基于 FP-growth 优化 SVM 分类器的跨站脚本攻击检测研究

李仁杰[1] 华 驰[1,2] 鲁志萍[1]
[1](江苏信息职业技术学院物联网工程学院 江苏无锡 214153)
[2](南京航空航天大学计算机科学与技术学院 南京 211106)
(liushaze@qq.com)

The Research of Detection XSS Attack Based on FP-growth Optimized SVM Classifier

Li Renjie[1], Hua Chi[1,2], and Lu Zhiping[1]
[1](*Jiangsu Vocational College of Information Technology*, *Wuxi*, *Jiangsu* 214153)
[2](*College of Computer Science and Technology*, *Najing University of Aeronautics and Astronautics*, *Nanjing* 211106)

Abstract XSS(cross-site scripting) is a Web-based security attack that is one of the most serious threats to Internet security today. By analyzing the principle of XSS attack detection based on SVM (support vector machine) classifier, paper proposes an association detection algorithm (FP-growth) optimizes the XSS attacker detection method, it is verified by experiments that this method can effectively improve the accuracy of XSS detection compared with the common SVM detection method.

Key words machine learning; XSS; SVM; FP-growth; attack; detection method

摘 要 跨站脚本(cross-site scripting, XSS)攻击是一种基于 Web 的安全攻击,是目前影响互联网安全的最为严重的威胁之一.通过对支持向量机(SVM)分类器的 XSS 攻击检测相关技术的研究,提出了一种基于关联分析算法(FP-growth)优化 SVM 分类器的 XSS 攻击自动检测策略,实验结果证明该策略有效提高了 XSS 攻击检测精准度.

关键词 机器学习;跨站脚本;支持向量机;关联分析算法;攻击;检测方法

中图法分类号 TP309

跨站脚本(cross-site scripting, XSS)攻击是一种注入式安全攻击,攻击者如果将数据(如恶意代码)注入到其他受信任的 Web 应用程序中,就会发生跨 XSS 攻击,该恶意脚本将被包含在传递给受害者浏览器的动态内容中,受害者将会在浏览器中执行恶意脚本,攻击者此时就可以利用此漏洞

收稿日期:2020-04-16
基金项目:江苏省"333"高层次人才工程项目(苏人才[2016]7号);江苏省高等职业院校高水平骨干专业建设工程项目(苏教高[2017]11号);江苏省"青蓝工程"优秀教学团队项目(苏教师[2017]15号);江苏省高等职业教育产教融合集成平台项目(苏教职〔2019〕31号);江苏省教育科学"十三五"规划项目(C-b/2016/03/10)

盗取用户账号、非法转账、网页挂马、控制受害者的机器进行 DDOS 攻击等，有着极大的安全隐患.

据北京奇虎科技有限公司(奇虎 360)发布的《2019 上半年中国政企机构网络安全形势分析报告》[1]披露，在 2019 年 1—12 月，XSS 漏洞被识别出次数为 51.9 万次，存在此漏洞的网站数为 4.3 万个，属于高危漏洞，且在所有的高危漏洞中，XSS 漏洞的扫出次数和漏洞网站数都是最多的，排名高危漏洞第一[1].

针对 XSS 攻击的防范，传统方法是使用跨站脚本过滤器 XSS-Filter 完成用户输入数据的过滤，该策略主要原理是将包含 XSS 攻击的源地址不断更新到数据库黑名单中，并拒绝黑名单中用户的访问. 这是一种静态的 XSS 攻击防范方法，该策略容易被黑客绕过[2]. 当前国内外信息安全研究人员针对 XSS 攻击检测积极开展了研究. Rietz 等人[3]将机器学习的方法用于 XSS 攻击检测，并基于支持向量机(support vector machines，SVM)算法完成了 XSS 和 CSRF 漏洞检测；Wang 等人[4]将 SVM，PCA，KNN 等机器学习算法进行实验对照，证明了 SVM 算法在 XSS 攻击检测中的优越性；王丹等人[5]基于隐马尔科夫模型(HMM)的攻击向量动态生成和优化方法；赵澄等人[6]提取最具代表性的五维特征并将这些特征向量化，然后这些特征向量化作为 SVM 算法的输入特征完成 XSS 漏洞检测，提高了 SVM 算法的检测准备率. 以上研究都为单一的机器学习算法完成对 XSS 攻击的检测，在前人研究的基础上，本文提出了一种基于关联分析算法(FP-growth)[7]通过分析 XSS 攻击日志中的关联关系，并将这种关系作为 SVM，KNN 等分类算法的特征提取依据之一来完成 XSS 攻击检测.

1 XSS 攻击的分类和特点

XSS 攻击类型一般分为反射型 XSS、存储型 XSS、DOM 型 XSS 这 3 种类型.

1) 反射型 XSS

反射型 XSS 的特点是攻击者通过多种方式诱使受害者点击包含恶意代码的超链接，从而发起反射 XSS 请求[8].

2) 存储型 XSS

存储型 XSS 中，包含恶意 XSS 的 Web 文件往往由攻击者部署在服务器上，当受害者访问该 Web 页面并执行相关操作后就会触发 XSS 攻击.

3) DOM 型 XSS

DOM 型 XSS 与 Web 页面无关，其主要出现在 DOM(文档对象模型)中，受害者访问 Web 应用程序过程中从 DOM 读取数据传送到浏览器. 如果未正确处理数据，则攻击者能够注入将作为 DOM 的一部分存储的有效负载，并在受害者从 DOM 读回数据时执行了攻击者的 XSS 漏洞代码.

2 基于 SVM 算法的 XSS 攻击检测设计

2.1 SVM 算法特点

SVM[3]是机器学习分类算法中使用范围最为广泛的算法之一，是定义在特征空间上间隔最大的线性分类器，对于线性二分类问题有着较好的解决能力，一般应用于模式识别、分类及回归分析中. 其原理是创建一条线或一个超平面，再通过线或超平面将数据分成多个类.

SVM 分类算法原始形式为：

$$\min_{w,b,\xi_i} \frac{1}{2}\|w\|_2^2 + c\sum_{i=1}^{m}\xi_i,$$
$$\text{s.t. } y_i(w\cdot\varphi(x_i)+b) \geqslant 1-\xi_i,$$
$$(i=1,2,\cdots,m),$$
$$\xi_i \geqslant 0(i=1,2,\cdots,m).$$

其中样本为$(x_1,y_1),(x_2,y_2),\cdots,(x_m,y_m)$，$m$ 为样本的个数. w,b 是分离超平面 $w\cdot\varphi(x_i)+b=0$ 的系数，ξ_i 为第 1 个样本的松弛系数，c 为惩罚系数，$\varphi(x_i)$为低维到高维的映射函数.

如果只有二维特征向量，区分普通用户和黑客用户可以用直线来划分，如图 1 所示.

此时解决的是线性划分(linear separable)问题，如果不仅仅有二维特征向量则称为不可线性区分(linear inseparable)问题，这时 SVM 算法需要在 2 个类的数据之间找到分离线(或超平面)；如果遇到不可线性区分的情况，就需升级到更高的维度进行划分，例如二维平面无法划分则需要拓展到三维平面进行划分，XSS 样本数据集是非线性的，需要将其转化为 SVM 算法可以处理的数

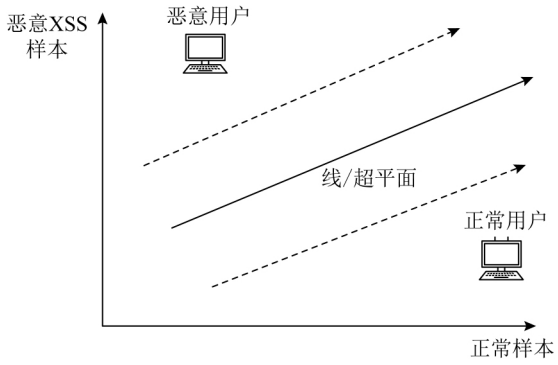

图1 SVM原理图

据集，根据Mercer定理[9]，可以通过核函数将非线性的数据样本映射到高维空间进行分类，从而使得原来的样本空间中非线性可分问题，转化为在特征空间中的线性可分问题. 在本研究中选择了sigmoid核函数.

2.2 传统SVM分类器的XSS攻击检测设计

SVM分类器的数据处理流程如图2所示，收集同样数量的正常Web访问数据和XSS攻击数据，分别对2类日志中的请求数据清洗并进行特征化，将正常数据标记为0，XSS攻击数据标记为1，形成特征数据集，将SVM算法的模型训练传递给分类器进行结果检测，生成训练模型.

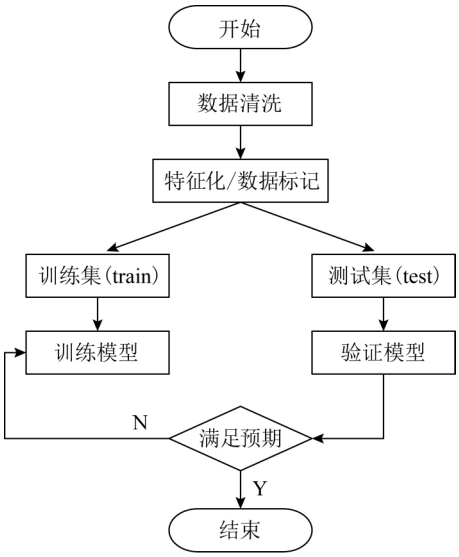

图2 SVM分类器的数据处理流程图

生成的训练模型可直接应用于对XSS攻击的识别，XSS攻击识别的准确度受限于训练模型，训练数据越大对XSS攻击的识别精准度越高.

3 FP-growth优化SVM分类器的XSS攻击检测设计

3.1 FP-growth算法挖掘XSS攻击相关参数

FP-growth算法是比较经典的频繁项集和关联规则挖掘的算法，相比较于Apriori算法对每个潜在的频繁项集都需要扫描数据集判定是否满足支持度，FP-growth算法只需要遍历2次数据集，因此效率会更高. 该算法可以应用于XSS相关参数的对比，从而找出XSS参数之间的关联性.

为了减少I/O次数，FP-growth算法引入了一些数据结构临时存储数据，这个数据结构主要为原始数据集、项头表、FP树.

假设有如图3所示的1组原始数据：

A	B	C	E	F	
A	C	G			
E	I				
A	C	D	E	G	
A	C	E	D	J	
E	J				
A	B	C	E	F	P
A	C	D			
A	C	E	G	M	
A	C	E	G	N	

图3 原始数据集

假设最小支持度为2，第1次读取数据集，得到所有频繁1项集计数，并删除支持度不满足最小支持度的项集，最后将剩余的满足最小支持度的频繁项集按照支持度降序排序. 这样就得到了项头表，如图4所示：

A:8
C:8
E:8
G:5
B:2
D:2
F:2

图4 项头表

第2次读取数据集，对于数据集的每一行剔除非频繁1项集的元素并按照支持度降序排序，经过排序后的数据集如图5所示.

有了项头表、处理后的数据集，就可以开始FP-tree的建立，FP-tree是一种紧凑的数据结构，

A	C	E	B	F
A	C	G		
E				
A	C	E	G	D
A	C	E	G	
E				
A	C	E	B	F
A	C	D		
A	C	E	G	
A	C	E	G	

图 5　处理后的数据集

1 棵 FP-tree 看上去与计算机中的其他树结构相类似，不过它是通过链接相似的元素，而这些被连起来的元素可以看作是一个链表. 将开始时的 FP-tree 设为空(NULL)，并将数据集中的单行样本按照顺序依次插入 FP-tree 中，在此过程中，排序靠前的节点为父节点，靠后的为子节点. 如果有共用的父节点，则对应的共用父节点计数加 1. 插入后，如果有新节点出现，则项头表对应的节点会通过节点链表链接上新节点. 直到所有的数据都插入到 FP-tree 后，FP 树也就建立完成. 建立完成的 FP-tree 如图 6 所示：

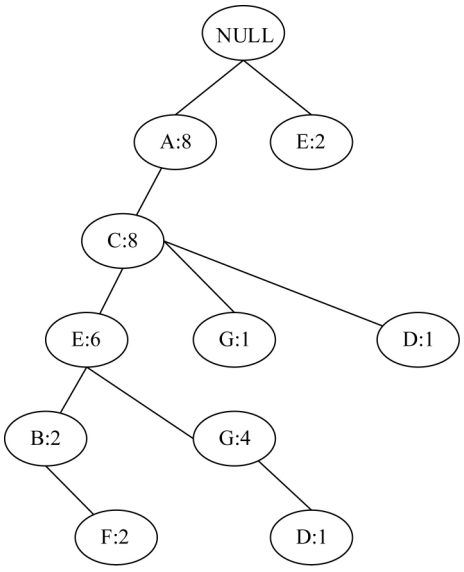

图 6　FP-tree

在建好 FP-tree 后，需要根据条件模式基查找频繁项集，所谓的条件模式基是以要挖掘的节点作为子节点所对应的 FP 子树，将 FP 子树中每个节点的计数设置为子节点的计数，删除计数低于支持度的节点. 通过图 7 所示的 F 对应的条件模式基的递归挖掘，便可以得到频繁项集.

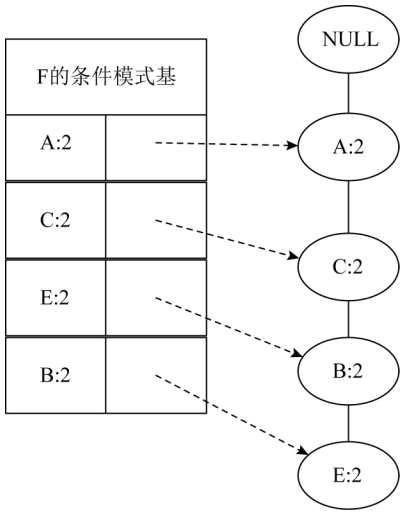

图 7　F 的条件模式基

该递归挖掘过程主要包括为以下步骤：1)挖掘频繁 2 项集，例如{A:2,F:2},{C:2,F:2};2)合并频繁 2 项集，得到频繁 3 项集，例如{A:2,C:2,F:2},{A:2,B:2,F:2};3) 重复步骤 2)，直到条件模式基为空，在本例中，最高可得频繁 5 项集{A:2,C:2,B:2,E:2,F:2}.

综上所述，通过 FP-growth 算法得到频繁项集具有以下步骤：

Step1. 第 1 次读取数据集，得到所有频繁 1 项集的计数. 删除支持度低于阈值的项，将 1 项频繁集放入项头表，并按照支持度降序排列.

Step2. 第 2 次读取数据集，将读到的原始数据剔除非频繁 1 项集，并按照支持度降序排列.

Step3. 将排序后的数据集每条插入 FP-tree 中，靠前为父节点，靠后为子节点. 同父节点，父节点加 1，新节点出现，项头表对应的节点会通过节点链表链接上新节点. 所有的数据都插入到 FP-tree 后，完成 FP-tree 建立.

Step4. 从项头表的底部项依次向上找到项头表项对应的条件模式基.

Step5. 从条件模式基递归获得符合要求的频繁项集.

基于 FP-tree 算法获得 XSS 攻击参数的频繁项集，挖掘其参数潜在关联，处理流程如图 8 所示.

利用 SVM 把超平面定义为 $\boldsymbol{W}\cdot\boldsymbol{X}+\boldsymbol{b}=\boldsymbol{0}$，$\boldsymbol{W}$ 是一个权重向量，$\boldsymbol{W}=(w_1,w_2,\cdots,w_n)$，其中 n 代

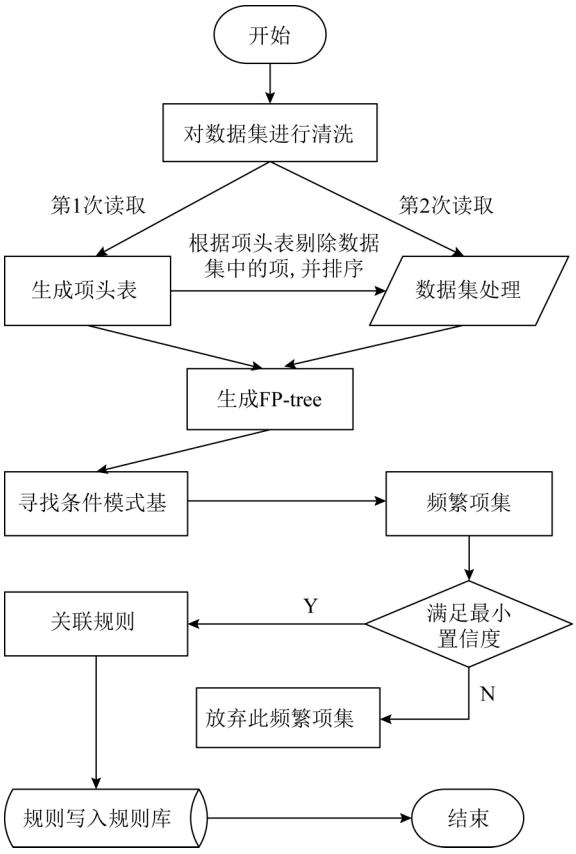

图8 FP-growth挖掘XSS攻击参数流程

表特征值的最大个数. SVM对XSS攻击与正常数据的分离准确度随着特征值个数的增加而提高，利用FP-growth算法计算出XSS参数中的支持度最高的参数：

$$Support(x_1,x_2,\cdots,x_n)=P(x_1x_2\cdots x_n)=\frac{number(x_1x_2\cdots x_n)}{num(ALLSamples)},$$

x表示样本中的所有参数，将支持度最高的参数应用到SVM中的特征值个数n中.

3.2 FP-growth算法优化SVM分类器的XSS攻击检测流程

利用FP-growth算法挖掘经数据清洗后的XSS参数间的潜在关联性，并将关联度较高的参数加入到SVM分类器的特征值数据中，增加SVM模型的特征值数量，从而提高检测的精准度，详细流程如图9所示.

图9中，SVM算法识别XSS攻击主要依赖于样本的特征化处理，此时通过FP-growth算法优化特征值参数，可以提高XSS攻击检测的精准度.

4 实验与结果分析

4.1 数据准备

基于AWVS(acunetix Web vulnerability scanner)漏洞扫描工具[10]扫描模拟靶场DVWA[11]中的XSS漏洞模块并进行日志搜集，搜集过程如图10所示，AWVS扫描被XSS攻击的网站，通过查看网站日志得到XSS样本并搜集整理，再从Web日志中搜集整理同样数量的正常访问日志.

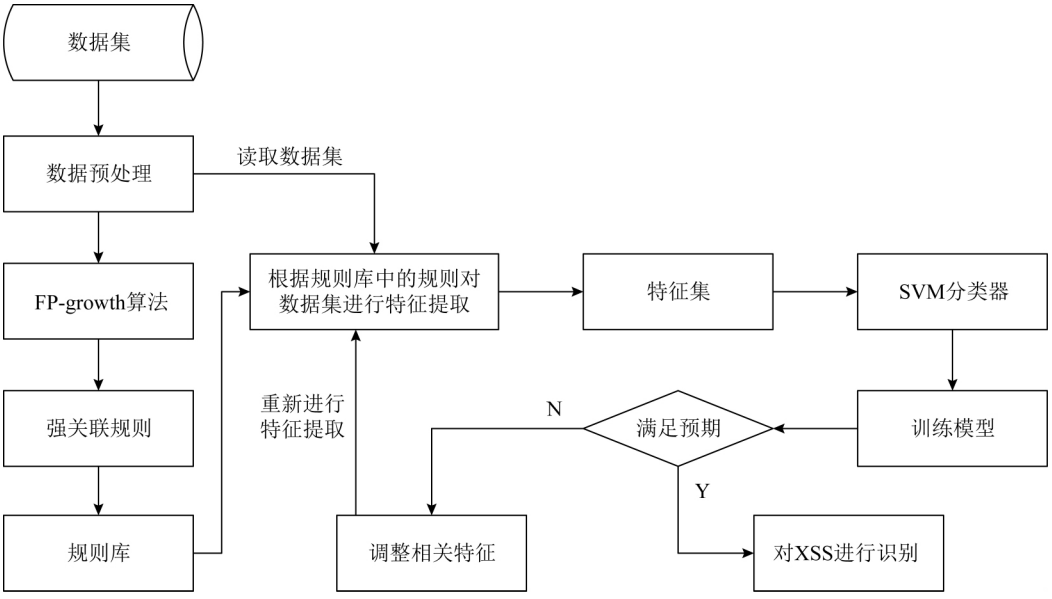

图9 FP-growth优化SVM分类器过程

```
"GET /dvwa/vulnerabilities/xss_r/\";NG328%0al<ScRiPt>Tyj3(9797)</ScRiPt>
HTTP/1.1" 403 289 "
ss_r/?name=llll" "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.21
(KHTML, like Gecko) Chrome/41.0.2228.0 Safari/537.21"
"GET /dvwa/vulnerabilities/xss_r/\"<WGUOUl>RX4AV%5b%21%2b
%21%5d</WGUOUl> HTTP/1.1" 403 281 "
ss_r/?name=llll" "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.21
(KHTML, like Gecko) Chrome/41.0.2228.0 Safari/537.21"
"GET /dvwa/vulnerabilities/xss_r/\"l<ScRiPt>prompt(957948)</ScRiPt>%3d%a2
HTTP/1.1" 403 288 "
ss_r/?name=llll" "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.21
(KHTML, like Gecko) Chrome/41.0.2228.0 Safari/537.21"
```

图 10　AWVS 扫描日志

4.2　数据清洗

由于日志无法被计算机直接识别，且根据观察，大部分的 XSS 样本中使用了 URL、十六进制、html 等编码去绕过检测，所以为了减少样本的复杂度，首先对日志进行解码，例如通过 python urllib 库 parse 类中的 unquote 方法进行 URL 解码，解码过程如图 11 所示：

```
原始日志: /64/portal.php?diy=yes%22%3E%3C/ScRiPt%3E%3CScRiPt%3Ealert
(/webscan/)%3C/ScRiPt%3E
解码: /64/portal.php?diy=yes"></ScRiPt><ScRiPt>alert(/webscan/)</ScRiPt>
```

图 11　日志进行解码

通过 python 中的 re. split 方法，将 URL 中例如“′”，“&”，“?”，“;”等常用的分隔符作为分割特征进行分割，分割结果如图 12 所示：

```
&searchTerm="><svg onload='a=alert;a
("XSSPOSED")'><"&defaultSearch=None&search=
['', 'searchTerm', '', '', '', 'svg onload', '', 'a', 'alert', 'a', '', 'XSSPOSED', '', '', '', '', '', '',
'defaultSearch', 'None', 'search', '', '']
```

图 12　日志文本化结果

4.3　FP-growth 算法挖掘 XSS 参数关联性

根据图 8 中 FP-growth 算法挖掘 XSS 参数流程，对经清洗后的数据进行循环计算，算出频繁项集，由于 FP-growth 算法发现关联规则的方法基于 apriori 算法，所以将最小置信度设置为 0.7[12]，参数同时出现概率大于 70%的情况将被认为具有关联性，再尝试从中找到 XSS 参数接近 100%的关联关系，部分结果如图 13 所示：

```
frozenset([f f, f/scriptf])——> frozenset ([f1f, f alertf, f scriptf]) conf:
0.741645244216
frozenset([f f, m: 3 Hmpl now:
|script *]) --> [frozenset ( [f V
frozenset ( [f1f, '/script', f alert 'alert', 'script']), frozenset ([
Hmpl:      [frozenset([f f,
frozenset([f/scriptf]) frozenset([f scriptf]) Hmpl after calculate: freqSet:
frozenset([f f frozenset([f/scriptf]) frozenset([f scriptf])
```

图 13　FP-growth 挖掘 XSS 参数关联性部分结果

经多次实验验证，结果基本一致，其中关联性最高的为(alert)|(script=)(%3c)|(%3e)|(%20)|(onerror)|(onload)|(eval)|(src=)|(prompt)参数，拟将其作为 SVM 的特征值提取，从而增加 SVM 特征值个数.

4.4　特征化

类似于 SVM 分类器的机器学习算法一般只能识别数字和向量，所以需把现实世界中的各类信息转换成为计算机可识别数字，该过程就是特征化，也称为向量化. 在本实验中，数据收集和数据清理是比较耗时的，而特征化则是最复杂也是最关键的.

由于各个向量之间的数据范围差别很大，一个数据过大的向量就可能影响到其他特征的结果，如果原始数据不服从高斯分布，在预测时表现可能不佳，这就需要把特征进行标准化. 一般方法为：1）标准化；2）均方差缩放；3）去均值. 本实验中，采用第 1 种方法，将日志中 URL 长度、URL 中包含第三方域名的个数、敏感字符的个数、敏感关键字的个数作为特征提取标准，使用 python 中 MinMaxScaler 或 MaxAbsScaler 函数完成实验数据集的标准化.

将 FP-growth 算法挖掘的 XSS 参数关联性部分结果加入 XSS 检测特征值参数的实现方法如下：len(re. findall(″(alert)|(script=)(%3c)|(%3e)|(%20)|(onerror)|(onload)|(eval)|(src=)|(prompt)″,url,re. IGNORECASE)).

增加特征后再次使用 SVM 算法对样本数据进行拆分、训练，生成训练模型，增加 SVM 对 XSS 攻击检测的准确度.

4.5　数据拆分

数据拆分需要将数据矩阵随机划分为训练子集和测试子集，并返回划分好的训练测试集样本和训练测试集标签，使用 python 中的函数 *cross_validation. train_test_split* 将数据划分为训练组和测试组. CrossValidator 函数首先将数据集拆分为 1 组折叠，这些折叠用作单独的训练和测试数据集. 当 $k=3$ 倍时，CrossValidator 将生成 3 个(训练，测试)数据集对，每个数据集对使用 2/3 的数据进行训练，1/3 进行测试. 通过在 3 个不同(训练，测试)数据集来计算 3 个模型的平均评估度量，所以一般使用数据的 60%作为训练组，40%作

为测试组，这个比例可以根据实际要求进行调节：

$x\_train, x\_test, y\_train, y\_test = cross\_validation.train\_test\_split(x, y, test\_size=0.4, random\_state=0)$.

使用函数 *train_test_split* 将矩阵进行随机划分，将全部训练集 S 分成 K 个不相交的子集，假设 S 中的训练样例个数为 m，那么每一个子集有 m/K 个训练样例，相应的子集称作 $\{s_1, s_2, \cdots, s_K\}$. 每次从分好的子集里，拿出 1 个作为测试集，其他 $K-1$ 个作为训练集，在 $K-1$ 个训练集上训练出学习器模型. 把这个模型放到测试集上，得到分类率，计算 K 次求得的分类率的平均值，作为该模型或者假设函数的真实分类率，这种方法充分利用了所有样本.

4.6 模型训练

由于 XSS 样本数据集是非线性的，需要对样本进行打标，XSS 攻击数据标记为 1，正常数据标记为 0，从而转化为线性数据. SVM 主要有线性核函数（linear）、径向基核函数（RBF kernel）、sigmoid 核函数和多项式核函数（polynomial kernel）4 种核函数. 由于本实验中样本特征维度为二维，所以采用性能良好的线性核函数 linear.

此处超平面被定义为 $\boldsymbol{W} \cdot \boldsymbol{X} + b = 0$，$\boldsymbol{W}$ 是一个权重向量，$\boldsymbol{W} = (w_1, w_2, \cdots, w_n)$，其中 n 是特征值的个数，$\boldsymbol{X}$ 是训练实例向量，假设二维特征向量：$\boldsymbol{X} = (x_1, x_2)$，$b$ 为额外的 w_0，此时超平面方程变为：$w_0 + w_1 x_2 + w_2 x_2 = 0$，调整权重向量，使超平面定义边际的 2 边：$y_i(w_0 + w_1 x_2 + w_2 x_2) \geqslant 1$，$\forall i$，都满足 $y_i(w_0 + w_1 x_2 + w_2 x_2) \geqslant 1$，经推导，MMH 可以表示“决定边界”$d(\boldsymbol{X}^{\mathrm{T}}) = \sum_{i=1}^{l} y_i \alpha_i \boldsymbol{X}_i \boldsymbol{X}^{\mathrm{T}} + b_0$，其中 $\boldsymbol{X}_i$ 是支持向量点及 MMH 边际的平面上的点. y_i 是支持向量点，$\boldsymbol{X}_i$ 的类别标记 $\boldsymbol{X}^{\mathrm{T}}$ 是要测试的实例，对于任何测试的实例将其作为 $\boldsymbol{X}^{\mathrm{T}}$ 代入以上公式，得出的符号的正负决定着它是在 MMH 的那一边，这样就可以完成 XSS 攻击数据和正常数据的识别模型训练.

4.7 模型验证

加载训练模型后，对测试集进行预测，并将预测结果进行对比，过程如图 14 所示.

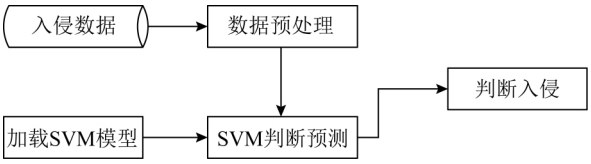

图 14 基于支持向量机入侵检测的模型

采用赵澄等人[6]提出的基于 SVM 分类器的 XSS 检测方法对包含 15 000 条 XSS 及正常样本数据集进行模型训练后，再对包含 1 000 条 XSS 及正常样本的数据集上进行模型验证，最后运行结果准确率为 99%以上；对样本容量分别为 1 000，2 000，4 000 条的 XSS 样本进行校验，准确率为 99%以上；运行结果分别如图 15、图 16 所示：

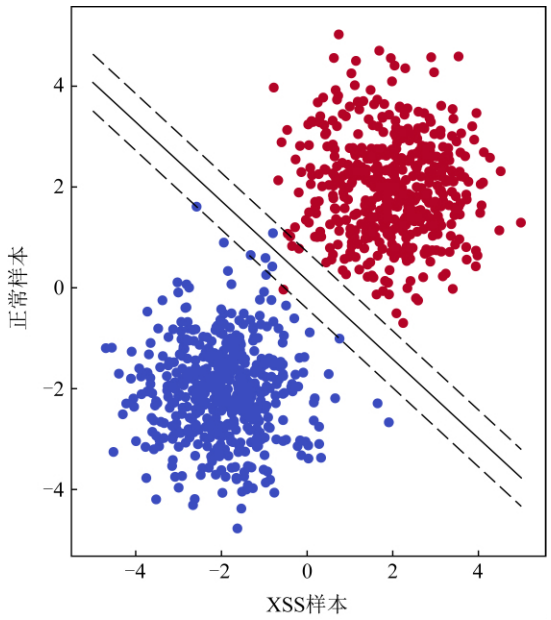

图 15 1 000 条包含 XSS 数据样本分类结果

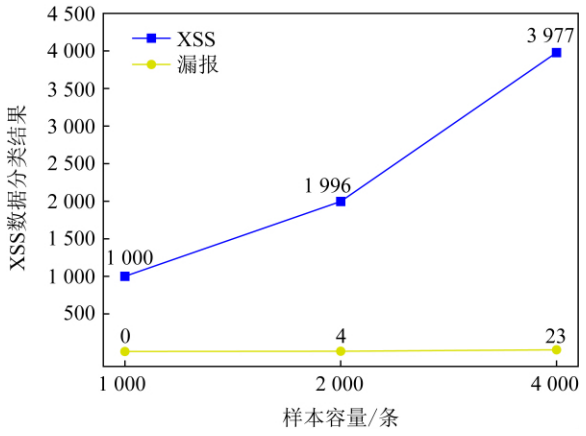

图 16 1 000，2 000，4 000 条 XSS 数据分类结果

基于本文提出的 FP-tree 优化 SVM 分类器的 XSS 检测方法对相同的 15 000 条 XSS 及正常

样本数据集进行模型训练后，再对样本容量分别为 1 000，2 000，4 000 的 XSS 样本进行校验，准确率为 100%，运行结果如图 17 所示，2 种方案的测试结果对比如表 1 所示.

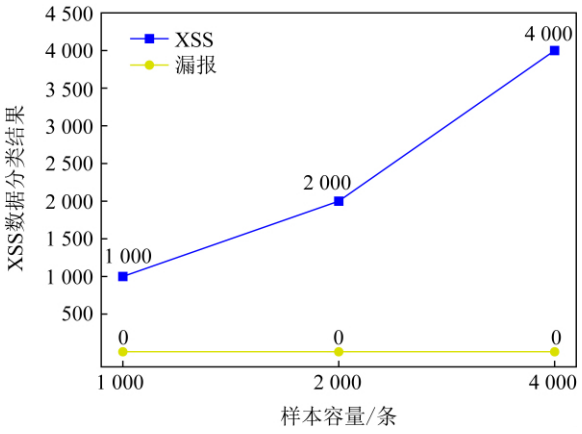

图 17　SVM 分类器优化后 XSS 数据分类结果

表 1　SVM 分类器优化前后测试结果　%

测试数据	SVM	SVM+FP-growth
15 000 条学习样本	√	√
1 000 条测试数据	100	100
2 000 条测试数据	100	100
4 000 条测试数据	99.67	100

实验证明，本文提出的 XSS 攻击检测方法可以有效提升 XSS 攻击检测的识别率，对于未知的 XSS 攻击的识别能力也有明显的优势.

4.8　XSS 攻击防范策略

根据本文所述的 XSS 攻击检测流程，训练 XSS 攻击检测模型，该模型可在 Web 应用防护系统(WAF)[14] 中部署，实现对 Web 访问流量的监控，进而识别出入侵数据，完成对 XSS 攻击的防范，具体流程如图 18 所示：

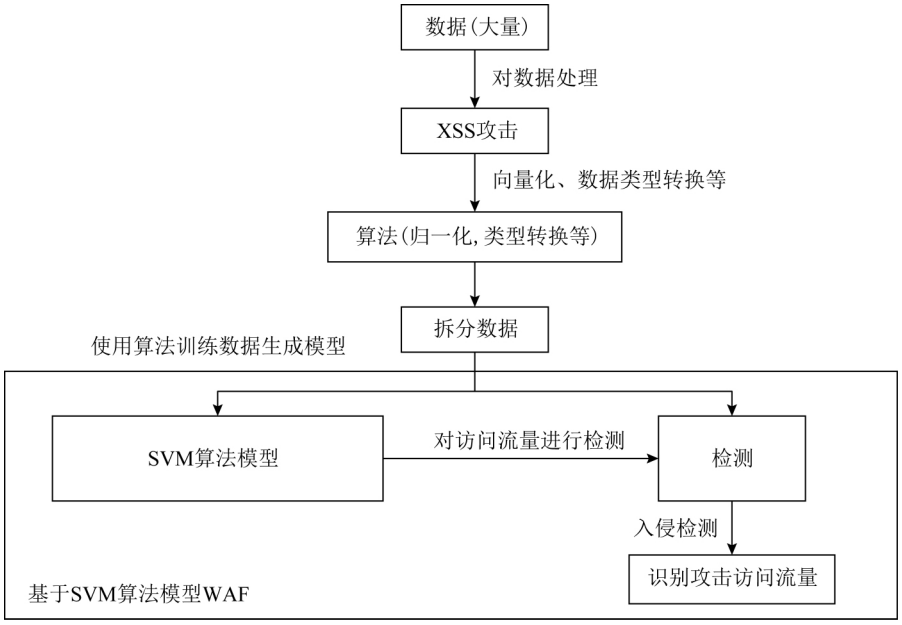

图 18　基于 SVM 算法模型 WAF 检测流程

5　结束语

本文在基于 SVM 分类器的 XSS 攻击检测技术原理的基础上，优化 SVM 分类器的检测方法，实现了 XSS 攻击的自动检测.

将 SVM 和 FP-growth 等机器学习算法应用于入侵检测是大势所趋[15]，将此类方法引入 WAF 进行入侵攻击检测，其最大优势就是可以不断学习新的攻击方式，来提高 XSS、SQL 注入、CSRF 等漏洞的识别率，实现 WAF 智能化学习及自防御，降低漏报率和人工干预，脱离繁琐的产品优化工作.

但在一些特定的入侵检测系统中，需要认真考虑特殊环境，如何构建健壮的网络入侵检测系统需要进一步研究.

参 考 文 献

[1] 360互联网安全中心. 2018上半年中国政企机构网络安全形势分析报告[EB/OL]. (2018-08-13)[2020-03-03]. http://zt.360.cn/1101061855.php?dtid=1101062514&did=491377741

[2] Deng Z, Choi K S, Jiang Y, et al. A survey on soft subspace clustering [J]. Information Sciences, 2016, 348(C): 84-106

[3] Rietz R, König H, Ullrich S, et al. Firewalls for the Web 2.0 [C] //Proc of IEEE Int Conf on Software Quality. Piscataway, NJ: IEEE, 2016

[4] Wang Wei, Liu Jiqiang, Pitsilis G, et al. Abstracting massive data for lightweight intrusion detection in computer networks [J]. Information Sciences, 2016, 433/434: 417-430

[5] 王丹, 顾明昌, 赵文兵. 跨站脚本漏洞渗透测试技术[J]. 哈尔滨工程大学学报, 2017, 38(11): 1769-1774

[6] 赵澄, 陈君新, 姚明海. 基于SVM分类器的XSS攻击检测技术[J]. 计算机科学, 2018, 45(11A): 356-360

[7] 高权, 步新玉. 基于增量式FP-Growth算法的关联规则挖掘模型设计[J]. 信息技术与信息化, 2020(3): 169-170

[8] Williams J, Manico J, Mattatall N. Cross-site Scripting (XSS)[S/OL]. [2020-03-03]. https://www.owasp.org/index.php/Cross-site Scripting (XSS)

[9] 雷亚国, 陈昊, 李乃鹏, 等. 自适应多核组合相关向量机预测方法及其在机械设备剩余寿命预测中的应用[J]. 机械工程学报, 2016, 52(1): 87-93

[10] 高晓峰. 漏洞你先知电脑更安全[J]. 计算机与网络, 2016, 42(13): 51-51

[11] 吴斌, 刘循. SQL注入攻击及漏洞检测防范技术[J]. 网络安全技术与应用, 2017(1): 76-78

[12] 钱雪忠, 孔芳. 关联规则挖掘中对Apriori算法的研究[J]. 计算机工程与应用, 2008, 44(17): 138-140

[13] 白洁, 田瑞丽, 张学军. Apriori算法在用户特性关联分析中的应用[J]. 计算机与网络, 2016, 42(12): 70-72

[14] 龚文涛, 郎颖莹. 基于主备模式的堡垒机网络架构[J]. 计算机系统应用, 2018, 27(3): 279-282

[15] Saxe J, Berlin K. Deep neural network based malware detection using two dimensional binary program features [C] //Proc of the 10th Int Conf on Malicious and Unwanted Software (MALWARE). Los Alamitos, CA: IEEE Computer Society, 2015: 11-20

李仁杰

本科生,主要研究方向为网络信息安全.

liushaze@qq.com

华 驰

博士研究生,教授,高级工程师,CCF会员,主要研究方向为计算机网络技术、数据挖掘及计算机教育.

huac@jsit.edu.cn

鲁志萍

副教授,主要研究方向为计算机计算机网络技术、网络信息安全.

luzp@jsit.edu.cn