计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.191000193

基于粗糙集聚类的报文格式推断方法

李毅豪　洪　征　林培鸿　冯文博
中国人民解放军陆军工程大学　南京 210000
(enhancelee@foxmail.com)

摘　要　报文聚类是报文格式推断的基础,现有的报文聚类方法大多以报文的全局相似性为聚类的标准,这类聚类方法的准确率往往不高,进而影响后续报文格式提取的准确率。针对这一问题,文中提出了一种基于粗糙集聚类的报文格式推断方法,该方法包括预处理、基于粗糙集的聚类、特征词提取和报文格式推断4个阶段。首先,通过数据预处理分离出目标报文中的业务类报文和控制类报文;其次,按照粗糙集理论中基于属性划分样本的方法对报文的统计特征进行聚类,这种聚类方法能够准确获取报文序列的局部特征,能够达到较好的聚类效果;然后,根据长度、频率和位置特征来提取协议特征词;最后,将协议特征词分为必选字段和可选字段,并用它们来描述报文格式。实验结果表明,该方法能够准确地获取协议的报文格式。

关键词:协议逆向工程;报文聚类;报文格式推断;粗糙集理论;特征词提取
中图法分类号　TP398.08

Message Format Inference Method Based on Rough Set Clustering

LI Yi-hao,HONG Zheng,LIN Pei-hong and FENG Wen-bo
Army Engineering University of PLA,Nanjing 210000,China

Abstract　Message clustering is an important procedure of message format inference. Most of the existing message clustering methods take message global similarity as the clustering criteria. However,the accuracy of such clustering methods is often not high enough,and affects the accuracy of subsequent message format extraction. To solve this problem,this paper proposes a message format inference method based on rough set clustering,which consists of preprocessing phase,rough-setbased clustering phase,feature word extraction phase and message format extraction phase. Firstly,messages are separated into business messages and control messages. Secondly,messages are clustered on the basis of position attributions according to rough set theory,and the clustering method considers local features of message sequences which ensures high accuracy of message clustering. Thirdly,protocol feature words are extracted according to length,frequency and position characteristics. Finally,protocol feature words are classified into mandatory fields and optional fields,and they are used to represent message formats. Experimental results show that the proposed method can extract message formats precisely.

Keywords　Protocol reverse engineering,Message clustering,Messages format inference,Rough set theory,Feature word extraction

1　引言

协议逆向工程[1]是指在不依赖协议描述的情况下,通过对协议实体的网络输入输出、系统行为和指令执行流程进行监控和分析,来提取协议语法、语义和同步信息的过程。它被广泛应用于网络协议仿真、软件安全审计、恶意软件协议分析和协议漏洞挖掘等领域[2]。

报文格式推断是协议逆向工程提取协议语法的过程。报文的格式通常由若干个固定字段和可变字段构成,而报文格式推断技术能够通过分析捕获的网络流量来获得未知协议的格式信息。

1.1　相关工作

PI项目(Protocol Information Project)[3]是一个经典的协议逆向项目。PI使用生物信息学中的序列比对算法来度量报文的相似性,然后建立相似性矩阵,并使用非加权成对群算数平均法(Unweighted Pair Group Method with Arithmetic Mean,UPGMA)对报文进行聚类,推导出目标协议的报文格式。但是,使用序列比对算法衡量报文相似性的过程既费时又消耗内存。

He等[4]和Lu等[5]在序列比对算法的基础上进行了改进,他们通过增加报文在序列比对时连续匹配的奖励,使得在衡量报文相似度时连续字节匹配的权重更高,从而实现更优

到稿日期:2019-10-29　返修日期:2020-04-12　本文已加入开放科学计划(OSID),请扫描上方二维码获取补充信息。
基金项目:国家重点研发计划基金资助项目(2017YFB0802900)
This work was supported by the National Key R&D Program of China (2017YFB0802900).
通信作者:洪征(hz5215@163.com)

化的报文相似度衡量方法。但是,在报文长度可变的情况下,使用该算法衡量报文的相似性会产生不公平的问题,这会导致聚类效果不好,进而使得报文格式推断的准确率较低。

Li 等[6]提出了一种基于报文顺序对协议报文进行聚类的方法。这种方法将网络会话中出现顺序相同的报文聚类在一起,然后对报文进行分析,之后对网络实体进行模糊测试。但是在网络会话中处于相同位置的报文可能并不是同一种类型,这就会导致报文聚类的精度较低,进而影响后续报文格式推断以及模糊测试过程。

Li 等[7]将报文转换为二进制图像,并根据图像的相似性利用改进的 DBSCAN 算法对报文进行聚类。但是其仍然依赖专家的专业知识来定义图像的长度和宽度,除此之外聚类算法的参数设置对聚类结果的精度有着重要影响,进而将影响后续特征词的提取以及报文格式推断的精度。

现有的报文格式推断方法在进行报文聚类时大多以衡量报文的全局相似性为主,忽略了报文的局部特征,导致聚类准确率和召回率不高,从而影响报文格式推断的准确率。

1.2 主要贡献

本文提出了一种基于粗糙集聚类的报文格式推断方法。该方法由预处理、基于粗糙集的聚类(Rough Set Theory Clustering,RSTC)、特征词提取和报文格式推断等 4 个阶段组成,该方法弥补了手工分析费时、繁琐、易出错等缺点。相比现有的自动化协议逆向方法,所提方法通过基于粗糙集的聚类方法对报文进行聚类,从而得到高准确率和高召回率的簇。基于这些簇提取关键词,再进行消息格式推断,有利于得到高准确率的推断结果。

2 方案设计

2.1 预处理阶段

在预处理阶段,首先将目标协议的流量包(Packet)从数据流中提取出来。应用层报文一般由 TCP 或 UDP 进行传输,每个 UDP 包的载荷被视为一个单独的应用层报文(Message)[8]。对于通过 TCP 传输的数据包,依据 TCP FIN 标志和 TCP SYN 标志将一个新的应用层报文与之前的应用层报文分隔开[9]。据此,我们将捕获的数据包转化为报文。

应用层报文通常可以分为控制类报文和业务类报文两种[10]。控制类报文是指与指令相关的报文,要求接收端根据报文内容执行相应指令,如 HTTP 中的"GET"类型报文,其功能是请求 HTTP 服务器发送相应页面到客户端。业务类报文是指传输数据负载的报文。如果同时分析这两种类型的报文,业务型报文中的数据内容容易与控制型报文中的关键词字段混淆,影响格式推断的准确率。如 HTTP 协议中"content=/0x01/0x02/0x03……"就属于业务型报文,这种报文的格式往往比较简单,可以单独分析处理。本文将控制类报文与业务类报文分隔开来,重点分析相对复杂的控制类报文。

协议设计者在设计协议时,为了提高网络实体解析协议的速度,通常会将控制类报文设计得较短。但是业务类报文传输的是通信数据,无法缩减数据内容,因此业务类报文往往偏长。基于这一特点,我们设计了一个长度阈值 $type\_threshold$ 来区分业务类报文和控制类报文。如果报文长度高于这一阈值,则判断该报文为业务类报文,如果报文长度低于这一阈值,则判断该报文为控制类报文。

在此基础上,使用分词算法将得到的控制型报文分割成小段。分词算法采用了 ProWord[11] 中的专家投票(Voting Experts,VE)算法来获取准确的分词结果。

本文将分词结果中的所有元素作为候选特征词,用 $cword_i(i\in N)$ 来表示,字段用 $field_i(i\in N)$ 来表示。例如,在 x_1 的分词结果中,"GET"在分词结果中处于第 1 个字段的位置,其位置表示为 $field_1$,"www.baidu.com"在分词结果中处于第 2 个字段的位置,其位置表示为 $field_2$。分别用 COWRD 和 FIELD 表示候选特征词集合与分词位置序列,$COWRD=\{cword_1,cword_2,\cdots,cword_n\}$,$FIELD=\langle field_1,field_2,\cdots,field_n\rangle$。

表 1 中,分词结果集合为{〈"GET","www.baidu.com","HTTP/1.1"〉,〈"POST","www.sina.cn","HTTP/1.1"〉,〈"GET","www.taobao.com","HTTP/1.1"〉}。候选特征词集合 CWORD={"GET","POST","www.baidu.com","www.sina.cn","www.taobao.com","HTTP/1.1"}。分词位置序列 $FIELD=\langle field_1,field_2,field_3\rangle$。

表 1 分词结果的示例

Table 1 Example of segmentation result

	$field_1$	$field_2$	$field_3$
x_1	GET	www.baidu.com	HTTP/1.1
x_2	POST	www.sina.cn	HTTP/1.1
x_3	GET	www.taobao.com	HTTP/1.1

2.2 基于粗糙集的聚类

报文聚类的目的是将同一类型的报文聚集在一起,为了提高报文聚类的准确率,本文设计了基于粗糙集的报文聚类方法。预处理阶段获得了候选词集合(COWRD)、分词位置序列(FIELD)以及分词结果集合,将这 3 个结果作为报文聚类的输入。

2.2.1 基于粗糙集聚类的基本思想

粗糙集理论是一种处理不确定性的数学理论[12],本文先介绍粗糙集理论的基本概念,并通过一个示例来说明基于粗糙集理论的聚类方法。

利用粗糙集理论解决问题,需要构建信息系统(Information System)S,它是一个四元组:

$$S=(U,Q,V,f) \tag{1}$$

其中,U 是样本的集合,Q 是属性的集合,V 是样本对应属性的值的集合,f 代表一个样本的属性到该属性取值的映射。

式(1)中,集合 U 包含了所有的待分析样本,本文使用 $x_i(i\in N)$ 表示待分析的样本,则 $U=\{x_1,x_2,\cdots,x_n\}$。属性集 Q 包含了样本的所有属性,属性集 Q 中的元素用 $a_j(j\in N)$ 来表示,属性集 $Q=\{a_1,a_2,\cdots,a_m\}$。V_{a_i} 表示属性 a_i 的值的集合。对于每一个 $a_i\in Q$,都有一个 V_{a_i},所有 V_{a_i} 的集合构成了属性值集 V,可以表示为 $V=\bigcup_{i=1}^{n}V_{a_i}$。$f$ 代表一种映射关系,它是一个样本的属性到该属性取值的映射。

$$f:U\times Q\rightarrow V \tag{2}$$

定义 1(不可分辨关系,Indiscernible Relation) x_i 与 x_j

在属性 a_k 上是不可分辨关系当且仅当 $f(x_i,a_k)=f(x_j,a_k)$。

考虑这样一个属性 a_i，如果有两个样本 x_i 和 x_j，对于该 a_i 有 $f(x_i,a_i)=f(x_j,a_i)$，即 x_i 和 x_j 在属性 a_i 上具有不可分辨关系，就将 x_i 和 x_j 放进同一个簇(Cluster)中，这是粗糙集聚类的基础。

一个属性集合 A 中可能包含多个属性 a_i，根据不同属性对样本集合进行聚类，可以得到不同的聚类结果。每一个聚类结果记作 $E_i(i\in N)$。每一个聚类结果都包含多个簇，簇记作 $e_i(i\in N)$。聚类结果的集合记作 U/A。

$$U/A=\{E_1,E_2,E_3,\cdots,E_n\} \tag{3}$$

$$E_i=\{e_1,e_2,e_3,\cdots,e_n\} \tag{4}$$

$$\bigcup_{i=1}^{n}e_i=U$$

$$\forall i\neq j,e_i\cap e_j=\varnothing \tag{5}$$

定义 2(聚类结果，Clustering Result) 聚类结果是指完成一次聚类后得到的集合，其中包含多个簇，且每个簇的交集为空，并集为样本全集。

下面给出一个基于粗糙集进行聚类的例子。设有 4 块积木，它们都有颜色、形状和大小 3 种属性，这 4 块积木在对应的 3 种属性上有不同的值，如表 2 所列。

表 2 积木的属性和值

Table 2 Attributions and values of building blocks

	a_1(颜色)	a_2(形状)	a_3(大小)
x_1	红	圆形	大
x_2	蓝	方形	大
x_3	红	方形	小
x_4	蓝	方形	小

在本例子中，$U=\{x_1,x_2,x_3,x_4\}$，$Q=\{a_1,a_2,a_3\}$，$V=$ {红，蓝，圆形，方形，大，小}。如果 A 中只有一个元素 a_1，即 $A=\{a_1\}$，那么 $U/A=\{\{x_1,x_3\},\{x_2,x_4\}\}$，其中 $\{x_1,x_3\}$ 和 $\{x_2,x_4\}$ 是两个簇，$\{\{x_1,x_3\},\{x_2,x_4\}\}$ 是一个聚类结果。聚类结果的数量与 A 中的元素数量一致。由于 x_1 和 x_3 的 a_1(颜色)属性都为红，因此它们在 a_1 属性上是不可分辨关系，故将它们聚为一个簇，同理 x_2 和 x_4 也是如此。如果此时 A 中有两个元素 $A=\{a_1,a_2\}$，那么 $U/A=\{\{\{x_1,x_3\},\{x_2,x_4\}\},\{\{x_1\},\{x_2,x_3,x_4\}\}\}$，这个结果是 $U/\{a_1\}$ 与 $U/\{a_2\}$ 的集合。

在存在多种聚类结果的情况下，我们需要根据聚类结果中簇内样本的数量，来决定需要保留的聚类结果和需要淘汰的聚类结果。

传统的聚类方法一般先衡量样本的相似性，然后建立样本间相似性矩阵进行聚类操作，但是粗糙集理论根据样本的属性对样本进行聚类，这种聚类方式不是根据样本间的“相似”，而是根据样本属性的“相同”进行聚类，基于粗糙集的聚类方式相比传统的聚类方式更为合理。

在报文聚类领域，相比基于序列比对的聚类方式，通过属性聚类的方式可以达到更高的准确率。因为序列比对算法在计算相似度的过程中，很难察觉到一两个字节的差别可能导致的报文种类的不同。如 HTTP 中状态码“200”表示请求报文成功接收，而状态码“400”则表示请求报文发生错误，这两个状态码之间仅仅只存在一个字节的差别。而基于粗糙集理论的聚类方法则要求样本在该属性的值完全一致才能将样本进行聚类，因此能够取得较好的聚类效果。

本文提出的基于粗糙集理论的聚类方法，由按属性分类和聚类结果简化两个步骤组成，下面将介绍这两个步骤的主要工作。

2.2.2 按属性分类

基于粗糙集理论的聚类方法，首先需要构建解决粗糙集问题的信息系统 S。本文将数据预处理阶段得到的每条报文的分词结果组成样本集合 U，如表 1 中的〈“GET”，“www.baidu.com”，“HTTP/1.1”〉就是第一条报文的分词结果，此分词结果即为样本 x_1。将候选词集合作为信息系统的属性值集合 V。

定义 3(邻接属性，Contiguous Attributions) 邻接属性是指在由多个属性构成的属性序列中，将相邻的几个属性结合形成的一个属性。

例如，有一个属性序列 $A=\langle a_1,a_2,a_3\rangle$，属性 a_1 与属性 a_2 在属性序列 A 中是相邻的，根据定义 3，将属性 a_1 与属性 a_2 相结合而形成的 $a_1\ a_2$ 称为邻接属性。

定义 4(N-邻接属性集，N-Contiguous Attributions Set，N-CAS) N-邻接属性集是指在属性序列 A 中，数量不超过 N 的邻接属性的集合。

例如，有属性序列 $A=\langle a_1,a_2,a_3\rangle$，根据定义 4，$2\text{-}CAS=\{a_1,a_2,a_3,a_1a_2,a_2a_3\}$。本文将信息系统 S 的属性集合 Q 定义为关键词位置序列 $FIELD$ 的 N-邻接属性集。

在数据预处理阶段，专家投票算法(Voting Experts，VE)算法可能因参数设置而导致分词粒度过小，即本来应该作为一个整体的候选特征词被分成了多个，候选特征词被分成的多个部分对应于不同的 *field*。为了使这些 *field* 结合成一个整体，本文提出了 N-邻接属性集的定义。N-邻接属性集通过结合相邻的 *field* 属性，能够在聚类过程中避免分词粒度小而引发的聚类准确率低的问题。

根据样本在属性集的各属性上的不可分辨关系对样本进行聚类，由于在属性集中有多少个元素就会有多少种聚类结果，因此该步骤完成时，结果中可能包含多个聚类结果。

算法 1 基于粗糙集的聚类算法

输入：S

输出：result

```
1. for a in Q:
2. h=0
3.   for any x_{1,2,3,...,n} in U:
4.     k=0
5. /* 如果 x_{1,2,3,...,m} 在属性 a 上为不可分辨关系，则将这些样本聚成一类 */
6.     if f(x_1,a)=f(x_2,a)=...=f(x_m,a):
7.       cluster x_1,x_2,...,x_m as cluster k
8.       k=k+1
9.     end if
10.    (U/a)_h= set(cluster_0,cluster_1,...,cluster_k)
11.    result.append((U/a)_h)
12.    h=h+1
```

13. end for
14. end for

算法 1 中，第 3－9 行表示在属性 a_i 确定的情况下，如果样本 $x_1, x_2, \cdots, x_m$ 是不可分辨关系，则将这些样本聚成一个簇；第 10－13 行表示对于每一个属性 a_i 都将有一个聚类结果；第 14 行表示最终将所有的聚类结果的集合作为输出。

2.2.3 聚类结果简化

一些属性并不合适作为报文聚类的标准，为了避免以这些不适合的属性为标准进行聚类，需要对上一步工作的结果 (U/Q) 进行简化。

每一个聚类结果 E_i 中都包含若干个簇 e_i。理想的聚类所形成的每一个 e_i 代表样本中的一种类型。如果一个属性的取值越多样，那么 e_i 的数量就较多，e_i 中的样本数量就较少；如果一个属性的取值越少，那么 e_i 的数量就较少，e_i 中的样本数量就相对较多。

考虑到一种协议的报文类型是有限的，相应地，理想的聚类结果 E_i 所得到的簇 e_i 的数量也不会太多。由于簇的数量少，在报文数量足够大的情况下，每个簇中的报文数量应该足够多。如果某个 e_i 中的报文数量很少，说明该簇对应的 $field$ 值出现的次数少，这个值在该 $field$ 的所有值中出现的概率低，其类型可能代表噪声，因此这样的 e_i 不能代表一种报文的类型，应该将其淘汰。

每个聚类结果 E_i 包含多个簇 e_i，簇 e_i 中包含若干个报文。在淘汰较小的簇之后，本文将所有聚类结果 E_i 中的所有簇 e_i 整合成为一个最终结果 SE，其中简化后的簇表示为 se_i。尽管 se_i 中的报文可能有重复，但是对后续的特征词提取却是有帮助的。

$$SE=\{se_1, se_2, se_3, \cdots, se_n\} \tag{6}$$

此时的簇是由在某位置属性上拥有相同值的报文构成的，而没有考虑报文的相似性。这种做法能够有效地分离报文内容相似但是报文类型不相同的报文。

例如有 7 条报文，经过分词后得到的结果如表 3 所列。

表 3 聚类结果简化的例子

Table 3 Example of simplifying clustering results

	$field_1$	$field_2$	$field_3$
x_1	GET	AAA	HTTP/1.1
x_2	GET	BBB	HTTP/1.0
x_3	GET	CCC	HTTP/1.0
x_4	GET	DDD	HTTP/1.0
x_5	POST	EEE	HTTP/1.1
x_6	POST	FFF	HTTP/1.1
x_7	POST	GGG	HTTP/1.1

根据每个位置属性聚类后得到的结果 $E_1=\{\{x_1, x_2, x_3, x_4\}, \{x_5, x_6, x_7\}\}$，$E_2=\{\{x_1\}, \{x_2\}, \{x_3\}, \{x_4\}, \{x_5\}, \{x_6\}, \{x_7\}\}$，$E_3=\{\{x_1, x_5, x_6, x_7\}, \{x_2, x_3, x_4\}\}$。由于 E_2 中的每个簇都只包含一个元素，因此需要对其进行简化，认为 E_2 中的簇都是按照不合适的位置属性中的值进行聚类的，因此将这些簇全部淘汰，并对剩余的簇进行整合，最终得到 $SE=\{\{x_1, x_2, x_3, x_4\}, \{x_5, x_6, x_7\}, \{x_1, x_5, x_6, x_7\}, \{x_2, x_3, x_4\}\}$。

2.3 特征词提取阶段

候选特征词集合 $COWRD$ 在预处理阶段通过 VE 算法获得。一般来说，协议特征词有以下 3 个特征：固定的位置、合适的长度以及在所在簇中的高支持度(Support Degree)。本阶段将根据这 3 个特征为每个候选特征词打分，从而提取出真正的协议特征词。

2.3.1 位置分数

通常来说，协议设计者出于方便解析协议、提高网络实体工作效率等目的，常常将协议关键词放在报文的固定位置上。根据这一特点，我们设计了位置分数，用 pos_score 来表示。

$$pos\_score=\begin{cases}1, & \text{only one pos}\\ 0, & \text{only appear once}\\ \dfrac{1}{\max(pos)-\min(pos)+1}, & \text{at least two different pos}\end{cases} \tag{7}$$

式(7)中，$\max(pos)$ 和 $\min(pos)$ 分别代表该候选词在簇中的分词结果位置上的最大值和最小值，即该候选词的位置属性($field_i$)中的最大值与最小值。如果该候选词仅在簇中出现过一次，那么位置分数记为 0。如果该候选词在簇中出现过一次以上，且是在同一个位置出现的，该分数记为 1。如果在不同位置出现，则记为 $1/(\max(pos)-\min(pos)+1)$。

例如，有两个分词结果，它们是同一个簇中的内容：

分词结果 1："GET""User-Agent""Host"；

分词结果 2："GET""Host""User-Agent"。

计算"Host"的位置分数的步骤如下："Host"在该簇中出现两次，在第一条中出现的位置是 $field_3$，在第二条中出现的位置是 $field_2$，由于 $field_i=i$，因此 $\max(pos)=2$，$\min(pos)=1$，该候选词的位置分数为 1/2。

2.3.2 长度分数

适合的长度是协议特征词的一个重要特点。一般来说，协议特征词不能够太短或者太长。如果一个协议特征词太长，那么网络实体对协议进行解析的时间耗费就很大，会消耗大量计算资源；如果协议特征词太短，就容易与数据部分混淆从而引发协议漏洞。为了衡量协议特征词的长度，本文设计了长度分数，记作 len_score。

$$len\_score=\begin{cases}1, & len\_min\leqslant len(cword)\leqslant len\_max\\ \dfrac{1}{len\_min-\mathrm{len}(cword)+1}, & len(cword)<len\_min\\ \dfrac{1}{\mathrm{len}(cword)-len\_max+1}, & len(cword)>len\_max\end{cases} \tag{8}$$

其中，len_min 和 len_max 是两个参数，它们表示特征词允许的最大长度与最小长度。如果候选特征词的长度大于 len_max，那么它的分数记为 $1/(len(cword)-len\_max+1)$；如果它的长度小于 len_min，那么它的分数记为 $1/(len\_min-len(cword)+1)$。

2.3.3 支持度分数

支持度是指在样本集合中，包含某个给定元素的样本的数量占总样本数量的比例。假设有一个集合，用 C 表示，且其中包含了多个样本 x_i，而每一个样本包含若干个元素 e_i。

$$C=\{x_1,x_2,x_3,\cdots,x_n\} \tag{9}$$

$$x=\{e_1,e_2,e_3,\cdots,e_n\} \tag{10}$$

接下来设拥有某个特定元素 e_i 的样本为 $x|e_i$，在集合 C 中包含该元素的所有样本的集合记作 $C|e_i$。

$$C|e_i=\{x_{i1}|e_i,x_{i2}|e_i,\cdots,x_{in}|e_i\} \tag{11}$$

那么该特定元素 e_i 的支持度表示为：

$$sup(C,ei)=\frac{card(C|ei)}{card(C)} \tag{12}$$

其中，$card(C)$表示集合 C 中元素的数量，$card(C|e_i)$表示集合 $C|e_i$ 中元素的数量。

每一个特征候选词的支持度分数都通过其所在簇计算出来，该支持度分数记作 sup_score。在计算出每一个候选特征词的支持度分数后，可以通过特征词的闭合[13]特征进一步对候选特征词集合进行简化。

定义 5（闭合候选特征词） 候选特征词（$Cword_i$）是闭合的当且仅当在特征词集合（$CWORD$）中，没有任何一个其他特征词是 $Cword_i$ 超序列并且其支持度等于 $Cword_i$ 的支持度。

根据定义 5，候选特征词集中的所有候选特征词都必须满足闭合候选特征词的条件，如果不满足该条件，则从候选特征词集中删除该候选特征词。这样做的目的是减少分词过程中的过分词现象，从而使得候选特征词集更小，确保能够精确地提取真正的协议特征词。

2.3.4 合计分数与排名

每一个候选特征词得到位置、长度与支持度分数之后，就需要计算其合计分数。其通过对这 3 个分数进行求和得到，记作 $total\_score$。

$$total\_score=pos\_socre+len\_score+sup\_score \tag{13}$$

一个候选特征词的 $total\_score$ 越高，表明该候选特征词是协议特征词的可能性越大。同一个候选特征词可能出现在不同的簇中，本文取每个候选特征词在所有簇中的最高分数作为该候选特征词的分数。

2.4 报文格式推断阶段

在报文格式推断阶段，将使用 Ding 等[14]提出的报文格式模型，利用前一阶段提取的协议特征词集构建报文格式。

2.4.1 报文格式模型

通常来说，报文格式是由若干个特征词字段（记作 $feature_i(i\in N)$）和数据字段（记作 $data_i(i\in N)$）构成的，并且这些字段一般在报文中交替出现，且 $feature_i$ 和 $data_i$ 都可能为空。这是由于网络协议实体在解析协议时，一般首先解析处理后续数据的方法，再利用对应方法解析紧跟在方法后的数据。报文格式模型如图 1 所示。

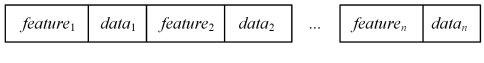

图 1 报文格式模型

Fig. 1 Message format model

例如，HTTP 的报文格式如图 2 所示。在格式的首行，“Method”是一个特征词字段，而 URL 是一个数据字段，在后面的几行中，“Key”是特征词字段，“Value”是一个数据字段。第一行的“Version”是一个特征词字段，而其后的数据字段为空。

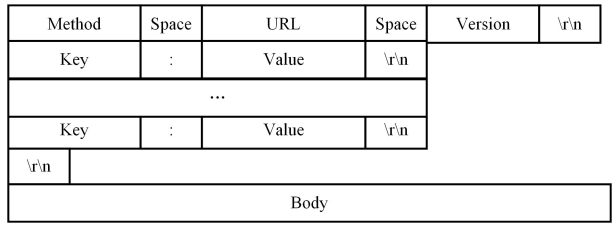

图 2 HTTP 的报文格式

Fig. 2 Message format of HTTP

2.4.2 报文格式推断

本文将使用有序的特征词来表示特征词字段，利用双下划线（“__”）表示数据字段。

本文根据提取出的协议特征词，对所有报文分词结果进行遍历，找出该协议特征词在报文分词结果中所处的 $field_i$ 位置。根据报文格式模型，将协议特征词填入模型的相应位置，通过此方法获得报文格式。

除此之外，对于特征词字段来说，区分必选字段（Mandatory Field）和可选字段（Optional Field）也十分重要[15]。

设簇内第 i 条报文的第 j 个协议特征词为 $feature_{ij}$，第 i 条报文的特征词集合为 $feature_i$。首先，本文对该簇内所有的 $feature_i$ 做一次并操作，得到没有重复项的所有协议特征词的集合，记为 $\bigcup_{i=1}^{n} feature_i$。然后对该簇内的所有 $feature_i$ 进行交操作，得到 $\bigcap_{i=1}^{n} feature_i$。在 $\bigcap_{i=1}^{n} feature_i$ 集合中的协议特征词都为必选字段的特征词。如果数据不够充足，可能将可选字段错误地认为是必选字段。最后，对 $\bigcap_{i=1}^{n} feature_i$ 与 $\bigcup_{i=1}^{n} feature_i$ 做一次差操作，从而得到 $\bigcup_{i=1}^{n} feature_i-\bigcap_{i=1}^{n} feature_i$，其中的特征词为可选字段的特征词。

例如，有 3 条 HTTP 的报文的协议特征词，它们被分在同一个簇内：

〈“GET”，“HTTP/1.1”，“Host:”〉

〈“GET”，“HTTP/1.1”，“Accept:”〉

〈“GET”，“HTTP/1.1”，“User-Agent”〉

那么就有，$\bigcup_{i=1}^{n} feature_i$ =〈“GET”，“HTTP/1.1”，“Host:”，“User-Agent:”，“Accept:”〉，$\bigcap_{i=1}^{n} feature_i$ =〈“GET”，“HTTP/1.1”〉，$\bigcup_{i=1}^{n} feature_i-\bigcap_{i=1}^{n} feature_i$ =〈“Host:”，“User-Agent:”，“Accept:”〉。因此，$\bigcap_{i=1}^{n} feature_i$ 中的特征词为必选字段的特征词，而 $\bigcup_{i=1}^{n} feature_i-\bigcap_{i=1}^{n} feature_i$ 中的特征词为可选字段的特征词。

通过上述方法识别出必选字段特征词与可选字段特征词后，将每个特征词的必选或者可选属性在最后的报文格式中标记出来。在每个特征词后分别使用“M”标记必选字段，使用“O”标记可选字段，如“GET(M)__HTTP/1.1(M)__Host:(O)__User-Agent:(O)__Accept:(O)__”。

3 方法评估

本节将通过实验来验证基于粗糙及理论的聚类方法、特

征词提取方法以及报文格式推断方法的有效性。为了说明本文方法的有效性，我们将使用公开的协议作为研究对象，进行仿真模拟实验。我们选择 HTTP，FTP 和 SMTP 3 种基于文本的协议作为研究对象。HTTP 的数据来自 WRCCDC[16]，FTP 和 SMTP 的数据来自 DARPA[17]。

3.1 基于粗糙集理论的聚类评估

实验选取了召回率和准确率作为聚类方法的评价指标。

3.1.1 召回率

召回率也称查全率，是指在衡量聚类算法的有效性时被分在相同簇中的同一类别的样本数。召回率的公式为：

$$Recall=\frac{TP}{TP+FN} \tag{14}$$

其中，TP 指预测为正、实际为正的样本数量，FN 指预测为负、实际为正的样本数量。由于本文设计的聚类方法的结果中可能包含多种聚类结果，在进行实验测试时，将样本数量保留最多的聚类结果作为召回率的评价对象。

3.1.2 准确率

准确率在衡量聚类算法的有效性中是指正确地分到同一个簇中的样本数。准确率的公式为：

$$Precision=\frac{TP}{TP+FP} \tag{15}$$

其中，TP 的定义与召回率中的一样，FN 指预测为正、实际为负的样本数量。由于本文聚类后形成的结果中可能包含多个聚类结果，在进行实验测试时，将样本数量保留最多的聚类结果作为准确率的评价对象。

3.1.3 实验结果

Netzob[18]是由 PI 项目的作者编写的一个协议逆向开源软件，其认可程度高，极具代表性[19]。因此，在实验中，本文选取了 RSTC 聚类方法以及 Netzob 内嵌的基于相似度的聚类方法作为横向比较的对象。RSTC 聚类方法的参数以及 Netzob 的参数设置如表 4 所列，其中 *threshold_1* 的值是根据多次对比实验得出的最优值。

表 4 参数设置

Table 4 Parameter setting

RSTC			
type_threshold	*threshold_1*	*len_min*	*len_min*
1400	0.02	2	15
Netzob			
similarity		*particle size*	
0.4		8 bit	

在格式逆向时，应尽可能搜集更多的报文数量[20]，因此实验将每类型协议的前 2000 条报文作为实验样本。各类型协议的报文聚类召回率与准确率如图 3－图 8 所示。可以发现，相比 Netzob，本文提出的 RSTC 方法的召回率与准确率都更高，原因是 RSTC 方法是在分词的基础上，采用基于样本的位置属性对样本进行聚类；而 Netzob 是基于对比样本的相似度进行聚类的，当这种聚类方法应用到报文上时，往往采取序列比对算法，其不但耗时，而且只能提取出序列字面上的意思，并不能深入挖掘报文类型。如 HTTP 中的状态码“200”和状态码“400”表示完全不同的报文类型，但是序列比对算法却会认为这两个状态码只相差一个字节，它们的相似度非常高。RSTC 方法却避免了这种情况的发生，通过 N-邻接属性集和闭合候选特征词，将可能发生过分词的情况淘汰，状态码“200”和状态码“400”被作为一个整体参与到聚类算法中，从而提高了聚类算法的召回率与准确率。

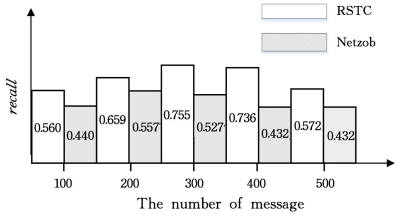

图 3 HTTP 的召回率结果

Fig. 3 Recall result of HTTP

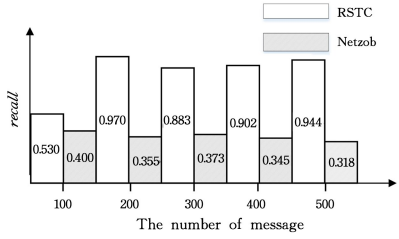

图 4 FTP 的召回率结果

Fig. 4 Recall result of FTP

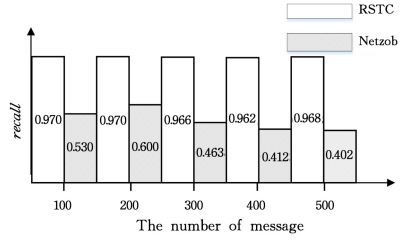

图 5 SMTP 的召回率结果

Fig. 5 Recall result of SMTP

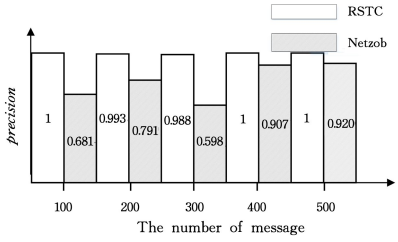

图 6 HTTP 的准确率结果

Fig. 6 Precision result of HTTP

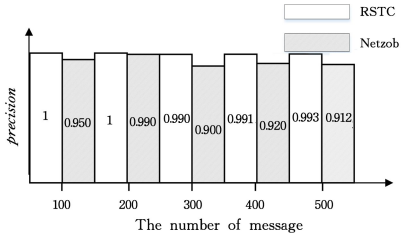

图 7 FTP 的准确率结果

Fig. 7 Precision result of FTP

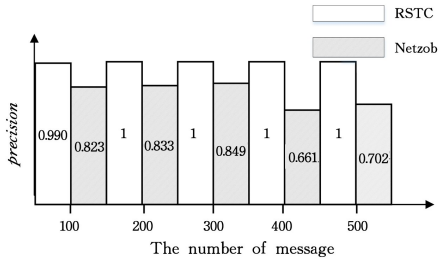

图 8 SMTP 的准确率结果

Fig. 8 Precision result of SMTP

此外，报文数量对 RSTC 的影响不大，而对 Netzob 的影响却很大。根据序列比对算法的特点，想要挖掘出报文字段的边界信息和关键的协议特征词信息，则需要更多的报文，而 RSTC 方法在聚类算法前就已经通过信息熵的方式对字段边界进行了划分，此时已经得到了大量关键的协议特征词信息。在聚类中产生的小簇和小聚类结果很有可能是在一个相同的分词位置上的值变化十分剧烈而导致的，因此删除这些小簇能够提升聚类的召回率与准确率。

3.2 协议特征词提取的评估

协议特征词提取在报文格式推断中是很重要的一部分，因为它是最终表示协议格式的基础。

本文用 Top-k 率来衡量协议特征词提取的有效性。Top-k 率指在排名前 k 的候选特征词中，真正的协议特征词占该协议本身拥有的所有特征词的比例。

Top-k 率的实验结果如图 9 所示。可以看出，对于 FTP 协议和 SMTP 协议，Top-k 率很快就到达了 1，这意味着排名靠前的候选特征词在真实的协议特征词中的占比很大；而对于 HTTP 来说，当 $k=60$ 时，Top-k 率到达 1。

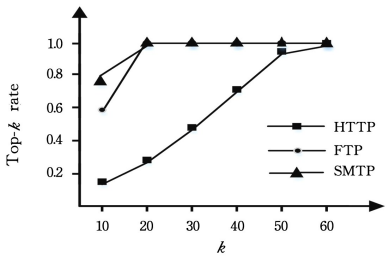

图 9 Top-k 率

Fig. 9 Top-k rate

3.3 报文格式推断的评估

报文格式是协议规范挖掘的基础。根据经验提取排序后的前 5%候选特征词作为协议特征词，对报文格式进行推断。

从表 5 可以看到，90%以上的报文格式都被推断出来，少量报文格式没有被推断出来，这主要是因为它们出现在数据集中的次数太少，或者本身在协议中的使用次数不多，属于较少使用的报文格式，推断不出来也不影响协议逆向结果的使用。

在提取得到的 FTP 和 SMTP 报文格式内，特征词都是必选的，在提取得到的 HTTP 报文格式内，既有可选特征词又有必选特征词。

表 5 报文格式推断的结果

Table 5 Result of message format inference

HTTP	GET(M)__HTTP/1.1(M) Connection:(O)__…… HTTP/1.1 200 ok(M)__
FTP	USER(M)__ 331(M)__ PASS(M)__ 230(M)__user logged in(M) SYST(M) 215(M)__ PORT(M)__ 200 PORT command successful(M) LIST(M) 150 Opening ASCII mode data connection for(M)__ 226 Transfer complete(M) CWD(M)__ 250 CWD command successful(M) TYPE(M)__ RETR(M)__ 221 Goodbye(M) NLST(M)
SMTP	220(M)__ HELO(M)__ 250(M)__ MAIL FROM(M)__ RCPT TO:(M)__ DATA(M) 354 Enter mail,end with "." On a line by itself(M) QUIT(M) 221 Closing connection(M) EHLO(M)__ 500 Command unrecognized(M)

结束语 本文提出了一种基于粗糙集理论的报文格式推断方法，该方法相比之前的工作具有更高的召回率、准确率和更低的时间耗费。

在预处理阶段，本文方法通过设置报文长度阈值，保留控制类报文进行分析，然后通过 VE 算法对控制类报文进行分割。在基于粗糙集理论的聚类阶段，通过分词结果中的位置属性，将分词结果进行聚类，然后对其进行简化，得到高准确率的聚类结果。在特征词提取阶段，使用位置、长度和支持度对每个候选特征词打分，然后提取出分值较高的候选特征词作为协议特征词。最后利用协议格式模型，将协议特征词匹配到协议格式模型中，并将特征词字段标识为必选和可选。

参 考 文 献

[1] WU L F, HONG Z, PAN F. Network Protocol Reverse Analysis and Application[M]// National Defense Industry Press. Beijing, China, 2016: 11-12.

[2] DUCHÊNE J, GUERNIC C L, ALATA E, et al. Protocol Reverse Engineering: Challenges and Obfuscation[C]// International Conference on Risks and Security of Internet and Systems. 2017.

[3] BEDDOE M. Protocol information project[EB/OL]. (2004-10-

05)[2019-06-25]. http://www.4tphi.net/~awalters/PI/PI.html.

[4] HE C, LIU F, ZENG X. Clustering Analysis of Unknown Protocol Message Sequence [J]. Communications Technology, 2017, 50(2): 277-286.

[5] LU Z Y, LI G S, SHEN Y Z, et al. Unknown protocol message clustering algorithm based on continuous features [J]. Journal of Shandong University (Natural Science), 2018, 54(5): 1-7.

[6] LI W M, ZHANG A F, LIU J C, et al. An Automatic Network Protocol Fuzz Testing and Vulnerability Discovering Method [J]. Chinese Journal of Computers, 2011, 34(2): 242-255.

[7] LI Y, LI Q, ZHAGN X. Outline Format Signature Construction Method Based on Separate Protocol Message [J]. Journal of Information Engineering University, 2018, 19(2): 134-139.

[8] YOUNG-HOON, GOO K S S, BYEONG-MIN CHAE, et al. Framework for precise protocol reverse engineering based on network traces [M]// 2018 IEEE/IFIP Network Operations and Management Symposium. 2018.

[9] BICHENG C, RENHUI L, YUNFEI Z, et al. Research on non-standard industrial control protocol formats reverse [J]. Computer Technology and Its Applications, 2018, 44(4): 126-129.

[10] YANG L, QING L, XIA Z. Automatic protocol format signature construction algorithm based on discrete series protocol message [J]. Journal of Computer Applications, 2017, 37(4): 954-969.

[11] ZHANG Z, ZHANG Z, LEE P P C, et al. Proword: An unsupervised approach to protocol feature word extraction[C]// IEEE Conference on Computer Communications. Toronto, Canada, 2014: 1393-1401.

[12] PAWLAK Z. Rough sets[J]. International Journal of Computer and Information Sciences, 1982, 11(5): 341-356.

[13] ZHANG H Z, HONG Z, WANG C, et al. Closed Sequential Patterns Mining Based Unknown Protocol Formal Inference Method[J]. Computer Science, 2019, 46(6): 80-89.

[14] X Z, DING S Y, LI O, et al. Keyword Sequence Extraction Based on Byte Entropy Iterative Segmentation [C]// presented at the 2017 3rd IEEE International Conference on Computer and Communications. Chengdu, China, 2017.

[15] KUROSE J F, ROSS K W. Computer Networking: A Top-Down Approach Featuring the Internet[M]. Addison-Wesley, 2002.

[16] WRCCDC Public Archive traces [DB/OL]. [2019-07-08]. https://archive.wrccdc.org/pcaps/2019/.

[17] MCHUGH J. Testing intrusion detection systems: a critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performedby lincoln laboratory[J]. ACM Transactions on Informationand System Security, 2000, 3(4): 262-294.

[18] BOSSERT G, HIET G, HENIN T. Modelling to simulate botnet command and control protocols for the valuation of network intrusion detection systems [C]// 2011 Conference on Network and Information System Security (SAR-SSI). La Rochelle: IEEE, 2011: 1-8.

[19] KLEBER S, MAILE L, KARGL F. Survey of Protocol Reverse Engineering Algorithms: Decomposition of Tools for Static Traffic Analysis [J]. IEEE Communications Surveys & Tuto-rials, 2019, 21(1): 526-561.

[20] NARAYAN J, SHUKLA S K, CLANCY T C. A Survey of Automatic Protocol Reverse Engineering Tools [J]. Acm Computing Surveys, 2015, 48(3): 1-26.

LI Yi-hao, born in 1996, postgraduate. His main research interests include cyberspace security and protocol reverse engineering.

HONG Zeng, born in 1979, Ph.D, associate professor. His main research interests include cyberspace security and protocol reverse engineering.