基于样本增强的网络恶意流量智能检测方法

陈铁明，金成强，吕明琪，朱添田
（浙江工业大学计算机科学与技术学院，浙江 杭州 310023）

摘 要：为解决现有网络流量异常检测方法需要投喂大量数据且泛化能力较差的问题，提出了基于样本增强的网络恶意流量智能检测方法。所提方法从训练集中提取关键词，且基于关键词回避策略对训练集进行样本增强，提高了方法提取文本特征的能力。实验结果表明，所提方法通过小型训练数据集即可提高网络流量异常检测模型的准确率与跨数据集检测能力，相较于其他方法，在显著降低计算复杂度的同时得到了更佳的检测能力。

关键词：样本增强；异常检测；流量检测；机器学习
中图分类号：TP309
文献标识码：A
doi: 10.11959/j.issn.1000−436x.2020122

Intelligent detection method on network malicious traffic based on sample enhancement

CHEN Tieming, JIN Chengqiang, LYU Mingqi, ZHU Tiantian
School of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023, China

Abstract: To address the problem that the existing methods of network traffic anomaly detection not only need a large number of training sets, but also have poor generalization ability, an intelligent detection method on network malicious traffic based on sample enhancement was proposed. The key words were extracted from the training set and the sample of the training set was enhanced based on the strategy of key word avoidance, and the ability for the method to extract the text features from the training set was improved. The experimental results show that, the accuracy of network traffic anomaly detection model and cross dataset can be significantly improved by small training set. Compared with other methods, the proposed method can reduce the computational complexity and achieve better detection ability.

Key words: sample enhancement, anomaly detection, traffic detection, machine learning

1 引言

随着 Web 攻击种类的增加，许多 Web 服务受到各种形式的安全威胁和网络攻击。近年来，不断出现的漏洞及漏洞利用方法使许多传统的恶意流量检测方法已经失效。在现代的网络攻击中，大部分攻击者都将 Web 服务作为主要攻击目标，进行数据窃取、漏洞攻击等。如何有效并且准确地检测此类流量是目前亟需解决的关键问题。

目前的恶意 Http 流量检测方法大致可分为两类：签名检测方法和异常检测方法[1]。签名检测方法主要构建已知攻击的检测模型，具有相应攻击签名的行为和流量将被识别为攻击流量，该方法虽然准确率高但无法检测未知攻击。异常检测方法的策略是根据正常行为的流量创建规则，凡是违规的流量都会被识别为攻击行为，该检测策略对未知的攻击具有一定的检测能力，但是其与白名单机制类似，所有的规则都被预先设定在一定范围内，一旦

收稿日期：2019–08–19；**修回日期**：2019–12–18
基金项目：国家自然科学基金资助项目（No.61202282, No.61772026）；国家自然科学基金与浙江省政府联合项目（No.U1509214）
Foundation Items: The National Natural Science Foundation of China(No.61202282, No.61772026), Joint Project of National Natural Science Foundation and Zhejiang Provincial Government(No.U1509214)

用户产生非预期但仍然属于正常范围的操作行为，流量就会被误判为恶意流量。因此，异常检测策略在一定程度上影响了用户的操作体验，且可能会对正常流量误判，同时，由于采用行为规则或流量语句规则匹配的方式，使攻击者易于通过编码、代码混淆和加密等途径绕过规则匹配，难以检测新型以及未知的 Web 攻击语句。

针对上述问题，可以通过机器学习提取异常流量特征，进而通过异常检测技术区别异常流量与正常流量。因为 Web 攻击具有时效性，所以当前异常检测方法往往需要大量的流量样本用于训练异常检测模型。然而，流量样本的获取难度很大，特别是恶意流量样本，导致通常情况下只能获取少量训练样本，而仅利用少量训练样本难以有效提取特征，导致模型过拟合和泛化能力弱（特别是深度学习模型）。

在相关研究工作中，虽然恶意 Web 流量检测方法在指定数据集上准确率最高达到 90%，但是在其他数据集上准确率会下降 10%~20%，对现实场景中恶意 Web 流量检测泛化能力也较弱，对混淆后的恶意 Web 流量无法检测。

因此，需要设计一种方法，仅通过少量训练样本训练模型就能够取得良好的恶意流量检测效果。目前，针对该问题比较有效的方法是利用样本增强的方式，基于少量训练样本产生更多训练样本扩充训练集，以使模型更好地提取特征。然而，目前样本增强的方法大多适用于图像或语音数据，例如，通过对图像的移位变换、视角变换、大小变换等，以及应用高斯噪声使图像样本增强中添加的噪声数据具有较低的信息失真水平。显然，这些方法不适用于文本类型数据的增强任务。在文本类型样本增强方面，有相关研究提出利用同义词替换[2]以及构造加权无向图[3]等方法，但是此类方法适用文章、问答等类型的文本，主要研究其语义关联特性。由于 Web 流量中大部分数据都属于计算机程序代码，例如“script”“for”“insert”等，因此，各类字符以及单词都无法采用同义词替换的方式进行样本增强。此外，恶意 Web 流量往往存在特定的语序结构及语义关联，例如，结构化查询语言（SQL, structured query language）注入会存在“select”“from”的组合，在跨站脚本（XSS, cross site scripting）攻击中存在<script></script>等，而这类语序结构在正常 Web 流量中一般不会存在，并且 Web 流量的语序结构难以通过简单的正则匹配提取特征。

综上所述，需要一种可行有效的样本增强方法应用于 Web 流量训练集，通过小容量的训练集实现恶意流量检测并且具备较好的泛化能力。因此，本文提出了一种基于关键词回避的流量样本增强方法，当训练样本容量较小时可对流量样本进行样本增强。首先，通过对流量训练样本根据特殊字符进行自动分割，提取并建立关键词库。然后，基于关键词对训练集进行关键词库回避加噪，实现训练样本增强。最后，将增强样本输入卷积神经网络用于机器学习。本文数据预处理方法可强化深度神经网络对关键特征的提取，优化异常流量检测能力，提高模型的稳健性与泛化能力。

本文的贡献总结如下。

1) 本文提出了全新的基于关键词库回避的样本增强方法。对已经标记的 Web 流量根据特殊字符进行自动分割，提取并建立关键词库。基于关键词库回避，选择对原始训练集的非关键词字符数据进行随机化的加噪处理，将噪声数据与原始训练数据混合产生神经网络训练集。

2) 本文设计了一种面向恶意 Web 流量检测的深度学习神经网络。该神经网络训练得到的检测模型具有较高的准确率，并且在其他数据集上依然表现较好，同时有较好的稳健性与泛化能力。

2　相关工作

本文的主要研究工作是利用深度神经网络对异常流量进行检测和识别，下面介绍近年来相关的研究现状。在数据预处理阶段，现有方法通过对大量数据进行语句简单分割以及字符归一化处理来增强数据，进而强化模型特征提取。现有异常流量检测方法可以分成两类：基于特征分布的异常流量检测和基于内容的异常流量检测。

2.1　文本样本增强

通过 Http 流量对异常流量检测，关键是采用文本数据分类的方法对 Http 流量进行二分类。但是 Web 攻击语句具有高度的时效性，并且可能存在各种方式的加密、混淆和替换，因此需要对训练数据进行适当样本增强从而使神经网络更好地提取异常流量中的特征。文本样本增强的主要意义是可以更好地提取文本特征，例如关键词、语序、语义关联等，同时文本样本增强技术使神经网络可以通过较少的数据提取相应的特征。Zhang 等[2]提出利用

同义词替换单词或短语，使用一个英语词库对文本数据进行扩充。Lu[3]为了提高支持向量机（SVM, support vector machine）在稀疏训练条件下的分类性能，利用高斯随机分布理论为基础，构造加权无向图的半监督学习算法来进行样本增强。

在文本样本增强方面，由于字符的顺序会影响语法与语义，因此使用图像或者语音识别的转换来增加数据是不合理的。而且 Web 流量数据大部分的字符是独立存在或无意义的编码以及非语义化单词，利用同义词替换反而会影响原有特征提取，破坏异常流量中存在的恶意关键词。因此，需要在保留异常流量中大部分关键词的基础上对 Web 流量进行样本增强。

2.2 异常流量检测

大部分异常流量都是在某些常见的恶意流量基础上进行编码以及增加少量变量变种，两者之间有大量的复用数据，具有恶意流量显著特征。将网络中的正常行为作为依据，凡是与预期的正常行为不一致的网络流量均被视为异常。大部分异常流量检测方法通过机器学习、数据挖掘或统计的方法来推知正常流量特征，并检测异常流量。

在异常流量检测方面，许多研究都使用了不同的模型与数据预处理方式来提取 Web 流量特征。Zolotukhin 等[4]通过对 Http 日志的分析，提出了一种针对 Web 攻击的异常检测方法。Park 等[5]提出了一种基于字符级二值图像变换的卷积自动编码器（CAE, convolutional auto encoder）对 Http 消息进行异常检测的方法，其使用了基于字符级分割的数据预处理方法，效果优于传统启发式选择输入特征的机器学习方法。Yu 等[6]提出了一种利用双向长短期存储器（Bi-LSTM, bidirectionallong short-term memory）和注意机制的深度神经网络模型，将 Http 流量建模为自然语言序列来检测恶意流量，其中数据预处理采用了基于特殊字符分割的方法。Yang 等[7]设计了一种卷积门控递归单元（CGRU, convolutional gated-recurrent-unit）神经网络，将常见的恶意统一资源定位符（URL, uniform resource locator）关键词建立词库，之后将该词库用于模型训练，将非关键词进行字符级分割，最终用基于字符作为文本分类特征的恶意 URL 检测。

在提取 Web 流量的有效数据方面，Arzhakov 等[8]利用蜜罐技术来收集攻击者行为的统计信息，用统计分类的方法来区分异常流量。Xu 等[9]则提出利用 Http 中 content-type 字段的不一致性来检测规避的网络攻击，可以有效地从流量中发现未知的恶意软件。Torrano-Giménez 等[10]提出了一种基于异常的网络流量入侵检测方法，利用可扩展标记语言（XML, extensible markup language）文件将传入的请求分类为正常请求和异常请求。这种方法利用 N-gram 模型，从 Http 日志的 Web source、query attributes 以及 user agents 这 3 个字段中提取相应的特征，利用机器学习中的 3 种算法来处理数据，分别为支持向量数据描述（SVDD, support vector data description）[11]、K-means[11]和基于密度的噪声应用空间聚类[12]。考虑到大部分恶意 Web 流量的攻击会出现在 Http 请求中，直接提取网络应用防火墙（WAF, Web application firewall）或网站设备中 Http 请求作为数据源是最可靠的方法，因为部分攻击者会利用攻击载荷修改 Web 服务器日志，因此利用 Web 服务器日志作为数据集并不可取。而在 Http 请求中，攻击者的攻击流量 90%以上出现在消息主体数据（entity-body data）和请求 URL（request-URL）中，因此可以从以上 2 个字段中提取相应特征，同时避免提取过多字段使数据特征过多导致过拟合。

在混合模型方面，Choraś 等[13]提出了一种机器学习方法来模拟 Web 应用程序的正常行为，同时检测网络攻击，该模型基于从 Http 请求中获取信息，利用基于图的分割技术和动态编程来产生模型。Kruegel 等[14]提出了一种检测网络攻击的多模型方法，通过分析 Http 请求，并使用基于不同特性的多重不同模型，包括属性长度、属性字符分布、结构推理、调用顺序等。Corona 等[15]设计了一个多分类器系统，通过对正常请求的建模来检测 Web 攻击，该系统采用一套预先定义的模型，把模型建立在 Http 请求中的不同字段上，采用统计分布模型和隐马尔可夫模型作为基本模型。Ringberg 等[16]提出了一种显式状态持续时间的非参数隐马尔可夫模型，用于 Http 会话进程的聚类和跟踪，这种方法按照会话规模分析 Http 流量，而不是按照特定的流量条目。Al-Obeidat 等[17]提出了一种新的基于模糊决策树和属性选择的监督式混合机学习方法用于流量分析。Erfani 等[18]提出一个混合模型，通过无监督的深度信念网络（DBN, deep belief network）提取通用的底层特征，用一个单类的 SVM 从 DBN 学习特征。LSTM 模型一般用于系统日志中的异常检测和诊断[19]。Zhang 等[20]提出利用多模型检查 Http 请

求消息来识别攻击，采用概率分布模型、隐马尔可夫模型和一类支持向量机模型 3 种机器学习模型对 Http 请求消息的不同字段进行检测，如果有一个模型报告异常，则总体检测结果为异常。

在对流量数据分组内容检测方面，通过对数据分组内容的解析、统计等方式来识别异常流量。Cretu-Ciocarlie[21]提出一种内容异常检测器，通过对高阶的 N-grams 的混合模型进行建模来检测异常流量和“可疑”的网络数据分组。

上述工作都是运用多种模型以及提取不同特征对流量进行建模分类，但是大部分工作在数据预处理部分采用简单的 N-grams 分词或数据统计等，使用简单数据统计对流量进行预处理将导致流量数据特征不能被较好地保留，同时使用 N-grams 方法将会产生维度爆炸的问题，并且大部分 Web 流量中存在无语义关联字符与编码字符，使用 N-grams 提取不同长度的关键词都会产生大量的编码字符以及无效字符串，而传统恶意关键词往往得不到保留。因此，本文提出将 Web 流量进行分割保留传统恶意关键词，再对不同种类的异常 Web 流量的关键词集合进行合并，去除其中的单字符，最大限度保留能代表异常 Web 流量的关键词，保留的异常 Web 流量关键词在样本增强方法上也起到关键性作用。

3　方法描述

3.1　总体架构

本文方法的总体架构如图 1 所示。训练数据集为正常流量数据与异常流量数据经过预处理后混合所得数据集，关键词库为根据特殊符号对训练数据集分割并保留高频字符串的集合，增强的训练数据集为通过基于关键词回避进行样本增强后的训练数据集，向量语句是将指定的数据通过 Word2vec 转化所得的向量。本文方法基本思路如下。首先，将提取正常流量与异常流量的请求数据混合得到训练数据，将训练数据集中所有数据基于特殊符号进行分割，保留分割所得字符串中出现次数较多的字符串得到关键词库。然后，基于关键词回避策略，选取部分训练数据集样本生成对应的增强样本，并且重新加入训练数据集得到增强的训练数据集，利用 Word2vec 将训练数据集中所有数据转化为向量语句。最后，将所得的向量语句输入基于文本分类的全监督学习模型进行训练得到检测模型。

利用本文方法进行检测时，先对待检测流量进行预处理提取请求数据，利用 Word2vec 转化为向量语句，再利用检测模型进行检测。

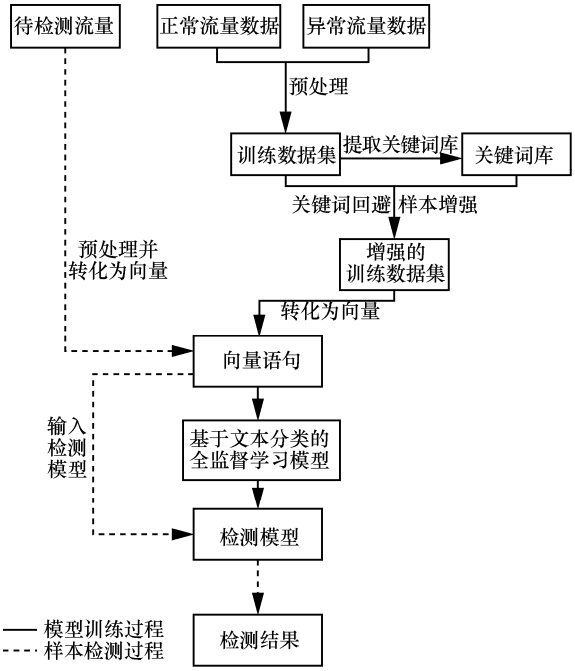

图 1　本文方法总体架构

3.2　基于数据加噪的样本增强

机器学习模型的性能受训练样本的制约。当训练样本数量和多样性不足时，机器学习模型容易产生过拟合，特别是参数数量多、学习能力强的深度学习模型[22]。因此，在训练样本中添加噪声有助于提高训练样本的多样性，有助于机器学习模型抵消噪声的影响，从而降低过拟合的风险。

目前已有的样本增强方法大都针对图像数据，通过对图像的移位变换、视角变换、大小变换，以及应用高斯噪声，使添加的噪声数据具有较低的信息失真水平。Web 流量数据与文本数据类似，针对文本数据，现有研究提出了利用同义词替换的方法[2]以及构造加权无向图的方法[3]。然而，此类方法适用于文章、问答等类型的文本，主要研究其语义关联特性，而 Web 流量中大部分数据都属于计算机程序代码，例如“script”“for”“insert”等，因此各类字符以及单词都无法采用同义词替换的方式进行样本增强，并且恶意 Web 流量中存在字符编码、混淆和加密等也难以适用同义词替换方法。

本文采用特征保留的思想，一方面保留关键特征文本，另一方面随机替换非关键特征文本，从而在实现样本增强的同时保留 Web 流量的关键信息不被破坏。

3.2.1 *数据解析*

将恶意 Http 流量划分成以下三类：跨站脚本（XSS,cross site scripting）攻击流量、SQL 注入流量和路径爆破流量。由于大部分 Web 攻击的攻击载荷都在 Http 流量的请求路径以及请求数据中，因此首先解析所有 Http 流量中的请求路径和请求数据来形成样本。为了正确地对样本中数据解析，需要对进行预处理，步骤如下。

1) 数据提取。提取 Get 请求中的请求路径和 Post 请求中的请求数据。

2) 数据清洗。对提取后的数据进行 URL 解码、字符小写、去除空格等无效字符，得到最终样本。

3.2.2 *关键词库生成*

为对文本型数据进行有效处理，需要先将文本分割成大量独立的单元。在以往相关工作中，通常将文本数据按照单字符分割，但是在复杂请求中会包含大量数据，仅通过单字符提取特征最终会归结到某一字符的分布情况与出现频率，脱离了整体请求数据内容的识别，导致在最终模型预测中容易出现过拟合。因此，需要将样本按照词进行分割，建立词之间的语义关联[23]。例如，data1 存在 SQL 注入攻击载荷。

data1 http://www.example.com?user=-1' union select 1,2,password from users#

为了正确识别 SQL 注入，本文关注 SQL 注入中存在的特征语句。例如，当请求数据中存在 SQL 查询语句时，极大可能为恶意请求。因此，可以将常见的 SQL 查询语句记录为关键词（同理，也可以将 XSS 攻击和路径爆破攻击中常见语句记录为关键词）。然而，人工记录 Web 流量关键词存在以下问题：首先，Web 流量关键词数量巨大，人工记录代价大且召回率低；其次，由于攻击手法以及攻击载荷会随着时间的推移发生变化，某些关键词会被混淆，导致无法人工记录。例如，空格使用“%0a”替换，“select”使用“/*!50000%53elect*/”替换，此时 data1 会被转换成 data2。

data2 http://www.example.com?user=-1'% 0 aunion%0a/*!50000%53elect*/%0a1,2, password %0 from% 0ausers#

综上所述，理想的方式是基于训练样本分析 Web 流量用词模式，自动挖掘 Web 流量关键词建立关键词库。许多研究采用 N-grams 来对文本进行关键词挖掘[24]，该方法能够有效地捕获字节频率分布和序列信息用于分析高频词的内容与序列，但容易产生大量毫无意义的词，且挖掘长度较长的关键词计算代价很大（例如，“select”是一个关键词，则“se”“ec”“sel”等“select”的子串也都是关键词）。因此，可基于“中文分词”的思想将 Web 流量语句分割成有意义的词，并从中挖掘关键词。考虑到 Web 流量中存在各种特殊符号，提取 2 个特殊符号中的子串作为一个词简化了分割过程，同时通过这种方式提取的词在 Web 流量中大都具有实际语义。因此，将 Web 流量中常见的 28 种特殊字符（包括“.”“,”“'”“"”“<”“>”“+”“-”“_”“*”“=”“{”“}”“(”“)”“[”“]”“~”“/”“\”“#”“:”“;”“?”“!”“-”“&”“@”）作为分割依据。如例 1 所示。

例 1 有效请求路径为/5o9zq43e.cfm?<script>cross_ site_ javascript.nasl</script>，分割后的数据为 5o9zq43e cfm script cross site javascriptnasl script。

从例 1 可知，“script”“javascript”等与 XSS 语义强相关的关键词会被分割出来。本文将恶意 Http 流量划分成三类：XSS 流量、SQL 注入流量以及路径爆破流量。对三类恶意 Http 流量分别进行字符串分割，并去除所有单字符，得到字符串集合 AS、BS、CS，最终恶意 Web 流量关键词集合 KS = AS ∪ BS∪CS。

考虑到简单分割提取得到的恶意 Web 流量关键词集合 KS 中可能会存在部分关键词也出现在正常 Web 流量中，这些关键词与恶意 Web 流量关联较弱，因此需要从 KS 中去除这些关键词。首先，对正常 Web 流量用相同方法分割提取得到关键词集合。考虑真实环境中正常 Web 流量数目远多于恶意 Web 流量，所以仅保留正常 Web 流量集合中出现频率大于 99% 的关键词，最终正常 Web 流量关键词集合记为 NS。然后，最终恶意流量关键词集合 W= KS–NS。

具体步骤如下。

1) 将异常 Http 流量分成三类，并提取有效数据得到 XSS 流量样本、SQL 注入流量样本和路径爆破流量样本，同时对正常 Http 流量提取样本。

2) 分别对 XSS 流量、SQL 注入流量和路径爆破流量样本分割字符串，得到 XSS 字符串集合 AS、SQL 注入字符串集合 BS、路径爆破字符串集合 CS。

3) 对正常流量有效数据分割字符串，得到正常流量字符串集合 N。

4) 根据式(1)进行计算，得到异常 Http 流量关

键词库 W，其中，j 表示集合中的字符串，P_j 表示字符串 j 出现的频率，NS 为所有出现频率大于 99% 的字符串集合。

$$W = (\mathrm{AS} \cup \mathrm{BS} \cup \mathrm{CS}) - \mathrm{NS}$$
$$\mathrm{NS} = \{j, P_j > 99\%, j \in N\} \quad (1)$$

3.2.3　基于关键词库的数据加噪

关键词库是将训练样本转化为词向量表示，而关键词库来源于训练样本，因此应该最大限度保留关键词，同时对非关键词字符进行随机替换，随机加噪替换规则如下。

1) 保留所有关键词库中的关键词；

2) “%”“$”“#”“@”等符号保持不变；

3) 数字字符随机替换为 0~9 的其他数字字符；

4) 英文字符随机替换为 a~z 的其他英文字符。

根据实验所得数据，关键词库中包含的“union”“from”在以下新型攻击载荷 data2 中可以检测到。提取其中的主体数据部分命名为 data3。

data3 -1'%0aunion%0a/*!50000%53elect*/%0a1, 2, password%0from%0ausers#

在 data3 中可以发现“–1”“1”“2”“password”“users”都是 SQL 查询中的常规变量，因此可以进行变量替换，得到 data4 。

data4 -5'%0aunion%0a/*!50000%53elect*/% 0a2, 5, name%0from%0alogs#

而 data3 与 data4 因为“union”“from”仍然存在，所构成的词向量表示矩阵并没有发生变化。未回避关键词进行字符替换时会得到 data5。可以发现 data5 的 SQL 语法已经不再符合规范（报错语法），并且此时只存在一个 where 语句疑似 SQL 注入语法，使 data5 的词向量表示矩阵权重下降，在模型训练时难以被有效判别。

data5 -5'%0awhere%0a/*!50000%53table*/%0 a 2,5,name %0alert%0alogs#

如果不对训练样本进行适度替换的方式进行加噪，由于训练样本往往采集于同一网络环境和靶场环境，而模拟攻击方往往采用自动化攻击方式，因此，如“password”“username”“passwd”等在训练样本中大量存在，同时“password”“username”“passwd”在正常请求中又会经常以变量的方式存在。最终导致正常请求训练样本中出现“password”等语句时会被识别为恶意流量。

除了关键词之外的其他字符发生变化有助于提高模型的泛化能力。因此，本文引入对数据的加噪，并且遵循随机加噪替换规则。

根据以上随机加噪替换规则，基于 XSS 攻击有效数据产生噪声数据结果如下。

有效数据为 /5o9zq43e. cfm?<script> cross_ site_scripting.nasl</script>

噪声数据为/6a0gs25e.ulc?<script>iakzm_qpzd_ zjladoznd. trn</script>

可以看到，产生的噪声数据中异常流量的关键词都得到了保留，而其他字符则发生了改变。

3.3　基于深度学习的恶意流量检测方法

本文采用基于卷积神经网络的 TextCNN[25]。因为相比浅层的机器学习方法（如朴素贝叶斯算法（NB, naive Bayes）和 SVM）卷积神经网络效果更好，特别是在数据集较大的情况下，并且不用人工提取特征。在处理文本分类上，卷积神经网络通常包含嵌入层、卷积层、池化层和全连接层。嵌入层可以将所有单词映射到指定的维度空间上；卷积层用于提取句子的特征；嵌入层会产生许多纬度较大的特征，用池化层可以很好地对特征进行降维，最常用的有最大值池化法和均值池化法；全连接层在整个模型中起到分类器的作用[26]。

卷积神经网络的核心思想是捕捉局部特征。对于文本来说，局部特征就是由若干单词组成的滑动窗口，类似 N-gram。卷积神经网络的优势在于能够自动地对 N-gram 特征进行组合和筛选，获得不同抽象层次的语义信息。因此，基于卷积神经网络的 TextCNN 在文本分类问题上有着更加卓越的表现

TextCNN 中嵌入层可以将单词映射到一组向量表示。对于数据集里的所有词，因为每个词都可以表征成一个向量，因此可以得到一个嵌入矩阵 $\boldsymbol{M}$，$\boldsymbol{M}$ 的每一行都是词向量。池化层用于提取主要特征，得到最终的特征向量。全连接层的输入为池化操作后形成的一维向量，经过激活函数输出，再加上 Dropout 层防止过拟合。

池化层和全连接层的处理过程如图 2 所示。由于需要对恶意流量中的字符提取潜在的语序、语义关联，例如“<script>alert(1)</script>”恶意语句中，“script”之后会出现第二个“script”产生闭合标签对。因此，使用 Word2vec 从加噪后的恶意流量训练集生成词向量文件；训练模型时将已经标记过的异常流量与正常流量训练集作为输入，同时在嵌入层输入词向量，构建语义关联；

最后得到检测模型。

检测模型采用一层嵌入层对训练样本进行向量化，一层窗口大小为 2 的池化层进行最大值池化，对卷积后得到的若干个一维向量取最大值，然后拼接起来作为本层的输出值，从而提取特征。然后，送入两层全连接层，第一个全连接层采用 ReLu 作为激活函数，第二个全连接层采用 Sigmoid 作为激活函数。最后，进行分类。

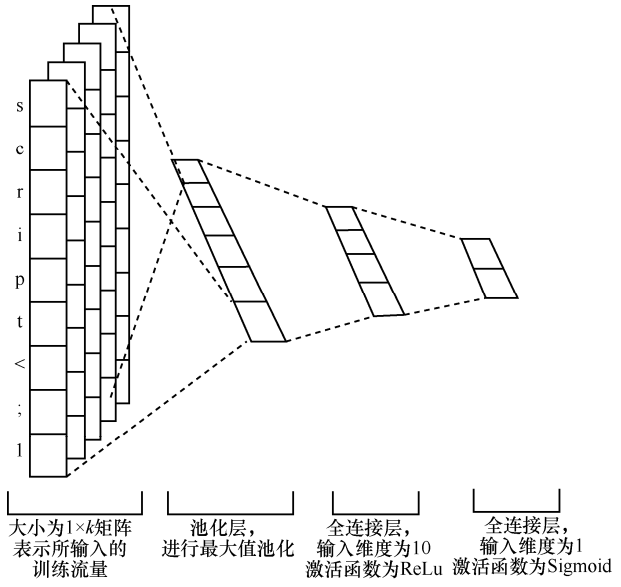

图 2 池化层和全连接层的处理过程

4 实验与分析

本文设计并实现了一个针对 Http 恶意流量检测的原型系统，并使用网络公开数据集进行了实验和评估。

根据文献[27]介绍的常见分类算法性能评估参数，在分类算法中，有以下基本指标：真正例（TP, true positive）、假正例（FP, false positive）、真负例（TN, true negative）以及假负例（FN, false negative），由这些基本指标构成其他更能准确表达分类器效果的指标。由这 4 个基本指标可以构成表 1 所示的常用指标。真正例率（TPR, true positive rate）表示能将正例正确分类的概率，本文中 TPR 表示被正确识别为异常流量的概率；假正例率（FPR, false positive rate）表示将正例分类错误的概率；精确度（precision）表示正确分类的正样本占总正样本的数量；召回率（recall）表示实际为正的样本被预测为正样本的概率；*F*-measure 表示精确度和召回率的调和均值；准确率（accuracy）表示所有正确分类样本与所有样本之比。

表 1 评估指标

名称	计算式
TPR	$\frac{TP}{TP+FN}$
FPR	$\frac{FP}{FP+TN}$
精确度	$\frac{TP}{TP+FP}$
召回率	$\frac{TP}{TP+FN}$
F-measure	$2\frac{precision \times recall}{precision + recall}$
准确率	$\frac{TP+TN}{TP+TN+FP+FN}$

4.1 实验数据集

本文收集了一套来自 Fsecrurify 基于 Http 协议的开源 WAF 请求数据集作为训练集与测试集。从中任意选取 4 万条确定具有恶意行为特征的恶意流量以及 4 万条正常流量，分别对正常流量、恶意流量按照 3:1 比例随机分割成训练集与测试集。同时，本文收集蜜罐服务器（2019 年 3 月份部署）记录的在野恶意流量（下文简称在野恶意流量）5 586 条作为泛化性能测试数据集。在本文恶意流量样本中，SQL 注入、XSS 漏洞以及路径爆破流量样本占 80%~90%，后门控制、隐私窃取流量样本等约占 10%.

4.2 系统环境

原型系统使用 Python 编程实现，开发及训练测试工作在一台计算机上进行，计算机的配置如下：GPU 为 GeForce GTX 1050 Ti 系列，CPU 为主频 3.20 GHz 的 AMD R5 1400，内存为 16 GB。

4.3 实验数据配置

训练集中恶意流量样本与正常流量样本比例为 1:1，均为 30 000 条流量样本。测试集中恶意流量样本与正常流量样本比例为 1:1，均为 10 000 条流量数据。在野恶意流量 5 586 条作为泛化性能测试数据集，其中，包括 614 条 SQL 注入流量样本、4 093 条 XSS 流量样本、879 条路径爆破流量样本。具体流量样本配置如表 2 所示。

4.4 实验结果评估

本文实验探究了训练集多种加噪比例对模型检测效果的影响，输入加噪训练集总数为 30 000 条，测试集包含各 10 000 条异常流量数据与正常流量

数据以及额外蜜罐服务器收集的 5 586 条在野恶意流量。实验使用相同的卷积神经网络模型检测，使用表 2 的指标作为评估指标。实验结果如表 3 和表 4 所示。

表 2　　实验数据配置

样本种类	样本数/条
恶意流量训练集	30 000
恶意流量测试集	10 000
正常流量训练集	30 000
正常流量测试集	10 000
在野 SQL 注入测试集	614
在野 XSS 测试集	4 093
在野路径爆破测试集	879

从表 3 和表 4 实验结果可以看出，基于未加噪训练集产生的模型出现过拟合，表 3 中可以发现其召回率虽然达到了 94.56%，但是准确率仅有 73.24%，虽然恶意流量的识别率较高但会将大部分正常流量也识别为恶意流量，整体误报率偏高。由此可知，在训练集未加噪时，训练所得模型泛化能力较差，对复杂流量的识别准确精度也不够高。当训练集中加入 10%噪声数据时，训练所得模型识别正常流量与恶意流量的准确率都得到了提高，同时泛化能力增强，整体性能得到了提升。为验证加入 10%噪声数据时模型性能最优，对加噪 10%区间附近的比例进行更精细的测试（即加入 5%噪声数据与 15%噪声数据进行测试），可以发现确实在训练集中加入 10%噪声数据能够使模型性能最优。而当训练集加噪比例在 20%~50%时，由于产生过多的噪声数据，噪声数据在整体数据集中比重过大进而使原有数据的特征难以被提取，因此对模型训练出现负面影响使模型出现过拟合，最终导致准确率和泛化率都有所下降。

分析表 3 和表 4 结果可知模型检测能力初期呈增长趋势，在到达顶点之后呈下降趋势。在检测能力呈增长趋势阶段，在训练数据集中应用少量样本增强，通过 3.2 节可知，本文方法生成的噪声数据将会保留异常流量的关键词，因此使增强后的训练数据集异常流量关键词数量增加，强化了模型对关键词出现的位置以及语序的检测，例如“select…from”的 SQL 注入语序及“<javascript>”“</javascript>”的 XSS 标签对语序，从而使模型检测能力随之提

表 3　　不同加噪比例对模型识别性能的影响

加噪比例	TPR	FPR	精确度	召回率	*F*-measure	准确率
0	94.56%	48.09%	66.31%	94.56%	77.95%	73.24%
5%	94.64%	20.67%	82.07%	94.64%	87.91%	86.98%
10%	95.63%	17.19%	84.75%	95.63%	89.86%	89.22%
15%	80.15%	16.02%	83.34%	80.15%	81.71%	82.06%
20%	16.71%	15.86%	51.27%	16.71%	25.21%	50.44%
30%	68.62%	20.60%	76.89%	68.62%	72.52%	74.01%
40%	51.91%	5.47%	90.45%	51.91%	65.96%	73.23%
50%	67.20%	13.28%	83.49%	67.20%	74.46%	76.97%

表 4　　不同加噪比例对模型识别泛化率的影响

加噪比例	路径爆破识别准确率	SQL 注入识别准确率	XSS 识别准确率
0	50.00%	50.00%	50.00%
5%	80.13%	80.84%	90.94%
10%	81.23%	81.92%	92.69%
15%	79.15	80.43%	91.71%
20%	45.96%	89.74%	86.42%
30%	45.96%	85.02%	82.97%
40%	100.00%	94.46%	96.63%
50%	86.23%	88.76%	93.31%

表 5 不同数据预处理方法对模型识别性能的影响

预处理方法	TPR	FPR	精确度	召回率	*F*-measure	准确率
本文方法	95.63%	17.19%	84.75%	89.86%	89.86%	89.22%
文献[5]方法	38.30%	0.75%	98.08%	38.30%	55.09%	68.79%
文献[6]方法	68.12%	28.98%	70.15%	68.12%	69.12%	69.57%
文献[7]方法	82.87%	4.37%	95.00%	82.87%	88.52%	89.25%

高。在检测能力呈下降趋势阶段，由于噪声数据比例超过一定阈值，导致训练数据集异常流量关键词数量过多，进而使模型更加强化了有无关键词的检测，关键词语序检测权重降低，从而出现模型检测误判。例如，正常流量中出现“javascript”“etc”等属于关键词的字符串，也会被模型误判为异常流量。

综上所述，在训练集中增加少量噪声数据，可以提高模型的准确率、泛化率，提升模型识别检测性能，但是超过比例阈值之后，模型会出现过拟合，模型的准确率或泛化率下降，整体识别检测性能均呈下降趋势。

4.5 相关工作比较

在数据预处理步骤中，使用本文所述方法（训练集加噪 10%）和文献[5-7]方法来得到对应的训练模型，通过在样本量、检测率、误报率、准确率 4 个方面，对不同数据预处理方法训练所得模型性能进行比较。不同数据预处理方法中使用相同训练集与测试集，同时使用相同的卷积神经网络进行训练，模型用于检测分类，对比结果如表 5 和表 6 所示。使用本文方法分别用未加噪的训练集与加噪 10%的训练集所得模型，在相同的测试集上进行检测分类，对比结果如表 7 所示。

表 6 不同数据预处理方法对模型识别泛化率的影响

预处理方法	在野恶意流量识别准确率
本文方法	72.86%
文献[5]方法	62.28%
文献[6]方法	60.08%
文献[7]方法	66.39%

精确度是衡量机器学习模型的预测结果中对正类的预测有多大可能性是正确的，而召回率是衡量机器学习模型对正类有多大可能性进行正确预测。因此，如果建立的机器学习模型精确度比较低，说明在该模型的预测结果中对正类的预测准确率很低；如果建立的机器学习模型召回率比较低，说明该模型对正类缺少预测能力，无法对正类进行准确预测。因此精确度和召回率也是一对矛盾的度量。一般说来，精确度高时，召回率往往偏低；而召回率高时，精确度往往偏低。良好的机器学习模型的目标之一就是在精确度和召回率之间获得一个平衡，一方面最大限度地提高对正类样本的预测准确率，另一方面尽量减少假阴性和误报的数量。由表 5 可知，文献[7]方法和文献[5]方法精确度都高于本文方法，但是其召回率低于本文方法。此时通过 *F*-measure 可以更好地度量精确度与召回率之间的平衡关系，而本文方法 *F*-measure 值最高，同时精确度与召回率的差值绝对值为 5.11%，而文献[7]方法为 12.13%，文献[5]方法为 59.78%，说明本文方法不仅能够使检测模型对异常流量有较高准确率，也保证了较低的误报率。另外，精确度与召回率差值绝对值过大说明文献[5]方法出现了过拟合，即将大量正常流量样本误判为异常流量。

表 7 加噪训练模型与未加噪训练模型泛化能力比较

模型	训练数据集准确率	在野恶意流量准确率
本文方法（10%噪声）	89.22%	83.54%
本文方法（无噪声）	73.24%	52.81%

综合表 5、表 6 数据可知，在使用小型训练流量样本的情况下，本文方法与其他方法相比，准确率高且误报率低，整体性能最优。文献[6]方法所得的模型虽然泛化能力较高，但是无法在复杂流量中区分恶意流量与正常流量。而文献[5]方法用于大量数据训练模型时，由于产生了数据维度爆炸，因此模型对恶意流量检测识别性能过拟合，将大量正确流量样本都识别为恶意流量，使整体准确率下降，区分异常流量能力最弱。而文献[7]方法虽然检测识别常规恶意流量能力好，但由于 Web 攻击语句的多样性和实效性，手工添加关键词库难以涵盖所有异常流量关键词以及潜在的攻击载荷，所以通过该数据预处理方法所得的模型泛化能力仍然较弱。而本文方法先将字符串按照特殊字符进行分割再通过统计建立关键词库，最大限度保留异常流量中的关

键词，再通过样本增强的方式来扩增训练数据集，让检测模型更易于提取其中的关键词特征以及相关的语序特征，较大的训练数据集同时也使检测模型在训练时降低了过拟合的风险。在野恶意流量数据集由近期部署的蜜罐捕获产生，而 Fsecrurify 数据集收集时间为 2017 年，因此二者相比而言在野恶意流量数据集中攻击者的攻击方式更加先进，例如通过多次编码对攻击数据进行混淆以及通过语法替换来绕过 WAF 对关键词的检测等。由于本文方法能够基于大量的恶意流量数据建立关键词字典，即使攻击者的攻击载荷中减少了部分关键词，难以做到回避所有的关键词，因此通过本文数据预处理的方式建立模型依然有良好的检测效果。而文献[7]方法面对此类攻击流量，预设的恶意流量关键词很容易被攻击者通过混淆、语法替换的方式绕过，例如“select”变为“se/**/lect”使关键词 select 无法被捕获，使模型检测能力下降。基于字符分割的方法由于产生维度较大并且忽略了本身关键词之间的语义关联，所以当攻击者对攻击数据进行高度混淆时，模型检测能力随之下降。文献[6]方法在处理存在大量特殊符号的无字母混淆攻击数据时，很容易忽略关键信息，例如“(~%8F%97%8F%96% 91%99%90)();”为“phpinfo()”；混淆后的攻击载荷，模型检测能力因此下降。

由表 7 可知，本文方法在加噪的情况下始终优于不加噪的情况。然而，加噪模型和不加噪模型在训练集上的准确率差异较小，而在新型在野数据集上的测试结果中不加噪模型的准确率远低于加噪模型。由此可见，加噪模型可以较好地适应新样本，保证模型在训练样本和测试样本中的性能相对稳定，泛化能力较强，说明加噪操作可有效改善模型的泛化能力。

综上所述，本文方法能够通过少量数据集自动提取恶意流量关键词，并以此建立关键词库用于机器学习的模型训练，所得模型的恶意流量检测识别准确率高，整体性能强，同时有效降低了模型过拟合的风险，提高了模型恶意流量检测的泛化能力。

5　结束语

本文研究了训练集加噪的数据预处理方法对恶意流量检测的影响，不同于文献[6]将特殊字符作为语句分割依据，将分割所得字符串直接使用 Word2vec 提取词向量的方法。本文将特殊字符对语句进行分割依据，将分割所的字符串提取并建立关键词字典，基于关键词字典回避，选择对训练集的非关键词字符进行随机化的加噪处理，最后将噪声数据与原始训练数据混合，利用卷积神经网络对恶意流量进行检测识别。

本文使用相同的训练集和测试集，对训练集不同加噪比例以及不同数据预处理方法进行比较，结果表明在加噪比例为 10%时本文方法有更好恶意流量检测性能。本文方法提高了恶意流量检测的准确率同时降低误报率并在优化了泛化能力，使整体性能得到较大提升。

在接下来的工作，将会尝试提取更多的有效特征以及尝试更多样的加噪方法，例如合理引入高斯加噪的方法，优化现有的深度学习模型，增加网络深度，以达到更好的检测分类效果。同时，尝试收集更多复杂流量样本，使用更大数据集进行实验，以进一步提高准确率、降低误报率。

参考文献：

[1] 谢逸, 余顺争. 基于Web用户浏览行为的统计异常检测[J]. 软件学报, 2007, 18(4): 967-977.
XIE Y, YU S Z. Anomaly detection based on Web users' browsing behaviors[J]. Journal of Software, 2007, 18(4): 967-977.

[2] ZHANG X, ZHAO J, LECUN Y. Character-level convolutional networks for text classification[C]//Advances in neural information processing systems. Massachusetts: MIT Press, 2015: 649-657.

[3] LU X, ZHENG B, VELIVELLI A, et al. Enhancing text categorization with semantic-enriched representation and training data augmentation[J]. Journal of the American Medical Informatics Association, 2006, 13(5): 526-535.

[4] ZOLOTUKHIN M, HÄMÄLÄINEN T, KOKKONEN T, et al. Analysis of http requests for anomaly detection of Web attacks[C]//2014 IEEE 12th International Conference on Dependable, Autonomic and Secure Computing. Piscataway: IEEE Press, 2014: 406-411.

[5] PARK S, KIM M, LEE S. Anomaly detection for HTTP using convolutional autoencoders[J]. IEEE Access, 2018, 6: 70884-70901.

[6] YU Y, LIU G, YAN H, et al. Attention-based Bi-LSTM model for anomalous HTTP traffic detection[C]//2018 15th International Conference on Service Systems and Service Management. Piscataway: IEEE Press, 2018: 1-6.

[7] YANG W, ZUO W, CUI B. Detecting malicious URLS via a keyword-based convolutional gated-recurrent-unit neural network[J]. IEEE Access, 2019, 7: 29891-29900.

[8] ARZHAKOV A V, TROITSKIY S S, VASILYEV N P, et al. Development and implementation a method of detecting an attacker with use of HTTP network protocol[C]//2017 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering. Piscataway: IEEE Press, 2017: 100-104.

[9] XU F, PAN H, CAO Z, et al. Identifying malware with HTTP content type inconsistency via header-payload comparison[C]//2017 IEEE 36th International Performance Computing and Communications

Conference. Piscataway: IEEE Press,, 2017: 1-7.
[10] TORRANO-GIMÉNEZ C, PEREZ-VILLEGAS A, ALVAREZ MARANÓN G. An anomaly-based approach for intrusion detection in Web traffic[J]. Journal of Information Assurance Security,2010, 5(4):446-454.
[11] TAX D M J, DUIN R P W.Support vector data description[J]. Machine learning, 2004, 54(1): 45-66.
[12] THANG T M, KIM J. The anomaly detection by using DBSCAN clustering with multiple parameters[C]//2011 International Conference on Information Science and Applications. Piscataway: IEEE Press, 2011: 1-5.
[13] CHORAŚ M, KOZIK R. Machine learning techniques applied to detect cyber attacks on Web applications[J]. Logic Journal of the IGPL, 2015, 23(1): 45-56.
[14] KRUEGEL C, VIGNA G. Anomaly detection of Web-based attacks[C]//Proceedings of the 10th ACM conference on Computer and communications security. New York: ACM Press, 2003: 251-261.
[15] CORONA I, TRONCI R, GIACINTO G. SuStorID: a multiple classifier system for the protection of Web services[C]//Proceedings of the 21st International Conference on Pattern Recognition. Piscataway: IEEE Press, 2012: 2375-2378.
[16] RINGBERG H, SOULE A, REXFORD J, et al. Sensitivity of PCA for traffic anomaly detection[C]//ACM SIGMETRICS Performance Evaluation Review. New York: ACM Press, 2007, 35(1): 109-120.
[17] AL-OBEIDAT F, EL-ALFY E S M. Hybrid multicriteria fuzzy classification of network traffic patterns, anomalies, and protocols[J]. Personal and Ubiquitous Computing, 2019, 23(5-6): 777-791.
[18] ERFANI S M, RAJASEGARAR S, KARUNASEKERA S, et al. High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning[J]. Pattern Recognition, 2016, 58: 121-134.
[19] DU M, LI F, ZHENG G, et al. Deeplog: anomaly detection and diagnosis from system logs through deep learning[C]//Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. New York: ACM Press, 2017: 1285-1298.
[20] ZHANG M, LU S, XU B. An anomaly detection method based on multi-models to detect Web attacks[C]//2017 10th International Symposium on Computational Intelligence and Design. Piscataway: IEEE Press, 2017, 2: 404-409.
[21] CRETU-CIOCARLIE G F, STAVROU A, LOCASTO M E, et al. Adaptive anomaly detection via self-calibration and dynamic updating[C]//International Workshop on Recent Advances in Intrusion Detection. Berlin: Springer, 2009: 41-60.
[22] WHITESON S, TANNER B, TAYLOR M E, et al. Protecting against evaluation overfitting in empirical reinforcement learning[C]//2011 IEEE symposium on adaptive dynamic programming and reinforcement learning. Piscataway: IEEE Press,2011: 120-127.
[23] JIN Y, XIE J, GUO W, et al. LSTM-CRF Neural Network With gated self-attention for Chinese NER[J]. IEEE Access, 2019, 7: 136694-136703.
[24] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. Imagenet classification with deep convolutional neural networks[C]//Advances in neural information processing systems. Massachusetts: MIT Press,2012: 1097-1105.
[25] KIM Y. Convolutional neural networks for sentence classification[J]. arXiv preprint arXiv:1408.5882, 2014:1746-1751.
[26] LECUN Y, BOTTOU L, BENGIO Y, et al. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86(11): 2278-2324.
[27] HAN J, KAMBER M. Data mining: concepts and techniques[M]. Berlin: Morgan Kaufmann Publishers, 2000.

[作者简介]

陈铁明（1978– ），男，浙江诸暨人，博士，浙江工业大学教授、博士生导师，主要研究方向为网络空间安全、大数据分析。

金成强（1995– ），男，浙江温州人，浙江工业大学硕士生，主要研究方向为信息安全。

吕明琪（1981– ），男，浙江杭州人，博士，浙江工业大学副教授，主要研究方向为数据挖掘与普适计算。

朱添田（1992– ），男，浙江慈溪人，博士，浙江工业大学讲师，主要研究方向为网络安全、系统安全。