第 46 卷 第 3 期 计 算 机 工 程 2020 年 3 月
Vol. 46 No. 3 Computer Engineering March 2020

·网络空间安全· 文章编号：1000-3428(2020)03-0129-09 文献标志码：A 中图分类号：TP391

基于类图像处理与向量化的大数据脚本攻击智能检测

张海军[a]，陈映辉[b]

（嘉应学院 a. 计算机学院；b. 数学学院，广东 梅州 514015）

摘 要： 通过类图像处理与向量化方法对访问流量语料库大数据进行词向量化处理，实现面向大数据跨站脚本攻击的智能检测。利用类图像处理方法进行数据获取、数据清洗、数据抽样和特征提取，设计一种基于神经网络的词向量化算法，得到词向量化大数据。在此基础上，提出多种不同深度的 DCNNs 智能检测算法。设置不同的超参数进行实验得到算法的识别率均值、方差和标准差，结果表明，该算法具有较高的识别率和稳定性。

关键词： Web 入侵检测；跨站脚本攻击；自然语言处理；大数据；网络空间安全

开放科学（资源服务）标志码（OSID）：

中文引用格式： 张海军，陈映辉. 基于类图像处理与向量化的大数据脚本攻击智能检测[J]. 计算机工程，2020，46(3)：129-137，143.

英文引用格式： ZHANG Haijun，CHEN Yinghui. Intelligent detection for big data scripting attack based on image processing inspired method and vectorization[J]. Computer Engineering，2020，46(3)：129-137，143.

Intelligent Detection for Big Data Scripting Attack Based on Image Processing Inspired Method and Vectorization

ZHANG Haijun[a]，CHEN Yinghui[b]

（a. College of Computer Science；b. School of Mathematics，Jiaying University，Meizhou，Guangdong 514015，China）

【Abstract】 In this paper，the methods similar to image processing and vectorization are used for the verctorization of access traffic corpus big data，and the intelligent detection for big data cross-site scripting attack is achieved. Besides，this paper uses methods similar to image processing for data acquisition，data cleaning，data sampling and feature extraction. Then，a word vectorization algorithm based on neural network is designed to obtain the big data of word vectorization. On this basis，the DCNNs intelligent detection algorithm with different depth is proposed. Finally，experiments with different hyper-parameter are conducted，and the obtained average recognition rate，variance and standard deviation show that the proposed algorithm has high recognition rate and stability.

【Key words】 Web intrusion detection；Cross-Site Scripting（XSS）attack；natural language processing；big data；cyberspace security

DOI： 10.19678/j.issn.1000-3428.0053360

0 概述

随着互联网、云计算、物联网、大数据等技术的快速发展以及数以万计网络接入点、移动终端和网络应用的出现，产生了大量蕴含较高价值的大数据，这给网络空间安全带来了前所未有的挑战。从服务器交易系统的数据库数据到各终端业务的系统数据，如各种流水操作、网购记录、网络浏览历史、播放的音视频、微博和微信等基于移动或 Web 应用的数据等，使得基于 Web 应用的攻击逐渐成为网络中的主要攻击，如跨站脚本（Cross-Site Scripting，XSS）攻击[1]，其表现为：网络钓鱼，盗取用户的账号和密码；盗取用户 Cookie 数据，获取用户隐私，或者利用用户身份进行进一步操作；劫持浏览器会话，从而冒充用户执行任意操作，如非法转账、强制发表博客；强制弹出页面广告，刷流量；进行恶意操作，如篡改页面信息、删除文章、传播跨站蠕虫脚本、网挂木马；进行基于大量客户端的攻击，如 DDOS 攻击；联合其他

基金项目： 国家自然科学基金（61171141，61573145）；广东省自然科学基金重点项目（2014B010104001，2015A030308018）；广东省普通高等学校人文社会科学省市共建重点研究基地课题（18KYKT11）；广东省嘉应学院自然科学基金重点项目（2017KJZ02）。

作者简介： 张海军（1978—），男，讲师、博士，主研方向为智能计算、自然语言处理、模式识别；陈映辉（通信作者），讲师。

收稿日期： 2018-12-10 **修回日期：** 2019-03-26 **E-mail：** nihaoba_456@163.com

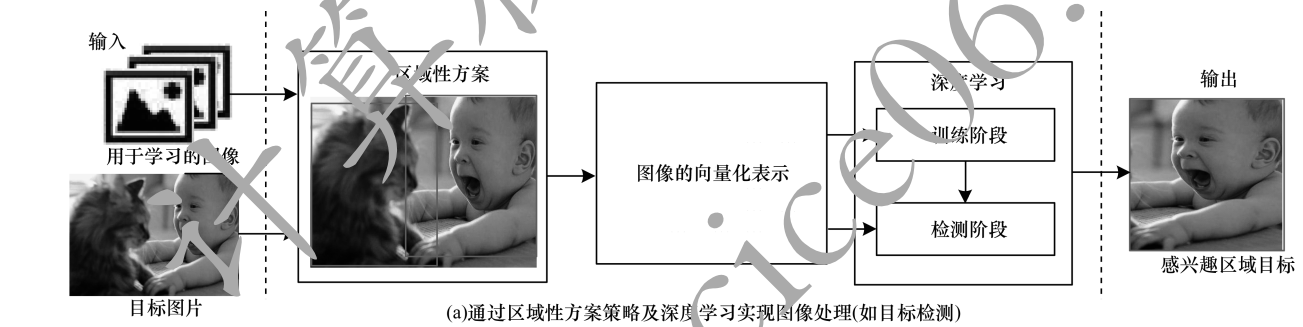
漏洞(如 CSRF);进一步渗透网站。

传统的计算机病毒检测方法主要利用病毒特征库中的已有特征,通过提取相应样本的特征,在病毒库中搜索并比较是否存在相匹配的特征,从而确定病毒是否存在。这种方法主要基于已知的病毒进行检测,难以发现新的病毒,特别是对于变形病毒其更加无能为力,而且针对大数据问题时效率较低。当前安全防护措施已经由过去的"80% 防护 +20% 检测及响应"变成了"20% 防护 +80% 检测及响应"。深度学习以其强大的自适应性、自学习能力在语音、图像、自然语言处理等方面取得了比传统机器学习方法更好的效果,特别是在解决大数据问题时,其效果更好。

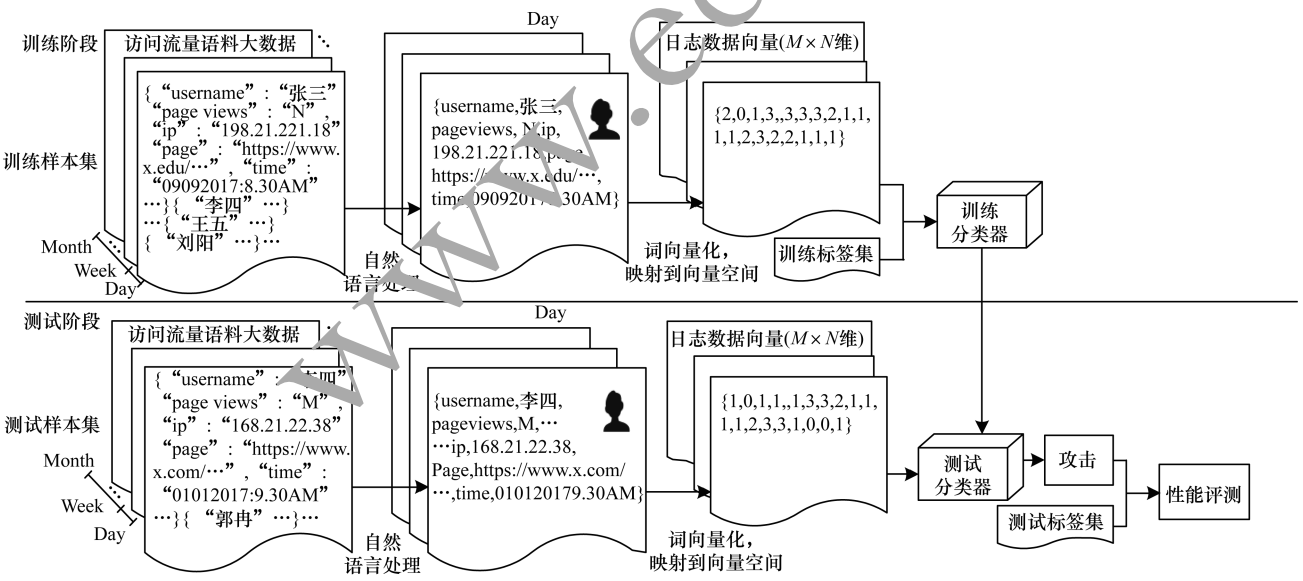
本文借鉴类图像处理过程,设计一种基于神经网络的词向量化算法,对访问流量语料库大数据进行词向量化处理,通过理论分析和推导实现多种不同深度的深层卷积神经网络算法,从而对大数据跨站脚本攻击进行智能检测。

1 语料大数据处理及向量化

Web 入侵检测本质上是基于日志文本的分析[2],即对访问流量语料大数据进行分析。首先,进行自然语言处理,由于当前基于安全防护的语料样本比较缺乏,标注好标签的样本更少,因此需要进行数据处理和建模,具体为:1)语料获取;2)语料预处理,包括清洗、分词、词性标注、去停用词;3)初级数据分析,如 URL 参数个数、字符分布、访问频率等分析。其次,进行词向量化,将词映射到向量空间,在计算机中,任何信息都是以 0 和 1 的二进制序列表示,如所有的字符(包括字母、汉字、英语单词等语言文字)都有一个编码。本文将大数据日志文本转换成数值数据并以矩阵表示,再基于词向量方法进行数据处理和分析,即将攻击报文转换成类似于图像数据(像素)的矩阵,也将字符串序列样本转换成具有一定维度值的向量,再对词向量进行数值化的特征提取和分析,如数据抽样、矩阵相关性维数约减、特征提取、降维、聚类等运算。最后,进行模型训练与数值分析,实现用户行为分析、网络流量分析和欺诈检测等。上述过程原理如图 1 所示,向量化处理过程详见 3.1 节实验部分,其实现了语料大数据的获取、处理、建模、分词和词向量化等[3]。



(a)通过区域性方案策略及深度学习实现图像处理(如目标检测)



(b)语义情景分析及向量化过程

图 1 大数据跨站脚本攻击智能检测原理

Fig. 1 Intelligent detection principle of big data cross-site scripting attack

2　算法设计

2.1　词向量化算法设计

利用 CBOW 实现词向量,即已知上下文词语来预测当前词语出现的概率。为此,需要最大化对数似然函数:

$$L=\sum_{w\in C}\lg p(w\mid C(w)) \tag{1}$$

其中,w 表示语料库 C 中的词。式(1)可以看作多分类问题,因为多分类是由二分类组合而成,所以可以使用 Hierarchical Softmax 方法进行求解。先计算 w 的条件概率,如下:

$$p(w\mid C(w))=\prod_{j=2}^{l^w}p(d_j^w\mid \boldsymbol{X}_w,\boldsymbol{\theta}_{j-1}^w) \tag{2}$$

其中,$\boldsymbol{X}_w$ 表示输入,p^w 表示路径,l^w 表示节点个数,p_1^w,p_2^w,…,$p_{l^w}^w$ 表示各节点,d_2^w,d_3^w,…,$d_{l^w}^w\in\{0,1\}$ 表示词 w 的编码,d_j^w 表示路径中第 j 个节点对应的编码,$\boldsymbol{\theta}_1^w$,$\boldsymbol{\theta}_2^w$,…,$\boldsymbol{\theta}_{l^w-1}^w\in\mathbb{R}^m$ 表示路径上非叶子节点对应的参数向量。式(2)右边的每一项都是一个逻辑斯特回归:

$$p(d_j^w\mid \boldsymbol{X}_w,\boldsymbol{\theta}_{j-1}^w)=\begin{cases}\sigma(\boldsymbol{X}_w^{\mathrm{T}}\boldsymbol{\theta}_{j-1}^w), & d_j^w=0\\ 1-\sigma(\boldsymbol{X}_w^{\mathrm{T}}\boldsymbol{\theta}_{j-1}^w), & d_j^w=1\end{cases} \tag{3}$$

其中,$\sigma(x)$ 为 sigmoid 函数。由于 d 只取 0 和 1,因此式(3)可以以指数的形式表示为:

$$p(d_j^w\mid \boldsymbol{X}_w,\boldsymbol{\theta}_{j-1}^w)=[\sigma(\boldsymbol{X}_w^{\mathrm{T}}\boldsymbol{\theta}_{j-1}^w)]^{1-d_j^w}\cdot[1-\sigma(\boldsymbol{X}_w^{\mathrm{T}}\boldsymbol{\theta}_{j-1}^w)]^{d_j^w} \tag{4}$$

将式(4)代入式(1)可得:

$$\begin{aligned}L=&\sum_{w\in C}\lg\prod_{j=2}^{l^w}\left\{[\sigma(\boldsymbol{X}_w^{\mathrm{T}}\boldsymbol{\theta}_{j-1}^w)]^{1-d_j^w}\right.\\&\left.[1-\sigma(\boldsymbol{X}_w^{\mathrm{T}}\boldsymbol{\theta}_{j-1}^w)]^{d_j^w}\right\}=\\&\sum_{w\in C}\sum_{j=2}^{l^w}\{(1-d_j^w)\cdot\lg[\sigma(\boldsymbol{X}_w^{\mathrm{T}}\boldsymbol{\theta}_{j-1}^w)]+\\&d_j^w\cdot\lg[1-\sigma(\boldsymbol{X}_w^{\mathrm{T}}\boldsymbol{\theta}_{j-1}^w)]\}\end{aligned} \tag{5}$$

式(5)中的每一项可以记为:

$$\begin{aligned}L(w,j)=&(1-d_j^w)\cdot\lg[\sigma(\boldsymbol{X}_w^{\mathrm{T}}\boldsymbol{\theta}_{j-1}^w)]+\\&d_j^w\cdot\lg[1-\sigma(\boldsymbol{X}_w^{\mathrm{T}}\boldsymbol{\theta}_{j-1}^w)]\end{aligned} \tag{6}$$

要最大化由多项式之和构成的式(5),可以分别最大化每一项,即式(6)。对每个节点的参数向量 $\boldsymbol{\theta}_{j-1}^w$ 和输出层的输入 $\boldsymbol{X}_w$ 两个参数使用随机梯度法,分别求偏导数得:

$$\begin{aligned}\frac{\partial L(w,j)}{\partial\boldsymbol{\theta}_{j-1}^w}=&\frac{\partial}{\partial\boldsymbol{\theta}_{j-1}^w}\left\{(1-d_j^w)\cdot\lg[\sigma(\boldsymbol{X}_w^{\mathrm{T}}\boldsymbol{\theta}_{j-1}^w)]+\right.\\&\left.d_j^w\cdot\lg[1-\sigma(\boldsymbol{X}_w^{\mathrm{T}}\boldsymbol{\theta}_{j-1}^w)]\right\}\end{aligned} \tag{7}$$

令 $\sigma'(x)=\sigma(x)[1-\sigma(x)]$,代入式(7)可得:

$$\begin{aligned}&(1-d_j^w)[1-\sigma(\boldsymbol{X}_w^{\mathrm{T}}\boldsymbol{\theta}_{j-1}^w)]\boldsymbol{X}_w-d_j^w\sigma(\boldsymbol{X}_w^{\mathrm{T}}\boldsymbol{\theta}_{j-1}^w)\boldsymbol{X}_w=\\&[1-d_j^w-\sigma(\boldsymbol{X}_w^{\mathrm{T}}\boldsymbol{\theta}_{j-1}^w)]\boldsymbol{X}_w\end{aligned} \tag{8}$$

对 $\boldsymbol{\theta}_{j-1}^w$ 进行迭代求值:

$$\boldsymbol{\theta}_{j-1}^w:=\boldsymbol{\theta}_{j-1}^w+\eta[1-d_j^w-\sigma(\boldsymbol{X}_w^{\mathrm{T}}\boldsymbol{\theta}_{j-1}^w)]\boldsymbol{X}_w \tag{9}$$

其中,η 为学习率。由式(6)可知 $\boldsymbol{X}_w$ 和 $\boldsymbol{\theta}_{j-1}^w$ 对称,因此,可以得到关于 $\boldsymbol{X}_w$ 的偏导数为:

$$\frac{\partial L(w,j)}{\partial\boldsymbol{X}_w}=[1-d_j^w-\sigma(\boldsymbol{X}_w^{\mathrm{T}}\boldsymbol{\theta}_{j-1}^w)]\boldsymbol{\theta}_{j-1}^w \tag{10}$$

由于 $\boldsymbol{X}_w$ 是上下文的词向量之和,在处理时将整个更新值应用到上下文每个单词的词向量上。

$$\boldsymbol{v}(\tilde{w}):=\boldsymbol{v}(\tilde{w})+\eta\sum_{j=2}^{l^w}\frac{\partial L(w,j)}{\partial\boldsymbol{X}_w},\tilde{w}\in C(w) \tag{11}$$

其中,$\boldsymbol{v}(\tilde{w})$ 表示上下文单词词向量。

基于上述算法建立模型,将原始语料作为输入,即可实现语料数据的词向量化。

2.2　深度卷积神经网络算法设计

在计算机视觉[4]、自然语言处理[5]等领域,相对于传统神经网络或其他 ML 算法,深度卷积神经网络(Deep Convolutional Neural Networks,DCNNs)具有更高的识别率、更强的鲁棒性以及更好的泛化性能[6]。为此,本文设计多种 DCNNs 算法,构建基于“输入层 + 卷积层 + 卷积层 + 池化层 + 卷积层 + 卷积层 + 池化层 + 全连接层 + 全连接层 + Softmax 层”10 层深度的结构,以实现大数据安全防护检测,并通过模型训练进行大数据智能检测。为减轻梯度消失等问题[7-8],本文选择 Relu 函数作为激活函数,其定义为

$$f(x)=\max(0,x) \tag{12}$$

通过式(13)可以求得卷积层的相应输出值,如下:

$$\boldsymbol{a}_{i,j}=f\left(\sum_{m=0}^{F-1}\sum_{n=0}^{F-1}w_{m,n}x_{i+m,j+n}+w_b\right) \tag{13}$$

其中,$x_{i,j}$ 表示向量的第 i 行第 j 列元素值,$w_{m,n}$ 表示卷积核第 m 行第 n 列的权值,w_b 表示卷积核的偏置项,F 是卷积核的大小(宽度或高度,两者相同)。卷积运算后得到下一特征层,其宽度和高度分别为:

$$\begin{cases}W_2=(W_1-F+2P)/S+1\\H_2=(H_1-F+2P)/S+1\end{cases} \tag{14}$$

其中,W_1 和 H_1 分别表示卷积前向量的宽度和高度,W_2 和 H_2 分别表示卷积后 Feature Map 的宽度和高度,P 表示在向量周围补 0 的圈数值,S 表示卷积运算时的步幅值。卷积前向量的深度可以大于 1,如表示为 D,则相应卷积核的深度也必须为 D,可以求得卷积后的相应输出值。

$$\boldsymbol{a}_{i,j}=f\left(\sum_{d=0}^{D-1}\sum_{m=0}^{F-1}\sum_{n=0}^{F-1}w_{d,m,n}x_{d,i+m,j+n}+w_b\right) \tag{15}$$

对于池化层和全连接层,输出运算相对简单。

在 DCNNs 进行训练时,先利用链式求导计算损失函数对每个权重的梯度,然后根据梯度下降公式更新权重。具体过程如下:

1)对于卷积层误差项的传递,假设步长 S、输入深度 D 和卷积核个数均为 1,它们间的关系如下:

$$\begin{cases}\boldsymbol{N}^l=\mathrm{conv}(\boldsymbol{W}^l,\boldsymbol{a}^{l-1})+\boldsymbol{w}^b\\\boldsymbol{a}_{i,j}^{l-1}=f^{l-1}(\boldsymbol{N}_{i,j}^{l-1})\end{cases} \tag{16}$$

其中,l 表示层数,$\boldsymbol{N}_{i,j}^{l-1}$ 表示加权输入,$\boldsymbol{a}_{i,j}^{l-1}$ 表示输出,conv 表示卷积操作。假设第 l 层中的每个误差项 $\boldsymbol{\delta}^l$ 值已知,根据链式求导法则可得:

$$\boldsymbol{\delta}_{i,j}^{l-1}=\frac{\partial E_d}{\partial \boldsymbol{N}_{i,j}^{l-1}}=\frac{\partial E_d}{\partial \boldsymbol{a}_{i,j}^{l-1}}\times\frac{\partial \boldsymbol{a}_{i,j}^{l-1}}{\partial \boldsymbol{N}_{i,j}^{l-1}} \tag{17}$$

其中，E_d 表示均方误差。通过上述卷积运算过程可知，计算$\frac{\partial E_d}{\partial \boldsymbol{a}_{i,j}^{l-1}}$相当于将第 l 层的 sensitive map 周围补一圈 0 再与 180°度翻转后的卷积核进行交叉相关 cross-correlation 运算：

$$\frac{\partial E_d}{\partial \boldsymbol{a}_{i,j}^{l-1}}=\sum_m\sum_n \boldsymbol{w}_{m,n}^{l}\boldsymbol{\delta}_{i+m,j+n}^{l} \tag{18}$$

由于 $\boldsymbol{a}_{i,j}^{l-1}=f(\boldsymbol{N}_{i,j}^{l-1})$，因此由式(17)和式(18)可得：

$$\boldsymbol{\delta}_{i,j}^{l-1}=\frac{\partial E_d}{\partial \boldsymbol{N}_{i,j}^{l-1}}=\frac{\partial E_d}{\partial \boldsymbol{a}_{i,j}^{l-1}}\times\frac{\partial \boldsymbol{a}_{i,j}^{l-1}}{\partial \boldsymbol{N}_{i,j}^{l-1}}=\sum_m\sum_n \boldsymbol{w}_{m,n}^{l}\boldsymbol{\delta}_{i+m,j+n}^{l}f'(\boldsymbol{N}_{i,j}^{l-1}) \tag{19}$$

可以将式(19)写成如下的卷积形式：

$$\boldsymbol{\delta}^{l-1}=\boldsymbol{\delta}^{l}*\boldsymbol{W}^{l}\circ f'(\boldsymbol{N}^{l-1}) \tag{20}$$

其中，符号∘表示将矩阵中的每个对应元素相乘。当步长 S、输入深度 D 和卷积核个数均大于 1 时，同理可得：

$$\boldsymbol{\delta}^{l-1}=\sum_{d=0}^{D}\boldsymbol{\delta}_d^{l}*W_d^{l}\circ f'(\boldsymbol{N}^{l-1}) \tag{21}$$

2）对于卷积核权重梯度，由于权值共享，根据全导数公式可得：

$$\frac{\partial E_d}{\partial \boldsymbol{w}_{i,j}}=\sum_m\sum_n \boldsymbol{\delta}_{m,n}\boldsymbol{a}_{i+m,j+n}^{l-1} \tag{22}$$

即用 sensitive map 作卷积核，对输入进行交叉相关 cross-correlation 运算。

3）基于上述分析计算，得出偏置项的梯度：

$$\frac{\partial E_d}{\partial \boldsymbol{w}_b}=\sum_i\sum_j \boldsymbol{\delta}_{i,j}^{l} \tag{23}$$

即偏置项的梯度是 sensitive map 所有误差项之和。获得所有的梯度之后，根据梯度下降法可以更新每个权值，从而实现卷积层训练。

对于池化层，一般有 max pooling 和 mean pooling 2 种，它们不用计算梯度，只需将误差项传递到上一层。通过分析可知，对于 max pooling 池化，下一层的误差项会按原值传递到上一层最大值对应的神经元，其他神经元的误差项为 0；对于 mean pooling 池化，下一层的误差项会平均分配到上一层对应区块的所有神经元，即可以用克罗内克积实现池化：

$$\boldsymbol{\delta}^{l-1}=\boldsymbol{\delta}^{l}\otimes\left(\frac{1}{n^2}\right)_{n\times n} \tag{24}$$

其中，n 表示池化层核的大小。

由此，利用已实现的卷积层和池化层结合全连接层，可以堆叠形成 DCNNs[9]，从而完成大数据脚本攻击的智能检测[10]。

3 实验结果与分析

3.1 实验大数据

3.1.1 语料大数据获取

用于本文实验的大数据包括两类[11-12]：正样本大数据（带有攻击行为），利用爬虫工具从网站 http://xssed.com/爬取获得，由 Payload 数据组成；负样本大数据（正常网络请求），为体现特殊性和普遍性，共收集了 2 份数据，一份来自嘉应学院网络中心 2017 年 5 月—12 月的访问日志大数据，另一份是从各网络平台通过网络爬虫获得，它们都是未经处理的语料大数据。

3.1.2 语料大数据处理及向量化

本文利用基于神经网络的词向量化工具连续词袋模型（Continous Bag of Words Model，CBOW）[13]实现大数据语料处理，进行文本切割、清洗、分词、词性标注、去停用词、词向量化，将独热编码的词向量映射为分布形式的词向量，从而降低维数和稀疏性，同时通过求向量间的欧氏距离或夹角余弦值得出任意词间的关联度[14]。具体处理过程如下：

1）首先遍历数据集，将数字都用“0”替换，将 http/、HTTP/、https/、HTTPS 用“http://”替换；其次按照 html 标签、JavaScript 函数体、http://和参数规则进行分词；接着基于日记文档构建词汇表，对单词进行独热编码。

2）构建基于神经网络的词向量化模型，包括输入层、投射层和输出层[15-16]，其结构及训练过程如图 2所示。然后输入样本，最小化损失函数并改变权值，训练模型并获得分布式词向量。

3）统计正样本词集，用词频最高的 3 000 个词构成词库，其他标记为“COM”。本文设定分布式特征向量的维数为 128，当前词与预测词最大窗口距离为 5，含 64 个噪声词，共进行 5 次迭代。

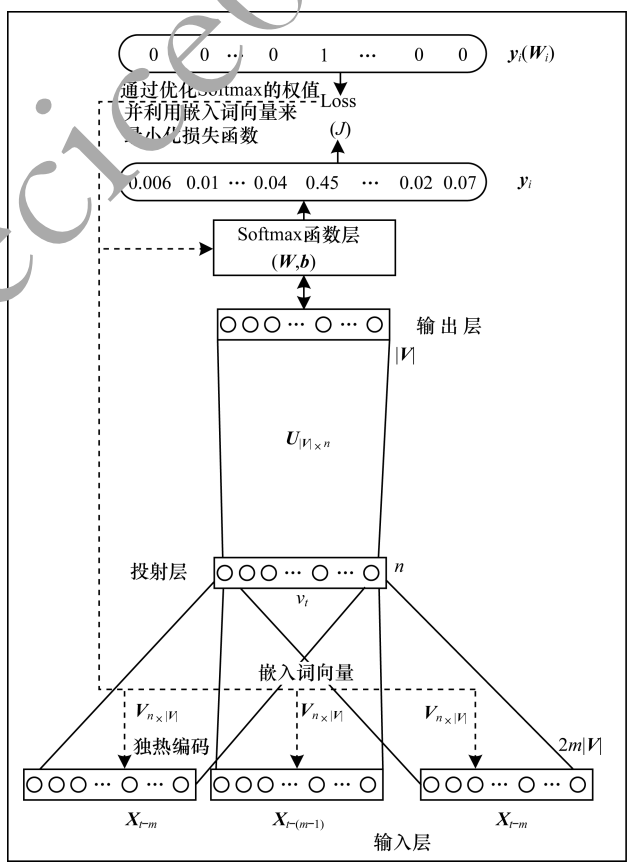


图 2 词向量化 CBOW 模型及训练过程

Fig. 2 Word vectorized CBOW model and training process

因为每条数据所占字符长度不同,所以本文以所占字符长度最大为标准,不足则以 -1 填充,在为数据集设计标签时,使用独热编码,正样本标签(即属于攻击样本)以 1 表示,负样本标签(即正常网络请求)以 0 表示。经过上述处理之后,共获得正样本数据集 40 637 条,负样本数据集分别为 105 912 条和 200 129 条,它们数量大、计算复杂性高,均为大数据[17-19]。为提高训练效果,将正样本集和两类负样本集分别进行合并,随机划分为训练集和测试集,数量比为 7∶3,并记为第Ⅰ类大数据集和第Ⅱ类大数据集。

3.2　实验检测与结果

为验证算法的有效性,设计多种 DCNNs 算法[20],构建基于"输入层 + 卷积层 + 卷积层 + 池化层 + 卷积层 + 卷积层 + 池化层 + 全连接层 + 全连接层 + Softmax 层"10 层深度的结构,并设计不同的超参数,包括样本块大小、学习率 μ 以及卷积层深度等[21-22],然后输入大数据集词向量样本进行训练和测试。为检验系统的稳定性,对每类数据分别进行 20 次实验,结果及分析如下:

1)基于各深层 DCNNs 设计不同的超参数,学习率 μ 为 0.001、0.01 和 0.1,对第Ⅰ类大数据集进行 20 次实验得到的识别率结果如表 1 所示。

表 1　各深层 DCNNs 基于不同学习率对第Ⅰ类大数据集的识别率结果

Table 1　Recognition rate results of each deep DCNNs for type Ⅰ big dataset based on different learning rates

实验次数	识别率		
	μ 为 0.001	μ 为 0.01	μ 为 0.1
1	0.981 4	0.993 4	0.831 3
2	0.991 2	0.994 2	0.830 8
3	0.992 0	0.994 2	0.831 0
4	0.992 8	0.994 5	0.830 9
5	0.993 5	0.994 8	0.830 5
6	0.993 6	0.994 9	0.830 5
7	0.994 0	0.995 1	0.830 6
8	0.994 0	0.994 9	0.830 3
9	0.994 3	0.995 3	0.830 1
10	0.994 5	0.992 2	0.830 1
11	0.994 5	0.994 9	0.830 0
12	0.994 7	0.994 7	0.830 4
13	0.995 0	0.994 8	0.830 7
14	0.994 9	0.994 8	0.831 4
15	0.994 7	0.994 9	0.829 8
16	0.995 5	0.994 7	0.826 9
17	0.995 8	0.994 8	0.829 4
18	0.995 8	0.994 9	0.830 1
19	0.995 3	0.995 0	0.829 7
20	0.995 8	0.995 0	0.829 8

从表 1 可以看出,当学习率为 0.001 和 0.01 时,算法都有很高的识别率,其中,最低识别率为 0.981 4,最高识别率为 0.995 8,且识别率随着训练次数的增加一直保持较高水平并趋于稳定;而当学习率为 0.1 时,算法识别率稍低,平均为0.830 2 左右,原因是学习率设置过大,导致训练时梯度下降过快从而越过了最优值,相比而言,当学习率较小时能得到全局最优或接近最优。识别率的曲线图表示如图 3 所示。

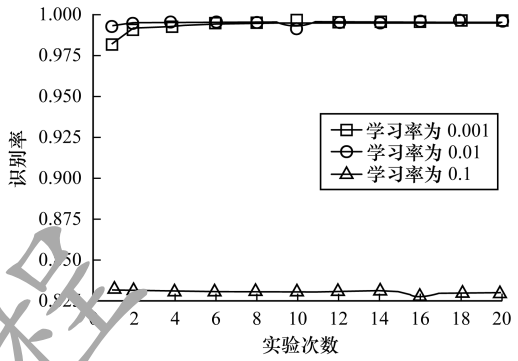


图 3　基于不同学习率对第Ⅰ类大数据集进行 20 次实验得到的识别率曲线

Fig. 3　Recognition rate curve obtained from 20 experiments for type Ⅰ big dataset based on different learning rates

2)基于不同的学习率,对第Ⅱ类大数据集进行 20 次实验得到的识别率结果如表 2 所示。从表 2 可以看出,当学习率为 0.001 时,算法一直有稳定的高识别率,学习率为 0.01 时除了中间几次稍低外其他都是高识别率,且识别率随着训练次数的增加总体上都保持增长状态并趋于稳定;同样当学习率为 0.1 时,识别率相对更低,平均为 0.831 0 左右。识别率的曲线图表示如图 4 所示。

表 2　各深层 DCNNs 基于不同学习率对第Ⅱ类大数据集的识别率结果

Table 2　Recognition rate results of each deep DCNNs for type Ⅱ big dataset based on different learning rates

实验次数	识别率		
	μ 为 0.001	μ 为 0.01	μ 为 0.1
1	0.980 8	0.983 3	0.830 8
2	0.996 7	0.994 8	0.830 9
3	0.997 3	0.993 6	0.831 4
4	0.997 6	0.985 7	0.830 9
5	0.997 7	0.975 3	0.831 0
6	0.997 9	0.988 3	0.831 0
7	0.998 1	0.989 1	0.831 1
8	0.998 2	0.989 2	0.830 9
9	0.998 4	0.989 4	0.831 1
10	0.998 3	0.947 0	0.831 1
11	0.998 5	0.836 2	0.830 8
12	0.998 5	0.824 8	0.831 1
13	0.998 7	0.806 8	0.831 3
14	0.998 8	0.898 4	0.831 1
15	0.998 7	0.924 9	0.831 2
16	0.998 7	0.926 6	0.831 0
17	0.998 8	0.926 7	0.830 9
18	0.999 0	0.926 5	0.831 2
19	0.998 8	0.927 3	0.831 1
20	0.998 6	0.941 2	0.831 0

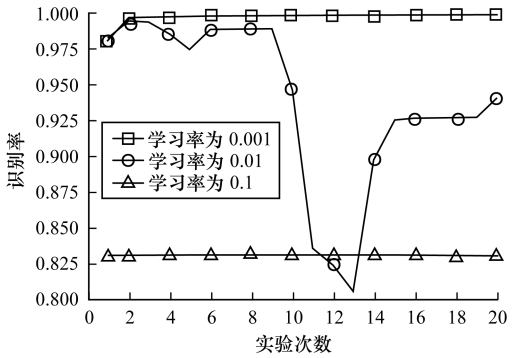


图 4 基于不同学习率对第Ⅱ类大数据集进行 20 次实验得到的识别率曲线

Fig. 4 Recognition rate curve obtained from 20 experiments for type II big dataset based on different learning rates

3)基于各深层 DCNNs 设计不同的超参数,样本块大小(BatchSize)分别为 50、100 和 500,对第Ⅰ类大数据集进行 20 次实验得到的识别率结果如表 3所示。从表 3 可以看出,BatchSize 为 100 和 500 时算法都有很高的识别率,其中,最低识别率为 0.988 6,最高识别率为 0.995 5,且随着训练次数的增加都保持稳定的高识别率;当 BatchSize 为 50 时,识别率相对较低,平均为 0.734 5。识别率的曲线图表示如图 5 所示。

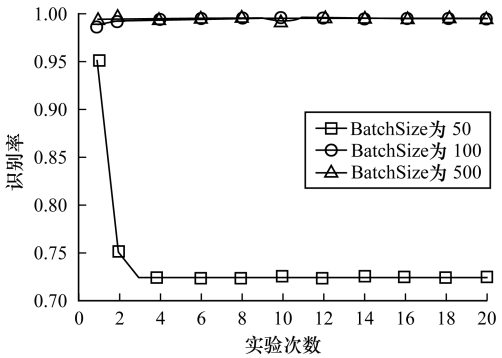


图 5 基于不同 BatchSize 对第Ⅰ类大数据集进行 20 次实验得到的识别率曲线

Fig. 5 Recognition rate curve obtained from 20 experiments for type I big dataset based on different BatchSizes

4)基于不同的 BatchSize,对第Ⅱ类大数据集进行 20 次实验得到的识别率结果如表 4 所示。从表 4 可以看出,当 BatchSize 为 500 时算法有最好的平均识别率,其中,最低识别率为 0.806 8,最高识别率为 0.994 8;当 BatchSize 为 50 时,识别率稍有下降,平均值约为 0.832 0;当 BatchSize 为 100 时,前 6 次识别率均接近 0.912 1,之后下降幅度较大,仅为0.169 0 左右。识别率的曲线图表示如图 6 所示。

表 3 各深层 DCNNs 基于不同 BatchSize 对第Ⅰ类大数据集的识别率结果

Table 3 Recognition rate results of each deep DCNNs for type I big dataset based on different BatchSizes

实验次数	识别率		
	BatchSize 为 50	BatchSize 为 100	BatchSize 为 500
1	0.951 1	0.988 6	0.993 4
2	0.748 9	0.991 6	0.994 2
3	0.721 7	0.992 1	0.994 2
4	0.721 7	0.992 6	0.994 5
5	0.721 7	0.993 1	0.994 8
6	0.721 7	0.993 4	0.994 9
7	0.721 7	0.993 9	0.995 1
8	0.721 7	0.993 9	0.994 9
9	0.721 7	0.994 2	0.995 3
10	0.721 7	0.994 4	0.992 2
11	0.721 7	0.994 6	0.994 9
12	0.721 8	0.994 5	0.994 7
13	0.721 7	0.994 6	0.994 8
14	0.721 6	0.994 9	0.994 8
15	0.721 7	0.995 1	0.994 9
16	0.721 7	0.995 3	0.994 7
17	0.721 7	0.995 0	0.994 8
18	0.721 7	0.995 4	0.994 9
19	0.721 7	0.995 2	0.995 0
20	0.721 7	0.995 5	0.995 0

表 4 各深层 DCNNs 基于不同 BatchSize 对第Ⅱ类大数据集的识别率结果

Table 4 Recognition rate results of each deep DCNNs for type II big dataset based on different BatchSizes

实验次数	识别率		
	BatchSize 为 50	BatchSize 为 100	BatchSize 为 500
1	0.850 8	0.908 8	0.983 3
2	0.830 9	0.917 8	0.994 8
3	0.831 4	0.912 1	0.993 6
4	0.830 9	0.909 5	0.985 7
5	0.831 0	0.909 5	0.975 3
6	0.831 0	0.899 9	0.988 3
7	0.831 1	0.168 9	0.989 1
8	0.830 9	0.169 1	0.989 2
9	0.831 1	0.168 9	0.989 4
10	0.831 1	0.168 9	0.947 0
11	0.830 8	0.169 2	0.836 2
12	0.831 1	0.168 9	0.824 8
13	0.831 3	0.168 7	0.806 8
14	0.831 1	0.168 9	0.898 4
15	0.831 2	0.168 8	0.924 9
16	0.831 0	0.169 0	0.926 6
17	0.830 9	0.169 1	0.926 7
18	0.831 2	0.168 8	0.926 6
19	0.831 1	0.168 9	0.927 3
20	0.831 0	0.169 0	0.941 2

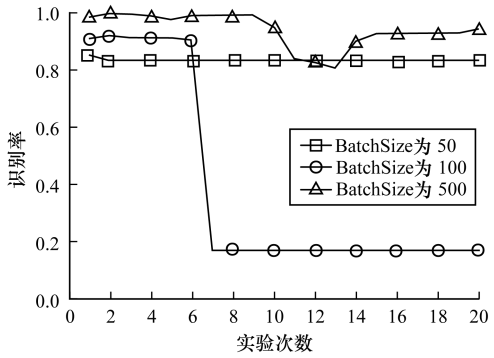


图 6　基于不同 BatchSize 对第Ⅱ类大数据集进行 20 次实验得到的识别率曲线

Fig. 6　Recognition rate curve obtained from 20 experiments for type II big dataset based on different BatchSizes

5)为进一步验证系统的相关特性,基于各深层 DCNNs 设计不同的卷积层深度,对第Ⅰ类大数据集进行 20 次实验得到的识别率结果如表 5 所示。从表 5 可以看出,算法总体都保持高识别状态,其中,最低识别率为 0.976 0,最高识别率为 0.993 0。识别率的曲线图表示如图 7 所示。

表 5　深层 DCNNs 对第Ⅰ类大数据集的识别率结果

Table 5　Recognition rate results of deep DCNNs on type I big dataset

实验次数	识别率	实验次数	识别率
1	0.990 0	11	0.981 8
2	0.992 0	12	0.981 9
3	0.993 0	13	0.981 9
4	0.976 0	14	0.981 9
5	0.979 0	15	0.981 9
6	0.980 0	16	0.981 9
7	0.981 0	17	0.981 9
8	0.981 4	18	0.982 0
9	0.981 6	19	0.982 0
10	0.981 7	20	0.982 0

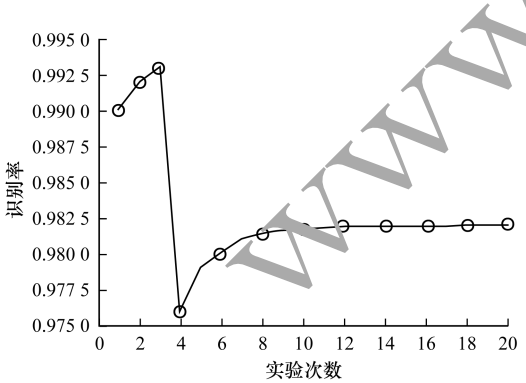


图 7　DCNNs 对第Ⅰ类大数据集进行 20 次实验得到的识别率曲线

Fig. 7　Recognition rate curve obtained from 20 experiments of DCNNs for type I big dataset

6)基于不同的卷积层深度,对第Ⅱ类大数据集进行 20 次实验得到的识别率结果如表 6 所示。从表 6 可以看出,随着训练的进行,算法识别率不断提高,其中,最低识别率为 0.980 8,最高识别率为0.999 0,最后趋于稳定。识别率的曲线图表示如图 8所示。

表 6　深层 DCNNs 对第Ⅱ类大数据集的识别率结果

Table 6　Recognition rate results of deep DCNNs for type II big dataset

实验次数	识别率	实验次数	识别率
1	0.980 8	11	0.998 5
2	0.996 7	12	0.998 5
3	0.997 3	13	0.998 7
4	0.997 6	14	0.998 8
5	0.997 7	15	0.998 7
6	0.997 9	16	0.998 7
7	0.998 1	17	0.998 8
8	0.998 2	18	0.999 0
9	0.998 4	19	0.998 8
10	0.998 3	20	0.998 6

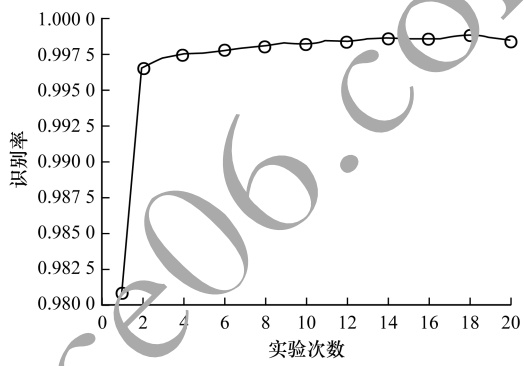


图 8　DCNNs 对第Ⅱ类大数据集进行 20 次实验得到的识别率曲线

Fig. 8　Recognition rate curve obtained from 20 experiments of DCNNs for type II big dataset

通过实验可以看出,各深层 DCNNs 基于不同学习率 μ 对第Ⅰ类大数据集的平均识别率为 99.366 5%,方差为 0.000 001,标准差为 0.000 944,如表 7 所示。各深层 DCNNs 基于不同学习率 μ 对第Ⅱ类大数据集的平均识别率为 93.875 5%,方差为 0.000 015,标准差为 0.003 952,如表 8 所示。通过实验也可以看出,各深层 DCNNs 基于不同 BatchSize 对第Ⅰ类大数据集的平均识别率为 99.389 5%,方差为 0.000 003,标准差为 0.001 670,如表 9 所示。各深层 DCNNs 基于不同 BatchSize 对第Ⅱ类大数据集的平均识别率为 83.204 5%,方差为 0.003 258,标准差为 0.058 559,如表 10 所示。另外,通过实验可以得到,各深层 DCNNs 基于不同卷积层深度对第Ⅰ类大数据集的平均识别率为 98.274 5%,方差为 0.000 016,标准差为 0.004 133,如表 11 所示。各深层 DCNNs 基于不同卷积层深度对第Ⅱ类大数据集的平均识别率为 99.740 5%,方差为 0.000 015,标准差为 0.003 952,如表 12 所示。

表 7 各深层 DCNNs 基于不同学习率对第Ⅰ类大数据集的平均识别率、方差和标准差

Table 7 Average recognition rate, variance and standard deviation of each deep DCNNs for type I big dataset based on different learning rates

学习率	平均识别率/%	方差	标准差
0.001	99.366 5	0.000 009	0.003 139
0.010	99.460 0	0.000 000	0.000 697
0.100	83.021 5	0.000 001	0.000 944

表 8 各深层 DCNNs 基于不同学习率对第Ⅱ类大数据集的平均识别率、方差和标准差

Table 8 Average recognition rate, variance and standard deviation of each deep DCNNs for type II big dataset based on different learning rates

学习率	平均识别率/%	方差	标准差
0.001	99.740 5	0.000 015	0.003 952
0.010	93.875 5	0.003 258	0.058 560
0.100	83.104 5	0.000 000	0.000 157

表 9 各深层 DCNNs 基于不同 BatchSize 对第Ⅰ类大数据集的平均识别率、方差和标准差

Table 9 Average recognition rate, variance and standard deviation of each deep DCNNs for type I big dataset based on different BatchSizes

BatchSize	平均识别率/%	方差	标准差
50	73.453 0	0.002 504	0.051 336
100	99.389 5	0.000 003	0.001 670
500	99.460 0	0.000 000	0.000 697

表 10 各深层 DCNNs 基于不同 BatchSize 对第Ⅱ类大数据集的平均识别率、方差和标准差

Table 10 Average recognition rate, variance and standard deviation of each deep DCNNs for type II big dataset based on different BatchSizes

BatchSize	平均识别率/%	方差	标准差
50	83.204 5	0.000 019	0.004 417
100	39.113 5	0.115 211	0.348 245
500	93.876 0	0.003 258	0.058 559

表 11 深层 DCNNs 对第Ⅰ类大数据集的平均识别率、方差和标准差

Table 11 Average recognition rate, variance and standard deviation of deep DCNNs for type I big dataset

卷积深度	平均识别率/%	方差	标准差
10	98.274 5	0.000 016	0.004 133

表 12 深层 DCNNs 对第Ⅱ类大数据集的平均识别率、方差和标准差

Table 12 Average recognition rate, variance and standard deviation of deep DCNNs for type II big dataset

卷积深度	平均识别率/%	方差	标准差
10	99.740 5	0.000 015	0.003 952

对于第Ⅰ类和第Ⅱ类大数据集，基于不同学习率 μ 的识别率均值如图 9 所示，标准差均值如图 10 所示。对于第Ⅰ类和第Ⅱ类大数据集，基于不同 BatchSize 的识别率均值如图 11 所示，标准差均值如图 12 所示。

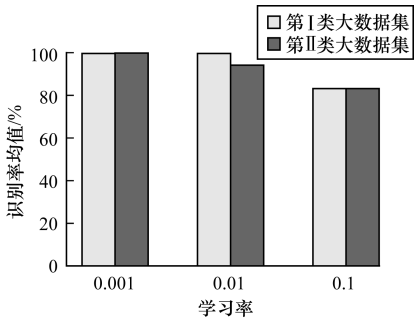


图 9 基于不同学习率对第Ⅰ类和第Ⅱ类大数据集的识别率均值

Fig. 9 Average recognition rate for type I and type II big datasets based on different learning rates

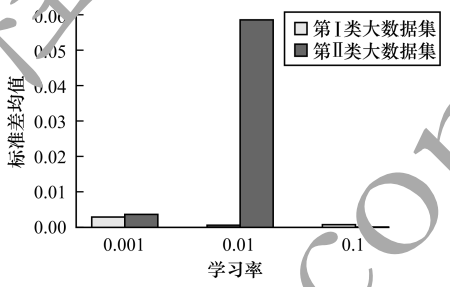


图 10 基于不同学习率对第Ⅰ类和第Ⅱ类大数据集的标准差均值

Fig. 10 Mean standard deviation for type I and type II big datasets based on different learning rates

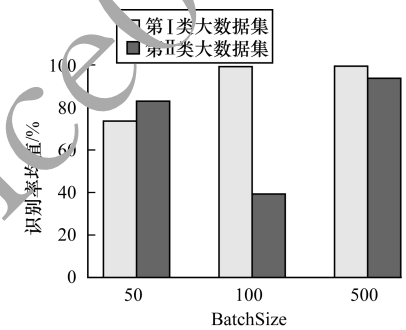


图 11 基于不同 BatchSize 对第Ⅰ类和第Ⅱ类大数据集的识别率均值

Fig. 11 Average recognition rate for type I and type II big datasets based on different BatchSizes

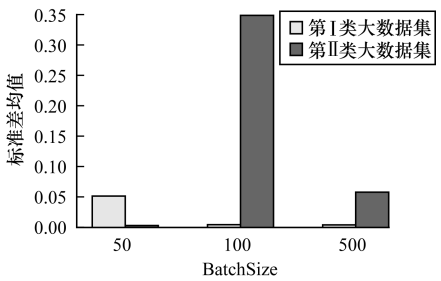


图 12 基于不同 BatchSize 对第Ⅰ类和第Ⅱ类大数据集的标准差均值

Fig. 12 Mean standard deviation for type I and type II big datasets based on different BatchSizes

系统识别率变化过程曲线图如图 13 所示，可以

看出,识别率随着训练的进行逐渐提高,随后降低然后又不断提高并趋于稳定,总体识别率较高。损失函数误差变化曲线图如图 14 所示,可以看出,随着训练的进行,损失函数误差先减少后增加然后不断地减小并趋于稳定,其与识别率的变化过程相一致。词向量样本余弦距离变化曲线图如图 15 所示,可以看出,随着训练的进行,余弦距离先减小后增加然后不断地减小并趋于稳定,这反映了词向量样本的相关性先增强后变小然后越来越强,其同识别率变化过程也一致。平均绝对误差变化过程曲线图如图 16 所示,可以看出,随着训练的进行,平均绝对误差先减小后增加然后不断地减小并趋于最小的稳定值。

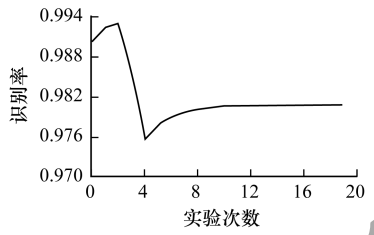


图 13　识别率变化曲线

Fig. 13　Curve of recognition rate change

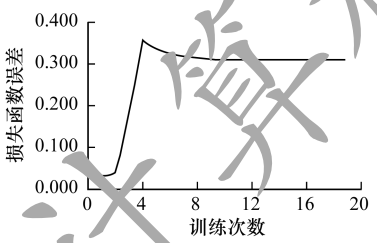


图 14　损失函数误差变化曲线

Fig. 14　Curve of loss function error change

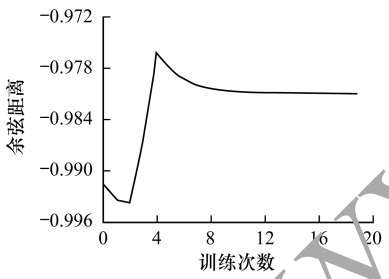


图 15　余弦距离变化曲线

Fig. 15　Curve of cosine distance change

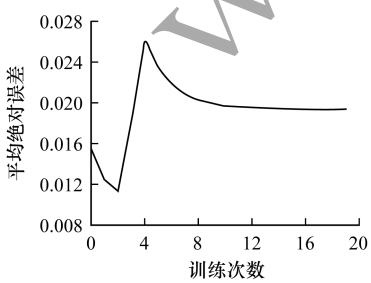


图 16　平均绝对误差变化曲线

Fig. 16　Curve of average absolute error change

4　结束语

传统的计算机病毒检测方法主要依据病毒特征库中的已有特征,通过特征匹配来确定病毒是否存在,其难以检测新出现的病毒以及变形病毒,而且检测效率较低。为解决该问题,本文通过类图像处理及向量化方法对访问流量语料库大数据进行词向量化处理,从而实现了面向大数据跨站脚本攻击的智能检测。实验结果表明,该方法具有识别率高、稳定性好、总体性能优良等优点。下一步将探讨小样本数据的智能检测问题,以更全面地进行入侵智能检测。

参考文献

[1] NAIR A. Prevention of cross site scripting and securing Web application atclient side[EB/OL]. [2018-11-20]. http://www.ijaerd.com/papers/special_papers/IJAERDE021.pdf.

[2] GERMÁN E R, BENAVIDES D E, TORRES J, et al. Cookie scout: an analytic model for prevention of cross-site scripting using a cookie classifier[C]//Proceedings of International Conference on Information Theoretic Security. Berlin, Germany: Springer, 2018: 497-507.

[3] WANG Guihua, QIN Xiangqing, CHEN Li, et al. A query recommendation algorithm for professional search engines[J]. Computer Engineering and Applications, 2013, 49(9): 144-149. (in Chinese)
王桂华,秦湘清,陈莉,等. 一种面向专业搜索引擎的查询推荐算法[J]. 计算机工程与应用,2013,49(9):144-149.

[4] ZHANG Haijun, ZHANG Nan, XIAO Nanfeng. Fire detection and identification method based on visual attention mechanism[J]. Optik, 2015, 126: 5011-5018.

[5] LI N Quan, LIANG Bin, XU Jin, et al. A deep hierarchical neural network model for aspect-based sentiment analysis[J]. Chinese Journal of Computers, 2018, 41(12): 2637-2652. (in Chinese)
刘全,梁斌,徐进,等. 一种用于基于方面情感分析的深度分层网络模型[J]. 计算机学报,2018,41(12):2637-2652.

[6] REHMAN Y A U, MAN P L, LIU M. LiveNet: improving features generalization for face liveness detection using convolution neural networks[J]. Expert Systems with Applications, 2018, 108: 159-169.

[7] ZHU Z A, LI Y Z, SONG Z. On the convergence rate of training recurrent neural networks[EB/OL]. [2018-11-20]. https://www.microsoft.com/en-us/research/uploads/prod/2018/12/on-the-convergence-rate.pdf.

[8] DU S S, ZHAI X, POCZOS B, et al. Gradient descent provably optimizes over-parameterized neural networks [EB/OL]. [2018-11-20]. https://arxiv.org/abs/1810.02054? context=cs.LG.

[9] LEVINE S, PASTOR P, KRIZHEVSKY A, et al. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection[J]. International Journal of Robotics Research, 2016 (10): 421-436.

(下转第 143 页)

(上接第 137 页)

[10] WANG Lei, ZHOU Qing, HE Dongjie, et al. Multi-source taint analysis technique for privacy leak detection of Android Apps[J]. Journal of Software, 2019, 30(2): 211-230. (in Chinese)
王蕾,周卿,何冬杰,等. 面向 Android 应用隐私泄露检测的多源污点分析技术[J]. 软件学报, 2019, 30(2): 211-230.

[11] GAO Yanjun, ZHANG Xueying, LI Fenglian, et al. Data allocation algorithm for large data with all-to-all comparison based on graph covering[J]. Computer Engineering, 2018, 44(4): 17-22, 27. (in Chinese)
高燕军,张雪英,李凤莲,等. 基于图覆盖的大数据全比较数据分配算法[J]. 计算机工程, 2018, 44(4): 17-22, 27.

[12] KWON O, LEE N, SHIN B. Data quality management, data usage experience and acquisition intention of big data analytics[J]. International Journal of Information Management, 2014, 34(3): 387-394.

[13] CHEN Q, SOKOLOVA M. Word2Vec and Doc2Vec in unsupervised sentiment analysis of clinical discharge summaries[EB/OL]. [2018-11-20]. https://arxiv.org/ftp/arxiv/papers/1805/1805.00352.pdf.

[14] Deep learning for Java[EB/OL]. [2018-11-20]. https://deeplearning4j.org/.

[15] CELESTI F, CELESTI A, WAN J, et al. Why deep learning is changing the way to approach NGS data processing: a review[J]. IEEE Reviews in Biomedical Engineering, 2018, 11: 68-76.

[16] SUNDHI G, ADAM K, RAGHU M. Learning one convolutional layer with overlapping patches[EB/OL]. [2018-11-20]. http://proceedings.mlr.press/v80/goel18a/goel18a.pdf.

[17] YU Yanwei, JIA Zhaofei, CAO Lei, et al. Fast density-based clustering algorithm for location big data[J]. Journal of Software, 2018, 29(8): 2470-2484. (in Chinese)
于彦伟,贾召飞,曹磊,等. 面向位置大数据的快速密度聚类算法[J]. 软件学报, 2018, 29(8): 2470-2484.

[18] ZHANG Haijun, XIAO Nanfeng. Parallel implementation of multilayered neural networks based on map-reduce on cloud computing clusters[J]. Soft Computing, 2016, 20(4): 1471-1483.

[19] TRIGUERO I, PERALTA D, BACARDIT J, et al. MRPR: a MapReduce solution for prototype reduction in big data classification[J]. Neurocomputing, 2015, 150: 331-345.

[20] DU S S, LEE J D, TIAN Y, et al. Gradient descent learns one-hidden-layer CNN: don't be afraid of spurious local minima[EB/OL]. [2018-11-20]. https://arxiv.org/pdf/1712.00779.pdf.

[21] JAEGUL C, SHIXIA L. Visual analytics for explainable deep learning[J]. IEEE Computer Graphics and Applications, 2018, 38(4): 84-92.

[22] MICHAËL G, CHEN J, BARRON J T, et al. Deep bilateral learning for real-time image enhancement[J]. ACM Transactions on Graphics, 2017, 36(4): 118-120.

编辑　吴云芳