计算机科学
COMPUTER SCIENCE

未知网络攻击识别关键技术研究

曹扬晨，朱国胜，孙文和，吴善超

引用本文

曹扬晨，朱国胜，孙文和，吴善超. 未知网络攻击识别关键技术研究[J]. 计算机科学, 2022, 49(6A): 581-587.

CAO Yang-chen, ZHU Guo-sheng, SUN Wen-he, WU Shan-chao. Study on Key Technologies of Unknown Network Attack Identification[J]. Computer Science, 2022, 49(6A): 581-587.

相似文章推荐（请使用火狐或 IE 浏览器查看文章）

Similar articles recommended (Please use Firefox or IE to view the article)

基于 SMOTE-SDSAE-SVM 的车载 CAN 总线入侵检测算法

SMOTE-SDSAE-SVM Based Vehicle CAN Bus Intrusion Detection Algorithm

计算机科学, 2022, 49(6A): 562-570. https://doi.org/10.11896/jsjkx.210700106

一种基于顺序和频率模式的系统调用轨迹异常检测框架

Anomaly Detection Framework of System Call Trace Based on Sequence and Frequency Patterns

计算机科学, 2022, 49(6): 350-355. https://doi.org/10.11896/jsjkx.210500031

基于节点相似性和网络嵌入的复杂网络社区发现算法

Complex Network Community Detection Algorithm Based on Node Similarity and Network Embedding

计算机科学, 2022, 49(3): 121-128. https://doi.org/10.11896/jsjkx.210200009

基于差分隐私的 *K*-means 算法优化研究综述

Review of *K*-means Algorithm Optimization Based on Differential Privacy

计算机科学, 2022, 49(2): 162-173. https://doi.org/10.11896/jsjkx.201200008

基于降噪自编码器和三支决策的入侵检测方法

Intrusion Detection Method Based on Denoising Autoencoder and Three-way Decisions

计算机科学, 2021, 48(9): 345-351. https://doi.org/10.11896/jsjkx.200500059

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.210400044

未知网络攻击识别关键技术研究

曹扬晨　朱国胜　孙文和　吴善超
湖北大学计算机与信息工程学院　武汉 430062
(943407866@qq.com)

摘　要　入侵检测是一种主动防御网络中攻击行为的技术,在网络管理方面起着至关重要的作用,而传统的入侵检测技术无法识别未知攻击,也是长期困扰本领域的难题。针对未知类型的入侵攻击,提出了 *K*-Means 与 FP-Growth 算法相结合的未知攻击识别模型,以实现对未知攻击的规则进行提取。首先,对于多种未知攻击混合的数据,根据样本间的相似性用 *K*-Means 进行聚类分析,引入轮廓系数评估聚类的效果,聚类完成之后,同种未知攻击被分到相同的簇中,人工提取未知攻击的特征,对特征数据进行预处理,将连续型特征离散化,然后用 FP-Growth 算法挖掘未知攻击数据的频繁项集和关联规则,最后对其进行分析,得出该未知攻击的规则,用规则对该类型的未知攻击进行检测,结果表明,所提模型的准确率可达 98.74%,优于其他相关模型。

关键词:入侵检测;未知攻击;*K*-Means;FP-Growth;关联规则
中图法分类号　TP181

Study on Key Technologies of Unknown Network Attack Identification

CAO Yang-chen,ZHU Guo-sheng,SUN Wen-he and WU Shan-chao
School of Computer and Information Engineering,Hubei University,Wuhan 430062,China

Abstract　Intrusion detection is a technology that proactively defends against attacks in the network and plays a vital role in network management. Traditional intrusion detection technology cannot identify unknown attacks,which is also a problem that has plagued this field for a long time. Aiming at unknown types of intrusion attacks,an unknown attack recognition model combining *K*-Means and FP-Growth algorithms is proposed to extract the rules of unknown attacks. First,for the data of a mixture of multiple unknown attacks,cluster analysis is performed with *K*-Means based on the similarity between samples,and the silhouette coefficient is introduced to evaluate the effect of clustering. After the clustering is completed,the same unknown attacks are classified into the same cluster,the feature of unknown attack is manually extracted,the feature data is preprocessed,the continuous feature is discretized,and then the frequent item sets and association rules of the unknown attack data are mined by the FP-Growth algorithm,and finally the rule unknown attack is obtained by analyzing it. The rules of attack are used to detect this type of unknown attack. The results show that the accuracy rate can reach 98.74%,which is higher than that of the related algorithms.

Keywords　Intrusion detection,Unknown attack,*K*-Means,FP-Growth,Association rules

1　引言

随着信息革命的不断推进,互联网以惊人的速度渗透到科技、文化、经济等领域。网络技术的高速发展一方面推动了社会经济的迅速发展,另一方面也给人类社会带来了前所未有的挑战。2020 年 4 月,欧洲能源巨头 EDP(Energias de Portugal)遭到勒索病毒的攻击,损失约 1090 万美元;2020 年 5 月,瑞士铁路机车制造商 Stadler 遭到网络攻击,被恶意软件入侵,攻击者窃取了很多敏感数据并造成了一定的威胁。2020 年 7 月,《CNCERT 互联网安全威胁报告》指出,在中国,超过 132 万个终端被病毒入侵,包含政府在内的被病毒篡改的网站有 18799 个,被植入后门的网站有 5048 个,包含政府的 9 个,根据国家信息安全漏洞共享平台收集整理的数据显示存在安全漏洞 1476 个,其中高危漏洞有 447 个。由此可见,随着互联网的发展,网络中的攻击行为也在不断发生,不仅造成了经济方面的巨大损失,甚至对国家的安全和社会的稳定发展造成了威胁。因此,如何发现黑客的攻击行为,尽可能识别、抵御网络上的恶意攻击,维护网络安全成为亟需解决的问题。

研究发现,网络中超过 40%的流量采用未知协议,并不公开协议的规范。考虑到通信的效率和安全性,未知协议被广泛运用到私密性较高的商业或者军队等领域,与此同时,也有一些不法分子利用未知协议发起网络攻击,不仅给网络管理带来了挑战,也使网络安全存在一定的隐患。为了维护

基金项目:赛尔网络下一代互联网技术创新项目;基于网络流量重构的校园区域舆情挖掘与监测系统(NGII20170210)
This work was supported by the CERNET Innovation Project and Campus Regional Public Opinion Mining and Monitoring System Based on Network Traffic Reconstruction(NGII20170210).
通信作者:朱国胜(zhuguosheng@hubu.edu.cn)

网络空间安全，识别未知的攻击，从中提取特征，分析出该攻击的规则，是非常值得研究且有意义的事情。

由于不知道未知攻击的特征规则，而人工提取特征的过程十分耗时，且十分依赖专家的水平，提出的特征质量也无法保证。面对海量的应用，基于数据挖掘的特征提取方法极大地提高了特征提取的效率。本文用聚类算法 *K*-Means 先对未知攻击进行聚类分析，使得未知的数据根据其相似性形成不同的簇，然后利用数据挖掘技术对同一簇未知攻击的数据进行频繁项集和关联规则的提取，从而提取出攻击的特征和规则，通过实验验证了方法的有效性。

本文第 2 节介绍未知入侵的现状，对已有的未知攻击检测技术进行描述；第 3 节介绍了本文提出的基于 *K*-Means 和 FP-Growth 的未知攻击识别方法，包括算法原理和模型建立的步骤；第 4 节为实验验证，即通过经典的入侵检测数据集 NSL-KDD 对检测方法进行校验与评估。

2 未知攻击检测技术研究现状

面对日益复杂的网络环境，入侵检测广泛使用机器学习的方法，通过对网络流量数据的训练来获得新的信息，更具智能性，提高了检测率，降低了漏报率和误报率，使相关的研究也层出不穷。

文献[1]提出一种基于改进 *K*-Means 聚类的入侵检测方法，该方法通过对未标记的数据进行训练，来检测新的攻击，在 KDD99 数据集上进行实验，实验结果表明，该方法提高了检测率，降低了误报率，使模型具有检测未知入侵的能力。文献[2]提出一种基于卷积神经网络的攻击检测方法，该方法通过从序列片段中学习攻击模式，从而对具有相同攻击频率的未知攻击进行检测，实验结果表明其准确率达到了 96.76%。文献[3]提出了零次学习模型，该模型描述所有攻击的语义信息，构建已知和未知攻击之间的关联关系，然后分类器对未知攻击进行分类，把基于稀疏自编码器的零次学习方法运用于未知攻击检测，使用 NSL-KDD 数据集进行检验，实验结果表明该方法的平均精度达到 88.3%。文献[4]提出一种基于遗传算法和聚类分析的入侵检测方法，该方法对能无标签的攻击数据进行有效检测，并在 CSIC 2010 数据集上进行了实验，结果表明其检测率可达 99.25%。文献[5]针对 Snort 仅能识别已知攻击且识别率低的问题，提出一种基于 *K* 近邻算法优化的 *K*-Means 算法和增加信任度指标的 Apriori 算法相结合的方式，将模型运用于 Snort 中，增加了 Snort 发现未知攻击的能力，并提高了检测的准确率。文献[6]针对传统的误用检测技术无法识别新的未知攻击，以及异常检测误报率高的问题，将改进后的 *K*-Means 运用到异常检测中，对网络中的异常先进行过滤，减少误用检测的数据量，再将改进的 Apriori 算法运用到误用检测中，提取新的攻击规则，更新规则库，并采用 KDD cup99 数据集验证了模型的合理性。文献[7]采用人工神经网络的方法检测已知和未知的 DDoS 攻击，使用 ANN 算法学习并检测出与训练数据相似的未知攻击，将最新的 DDoS 攻击生成攻击模式用于检测攻击行为，实验结果表明，训练的攻击模式越多，对已知和未知攻击的检测精度越高，检测精度最高可达 98%。文献[8]介绍了一种无监督的网络入侵检测系统，提出基于子空间聚类和 EAC(Evidence Accumulation)算法的无监督异常值检测方法，用 KDD99 数据集和两个网络真实流量数据集评估其检测未知攻击的能力，结果显示该系统可以检测超过 90%的攻击。文献[9]提出一种自适应威胁检测架构，用于解决模型不能及时识别未知攻击的问题，使用蜜罐收集未知攻击的数据，将攻击数据实时添加到模型训练中，最后使用真实的数据集进行评估，分类准确率超过 90%，但此方法依赖蜜罐中提取的信息。

综上所述，机器学习在入侵检测中的应用十分广泛，对于未知的攻击，由于没有标签，可通过无监督学习的算法进行聚类分析以及利用数据挖掘的方法发现未知攻击的特征规则，或者通过深度学习的方式对其进行研究。

本文将 *K*-Means 与 FP-Growth 结合，用 *K*-Means 将多种未知攻击进行聚类，再对同一种未知攻击采用 FP-Growth 算法进行特征规则挖掘，通过实验和人工分析实现了对未知攻击的特征和规则的提取，不仅提高了未知攻击识别的准确率，也提高了时间效率。

3 基于 *K*-Means 和 FP-Growth 的未知攻击识别模型

本节首先对聚类算法 *K*-Means、Apriori 算法以及 FP-Growth 算法进行简单的介绍，然后详细描述未知攻击识别模型的构建过程。

3.1 聚类算法 *K*-Means

K-Means 是一种经典的聚类算法，在数据集没有标签的情况下，通常可以通过数据间的相似性将数据进行聚类分析。*K*-Means 算法将 N 个样本划分到不同的 k 个簇中，每个簇的质心为该簇中所有样本数据的均值。*K*-Means 的主要思想是，首先确认 k 值，然后生成 k 个质心，将样本划分到距离最近的质心所在的簇中。簇的数量 k 一般通过枚举的方式，以轮廓系数来确认。算法的详细步骤：

(1)首先随机生成 k 个质心 $a=a_1,a_2,\cdots,a_k$。

(2)计算每个样本 x_i 到所有质心的距离，该样本被划分到距离最短的簇中，聚类的簇为 $c=c_1,c_2,\cdots,c_k$。

(3)根据被分到同一个簇中的所有样本点，重新计算每个簇的簇心 a_j，按如下式进行计算：

$$a_j=\frac{1}{|c_i|}\sum_{x\in c_i}x \tag{1}$$

(4)重复步骤 2 和步骤 3，当质心的位置不再变化时说明聚类完成。

同一个簇中的样本相似性较大，全部样本到质心的距离之和越小，说明簇内的差异越小，假设 x 为簇中的一个样本，μ 为簇中的质心，n 为样本的特征个数，i 为样本的每个特征，通过欧几里得距离对簇中所有样本点到质心的距离进行计算，计算式如下：

$$d(x,\mu)=\sqrt{\sum_{i=1}^{n}(x_i-\mu_i)^2} \tag{2}$$

选取欧几里得距离计算，则同一个簇中所有样本到质心的距离平方和公式为：

$$ClusterSumofSquare(CSS)=\sum_{j=0}^{m}\sum_{i=1}^{n}(x_i-\mu_i)^2 \tag{3}$$

$$TotalClusterSumofSquare=\sum_{l=1}^{k}CSS_l \tag{4}$$

其中，m 为一个簇中样本点的个数，k 为簇的个数，由式(3)计算簇类平方和，又叫 inertia，由式(4)计算所有簇的簇内平方和之和，得到的是整体平方和，整体平方和越小，说明簇内数据相似度越高，簇内平方和可作为模型评估的指标，当簇内

平方和不再变化时，质心的位置便确定了。

在分类算法中，分类的结果有正确和错误之分，可以用模型分类的准确率来评估，聚类算法没有绝对的正误之分，随着簇的个数越多，簇内平方和越小，当簇的个数与样本个数相等时，簇内平方和为 0。对于聚类算法而言，簇内的稠密程度与簇外的离散程度可以看出聚类效果，聚类算法中引入轮廓系数作为模型的评估指标。每个样本都有自己的轮廓系数，假设一个样本与同一个簇中其他样本的平均距离为 a，与离的最近的不同簇中样本之间的平均距离为 b，那么单个样本的轮廓系数公式为：

$$s=\frac{b-a}{\max(a,b)} \tag{5}$$

由式(5)可知，当 a 大于 b 时，说明该样本与其他簇的距离比自己所在簇距离更远，聚类效果不佳；而当 a 小于 b 时，轮廓系数越大，说明聚类效果越好；当 a 等于 b 时，说明两个簇的相似度一致，轮廓系数的取值在(−1,1)之间。轮廓系数大的样本越多，说明整体的聚类越好，轮廓系数小的样本越多，说明簇的个数不适合。

3.2 关联规则

关联规则指两个或者多个事务存在关联，形如 $X->Y$ 的蕴含式，X 为先导规则，而 Y 为后继规则，其中，X 发生之后，可以对 Y 进行预测。数据挖掘技术可以对一些未知数据之间的关系进行关联规则的挖掘，下面对一些理论基础的概念进行描述。

为了更好地表示关联规则，以下进行形式化描述，假设 $I=\{I_1,I_2,\cdots,I_m\}$ 是 m 个项的集合，事务 T 表示 I 的一个非空子集，$D=\{T_1,T_2,\cdots,T_n\}$，若一个项或者项集 X 在事务集合 D 中出现的次数记为 $\#X$，则支持度可表示如下：

$$Support(X)=\frac{\#X}{n} \tag{6}$$

将支持度大于或等于给定阈值的项集称为频繁项集。关联规则 $X->Y$ 的支持度，即为 X 和 Y 同时出现的次数 $\#(X\cup Y)$ 与事务总数 n 的比值：

$$Support(X->Y)=\frac{\#(X\cup Y)}{n} \tag{7}$$

置信度为 X 和 Y 同时出现的次数与 X 在事务集合 Y 中出现次数的比值，表示为：

$$Confidence(X->Y)=\frac{\#(X\cup Y)}{\#(X)} \tag{8}$$

关联规则中的支持度和置信度是两个关键的指标，支持度表示项集在总的数据中所占的比例，置信度表示关联规则的可信程度，在一般情况下，这两个指标都需达到一定的值才能表示可靠的关联规则。

3.3 Apriori 算法

Apriori 算法是数据挖掘十大经典算法之一，其因算法简单、易实现而被广泛运用于各个领域，在未知攻击方面更是能发挥其优势。Apriori 算法通过挖掘大量未知攻击数据之间的关联，通过分析行为规则，更新攻击的规则库，提高检测网络攻击的能力。Apriori 算法利用第 k 次循环得出的长度为 k 的频繁项集，构造长度为 $k+1$ 的候选项集，降低了算法时间复杂度，算法遵守向下闭包的规则，即任何频繁项集的子集均为频繁项集，不频繁项集的任何超集也是不频繁的。算法的步骤为：

(1)扫描数据库中所有的数据，可以得到初始的项集，此时项集的大小 $k=1$。

(2)计算每个 k 项集支持度，筛选出支持度大于给定阈值的频繁 k 项集；若得到的频繁 k 项集大小为空，则直接返回频繁 $k-1$ 项集的集合；若得到的频繁 k 项集数量为 1，则返回该频繁 k 项集。

(3)令 $k=k+1$，重复步骤(2)，可得到所有的频繁项集。

(4)计算每个得到的频繁项集的置信度，即可得到关联规则。

Apriori 算法的缺点是，其需要在每一轮迭代时遍历数据库，当数据量较多时，频繁遍历数据库会导致算法效率降低，运行速度变慢。

3.4 FP-Growth 算法

FP-Growth 算法是基于 Apriori 算法的改进，FP-Growth 采用树的结构存储数据，只需要进行两次遍历，即可提取频繁项集，大大加快了算法的速度，比 Apriori 效率更高。FP-Growth 的主要思想是将各条记录之间频繁项的关系压缩在频繁模式树中，然后将频繁模式树根据条件模式拆分成一组条件模式树，再挖掘其频繁项集，主要分为两个步骤：

(1)构建 FP 树：构建 FP 树的过程中，首先统计数据集中的每个项的频数，将小于最小支持度的项删除，剩下的元素被称为频繁项，然后用剩下元素的每条记录来构造 FP 树，同时更新头指针表，头指针表中的内容为频繁项以及频繁项的频数，头指针表指向记录中的相同元素，向 FP 树中插入记录，当插入的记录与 FP 树中的路径相同时，则更新项的频数，不相同时则在不同的地方分叉，并创建新的结点，然后从 1 开始重新记录项的频数。

(2)从 FP 树中挖掘出频繁项集：FP 树构建完成之后，从 FP 树中挖掘出每一个频繁项的频繁项集，首先，获得频繁项的前缀路径，以此路径构建新的条件 FP 树，然后获得条件 FP 树中的频繁项的前缀路径，依次迭代，直到条件 FP 树中只剩下一个频繁项集时停止。

3.5 基于 *K*-Means 和 FP-Growth 的未知攻击特征规则提取

对于网络入侵中的未知攻击行为，在规则库中没有攻击的特征，无法对其进行识别，K-Means 只能根据同种攻击数据之间的相似性对多种攻击类型的数据进行聚类，而无法识别未知攻击具体的攻击类型，FP-Growth 算法可以通过对同种攻击类型的数据集进行规则挖掘，然后对挖掘出来的频繁项集和关联规则进行分析，得出未知攻击的特征规则，具体的实现步骤如下：

(1)首先，对多种未知攻击数据进行数据的预处理，将字符型数据进行数值转换。

(2)然后进行聚类分析，采用 K-Means 将预处理完的未知攻击数据中的同一种类型的攻击数据聚到一个簇里面，对于聚类簇数 K 值的确定，根据“簇内差异小，簇外差异大”的原则，采用轮廓系数来评估聚类的效果，从而将多种攻击分成了几个簇，每个簇里面包含一种类型的攻击。

(3)将聚类好的攻击数据与正常的数据进行对比，初步分析出攻击数据的特征，此步骤依赖人工的经验知识。

(4)FP-Growth 算法只能针对离散化的数据进行频繁项集和关联规则的挖掘，因此将人工提取出来的攻击数据特征进行分箱处理。

(5)对同种未知攻击的数据进行预处理之后，导入 FP-Growth 算法中，调整支持度和置信度，对攻击数据的特征和

规则进行提取。

模型图如图 1 所示。

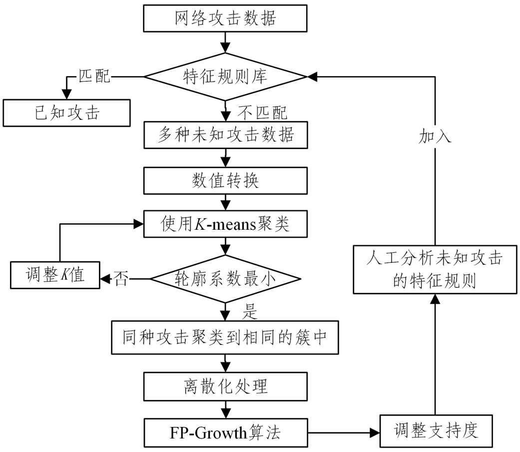


图 1　基于 *K*-Means 和 FP-Growth 的未知攻击规则提取模型图

Fig. 1　Extraction model diagram of unknown attack rules based on *K*-Means and FP-Growth

4　实验部分

4.1　实验过程

实验中将 python 作为编程语言，Jupyter lab 作为训练算法的工具，用到的 python 模块包括 Scikit-learn，Pandas，Matplotlib。采用 NSL-KDD 数据集中的未知攻击部分来进行下面的实验。

实验提取 NSL-KDD 数据集的测试集中的未知攻击部分，进行数据的预处理，将数据集中的非数值型特征进行数值转换，去掉标签列和表示分类难度的列，提取剩下的 41 列特征数据，构建 *K*-Means 聚类模型。在 *K*-Means 算法的调参上，n_cluster 为算法中的 k 值，即分为几簇，将 n_clusters 范围设置为 2～10 进行训练，n_clusters 越大，inertia 越小。引入模块 sklearn.metrics 中的类 sihouette_score 来计算轮廓系数，可以得到所有样本轮廓系数的均值，在此实验中，当 n_clusters 设置为 2～10 时，轮廓系数的值如图 2 所示。

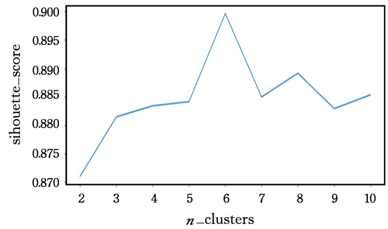


图 2　轮廓系数随 n_clusters 变化图

Fig. 2　Contour coefficient changes with n_clusters

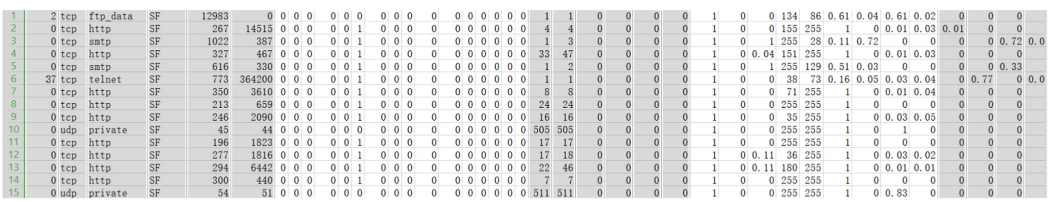
当 n_clusters 为 6 时，轮廓系数 sihouette_score 的值最大为 0.89，说明簇数为 6 时，聚类效果最好，此时每一簇中的样本数如表 1 所列。

表 1　*K*-Means 聚类结果

Table 1　*K*-Means clustering results

标签	样本数/条
0	996
1	737
2	685
3	331
4	293
5	133

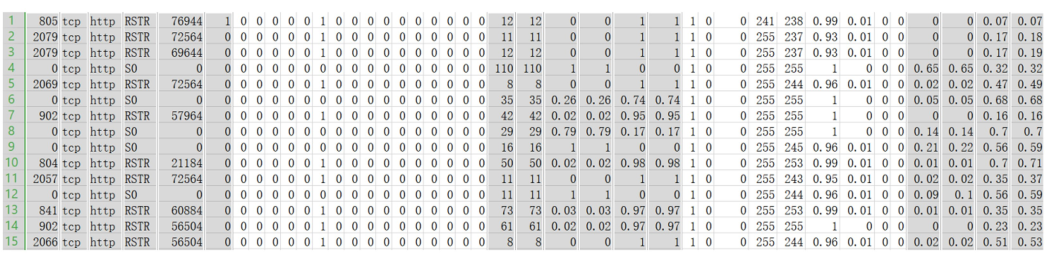
完成聚类后，每一类包含一种类型的未知攻击，接下来对标签为 1 和标签为 5 的未知攻击进行分析。首先，将未知攻击和正常的数据进行对比，通过人工分析，去除对识别该攻击类型的无用特征，下面用标签为 1 的攻击数据在 NSL-KDD 数据集中的特征来进行实验，正常数据和标签为 1 的攻击数据分别如图 3、图 4 所示。



1	2	tcp	ftp_data	SF	12983	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	0	0	134	86	0.61	0.04	0.61	0.02	0	0	0	
2	0	tcp	http	SF	267	14515	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	4	4	0	0	0	0	1	0	0	155	255	1	0	0.01	0.03	0.01	0	0	
3	0	tcp	smtp	SF	1022	387	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	3	0	0	0	0	1	0	1	255	28	0.11	0.72	0	0	0	0	0.72	0.0
4	0	tcp	http	SF	327	467	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	33	47	0	0	0	0	1	0	0.04	151	255	1	0	0.01	0.03	0	0	0	
5	0	tcp	smtp	SF	616	330	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	2	0	0	0	0	1	0	1	255	129	0.51	0.03	0	0	0	0	0.33	
6	37	tcp	telnet	SF	773	364200	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	0	0	38	73	0.16	0.05	0.03	0.04	0	0.77	0	0.0
7	0	tcp	http	SF	350	3610	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	8	8	0	0	0	0	1	0	0	71	255	1	0	0.01	0.04	0	0	0	
8	0	tcp	http	SF	213	659	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	24	24	0	0	0	0	1	0	0	255	255	1	0	0	0	0	0	0	
9	0	tcp	http	SF	246	2090	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	16	16	0	0	0	0	1	0	0	35	255	1	0	0.03	0.05	0	0	0	
10	0	udp	private	SF	45	44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	505	505	0	0	0	0	1	0	0	255	255	1	0	1	0	0	0	0	
11	0	tcp	http	SF	196	1823	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	17	17	0	0	0	0	1	0	0	255	255	1	0	0	0	0	0	0	
12	0	tcp	http	SF	277	1816	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	17	18	0	0	0	0	1	0	0.11	36	255	1	0	0.03	0.02	0	0	0	
13	0	tcp	http	SF	294	6442	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	22	46	0	0	0	0	1	0	0.11	180	255	1	0	0.01	0.01	0	0	0	
14	0	tcp	http	SF	300	440	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	7	7	0	0	0	0	1	0	0	255	255	1	0	0	0	0	0	0	
15	0	udp	private	SF	54	51	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	511	511	0	0	0	0	1	0	0	255	255	1	0	0.83	0	0	0	0	

图 3　正常数据集

Fig. 3　Normal data set



1	805	tcp	http	RSTR	76944	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	12	12	0	0	1	1	1	0	0	241	238	0.99	0.01	0	0	0	0	0.07	0.07
2	2079	tcp	http	RSTR	72564	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	11	11	0	0	1	1	1	0	0	255	237	0.93	0.01	0	0	0	0	0.17	0.18
3	2079	tcp	http	RSTR	69644	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	12	12	0	0	1	1	1	0	0	255	237	0.93	0.01	0	0	0	0	0.17	0.19
4	0	tcp	http	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	110	110	1	1	0	0	1	0	0	255	255	1	0	0	0	0.65	0.65	0.32	0.32
5	2069	tcp	http	RSTR	72564	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	8	8	0	0	1	1	1	0	0	255	244	0.96	0.01	0	0	0.02	0.02	0.47	0.49
6	0	tcp	http	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	35	35	0.26	0.26	0.74	0.74	1	0	0	255	255	1	0	0	0	0.05	0.05	0.68	0.68
7	902	tcp	http	RSTR	57964	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	42	42	0.02	0.02	0.95	0.95	1	0	0	255	255	1	0	0	0	0	0	0.16	0.16
8	0	tcp	http	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	29	29	0.79	0.79	0.17	0.17	1	0	0	255	255	1	0	0	0	0.14	0.14	0.7	0.7
9	0	tcp	http	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16	16	1	1	0	0	1	0	0	255	245	0.96	0.01	0	0	0.21	0.22	0.56	0.59
10	804	tcp	http	RSTR	21184	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	50	50	0.02	0.02	0.98	0.98	1	0	0	255	253	0.99	0.01	0	0	0.01	0.01	0.7	0.71
11	2057	tcp	http	RSTR	72564	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	11	11	0	0	1	1	1	0	0	255	243	0.95	0.01	0	0	0.02	0.02	0.35	0.37
12	0	tcp	http	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11	11	1	1	0	0	1	0	0	255	244	0.96	0.01	0	0	0.09	0.1	0.56	0.59
13	841	tcp	http	RSTR	60884	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	73	73	0.03	0.03	0.97	0.97	1	0	0	255	253	0.99	0.01	0	0	0.01	0.01	0.35	0.35
14	902	tcp	http	RSTR	56504	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	61	61	0.02	0.02	0.97	0.97	1	0	0	255	255	1	0	0	0	0	0	0.23	0.23
15	2066	tcp	http	RSTR	56504	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	8	8	0	0	1	1	1	0	0	255	244	0.96	0.01	0	0	0.02	0.02	0.51	0.53

图 4　攻击数据集

Fig. 4　Attack data set

每列的特征依次是连接维持的时间、协议类型、目的主机服务类型、连接状态、源主机发送给目标主机的字节数、目标主机发送给源主机的字节数、源地址和目的地址相同为 1（不同为 0）、错误分段的数量、紧急数据包个数；访问系统敏感文件和目录的次数、登录失败的次数、登录成功为 1（失败为 0）、compromised 出现次数、已 root 权限执行 shell 为 1（否则为 0）、执行 su root 指令为 1（否则为 0）、root 用户访问次数、创建文件的数量、shell 命令使用次数、访问控制文件的次数、

一个 ftp 会话出站连接次数、登录用户是否存在于 hot 列表、guest 用户登录为 1(否则为 0)；在过去两秒内，同主机的连接数、同服务的连接数、同主机出现“SYN”错误百分比、同服务出现“SYN”错误百分比、同主机出现“REJ”错误百分比、同服务出现“REJ”错误百分比、同主机中同服务的连接百分比、同主机中不同服务的连接百分比、同服务中不同主机的连接百分比；前 100 个连接中，同主机的连接数、同主机同服务的连接数、同主机同服务的连接百分比、同主机不同服务的连接百分比、同主机同源端口的连接百分比、不同源主机的连接百分比、同主机出现“SYN”错误百分比、同服务出现“SYN”错误百分比、同主机出现“REJ”错误百分比、同服务出现“REJ”错误百分比。

将攻击数据与正常的数据进行对比，可以看出，相比正常数据，攻击数据大部分连接持续时间较长，集中协议类型均为 tcp，目标主机服务类型均为 http，连接状态分为 RSTR，S0，S3，SF，正常连接的状态为 SF，源主机到目标主机的字节数较大，而目标主机没有返回给源主机的数据，在两秒内在连接中出现“SYN”错误和出现“REJ”错误的百分比大多数都大于 0，在前 100 个连接中，出现“REJ”的百分比和出现“SYN”的百分比大多数大于 0，在进行分析时不能确定是否有关系的特征：过去两秒内，与当前连接具有相同目标主机的连接数量与当前连接具有相同服务的连接数。提取出的正常数据和攻击数据相差较大的特征如表 2 所列。

表 2 正常数据和攻击数据相差较大的特征

Table 2 Characteristics with great difference of normal data and attack data

编号	特征	正常数据集	攻击数据集	类型
0	duration	连接时间较短	大部分连接时间较长	连续
1	protocol_type	tcp，icmp，udp	tcp	离散
2	service	ftp，http，private 等	http	离散
3	flag	SF 等多种状态	RSTR，S0，S3，SF	离散
4	src_bytes	字节数偏小	字节数偏大	连续
5	dst_bytes	返回正常字节数	基本为 0	连续
6	count	分布范围广	分布均匀	连续
7	src_count	分布范围广	分布均匀	连续
8	serror_rate	百分比大多数为 0	百分比大多数大于 0	连续
9	srv_serror_rate	百分比大多数为 0	百分比大多数大于 0	连续
10	rerror_rate	百分比大多数为 0	百分比大多数大于 0	连续
11	srv_rerror_rate	百分比大多数为 0	百分比大多数大于 0	连续
12	dst_host_serror_rate	百分比大多数为 0	百分比大多数大于 0	连续
13	dst_host_srv_serror_rate	百分比大多数为 0	百分比大多数大于 0	连续
14	dst_host_rerror_rate	百分比大多数为 0	百分比大多数大于 0	连续
15	dst_host_srv_rerror_rate	百分比大多数为 0	百分比大多数大于 0	连续

分析之后提取的特征包含 3 个离散变量和 13 个连续变量，在用 FP-Growth 算法进行分析时，需要将连续型变量离散化，对于不同的连续型数据，采用不同的离散化策略，使用 sklearn. prepocessing. KBinsDiscretizer 类中的 uniform 和 quantitle 进行不同离散化处理，其中 uniform 策略使得样本量之间具有相同的宽度，quantitle 策略使得每个特征上每箱的样本量基本相同。本实验对每个特征采取的离散化策略如表 3 所列。

由于每列数据之间重复的数值较多，因此给每列离散化之后的数据加上了不同的前缀，代表该数据所属的列，每一列的特征与表 2 保持一致，为之后出现的频繁项集的分析做准备。

对于预处理完成之后的数据，用 FP-Growth 算法对其中的未知攻击数据进行特征的挖掘。一共有 737 条攻击数据进行频繁项集和关联规则的挖掘。

表 3 特征分箱策略

Table 3 Feature binning strategy

特征	分箱策略	离散化结果
duration	uniform	[0，1 050，2 100]
src_bytes	uniform	[0，50 882，101 764]
dst_bytes	uniform	[0，127.5，255]
count	quantitle	[1，11，29，45，72.8，130]
src_count	quantitle	[1，11，29，45，72.8，130]
serror_rate	uniform	[0，0.5，1]
srv_serror_rate	uniform	[0，0.5.1]
rerror_rate	uniform	[0，0.5，1]
srv_rerror_rate	uniform	[0，0.5，1]
dst_host_serror_rate	uniform	[0，0.42，0.84]
dst_host_srv_serror_rate	uniform	[0，0.42，0.84]
dst_host_rerror_rate	uniform	[0，0.37，0.74]
dst_host_srv_rerror_rate	uniform	[0，0.37，0.74]

经过离散化处理之后的数据集展示如图 5 所示，特征对应表 2 中的数据。

1	0-0.0	tcp	http	RSTR	4-1.0	5-0.0	6-1.0	7-1.0	8-0.0	9-0.0	10-1.0	11-1.0	12-0.0	13-0.0	14-0.0	15-0.0
2	0-1.0	tcp	http	RSTR	4-1.0	5-0.0	6-1.0	7-1.0	8-0.0	9-0.0	10-1.0	11-1.0	12-0.0	13-0.0	14-0.0	15-0.0
3	0-1.0	tcp	http	RSTR	4-1.0	5-0.0	6-1.0	7-1.0	8-0.0	9-0.0	10-1.0	11-1.0	12-0.0	13-0.0	14-0.0	15-0.0
4	0-0.0	tcp	http	S0	4-0.0	5-0.0	6-4.0	7-4.0	8-1.0	9-1.0	10-0.0	11-0.0	12-1.0	13-1.0	14-0.0	15-0.0
5	0-1.0	tcp	http	RSTR	4-1.0	5-0.0	6-0.0	7-0.0	8-0.0	9-0.0	10-1.0	11-1.0	12-0.0	13-0.0	14-1.0	15-1.0
6	0-0.0	tcp	http	S0	4-0.0	5-0.0	6-2.0	7-2.0	8-0.0	9-0.0	10-1.0	11-1.0	12-0.0	13-0.0	14-1.0	15-1.0
7	0-0.0	tcp	http	RSTR	4-1.0	5-0.0	6-2.0	7-2.0	8-0.0	9-0.0	10-1.0	11-1.0	12-0.0	13-0.0	14-0.0	15-0.0
8	0-0.0	tcp	http	S0	4-0.0	5-0.0	6-2.0	7-2.0	8-1.0	9-1.0	10-0.0	11-0.0	12-0.0	13-0.0	14-1.0	15-1.0
9	0-0.0	tcp	http	S0	4-0.0	5-0.0	6-1.0	7-1.0	8-1.0	9-1.0	10-0.0	11-0.0	12-0.0	13-0.0	14-1.0	15-1.0
10	0-0.0	tcp	http	RSTR	4-0.0	5-0.0	6-3.0	7-3.0	8-0.0	9-0.0	10-1.0	11-1.0	12-0.0	13-0.0	14-1.0	15-1.0
11	0-1.0	tcp	http	RSTR	4-1.0	5-0.0	6-1.0	7-1.0	8-0.0	9-0.0	10-1.0	11-1.0	12-0.0	13-0.0	14-0.0	15-1.0
12	0-0.0	tcp	http	S0	4-0.0	5-0.0	6-1.0	7-1.0	8-1.0	9-1.0	10-0.0	11-0.0	12-0.0	13-0.0	14-1.0	15-1.0
13	0-0.0	tcp	http	RSTR	4-1.0	5-0.0	6-4.0	7-4.0	8-0.0	9-0.0	10-1.0	11-1.0	12-0.0	13-0.0	14-0.0	15-0.0
14	0-0.0	tcp	http	RSTR	4-1.0	5-0.0	6-3.0	7-3.0	8-0.0	9-0.0	10-1.0	11-1.0	12-0.0	13-0.0	14-0.0	15-0.0
15	0-1.0	tcp	http	RSTR	4-1.0	5-0.0	6-0.0	7-0.0	8-0.0	9-0.0	10-1.0	11-1.0	12-0.0	13-0.0	14-1.0	15-1.0

图 5 离散化之后的数据集

Fig. 5 Data set after discretization

4.2 实验结果与分析

将支持度和置信度从 1 开始调整，每次下降 0.01。当下降 0.01 时，无新的频繁项集和关联规则产生，则跳过该值，继续下调 0.01。

当支持度和置信度设置为 1 时，实验得出的频繁项集和关联规则如表 4 所列。

表 4 支持度和置信度设置为 1 时的频繁项集和关联规则

Table 4 Frequent itemsets and association rules when support and confidence are set to 1

频繁项集	关联规则
tcp	tcp->http
http	http->tcp

此结果表明，http 和 tcp 是每条记录都出现的特征值，说明该攻击使用的是 tcp 协议和 http 协议。

当支持度和置信度设置为 0.99 时，新增的频繁项集和关联规则如表 5 所列。

表 5 支持度和置信度设置为 0.99 时新增的频繁项集和关联规则

Table 5 Added frequent itemsets and association rules when support and confidence are set to 0.99

频繁项集	关联规则
'5-0.0'	'5-0.0'->tcp,http

'5-0.0'为第 5 列特征，即目标主机返回给源主机的字节数，结果表明 99%的样本中此特征的值在区间[0,127.5]范围内。

当支持度和置信度设置为 0.86 时，新增的频繁项集和关联规则如表 6 所列。

表 6 支持度和置信度设置为 0.86 时新增的频繁项集和关联规则

Table 6 Added frequent itemsets and association rules when support and confidence are set to 0.86

频繁项集	关联规则
'12-0.0'	'12-0.0'->tcp,'5-0.0'
'13-0.0'	—

'12-0.0'表示第 12 列特征值在区间[0,0.42]上，实验表明在 86%的样本中，100 个连接中同主机出现"SYN"错误的百分比在区间[0,0.42]上；'13-0.0'表示第 13 列特征值在区间[0,0.42]上，实验表明在 86%的样本中，100 个连接中同服务出现"SYN"的百分比在区间[0,0.42]上。

当支持度和置信度设置为 0.71 时，新增的频繁项集和关联规则如表 7 所列。

表 7 支持度和置信度设置为 0.71 时新增的频繁项集和关联规则

Table 7 Added frequent itemsets and association rules when support and confidence are set to 0.71

频繁项集	关联规则
'8-0.0'	—
'9-0.0'	'9-0.0'->tcp,http

'8-0.0'表示第 8 列特征值在区间[0,0.5]上，实验表明在 71%的样本中，两秒内同主机出现"SYN"错误的百分比在区间[0,0.5]上；'9-0.0'表示在第 9 列特征值在区间[0,0.5]上，实验表明在 71%的样本中，两秒内同服务出现"SYN"错误的百分比在区间[0,0.5]上。

当支持度和置信度设置为 0.70 时，新增的频繁项集和关联规则如表 8 所列。

表 8 支持度和置信度设置为 0.70 时新增的频繁项集和关联规则

Table 8 Added frequent itemsets and association rules when support and confidence are set to 0.70

频繁项集	关联规则
'10-1.0'	'10-1.0'->'9-0.0','5-0.0',http
'11-1.0'	—

'10-1.0'表示第 10 列特征值在区间[0.5,1]上，实验表明有 70%的样本在两秒内同主机出现"REJ"错误的百分比在区间[0.5,1]上；'11-1.0'表示第 11 列特征值在区间[0.5,1]上，实验表明在两秒内同服务出现"REJ"错误的百分比在区间[0.5,1]上。综上，随着支持度的降低，频繁项集和关联规则会逐渐增多，需要从中找出有价值的特征和规则，通过以上分析，可以得出该未知攻击类型的规则如下：1)在协议上类型上使用 tcp 协议；2)目标主机的网络服务类型为 http；3)目标主机返回给源主机的字节数较少，在区间[0,127.5]范围内；4)在 100 个连接中，相同目标主机和相同服务中 86%的样本会出现"SYN"错误，百分比区间为[0,0.42]；5)在两秒内相同目标主机和相同服务中，71%的样本会出现 "SYN"错误，百分比区间为[0,0.5]；6)在两秒内，相同目标主机和相同服务中，70%的样本会出现的"REJ"错误，百分比区间为[0.5,1]。

下面验证提取的特征规则，选取正常数据 3500 条，和以上分析类型的未知攻击数据 737 条数据，分别计算下面的实验指标：TP(True Positive)表示被正确分类的正常样本；FN(False Negative)表示被错误分到未知攻击的正常样本；TN(True Negative)表示被正确分类的未知攻击样本；FP(False Positive)表示被错误分到正常的未知攻击样本。准确率(Accuracy)、召回率(Recall)、精准率(Precision)、F1 分数计算式如下：

$$Accuracy=\frac{TP+TN}{TP+FP+TN+FN} \tag{9}$$

$$Recall=\frac{TP}{TP+FN} \tag{10}$$

$$Precision=\frac{TP}{TP+FP} \tag{11}$$

$$F1=\frac{2*Precision*Recall}{Precision+Recall} \tag{12}$$

准确率即被正确分类的样本所占的比例，召回率即被正确分类到正常样本占所有正常样本的比例，精准率即被正确分类到正常样本占全部分类为正常样本的比例，F1 分数是模型精确率和召回率的一种调和平均。

F1 值作为综合评估指标，可以平衡准确率和召回率的影响，不同支持度和置信度下的 F1 值变化散点图如图 6 所示，数据与表 9 对应。

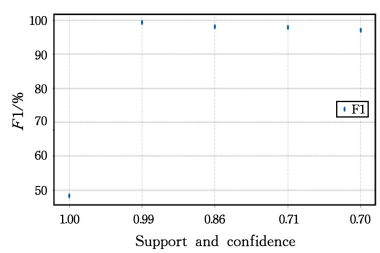


图 6 未知攻击 1 的 F1 值变化散点图

Fig. 6 Scatter plot of F1-Score change of unknown attack 1

表 9 不同支持度和置信度下的实验指标

Table 9 Experimental indexes under different support and confidence

Support and Confidence	*Accuracy*/%	*Recall*/%	*Precision*/%	*F1*/%
1	43.66	31.80	1	48.25
0.99	8.74	98.57	99.91	99.23
0.86	96.60	98.33	97.19	98.01
0.71	96.34	98.91	96.73	97.81
0.70	94.87	99.14	94.88	96.96

支持度和置信度为 1 时,实验指标数据表明提取特征的效果不佳,此特征规则不能很好地识别攻击,说明这种特征规则在正常样本中也是合适的;而当置信度设置为 0.99 时,提取出来的特征规则检测该攻击达到了较好的效果,随着特征规则增多,准确率、精准率和 F1 值在下降,说明特征规则选择超过合适的数量之后,可能会漏掉一些攻击样本,使得识别的效果变差。

将不同算法的实验结果进行对比,结果如表 10 所列。

表 10 不同算法的实验结果对比

Table 10 Comparison of experimental results of different algorithms

算法	准确率/%	时间/s
K-Means,Apriori[13]	92.62	5.99
聚类分析与关联规则结合技术[14]	92.03	—
本文算法	98.74	0.51

通过对比以上数据可知,本文算法有较高的准确率,将 *K*-Means 和 FP-Growth 算法结合,提取未知攻击特征规则进行识别,通过对支持度和置信度的调整,然后人工分析提取出未知攻击的特征规则,准确率可达 98.74%,说明聚类的效果较好,人工提取的特征具有很好的参考价值,与其它文献中的模型相比,本文选择的特征更加合理,在调整支持度和置信度后,选取的值能够得出用于检测未知攻击的特征规则,并且准确率较高。FP-Growth 与 Apriori 比较,减少了对内存的读取次数,算法所需要的时间更短,在时间上也占有一定的优势。

结束语 随着互联网的高速发展,网络入侵行为千变万化,也出现了很多新型攻击,由于规则库里不包含未知攻击的特征规则,无法对其进行识别,因此本文从大量数据中发现规律,挖掘出未知攻击的规则,达到对未知网络入侵进行识别的效果。此方法对未知攻击的识别率较高,但是也存在一定的问题,如在数据离散化处理的过程中造成了一定的信息损失;面对海量的数据,对于攻击行为中一些出现次数不多的特征值分析不足。最后本文使用 NSL-KDD 数据集得到的仿真结果在对实际网络的适应性上还需要加强。针对以上模型的不足之处,未来的研究工作可以从以下几个方面展开:

(1)采用监督学习和无监督学习相结合的方式实现网络入侵检测,利用监督学习对已知攻击进行识别,对于分类模型无法识别的攻击,采用无监督学习的方法进行聚类分析。

(2)深度学习技术不需要复杂的特征工程,对未知攻击具有识别能力,且泛化能力强。

(3)用自动化的方式对未知攻击的特征进行提取,以算法来补充专家经验的稀缺性,挖掘出大量数据中隐藏的规律。

参 考 文 献

[1] WANG S. Research of intrusion detection based on an improved K-means algorithm[C]// 2011 Second International Conference on Innovations in Bio-inspired Computing and Applications. IEEE,2011:274-276.

[2] CHEN Y,ZHANG M J,XU F J. HTTP slow DoS attack detection method based on one-dimensional convolutional neural network[J]. Journal of Computer Applications,2020,40(10):2973-2979.

[3] ZHANG Z,LIU Q,QIU S,et al. Unknown Attack Detection Based on Zero-Shot Learning[J]. IEEE Access,2020,8:193981-193991.

[4] ZHENG M X. Research on Intrusion Detection and Defense of Campus Network Based on Clustering[D]. Hangzhou:Zhejiang University,2020.

[5] LI E Y. Research on Intrusion Detection System Based on Classic Clustering Algorithm and Association Algorithm [D]. Chongqing:Chongqing University of Posts and Telecommunications,2020.

[6] ZHAO S. Research on Intrusion Detection System Based on Cluster Analysis and Association Rules[D]. Tangshan:North China University of Technology,2019.

[7] SAIED A,OVERILL R E,RADZIK T. Detection of known and unknown DDoS attacks using Artificial Neural Networks[J]. Neurocomputing,2016,172(C):385-393.

[8] CASAS P,MAZEL J,OWEZARSKI P. Unsupervised network intrusion detection systems:Detecting the unknown without knowledge[J]. Computer Communications,2012,35(7):772-783.

[9] LOBATO A G P,LOPEZ M A,SANZ I J,et al. An adaptive real-time architecture for zero-day threat detection[C]// 2018 IEEE International Conference on Communications (ICC). IEEE,2018:1-6.

CAO Yang-chen, born in 1996, postgraduate. Her main research interests include machine learning and network traffic analysis.

ZHU Guo-sheng, born in 1972, Ph. D, professor. His main research interests include next-generation Internet and software-defined networks.