计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.181202455

# 基于 CNN 的恶意 Web 请求检测技术

崔艳鹏[1,2]　刘　咪[1]　胡建伟[1,2]
1 西安电子科技大学网络与信息安全学院　西安 710071
2 西安电子科技大学网络行为研究中心　西安 710071

**摘　要**　目前,基于卷积神经网络的 Web 恶意请求检测技术领域内只有针对 URL 部分进行恶意检测的研究,并且各研究对原始数据的数字化表示方法不同,这会造成检测效率和检测准确率较低。为提高卷积神经网络在 Web 恶意请求检测领域的性能,在现有工作的基础上将其他多个 HTTP 请求参数与 URL 合并,将数据集 HTTP data set CSIC 2010 和 DEV_ACCESS 作为原始数据,设计对比实验。首先采用 6 种数据数字向量化方法对字符串格式的原始输入进行处理;然后将其分别输入所设计的卷积神经网络,训练后可得到 6 个不同的模型,同时使用相同的训练数据集对经典算法 HMM,SVM 和 RNN 进行训练,得到对照组模型;最后在同一验证集上对 9 个模型进行评估。实验结果表明,采用多参数的 Web 恶意请求检测方法将词汇表映射与卷积神经网络内部嵌入层相结合对原始数据进行表示,可使卷积神经网络取得 99.87%的准确率和 98.92%的 F1 值。相比其他 8 个模型,所提方法在准确率上提升了 0.4～7.7 个百分点,在 F1 值上提升了 0.3～13 个百分点。实验充分说明,基于卷积神经网络的多参数 Web 恶意请求检测技术具有明显的优势,且使用词汇表映射和网络内部嵌入层对原始数据进行处理能使该模型取得最佳的检测效果。

**关键词**:卷积神经网络;深度学习;Web 安全;恶意 Web 请求检测
**中图法分类号**　TP183

## Malicious Web Request Detection Technology Based on CNN

CUI Yan-peng[1,2],LIU Mi[1] and HU Jian-wei[1,2]
1 School of Cyber Engineering,Xidian University,Xi'an 710071,China
2 Network Behavior Research Center,Xidian University,Xi'an 710071,China

**Abstract**　At present,in the field of Web malicious requests detection technology based on convolutional neural network,malicious requests are detected only for the URL part,and each research has different digital representation methods for the original data,which will result in low detection efficiency and detection accuracy. In order to improve the performance of the convolutional neural network in web malicious request detection,this paper introduced other HTTP request parameters to be merged with URLs,and used the dataset HTTP data set CSIC 2010 and DEV_ACCESS as the raw data. The comparative experiment first used six digital representation methods to represent the raw input of the string format,and then put them to the designed convolutional neural network to obtain six different models. At the same time,the classical algorithms HMM,SVM and RNN were trained on the same training data set to obtain the control models. Finally,the nine models were evaluated on the same test data set. The experimental results show that in the multi-parameter Web malicious request detection method,the convolutional neural network using the combination of the vocabulary mapping and the internal embedding layer to represent the original data achieves 99.87% accuracy and 98.92.% F1 score,therefore,the accuracy is improved by 0.4～7.7 percentage points and the F1 value is improved by 0.3～13 percentage points. The experiment fully demonstrate that the multi-parameter Web malicious request detection technology based on convolutional neural network has obvious advantages,and using the vocabulary mapping and the internal embedding layer of the network to represent the original data can make the model achieve the best detection performance.

**Keywords**　Convolutional neural network,Deep learning,Web security,Malicious Web request detection

## 1　引言

随着 Web 应用技术的迅猛发展,Web 攻击对个人和企业的影响更加严重,同时 Web 攻击也变得更加复杂且难以检测,因此对恶意 Web 请求检测技术的研究显得尤为重要。在此情况下,Web 服务端若能在处理用户请求前对这些请求进

到稿日期:2018-12-31　返修日期:2019-03-22　本文已加入开放科学计划(OSID),请扫描上方二维码获取补充信息。
通信作者:崔艳鹏(ypcui@mail.xidian.edu.cn)

行恶意检测，则能在很大程度上对 Web 应用起到保护作用。

随着大数据时代的到来和 Web 应用的复杂化，Web 攻击方法也变得多元化和复杂化。基于规则的传统恶意检测技术由于需要人工维护大量的规则库而十分低效；同时，这种方法需要经过专家分析，受限于人力资本，往往不能在第一时间感知到新的攻击方法。在这种情况下，基于机器学习技术的恶意 Web 请求检测技术成为了研究和开发的热点，其效率和准确度都有所提升，并且可以检测出新类型的攻击，因此受到 Web 安全领域专家的重视。

深度学习是机器学习的一个分支。近年来，随着深度学习的飞速发展，基于各类深度学习神经网络的 Web 恶意检测技术成为了研究热点。2015 年，Atienza 等从可视化的角度利用神经网络来分析 HTTP 数据，并试图检测 Web 攻击[1]。2017 年，Zhang 等提出了一个对恶意 URL 进行分类的单词级 CNN 网络[2]。Saxe 等提出了一个字符级 CNN 网络 Explosure 用于对恶意 URL、文件路径和注册表键进行检测[3]。Adam 等提出了使用 RNN 实现恶意 Web 请求检测的 LSTM 神经网络[4]。2018 年，Le 等提出了一个字符级的卷积神经网络(Convolutional Neural Network，CNN) URLNet，用于对恶意 URL 进行检测[5]。Chen 等提出了一种端到端的可训练的树形深度神经网络 TSDNN[6]。

基于神经网络的恶意 Web 请求检测技术大都取得了令人满意的结果。其中，CNN 由于具有强大的分类能力，因此基于 CNN 的方法取得的效果更为突出。CNN 是一种前馈神经网络，在图像处理问题上取得了突破性成果[7-8]，之后在语音识别和文本分类等领域也有出色的表现。自 2014 年 Kim 提出使用 CNN 进行文本分类后[9]，CNN 在句子分类[10-11]、情感分析[12-13]等方面的研究成果层出不穷，Web 安全研究人员也开始使用 CNN 分类方法对恶意 Web 请求检测进行研究。在该问题上，CNN 由于权值共享，需要考量的参数更少，且不用手动提取特征，因而成为了一种更具有吸引力的分类网络。因此，本文对基于 CNN 的恶意 Web 请求检测技术进行了进一步的研究和完善。

首先，已有工作都只对 Web 请求中的 URL 进行了检测。虽然大部分 Web 请求的恶意载荷都位于 URL 及 URL 参数，但位于其他 HTTP 参数的 Web 攻击也无法忽视。同时，恶意 HTTP 请求的参数之间必然存在一定的关联性，因此对整个 HTTP 请求进行建模显然比检测恶意 URL 能更加准确、全面地识别出恶意攻击。其次，已有工作各自采用了不同的方法来对原始输入数据进行数字化表示，无法得出最佳表示方法。

为解决这些问题，本文在利用 CNN 检测恶意 URL 的基础上，将 HTTP 请求中的 URL 和其他一些 Web 请求参数合并后作为 CNN 网络的输入，这些参数包括 content-length，timestamp，referer，remoteAdress，responsePayload，method。同时，本文参考了文本分析、情感分析、恶意软件检测等基于 CNN 的文本分类的相关文章，采用了 6 种表示方法对原始数据进行字符级别的离散向量化处理来进行对比实验。实验结果表明，本文方法取得了最高的精确度 99.87%，并且 CNN 输入层数据的最佳处理方法为词汇表映射与 CNN 网络内部嵌入层的结合。

## 2 相关工作

### 2.1 卷积神经网络

#### 2.1.1 卷积神经网络的分类原理

CNN 最初是为解决图像处理问题而提出的算法，它受到动物眼睛工作方式的启发，在图像处理问题上取得突破性成果后，在语音识别和文本分类等领域也都有出色表现。

在图像识别的工作中，CNN 输入图像是由像素点组成的二维或三维数字矩阵，3 个维度分别是图像宽度、图像高度和 RGB 三色通道。该输入保留了图像原有的空间结构，不会丢失任何信息。

CNN 通过使用一系列卷积核来对输入矩阵进行特征提取。每个卷积核即一个数字矩阵，该卷积核在整个图像上进行滑动窗口移位，所有像素至少被覆盖一次。卷积核每覆盖到输入矩阵的一部分，会计算该部分区域和卷积核的权重矩阵之间的点积。设输入为一个 $l\times h$ 的二维矩阵 $\boldsymbol{X}$，卷积核 $\boldsymbol{W}$ 为 $k\times k$ 的矩阵，当卷积核覆盖到 $\boldsymbol{X}$ 的局部 $X_{ij}$ 时，得到的特征图 $C_{ij}$ 可表示为：

$$C_{ij}=\boldsymbol{W}\cdot X_{ij}+b \tag{1}$$

其中，$b$ 为计算过程中的常量偏置。最终得到的特征图 $\boldsymbol{C}$ 为由组 $C_{ij}$ 成的矩阵，该矩阵的大小 $G$ 为：

$$G=(\frac{l-k}{s}+1)\times(\frac{h-k}{s}+1) \tag{2}$$

其中，$s$ 为卷积核移动的步长。

这个过程可以理解为卷积核在输入矩阵上滑动搜索特定的特征，这种行为类似于从原始图像矩阵中提取特定信息的图像滤波器，因此在 CNN 中卷积核也被称为滤波器。在开始训练网络时，这些卷积核的参数是完全随机的，不能检测出任何特征，每个卷积核的参数都是在训练过程中通过反向传播算法进行逐步优化得到的。

CNN 的训练过程需要定义一个损失函数和一个优化器。损失函数(Loss Function)用于估量模型的预测值 $f(x)$ 与真实值 $Y$ 的不一致程度，它是一个非负实值函数，损失函数越小，模型的鲁棒性就越好。在训练过程中，使用损失函数计算网络输出的损失，并计算误差梯度，随后该误差会被反向传播用以更新网络的相关参数，从而使得卷积核能够提取对应的特征。常用的损失函数有 Softmax Cross Entropy Loss 和 Categorical Cross Entropy 等。优化器的作用是采用优化算法更新网络参数，以最小化损失函数。CNN 中常用的优化器有 ADAM[14]，SDG，Momentum 等。

上述卷积过程是 CNN 网络的核心部分，但是该过程只能从原始图像中提取特征并减少参数的数量。为了生成最终输出，还需要全连接层来完成分类和结果输出。

#### 2.1.2 卷积神经网络的网络结构

CNN 是由多个层构成的深度学习网络，每个层又由多个神经元组成。

1）输入层：输入为 $n$ 维的数字向量，其输出作为嵌入层的输入。

2）嵌入层：嵌入即将输入矩阵映射为固定大小的向量。在 CNN 网络中，嵌入层可以随着整个网络的训练更新来优化其表示参数，从而得到更好的数据表示结果。

3）卷积层：卷积层用来对输入进行特征提取，是 CNN 网络的核心。该层一般由一些卷积核和一个激活函数构成。卷积核实现了 CNN 的局部感知思想，一个卷积核即代表一个局部视野，对应地可以提取出一个特征图。每个卷积层可以有多个卷积。由于卷积层的卷积计算过程只存在线性变化，为了引入非线性变化，需要加入激活函数。CNN 中最常用的激活函数为 ReLU 函数，它是 Alex 于 2012 年提出的一种新的激活函数。相比其他激活函数，ReLU 函数加快了 CNN 的收敛速度。该函数的提出很大程度上解决了 BP 算法在优化深层神经网络时的梯度耗散问题[7]。ReLU 函数的计算公式为：

$$f(x)=\max(0,x) \tag{3}$$

4）池化层：该层通过局部采样对输入的特征图进行压缩。采样方式有两种，即取池化窗口中的最大值或平均值为采样值。在 CNN 中使用池化层，一方面可以使特征图变小，简化网络的计算复杂度；另一方面可以改善过拟合现象。

5）Dropout 层：该层会随机丢弃一些神经元的输入，减轻过拟合的程度，从而提高神经网络在多种监督学习任务上的性能[15]。

6）全连接层：该层连接所有的特征，并将最后的输出输送给分类器（如 Softmax）进行分类。分类器使用 Softmax 函数得到最终的输出。Softmax 函数也称归一化指数函数，它能将一个含任意实数的 $K$ 维向量 $\boldsymbol{Z}$“压缩”到另一个 $K$ 维实向量 $\sigma(\boldsymbol{Z})$ 中，使得每一个元素的范围都在（0，1）之间，并且所有元素的和为 1。也就是说，它将多个神经元的输出映射到（0，1）区间内，即对于 $K$ 个输入项 $Z_j(j=1,\cdots,K)$，softmax 函数得到的输出结果如下式所示：

$$\sigma(z)_j=\frac{e^{z_j}}{\sum_{k=1}^{K}e^{z_k}} \tag{4}$$

在构建好 CNN 的各个层以后，指定对应的 Loss 函数和优化器来对网络进行训练。

由上可知，使用 CNN 进行恶意 Web 请求分类，可以忽略复杂的特征工程，得到一个端到端的恶意检测模型。

### 2.2 原始数据表示

本文设计的 CNN 网络的输入是一个 $l\times h$ 的二维数字矩阵，因此首先需要把原始数据表示为数字矩阵。因为该过程在基于 CNN 的句子分类、情感分析、恶意软件检测等应用中是基本相同的，所以本文参考了多种应用中的方法来进行对比实验。

在上述各类应用中，One-Hot 是使用得最多的方法[10-12,16-17]。One-Hot 又称一位有效编码，其原理是使用 $N$ 位状态寄存器来对 $N$ 个状态进行编码，每个状态对应一个独立的一位有效寄存器位。One-Hot 编码可以消除距离的影响，使特征之间的距离计算更加合理，但是编码结果比较稀疏，维度较高，在深度学习中效率较低。

Word2Vec 在自然语言处理中取得了令人满意的成果，之后被用于 CNN 网络的输入处理[9,13]。Word2Vec 是 Mikolov 在 Google 带领的研究团队于 2013 年开发的一款将词表征为实数值向量的高效词嵌入工具，它基于 CBow 算法和 Skip-gram 算法[18-19]，将字符串作为输入进行训练得到一个模型。该模型将每个词映射到一个向量，也可以用来表示词与词之间的关系。

2015 年，机器学习库 Keras 提供了网络内部的嵌入层。该层可以把一个字符映射为一个向量，并随着整个网络的训练对其表示参数进行优化更新，使得参数的数字化更加准确。但是，由于 CNN 的输入不接受原生字符串，只能是数字，因此需要在网络输入前得到一个数字化的一维向量，再经过词嵌入层把该一维向量的每一位映射为定长向量，最终得到一个二维的矩阵。在 Saxe 等提出的 Explosure[9] 和 Le 等提出的 URLNET[5] 中，每个字符被映射为一个（1，$len(V)$）范围内的整数，然后将其输送给 CNN 嵌入层，其中 $V$ 为原始数据生成的词汇表。该映射方法，即传统机器学习中特征表示的词汇表方法。词汇表方法会对文档生成一个词汇表，该词汇表中的每个字符都有一个对应的数字编号。该方法对一个要进行编码的字符串逐一使用对应的编号来代替原字符，从而得到一个一维的数字化向量。

受已有相关工作的启发，本文采用了其他几种经典的传统机器学习中的特征表示方法对原始数据进行数字化，然后进行嵌入。这几种方法分别为 SoW（Set of Words）、BoW（Bag of Words）、TFIDF（Term Frequency-Inverse Doc Frequency）和词汇表映射。

SoW 模型假定对于一个文档，忽略字符的顺序、频率和语法等要素，仅将其看作若干个词汇的集合，且文档中每个单词的出现都是独立的，不依赖于其他单词是否出现，用 0 或 1 表示单词是否出现来进行编码。与 SoW 相比，BoW 考虑了字符出现的频率，使用频率来进行编码。而 TFIDF 是一种统计方法，用于评估一个字词对一个文件或文件集的重要程度。其主要思想是如果某个词在一篇文章中出现的频率很高，并且在其他文章中很少出现，则认为该词具有很好的分类能力，适合用于分类。

最终，本文得到的所有表示方法有 One-Hot，Word2Vec，SoW + Embedding，BoW + Embedding，TFIDF + Embedding 和 Vob + Embedding 共 6 种，这些方法都实现了原始数据到二维数字矩阵的映射，得到的映射结果可输入 CNN 网络进行训练。

## 3 基于 CNN 的深度学习网络设计

### 3.1 数据收集和预处理

本文将研究的关键问题集中到对 Web 请求进行二值分类的问题上。因此，本文选择 Web 请求数据的基本要求是：带有明确的 0 或 1 的分类标签，包含丰富的 Web 请求参数，

包含多种 Web 攻击手段，并且数量不能过小。在此要求下，本文选择了两个数据集进行合并，得到 123838 条数据。第一个数据集是经典数据集 HTTP data set CSIC 2010[20]，该数据集共有 97065 条数据，包含 72000 条正常 Web 请求和 25065 条恶意请求。第二个数据集为 DEV_ACCESS，该数据集是 Adam 利用特殊构造的 API 收集的数据，用于实现 RNN 恶意检测网络[4]。该数据集共有 26773 条数据，包含 13413 条正常 Web 请求和 13360 条恶意请求。

这两个数据集均包含多种攻击手段和多个 Web 参数，本文对其分别进行预处理。删除 HTTP data set CSIC 2010 数据集中内容重复的参数，并使用空字符串填充一些缺失的字段，最后将处理后的字符串转化为标准 JSON 格式。对 DEV_ACCESS 数据集，亦删除了一些内容重复的参数。处理后，两个数据集保留的参数和数据格式都相同，具体格式如以下样例数据所示。

```
{"timestamp":1502738402856,
    "method":"post",
    "query":{},
    "path":"/login",
    "statusCode":401,
    "source":{
      "remoteAddress":"87.142.27.75"},
    "route":"/login",
    "requestPayload":{"username":"oh",
        "password":"Mc2402"},
    "responsePayload":{"statusCode":401,
      "error":"Unauthorized",
      "message":"Invalid Login"}
}
```

### 3.2 网络设计

#### 3.2.1 原始数据表示

本文设计的网络是一个字符级别的网络。本文没有使用字级嵌入的原因是：1）无法在测试时获得新单词的嵌入；2）太多独特的单词在学习单词嵌入时将导致内存不足；3）字级网络需要复杂的分词过程，而且难以找到合适的分词方法。例如，文献[2]中使用“/”“&”“=”“+”等字符对 URL 进行分词后得到的序列就抛弃了这些字符，而相当一部分的恶意 URL 的 Payload 都包含了这些特殊字符。这样的分词方式会对原始数据造成损失。

字符级别向量化的优点是不需要对语言的语法进行理解，可以去掉复杂的分词过程，实现一个端到端的网络。根据第 2 节介绍的 6 种方法将图 1 中的数据向量化。经过数据分析，本文对每条数据进行截取或填充，使其长度为 480 个字符，即 $l$ 为 480。

1）One-Hot：每个字符使用长为 100 个字符的向量来表示，即 $h$ 为 100，这样就得到一个 480×100 的输入向量。对 3.1 节中的样例数据进行 One-Hot 编码后得到的输入如下所示，其特点是矩阵非常稀疏，每一行数据只有一位为 1。

```
[[0.  0.  0.  …  0.  0.  0.]
 [0.  1.  0.  …  0.  0.  0.]
 [0.  0.  0.  …  0.  0.  0.]
 …
 [0.  0.  0.  …  0.  0.  0.]
 [0.  0.  0.  …  0.  0.  0.]
 [0.  0.  0.  …  0.  0.  0.]]
```

2）使用 Word2Vec，令 $h$ 为 100，对样例数据进行处理，得到如下结果：

```
[[ 2.2803 0.9445 … -0.5193 -2.9386 -2.0482]
 [-0.2632 1.6008 … -0.3007  0.2960 -1.0050]
 [-0.3907 0.3168 … -0.3218  1.9013 -2.3852]
 …
 [0.  0.  …  0.  0.  0.]
 [0.  0.  …  0.  0.  0.]
 [0.  0.  …  0.  0.  0.]]
```

3）使用词汇表映射方法对样例数据进行长为 480 个字符的截断和填充，结果数据如下。这个一维向量经过嵌入层后，可得到一个 480×100 的向量。

```
[[21.  1.  25. … 13.  4.  13.  34.]
 [5.  16.  1.  … 28.  3.  0.  0.]
 [8.  4.  30.  … 35.  6.  10.  12.]
 …
 [1.  7.  3.  … 13.  10.  0.  0.]
 [1.  7.  3.  … 1.  11.  0.  0.]
 [1.  18.  10. … 7.  3.  2.  0.]]
```

4）分别使用 SoW，BoW，TFIDF 对样例数据进行编码。由于这 3 种方法都是基于一个字符集合，这些集合的长度都接近 100 个字符，因此 3 种方法对输入字符串映射后再进行填充均得到一个长为 100 的向量，再经过嵌入层可得到一个 100×100 的输入矩阵。这些向量分别如下所示：

```
SoW:[0.  1.  1.  1.  …  0.  0.  0.]
Bow:[0.  48.  29.  27.  …  0.  0.  0.]
TFIDF:[0.  3.3763  3.0270  2.9775 …  0.  0.  0.]
```

#### 3.2.2 网络架构设计

本文搭建的 CNN 网络的核心结构如图 1 所示。

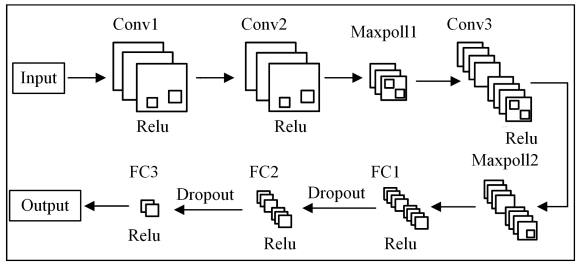


图 1 CNN 网络结构

Fig. 1 CNN network structure

该网络采用了 3 个卷积层、2 个 MaxPool 层、2 个 Dropout 层和 3 个全连接层。该网络的优化器采用了 Adam。Adam 是深度学习领域十分流行的算法，它能较快地实现网络的

收敛。本文采用 Categorical_Crossentropy 损失函数，它是 CNN 中常用的一种损失函数，常用于互相排斥的分类任务中，其计算公式为：

$$H(y,t)=H_t(y)=-\sum_i t_i \log y_i \tag{5}$$

构造上述网络后，将第 2 节得到的 6 种矩阵作为输入层，得到 6 个网络，并设置学习率为 0.0002，*Batch size* 为 64，对这些网络进行训练。实验的具体流程如图 2 所示。

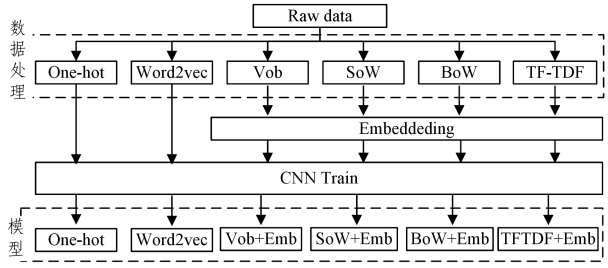


图 2 实验流程

Fig. 2 Experiment process

## 4 实验及结果分析

### 4.1 实验数据

本文把实验数据随机分为两部分进行训练和验证，将 75% 的数据作为训练集，剩余的 25% 的数据作为验证集，并且所有模型使用相同的训练集和验证集。

### 4.2 评估标准

本文最终记录并计算的评估参数有 FP，FN，Precision，Recall 和 F1 值。其中，FP 为预测所检测的 Web 请求为恶意请求，但实际为正常请求的样本个数；FN 为预测所检测的 Web 请求为正常请求，但实际为恶意请求的样本个数；Precision 表示预测为恶意的样本中所包含的真正恶意样本的个数，其值越大说明模型对正常请求样本的区分能力越强；Recall 体现恶意样本中被预测正确的情况，体现模型对恶意样本的识别能力，其值越大说明模型对恶意样本的识别能力越强；由于本文所设计的网络为二值分类网络，只有精确率和回归值无法准确地衡量网络的表现，因此计算了 F1 值，其是精确率和回归值的综合，值越大说明该分类模型越稳健，综合分类性能越好。

相关指标的计算公式如下：

$$Precision=\frac{TP}{TP+FP} \tag{6}$$

其中，*TP* 表示预测为恶意请求，实际也是恶意请求的样本个数。

$$Recall=\frac{TP}{TP+FN} \tag{7}$$

$$F1=\frac{2TP}{2TP+FN+FP}=\frac{2\times Precision\times Recall}{Precision+Recall} \tag{8}$$

### 4.3 实验结果

在对所有模型进行了 15 个或 25 个 Epoch 的训练后，模型都基本收敛，相关训练情况如图 3 所示。

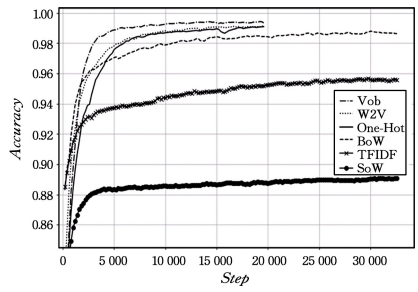


图 3 CNN 模型训练过程中的准确度曲线

Fig. 3 Accuracy curve during training of CNN models

模型训练完成后，使用验证集对其进行评估。验证集共 30960 条数据，包括 21462 条正常数据和 9498 条恶意数据。

此外，为体现该模型的优越性，在该验证集上评估了使用相同训练集训练得到的 HMM，SVM 和 RNN 对照模型，结果如表 1 所列。

表 1 实验结果

Table 1 Experimental results

| | *FP* | *FN* | *Precision* | *Recall* | *F1* |
|---|---|---|---|---|---|
| Vob+Embedding | 12 | 191 | 0.99871 | 0.97989 | 0.98921 |
| One-Hot | 48 | 217 | 0.99485 | 0.97715 | 0.98592 |
| Word2Vec | 66 | 279 | 0.99289 | 0.97062 | 0.98163 |
| BoW+Embedding | 207 | 211 | 0.97819 | 0.97778 | 0.97799 |
| TFIDF+Embedding | 238 | 1193 | 0.97214 | 0.87439 | 0.92068 |
| SoW+Embedding | 185 | 2988 | 0.97236 | 0.68540 | 0.80405 |
| HMM | 310 | 599 | 0.96634 | 0.93693 | 0.95141 |
| RNN | 387 | 635 | 0.95817 | 0.93314 | 0.94549 |
| SVM | 617 | 1643 | 0.92717 | 0.82702 | 0.87423 |

综合图 3 和表 1 的结果可以看出，网络 Vob+Embedding 的各个指标都取得了最佳的结果，SoW+Embedding 的结果最差。由此可以得到如下结论：

1）将 CNN 神经网络应用到 Web 恶意检测是可行的，可以直接把整个请求作为一个 JSON 字符串，在其进行数字向量化后作为输入。

2）将 CNN 神经网络应用到 Web 恶意检测可以取得非常令人满意的结果。该方法具有很高的准确性和良好的泛化能力，是非常有价值的研究方法。

3）在使用 CNN 进行 Web 恶意检测的数据处理方法中，Vob+Embedding 以较明显的优势优于其他方法。在这些表示方法中，数据的信息丢失会对 CNN 的结果造成不良的影响。在数据没有损失的情况下，采用网络内嵌入层会更好地表示数据，得到最佳结果。

**结束语** 本文在已有的利用 CNN 检测恶意 URL 工作的基础上，引入了其他 Web 请求参数对整个 Web 请求进行了基于 CNN 的恶意检测。该方法能够检测出位于除 URL 外的其他参数位置的恶意攻击。同时，本文通过对比实验得出了 CNN 输入层数据的最佳处理方法为词汇表方法与网络内部词嵌入层的结合。该网络在精确度、召回率等评估结果中都取得了极高的分数。同时，该网络在不同的数据集上也能得到良好的泛化结果，为利用 CNN 进行 Web 恶意检测提供了一种思路。

本文采用的数据集攻击类型仍然不够全面，未来需要攻击类型更完善和数据量更大的数据集来进行训练。其次，使用 CNN 实现 Web 恶意检测的过程具有一定的不透明性[21]，因此研究如何提高 CNN 的透明性和可解释性，对于将神经网络应用于安全领域是十分重要的。

## 参 考 文 献


[1] ATIENZA D, HERRERO Á, CORCHADO E. Neural analysis of http traffic for web attack detection[C]// Computational Intelligence in Security for Information Systems Conference. Cham: Springer, 2015: 201-212.

[2] ZHANG M, XU B, BAI S, et al. A Deep Learning Method to Detect Web Attacks Using a Specially Designed CNN[C]// International Conference on Neural Information Processing. Springer, 2017: 828-836.

[3] SAXE J, BERLIN K. eXpose: A character-level convolutional neural network with embeddings for detecting malicious URLs, file paths and registry keys[J]. arXiv: 1702.08568, 2017.

[4] KUSEY A. Detecting Malicious Requests with Keras & Tensorflow[EB/OL]. (2017-09-12)[2018-06-10]. https://medium.com/slalom-engineering/detecting-malicious-requests-with-keras-tensorflow-5d5db06b4f28.

[5] LE H, PHAM Q, SAHOO D, et al. URLNet: Learning a URL Representation with Deep Learning for Malicious URL Detection[J]. arXiv: 1802.03162, 2018.

[6] CHEN Y C, LI Y J, TSENG A, et al. Deep learning for malicious flow detection[C]// Personal, Indoor, and Mobile Radio Communications. 2017: 1-7.

[7] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. Imagenet classification with deep convolutional neural networks[C]// Advances in Neural Information Processing Systems. 2012: 1097-1105.

[8] SZEGEDY C, LIU W, JIA Y, et al. Going deeper with convolutions[C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2015: 1-9.

[9] KIM Y. Convolutional neural networks for sentence classification[J]. arXiv: 1408.5882, 2014.

[10] KALCHBRENNER N, GREFENSTETTE E, BLUNSOM P. A convolutional neural network for modelling sentences[J]. arXiv: 1404.2188, 2014.

[11] ZHANG X, ZHAO J, LECUN Y. Character-level convolutional networks for text classification[C]// Advances in Neural Information Processing Systems. 2015: 649-657.

[12] DOS SANTOS C, GATTI M. Deep convolutional neural networks for sentiment analysis of short texts[C]// Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers. 2014: 69-78.

[13] SEVERYN A, MOSCHITTI A. Unitn: Training deep convolutional neural network for twitter sentiment classification[C]// Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015). 2015: 464-469.

[14] KINGMA D P, BA J. Adam: A method for stochastic optimization[J]. arXiv: 1412.6980, 2014.

[15] SRIVASTAVA N, HINTON G, KRIZHEVSKY A, et al. Dropout: a simple way to prevent neural networks from overfitting[J]. The Journal of Machine Learning Research, 2014, 15(1): 1929-1958.

[16] ATHIWARATKUN B, STOKES J W. Malware classification with LSTM and GRU language models and a character-level CNN[C]// IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2017: 2482-2486.

[17] HENDLER D, KELS S, RUBIN A. Detecting Malicious PowerShell Commands using Deep Neural Networks[C]// Proceedings of the 2018 on Asia Conference on Computer and Communications Security. ACM, 2018: 187-197.

[18] JOHNSON R, ZHANG T. Effective use of word order for text categorization with convolutional neural networks[J]. arXiv: 1412.1058, 2014.

[19] MIKOLOV T, SUTSKEVER I, CHEN K, et al. Distributed representations of words and phrases and their compositionality[C]// Advances in Neural Information Processing Systems. 2013: 3111-3119.

[20] GIMÉNEZ C T, VILLEGAS A P, MARAÑÓN G Á. HTTP data set CSIC 2010[J]. Information Security Institute of CSIC (Spanish Research National Council), 2010.

[21] JOSEPH A D, LASKOV P, ROLI F, et al. Machine learning methods for computer security (Dagstuhl Perspectives Workshop 12371)[C]// Dagstuhl Manifestos. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik. 2013: 3.


**CUI Yan-peng**, born in 1978, Ph.D, associate professor. Her main research interests include network attack and defense.