基于自编码器的未知协议分类方法

顾纯祥[1,2]，吴伟森[1]，石雅男[1]，李光松[1]
（1. 信息工程大学网络空间安全学院，河南 郑州 450001；2. 网络密码技术河南省重点实验室，河南 郑州 450001）

摘 要：针对互联网中存在的大量未知协议导致网络管理和维护网络安全十分困难的问题，提出了一种未知协议的分类识别方法。结合自编码器技术和改进的 K-means 聚类技术针对网络流量实现了未知协议的分类识别。利用自编码器对网络流量进行降维和特征提取，使用聚类技术对降维后数据进行无监督的分类，最终实现对网络流量的无监督识别分类。实验结果表明，所提方法分类效果优于传统的 K-means、DBSCAN、GMM 算法，且具有更高的效率。

关键词：未知协议分类；自编码器；无监督分类；特征提取
中图分类号：TP181
文献标识码：A
doi: 10.11959/j.issn.1000−436x.2020123

Method of unknown protocol classification based on autoencoder

GU Chunxiang[1,2], WU Weisen[1], SHI Ya'nan[1], LI Guangsong[1]
1. School of Cyberspace Security, Information Engineering University, Zhengzhou 450001, China
2.Henan Key Laboratory of Network Cryptography Technology, Zhengzhou 450001, China

Abstract: Aiming at the problem that a large number of unknown protocols exist in the Internet, which makes it very difficult to manage and maintain the network security, a classification and identification method of unknown protocols was proposed. Combined with the autoencoder technology and the improved K-means clustering technology, the unknown protocol was classified and identified for the network traffic. The autoencoder was used to reduce dimensionality and select features of network traffic, clustering technology was used to classify the dimensionality reduction data unsupervised, and finally unsupervised recognition and classification of network traffic were realized. Experimental results show that the classification effect is better than the traditional K-means, DBSCAN, GMM algorithm, and has higher efficiency.

Key words: unknown protocol classification, autoencoder, unsupervised classification, feature extraction

1 引言

近年来，随着互联网和通信技术的快速发展，网络应用和服务种类越来越多，网络流量增长异常迅猛，网络攻击和恶意服务不断滋生，维护网络安全变得十分重要。网络流量分类是进行网络管理和维护网络安全的基础，其目的是识别不同网络流量，实现合理分配网络资源，提升网络服务质量和网络安全。

为了提供多种网络服务，互联网中使用了各种各样的网络协议，除了大量标准化的网络协议外，还存在很多未知协议。不少网络通信软件厂商或个人出于经济利益、安全、隐私等方面的考虑没有公开协议细节。此外，一些恶意通信软件也采用了其研发的私有协议防止被跟踪和分析[1]。网络中大量恶意服务和攻击使用的是标准没有公开的未知协

收稿日期：2019−11−30；**修回日期**：2020−04−03
基金项目：国家自然科学基金资助项目（No.61772548）；国家自然科学基金创新研究群体资助项目（No.61521003）；信息保障技术重点实验室开放基金资助项目（No.KJ-17-001）
Foundation Items: The National Natural Science Foundation China(No.61772548), Innovative Research Groups of the National Natural Science Foundation of China(No.61521003), Foundation of Science and Technology on Information Assurance Laboratory(No.KJ-17-001)

议，为了识别该类恶意攻击和服务，在网络流量中识别未知协议是十分必要的工作。

流量识别主要分为基于端口的方法、基于深度分组检测的方法、基于概率的方法三大类。未知流量分类主要分为将非已知协议标记为未知协议的方法、根据已知协议信息进行聚类的半监督方法、直接聚类的无监督方法。

基于端口和深度分组检测的方法已经不适用于如今的网络环境。目前研究主要集中于机器学习方法和人工提取特征相结合[2]。人工提取特征在进行流量识别时面临的主要问题是需要大量的专业经验，而且在对未知协议进行人工特征提取时，由于缺乏足够的经验，很难提取出合适的特征。

自编码器是一种自监督模型，可以在强制降维过程中学习数据中的特征，并自动完成降维工作[3]。

本文针对协议特征提取困难的问题，利用自编码器自动提取协议有效载荷数据特征，结合经典无监督分类模型 K-means，提出了一个基于无监督的未知协议识别模型。该模型由一个编码器和一个分类器组成。分类器使用的分类数据是由编码器压缩的包含协议头的原始协议数据。压缩的数据既去除了冗余数据又保留了原始数据的主要特征，因此能够在快速分类的同时提升准确率。这种方法同原始的聚类算法相比能够很好地加快协议识别的速度，而且对分类的效果也有一定提升。

2　相关工作

2.1　流量识别

目前已有很多网络流量识别的研究，这些工作主要分为三类：基于端口的方法、基于深度分组检测的方法、基于概率的方法。

1) 基于端口的方法。基于端口的网络流量识别方法是最早出现的方法[4]，对传输控制协议/用户数据报协议（TCP/UDP, transmission control protocol/user datagram protocol）分组的端口号进行考察，并和互联网号码分配局（IANA, Internet Assigned Numbers Authority）给定的端口号进行比较来实现流量分类，通常应用于防火墙规则和访问控制列表[4]。但是随着动态端口技术[5]和端口伪装技术[6]的成熟，基于端口协议识别变得不再可靠。

2) 基于深度分组检测的方法。该方法通过收集网络流量数据分组中的内容，并对网络流量数据进行统计建模，对于不同的分类要求建立不同的分类方案和模型。Chung 等[7]提取了 Bittorrent 协议中的特征字，通过检测流量中的特征字匹配字段实现对 Bittorrent 的识别。Rocha 等[8]通过对捕获的网络流量的统计特性进行分析，实现流量的协议分类。

3) 基于机器学习和深度学习的方法。研究者通过结合机器学习的各种算法如随机森林、支持向量机、逻辑回归、k 近邻等，提取出网络流量的统计特征来实现对网络流量的分类识别[9]。Blake 等[10]通过提取流元数据、分组长度分布、时间分布、字节分布和未加密的安全传输层（TLS, transport layer security）协议分组头信息作为联合特征，使用 Logistic 回归算法识别恶意软件加密流量。Wang 等[11]使用原始数据流量经过剪切处理后作为卷积神经网络（CNN, convolutional neural network）的输入实现流量分类。Yang 等[12]使用卷积神经网络提取流量高维特征，使用自编码器提取流量的代表性特征，进行流量分类识别。

2.2　未知流量分类

有监督的未知流量方法是通过排除已知流量来找出未知流量。Ma 等[13]根据 10 种已知协议训练神经网络模型，将那些对任何已知协议都不匹配的数据认为是未知协议，虽然能够很好地识别未知协议，但是对未知协议内部没有分类。

半监督的未知流量分类方法通过部分已知协议的信息来指导未知协议分类。Zhang 等[14]根据已知协议的三元组为具有相同三元组的未知协议打上标签，再对训练集进行聚类，根据每一类中最多的标签决定该类的标签，如果无标签则为未知协议类。Zhu 等[15]根据已知协议，从协议流中获得特征，利用这些特征进行未知协议分类。该类方法依赖于选择的已知协议是否与未知协议特征相符合。

无监督的方法包括 K-means 聚类算法、基于密度的聚类（DBSCAN, density-based spatial clustering of applications with noise）算法、混合高斯模型（GMM, Gaussian mixed model）算法、EM（expectation-maximization）算法、Autoclass 算法等[16]。无监督方法可以根据 TCP 前 k 个分组的大小和方向作为输入来对应用类别进行分类[17]。卢政宇等[18]使用将最长相同字段作为距离的 K-means 聚类算法对协议进行分类。该类方法目前没有很好的特征提取方法。

3　基础知识

为了解决协议流量维度过高的问题，需要对初始的输入数据进行降维，提取特征后才能更好地进

行分类识别工作[19]。本文采用神经网络中的自编码器对数据进行自动降维处理[20-21]，使用 K-means 聚类方法对降维后的数据分类。本节将介绍自编码器和 K-means 聚类相关知识。

3.1 自编码器

自编码器是一种自监督的神经网络生成方法，主要用于数据去噪、特征提取和数据降维[22-23]。自编码器为了学习数据内部特征，限制了内部表示空间的维度，迫使模型必须学习一些低维的特征来对输入进行表示从而实现了数据降维和特征提取[24]。

自编码器分为编码器和解码器两部分。编码器将输入数据压缩到潜在空间中，可以用下面的函数关系来表示：$\boldsymbol{h}=f(\boldsymbol{x}),\boldsymbol{x}\in F^n,\boldsymbol{h}\in F^m,m<n$，$n$ 为输入编码器的数据维度，m 为编码器的输出数据维度，$\boldsymbol{x}$ 和 $\boldsymbol{h}$ 分别为编码器的输入数据和输出数据，F 为输入数据和输出数据包含元素的集合。解码器的目的是将潜在空间中的数据解码到原来的输入，可以用下面的函数关系来表示 $\boldsymbol{r}=g(\boldsymbol{h}),\boldsymbol{r}\in F^n,\boldsymbol{h}\in F^m$，$m<n$，$n$ 为解码器输出数据维度，m 为解码器输入数据维度，$\boldsymbol{h}$ 和 $\boldsymbol{r}$ 分别为解码器的输入数据和输出数据。因此，自编码器可以表示为 $\boldsymbol{r}=g(f(\boldsymbol{x})),\boldsymbol{r},\boldsymbol{x}\in F^n$,其目的是使输入 $\boldsymbol{x}$ 和输出 $\boldsymbol{r}$ 尽可能相近。该过程可以看作对输入数据的压缩编码，将高维的原始数据用低维的向量表示，使压缩后的低维向量保留输入数据的典型特征，从而能够较为方便地恢复原始数据。需要注意的是，这里增加了一个约束条件，即在对数据进行编码和解码时，使用的是同一个参数矩阵。该约束用于减少参数的个数，控制模型的复杂度。

编码器和解码器一般都是由神经网络构成。图 1 表示了一个简单的自编码器模型，其具有一个 10 维的输入层，一个 3 维隐藏层和一个 10 维的输出层，即 n=10，m=3，m<n。编码器是一个输入为 n 维、输出为 m 维的全连接层，解码器是一个输入为 m 维、输出为 n 维的全连接层。将 n 维的数据输入自编码器中，得到一个 n 维的输出向量，通过比较输入和输出向量之间的差别，能够对模型进行训练，由于隐藏层的维数小于输入层和输出层，自编码器只能通过学习输入数据的典型特征才能使解码得到的输出向量和输入数据相似，因此可以得到一个 m 维的典型特征向量（降维处理后的数据）。

3.2 K-means 聚类算法

聚类算法属于无监督学习，不需要预先知道样本中的数据分类，只需要知道样本数据就可以进行训练。其根据数据的相似性将一个数据集分割为不同的类或簇，使在一个簇中的数据对象相似性尽量大，在不同簇之间的数据对象差异性尽量大。

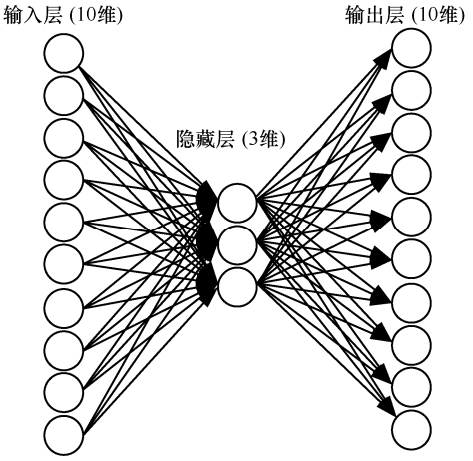

图 1 自编码器模型

聚类的一个关键问题是如何衡量数据之间的相似度，一般使用距离进行度量，如欧氏距离、Minkowoski 距离、曼哈顿距离，还可以使用相关系数来度量。本节介绍以欧氏距离为度量的 K-means 聚类[25]。

给定具有 n 个数据的样本集 $\boldsymbol{D}$，$\boldsymbol{D}=(\boldsymbol{x}_1,\boldsymbol{x}_2,\boldsymbol{x}_3,\cdots,\boldsymbol{x}_n)$，每个数据对象 $\boldsymbol{x}_i$ 为 d 维向量。K-means 聚类算法的目的是在给定 k 个分组数量的情况下，将 n 个数据对象分为 k 簇，即 $S=\{S_1,S_2,S_3,\cdots,S_k\}$，并使目标函数 E 最小。目标函数如式(1)所示。

$$E=\sum_{i=1}^{k}\sum_{\boldsymbol{x}\in S_i}\|\boldsymbol{x}-\boldsymbol{\mu}_i\|^2$$
$$\boldsymbol{\mu}_i=\frac{1}{|S_i|}\sum_{\boldsymbol{x}\in S_i}\boldsymbol{x} \tag{1}$$

其中，E 表示所有簇间点之间的距离平方差之和。E 最小，则可以认为得到一个簇内距离最小的分类。

K-means 聚类算法步骤如下。

步骤 1 从 $\boldsymbol{D}$ 中随机取 k 个元素，作为 k 个簇的中心。

步骤 2 分别计算其他元素到 k 个簇中心的距离，将这些元素归到与簇中心距离最小的簇中。

步骤 3 根据聚类结果，重新计算 k 个簇的中心点，计算方法为取簇中所有元素各自维度的算术平均数。

步骤 4　将 $\boldsymbol{D}$ 中全部元素按照新的中心重新聚类。

步骤 5　重复步骤 2～步骤 4，直到聚类结果不再变化。

步骤 6　输出结果。

K-means 聚类算法保证了每次重新计算中心后的 E 值都是最小的，这样一定可以得到 E 的极小值，从而保证了算法的收敛性[26]。虽然无法得到 E 的最小值，但是 K-means 聚类算法复杂度较低，在分组数 k 不是很大的时候，计算速度较其他聚类算法更快，而且 K-means 聚类算法会产生紧密度比较高的簇。K-means 聚类算法的初始分组数目 k 很难确定，对噪声和离群值比较敏感，而且初始中心点的选取也影响 K-means 聚类的好坏。

4　基于自编码器的网络流量无监督聚类方法

本文使用自编码器来实现数据的特征提取和降维，并使用了改进后的 K-means 聚类算法来进行协议无监督分类，最后采用基于自编码器的无监督聚类方法 DEC（deep embedded clustering）对网络协议进行分类识别，方法流程如图 2 所示。

4.1　流量数据预处理

网络协议流量数据包含链路层数据、TCP/UDP 头、应用层协议数据。本文进行协议识别主要是关注应用层协议头的部分数据，而不是关注载荷部分数据。因此只选择了包含协议头部的数据分组作为分类的流量数据。协议流量数据形式如图 3 所示，每一条数据都是一帧流量。

```
08 00 20 89 BA 28 00 C0 4F A3 57 DB 08 00 45 00 00 3F 20
4E 00 00 40 11 1E 96 AC 10 70 95 AC 10 73 14 05 C0 00 35
00 2B 34 78 B0 04 01 00 00 01 00 00 00 00 00 00 06 6C 61
6D 62 64 61 06 6F 72 61 6E 67 65 03 63 6F 6D 00 00 01 00
01
00 10 5A 9C B2 8E 08 00 20 89 BA 28 08 00 45 00 00 3F F5
7A 40 00 FF 11 A5 51 AC 10 73 14 C0 A8 01 14 80 0C 00 35
00 2B 6E FD 57 1C 00 00 00 01 00 00 00 00 00 00 06 6C 61
6D 62 64 61 06 6F 72 61 6E 67 65 03 63 6F 6D 00 00 01 00
01
```

图 3　协议流量数据形式

为了保证输入数据长度固定，需要将输入数据进行截断和填充。例如，选择长度 l 作为输入长度，需要对长度短于 l 的消息数据进行阶段处理，对其填充 0 使长度为 l。这样会导致数据的损失和噪声的产生。选择的长度需要在减少数据损失和噪声产生的基础上尽可能选取全部有效的控制信息。对于数据集 $\boldsymbol{I}=(\boldsymbol{m}_1,\boldsymbol{m}_2,\boldsymbol{m}_3,\cdots,\boldsymbol{m}_n)$，这里每个数据对象 $\boldsymbol{m}_i=(m_{i1},m_{i2},m_{i3},\cdots,m_{ij}\cdots,m_{i|\boldsymbol{m}_i|})$ 为不定长的消息，其中 m_{ij} 代表消息 $\boldsymbol{m}_i$ 的第 j 个字节。为了保证能够不损失有效信息并减少填充噪声，截取长度 l 设定为

$$l=\max\left(\arg\min\sum_{i=1}^{n}\left|\left|\boldsymbol{m}_i\right|-l\right|,t\right) \tag{2}$$

其中，t 是包括所有控制信息的最短长度。为了便于数值计算，需要将长度对齐后的数据集从十六进制转化为十进制。为了便于训练神经网络时进行矩阵运算，需要将向量进行归一化操作，如式(3)所示。

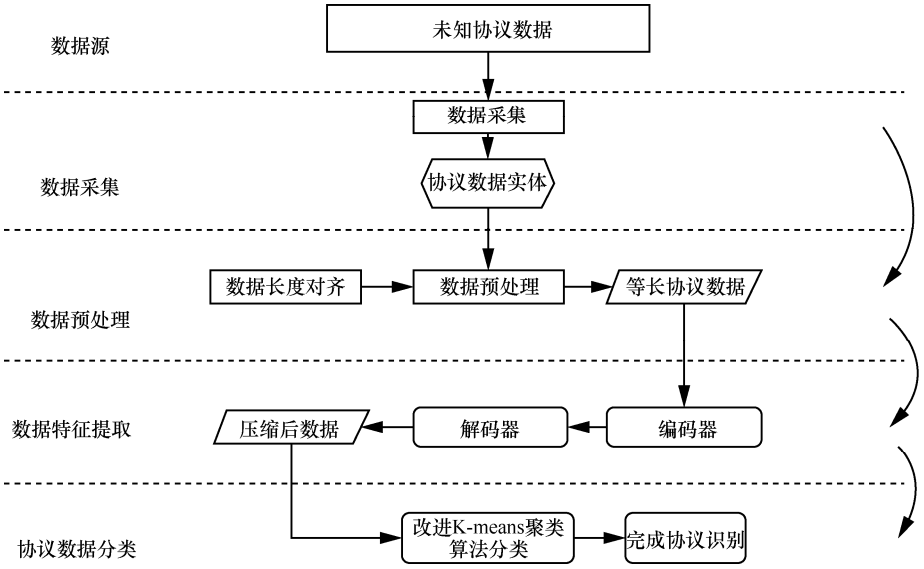

图 2　基于自编码器的无监督聚类方法流程

$$\boldsymbol{m}_i' = \frac{\boldsymbol{m}_i}{\max\limits_{1<i<n,1<j<l} m_{ij}} \tag{3}$$

4.2 DEC 模型

本文使用的 DEC 模型如图 4 所示，模型由一个编码器层和一个分类器层组成。算法过程由两步组成。第一步，将数据输入自编码器中进行训练，得到由编码器压缩过的数据；第二步，将压缩过的数据通过改进的 K-means 聚类算法进行分类。下面详细描述这两步的工作过程。

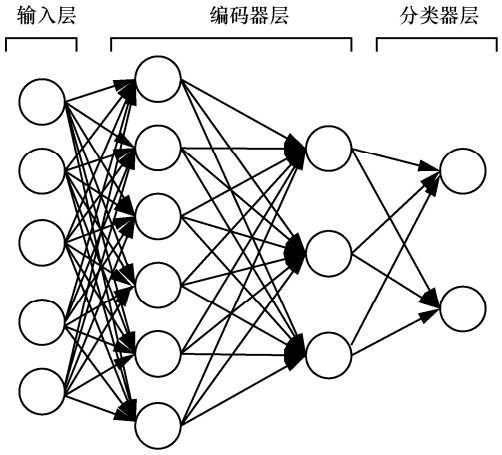

图 4 DEC 模型

第一步，将预处理后的数据进行特征提取。经过预处理后的流量数据为 $\boldsymbol{I}' = (\boldsymbol{m}_1', \boldsymbol{m}_2', \boldsymbol{m}_3', \cdots, \boldsymbol{m}_n')$，其中选择每个数据对象 $\boldsymbol{m}_i' = (m_{i1}', m_{i2}', m_{i3}', \cdots, m_{il}')$ 是长度为 l 且经过归一化的向量。在进行特征提取时，先将数据集输入自编码器中进行训练。训练后的自编码器中编码器可表示为一个输入为 l 维数据向量，输出为 m 维特征向量的压缩数据的函数，即 $\boldsymbol{h} = f(\boldsymbol{x}), \boldsymbol{x} \in F^l, \boldsymbol{h} \in F^m, m < l$。将数据集 $\boldsymbol{I}'$输入编码器 $f(\boldsymbol{x})$中得到压缩后的数据 $\boldsymbol{H} = (\boldsymbol{h}_1, \boldsymbol{h}_2, \boldsymbol{h}_3, \cdots, \boldsymbol{h}_n)$。

第二步，将压缩后的数据进行分类。DEC 模型选择使用以相对熵为度量的改进 K-means 聚类算法作为分类器对第一步中得到的压缩数据根据指定的分类数 k 进行分类。分类数 k 的选择方法在 4.3 节描述。下面详细介绍如何改进 K-means 聚类算法。

分类器比较辅助概率 $P(x)$和类型分配概率 $Q(x)$ 之间的差距，两者概率分布越相似，相对熵越低，$P(x)$和 $Q(x)$的相对熵为

$$\mathrm{KL}(P \mid Q) = \int P(x) \mathrm{lb} \frac{P(x)}{Q(x)} \mathrm{d}x \tag{4}$$

使用相对熵需要计算数据对象 $\boldsymbol{h}_i$ 被分类到簇 v_j 的概率 $Q(x)=(q_{ij})$，考虑到 t 分布是多个均值相同的高斯分布的叠加，能够很好地体现协议中类的分布，分类模型使用了 t 分布作为核来衡量数据对象 $\boldsymbol{h}_i$ 被分为簇 v_j 的概率[27]，如式(5)所示。

$$q_{ij} = \frac{\left(1 + \dfrac{\left\|\boldsymbol{h}_i - \boldsymbol{\mu}_j\right\|^2}{\alpha}\right)^{-\frac{\alpha+1}{2}}}{\sum\limits_{1 \leqslant j' \leqslant k} \left(1 + \dfrac{\left\|\boldsymbol{h}_i - \boldsymbol{\mu}_{j'}\right\|^2}{\alpha}\right)^{-\frac{\alpha+1}{2}}} \tag{5}$$

其中，$\boldsymbol{h}_i = f(\boldsymbol{m}_i)$ 是 $\boldsymbol{m}_i$ 经过编码器处理后的特征向量，$\boldsymbol{\mu}_j$ 是簇 v_j 的中心点，α 为 t 分布的自由度。由于无法在无监督下进行交叉验证，所以不需要特别考虑 α 的取值[28]，本文令 $\alpha = 1$。数据点距离一个簇的中心越近，其分配到该簇的概率越大。

使用相对熵还需要一个概率分布 $P(x)=(p_{ij})$作为辅助概率分布，并且 $P(x)$的选择影响分类器分类的效果。这里希望 $P(x)$具有以下特点：1) 提高分类的纯度；2) 更加强调具有高置信度的数据点；3) 将每个中心的损失贡献归一化，以防止大簇对隐藏特征空间的扭曲。为了满足以上的条件，在分类器模型中，使用 $Q(x)$在每次迭代中定义 $P(x)$为

$$p_{ij} = \frac{\dfrac{q_{ij}^2}{t_j}}{\sum\limits_{1 \leqslant j' \leqslant k} \dfrac{q_{ij'}^2}{t_{j'}}}, t_j = \sum_{1 \leqslant i \leqslant k} q_{ij} \tag{6}$$

通过将 q_{ij} 取平方，能够使对象 $\boldsymbol{h}_i$ 分类到簇 v_j 的概率 q_{ij} 在接近 0 时变小，减少了将消息误分的概率，并且提高了分配所需的阈值，增加了消息分类的置信度；然后，通过除以 t_j 来进行归一化。

最终得到改进 K-means 的目标函数为 $P(x)$和 $Q(x)$的相对熵，如式(7)所示。

$$L = \mathrm{KL}(P \mid Q) = \sum_i \sum_j p_{ij} \mathrm{lb}\left(\frac{p_{ij}}{q_{ij}}\right) \tag{7}$$

每次迭代都为数据点 $\boldsymbol{h}_i$ 重新分配簇 v，如式(8)所示。

$$v = \arg\max_{1 \leqslant j \leqslant k} \left\{q_{ij} \mid q_{ij} = P(\boldsymbol{h}_i \in v_j)\right\} \tag{8}$$

重新计算每个簇的中心，计算簇 v_j 的中心 $\boldsymbol{\mu}_j$，如式(9)所示。

$$\boldsymbol{\mu}_j = \arg\min_{\boldsymbol{h} \in v_j} \left\{\sum_{\boldsymbol{h}_i \in v_j} \left\|\boldsymbol{h}_i - \boldsymbol{h}\right\|^2\right\} \tag{9}$$

分类器通过不停迭代使目标函数 L 值越来越小，分类越来越具有准确性。最终得到的 k 类簇就是分类结果。

经过改进后的分类算法由于根据点分配到簇的概率来衡量，因此受初始中心点选取的影响小，具有稳健性。输入的数据是提取典型特征后的压缩数据，要比原始数据的维度小很多，因此计算速度也比直接输入高维数据的聚类方法快。

4.3 k 值选取

DEC 模型需要预先选择数据分类的簇数 k，由于对协议的相关信息一无所知，无法结合标签选择簇数 k，因此只能通过使用内部信息评价指标来对不同簇数 k 选择后的结果进行评价，从中选择效果最优的 k 值。本文使用了 S_Dbw 指标来评价[29]分类效果，从而确定合适的 k 值。

给定一个样本集 $\boldsymbol{D}=(\boldsymbol{x}_1,\boldsymbol{x}_2,\boldsymbol{x}_3,\cdots,\boldsymbol{x}_n)$，选择簇数为 k 的 S_Dbw 指标为

$$\mathrm{S\_Dbw}_k(\boldsymbol{D})=\mathrm{Dens\_bw}(k)+\mathrm{Scat}(k) \tag{10}$$

$$\mathrm{Dens\_bw}(k)=\frac{1}{k(k-1)}\sum_{i=1}^{k}\sum_{j=1,j\neq i}^{k}\frac{\mathrm{density}(v_{i,j})}{\max\left(\mathrm{density}\left(v_i\right),\mathrm{density}\left(v_j\right)\right)} \tag{11}$$

$$\mathrm{Scat}(k)=\frac{1}{k}\sum_{i=1}^{k}\frac{\left\|\delta\left(v_i\right)\right\|}{\left\|\delta\left(\boldsymbol{D}\right)\right\|} \tag{12}$$

其中，$\delta(\boldsymbol{D})$ 代表样本集的方差，$\mathrm{density}\left(v_i\right)=\sum_{i=1}^{|v_i|}r_{ij}$，如果点 $\boldsymbol{x}_i$ 距离簇 v_j 中心小于平均标准差，则 r_{ij} 的值为 1；否则 r_{ij} 的值为 0，$v_{i,j}$ 表示 v_i 和 v_j 的并集。Dens_bw(k)计算了簇间密度和簇内密度比例之和，因此簇间密度越小，簇内密度越大时，Dens_bw(k)越小。Scat(k)计算了簇内方差和整体方差比例之和，整体方差不会改变，每个簇内的方差越小，Scat(k)也越小。好的聚类应该簇间密度很小，簇内方差也很小，因此 S_Dbw 指标越小，聚类的效果越好。

对于数据集 $\boldsymbol{I}'=\left(\boldsymbol{m}_1',\boldsymbol{m}_2',\boldsymbol{m}_3',\cdots,\boldsymbol{m}_n'\right)$，经过压缩后得到 $\boldsymbol{H}=(\boldsymbol{h}_1,\boldsymbol{h}_2,\boldsymbol{h}_3,\cdots,\boldsymbol{h}_n)$ 输入到分类器中进行分类，最终进行 k 值选择时本文选择 $\mathrm{DS\_Dbw}(k)=\mathrm{S\_Dbw}_k(\boldsymbol{I}')+\mathrm{S\_Dbw}_k(\boldsymbol{H})$ 来作为评价指标，DS_Dbw 越小分类效果越好。

5 实验仿真

本节通过实验仿真对提出的 DEC 模型进行测试。下面先介绍实验数据和实验环境，之后展示基于 DEC 的流量分类结果，并与其他无监督学习方法进行比较。

5.1 实验环境和数据

实验平台是使用以 tensorflow 为后端的 keras 框架，Python 版本为 3.6，运行在 Windows10 操作系统上；实验硬件方面，CPU 为 inter-core I5 8th Gen，内存为 8 GB，GPU 为 NVIDIA GeForce GTX1060Ti。

实验数据选择 1998 年林肯实验室的公开流量数据集和 2012 年加拿大新不伦瑞克大学的信息安全中心发布的入侵检测数据集 ISCX2012。这 2 个数据集都是原始流量数据，相隔年份很大，数据端口和 IP（Internet Protocol）地址也不相同，可以有效测试分类模型的通用性。本文从这 2 个数据集中提取出 DNS（domain name system）、HTTP（hyper text transfer protocol）、NetBIOS（network basic input/output system）、SMTP（simple mail transfer protocol）、SSH（secure shell）、ARP（address resolution protocol）、TLS（transport layer security）共计 7 种协议构建了数据集 1 进行分类识别。其中 ARP、DNS、SSH、TLS 都有 10 000 条协议分组数据，HTTP 有 6 637 条协议分组数据，SMTP 有 4 714 条协议分组数据，NetBIOS 有 3 771 条协议分组数据。

由于不同协议处在不同的流之间，因此其交互双方的 IP 地址和端口号也与协议相关，为了防止 IP 地址和端口号对分类造成影响，本文选取了 DNS、HTTP、SMTP、SSH、TLS 共 5 种协议，并去掉前 34 B 的 IP 地址和 TCP/UDP 数据分组头部分，构建了数据集 2 来观察 IP 地址和端口号是否会对分类模型产生影响。为了确定模型的通用性，构建了去除 IP 地址和端口号的包含 DNS、HTTP、SSH、TLS 各 3 000 条协议分组的数据集 3 来进行对比实验。

5.2 评价指标

对于无监督分类，由于没有准确标签对分类效果进行评判，如何评价分类的好坏就成了一个关键性问题。

当模型参数确定后，需要进行外部信息评价聚类的真实效果。本文使用了调整兰德系数（ARI, adjusted Rand index）和纯度（purity）来评价聚类效果的好坏[30]。

调整兰德系数。给定具有 n 个数据的样本集 $\boldsymbol{D}=(\boldsymbol{x}_1,\boldsymbol{x}_2,\boldsymbol{x}_3,\cdots,\boldsymbol{x}_n)$，假设 $\boldsymbol{U}=(\boldsymbol{u}_1,\boldsymbol{u}_2,\cdots,\boldsymbol{u}_r)$ 和

$\boldsymbol{V}=(\boldsymbol{v}_1,\boldsymbol{v}_2,\cdots,\boldsymbol{v}_c)$ 表示 $\boldsymbol{D}$ 的 2 个不同划分并且满足 $\bigcup_{i=1}^{r}\boldsymbol{u}_i=\boldsymbol{D}=\bigcup_{i=1}^{c}\boldsymbol{v}_i$，$\boldsymbol{u}_i\cap\boldsymbol{u}_{i'}=\varnothing=\boldsymbol{v}_j\cap\boldsymbol{v}_{j'}$，$1\leqslant i\neq i'\leqslant r$，$1\leqslant j\neq j'\leqslant c$，$\varnothing$ 为空集。假设 $\boldsymbol{U}$ 是外部评价标准，而 $\boldsymbol{V}$ 是聚类后的分类结果。设定 4 个统计量。

a 为在 $\boldsymbol{U}$ 中为同一类，且在 $\boldsymbol{V}$ 中也为同一类的数据点对数。

b 为在 $\boldsymbol{U}$ 中为同一类，但在 $\boldsymbol{V}$ 中属于不同类的数据点对数。

c 为在 $\boldsymbol{U}$ 中不在同一类，但在 $\boldsymbol{V}$ 中为同一类的数据点对数。

d 为在 $\boldsymbol{U}$ 中不在同一类，且在 $\boldsymbol{V}$ 中也不属于同一类的数据点对数。

兰德系数 $\mathrm{RI}=\dfrac{a+d}{a+b+c+d}$，对于 2 个随机的划分，其 RI 不是一个接近于 0 的常数。为了解决这一问题提出了调整兰德系数，假设 n_{ij} 表示同时在类别 u_i 和簇 v_j 内的数据点数目，$n_{i.}$ 为类 u_i 包含的数据点数目，$n_{.j}$ 为簇 v_j 包含的数据点数目，则 $a+d=\sum_{i,j}\binom{n_{ij}}{2}$，调整兰德系数为

$$\mathrm{ARI}=\frac{\mathrm{RI}-\mathrm{E}(\mathrm{RI})}{\max(\mathrm{RI})-\mathrm{E}(\mathrm{RI})} \tag{13}$$

$$\mathrm{E}(\mathrm{RI})=\mathrm{E}\left(\sum_{i,j}\binom{n_{ij}}{2}\right) \tag{14}$$

$$\max(\mathrm{RI})=\frac{1}{2}\left[\sum_{i}\binom{n_{i.}}{2}+\sum_{j}\binom{n_{.j}}{2}\right] \tag{15}$$

ARI 能够评价 2 个分布之间的相似性，ARI 取值为[−1,1]，越接近 1，则聚类结果越符合真实分布[31]。

纯度。假设 k 是聚类的簇数，r 为真实类别数目，n 为样本个数，n_{ij} 表示同时在类别 u_i 和簇 v_j 内的数据点数目，则有

$$\mathrm{purity}=\sum_{i=1}^{k}\frac{\max\limits_{1\leqslant j\leqslant r}n_{ij}}{n} \tag{16}$$

purity 表示被成功分类的数据占总的数据的百分比。

5.3 结果分析

本文对 k=3, 4,…, 18 进行实验来选择合适的 k 值。图 5 为数据集 1 的 S_Dbw 和 DS_Dbw 随 k 的变化情况。

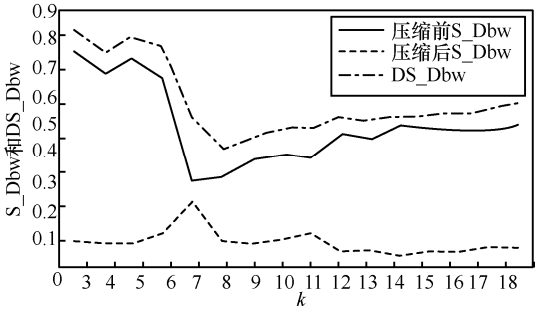

图 5 数据集 1 压缩前后 S_Dbw 和 DS_Dbw 随 k 的变化

图 5 中压缩前数据是数据集 1 的预处理后的数据，压缩后数据是数据集 1 经过编码器编码后的数据，最后根据 DS_Dbw 值确定 k 值为 8。发现压缩前数据的 S_Dbw 值普遍高于压缩后的 S_Dbw 值，这是由于压缩后的数据是压缩前的数据投影到低维空间的结果，因此压缩后的数据距离也要普遍小于压缩前的数据，导致 Dens_bw 值和 Scat 都相应变大。

图 6 为数据集 2 的 S_Dbw 和 DS_Dbw 随 k 的变化情况。其中，压缩前数据是数据集 2 的预处理后的数据，压缩后数据是数据集 2 经过编码器编码后的数据，最后根据 DS_Dbw 值，最终确定 k 值为 6。图 7 为数据集 3 的 S_Dbw 和 DS_Dbw 随 k 的变化情况，在数据集 3 中根据 DS_Dbw 值选择 k 值为 6。根据提供的数据，数据集 1、数据集 2 和数据集 3 期望选择的 k 值应该分别为 7、5 和 4，和最终选择的 8、6 和 6 有所差别。

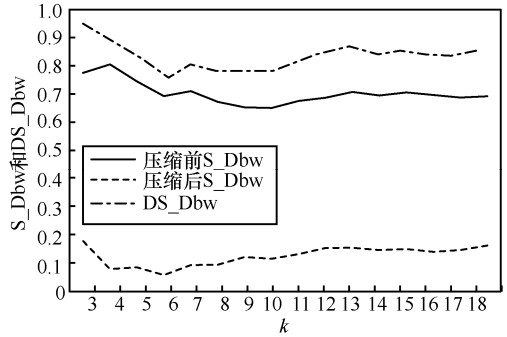

图 6 数据集 2 压缩前后 S_Dbw 和 DS_Dbw 随 k 的变化

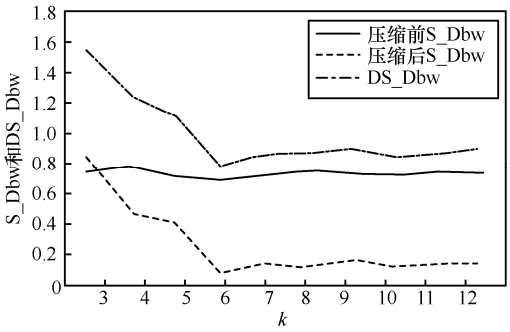

图 7 数据集 3 压缩前后 S_Dbw 和 DS_Dbw 随 k 的变化

图 8 是 DEC 模型在数据集 1 上分类结果的可视化矩阵。横轴每个格都为一个模型分出的簇，纵轴每个格都为一个为真实的类，矩阵元素为属于同一类并被模型分到同一簇的数据点个数。可以发现 DNS 协议被分为两类，一类是 DNS 的询问，另一类是 DNS 的回答，其他协议被聚类算法各自分为一类。这个分类结果验证了 k 值选取的有效性和可行性。

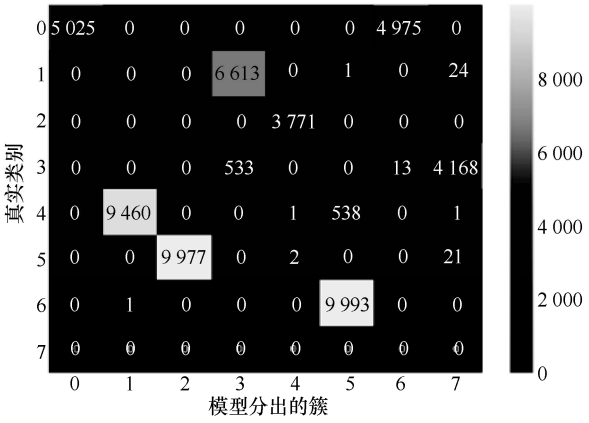

图 8　数据集 1 DEC 模型分类结果

图 9 和图 10 分别是 DEC 模型数据集 2 和数据集 3 上分类结果的可视化矩阵。数据集 2 中 DNS 中并没有被分为两类。聚类算法将 SMTP 和 SSL 中的一部分产生了混淆，而 SSL 这一部分在数据集 1 中和 TLS 被分为一类。这是由于 SSL 数据中特征不同且数量小的 SSL 数据和剩余 SSL 数据的距离远远大于与 SMTP 和 TLS 数据的距离，而数据数量差别又很大，其中一部分只占了 SSL 总数据的 5%，因此在考虑分类种类时没有特别地将其分为一类，最终导致这一小部分的 SSL 数据和其他数据混杂在一起。在数据集 3 中模型将 DNS 协议分为两类，并且其他协议也被各自分为一类，证明了模型在不同的环境和数据下也具有良好的分类效果。

5.4　与其他无监督分类方法比较

本文通过与部分已有的经典无监督分类模型 K-means、DBSCAN、GMM，还有 PRISMA 中的协议聚类方法进行比较来评价聚类的效果。PRISMA 采用非负矩阵分解（NMF，nonnegative matrix factorization）来获得协议特征。评价标准是纯度和 ARI，其他模型的簇数和 DEC 模型 k 值相同。

图 11～图 13 分别展示了不同的聚类方法对于数据集 1、数据集 2 和数据集 3 分类结果的纯度和 ARI 的值。可以看到 DBSCAN 方法的分类效果很差，这是由于流量数据的维度过高，而 DBSCAN 方法在高维空间的表现不佳，导致分类结果很差。对于数据集 1～数据集 3，DEC 分类效果好于 GMM、K-means 和 PRISMA。数据集 2 虽然整体上分类效果要弱于数据集 1 上的分类效果，但 DEC 分类结果的纯度和 ARI 仍好于 GMM，K-means 和 PRISMA。并且 PRISMA 效果在无地址的情况下优于 K-means。实验结果证明了获得协议特征能够分辨除了 IP 地址外的数据特征，准确地对协议进行分类。在数据集 3 中 PRISMA 的 ARI 要高于 DEC，但是纯度低于 DEC，说明 PRISMA 更适合处理小型数据。

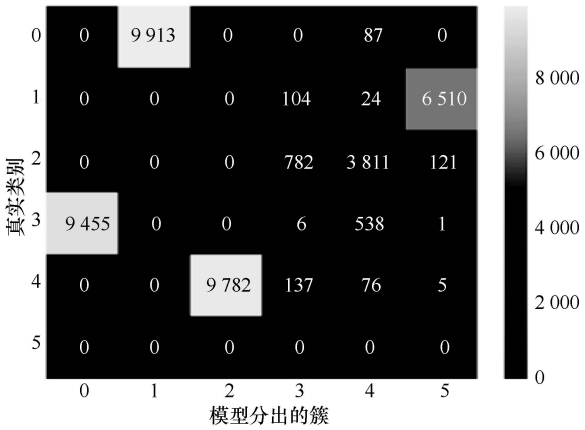

图 9　数据集 2 DEC 模型分类结果

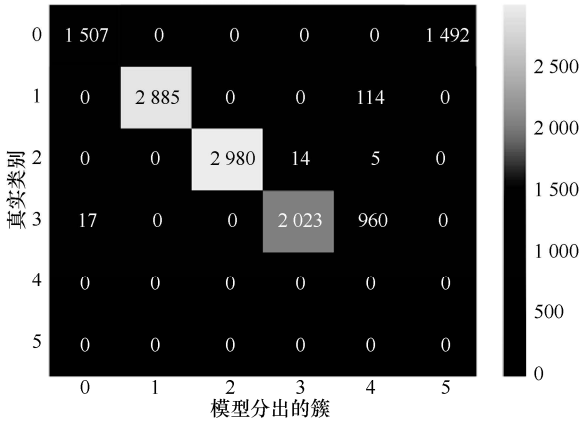

图 10　数据集 3 DEC 模型分类结果

表 1 展示了在 3 个数据集中不同方法分类的运行时间。由于对类似的环境自编码器不需要重新训练，因此在 DEC 模型已经提前训练好自编码器的情况下，只需测试进行分类迭代时所需要的时间。数据表明 DEC 明显快于其他方法。

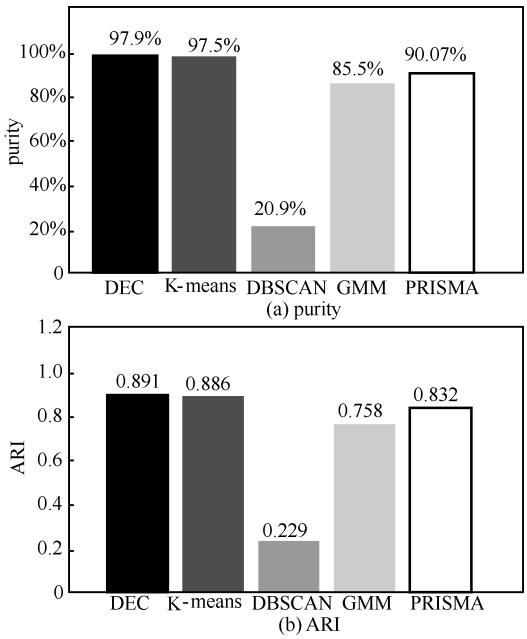

图 11 在数据集 1 上不同聚类方法效果 purity 和 ARI 值

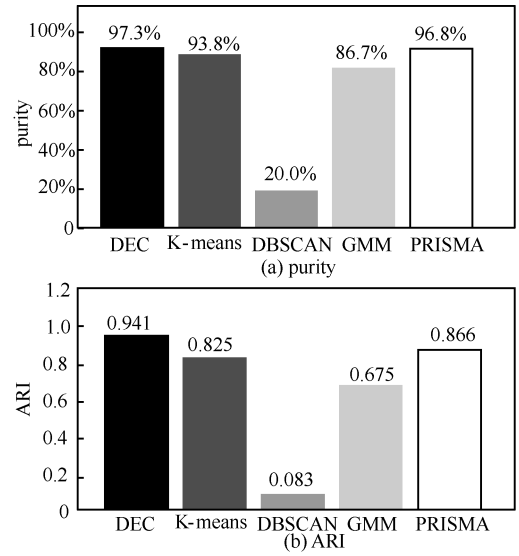

图 12 在不含 IP 地址的数据集 2 上不同聚类方法效果 purity 和 ARI 值

6 结束语

针对未知流量分类问题，本文提出一种基于自编码器的无监督聚类方法的网络协议分类识别模型。DEC 模型的优点在于其对输入的初始变量不敏感具有稳健性。DEC 模型能够对数据进行降维后聚类，实验测试表明自编码器训练完成后聚类的速度要快于 K-means、DBSCAN、GMM、PRISMA 这 4 种方法。而且对于不同环境下的网络数据，只要将新的数据输入自编码器中重新训练就可以完成模型的迁移，不需要再用人工提取特征来适应新的数据集。

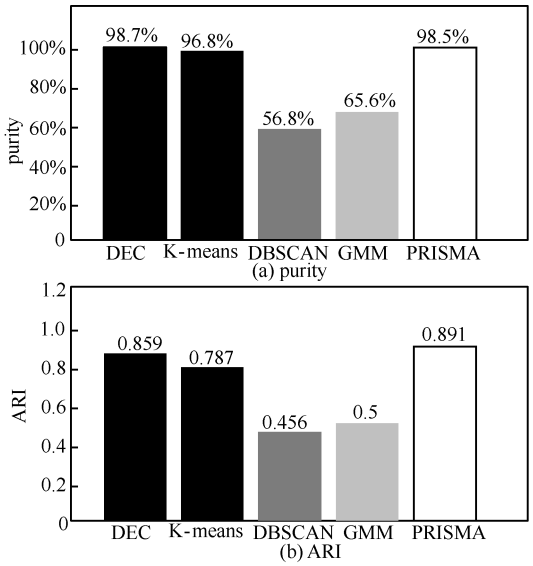

图 13 在数据集 3 上不同聚类方法效果纯度和 ARI 值

本文中 DEC 模型的自编码器结构仍然比较粗糙，提取的特征不够精简。下一步可以考虑寻找出更好的自编码器结构来提取数据特征。另外，现有的网络数据越来越多地使用加密算法进行保护，而本文的工作只适用于明文协议的分类，对于加密未知协议的识别，可以综合利用协议的前几条握手信息和协议流数据的特征来进行分类。

参考文献：

[1] 吴礼发，洪征，潘瑶. 网络协议逆向分析及应用[M]. 北京: 国防工业出版社,2016.
WU L F, HONG Z, PAN Y. Network protocol reverse analysis and application [M]. Beijing: National Defense Industry Press, 2016.

[2] ANDERSON B, MCGREW D.Machine learning for encrypted malware traffic classification: accounting for noisy labels and non-stationarity[C]//Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New

表 1 不同方法运行时间

数据集	DEC 模型/s	K-means/s	DBSCAN/s	GMM/s	PRISMA/s
数据集 1	5.662 36	7.325 22	180.059 59	16.448 28	5.876 94
数据集 2	4.182 25	4.601 08	27.051 56	8.413 99	4.536 88
数据集 3	0.682 61	2.132 34	3.968 55	0.934 93	2.125 45

York: ACM Press, 2017: 1723-1732.
[3] HINTON G, SALAKHUTDINOV R. Reducing the dimensionality of data with neural networks[J]. Science, 2006, 313(5786):504-507.
[4] QI Y, XU L, YANG B, et al. Packet classification algorithms: from theory to practice[J]. Proceedings - IEEE INFOCOM, 2009, 13(10):648-656.
[5] FIVOS C, PANAYIOTIS M. Identifying known and unknown peer-to-peer traffic[C]// Proceedings of IEEE International Symposium on Network Computing & Applications. Piscataway: IEEE Press, 2006: 93-102.
[6] THAY C, VISOOTTIVISETH V, MONGKOLLUKSAMEE S.P2P traffic classification for residential network[C]// Computer Science & Engineering Conference. Piscataway: IEEE Press, 2016: 1-6.
[7] CHUNG J, PARK B, WON Y, et al. Traffic classification based on flow similarity[C]// IEEE International Workshop on IP Operations & Management. Berlin: Springer, 2009: 65-77.
[8] ROCHA E, SALVADOR P, NOGUEIRA A. Detection of illicit network activities based on multivariate Gaussian fitting of multi-scale traffic characteristics[C]//2011 IEEE International Conference on Communications. Piscataway: IEEE Press, 2011:1-6.
[9] TAYLOR V,SPOLAOR R, CONTI M, et al. Robust smartphone App identification via encrypted network traffic analysis[J]. IEEE Transactions on Information Forensics & Security, 2017, 13(1):63-78.
[10] BLAKE A, SUBHARTHI P, DAVID M. Deciphering malware's use of TLS (without decryption)[J]. arXiv Preprint, arXiv: 1607.01639, 2017.
[11] WANG, W, ZHU M, ZENG X, et al. Malware traffic classification using convolutional neural network for representation learning[C]// 2017 International Conference on Information Networking. Piscataway: IEEE Press, 2017: 712-717.
[12] YANG Y, KANG C, GOU G, et al. TLS/SSL encrypted traffic classification with autoencoder and convolutional neural network[C]// 2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS). Piscataway: IEEE Press, 2018: 362-369.
[13] MA R, QIN S. Identification of unknown protocol traffic based on deep learning[C]// 2017 3rd IEEE International Conference on Computer and Communications. Piscataway: IEEE Press, 2017: 1195-1198.
[14] ZHANG J, CHEN C, XIANG Y, et al. An effective network traffic classification method with unknown flow detection[J]. IEEE Transactions on Network and Service Management, 2013, 10(2):133-147.
[15] ZHU P, ZHANG S, LUO H, et al. A semi-supervised method for classifying unknown protocols[C]// 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference. Piscataway: IEEE Press, 2019: 1246-1250.
[16] ZANDER S, NGUYEN T, ARMITAGE G. Automated traffic classification and application identification using machine learning[C]// IEEE Conference on Local Computer Networks. Piscataway: IEEE Press, 2005:250- 257.
[17] ERMAN J, ARLITT M, MAHANTI A. Traffic classification clustering algorithms[C]// Proceedings of SIGMETRICS. New York:ACM Press, 2006: 281-286.
[18] 卢政宇, 李光松, 申莹珠, 等. 基于连续特征的未知协议消息聚类算法[J]. 山东大学学报(理学版), 2019, 54(5): 37-43.
LU Z Y, LI G S, SHEN Y Z, et al. Clustering algorithm of unknown protocol messages based on continuous features [J]. Journal of Shandong University (Science Edition), 2019, 54 (5): 37-43.
[19] DING C, HE X. Cluster structure of K-means clustering via principal component analysis[J]. Lecture Notes in Computer Science, 2004, 46(4):414-418.
[20] CHEN X, KINGMA D, SALIMANS T, et al. Variational lossy autoencoder[J]. arXiv preprint arXiv:1611.02731, 2016.
[21] DENG J, ZHANG Z, EYBEN F, et al. Autoencoder-based unsupervised domain adaptation for speech emotion recognition[J]. IEEE Signal Processing Letters, 2014, 21(9):1068-1072.
[22] BENGIO Y, LAMBLIN P, POPOVICI D, et al. Greedy layer-wise training of deep networks[C]//Neural Information Processing Systems. Massachusetts: MIT Press, 2007: 153-160.
[23] VINCENT P, LAROCHELLE H, BENGIO Y, et al. Extracting and composing robust features with denoising autoencoders[C]// Machine Learning, Proceedings of the Twenty-Fifth International Conference. New York: ACM Press, 2008: 1096-1103.
[24] RIFAI S, VINCENT P, MULLER X, et al. Contractive auto-encoders: explicit invariance during feature extraction[C]// Proceedings of the 28th International Conference on Machine Learning. New York: ACM Press, 2011: 833-840.
[25] HARTIGAN J, WONG M. Algorithm AS 136: a K-means clustering algorithm[J]. Journal of the Royal Statistical Society. Series C (Applied Statistics), 1979, 28(1):100-108.
[26] SELIM S, ISMAIL M.K-means-type algorithms: a generalized convergence theorem and characterization of local optimality[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1984, 6(1):81-87.
[27] LAURENS V, HINTON G. Visualizing data using T-SNE[J]. Journal of Machine Learning Research, 2008, 9(2605):2579-2605.
[28] MAATEN L. Learning a Parametric embedding by preserving local structure[J]. Journal of Machine Learning Research, 2009(5):384-391.
[29] HALKIDI M, VAZIRGIANNIS M. Clustering validity assessment: finding the optimal partitioning of a data set[C]// IEEE International Conference on Data Mining. Piscataway: IEEE Press, 2001: 187.
[30] LIU Y, LI Z, XIONG H, et al. Understanding of internal clustering validation measures[C]// 2010 IEEE International Conference on Data Mining. Piscataway: IEEE Press, 2010: 911-916.
[31] HUBERT L, ARABIE P. Comparing partitions[J]. Journal of Classification, 1985, 2(1):193-218.

[作者简介]

顾纯祥（1976– ），男，安徽霍山人，博士，信息工程大学教授、博士生导师，网络密码技术河南省重点实验室主任，主要研究方向为密码学与网络安全。

吴伟森（1996– ），男，浙江天台人，信息工程大学硕士生，主要研究方向为网络安全、机器学习。

石雅男（1982– ），女，河南安阳人，信息工程大学讲师，主要研究方向为安全协议分析。

李光松（1977– ），男，山东德州人，博士，信息工程大学副教授，主要研究方向为网络协议分析、区块链、无线网络安全。