文章编号：2095-2163(2020)03-0356-06 中图分类号：TP393.08 文献标志码：A

基于机器学习的 SQL 攻击检测技术研究

张泽亚，翟健宏
（哈尔滨工业大学 计算机科学与技术学院，哈尔滨 150001）

摘 要：SQL 注入是网络上使用非常广泛的攻击手段，也是防御难度极大的网络攻击方式。在信息安全领域中，SQL 注入因其适用范围广，操作门槛低，可造成的损失大而被视为对网络安全威胁极大的一类攻击方式。本论文的目的在于测试不同的机器学习算法对于 SQL 注入攻击的区分能力。研究搜集了大量的 SQL 注入攻击语句，选择 4 种不同的机器学习模型建立了分类器，并使用上面收集的数据对其进行了训练。最后，对 4 种算法所建立的分类器进行了测试，得出了最适合检测 SQLMAP 的机器学习算法是卷积神经网络(CNN)算法。

关键词：SQL 注入攻击；决策树；机器学习；分类器算法；SQLMAP

Research on SQL attack detection technology based on machine learning

ZHANG Zeya, ZHAI Jianhong
(School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China)

【Abstract】 In the field of information security, SQL injection is regarded as a kind of attack mode which threatens the network security greatly because of its wide application range, low operation threshold and great loss. The purpose of this paper is to test different machine learning algorithms for distinguishing SQL injection attacks. The paper collects a large number of SQL injection attack statements, selects four different machine learning models to build classifiers, and uses the data collected above to train them. Finally, the paper tests the classifiers built by the four algorithms and concludes that the most suitable machine learning algorithm for detecting SQLMAP is convolutional neural network (CNN) algorithm.

【Key words】 SQL injection attack; decision-making tree; machine learning; classifier algorithm; SQLMAP

0 引 言

SQL 注入是一种针对 Web 中数据库漏洞的注入技术。其工作原理是，把想要执行的命令添加到 Web 流量、域名或者查询的字符串中来达到恶意欺骗的目的。攻击者可以基于用户发送的信息而得到的回复和反应，把攻击性的代码发送到 Web 数据库服务器，其结果会导致系统崩溃、数据被破坏以及非公开的信息被偷窃。

在国际上享有高度权威性的非营利性组织开放式 Web 应用安全项目 Open Web Application Security Project(OWASP)发布的十大攻击行为中，SQL 注入因其危害高，攻击成本小，入门门槛低，适用范围广而多年来一直排名第一位。在可预见的数年内，SQL 注入上升趋势明显，将会是一段时间内互联网安全的重大威胁。

SQL 注入的破坏性极大。由于 Web 语言自身的缺陷，编程开发人员的疏忽和安全意识不足，大多数的 Web 应用系统都有被 SQL 注入攻击的可能性。而一旦攻击成功，那么攻击者就可以在被攻击的数据库中随意地修改、窃取、删除数据，甚至可以让系统陷于瘫痪。SQL 注入攻击可以对人们财产、公司信誉、国家安全造成不可挽回的损失。

综上所述，SQL 注入的防御技术是一个极具研究价值的课题方向。在目前情况下，仍有很多问题亟待解决：检测率不够高，误报率过大，检测效率不高，以及静态防御条件下，数据库更新慢，更新频率快等。

本文研究旨在建立一个 SQL 注入攻击的分类器。方法是使用机器学习的算法创建针对 SQL 注入攻击的分类器，再将收集的样本作为训练样本对分类器进行训练，最终得到检测 SQL 注入的分类器。之后，研究将通过实验验证检测比较不同机器学习算法在检测 SQL 注入时的有效性。

1 相关工作

1.1 数据集的收集

原始数据的收集，是研究的基础，也是重要的组成部分。本实验选择的数据是带有 SQL 注入攻击性的语句，但是这种数据在网上很少，根本无法满足

作者简介：张泽亚（1989-），男，硕士研究生，主要研究方向：网络内容安全、网络攻防实践；翟健宏（1968-），男，副教授，硕士生导师，主要研究方向：网络安全、云安全、工业信息安全等。

收稿日期：2019-06-10

实验的需要。而且，网上的数据还可能存在不全面，及未能包含所有种类的 SQL 注入攻击等缺点。为了使本文的数据集数量充足、种类全面、实用性强、具有说服力，收集数据的方法和使用如图 1 所示。由图 1 可见，对此可做阐释分述如下。

（1）利用 SQLMAP 扫描特定的网站，再利用 wireshark 进行捕获；研究选择的是 http://www.shiyanbar.com 实验吧网站。文中扫描使用过的命令详见表 1。

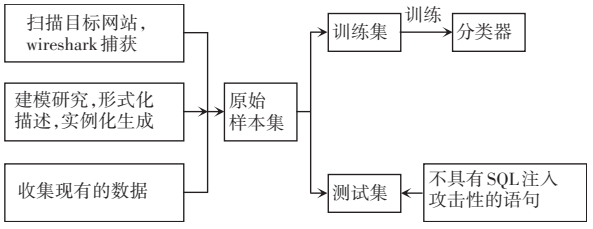

图 1　收集数据的方法和使用方法

Fig. 1　Methods of collecting and using data

表 1　使用的 SQLMAP 命令

Tab. 1　Using the SQLMAP command

输入的命令	执行的结果
-u “106.2.25.10？id=1”	测试是否存在安全漏洞
-u “106.2.25.10？id=1” --batch -password	窃取数据库密码
-u “106.2.25.10？id=1” --batch -dbms=mysql --level 3	指定测试安全级别和后台数据库的测试
-u “106.2.25.10？id=1” --password	获取用户密码哈希值

研究使用 wireshark 软件来捕获 SQLMAP 与网站之间的通讯流量，观察 SQLMAP 产生的部分注入请求及攻击类型见表 2。

表 2　部分 SQL 注入请求及攻击类型

Tab. 2　Partial SQL injection requests and attack types

注入攻击语句	攻击种类
select * from users where id=1 or " $#" or 1=1 -- 1	永真式
select * from users where id=1 or \.<1 union select 1,@@VERSION -- 1	联合式注入
106.2.25.10id=3 AND 9675=3571,SLEEP(2),3788	基于时间盲注

通过这种办法，一共收集了大约 7 000 条 SQL 注入攻击语句。

（2）对 SQL 注入的语句进行建模与分析，对攻击语句得出形式化描述，再通过实例化产生最终样本。本文使用的模型是 Wang 等人[1]提出的增广攻击树的 SQL 注入攻击模型的改进型如图 2 所示。

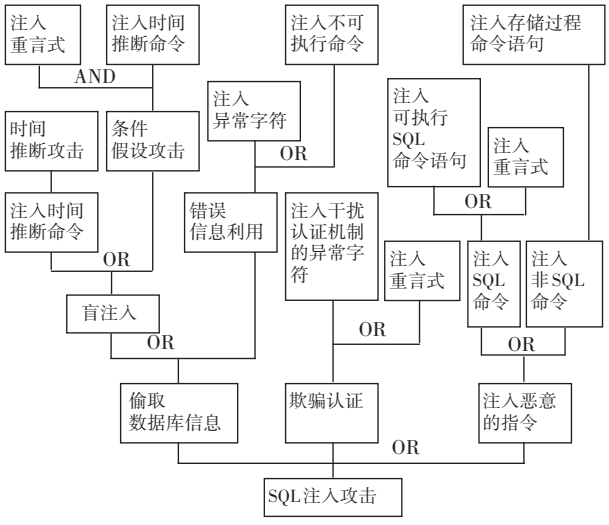

图 2　增广攻击树的 SQL 注入攻击模型

Fig. 2　SQL injection attack model of attack tree

基于上述模型，研究中可使用形式化语言，即以形式化语言描述 SQL 注入的攻击语言中出现何种反映特征时才可判定其具有 SQL 注入安全漏洞。实例化的数据将为 SQL 注入检测机器学习模型训练提供指导。SQL 注入的测试用例是由一些基本的攻击载荷以一定的规则组合而成的，因此，这里将给出攻击载荷的形式化描述。其中，$S(x)$ 表示某类攻击输入集合。特别地，当示例中有多个示例时，以符号“/”分割，内容详情见表 3。

定义了攻击载荷后，为了将攻击载荷以一定规则组合为攻击输入，还要定义攻击载荷之间的运算符，以描述实现某类 SQL 注入攻击时，攻击载荷之间的规律及关系。为此定义如下操作符：

①|| 为攻击载荷的或操作，$S(1) \,||\, S(2) = \{x \mid x \in S(1) \vee x \in S(2)\}$，表示 $S(1)$ 和 $S(2)$ 两种攻击载荷任取其一即可。

②&& 为攻击载荷的与操作，$S(1)\&\&S(2) = \{x,y \mid x \in S(1) \wedge y \in S(2)\}$，表示 $S(1)$ 和 $S(2)$ 两种攻击载荷需要同时使用。

③ * 为攻击载荷之间的复合运算，$S(1) * S(2) = \{x \mid x \in S(1) \wedge S(2)\}$，表示使用 $S(1)$ 对 $S(2)$ 的攻击载荷进行处理，生成新的或者复合的攻击载荷，运算顺序为从右往左，即 $S(1) * S(2) * S(3) = S(1) * (S(2) * S(3))$。

接下来，也需定义预算的优先级。在各类运算符中，括号的优先级是最高的，* 的运算级要高于 || 和 &&。

其中,复合运算符 * 的优先级高于 ‖ 和 &&,&& 运算符的优先级高于 ‖。现举一例,即如 $S(1) \| S(2) \&\& S(3) * S(4)$ = $S(1) \| (S(2) \&\& (S(3) * S(4)))$。

研究中以此为基础进行化简,可推演得到 SQL 注入语句的表达式参见表 4。

表 3 攻击载荷的形式化描述及实例

Tab. 3 Formal description and example of attack load

攻击载荷	定义	实例
S(DS)	异常字符集合	and1 = 1#; * ;@ ;
S(CON)	条件式重合	= ;< ;> ;between; ! ;
S(IE)	重言式重合	1 = 1;1<2;'a'<'b';
S(NE)	矛盾式重合	1 = 2;1>2;2<1;
S(SC)	选取可执行 SQL 语句	select * from…;union select from;
S(OC)	选取引起 DNS 请求的 SQL 命令	select load_file
S(OP)	选取引起 DNS 请求的 SQL 存储过程	select dbms_ldap...from dual;
S(TI)	时间延迟函数集合	Sleep;delay;
S(LG)	逻辑连接词集合	And;or;&&;
S(CN)	条件判断	If;ifnull;
S(DC)	异常命令集合	Select @ @ version/having 1 = 1;union select user( )
S(WAF)	对用例进行变形来绕过 WAF	select/%75%45%65%43%73;union/ * ! 4214;

表 4 实例化时使用的 5 个表达式

Tab. 4 Five expressions used in instantiation

SQL 注入方式	注入用例表达式
显错注入	$(S \| S(\text{WAF})) * (S(\text{DS}) \| S(\text{DC}))$
联合查询	$(S \| S(\text{WAF})) * S(\text{SC})$
基于时间的盲注	$(S(\text{SC}) * S(\text{TI}) \| S(\text{TI})) * (S(\text{CON}) \| S(\text{CN}))$
布尔类型的盲注	$(S \| S(\text{WAF})) * (S(\text{LG}) * S(\text{IE}) \&\& S(\text{LG}) * S(\text{NE}))$
执行 SQL 命令或者存储过程	$(S(\text{SC}) \| S(\text{LG}) * S(\text{IE})) * (S(\text{OC}) \| S(\text{OP}))$

此外,仍需考虑一个问题,即攻击的语句是无穷无尽的,每种攻击方式也有许多重复和相似的表达方式。例如,重言式攻击,只要条件中有永真的等式就可以了。但是“1 = 1”和“999 = 999”,虽是 2 个等式,但显而易见的是将这两者都列举出来是没有意义的,研究中只需选取其中之一就可以代表这一类。至此,研究得到的攻击载荷的判断依据见表 5。

综上论述可知,测试用例的生成由表达式和分类依据共同决定,表达式决定测试用例由哪些攻击载荷根据何种规则生成,分类依据决定实际使用的攻击载荷。例如,对表达式 $S(\text{LG}) * S(\text{IE}) \&\& S(\text{LG}) * S(\text{NE})$ 的实例化可具体表述为:S(LG)中任选一个关键字,如“and”,S(IE)中任选一类攻击载荷,如“1 = 1”,S(NE)中任选一类攻击载荷,如“2 <1”,则生成的攻击输入为“and 1 = 1 and 2< 1”,该攻击输入即可作为检测基于布尔盲注的测试用例。以此类推,不断循环 S(LG)、S(IE)和 S(NE)中的载荷并根据表达式生成测试用例,最终生成全部的用于检测基于布尔盲注的测试用例。

表 5 攻击载荷的判断依据

Tab. 5 Judgment basis of attack load

攻击载荷	判别依据
S(DS)	选取常用异常字符串
S(CON)	比较运算符和条件式类别
S(IE)	比较运算符和重言式类别
S(NE)	比较运算符和不等式类别
S(SC)	SQL 命令动词关键字
S(OC)	SQL 函数名
S(OP)	存储过程
S(TI)	时间函数
S(LG)	选取常用的逻辑连接词
S(CN)	条件函数名
S(DC)	选取常用异常命令
S(WAF)	根据编码规则

通过上述化简操作过程,研究中实例化生成了共 7 000 条 SQL 注入的命令语句。

(3)在网上收集了一些现有的 SQL 注入语句。主要是在 Github 上和国外漏洞提交平台 exploit-db 上收集 SQL 注入的攻击语句。但是数量有限,大约有 2 000 条。

1.2　分类器的建立

分类器是本系统的核心部分，并将直接决定最后分类器的效果。本文将分别尝试 LSTM、CNN、SVM 和 KNN 四种算法，比较其性能效果。文中将针对这 4 种模型，研究推得剖析概论如下。

(1)长短时记忆算法(LSTM)模型。这是一种特殊的神经网络模型，最早由 Hochreiter 和 Schmidhuber 提出的[2]，并经多次演变改进而得以完善。已在各研究领域得到了广泛使用。

LSTM 的特点是可以将前面的所有单个样本都作为一份“经验”，用于处理下一个样本。但又不会将其完全继承，而是有选择性地“遗忘”掉其中一部分。

(2)卷积神经网络(CNN)模型。这是一种多层的神经网络，由卷积层、池化层、全连接层、输出层四个部分组成。一般情况下，卷积神经网络可以被视作是一种使用重复神经元的许多相同拷贝的运算网络结构。其特点是允许网络拥有大量神经元并表达计算大型模型，同时保持实际参数的数量来描述神经元行为方式的值，而且只需要相当小的学习。这种具有相同神经元的多个拷贝技巧大致类似于数学和计算机科学中的函数的抽象。类似地，卷积神经网络学习过一次的神经元会在其它结构中多次重复地得到使用，这也使得模型的准确率和学习效率会更高。

(3) Support Vector Machine(SVM)模型，即支持向量机。这是一种有监督的模式分类方法。SVM 的研究论题可以用一个经典的二分类问题加以描述。一个二分类问题示意如图 3 所示。在图 3 中，两类的圆点显然是可以被一条直线分开的，是模式识别领域中的线性可分问题。但是也有许多条直线可以将 2 类数据分开。图 3(b)和图 3(c)分别给出了 2 种不同的分类方案，其中黑色实线为分界线，研究将其称为决策面。每个决策面对应了一个线性分类器。虽然 2 种分类方法的结果相同，但如果考虑潜在的其它数据，则两者性能却是有差别的。

(a) 经典的二分类问题　(b) 分类方案 1　(c) 分类方案 2

(a) Classical binary classification problem　(b) Classification scheme 1　(c) Classification scheme 2

图 3　一个二类分类问题中的最优决策面

Fig. 3　Optimal decision surface in a two - class classification problem

经由 SVM 算法的评判可知，图 3 中的分类器(b)在性能上要优于分类器(c)，判别依据是图 3(b)的分类间隔比图 3(c)要大。这里就要用到第一个 SVM 中的一个概念：分类间隔。在保证决策面方向不变、且分类正确的情况下移动决策面，图中 2 条虚线之间的实线与两条虚线的距离相等，在决策面不改变的条件下，这条实线就是求解该问题的最优决策面。2 条虚线到实线的距离就称为分类间隔。一般来说，分类间隔越大，得到的分类效果越好，故而最大间隔就是 SVM 的最优解。从图 3 中可以看到，有一些样本点是正好穿过虚线的，可以说，这些样本点决定了虚线、及实线的位置。这样的样本点就是支持向量。通过前文的例子可以看出，支持向量最后决定了最优决策面的位置。

简单地说，SVM 方法就是提升样本的维度，使得原先在低维线性不可分的数据通过升维的方法，在高维变得线性可分，其分类的最优解就是最优超平面。升维就是把样本投向高维的映射。如果在低维度时，由于分类样本过于复杂而无法分类，那么支持向量机的做法就是提高其维度，使不同的样本具有更多的特征提取因素，从而在高维空间进行分类。高维空间中的分类间隔不再是一条线，而是一个超平面。核函数可以提升样本的维度，但同时尽量维持不增加样本的计算难度。SVM 方法中核函数的加入还可以避免维数灾难。本文使用的是 sigmoid 核函数。

(4)邻近算法(KNN)。或者说 K 最近邻分类算法，这是一种经典的机器学习算法。所谓 K 最近邻，就是当要对某一样本个体进行归类时，研究判断其属于哪一类的依据是，选择 k 个与其最接近的若干个样本，再依据这些样本归属的类别，来判断此研究个体属于哪一类。

这里，举出一个 K 邻近的实例如图 4 所示。当要判断样本个体 X_u 属于哪一类时，判断的方法可表述为：先选择 5 个距离 X_u 最近的样本(这里的“5”就是研究中的 K 值)，然后观察到这 5 个样本中，4 个属于 w_1，1 个属于 w_3，因此判定 X_u 属于 w_1 类。再经分析可知在距离 X_u 最近的 5 个样本中，属于 w_1 的最多，故而判断 X_u 属于 w_1 的概率最大。

通过上面的例子可以看出，KNN 算法在判断某一未知分类的样本属于哪一类时，其方法就是选择距离此样本最近的 k 个已知类别的样本，这里的 k 由研究者本人决定。k 值不同，分类结果也有可能不同。统计这 k 个已知样本的分类结果，即可判断该

未知样本归属为 k 个样本中个数最多的那一类。

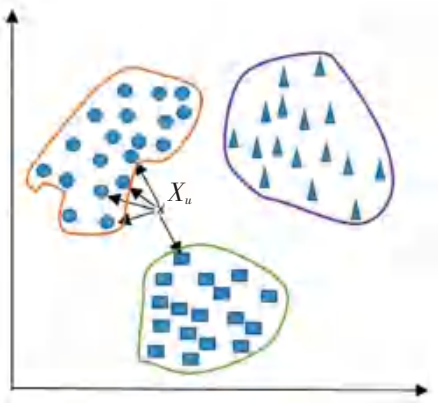

图 4 一个 K 邻近的例子

Fig. 4 An example of K-means

2 实验测试

2.1 评估的标准

研究中,要对测试样本进行标记。样本分为 2 类。一类标记为 0,即普通数据;另一类标记为 1,即具有 SQL 注入攻击性质的数据。如图 5 所示。

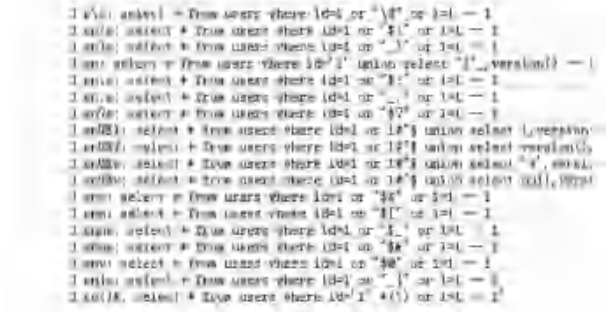

图 5 对数据添加标签

Fig. 5 Sign in data

将测试样本分别输入 3 个机器学习模型后,程序会将模型的标签重新写入。测试集分类结束之后,只需找出前后标签不一致的即可。针对分类效果优劣的评判标准可做定义表述如下。

(1)漏报率。数学定义可写为:

$$\text{漏报率}=\frac{\text{未检测出的 SQL 注入语句的数量}}{\text{测试集数据总数}}, \quad (1)$$

即开始时标签是 1,但分类结束后标签是 0 的数据占总数据量百分比。

(2)误报率。数学定义可写为:

$$\text{误报率}=\frac{\text{错将正常语句鉴定为 SQL 注入语句的数量}}{\text{测试集数据总数}}. \quad (2)$$

即开始时标签为 0,但结束后标签为 1 的数据占总数据量的百分比。

(3)准确率。定义为前后标签一致的数量,占总数据量的百分比。

2.2 测试环境

在本次实验中,计算机 CPU 为:Inter(R) Core(TM) i3-2100@ 3.10 GHz,系统为:Window7 旗舰版。软件运行环境为:pycharm2017, python3.6.4, ananconda2, tensorflow1.8.0。

2.3 分类器训练

在 15 000 条数据中,选择了 10 000 条用来作为训练集,分别对 4 个模型进行训练,并记录训练时间。4 种模型的训练时间如图 6 所示。需要强调的是,这 10 000 条数据是从上述 3 种 SQL 注入语句来源中大致平均抽取的,如此一来则使得训练的样本的选取数据范围更广,训练出的分类器性能也更好,更加具有说服力。

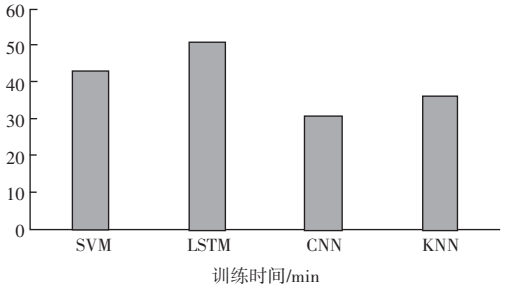

图 6 4 种模型的训练时间

Fig. 6 The time of training four models

由图 6 可以发现,使用相同的 10 000 条数据对 4 个分类器进行训练,训练时间最短的是 CNN。这说明,在同样的条件下,如果使用 CNN 作为分类器的算法,则分类的效率是最高的。

2.4 模型的测试

在模型训练结束后,接着对其分类效果进行了测试。测试方法是使用 wireshark 捕获 5 000 条正常的数据,与原始样本中剩下的 SQL 注入攻击语句样本混合在一起,使 3 个分类器分别进行分类,观察其分类效果。研究后得到的 4 种模型的测试结果如图 7 所示。4 种模型的错报率和误报率的结果数值见表 6。4 种模型的准确率如图 8 所示。

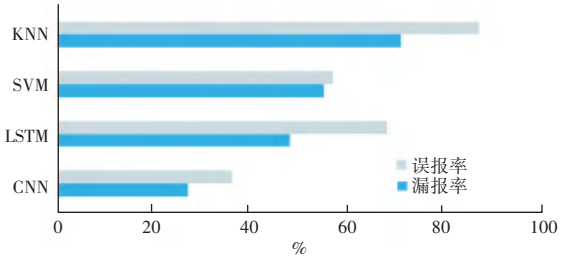

图 7 4 种模型的漏报率与误报率

Fig. 7 The false negative rate and false positive rate of models

表 6　4 种模型的错报率与误报率

Tab. 6　The false negative rate and false positive rate of models

	漏报数量	漏报率/%	误报数量	误报率/%
CNN	27	27	36	36
LSTM	48	48	68	68
SVM	55	55	57	57
KNN	71	71	87	87

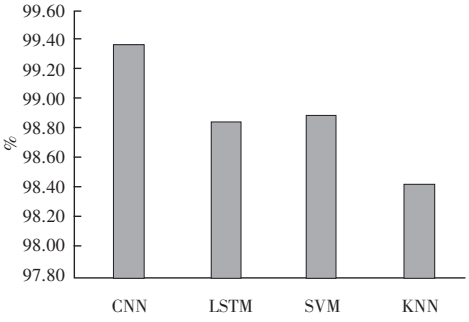

图 8　4 种模型的准确率

Fig. 8　Accuracy of four kinds of models

由图 7、图 8、表 6 分析可知，无论是从漏报率、误报率，还是从准确率上看 CNN 模型是针对 SQL 注入检测的最好模型，；SVM 的分类效果次之，LSTM 相比于传统的机器学习算法（SVM 与 KNN）并没有明显的优势，KNN 的分类效果较差。

研究中，还将各个模型分类错误的 SQL 语句（包括漏报和错报的）进行了梳理和分类，分类结果如图 9 所示。

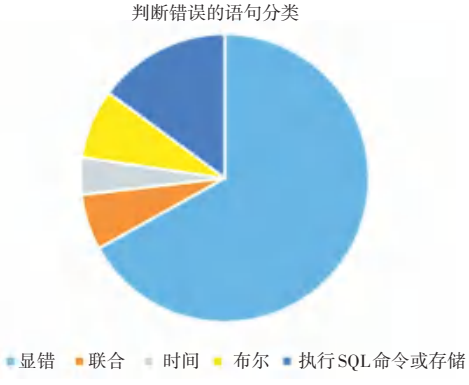

图 9　半段错误的 SQL 注入语句分类结果

Fig. 9　Classification results of erroneous SQL injection statements

由图 9 可以看出，4 种模型在联合式注入、基于时间盲注和基于布尔的盲注入检测时效果非常好，对于显错式注入的判断效果是最差的。

3　结束语

研究中，设计提出了一个模型来实例化生成 SQL 注入攻击语句，在收集到足够多的语句后，研究比较了 4 种不同机器学习算法在 SQL 注入攻击的检测上的性能。通过实验发现，CNN 算法的训练时间最短，分类的效果最好。而 LSTM 在 SQL 注入分类的效果上，却并未显现出明显优势。

参考文献

[1] WANG Jie, PHAN R C W, WHITLEY J N, et al. Augmented attack tree modeling of SQL injection attacks[C]//2010 2nd IEEE International Conference on Information Management and Engineering. Chengdu, China: IEEE, 2010, 437: 1009.

[2] HOCHREITER S, SCHMIDHUBER J. Long short-term memory [J]. Neural Computation, 1997, 9(8): 1735.

[3] MCWHIRTER P R, KIFAYAT K, SHI Qi, et al. SQL injection attack classification through the feature extraction of SQL query strings using a Gap-Weighted String. Subsequence Kernel [J]. Journal of Information Security and Applications, 2018, 40: 199.

[4] 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.

[5] 张燕. 数据挖掘提取查询树特征的 SQL 注入攻击检测[J]. 电子技术应用，2016, 42(3)：90.

[6] 王苗苗，钱步仁，许莹莹，等. 基于通用规则的 SQL 注入攻击检测与防御系统的研究[J]. 电子设计工程，2017, 25(5)：24.

[7] 韩宸望，林晖，饶绪黎，等. 基于代理模式的 SQL 注入过滤方法[J]. 计算机系统应用，2018, 27(1)：98.

[8] 张慧琳，丁羽，张利华，等. 基于敏感字符的 SQL 注入攻击防御方法[J]. 计算机研究与发展，2016, 53(10)：2262.

[9] 韩涛. 基于解析树的 SQL 注入检测方法研究[J]. 哈尔滨：哈尔滨工业大学，2013.

[10] LIU Pengfei, QIU Xipeng, CHEN Xinchi, et al. Multi-timescale long short-term memory neural network for modelling sentences and documents[C]// Conference on Empirical Methods in Natural Language Processing. Lisbon, Portugal: IEEE, 2015: 2326.

（上接第 355 页）

[3] 李保强，李忠华，白培康，等. 选区激光熔化 AlSi10Mg 应力场数值模拟研究[J]. 应用激光，2019(2)：211.

[4] 万华亮，王奇志. 增材制造铝镁合金 AlSi10Mg 的疲劳性能研究[J]. 强度与环境，2019(3)：20.

[5] 刘明辉，肖伟，王建伟，等. 铝合金中刃型位错与合金元素相互作用的分子动力学模拟研究[J]. 稀有金属，2017, 41(3)：233.

[6] 张宁，杨新华，陈传尧. 含球形孔洞双晶铜单向拉伸性能的分子动力学模拟[J]. 计算力学学报，2010(2)：330.

[7] 梁华，李茂生. 孔洞和空位对单晶铝力学性能影响的分子动力学研究[J]. 计算物理，2019, 36(2)：211.

[8] 周继凯，朱清华. Fe-C 合金动态拉伸力学性能温度和应变率效应分子动力学[J]. 科学技术与工程，2019, 19(11)：61.

[9] JELINEK B, GROH S, HORSTEMEYER M F, et al. MEAM potentials for the Al, Si, Mg, Cu, and Fe alloys[J]. Physical Review B, 2012, 85(24): 245102.

[10] LARSEN P M, SCHMIDT S, SCHIφTZ J. Robust structural identification via polyhedral template matching[J]. Modelling and Simulation in Materials Science and Engineering, 2016, 24(5): 055007.