型进行评价，计算式如式(25)－式(28)所示。

准确率 Accuracy 是正确判定为僵尸网络流量的样本占所有样本的比值。

$$Accuracy=\frac{TP+TN}{TP+TN+FP+FN} \tag{25}$$

其中，TP，TN，FP，FN 为衡量检测模型的通用指标。TP（True Positives）指检测模型将真实的僵尸网络流量正确判定为僵尸网络流量的样本数量；TN（True Negatives）指检测模型将良性网络流量正确判定为良性网络流量的样本数量；FP（False Positives）指检测模型将良性网络流量错误判定为僵尸网络流量的样本数量；FN（False Negatives）指检测模型将真实的僵尸网络流量错误判定为良性网络流量的样本数量。

精确率 Precision 是正确判定为僵尸网络流量的样本占所有判定为僵尸网络流量样本的比值。

$$Precision=\frac{TP}{TP+FP} \tag{26}$$

召回率 Recall 是正确判定为僵尸网络流量的样本占所有僵尸网络流量样本的比值。

$$Recall=\frac{TP}{TP+FN} \tag{27}$$

F1 分数能够综合判断僵尸网络流量检测模型的好坏。

$$F1=\frac{2TP}{2TP+FP+FN} \tag{28}$$

漏检率（False Negative Rate，FNR）是误判为良性网络流量的样本占所有僵尸网络流量样本的比值。

$$FNR=\frac{FN}{FN+TP} \tag{29}$$

误报率（False Positive Rate，FPR）是误判为僵尸网络流量的样本占所有良性网络流量样本的比值。

$$FPR=\frac{FP}{FP+TN} \tag{30}$$

如图 5 所示，在 CTU-13 数据集上，模型在第 10 轮训练结束后，训练损失和验证损失趋于 0，准确率、精确率、召回率及 F1 值趋于 1（见图 6 和表 3）。精确率趋于 1，意味着 $FP=0$，那么误报率趋于 0，同理，召回率趋于 1，漏检率趋于 0。在 ISCX 僵尸网络数据集上，模型的准确率、精确率、召回率及 F1 值均大于 0.99。如表 3 所列，本文模型在两个数据集上的检测结果均优于文献[20]和文献[21]提出的方法。

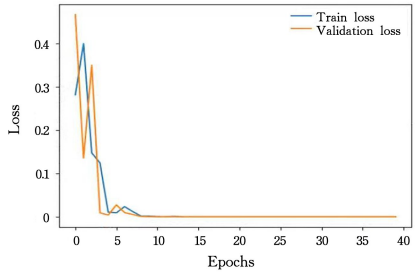


图 5 M-AttLSTMBotModel 训练损失和验证损失

Fig. 5 Training loss and validation loss of M-AttLSTMBotModel

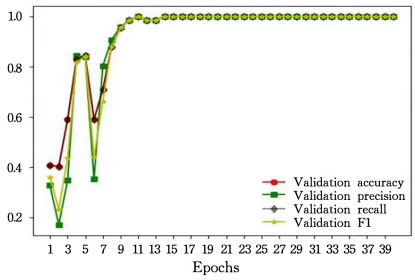


图 6 本文方法的验证结果

Fig. 6 Validation results of the proposed method

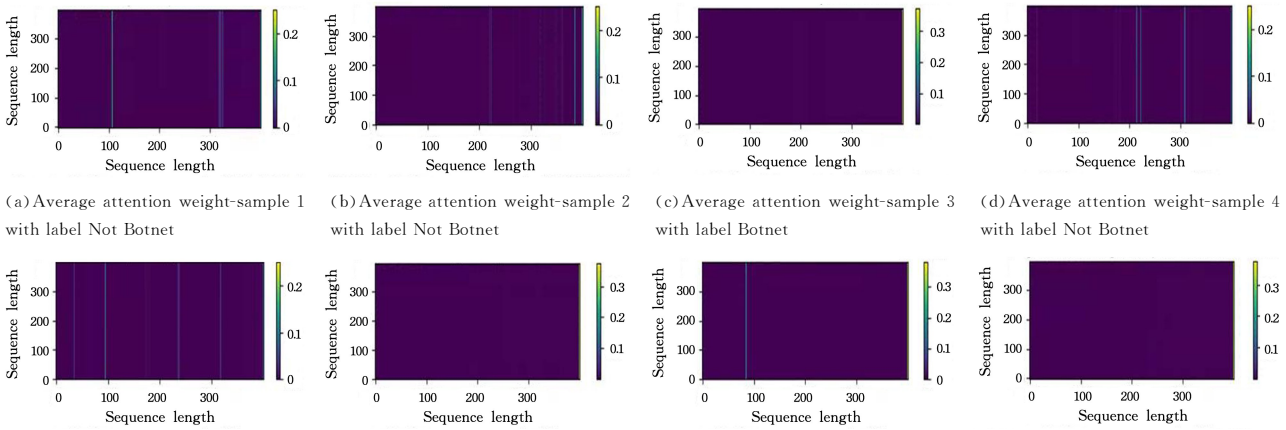
表 3 模型检测分析结果

Table 3 Analysis result of model detection

Dataset	Method	*Accuracy*	*Precision*	*Recall*	*F*1
CTU-13	BiLSTM-GAN[20]	0.8551	0.9155	0.8238	0.8672
	1D-CNN[21]	0.9330	0.9750	0.9210	0.9950
	LSTM[21]	0.9764	0.9814	0.9698	0.9990
	Ours	1.0000	1.0000	1.0000	1.0000
ISCX	BiLSTM[20]	0.8345	0.9527	0.7577	0.8441
	Ours	0.9925	0.9927	0.9925	0.9925

为了分析多头注意力机制在本文模型中的作用，本文对测试样本中测试时对多头注意力机制输出的平均权重进行了可视化展示。图 7 是分析 Neris 僵尸样本的网络流量和良性网络流量的测试样本时多头注意力机制输出的平均权重可视化图。可以观察出，模型分析良性网络流量时针对不同的特征采用不同的注意力，即多头注意力机制的平均权重值不同（图中有 2～3 条不同色系的竖线出现）；而模型在分析僵尸网络流量时对不同的特征采用了近似相同的注意力，即多头注意力机制的平均权重值相同。上述观察结果表明，M-Att-LSTMBotModel能够较好地挖掘僵尸网络流量与良性网络流量在本文设计的特征维度上存在的差异性，进而基于这些差异性识别僵尸网络流量。

为了对比局部粗粒度特征和全局细粒度画像对检测能力的影响，本文基于 CTU-13 和 ISCX 僵尸网络数据集进行了消融实验，分别依靠局部粗粒度特征、全局细粒度画像和本文全部特征进行检测，结果如表 4 所列。通过消融实验可以发现，全局细粒度画像特征对检测能力的重要性大于局部粗粒度特征。



(a) Average attention weight-sample 1 with label Not Botnet

(b) Average attention weight-sample 2 with label Not Botnet

(c) Average attention weight-sample 3 with label Botnet

(d) Average attention weight-sample 4 with label Not Botnet

(e) Average attention weight-sample 5 with label Not Botnet

(f) Average attention weight-sample 6 with label Botnet

(g) Average attention weight-sample 7 with label Botnet

(h) Average attention weight-sample 8 with label Botnet

图 7 基于部分测试样本 M-AttLSTMBotModel 输出的多头注意力平均权重图

Fig. 7 Average attention weight maps of the multi-head attention ouput of M-AttLSTMBotModel based on part test samples

表 4 M-AttLSTMBotModel 的消融实验结果

Table 4 Ablation study results of the M-AttLSTMBotModel

数据集	输入特征	*Accuracy*	*Precision*	*Recall*	*F*1
CTU-13	仅有局部粗粒度特征	0.558	0.311	0.558	0.400
	仅有全局细粒度画像	0.961	0.953	0.961	0.957
	本文全部特征	1.000	1.000	1.000	1.000
ISCX	仅有局部粗粒度特征	0.560	0.318	0.560	0.405
	仅有全局细粒度画像	0.955	0.959	0.955	0.955
	本文全部特征	0.993	0.993	0.993	0.993

结束语 本文针对当前僵尸网络流量检测方法易被绕过及误报率高的局限性，提出了一种基于多粒度统计特征的僵尸网络流量智能检测方法。首先提取时间片内的局部网络流粗粒度统计特征，构建面向源 IP 的全局网络流细粒度画像特征；然后利用具有多头注意力机制的长短期记忆网络模型充分分析网络流间的依赖关系及上下文信息获取僵尸网络流和良性网络流存在的差异性；最后基于这些差异实现僵尸网络流量的检测。本文基于 CTU-13 和 ISCX 数据集验证了所提方法的可行性及有效性。

后续研究工作包括引入自动特征工程从原始网络流量的元数据中提取特征以突破当前人工特征工程的局限性，并尝试利用自然语言大模型提升对未知僵尸网络流量的检测效果。

参 考 文 献

[1] Office of Public Affairs. Qakbot Malware Disrupted in International Cyber Takedown[EB/OL]. https://www.justice.gov/opa/pr/qakbot-malware-disrupted-international-cyber-takedown.

[2] MAHMOUD M, NIR M, MATRAWY A. A survey on botnet architectures, detection and defences[J]. International Journal of Network Security, 2015, 17(3): 264-281.

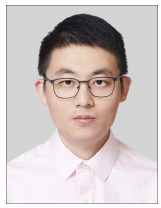
[3] BEER F, BÜHLER U. Feature selection for flow-based intrusion detection using rough set theory[C]// 2017 IEEE 14th International Conference on Networking, Sensing and Control(ICNSC). IEEE, 2017: 617-624.

[4] CLAISE B. Cisco systems netflow services export version 9[R].

2004.
[5] KONDO S,SATO N. Botnet traffic detection techniques by C&C session classification using SVM[C]// International Workshop on Security. Berlin:Springer,2007:91-104.
[6] AYO F E,AWOTUNDE J B,FOLORUNSO S O,et al. A genomic rule-basedKNN model for fast flux botnet detection[J]. Egyptian Informatics Journal,2023,24(2):313-325.
[7] ALHIJAJ T B,HAMEED S M,BARA'A A A. A decision tree-aware genetic algorithm for botnet detection[J]. Iraqi Journal of Science,2021,62(7):2454-2462.
[8] HOANG X D,VUX H. An improved model for detecting DGA botnets using random forest algorithm[J]. Information Security Journal:A Global Perspective,2022,31(4):441-450.
[9] ZANG X,CAO J,ZHANG X,et al. BotDetector:a system for identifying DGA-based botnet with CNN-LSTM[J]. Telecommunication Systems,2023,85(2):207-223.
[10] ABOU DAYA A,SALAHUDDIN M A,LIMAM N,et al. A graph-based machine learning approach for bot detection [C]// 2019 IFIP/IEEE Symposium on Integrated Network and Service Management(IM). IEEE,2019:144-152.
[11] UMER M F,SHER M,BI Y. Flow-based intrusion detection:Techniques and challenges[J]. Computers & Security,2017,70:238-254.
[12] THASEEN S,KUMAR C A. An analysis of supervised tree based classifiers for intrusion detection system[C]// 2013 International Conference on Pattern Recognition, Informatics and Mobile Engineering. IEEE,2013:294-299.
[13] HEGNA A. Visualizing spatial and temporal dynamics of a class of irc-based botnets[EB/OL]. https://ntnuopen. ntnu. no/ntnu-xmlui/bitstream/handle/11250/262293/353050 _ FULLTEXT. 01pdf?sequence=2&isAllowed=y.
[14] GARCIA S,GRILL M,STIBOREK J,et al. An empirical comparison of botnet detection methods[J]. Computers & Security,2014,45:100-123.
[15] HOCHREITER S,SCHMIDHUBER J. Long short-term memory[J]. Neural Computation,1997,9(8):1735-1780.
[16] ZHANG A,LIPTON Z C,LI M,et al. Dive into deep learning [M]. Cambridge:Cambridge University Press,2023.
[17] VASWANI A,SHAZEER N,PARMAR N,et al. Attention is all you need[C]// Proceedings of the 31st International Conference on Neural Information Processing Systems. 2017:6000-6010.
[18] GARCÍA S. Identifying,modeling and detecting botnet behaviors in the network[EB/OL]. https://www. stratosphereips. org/publications/2014/12/5/identifying-modeling-and-detecting-botnet-behaviors-in-the-network.
[19] BEIGI E B,JAZI H H,STAKHANOVA N,et al. Towards effective feature selection in machine learning-based botnet detection approaches[C]// 2014 IEEE Conference on Communications and Network Security. 2014.
[20] ZOU F T,TAN Y,WANG L,et al. Botnet detection based on generative adversarial network[J]. Journal on Communications,2021,42(7):95-106.
[21] SRINARAYANI K,PADMAVATHI B,KAVITHA D. Detection of Botnet Traffic using Deep Learning Approach[C]// 2023 International Conference on Sustainable Computing and Data Communication Systems(ICSCDS). IEEE,2023:201-206.



ZHANG Haixia, born in 1981, Ph.D, associate professor. Her main research interest is cyber information security.



HUANG Kezhen, born in 1988, Ph.D, associate professor. His main research interests include cyber security situation and cyber threat intelligence.

(责任编辑:何杨)