基于 CNN-BiGRU 的恶意域名检测方法

林梓宇 凌 捷
(广东工业大学计算机学院 广东 广州 510006)

摘 要 恶意域名检测对于防范僵尸网络等网络攻击具有重要意义。该文提出一种基于 CNN 和 BiGRU 的恶意域名检测方法 CNN-BiGRU-Focal,利用卷积神经网络和双向门控循环单元网络来进行特征的融合学习,并引入改进的 Focal Loss 函数用以解决数据不平衡问题。与 LSTM、CNN、GRU、ATT-CNN-BiLSTM 方法的对比实验表明,文章方法在多分类实验中检测准确率分别提高 1.43 百分点、2.89 百分点、1.27 百分点、2.43 百分点,在二分类实验中检测准确率分别提高 0.19 百分点、0.12 百分点、1.41 百分点、0.3 百分点。实验表明 CNN-BiGRU-Focal 方法在恶意域名的检测上有着更好的性能。

关键词 域名生成算法 深度学习 卷积神经网络 双向门控循环单元网络

中图分类号 TP393 **文献标志码** A DOI:10.3969/j.issn.1000-386x.2024.06.048

MALICIOUS DOMAIN DETECTION METHOD BASED ON CNN-BIGRU

Lin Ziyu Ling Jie
(*School of Computer, Guangdong University of Technology, Guangzhou 510006, Guangdong, China*)

Abstract Malicious domain name detection is of great significance to prevent botnet and other network attacks. This paper proposes a malicious domain name detection method called CNN-BiGRU-Focal. Convolutional neural network and bidirectional gated cyclic unit network were used for feature fusion learning, and an improved focal loss function was introduced to solve the problem of data imbalance. Compared with LSTM, CNN, GRU and ATT-CNN-BiLSTM method, the detection accuracy of the proposed method is improved by 1.43, 2.89, 1.27 and 2.43 percentage points in multi-classification experiments, and 0.19, 0.12, 1.41 and 0.3 percentage points in binary classification experiments. Experiments show that CNN-BiGRU-Focal method has better performance in the detection of malicious domain names.

Keywords DGA Deep learning CNN BiGRU

0 引 言

目前,域名生成算法(Domain Generation Algorithms, DGA)可以产生许多恶意域名、勒索软件等利用产生的域名与它的命令控制服务器进行连接,一旦连接成功后,恶意软件就会在宿主机器上进行各种恶意活动。除此之外,僵尸网络[1]也会利用 DGA 来生成恶意域名,在僵尸程序中预先植入编写好的 DGA 种子,之后僵尸程序利用该算法产生大量的恶意域名,使得攻击者会利用其中产生的某些域名,注册成为某个僵尸网络的命令与控制服务器。如果僵尸程序顺利地与其中的域名连接成功,就表示僵尸主机与命令控制服务器通信成功,接下来就开始执行攻击者的命令。由于 DGA 可以在短时间内产生大量的随机域名,通过黑名单过滤的防御方式在效果上十分有限,针对 DGA 生成的恶意域名检测方法研究,对于恶意软件与僵尸网络的防御有着十分重要的意义。

本文提出一种基于 CNN-BiGRU 的恶意域名检测方法 CNN-BiGRU-Focal,通过卷积神经网络对字符级的域名进行特征提取以及循环神经网络提取的时序特征,通过特征融合的方式结合两者的优势,之后使用

收稿日期:2021-01-28。广东省重点领域研发计划项目(2019B010139002);广州市重点领域研发计划项目(202007010004)。
林梓宇,硕士生,主研领域:网络信息安全技术。**凌捷**,教授。

Focal Loss 函数,用以提高分类的准确率。

1　相关工作

早期传统的恶意域名检测方法包括使用逆向工程技术和蜜罐技术[2],这些方法耗费资源比较大、周期长,一旦 DGA 变化,就难以进行准确识别。文献[3]利用聚类的方法,给域名数据集设定数据属性,通过不断训练与测试之后生成相应的规则库来识别潜在的恶意域名。文献[4-5]对恶意域名进行字符串分析,针对字符建立一些人工特征,再利用分类器来进行预测。但是这种方法遇到域名的算法种子进行更新时就要从头设计特征,导致后续计算量加大并且人工提取特征太过于耗时,效率低下。随后有学者利用机器学习来检测恶意域名,但机器学习方法严重依赖于人工构造的特征,一旦被识破,就容易被 DGA 规避,而且方法效率较低[6]。近年来,随着深度学习的飞速发展以及在不同应用场景的突出表现,使得相关研究人员开始使用深度学习的方法进行 DGA 产生的恶意域名的检测,文献[7]提出采用长短期记忆网络(Long Short Tem Memory,LSTM)模型来对恶意域名进行检测,长短期记忆网络的检测效果虽然较好,但训练与测试时间比较长。文献[8]则使用门控循环单元(Gated Recurrent Unit,GRU)来进行恶意域名检测,进一步提升了模型的收敛速度与鲁棒性。文献[9]使用 GRU 结合注意力的方法,有效提升了对低随机恶意域名的识别率。文献[10]采用混合词向量的方法来提高深度学习模型在恶意域名多分类上的性能,但仍然存在部分样本识别率低下的问题。

深度学习模型在恶意域名检测任务中都取得了不错的效果,但是因为不同 DGA 类别的样本数量以及生成算法的不同,如 suppobox 家族,它是随机地从英文单词表中选取两个单词拼接在一起;symmi 家族组合日期,常量以及随机数生成种子来干扰识别;matsnu 家族通过随机挑选英文单词与连接线来生成相应的恶意域名等,导致深度学习模型在多分类任务上,特别是针对某些恶意域名家族类别上仍然存在识别率低的问题。

卷积神经网络最早应用于图像的处理,在文献[11]中被用于检测恶意域名并且取得了不错的效果,说明整个卷积模型也适用于词嵌入层处理的字符级文本序列。文献[12]说明卷积神经网络在字符级别的文本分类上可以取得不错的效果,文献[13-14]对比了多种字符级别的文本分类算法,发现采用并行的 CNN 模型的效果最佳。为了对域名这种短字符进行分类,本文模型采用了 4 个并行的卷积神级网络,并且每个卷积神经网络的卷积核大小都不一样,参数大小分别设置 2、3、4、5,因为不同大小的卷积核可以从字符序列中提取到不同大小的特征粒度,可以更加有效地进行后续的分类。

卷积神经网络对于局部特征提取更为有效,而不易发现文本序列特征,而循环神经网络能够更好地利用上下文来进行提取特征。循环神经网络作为广泛应用于文本处理的深度学习模型,被经常用来解决一些时序问题,RNN 有两个变体,一个是 LSTM,一个是 GRU。

为了使得模型能够自动发现恶意域名字符的隐含特征,本文用两种深度学习模型相结合的方式来提取特征并进行分类任务,提高恶意域名检测的准确率。文献[15]通过实验对比了多种深度学习模型在恶意域名检测上的效果,发现通过结合 CNN 与 BiGRU 或 BiLSTM 的方式可以有效地提升模型检测的性能,其中 BiGRU 与 BiLSTM 效果相差无几,但是训练时间更短,模型收敛速度更快,所以本文采用 CNN 与 BiGRU 相结合的方式来搭建神经网络模型。

2　方法描述

2.1　CNN 模型

卷积神经网络的核心是提取局部特征。卷积神经网络能够高效率地提取字符在整个域名序列中的局部上下文特征,特别对于卷积操作,通过选择不同大小的卷积核,模型可以很好地提取不同程度的信息,得到粒度不同的域名字符特征信息。对于不同大小的卷积核来说,能够提取相应的某一类特征。对于不同的文本序列,通常情况它的上下文信息并不一致,所以本文模型使用多核卷积来获取局部特征,卷积核大小分别设置为 2、3、4、5,滤波器数量为 256。

图 1 为一维卷积层提取字符特征的过程,此处的一维卷积层设置了滤波器数量为 256,卷积核大小设置为 3,步幅 stride 设置为 1,该层每次从字符序列中选取 3 个字符后送入滤波器,然后再接着从序列中选取下一个字符进行特征提取。该神经网络相应地生成 3-grams 的字符特征,这些将成为接下来区分这些域名字符的有效特征。通过在整个域名文本序列上应用相同大小的滤波器,所需的计算量与传统的多层感知机相比,时间可以大幅度减少。另外,由于每个卷积核都是独立地在每 3 个字符上同时进行卷积计算。这个并行

化的处理方式带来的计算时间的减少是卷积神经网络相比循环神经网络在处理文本分类上所带来的主要优势之一。

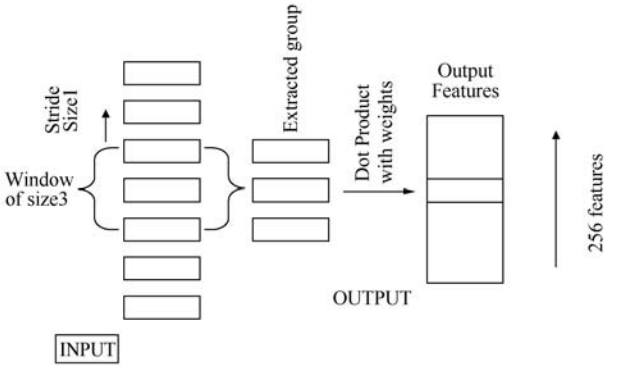

图 1 一维卷积提取字符特征的过程

卷积层作为卷积神经网络的核心,其通过对单词进行卷积运算来获取更加高级的特征表示,通过设置卷积核的大小可以保留域名字符串的局部语序信息,并且不会带来特征稀疏的问题,从而省去了降维的过程,避免了在数据降维过程中信息的丢失。每个卷积核与输入特征的不同局部窗口进行卷积操作,将运算得到的特征向量经过非线性激活函数 f 处理后便产生本层要输出的特征,公式如下:

$$\boldsymbol{S}_t = f(\boldsymbol{WX} + \boldsymbol{b}) \tag{1}$$

式中:$\boldsymbol{X}$ 为输入向量;$\boldsymbol{W}$ 是卷积核;参数 $\boldsymbol{b}$ 是偏置。激活函数有 Sigmoid 或者 ReLU 等。此处本文采用了 ReLU 函数,因为该函数可以降低模型学习周期,加快损失函数的收敛速度,减少计算量。公式如下:

$$f = \max(0, \boldsymbol{X}) \tag{2}$$

式中:$\boldsymbol{X}$ 表示上一层网络的输出向量。ReLU 函数把输入的负值变 0,正值保持不变,使得网络具有稀疏性,缓解过拟合。接下来将卷积层提取到的字符特征送入池化层。池化操作通常有两种:最大池化和平均池化。此处为了结合两种池化方法的优点,引进了 K-Max average pooling[16] 池化方法,公式如下:

$$\hat{c} = \frac{\sum_{i=1}^{k} \max_i \{c_1, c_2, \cdots, c_{n-h+1}\}}{k} \tag{3}$$

式中:$\hat{c}$ 代表了 K-Max 平均池化的输出结果,K-Max 平均池操作通过获取采样过程中的 K 个最大值,然后计算它们的平均值并作为最终的输出值。这种方法避免了在最大池化操作中仅仅挑选出最大的特征,使得模型可以考虑其他特征的影响。同样该方法也可以避免平均池化操作中距离较大的小特征对于整体特征强度的削弱。因此,K-Max 平均池化方法既可以保留原来的特征强度,又可以保证从样本特征中提取出最有效的特征。

2.2 BiGRU 模型

卷积神经网络对于输入的时间序列不敏感,为了克服该问题引入了循环神经网络。本文模型选用 BiGRU,相比较于 LSTM,GRU 同样也是基于门控制原理,但是简化了整个神经网络的结构,减低了模型的复杂度,减少了整个网络的训练时间,并且效果与 LSTM 基本相同,如图 2 所示。

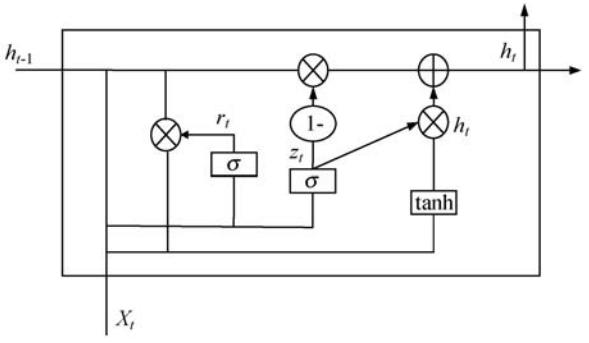

图 2 GRU 单元门结构

计算公式如下:

$$\boldsymbol{z}_t = \sigma(\boldsymbol{W}_z \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{x}_t]) \tag{4}$$

$$\boldsymbol{r}_t = \sigma(\boldsymbol{W}_r \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{x}_t]) \tag{5}$$

$$\tilde{\boldsymbol{h}}_t = \tanh(\boldsymbol{W} \cdot [\boldsymbol{r}_t * \boldsymbol{h}_{t-1}, \boldsymbol{x}_t]) \tag{6}$$

$$\boldsymbol{h}_t = (1 - \boldsymbol{z}_t) * \boldsymbol{h}_{t-1} + \boldsymbol{z}_t * \tilde{\boldsymbol{h}}_t \tag{7}$$

式中:$\boldsymbol{z}_t$ 代表更新门,更新门用于控制前一时间节点保存到当前时间节点的信息量,值越大表示前一节点隐藏状态的信息对当前节点影响越大。式(4)中,更新门接收 $\boldsymbol{h}_{t-1}$ 和 $\boldsymbol{x}_t$ 进行计算并且使用激活函数处理,激活结果越大说明越多的信息被保存。$\boldsymbol{r}_t$ 代表重置门,控制前一节点的信息量被写入当前候选集 $\tilde{\boldsymbol{h}}_t$,如果值为 0,则可以过滤掉之前节点状态信息。式(5)中,重置门对 $\boldsymbol{h}_{t-1}$ 与 $\boldsymbol{x}_t$ 进行与更新门类似的操作,激活结果越大说明越多的信息被写入。

接下来式(6)中 GRU 将先前重置门输出的 $\boldsymbol{r}_t$ 和前一时刻的输入状态 $\boldsymbol{h}_{t-1}$,当前时刻输入 $\boldsymbol{x}_t$,进行运算形成当前节点新的记忆内容 $\tilde{\boldsymbol{h}}_t$,式(6)中 GRU 结合前一时刻的输入状态 $\boldsymbol{h}_{t-1}$ 与当前新的记忆内容 $\tilde{\boldsymbol{h}}_t$ 来确定当前节点的状态 $\boldsymbol{h}_t$。σ 为 Sigmoid 函数,tanh 为双曲正切函数,W 为权重,$\boldsymbol{x}_t$ 代表了 t 时刻输入的数据,$\boldsymbol{h}_{t-1}$ 代表上一时刻输入的数据,$\boldsymbol{h}_t$ 为当前时刻的输出。

对于单向的 GRU 神经网络,它的状态从前往后进行输出的,导致这种结构在处理序列信息时会出现弊端,因为有时候对于需要处理的序列信息它是由前面的输入信息以及后面的输入信息共同起作用的,所以需要借助双向门控循环网络来解决这一问题。双向 GRU 神经网络借鉴了双向 RNN 以及 LSTM 的优点做了进一步的改善,如图 3 所示。

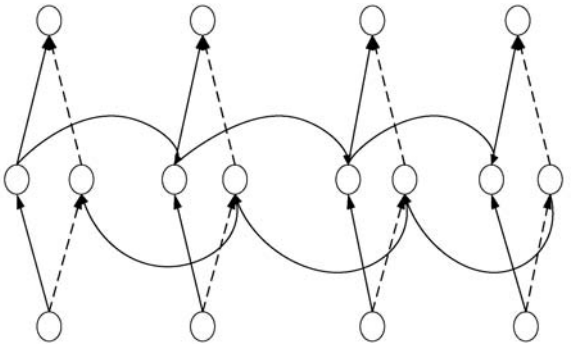

图 3　双向 GRU 神经网络结构

图 3 中从左往右整个神经网络的更新公式为:

$$\overrightarrow{\boldsymbol{h}}_t = f(\overrightarrow{\boldsymbol{W}}\boldsymbol{x}_t + \overrightarrow{\boldsymbol{V}}\boldsymbol{h}_{t-1} + \overrightarrow{\boldsymbol{b}}) \tag{8}$$

从右往左整个神经网络的更新公式为:

$$\overleftarrow{\boldsymbol{h}}_t = f(\overleftarrow{\boldsymbol{W}}\boldsymbol{x}_t + \overleftarrow{\boldsymbol{V}}\boldsymbol{h}_{t+1} + \overleftarrow{\boldsymbol{b}}) \tag{9}$$

双向 GRU 最终输出公式为:

$$\boldsymbol{y}_t = g(\boldsymbol{U}[\overrightarrow{\boldsymbol{h}}_t; \overleftarrow{\boldsymbol{h}}_t] + \boldsymbol{c}) \tag{10}$$

式中:$\boldsymbol{W}$、$\boldsymbol{V}$、$\boldsymbol{U}$ 均代表了权重矩阵;$\boldsymbol{b}$、$\boldsymbol{c}$ 代表偏置矩阵。为了防止模型的过拟合,在 BiGRU 层后面使用 Dropout 技术。该技术可以在模型训练时随机地从神经网络中移除一定比例的 BiGRU 单元,实现模型的正则化。

2.3　本文方法

本文提出的恶意域名检测方法简称 CNN-BiGRU-Focal,其结构如图 4 所示,由三部分组成:词嵌入、特征提取、分类。

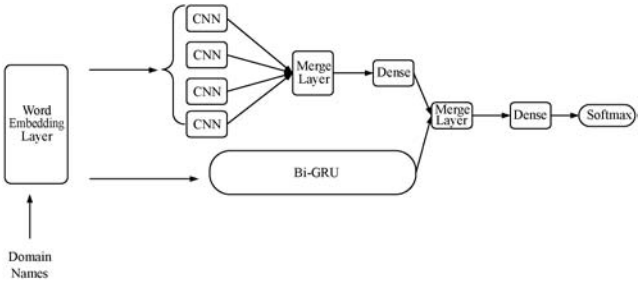

图 4　CNN-BiGRU-Focal 结构

2.3.1　词嵌入

深度神经网络只能处理数值化的向量,对于是字符形式的域名数据是无法直接进行处理的,常采用词嵌入的技术对字符进行向量化,首先使用 one-hot 编码技术对原始域名数据进行编码,但是这种编码后的数据如果直接送入模型进行学习的话,会导致模型参数数量巨大且带来数据稀疏的问题。将 one-hot 编码过的域名字符送入模型之前,要经过词嵌入层来生成单词的词向量。在自然语言处理中,经常使用 Glove、Word2Vec 等方法来生成词向量,这些预训练方法将网络上的大量网页资料作为学习语料,进行语言模型训练,学习自然语言中含有语义信息的词向量。

但恶意域名数据都是由杂乱的字符所组成的,如 nymaim 家族所产生的域名 qpwaruus. net,所以实际上 DGA 产生的域名大多数并不是一个自然语言中存在的单词,所以直接使用 Glove 等方法是难以进行的。因此本文对域名字符串中的字符进行单独处理,将每个恶意域名字符串进行拆分,得到单个字符组成的[…,['q','p','w','a','r','u','u','s','.','n','e','t'],….]字符列表。随后将经过 one-hot 编码的字符送入词嵌入层。

2.3.2　特征提取

特征提取层采用 CNN 和 BiGRU 这两种深度学习模型,对于词嵌入层输出的域名矩阵进行自动的特征提取。本文方法中的并行 CNN 模型使用的卷积核(kernel)大小分别为 2、3、4、5,滤波器(filter)的数量为 256。CNN 首先使用多核卷积来提取域名字符的局部特征,卷积核在输入的域名矩阵上通过滑动操作来进行特征的提取,此处的特征提取效果相当于传统的 n-grams 方法。卷积操作得到的结果接下来送入池化层,此处采用的是 K-Max 平均池化操作,剔除掉一些不必要的冗余特征信息。

BiGRU 模型可以实现字符序列的正向与反向的处理,与单独的 GRU 模型相比可以提取出更多的上下文时序信息。BiGRU 通过利用并行通道的方式,保证了模型可以获取正向累积的上下文信息,又可以获取反向的累积上下文信息。通过这种方式,BiGRU 可以从输入中提取出更加丰富的特征信息。当两种模型分别提取出相应特征后,使用向量连接的操作,对 CNN 和 BiGRU 两种网络结构输出的特征向量矩阵进行学习运算,使得这两个进行独立特征学习的分支融合起来,进行最终的域名分类时可以使用不同粒度的特征。

2.3.3　分　类

基于 CNN 与 BiGRU 两种模型融合的特征后,完成最后的域名分类任务。目前来说,针对恶意域名检测可以分为两大类,一类属于二分类任务,此时全连接层仅仅设置为一个节点,另一类属于多分类任务,此时的全连接层节点数设置为域名的家族种类数量。

常用的交叉熵损失函数它的函数计算公式如下:

$$L = \begin{cases} -\log y' & y = 1 \\ -\log(1 - y') & y = 0 \end{cases} \tag{11}$$

式中:y 代表了样本的真实标签值;y'代表了模型预测的样本标签值。对于一个正类,如果它的预测值越是接近于 1,那么损失函数就越小;对于一个反类,预测值越接近于 0,同样的损失函数也越小。当训练集中存在大量容易分类的样本时,会使得模型的预测结果容易接近真实值,导致整个模型的最终优化效果很差。为了解决这个问题,文献[17]提出了一个优化的交叉

熵损失函数 Focal Loss,它聚焦于样本集中难以分类的项,它的计算公式如下:

$$L_{fl}=\begin{cases}-(1-y')^{\gamma}\log y' & y=1\\ -y'^{\gamma}\log(1-y') & y=0\end{cases} \tag{12}$$

考虑到在恶意域名检测中,每个域名家族的底层生成算法不同,所产生的恶意域名的字符的异常度也不同,因此在整个恶意域名样本集中,存在一部分难分类的恶意域名家族样本集合。为了减少数据的不平衡对模型准确率的影响,此处模型使用了改进的Focal Loss 损失函数,用于提高分类的准确率公式如下:

$$L_{fl}=\begin{cases}-\alpha(1-y')^{\gamma}\log y' & y=1\\ -(1-\alpha)y'^{\gamma}\log(1-y') & y=0\end{cases} \tag{13}$$

式中:y'代表了模型输出预测值;y 是样本数据的真实标签值;α 是加权因子。

3 实验与结果分析

3.1 数据集与测试环境

实验数据来自于公开数据集,其中正常样本数据集来自于 Alexa 网站,恶意样本数据集来自于 360 NetLab DGA 开放数据。本文选取 Alexa 数据集中排名前 10 万的域名数据作为正样本数据,360 NetLab DGA 域名数据中包含了 40 多种家族的 DGA 数据。但是由于少部分域名种类的数据量过少,不足以支撑深度学习分类模型的验证,所以本文选取了其中数据量较为充足的 41 个类来进行实验。实验数据划分为三部分,训练集数量占总数据量的 80%,验证集占 10%,测试集占 10%。

本实验的测试环境:处理器为英特尔酷睿 i5-9300H @2.2 GHz,图形加速卡为 NVIDIA GeForce GTX 1650,内存为 16 GB,操作系统为 Windows 10。使用 Keras 和 TensorFlow 来搭建神经网络模型。

3.2 评价指标

本文采用的实验评估指标为:准确率、精确率、召回率、F1 值。

1)准确率表示预测结果中预测正确所占的比例:

$$A_{\text{ccuracy}}=\frac{T_P+T_N}{T_P+T_N+F_P+F_N} \tag{14}$$

2)精确率表示在预测结果中为正类且预测正确所占的比例:

$$P_{\text{recision}}=\frac{T_P}{T_P+F_P} \tag{15}$$

3)召回率表示在所有正类中预测结果为正确的所占的比例:

$$R_{\text{ecall}}=\frac{T_P}{T_P+F_N} \tag{16}$$

4)F1 值为精确率与召回率的调和评价值:

$$F_1=\frac{P_{\text{recision}}\times R_{\text{ecall}}\times 2}{P_{\text{recision}}+R_{\text{ecall}}} \tag{17}$$

3.3 实验结果分析

在多分类实验中,选取了 LSTM 方法[7]、CNN[15]方法、GRU[9]方法、ATT-CNN-BiLSTM[18]方法以及本文设计的 CNN-BiGRU-Focal 方法进行对比实验,CNN-BiGRU-Focal 方法方法参数设置为:词嵌入层为 128 维,BiGRU 选取 256 个单元,Dropout 失活率均设置为 0.5。

通过表 1 可以看出,CNN-BiGRU-Focal 方法在恶意域名多分类上均优于其他方法,检测准确率分别提高了 1.43 百分点、2.89 百分点、1.27 百分点、2.43 百分点。分析表 2 可以发现,对于大部分类别,几种方法的检测精确率均高于 95%,CNN-BiGRU-Focal 方法对于这 10 种类别的检测精确率都在 80% 及以上。特别是对于 mydoom、nymaim、suppobox、tinba,其他 4 种方法的检测精确率均处于偏低水平,甚至出现了无法识别的情况,而 CNN-BiGRU-Focal 方法的检测精确率达到了 80%、86.48%、96.33%、91.23%。

表 1 多分类实验对比表(%)

方法	Accuracy	Precision	Recall	F1
LSTM	91.66	92.77	90.20	90.83
CNN	90.20	93.41	91.66	91.97
GRU	91.82	93.50	91.82	92.04
ATT-CNN-BiLSTM	90.66	93.11	90.66	91.35
CNN-BiGRU-Focal	93.09	94.24	93.09	93.36

表 2 在 10 种类别上的多分类检测精确率对比表(%)

类别	LSTM	CNN	GRU	ATT-CNN-BiLSTM	CNN-BiGRU-Focal
emotet	99.51	97.89	99.61	99.70	99.80
murofet	93.56	88.54	93.59	92.05	94.17
mydoom	0	0	0	50.00	80.00
nymaim	0	0	0	0	86.48
pykspa_v1	97.60	98.28	98.98	97.69	99.08
qadars	99.23	99.49	98.83	98.24	99.94
rovnix	99.90	93.97	100.00	99.80	100.00
simda	99.07	98.64	99.57	97.82	99.47
suppobox	93.63	80.21	87.92	56.00	96.33
tinba	88.81	85.40	88.34	87.07	91.23

实验的二分类结果如表 3 所示,可以看到 5 种方法都取得了很好的效果,尤其是 CNN-BiGRU-Focal 方法在四项指标中均好于其他 4 种方法,检测准确率相比于其他 3 种方法分别提高了 0.19 百分点、0.12 百分点、1.41 百分点、0.3 百分点。AUC 代表了 ROC 曲线和坐标轴围成面积大小,如果值越大,就说明模型分类性能更好,可以看到 CNN-BiGRU-Focal 方法的 AUC 值在所有方法中是最大的。以上对比实验说明本文提出的 CNN-BiGRU-Focal 方法对于检测恶意域名,特别是针对难识别的恶意域名方面是有效的。

表 3　二分类实验对比表(%)

方法	Accuracy	Precision	Recall	F1	AUC
LSTM	97.28	97.66	98.47	98.07	96.49
CNN	97.35	97.53	98.71	98.12	96.44
GRU	96.06	96.67	97.74	97.20	94.94
ATT-CNN-BiLSTM	97.17	98.52	97.43	97.97	97.01
CNN-BiGRU-Focal	97.47	98.88	97.50	98.18	97.46

4　结　语

本文提出了一种基于 CNN-BiGRU 的恶意域名检测方法,通过使用模型并行学习的方法,同时考虑到恶意域名的字符数据特征以及样本数量不平衡的特点,结合 CNN 对字符级域名的高效特征提取以及 BiGRU 在时间序列处理上的优势,在分类时使用改进的 Focal Loss 函数来解决样本数量不平衡问题。实验结果表明,本文方法在多分类的检测任务上检测准确率与对比方法相比分别提高了 1.43 百分点、2.89 百分点、1.27 百分点、2.43 百分点,在二分类的检测任务上与对比方法相比分别提高了 0.19 百分点、0.12 百分点、1.41 百分点、0.3 百分点。这说明本文方法在恶意域名多分类检测上比现有检测方法有明显的优势。

参　考　文　献

[1] Abu R M, Zarfoss J, Monrose F, et al. A multifaceted approach to understanding the botnet phenomenon[C]//6th ACM SIGCOMM Conference on Internet Measurement,2006:41 - 52.

[2] Wang T S, Lin H T, Cheng W T, et al. DBod: Clustering and detecting DGA-based botnets using DNS traffic analysis[J]. Computers and Security,2017,64:1 - 15.

[3] Tong V, Nguyen G. A method for detecting DGA botnet based on semantic and cluster analysis[C]//7th Symposium on Information and Communication Technology,2016:272 - 277.

[4] 张维维,龚俭,刘茜,等. 基于词素特征的轻量级域名检测算法[J]. 软件学报,2016,27(9):2348 - 2364.

[5] Erquiaga M J, Catania C, García S. Detecting DGA malware traffic through behavioral models[C]//IEEE Biennial Congress of Argentina,2016:1 - 6.

[6] Chang J, Venkatasubramanian K, West A G, et al. Analyzing and defending against web-based malware[J]. ACM Computing Surveys,2013,45(4):1 - 35.

[7] Woodbridge J, Anderson H S, Ahuja A, et al. Predicting domain generation algorithms with long short-term memory networks[EB]. arXiv:1611.00791,2016.

[8] 陈立国,张跃冬,耿光刚,等. 基于 GRU 型循环神经网络的随机域名检测[J]. 计算机系统应用,2018,27(8):198 - 202.

[9] 陈立皇,程华,房一泉. 基于注意力机制的 DGA 域名检测算法[J]. 华东理工大学学报(自然科学版),2019,45(3):478 - 485.

[10] 杜鹏,丁世飞. 基于混合词向量深度学习模型的 DGA 域名检测方法[J]. 计算机研究与发展,2020,57(2):443 - 446.

[11] Yu B, Gray D L, Pan J, et al. Inline DGA detection with deep networks[C]//IEEE International Conference on Data Mining Workshops,2017:683 - 692.

[12] Zhang X, Zhao J B, LeCun Y. Character-level convolutional networks for text classification[C]//28th International Conference on Neural Information Processing Systems,2015:649 - 657.

[13] Yu B, Pan J, Hu J M, et al. Character level based detection of DGA domain names[C]//International Joint Conference on Neural Networks,2018:1 - 8.

[14] Saxe J, Berlin K. Expose: A character-level convolutional neural network with embeddings for detecting malicious URLs, file paths and registry keys[EB]. arXiv:1702.08568,2017.

[15] 裴兰珍,赵英俊,王哲,等. 采用深度学习的 DGA 域名检测模型比较[J]. 计算机科学,2019,46(5):111 - 115.

[16] Zhang Y S, Zheng J, Jiang Y R, et al. A text sentiment classification modeling method based on coordinated CNN-LSTM-attention model[J]. Chinese Journal of Electronics,2019,28(1):120 - 126.

[17] Lin T Y, Goyal P, Girshick R, et al. Focal loss for dense object detection[C]//IEEE International Conference on Computer Vision,2017:2980 - 2988.

[18] Ren F L, Jiang Z W, Wang X R, et al. A DGA domain names detection modeling method based on integrating an attention mechanism and deep neural network[J]. Cybersecurity,2020,3(1):1 - 13.