文章编号:1674-2974(2020)08-0031-11 DOI:10.16339/j.cnki.hdxbzkb.2020.08.004

SQL 注入行为实时在线智能检测技术研究

李铭[1],邢光升[2],王芝辉[2†],王晓东[2]
(1. 国防科技大学 智能科学学院,湖南 长沙 410073;
2. 国防科技大学 计算机学院,湖南 长沙 410073)

摘 要:为了解决传统手段在实时高速网络流量环境下 SQL 注入行为检测准确度和效率之间无法达到较好平衡的问题,提出一种基于深度学习模型的 SQL 注入行为实时在线检测的方法,构建了以卷积神经网络为基础并引入快速傅里叶变换层的复合检测网络模型 SQLNN,并以此模型为基础提出 SQL 注入行为在线检测与自适应训练框架.该框架对于 SQL 注入语句的检测正确率达到了 99.98%,每秒可完成对一万条左右含有 SQL 语句的数据包的检测,满足了 SQL 注入攻击的实时在线检测对检测准确度和效率的要求.

关键词:SQL 注入;实时检测;卷积神经网络;快速傅里叶变换
中图分类号:TP391 **文献标志码**:A

Research on Real-time Online Intelligent Detection Technology of SQL Injection Behavior

LI Ming[1], XING Guangsheng[2], WANG Zhihui[2†], WANG Xiaodong[2]
(1. College of Intelligence Science and Technology, National University of Defence Technology, Changsha 410073, China;
2. College of Computer Science and Technology, National University of Defence Technology, Changsha 410073, China)

Abstract: In order to solve the problem that traditional methods cannot achieve a good balance between the accuracy and efficiency of SQL injection behavior detection in the real-time high-speed network traffic environment, this paper proposes a method for real-time detection method of SQL injection behavior based on deep learning construction model, and constructs a detection network model called SQLNN based on Convolutional Neural Networks (CNN) and introduces a fast Fourier transform layer. Based on this model, an online detection and adaptive training framework for SQL injection behavior is proposed. For our detection framework, the detection accuracy of the SQL injection statements reaches 99.98%, and it can detect about 10 000 packets containing SQL statements per second. Therefore, it can satisfy the requirements of real-time online detection of SQL injection attacks for detection accuracy and efficiency.

Key words: SQL injection; real-time detection; Convolutional Neural Networks (CNN); Fast Fourier Transformation(FFT)

* **收稿日期**:2019-07-18
作者简介:李铭(1972—),男,山东泰安人,国防科技大学博士研究生
† 通讯联系人,E-mail:wangzhihui13@nudt.edu.cn

随着互联网技术的快速发展，各种网络攻击手段层出不穷,SQL 注入攻击作为网络安全领域中最常见的攻击方式已有十多年的历史，根据开放 Web 应用安全项目组织（Open Web Application Security Project, OWASP）正式发布的十大 Web 应用安全风险报告,SQL 注入已经成为对 Web 业务系统威胁最严重的行为之一. 攻击者常常利用网络应用漏洞,渗透并进一步获取后台服务器控制权，然后执行篡改页面、窃取数据等恶意操作. SQL 注入方式主要是通过构建恶意的 SQL 语法组合作为参数输入 Web 应用程序，应用程序执行 SQL 语句时会同步执行恶意操作,进而发生 SQL 注入攻击. 如果应用程序代码执行过程中将含有攻击者输入的恶意字符串进行存储或传递,也会导致 SQL 注入的发生. SQL 注入有多种方式,按照构造参数类型划分主要有 Get 注入、POST 注入、Cookie 注入、宽字节注入、基于时间注入、布尔型注入、报错型注入、联合查询注入、多语句查询注入等. 虽然注入类型比较多，但是所有的注入类型 SQL 语句均需遵循相关语法并且含有特定的关键词或字符,这就为 SQL 注入行为检测提供了基础条件.

大数据时代,网络信息呈现井喷式增长,在通信主干网和端到端的实时线路上网络数据均呈现高速、大流量的特点,对数据的实时处理提出更高要求. 对于实时在线网络安全检测设备来说，必须确保能够及时发现安全威胁,尽可能检测出所有可疑行为,保证后端设备安全，在此前提下可以适当容忍误检情况的发生.这首先决定了安全检测设备需采取有效的手段进行流量降速，目前业界常规做法是基于特定字符(串)或流量特征对海量数据进行高速筛选;其次是安全设备中的检测模型不宜太复杂，同时又要保证检测的有效性，在保证对可疑行为高检测率的前提下尽量减少误检情况的发生. 目前通用的实时在线安全检测设备大部分是以正则匹配的关键字符串过滤为基础，该类模型通常部署于基于深度包检测技术(Deep Packet Inspection, DPI)的硬件平台上，可以做到基于批量加载规则的实时大流量数据在线检测,但是该类设备误检率较高.

近几年人工智能技术蓬勃发展,在图像处理、语言文本识别等方面以深度学习为基础的相关处理技术有着卓有成效的应用. 网络实时流量中的 SQL 语句具有网络数据和脚本语言双重特性，可以借鉴图像和文本处理中的思路.本文提出一种基于深度学习构建模型对 SQL 注入行为进行实时检测的方法,构建了以卷积神经网络为基础，引入快速傅里叶变换层的复合检测网络模型 SQLNN,并以此模型为基础提出基于 SQLNN 模型的 SQL 注入行为检测模型实时在线检测及迭代训练框架.

本文第一部分梳理了国内外 SQL 注入攻击行为检测技术相关研究以及其他网络脚本或网络数据检测技术的相关研究；第二部分设计了 SQL 注入行为检测模型实时在线检测及迭代训练框架，满足模型实时在线检测的相关需求，同时使模型能够在不影响实时在线运行的前提下实现迭代更新，有效应对网络链路变化或者各类 SQL 脚本协议更新后带来的数据类型和特征的变化；第三部分介绍了相关训练测试数据集的来源、特点和数据预处理过程;第四部分详细介绍 SQLNN 模型设计有关内容;第五部分对 SQLNN 模型与其他模型的实验对比测试的内容和方法进行阐述,并对实验结果进行梳理总结.

1 相关研究

针对 SQL 注入攻击检测和防范，国内外均有大量研究. 王丹等[1]介绍了 SQL 注入式安全漏洞分类,总结了漏洞成因,阐述了相关检测关键技术. 从研究对象来看，主要是从前后端的相关应用程序源码和包含 SQL 注入语句的网络数据两个方面开展相关研究. 对于实时在线检测来说,更多的是基于数据开展研究,因此本文只关注基于数据层面的研究.

1.1 传统检测方法

在传统检测方法上，主要有基于关键字符串构建相应的特征表达式或者模型，基于 SQL 语义语法规则构建模型两种方法对 SQL 注入行为进行检测.

基于关键字符串方面，Halfond 等[2]通过静态分析方法找出程序中可能的注入点，然后针对这些可能的注入点建立合法状态模型，使用静态分析方法建立合法 SQL 语句的模型，动态分析检测在线执行 SQL 语句是否与建立的静态模型一致，但是这种方法只适用于相关的特征应用，类似于建立黑白名单机制,实时在线检测能力受限. 张卓[3]、方爽[4]均是通过正则表达式对网络数据内容进行验证,过滤 SQL 注入攻击的关键字符串，通过分析 SQL 注入行为

语句构造特征,利用正则表达式来分辨 SQL 注入语句和合法语句,该方法实现相对容易,但误检率高,可转化为实时在线检测模型的前置处理模块有效降低网络流量.

基于 SQL 语义语法规则方面,孙义等[5]设计了一种根据模式匹配及序列对比 SQL 注入检测的新思路,使用 Needleman-Wunsch 算法(用于 RNA 和 DNA 序列分析)检测和预防 SQL 注入攻击,通过分析运行过程中含注入命令和正常的命令模式字符串是否能够完全匹配达到检测目的,该方法有效降低检测模型的时空复杂度,但只适用于具体应用中的 SQL 注入数据检测. 石聪聪等[6]使用语法树特征匹配的方法检测 SQL 注入,将执行 Web 应用程序功能的所有 SQL 语句通过语法分析生成 SQL 语法解析树,然后选择出能够识别 SQL 注入行为的特征,构建知识库,通过验证实际执行的 SQL 语句和知识库能否模式匹配成功,进而判断这条 SQL 是否存在注入行为. 王杰 [7] 提出基于抽象语法树的 SQL 注入防御方法,使用 EBNF 范式来描述词法语法规则,通过对生成语法树的规则进行重写并去掉对检测无用节点,构造抽象语法树,基于抽象语法树的 hash 值比对来判断输入语句是否存在 SQL 注入行为. 上述两个研究在相关应用中的 SQL 注入检测中效果较好,但基于语法树特征的检测方式需要研究者花费大量时间构建特征模型,且模型只针对特定应用类型的 SQL 语句,在更新模型时花费时间较多,模型效果好坏取决于研究者对相关知识的理解和模型的设计能力.

1.2 机器学习方法

在基于机器学习的 SQL 注入行为检测方面,国内外也有相关研究,在特征选择上主要针对原始数据特征或数据流信息. Joshi 等[8]使用贝叶斯算法建造了一个区分恶意和非恶意分类查询的分类器,用于训练的数据集包含恶意和非恶意的查询,对正常的 SQL 语句和恶意的 SQL 语句进行建模识别. 张登峰[9]对原始数据从访问者、被访问者、Url 字段 3 个方面进行特征构建和特征化描述,采取贝叶斯模型和决策树模型相结合的集成学习方式,提升了机器学习模型对 SQL 注入行为检测的准确性,实现对大量正常数据快速识别的效果. 上述两种方式均是基于原始数据进行特征提取,检测准确率相对传统方法有所提高,但是对于实时在线检测来说仍然不够. Kim 等[10]通过 SVM 算法对基于数据库日志中内部查询树提取的特征向量进行建模,对已经发生的 SQL 注入行为进行检测,不针对实时在线检测. 赵宇飞等[11]提出一种名为 LFF(Length-Frequency-Feature)的 SQL 注入行为检测方法,针对每条 HTTP 请求检测其请求长度和连接频率,然后采用特征串匹配算法,对请求语句进行匹配,经投票决策,确定 HTTP 请求中是否有 SQL 注入.张志超等[12-13]在研究中均采用多层神经网络对 SQL 注入进行检测,通过选择能够识别 SQL 注入行为的特征合成特征向量,使用多层神经网络对特征进行训练识别. 上述方法虽然检测准确率较高,但是由于其特征是基于对象通联关系或数据载荷的,在不同网络环境和应用中这两类特征会有变化,因此模型只适用于特定网络环境和某些特定应用中.

1.3 深度学习方法

相对于机器学习,深度学习神经网络更有深度,随着训练数据集规模的增大,能够从数据中直接学习到高等级的特征,且不需要人工总结提取. 目前在网络脚本和网络数据识别处理上已有初步应用,尽管在公开的文献中还没有基于深度学习的 SQL 注入行为的实时在线检测方法的研究,但是在 Github 上已经有众多研究源代码公开. 在网络脚本检测方面,方忠庆[14]提出一种 CNN+LSTM 模型对跨站脚本攻击(XSS)进行检测;傅建明等[15]基于卷积神经网络的检测模型对 Webshell 进行检测;付垒朋等[16]基于多类特征提取和 PNN(概率神经网络)模型算法,实现对 JavaScript 恶意脚本的检测;潘司晨等[17]和陈康等[18]均是利用卷积神经网络对恶意 URL 进行识别检测,两个研究差别在于数据的特征提取处理方法上.在网络数据识别处理方面,深度学习技术也得到广泛应用,国内外研究[19-23]均是利用流量数据类图像性质,将原始数据输入,由深度神经网络完成数据表征学习从而实现相关类型数据的检测.上述研究利用深度学习技术构建检测识别模型,均取得很高的准确率. 目前在 Github 开源项目中已经出现基于深度学习的 SQL 注入行为识别检测方面的相关应用和代码,经过代码阅读研究,已有的开源研究均是基于构建词向量的方式产生训练数据,需要构建和训练词向量模型和检测模型两种模型,处理流程较为繁琐.

针对网络脚本和网络数据识别处理还有一种新

兴的方法是使用模糊神经网络，它是神经网络和模糊逻辑结合形成的混合智能系统，通过将模糊系统的类人推理方式与神经网络的学习和连接结构相融合来协同这两种技术.当前已出现许多研究[24-28]将该种神经网络应用于分类、模型识别等 SQL 注入检测相关的问题中，如利模糊神经网络创建专家系统来检测 SQL 注入等,但当前该方法尚未成熟.

2 基于深度学习的 SQL 注入行为在线检测与自适应训练框架

在网络数据脚本识别检测应用上，采用深度学习技术可以在检测与训练集同特征类型数据脚本时取得很高的准确率，但是由于采用深度学习训练生成的检测模型相比于传统检测模型结构复杂，计算量大,如果直接加载于高速实时网络链路中使用,必然会导致数据拥塞,效率低下. 同时由于网络技术的不断发展，各类网络数据脚本协议也在不断完善修订当中，再加上不同的网络线路中数据类型和数据载荷也不相同，因此个别线路采集的数据也不具有完全代表性，导致训练数据集不可能覆盖相关脚本数据的所有特征类型，可能会造成模型在某些应用场景表现的不够理想，这就需要模型在不同的应用场景时可以迭代训练,以更好地适应不同场景.为了有效解决模型在高速线路中检测效率和模型的迭代训练问题，提高模型在线检测的灵活性和检测数据类型的可扩展性，本文提出一种 SQL 注入行为在线检测与自适应训练框架,如图 1 所示,从流程上可以分为在线检测流程和自适应训练流程.

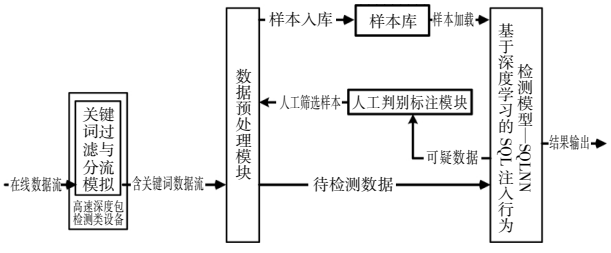

图 1 SQL 注入行为在线检测与自适应训练框架

Fig.1 The online detection and adaptive training framework of SQL injection behavior

2.1 在线检测流程

高速深度包检测类设备可以输入任何端到端实时网络环境数据流或者核心网高速分接链路，该类设备基于硬件实现相关功能，通常基于 FPGA 开发板和专用网络数据处理芯片两种方法，已经有较多成熟方案,并且得到广泛应用,性能较好,可实时处理 10 Gbps 甚至 100 Gbps 以上的网络流量，实时加载检测规则几千条甚至上万条以上,因此在框架中,关键词过滤与分流模块即该类设备加载的相关规则能够应对高速流量环境并确保所有可疑数据进入 SQL 注入行为检测模块，可大幅降低所需后续 SQLNN 模型检测的数据量.

在线检测流程对实时线路数据进行检测，线路数据首先经过告诉深度包检测类设备的关键词,过滤与分流模块进行数据筛选与流量降速，筛选出的含有 SQL 关键词的数据流进入数据预处理模块进行处理后，形成待检测数据进入 SQL 注入行为检测模型——SQLNN 进行检测，将检测判别结果及数据包特征信息(IP 五元组信息等)输出,整个流程如图 2 所示.

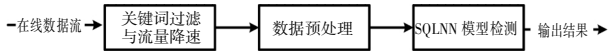

图 2 在线检测流程

Fig.2 Online inspection process

2.2 自适应训练流程

SQL 注入行为检测模型 SQLNN 对预处理后的实时数据进行检测，所有模型判定为 SQL 注入行为数据进入人工判别生成模块，通过人工判别正反例并标注，经过处理后的人工筛选数据经过数据预处理模块后生成训练样本导入到样本库供训练使用,自适应训练流程如图 3 所示.

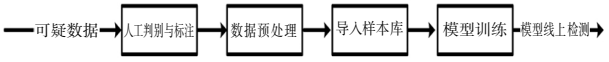

图 3 自适应训练流程

Fig.3 Adaptive training process

自适应训练过程是独立的，不会影响模型在实时线路上的检测运行，模型在线运行时所检测出的可疑数据可以不断丰富样本库，因此不管链路特征如何变化,即使相关数据特征协议发生变化时,只要可疑数据仍具有 SQL 基本特性即相关关键词不变,经过一段时间的自适应训练后，样本库中的数据在当前检测线路中能够具有较好的代表性，多次迭代训练出的模型可实现较好的检测效果. 当然实现上

述目标也要求 SQLNN 模型必须对相关关键字符串具有高度敏感性,在遇到非训练集脚本特征数据时,有一定的过拟合性,以减少可疑数据漏检率,为人工判别标注提供完善的数据.

3 数据预处理

3.1 数据来源

本文中的原始启动训练数据集主要来自于 Github 上基于关键词“SQL detection”搜索得到开源项目的数据样本以及对常用注入攻击工具——明小子 4.3、SQLmap1.3.5 执行注入操作时进行实时抓包分析提取得到的数据样本，同时还有非 SQL 注入脚本的其他网络数据样本，分别将上述两类样本称为 SQL 样本和 normal 样本，从原始数据样本可以看出 SQL 样本有高频特征词以及相对应的上下文语义环境. 对 SQL 样本进行高频特征词统计，总计有脚本 95 602 条,统计中关键词不区分大小写,统计结果见表 1.

表 1　SQL 样本关键词统计

Tab.1　SQL sample keyword statistics

关键词类型	脚本/条	关键词类型	脚本/条
select	85 984	group	4 132
union	69 780	sleep	2 205
all	49 519	set	1 212
and	32 802	drop	1 068
from	22 166	cast	823
or	19 739	binary	337
where	19 093	passwd	301
version	10 833	creat	209
like	9 801	declare	116
when	8 108	exec	61
password	4 732	into	41
table	4 330	length	12
char	4 214	alter	3

在检测模型实际线上运行时，为了有效地降低流速,减轻模型计算负载,会提取含有关键字的数据包导入模型中进行识别检测. 为了提高后续模型训练过程中正负样本之间的对比性,本文删除不含关键词的 normal 脚本得到 normal 原始样本,总计有 57 624 条.

为了测试模型的泛化检测能力——即测试模型对于非训练集中数据特征的可疑 SQL 脚本的识别能力，本文另行在网上搜集其他类型的 SQL 脚本和含有 SQL 关键词并具有类似语法结构的网络脚本来构建泛化测试集，训练集和泛化测试集中 SQL 脚本和 normal 脚本示例如表 2 所示.

表 2　脚本示例

Tab.2　Script example

脚本类型	脚本示例
训练 SQL 脚本	articleuservote.php?id =1&ajax_re quest = 1488931310945% 27% 20LIMIT% 201% 2C1% 20UNION%20ALL%20SELECT%20NULL%2C%20NULL%23 ucavatar.php?uid =-2462% 22% 20UNION% 20ALL% 20SELECT%209358%2C%209358%2C%209358%2C%209358% 2C% 209358% 2C% 209358% 2C% 209358% 2C% 209358% 2C% 209358% 2C% 209358% 2C% 209358% 2C% 209358-%20
训练 normal 脚本	dm% 3D8.qzone.qq.com.hot% 26url% 3D/feeds% 26tt%3D-%26hottag% _cbFunction% 3Dfn_3zedV2u7% 26tokenid% 3Duw8GE-VUMliYCAWolzAEbFnxaB6Xqkav9% 26a cookie% 3Duw8GEVUMliYCAWolzAEbFnxa%26n%3Dcallback
泛化测试 SQL 脚本	GRANT ALL PRIVILEGES ON‘test_%’. * TO‘user_test’@‘localhost’; INSERT INTO ‘items’(‘name’, ‘art’, ‘count_sold’, ‘count_boxes’) values(‘Ручка Ico Omega’,4678,95,525);
泛化测试 normal 脚本	selected works from this collection select 字段 1, 字段 2, 字段 3,count (姓名)from 表 where 字段 3=0(统计为 0 的总数)group by 字段 1,字段 2,字段 3,姓名

3.2 特征提取处理

基于已有研究总结，常见的网络脚本特征提取方法主要分为 4 类：第 1 类是基于承载网络脚本的数据流相关特征进行提取分析，例如数据对象的通联时间、通联次数等;第 2 类是利用网络数据的类图像数据特性，直接利用原始数据按字节输入处理[18-23];第 3 类是利用网络脚本类自然语言的特性,基于脚本分词并转换词向量,方法主要有 one-hot 编码(或类似方法)[17]、Word2Vec[14]等;第 4 类是混合利用前两类方法提取混合特征进行处理. 综合来看,第一类方法需对网络实时数据进行一段积累才能进行

特征提取分析，在端到端的复杂网络环境下数据实体来源较多,数据链路容易发生变化,对特征提取时间窗口的设置提出较高要求，同时也考验研究者对于原始数据和相关数据脚本的解读能力. 第 3 类方法基于数据(词向量)特征进行分析检测,虽然在相关研究[14]中可达到极高的准确率,但是在分词和词向量转换时数据会成倍地膨胀，在流量较大的网络中会对模型的计算能力和实时检测能力提出挑战,同时该类方法在更新迭代时需要重新进行分词，构建模型训练词向量环节较多,更新任务量大.本文特征提取方式主要是基于原始数据的字符编码作为输入特征，同时针对脚本中的相关特性在特征转换前需对脚本进行 URL 编码转换、数字去重替换、脚本去重、长度统一化等操作，同时在模型构建时增加的 FFT 层可对数据特征深度转换处理，在实验测试中能够看出本文的特征提取方式和词向量特征提取方式下模型测试结果的差异.

本文对 SQL 样本和 normal 样本进行特征研究，对每条样本以 400 字节长度进行提取或填充，转换生成 20×20 灰度图，并利用最近邻算法对灰度图进行放大获得一系列样本数据灰度图.

通过分析灰度图，两类样本在直观视觉特征上存在一定差异，但是由于原始样本中有的脚本利用 URL 编码有的没有利用，脚本中有数字表达形式等情况,这些增加了样本中无用特征数量,因此为了强化原始样本的特征,需要针对原始样本进一步处理,主要包括 URL 编码转换、数字去重替换、脚本去重、长度统一化,流程如图 4 所示.

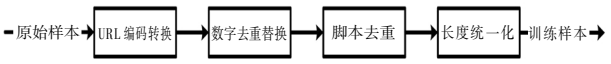

图 4 原始样本处理流程

Fig.4 Original sample processing flow

3.2.1 URL 编码转换

在网络传输中,有些符号在 URL 脚本中是不能直接传递的,需要进行编码,编码的格式为:%加字符 ASCII 值十六进制形式,例如空格的编码值是“%20”,由于此类编码的存在,导致在原始样本中相同含义的脚本有不同的特征表达方式，这种冗余的特征会增加后续模型训练成本.在原始样本中混合有大量的此类编码方式，例如“%22%29%29%29%20UNION%20ALL%20SELECT%206781%2C6781%2C6781%2C6781%2C6781%2C6781%2C6781%23”，因此需要对该类编码格式进行解码,减少样本特征数量.

3.2.2 数字去重替换

样本中存在大量的十进制或十六进制数字,在永真式注入构造如“1=1”可以替换成任意数字,同样可以达到注入效果,因此为了更好地提取样本特征,必须相关数字字段用特定的字符编码替换. 如“union select1,2,3, 4, 5, 6, 7, 8, 9,table_name,11,12,13 from information_schema.tables”“(0×7 179 647 371, 0× 6141534f415555665645,0× 717a687371),NULL, NULL,NULL,NULL,NULL”,该脚本片段中十进制数字部分用“DD”替换,十六进制部分用“FF”替换.上述脚本片段经转换后变成“union select DD,DD,DD, DD,DD,DD,DD,DD,DD,table_name,DD,DD,DD from in formation_schema.tables”“(DD×FF,DD×FF, DD× FF),NULL,NULL,NULL,NULL,NULL”形式.

3.2.3 脚本去重

经过上述两个步骤处理后，原始样本中会存在脚本重复,需要进行脚本去重,去重后 SQL 样本有脚本 76 666 条,normal 样本有脚本 38 471 条.

3.2.4 长度统一化

深度学习模型原始输入特征必须长度固定,因此需要对样本中的脚本长度进行统一，本文对 SQL 样本中的脚本长度分布进行统计,结果如图 5 所示.

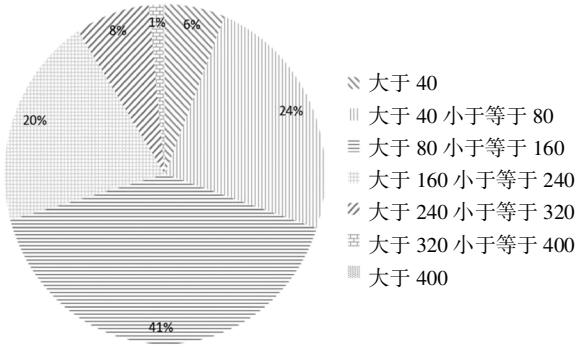

图 5 SQL 脚本长度统计

Fig.5 SQL script length statistics

基于统计结果,分别选取 80、240、400 长度作为样本长度统一化基准，分别生成相应长度样本以便后续实验对比测试,在统一化过程中,会对长度不够的样本执行补“0”操作,对过长的样本执行截取操作，以滑动窗口的形式将脚本数据分别置于不同位置,同时要删除不含关键词的脚本.例如“union select DD,DD”含 18 个字符(空格算 1 个字符),进行长度为 14 的统一操作后得到脚本序列:“union select D”“nion select DD”“ion select DD,”“on select DD,D”“n

select DD,DD”由于脚本中均包含关键字 select 因此不需要执行无关键词脚本删除操作. 进行长度为 22 的统一操作后得到脚本序列:“union select DD,DD0000”“0union select DD,DD000”“00union select DD,DD00”“000union select DD,DD0”“0000union select DD,DD”.

长度统一化后，将训练集中的 SQL 样本与 normal 样本打标签,SQL 脚本标注为“0”,normal 脚本标注为 “1”. 然后分别将两类样本按照 10∶1 分成训练样本集合和测试样本集合，其中训练样本集合中的脚本转换成 tensorflow 能够高效读取的 tfrecord 格式，训练集中的两类样本主要作为模型训练和测试使用.对泛化测试集执行相同上述预处理操作但不需要转换 tfrecord 格式,供后续泛化测试使用.

4 SQLNN 模型设计

4.1 快速傅里叶变换层

原始训练脚本数据首先要经过初始变换，将原始字符数据流由数值 0~255 之间整数变成 0~1 之间的浮点数类型. 为了更好的表征相邻字符之间的关系，本文引入了图像处理领域经常用到的快速傅里叶变换(FFT)将原始类似于时域上的数值数组转换为频域上的数值.快速傅里叶变换(FFT)是离散傅里叶变换(Discrete Fourier Transformation,DFT)的快速算法,基本方法是把原始的 N 长度序列,分解成短序列 $x(n)$,然后利用 DFT 公式 ,求出相应的 DFT 并进行适当组合 $X(K)$,达到减少计算和简化结构的目的[29]. 计算机进行 DFT 时，输入值是时域的信号值,输入值的数量决定了转换计算规模. 变换后输出的频域值数量相同. 该方法已经在 tensorflow 中得到封装,可以直接调取使用,在整体模型中 FFT 层实现了对数据特征进一步提取的作用.

4.2 模型整体设计

SQLNN 模型是一种双层 CNN 模型,输入分别经过输入层和 FFT 层转换后的一维数组，按照二分类方法对 SQL 注入样本和 normal 样本进行训练. 本文中模型使用 tensorflow-1.4,python3.5 搭建及代码实现,模型结构如图 6 所示,各部分概况及数据维度变化情况见表 3,由于输入数据长度在对比测试中分为 80、240、400,表中以 length 代替.

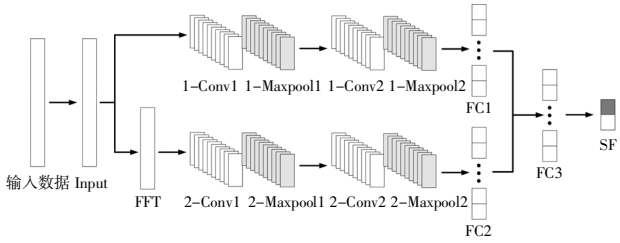

图 6　SQLNN 模型架构

Fig.6　SQLNN model structure

本文中模型的起始输入是经过字符编码直接转换后形成数组,维度为[1,length]. 模型两层结构在经过一系列卷积池化操作后，各自经过全连接层后执行数据拼接操作,将 2 个[1,1 024]维度数组拼接成[1,2 048]维度数组,拼接后的数组输入到全局全连接层中.在 Softmax 层中,假设输入是 $[x_1,x_2]$,通过 softmax 层后输出变为$[y_1,y_2]$且 $y_1+y_2=1$. 本文设置阈值为 0.5,当 $y_1>0.5$ 时,则判定原始样本为 SQL 注

表 3　SQLNN 模型各阶段数据处理概况

Tab.3　Overview of data processing at each stage of the SQLNN model

名称	功能	尺寸大小	步长	输入数据维度	输出数据维度
Input	模型的起始输入	[1,length,1]	—	[length]	[1,length,1]
FFT	FFT 转换	[1,length,1]	—	[1,length,1]	[1,length,1]
1-Conv1 2-Conv1	卷积层,Relu	[1,5,1,32]	[1,1,1,1]	[1,length,1]	[1,length,32]
1-Maxpool1 2-Maxpool1	池化层	2×2	—	[1,length,32]	[1,length/2,32]
1-Conv2 2-Conv2	卷积层,Relu	[1,5,1,64]	[1,1,1,1]	[1,length/2,32]	[1,length/2,64]
1-Maxpool2 2-Maxpool2	池化层	2×2	—	[1,length/2,64]	[1,length/4,64]
FC1 FC2	全连接层	[length×16,1 024]	—	[1,length/4,64]	[1,1 024]
FC3	全连接层	[2 048,2]	—	[1,2 048]	[1,2]
SF	Softmax 分类	2	—	[1,2]	[1,2]

入样本,否则视为正常.

在训练中,每次迭代的 batch 大小为 128,模型卷积层采用 Relu 激活函数, 利用 Adam 算法对参数进行反向传播优化,在每层结构中的全连接层(FC1、FC2)采用 Dropout 策略,参数为 0.5,使每轮训练中一半神经网络单元失效,防止产生过拟合,由于模型结构较浅,在其他层网络单元采取全学习策略.

在 1.3 节的介绍中,有很多研究将深度学习方法应用于网络脚本和网络数据识别处理中, 用来检测各种类型的攻击. 与 1.3 节中的模型相比, 本文所提方法的创新性主要有以下几点:

1)根据 SQL 脚本以及攻击脚本的特征对数据进行预处理,加入了 URL 转换和数字去重替换等预处理过程.

2)对数据处理过程中加入快速傅里叶变换层,利用该层可以进一步提取数据的特征,而其他深度学习方法则是直接利用词向量等方法处理脚本中的文本,然后以词向量为基础对模型进行训练.

3)引入了自适应训练模块,将模型检测出的可疑数据经过人工识别后加入到样本库中, 对模型进行迭代训练,适应网络中数据的变化.

5 实验对比测试

5.1 SQLNN 模型效果检测

一个有效的数据检测模型应该在已知范围的检测数据范围内具有极高的准确率, 在未知的泛化数据集上尽量有较高的准确率, 如果达不到要求即不能很好地识别出正反例数据, 也应该将所有可疑的数据识别出来即保证有较高的召回率. 总体上看,在评价标准上应该遵循准确率和召回率均衡, 召回率优先.

本文采用准确率 A、召回率 R、F_1 值对相关模型进行测试评估,准确率和查全率模型的识别效果,F_1 是根据准确率 A 和召回率 R 计算得到的综合评价指标[30],三者值越高越好,相关参数值见表 4.

首先确定模型在不同输入数据长度下的表现,确定最适合的长度,对输入数据 80、240、400 长度分别使用测试集和泛化测试集进行检测, 得到测试集下的准确率 A,泛化测试集下的准确率 A、召回率 R. 结果如图 7 所示.

为了减少测试过程中的随机误差, 图 7 所示结果均为多次测试后所获得的平均值, 其方差如表 5 所示.

表 4 相关参数值

Tab.4 Related parameter value

名称	含义
TP	数据集中被模型预测为 SQL 注入样本,实际是 SQL 注入样本的脚本数量
FP	数据集中被模型预测为 SQL 注入样本,实际是正常样本的脚本数量
FN	数据集中被模型预测为正常样本,实际是 SQL 注入样本的脚本数量
TN	数据集中被模型预测为正常样本,实际是正常样本的脚本数量
准确率 A	$\frac{TP + TN}{TP + FP + FN + TN}$
召回率 R	$\frac{TP}{TP + FN}$
F_1	$\frac{2AR}{A + R}$

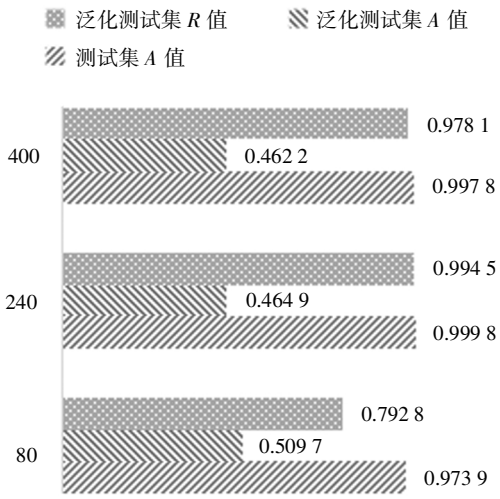

图 7 SQL 脚本长度测试结果

Fig.7 SQL script length test results

表 5 不同长度下结果方差

Tab.5 Variance of results at different lengths

脚本长度(字符)	测试集准确率/10^{-6}	泛化测试集准确率/10^{-6}	泛化测试集召回率/10^{-6}
80	8.1	217.4	95.3
240	0.008	168.3	2.9
400	3.7	180.9	5.4

通过观察上述结果,在测试集下,长度为 240 时准确率最高,达到 99.98%.对于泛化测试集,在 3 种长度下准确率都不高, 但是在长度 240 下召回率很高,达到 99.45%,因此选择 240 作为脚本输入长度.

对以上结果进行分析，通过观察 SQL 脚本长度的统计结果可以发现，绝大部分的脚本长度都在 80~240 字符的范围内,当长度固定为 80 时,大部分脚本都会因为截取操作而导致很多信息丢失；而当脚本长度固定为 400 时，很多脚本又会因为长度不够,添加了过多的冗余信息. 因此,将长度固定为 240 时能够适应大多数脚本，既有效保留了脚本中的有用信息,同时也不会有过多的冗余信息.

为了对比本文使用的数据特征提取处理以及模型构建的有效性，基于 Github 开源项目搭建基于 Word2Vec 模型提取词向量特征的方式对 CNN、LSTM、GRU、MLP、SVM 模型进行对比训练测试,通过针对不同模型层数和模型参数大量的调优实验,获得相应类型模型的最佳测试结果. 其中 CNN 模型也使用了与 SQLNN 相同的两个卷积层;LSTM 模型尝试了单向和双向两种结构，双向 LSTM 的效果更好,当模型的隐藏单元数为 512 时取得的效果最好;GRU 模型的结构与 LSTM 模型相同，只是将 LSTM 单元改为了 GRU 单元. 另外,本文也尝试了将 FFT 层应用于上述各类模型,但加了 FFT 层并没有改善模型的检测效果,只有在 CNN 中使得模型效果得到了改进,构成了本文中提出的 SQLNN 模型.

对比 SQLNN 模型与上述模型在测试集下的准确率,泛化测试集下的准确率 A、召回率 R,计算 F_1 值. 结果如图 8 所示.

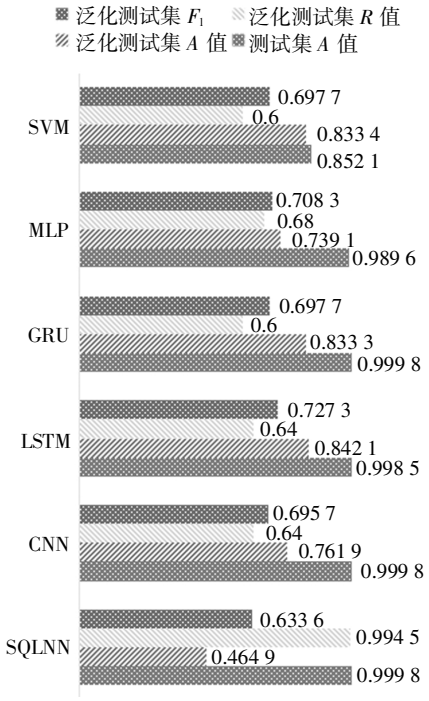

图 8　模型对比测试结果

Fig.8　Model comparison test results

对于上述结果,同样是在多次测试后选取了平均值,其方差如表 6 所示.

表 6　不同模型下结果方差

Tab.6　Variance of results under different models　10^{-6}

模型	测试集准确率	泛化测试集准确率	泛化测试集召回率	泛化测试集 F_1 值
SVM	17.4	38.2	35.2	47.6
MLP	2.6	24.1	26.1	49.1
GRU	0.01	143.7	15.7	69.8
LSTM	0.2	95.6	58.6	96.4
CNN	0.1	113.7	13.6	31.3
SQLNN	0.008	168.3	2.9	56.9

在测试集下,模型 SQLNN 和基于词向量特征提取的模型 CNN、LSTM、GRU 准确率高，超过 99.5%. 在泛化测试集下,本文所提出的 SQLNN 模型相对于其他模型准确率较低,F_1 值相对较低，但 SQLNN 模型的召回率很高,达到 99.45%.

SQLNN 模型优于其他模型的主要原因有以下 2 个方面:

1)在对攻击行为进行检测时,应尽可能不漏掉攻击脚本,SQLNN 模型可以确保在运行时不漏掉可疑 SQL 注入行为,保证应用运行的安全性.

2)SQLNN 模型虽然在泛化测试集下的正确率不高. 在泛化数据集上 SQLNN 相比其他模型误检率较高，但是其可以确保检测系统在运行时不漏掉可疑 SQL 注入行为，其他模型会存在比较明显的漏检情况. 此外,在系统运行过程中,SQLNN 模型检测出的可疑 SQL 注入脚本会由人工进行进一步判别,并将结果加入到样本库中，作为新数据对模型进一步训练从而适应新数据,不断提升 SQLNN 模型对于新数据的检测效果.随机抽取部分泛化数据对模型进行二次训练后,模型对泛化数据的检测效果明显提升,达到了 97.58%. 虽然其他模型经过二次训练后效果也会有提升，但是它们因为召回率低而不能有效检测出可疑 SQL 注入脚本,因此 SQLNN 模型能够更好的适用于 SQL 注入行为实时在线检测.

此外，在实时线路中，当前最常用的方法有两种,第一种是关键词匹配过滤,该方法的缺点是正确率比较低. 第二种方法则是利用正则函数加流量分析，该方法虽然正确率比关键词匹配过滤方法高很多,但是只能针对特定的注入攻击类型检测,不具有

广泛代表检测能力. 本文所提的 SQLNN 模型检测正确率较高，对于各种注入攻击类型的脚本都可以检测,因此优于其他方法.

5.2 实时性效果检测

以 10 Gbps 的大流量线路为例，峰值情况下每秒大约有 2 000 000 个数据包. 对于线路上的数据包,并不全包含有 SQL 语句,因此首先需要筛选出含有 SQL 语句的数据包.

对于数据包的筛选,包含如下两步:

第一步是通过网络协议进行筛选，对于包含 SQL 语句的脚本,所使用的协议均为 TCP 协议.本文利用两条 10 Gbps 线速环境的线路进行实验，利用网络协议进行筛选后，两条线路每秒钟所包含的 TCP 数据包数量约为 10 000 条.

第二步则是利用关键词对第一步的筛选结果进行进一步的筛选,筛选出包含有 SQL 关键词的数据,作为需要 SQLNN 模型检测的数据.

经过上述两步，两条线路中所包含的关键词命中包分别为 100 条左右与 30 条左右. 利用高速深度包检测类设备完成上述操作所需的时间为秒级.完成数据筛选后，对于实时线路中被检测的数据仅包含 URL 转换和长度统一化两个操作，且需要处理的数据的数量级在 10^2 级,因此对数据的筛选和处理所需的时间为秒级,利用 SQLNN 模型对 10^2 级数据进行检测所需的时间可以忽略不计，因此本文提出的检测框架完全可以达到实时检测的效果，在效率上满足检测要求.

6 结 论

本文提出一种基于深度学习的 SQL 注入行为在线实时检测方法,构建了检测模型 SQLNN,该模型引入 FFT 层丰富输入数据表达特征，在测试集上可以达到与基于 Word2Vec 词向量转换特征表达方法相当的高准确率，在泛化测试集上召回率有明显优势，可以有效减少在面临新 SQL 注入形式时漏检情况发生. 同时本文以 SQLNN 模型为基础提出 SQL 注入行为在线检测与自适应训练框架，满足 SQL 注入攻击实时在线检测对检测准确度和效率的要求,并可实现模型的迭代演进，该框架对其他网络脚本类数据检测模型的训练及在线运行也有一定的参考价值.未来将进一步优化模型结构,使模型能够在泛化测试集上有更好的表现，并逐步扩展到其他网络脚本类攻击行为检测，同时基于检测结果加入后续数据流量特征分析以进一步分析可疑行为.

参考文献

[1] 王丹,赵文兵,丁治明. Web 应用常见注入式安全漏洞检测关键技术综述 [J]. 北京工业大学学报,2016,42(12):1822—1832.
WANG D,ZHAO W B,DING Z M. Review of detection for injection vulnerability of web applications [J]. Journal of Beijing University of Technology,2016,42(12):1822—1832. (In Chinese)

[2] HALFOND W G J,ORSO A. AMNESIA: analysis and monitoring for neutralizing SQL-injection attacks [C]//The Proceedings of the 20th IEEE/ACM International Conference on Automated Software Engineering. Long beach,California: ACM,2005:174—183.

[3] 张卓. SQL 注入攻击技术及防范措施研究[D]. 上海:上海交通大学信息安全工程学院,2007:52—64.
ZHANG Z. SQL injection attack techniques and countermeasures analysis [D].Shanghai: School of Information Security Engineering,Shanghai Jiaotong University,2007:52—64. (In Chinese)

[4] 方爽. 基于特征匹配的 WEB 应用防火墙的研究与实现 [D].合肥:安徽大学计算机科学与技术学院,2014:36—47.
FANG S. Research and implementation of web application firewall based on feature matching [D]. Hefei: School of Computer Science and Technology,Anhui University,2014:36—47. (In Chinese)

[5] 孙义，胡雨雳，黄皓. 基于序列比对的 SQL 注入攻击检测方法[J]. 计算机应用研究,2010,27(9):3525—3528.
SUN Y,HU Y J,HUANG H. SQL injection attack detection method based on sequence alignment [J].Computer Application Research,2010,27(9):3525—3528. (In Chinese)

[6] 石聪聪,张涛,余勇. 基于语法树特征匹配的 SQL 注入防护方法研究与实现[C]//第三届计算智能与工业应用国际学术研讨. 武汉:电气电子工程师协会,2010:192—196.
SHI C C,ZHANG T,YU Y. Research and Implementation of SQL injection protection method based on syntax tree feature matching [C]//Proceedings of 2010 The 3rd International Conference on Computational Intelligence and Industrial Application. Wuhan: IEEE,2010: 192—196. (In Chinese)

[7] 王杰. 基于抽象语法树的 SQL 注入防御研究[D].武汉:武汉邮电科学研究院,2018:28—40.
WANG J. Research on SQL injection defense based on abstract syntax -tree [D]. Wuhan: Wuhan Institute of Posts and Telecommunications,2018: 28—40. (In Chinese)

[8] JOSHI A,GEETHA V. SQL injection detection using machine learning[C]//2014 International Conference on Control,Instrumentation,Communication and Computational Technologies (ICCICCT). Tamilnadu,India:IEEE,2014:1111—1115.

[9] 张登峰. 基于机器学习的 SQL 注入检测[D].重庆:重庆邮电大学计算机学院,2017:31—45.
ZHANG D F. SQL injection detection based on machine learning [D].Chongqing: School of Computer,Chongqing University of Posts and Telecommunications,2017: 31—45. (In Chinese)

[10] KIM M Y,DONG H L. Data-mining based SQL injection attack de-

tection using internal query trees [J]. Expert Systems with Applications,2014,41(11):5416—5430.

[11] 赵宇飞,熊刚,贺龙涛,等. 面向网络环境的 SQL 注入行为检测方法[J]. 通信学报,2016,37(2):88—97.
ZHAO Y F,XIONG G,HE L T,*et al*. SQL injection behavior detection method for network environment [J]. Journal of Communications,2016,37(2):88—97. (In Chinese)

[12] 张志超. 基于神经网络的 SQL 注入式攻击漏洞检测问题的研究与实现[D].北京:北京工业大学计算机学院,2016:17—26.
ZHANG Z C. Research and implementation of SQL injection attack vulnerability detection based on neural network [D]. Beijing: School of Computer,Beijing University of Technology,2016: 17—26. (In Chinese)

[13] 张志超,王丹,赵文兵,等. 一种基于神经网络的 SQL 注入漏洞的检测模型[J]. 计算机与现代化,2016(10):67—71.
ZHANG Z C,WANG D,ZHAO W B,*et al*. A SQL injection vulnerability detection model based on neural network [J].Computer and Modernization,2016(10):67—71. (In Chinese)

[14] 方忠庆. 基于深度学习的跨站脚本攻击检测研究[D].长沙:湖南大学信息科学与工程学院,2018:33—43.
FANG Z Q. Research on cross-site scripting attack detection based on deep learning [D]. Changsha: School of Information Science and Engineering,Hunan University,2018:33—43. (In Chinese)

[15] 傅建明,黎琳,王应军. 基于 CNN 的 Webshell 文件检测[J]. 郑州大学学报(理学版),2019,51(2):1—8.
FU J M,LI L,WANG Y J. Webshell file detection based on CNN [J]. Journal of Zhengzhou University (Science Edition),2019,51(2):1—8. (In Chinese)

[16] 付垒朋,张瀚,霍路阳. 基于多类特征的 JavaScript 恶意脚本检测算法[J]. 模式识别与人工智能,2015,28(12):1111—1117.
FU L P,ZHANG H,HUO L Y. JavaScript malicious script detection algorithm based on multi-class features [J]. Pattern Recognition and Artificial Intelligence,2015,28 (12):1111—1117. (In Chinese)

[17] 潘司晨,薛质,施勇. 基于卷积神经网络的恶意 URL 检测[J].通信技术,2018,51(8):1919—1923.
PAN S C,XUE Z,SHI Y. Malicious URL detection based on convolutional neural network [J]. Communication Technology,2018,51(8):1919—1923. (In Chinese)

[18] 陈康,付华峥,向勇. 基于深度学习的恶意 URL 识别[J].计算机系统应用,2018,27(6):27—33.
CHEN K,FU H Z,XIANG Y. Malicious URL recognition based on deep learning [J].Application of Computer Systems,2018,27(6):27—33. (In Chinese)

[19] TORRES P,CATANIA C,GARCIA S,*et al*. An analysis of recurrent neural networks for botnet detection behavior [C]//Biennial Congress of Argentina (ARGENCON).Buenos Aires,Argentina: IEEE,2016:1—6.

[20] WANG W Y,ZENG X,YE X,*et al*. Malware traffic classification using convolutional neural network for representation learning [C]// The 31st International Conference on Information Networking (ICOIN). Da Nang,Vietnam: IEEE,2017:712—717.

[21] 马若龙. 基于卷积神经网络的未知和加密流量识别的研究与实现[D]. 北京:北京邮电大学网络技术研究院,2017:17—44.
MA R L. Research and implementation of unknown and encrypted traffic recognition based on convolutional neural networks [D]. Beijing:Institute of Network Technology,Beijing University of Posts and Telecommunications,2017:17—44. (In Chinese)

[22] 张路煜,廖鹏,赵俊峰,等. 基于卷积神经网络的未知协议识别方法[J]. 微电子学与计算机,2018,35(7):106—108.
ZHANG L Y,LIAO P,ZHAO J F,*et al*. Unknown protocol recognition method based on convolutional neural network [J]. Microelectronics and Computers,2018,35(7):106—108. (In Chinese)

[23] 王伟. 基于深度学习的网络流量分类及异常检测方法研究[D]. 合肥:中国科学技术大学信息学院,2018:45—67.
WANG W. Research on network traffic classification and anomaly detection method based on deep learning [D]. Hefei: School of Information,University of Science and Technology of China,2018:45—67. (In Chinese)

[24] BATISTA L O,SILVA G A D, VANESSAÙJO V S,*et al*. Fuzzy neural networks to create an expert system for detecting attacks by SQL injection [J]. The International Journal of Forensic Computer Science,2018,13(1):8—21.

[25] SOUZA P V C. Regularized fuzzy neural networks for pattern classification problems [J]. International Journal of Applied Engineering Research,2018,13(5): 2985—2991.

[26] SOUZA P V C,OLIVEIRA P F A. Regularized fuzzy neural networks based on nullneurons for problems of classification of patterns[C]// 2018 IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE).Penang Island,Malaysia: IEEE,2018: 25—30.

[27] SOUZA P V C,GUIMARAES A J,ARAÙJO V S,*et al*. Fuzzy neural networks based on fuzzy logic neurons regularized by resampling techniques and regularization theory for regression problems [J]. Inteligencia Artificial,2018,21(62): 114—133.

[28] MA B H,ZHONG G,CHEN Q N,*et al*. A fuzzy neural network system for architectural foundation selection [C]//Proceedings of the 2nd International Symposium on Asia Urban Geo Engineering. Springer,Singapore, 2018: 651—664.

[29] 虞湘宾,董涛. 一种离散小波变换的快速分解和重构算法[J]. 东南大学学报(自然科学版),2002,32(4):564—568.
YU X B,DONG T. A fast decomposition and reconstruction algorithm of discrete wavelet transform [J]. Journal of Southeast University (Natural Science Edition),2002,32 (4):564—568. (In Chinese)

[30] 王泳, 胡包钢. 应用统计方法综合评估核函数分类能力的研究[J]. 计算机学报,2008,31(6):942—952.
WANG Y,HU B G. Study on comprehensive evaluation of kernel function classification ability using statistical methods [J]. Journal of Computer,2008,31(6):942—952. (In Chinese)