Journal of System Simulation

Volume 33 | Issue 3 Article 17

3-18-2021

Network Traffic Anomaly Detection Method for Imbalanced Data

Shuqin Dong
1. SSF Information Engineering University, Zhengzhou 450001, China; ;2. Henan Key Laboratory of Information Security, Zhengzhou 450001, China;

Bin Zhang
1. SSF Information Engineering University, Zhengzhou 450001, China; ;2. Henan Key Laboratory of Information Security, Zhengzhou 450001, China;

Network Traffic Anomaly Detection Method for Imbalanced Data

Abstract

Abstract: Aiming at the poor detection performances caused by the low feature extraction accuracy of rare traffic attacks from scarce samples, a network traffic anomaly detection method for imbalanced data is proposed. *A traffic anomaly detection model is designed, in which the traffic features in different feature spaces are learned by alternating activation functions, architectures, corrupted rates and dropout rates of stacked denoising autoencoder (SDA), and the low accuracy in extracting features of rare traffic attacks in a single space is solved. A batch normalization algorithm is designed, and the Adam algorithm is adopted to train parameters of SDAs to extract multifarious traffic features. The Softmax classifier is trained by combining the extracted features, so that the rare traffic attacks can be detected with a high detection precision.* The experimental results show that, compared with the methods based on random forest, single SDA and feature fusion, the proposed method has higher classification accuracy, higher detection rate of rare traffic attacks and the detection performances are stable.

Keywords

anomaly detection, imbalanced traffic classification, deep learning, stacked denoising autoencoder

Recommended Citation

Dong Shuqin, Zhang Bin. Network Traffic Anomaly Detection Method for Imbalanced Data[J]. Journal of System Simulation, 2021, 33(3): 679-689.

第 33 卷第 3 期 系统仿真学报© Vol. 33 No. 3
2021 年 3 月 Journal of System Simulation Mar. 2021

面向不平衡数据的网络流量异常检测方法

董书琴[1,2]，张斌[1,2]
（1. 战略支援部队信息工程大学，河南 郑州 450001；2. 河南省信息安全重点实验室，河南 郑州 450001）

摘要：针对小流量攻击样本稀少导致特征提取准确性低进而影响检测性能的问题，提出一种面向不平衡数据的网络流量异常检测方法。*设计流量异常检测模型：变换堆叠降噪自编码器(Stacked Denoising Autoencoder，SDA)激活函数、结构、噪声比例及 dropout 率，学习不同特征空间流量特征，解决单一空间小流量攻击特征提取准确性低的问题；设计批标准化算法，采用 Adam 算法训练 SDA 参数，提取多样性流量特征；联合所提特征对 Softmax 进行训练，提高小流量攻击检测精度。*实验结果表明：相比随机森林、单 SDA 和现有特征融合方法，所提方法分类准确率和小流量攻击检测率较高，且检测性能稳定。

关键词：异常检测；不平衡流量分类；深度学习；堆叠降噪自编码器
中图分类号：TP393; TP391 文献标志码：A 文章编号：1004-731X (2021) 03-0679-11
DOI: 10.16182/j.issn1004731x.joss.19-0573

Network Traffic Anomaly Detection Method for Imbalanced Data

Dong Shuqin[1,2], *Zhang Bin*[1,2]
(1. SSF Information Engineering University, Zhengzhou 450001, China; 2. Henan Key Laboratory of Information Security, Zhengzhou 450001, China)

Abstract: Aiming at the poor detection performances caused by the low feature extraction accuracy of rare traffic attacks from scarce samples, a network traffic anomaly detection method for imbalanced data is proposed. *A traffic anomaly detection model is designed, in which the traffic features in different feature spaces are learned by alternating activation functions, architectures, corrupted rates and dropout rates of stacked denoising autoencoder (SDA), and the low accuracy in extracting features of rare traffic attacks in a single space is solved. A batch normalization algorithm is designed, and the Adam algorithm is adopted to train parameters of SDAs to extract multifarious traffic features. The Softmax classifier is trained by combining the extracted features, so that the rare traffic attacks can be detected with a high detection precision.* The experimental results show that, compared with the methods based on random forest, single SDA and feature fusion, the proposed method has higher classification accuracy, higher detection rate of rare traffic attacks and the detection performances are stable.

Keywords: anomaly detection; imbalanced traffic classification; deep learning; stacked denoising autoencoder

引言

随着新型网络应用的不断涌现，网络流量类型愈发繁多，且不同类型流量在数量分布上呈现较大的不平衡性，正常业务流量较多，而恶意攻击流量较少[1]，特别是 R2L(remote to local)，U2R(user to root)等危害性较大的小流量攻击通常隐藏于大量

收稿日期：2019-11-01 修回日期：2020-01-17
基金项目：河南省基础与前沿技术研究计划(142300413201)，信息工程大学新兴科研方向培育基金(2016604703)，信息工程大学科研团队发展基金(2019f3303)
第一作者：董书琴(1990-)，男，博士，讲师，研究方向为网络安全态势感知。E-mail：dongshuqin377@126.com

的正常业务流量中，给流量异常检测带来了巨大的挑战。如何准确地从大量的正常业务流量中检测出网络攻击，特别是小流量攻击，是当前网络流量异常检测领域亟需解决的重点问题。

不平衡数据分类方法可通过对不平衡数据的重抽样或对传统分类算法的改进，提高少数类数据分类的准确性，为解决小流量攻击检测问题提供了一种有效方法。常用的不平衡数据分类方法主要可分为数据层和算法层两个方面。数据层包含欠采样、过采样和混合采样等方法[2]，该类方法可通过增加小流量攻击样本或减少大流量数据样本得到准平衡的数据集，然后采用传统分类算法对攻击流量进行检测，从而有效提高小流量攻击的检测性能，但会改变原始数据分布，容易丢失大流量数据中的重要信息，或引入大量相似性较大的小流量攻击样本导致过拟合问题。算法层方法可在不改变数据分布的前提下，通过增加对少数类数据的关注，实现不平衡数据的分类，从而有效提高小流量攻击的检测性能，其中具有代表性的方法主要有集成学习法和特征选择法等，集成学习法[3-4]可通过集成多个具有一定差异性的基分类器，并对不同分类器的输出结果进行融合，从而有效提高小流量攻击的检测率，但该类方法在训练检测模型时需要大量的标签数据，无法适应不断出现的新型攻击检测需求；特征选择法[5]可通过在现有流量特征中选择与小流量攻击相关的特征，提高小流量攻击检测性能，但在特征选择过程中需要不断对选择的特征子集进行验证分析，同样需要大量标签数据进行训练。

针对上述问题，将深度学习方法引入流量异常检测领域[6]，通过对大量无标签流量数据的逐层学习，可提取具有较高准确性的流量特征，进而提高攻击流量的总体检测性能，其中堆叠降噪自编码器(Stacked Denoising Autoencoder，SDA)[7]具有流量特征提取准确性高、鲁棒性好的特点，成为当前研究的热点。然而，在不平衡流量环境下，由于训练过程中小流量攻击样本稀少，导致其特征无法在单一特征空间中得到充分表达，在一定程度上降低了单个SDA提取小流量攻击特征的准确性，通过集成多个激活函数不同的自编码器可有效拓展小流量攻击样本的特征空间，从而提高其特征提取的准确性[8]。为此，本文从 SDA 出发，提出一种不平衡流量环境下基于多重特征学习的流量异常检测方法(Traffic Anomaly Detection Method Based on Multi Features Learning，TADMFL)，主要工作如下：

(1) 设计一种基于多重特征学习的流量异常检测模型。采用 3 个激活函数不同的 SDA，在不同特征空间中对网络流量特征进行学习，并通过变换 SDA 结构和训练噪声比例提高提取流量特征的准确性及鲁棒性；然后，基于多重流量特征对 Softmax 分类器进行训练，并通过变换 dropout 率强化 Softmax 的分类能力，实现对不平衡流量环境下攻击流量的检测，解决单个 SDA 对小流量攻击特征提取准确性低，导致其检测性能差的问题。

(2) 提出一种基于 SDA 的流量特征提取方法。采用小批量 Adam[9]算法对 SDA 进行逐层无监督训练，并在设计批标准化算法对 SDA1 和 SDA2 的输入样本特征进行标准化处理的基础上，给出采用不同激活函数时降噪自编码器(Denoising Autoencoder，DA)的权重和偏置参数更新规则，进而实现对流量特征的提取，同时缓解梯度消失导致的流量特征提取准确性低的问题。

(3) 提出一种基于 Softmax 的流量异常检测方法。在采用 3 个 SDA 提取少量有标签验证样本流量特征的基础上，基于联合特征对 Softmax 分类器进行训练，并在训练过程中引入 dropout 技术对 SDA 进行处理，有效提高分类器的泛化能力，避免过拟合情况导致的分类器检测性能差的问题。基于 NSL-KDD[10]数据集的实验结果表明：相比基于随机森林及单个 SDA 的流量异常检测方法，TADMFL 方法可有效提高小流量攻击的检测性能，且检测性能稳定。

DOI: 10.16182/j.issn1004731x.joss.19-0573

1 基于多重特征学习的流量异常检测模型

基于多重特征学习的流量异常检测模型如图 1 所示，主要包含数据预处理、流量特征提取和流量异常检测等 3 个组件。

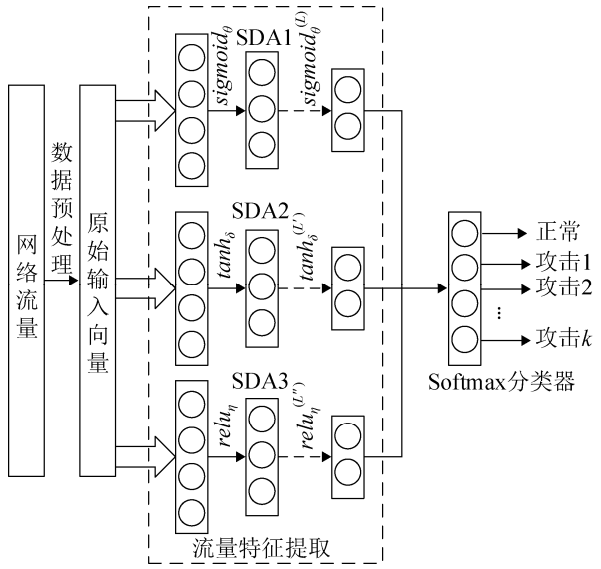


图 1 基于多重特征学习的流量异常检测模型
Fig. 1 Traffic anomaly detection model based on multi features learning

数据预处理组件主要完成对网络流量属性的数值化、归一化处理等。

流量特征提取组件通过构建 3 个结构、激活函数和训练噪声比例各不相同的 SDA，在多个特征空间对网络流量特征进行提取，其中 SDA1 含有 $L(L\in\mathrm{N}$，且 $L\geqslant 1)$个隐藏层，相邻两层之间的激活函数为 $sigmoid(x)=1/(1+\mathrm{e}^{-x})$，对于第 $l(l\in\mathrm{N}$，且 $1\leqslant l\leqslant L)$ 个隐藏层而言，其编码参数 $\theta^{(l)}=\{\boldsymbol{W}^{s(l)},\boldsymbol{b}^{s(l)}\}$，$\boldsymbol{W}^{s(l)},\boldsymbol{b}^{s(l)}$ 分别为编码权重矩阵和偏置向量，参数训练过程中添加的噪声比例为 ρ_1；SDA2 含有 L'（$L'\in\mathrm{N}$，且 $L'\geqslant 1$)层，相邻两层之间的激活函数为 $tanh(x)=(\mathrm{e}^{x}-\mathrm{e}^{-x})/(\mathrm{e}^{x}+\mathrm{e}^{-x})$，对于其第 l'（$l'\in\mathrm{N}$，且 $1\leqslant l'\leqslant L'$)个隐藏层而言，其编码参数 $\delta^{(l')}=\{\boldsymbol{W}^{t(l')},\boldsymbol{b}^{t(l')}\}$，参数训练过程中添加的噪声比例为 ρ_2；SDA3 含有 L''（$L''\in\mathrm{N}$，且 $L''\geqslant 1$)个隐藏层，相邻两层间的激活函数为 $relu(x)=\max(0,x)$，对于其第 l''（$l''\in\mathrm{N}$，且 $1\leqslant l''\leqslant L''$）个隐藏层而言，其编码参数 $\eta^{(l'')}=\{\boldsymbol{W}^{r(l'')},\boldsymbol{b}^{r(l'')}\}$，参数训练过程中添加的噪声比例为 ρ_3。

流量异常检测组件主要基于多重流量特征实现 Softmax 分类器的训练，并在训练过程中采用 dropout 技术对 SDA 的隐藏层神经元进行变换，进一步提高 Softmax 分类器的泛化能力，从而准确将网络流量分为正常流量和 $k(k\in\mathrm{N}$，且 $k\geqslant 1)$类攻击流量。

2 基于 SDA 的流量特征提取方法

采用无标签训练样本依次对 3 个 SDA 进行训练，进而提取多样性流量特征。对于 SDA1 而言，其逐层训练的过程如图 2 所示[11]，图中 $\boldsymbol{y}^{(l-1)},\boldsymbol{z}^{l}$ $(1\leqslant l\leqslant L)$分别表示其第 l 层输入向量和输出向量。

给定第 l $(1\leqslant l\leqslant L\text{-}1)$个 DA 的 d^{l} 维原始输入向量 $\boldsymbol{y}^{(l-1)}$，首先按比例 ρ_1 经 masking 加噪函数 q_D 映射后得到含噪输入向量 $\boldsymbol{y}^{(l-1)}$；然后采用编码函数 $f_{\theta^{(l)}}$ 对 $\boldsymbol{y}^{(l-1)}$ 进行编码，获得 $d^{(l+1)}$ 维隐藏层向量 $\boldsymbol{y}^{l}=f_{\theta^{(l)}}(\overline{\boldsymbol{y}^{(l-1)}})=sigmoid(\boldsymbol{W}^{s(l)}\overline{\boldsymbol{y}^{(l-1)}}+\boldsymbol{b}^{s(l)})$；接下来采用解码函数 $g_{\theta'^{(l)}}$ 对 $\boldsymbol{y}^{l}$ 进行解码，获得 d^{l} 维重构向量 $\boldsymbol{z}^{l}=g_{\theta'^{(l)}}(\boldsymbol{y}^{l})=sigmoid\quad(\boldsymbol{W}^{s'(l)}\boldsymbol{y}^{l}+\boldsymbol{b}^{s'(l)})$，其中 $\theta'^{(l)}=\{\boldsymbol{W}^{s'(l)},\boldsymbol{b}^{s'(l)}\}$ 为解码参数，$\boldsymbol{W}^{s'(l)},\boldsymbol{b}^{s'(l)}$ 分别为解码权重矩阵和偏置向量，且 $\boldsymbol{W}^{s'(l)}=(\boldsymbol{W}^{s(l)})^{\mathrm{T}}$；最后，通过最小化原始输入向量与重构向量间的损失函数 $L(\boldsymbol{y}^{(l-1)},\boldsymbol{z}^{l})$ 对 DA 的编解码参数进行反向训练(当 $l=1$ 时，d^{l}，$\boldsymbol{y}^{(l-1)}$，$\boldsymbol{y}^{l}$，θ^{l}，$\boldsymbol{W}^{s(l)}$，$\boldsymbol{b}^{s(l)}$，$\boldsymbol{z}^{l}$，θ'^{l}，$\boldsymbol{W}^{s'(l)}$，$\boldsymbol{b}^{s'(l)}$ 可分别简化为 d，$\boldsymbol{x}$，$\boldsymbol{y}$，θ，$\boldsymbol{W}^{s}$，$\boldsymbol{b}^{s}$，$\boldsymbol{z}$，θ'，$\boldsymbol{W}^{s'}$，$\boldsymbol{b}^{s'}$)。训练完成后将其编码得到的隐藏层向量 $\boldsymbol{y}^{(l)}$ 作为第 l+1 个 DA 的输入向量，依次完成所有 DA 的训练，最终将原始输入神经元及各个 DA 隐藏层神经元逐层进行堆叠便得到 SDA1 模型。

具体地，SDA1 中每个 DA 的参数 $\boldsymbol{W}^{s(l)},\boldsymbol{b}^{s(l)},\boldsymbol{b}^{s'(l)}$ $(1\leqslant l\leqslant L)$均采用小批量 Adam 算法

进行反向迭代训练。训练过程中，首先引入批标准化[12](Batch Normalization，BN)方法对每个小批量训练集中 m 个样本的特征进行标准化处理，有效避免梯度消失导致 $\boldsymbol{W}^{s(l)},\boldsymbol{b}^{s(l)},\boldsymbol{b}^{s'(l)}$ 过早收敛使得含噪数据重构向量与原始输入向量间的误差较大，进而影响流量特征提取准确性的问题。BN 处理过程如算法 1 所示，其中 $\boldsymbol{y}_i^{(l-1)}(i=1,2,\cdots,m)$ 表示每个小批量中第 i 个训练样本的输入特征向量，$y_i^{(l-1)}(j)$ $(j=1,2,\cdots,d^l)$ 表示输入特征向量 $\boldsymbol{y}_i^{(l-1)}$ 中的第 j 个元素。

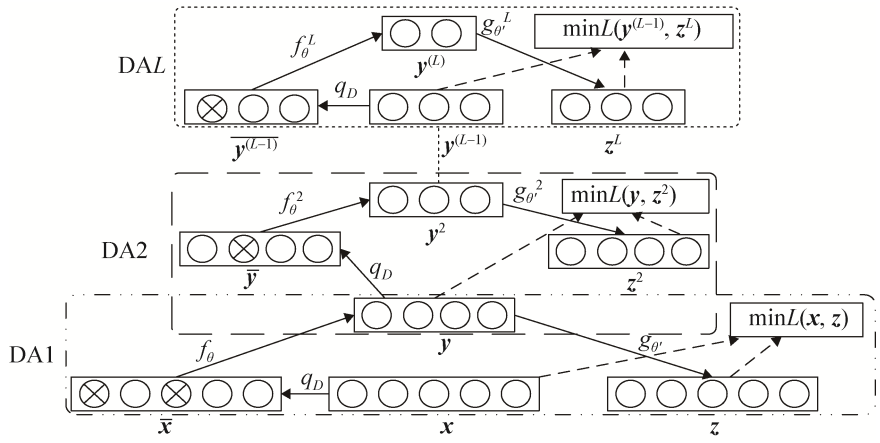


图 2 SDA 逐层训练过程

Fig. 2 Layer-wise training process of SDA

算法 1 流量特征批标准化算法

输入：小批量训练集 $B=\{\boldsymbol{y}_1^{(l-1)},\boldsymbol{y}_2^{(l-1)},\cdots,\boldsymbol{y}_m^{(l-1)}\}$

输出：标准化小批量训练集 $\overline{\overline{B}}$

for $i=1$ to m do//遍历 B 中所有样本

for $j=1$ to d^l do//遍历 B 中第 i 个样本的输入特征

$$\mu(j)\leftarrow\frac{1}{m}\sum_{i=1}^{m}y_i^{(l-1)}(j)\text{//计算 }B\text{ 中特征 }j\text{ 的均值}$$

$$\sigma^2(j)\leftarrow\frac{1}{m}\sum_{i=1}^{m}(y_i^{(l-1)}(j)-\mu(j))^2\text{//计算 }B\text{ 中特征 }j\text{ 的方差}$$

$$\overline{\overline{y_i^{(l-1)}(j)}}\leftarrow\frac{y_i^{(l-1)}(j)-\mu(j)}{\sigma(j)}\text{//计算样本 }\boldsymbol{y}_i^{(l-1)}\text{ 中特征 }j\text{ 的标准化值}$$

end for

$$\overline{\overline{\boldsymbol{y}_i^{(l-1)}}}\leftarrow\left[\overline{\overline{y_i^{(l-1)}(1)}},\overline{\overline{y_i^{(l-1)}(2)}},\cdots,\overline{\overline{y_i^{(l-1)}(d^l)}}\right]\text{//采用标准化特征值构建标准化样本 }\boldsymbol{y}_i^{(l-1)}$$

end for

$$\overline{\overline{B}}\leftarrow\left\{\overline{\overline{\boldsymbol{y}_1^{(l-1)}}},\overline{\overline{\boldsymbol{y}_2^{(l-1)}}},\cdots\overline{\overline{\boldsymbol{y}_i^{(l-1)}}},\overline{\overline{\boldsymbol{y}_m^{(l-1)}}}\right\}\text{//采用标准化样本构建标准化小批量训练集 }\overline{\overline{B}}$$

return $\overline{\overline{B}}$

然后，在将 $\boldsymbol{b}^{s(l)},\boldsymbol{b}^{s'(l)}$ 初始化为 0，并采用 Xavier[13]方法初始化 $\boldsymbol{W}^{s(l)}$ 的基础上，采用式(1)~(3)所示规则对 $\boldsymbol{W}^{s(l)},\boldsymbol{b}^{s(l)},\boldsymbol{b}^{s'(l)}$ 进行更新。

$$\begin{cases} g^{\boldsymbol{W}^{s(l)}}=\dfrac{1}{m}\sum_{i=1}^{m}\dfrac{\partial L(\boldsymbol{y}_i^{(l-1)},\boldsymbol{z}_i^l)}{\partial\boldsymbol{W}^{s(l)}}, \\ \boldsymbol{r}^{\boldsymbol{W}^{s(l)}}=\dfrac{\beta_1\boldsymbol{r}^{\boldsymbol{W}^{s(l)}}+(1-\beta_1)g^{\boldsymbol{W}^{s(l)}}}{1-\beta_1^t}, \\ \boldsymbol{v}^{\boldsymbol{W}^{s(l)}}=\dfrac{\beta_2\boldsymbol{v}^{\boldsymbol{W}^{s(l)}}+(1-\beta_2)g^{\boldsymbol{W}^{s(l)}}*g^{\boldsymbol{W}^{s(l)}}}{1-\beta_2^t}, \\ \boldsymbol{W}^{s(l)}=\boldsymbol{W}^{s(l)}-\varepsilon\dfrac{\boldsymbol{r}^{\boldsymbol{W}^{s(l)}}}{\sqrt{\boldsymbol{v}^{\boldsymbol{W}^{s(l)}}}+\xi}, \end{cases} \tag{1}$$

$$\begin{cases} g^{\boldsymbol{b}^{s(l)}}=\dfrac{1}{m}\sum_{i=1}^{m}\dfrac{\partial L(\boldsymbol{y}_i^{(l-1)},\boldsymbol{z}_i^l)}{\partial\boldsymbol{b}^{s(l)}}, \\ \boldsymbol{r}^{\mathbf{b}^{s(l)}}=\dfrac{\beta_1\boldsymbol{r}^{\boldsymbol{b}^{s(l)}}+(1-\beta_1)g^{\boldsymbol{b}^{s(l)}}}{1-\beta_1^t}, \\ \boldsymbol{v}^{\boldsymbol{b}^{s(l)}}=\dfrac{\beta_2\boldsymbol{v}^{\boldsymbol{b}^{s(l)}}+(1-\beta_2)g^{\boldsymbol{b}^{s(l)}}*g^{\boldsymbol{b}^{s(l)}}}{1-\beta_2^t}, \\ \boldsymbol{b}^{s(l)}=\boldsymbol{b}^{s(l)}-\varepsilon\dfrac{\boldsymbol{r}^{\boldsymbol{b}^{s(l)}}}{\sqrt{\boldsymbol{v}^{\boldsymbol{b}^{s(l)}}}+\xi}. \end{cases} \tag{2}$$

$$\begin{cases} g^{\boldsymbol{b}'^{s(l)}} = \dfrac{1}{m}\sum\limits_{i=1}^{m}\dfrac{\partial L(\boldsymbol{y}_i^{(l-1)},\boldsymbol{z}_i^l)}{\partial \boldsymbol{b}'^{s(l)}}, \\ \boldsymbol{r}^{\boldsymbol{b}'^{s(l)}} = \dfrac{\beta_1 \boldsymbol{r}^{\boldsymbol{b}'^{s(l)}} + (1-\beta_1) g^{\boldsymbol{b}'^{s(l)}}}{1-\beta_1^t}, \\ \boldsymbol{v}^{\boldsymbol{b}'^{s(l)}} = \dfrac{\beta_2 \boldsymbol{v}^{\boldsymbol{b}'^{s(l)}} + (1-\beta_2) g^{\boldsymbol{b}'^{s(l)}} * g^{\boldsymbol{b}'^{s(l)}}}{1-\beta_2^t}, \\ \boldsymbol{b}'^{s(l)} = \boldsymbol{b}'^{s(l)} - \varepsilon \dfrac{\boldsymbol{r}^{\boldsymbol{b}'^{s(l)}}}{\sqrt{\boldsymbol{v}^{\boldsymbol{b}'^{s(l)}}} + \xi}, \end{cases} \tag{3}$$

式(2)~(3)中：$L(\boldsymbol{y}_i^{(l-1)},\boldsymbol{z}_i^l)$ 为 $\overline{\overline{B}}$ 中第 i 个样本训练过程中的损失函数，计算方式如式(4)所示；$g^{\boldsymbol{W}^{s(l)}}, g^{\boldsymbol{b}^{s(l)}}, g^{\boldsymbol{b}'^{s(l)}}$ 分别为 $L(\boldsymbol{y}_i^{(l-1)},\boldsymbol{z}_i^l)$ 关于 $\boldsymbol{W}^{s(l)}, \boldsymbol{b}^{s(l)}, \boldsymbol{b}^{s'(l)}$ 的梯度；$\boldsymbol{r}^{\boldsymbol{W}^{s(l)}}, \boldsymbol{r}^{\boldsymbol{b}^{s(l)}}, \boldsymbol{r}^{\boldsymbol{b}'^{s(l)}}$ 分别为 $\boldsymbol{W}^{s(l)}, \boldsymbol{b}^{s(l)}, \boldsymbol{b}^{s'(l)}$ 的一阶矩变量，$\boldsymbol{v}^{\boldsymbol{W}^{s(l)}}, \boldsymbol{v}^{\boldsymbol{b}^{s(l)}}, \boldsymbol{v}^{\boldsymbol{b}'^{s(l)}}$ 分别为 $\boldsymbol{W}^{s(l)}, \boldsymbol{b}^{s(l)}, \boldsymbol{b}^{s'(l)}$ 的二阶矩变量，其初始值均为 0；β_1, β_2 为矩估计指数衰减速度；*表示逐元素相乘；t 表示 SDA 训练过程中的迭代次数；ε 为学习率；ξ 为用以保持数值稳定性的小常数。

$$L(\boldsymbol{y}_i^{(l-1)},\boldsymbol{z}_i^l) = -\sum_{j=1}^{d^l}\Big[y_i^{(l-1)}(j)\ln z_i^l(j) + \Big(1-y_i^{(l-1)}(j)\Big)\ln\Big(1-z_i^l(j)\Big)\Big], \tag{4}$$

式中：$z_i^l(j) = sigmoid(h_i^{s(l)}(j))$ 是 $\boldsymbol{z}_i^l$ 中的第 j 个元素；$h_i^{s(l)}(j) = \sum\limits_{j'=1}^{d^{(l+1)}} W_{j,j'}^{s'(l)} y_i^l(j') + b^{s'(l)}(j)$ 为 $z_i^l(j)$ 总的输入项，$W_{j,j'}^{s'(l)}$ 是 $\boldsymbol{W}^{s'(l)}$ 的第 j 行第 j' 列元素；$y_i^{(l)}(j')$ 为隐藏层向量 $\boldsymbol{y}_i^l$ 的第 j' 个元素；$b^{s'(l)}(j)$ 为 $\boldsymbol{b}^{s'(l)}$ 的第 j 个元素；$y_i^l(j') = sigmoid(a_i^{s(l-1)}(j'))$，$a_i^{s(l-1)}(j') = \sum\limits_{j=1}^{d^l} W_{j',j}^{s(l)} \overline{\overline{y_i^{(l-1)}}}(j) + b^{s(l)}(j')$ 为 $y_i^{(l)}(j')$ 总的输入项；$W_{j',j}^{s(l)}$ 为 $\boldsymbol{W}^{s(l)}$ 的第 j' 行第 j 列元素；$\overline{\overline{y_i^{(l-1)}}}(j)$ 为 $\overline{\overline{B}}$ 中 $\overline{\overline{\boldsymbol{y}_i^{(l-1)}}}$ 含噪输入向量 $\overline{\overline{\boldsymbol{y}_i^{(l-1)}}}$ 的第 j 个元素；$b^{s(l)}(j')$ 为 $\boldsymbol{b}^{s(l)}$ 的第 j' 个元素。

由矩阵求导法则可知，计算 $\dfrac{\partial L(\boldsymbol{y}_i^{(l-1)},\boldsymbol{z}_i^l)}{\partial \boldsymbol{W}^{s(l)}}$，$\dfrac{\partial L(\boldsymbol{y}_i^{(l-1)},\boldsymbol{z}_i^l)}{\partial \boldsymbol{b}^{s(l)}}$，$\dfrac{\partial L(\boldsymbol{y}_i^{(l-1)},\boldsymbol{z}_i^l)}{\partial \boldsymbol{b}^{s'(l)}}$ 相当于 $L(\boldsymbol{y}_i^{(l-1)},\boldsymbol{z}_i^l)$ 分别对 $\boldsymbol{W}^{s(l)}$，$\boldsymbol{b}^{s(l)}$，$\boldsymbol{b}^{s'(l)}$ 中的每个元素求偏导。具体地，对于 $\boldsymbol{W}^{s(l)}$ 中任意元素 $W_{j',j}^{s(l)}$，$\boldsymbol{b}^{s(l)}$ 中任意元素 $b^{s(l)}(j')$，以及 $\boldsymbol{b}^{s'(l)}$ 中任意元素 $b^{s'(l)}(j)$，有：

$$\begin{cases} \dfrac{\partial L(\boldsymbol{y}_i^{(l-1)},\boldsymbol{z}_i^l)}{\partial W_{j',j}^{s(l)}} = \\ \sum\limits_{j=1}^{d^l}\dfrac{\partial L(\boldsymbol{y}_i^{(l-1)},\boldsymbol{z}_i^l)}{\partial z_i^{(l)}(j)}\dfrac{\partial z_i^{(l)}(j)}{\partial y_i^{(l)}(j')}\dfrac{\partial y_i^{(l)}(j')}{\partial W_{j',j}^{s(l)}}, \\ \dfrac{\partial L(\boldsymbol{y}_i^{(l-1)},\boldsymbol{z}_i^l)}{\partial b^{s(l)}(j')} = \\ \sum\limits_{j=1}^{d^l}\dfrac{\partial L(\boldsymbol{y}_i^{(l-1)},\boldsymbol{z}_i^l)}{\partial z_i^l(j)}\dfrac{\partial z_i^l(j)}{\partial y_i^l(j')}\dfrac{\partial y_i^l(j')}{\partial b^{s(l)}(j')}, \\ \dfrac{\partial L(\boldsymbol{y}_i^{(l-1)},\boldsymbol{z}_i^l)}{\partial b^{s'(l)}(j)} = \\ \dfrac{\partial L(\boldsymbol{y}_i^{(l-1)},\boldsymbol{z}_i^l)}{\partial z_i^l(j)}\dfrac{\partial z_i^l(j)}{\partial b^{s'(l)}(j)}, \end{cases} \tag{5}$$

式中：$\dfrac{\partial L(\boldsymbol{y}_i^{(l-1)},\boldsymbol{z}_i^l)}{\partial z_i^l(j)} = \dfrac{z_i^l(j)-y_i^{(l-1)}(j)}{z_i^l(j)(1-z_i^l(j))}$，$\dfrac{\partial z_i^{(l)}(j)}{\partial y_i^{(l)}(j')} = z_i^{(l)}(j)'W_{j,j'}^{s'(l)}$，$\dfrac{\partial y_i^{(l)}(j')}{\partial W_{j',j}^{s(l)}} = y_i^{(l)}(j')'\overline{\overline{y_i^{(l-1)}}}(j)$，$\dfrac{\partial y_i^{(l)}(j')}{\partial b^{s(l)}(j')} = y_i^{(l)}(j')'$，$\dfrac{\partial z_i^{(l)}(j)}{\partial b^{s'(l)}(j)} = z_i^{(l)}(j)'$，且 $z_i^{(l)}(j)' = z_i^{(l)}(j)(1-z_i^{(l)}(j))$，$y_i^{(l)}(j')' = y_i^{(l)}(j')(1-y_i^{(l)}(j'))$。

采用式(5)进行逐项更新后即得到 $L(\boldsymbol{y}_i^{(l-1)},\boldsymbol{z}_i^l)$ 关于 $\boldsymbol{W}^{s(l)}, \boldsymbol{b}^{s(l)}, \boldsymbol{b}^{s'(l)}$ 的偏导数，最后将得到的偏导数分别代入式(1)~(3)并进行设定的迭代次数，即得到训练阶段参数 $\boldsymbol{W}^{s(l)}, \boldsymbol{b}^{s(l)}, \boldsymbol{b}^{s'(l)}$ 的最优解。SDA1 中所有 DA 的参数 $\boldsymbol{W}^{s(l)}, \boldsymbol{b}^{s(l)}, \boldsymbol{b}^{s'(l)}$ 训练完成后，可利用该 SDA 的编码函数和编码参数 $\boldsymbol{W}^{s(l)}, \boldsymbol{b}^{s(l)}$，通过前向传播过程逐层计算 SDA1 的隐藏层向量，并最终得到原始输入向量 $\boldsymbol{x}$ 采用 SDA1 提取的流量特征 $\boldsymbol{y}^L$。

对于 SDA2 而言，其训练方法与 SDA1 相似，此处不再赘述。反向迭代训练过程中 $z_i^l(j)' = 1-(z_i^l(j))^2$，$y_i^l(j')' = 1-y_i^l(j')^2$，SDA2 训练完成后，可通过前向传播过程获得原始输入向量 $\boldsymbol{x}$ 的流量特征 $\boldsymbol{y}^{L'}$。

对于 SDA3 而言，其训练方法在 SDA1 的基础上减少对每个小样本训练集的批标准化处理，因为其激活函数 $relu(x)$ 本身具有一定的减缓梯度消失的作用，其他训练过程与 SDA1 相似，此处不再赘

述。反向迭代过程训练过程中 $z_i^l(j)'=\begin{cases}1, & h_i^{r(l)}(j)>0\\ 0, & h_i^{r(l)}(j)\le 0\end{cases}$，$y_i^{(l)}(j')=\begin{cases}1, & a_i^{r(l-1)}(j')>0\\ 0, & a_i^{r(l-1)}(j')\le 0\end{cases}$，其中 $h_i^{r(l)}(j)$、$a_i^{r(l-1)}(j')$ 分别表示采用 *relu*(*x*)激活函数时 SDA3 中 $z_i^l(j)$、$y_i^l(j')$ 总的输入项。SDA3 训练完成后，可通过前向传播计算获得原始输入向量 $\boldsymbol{x}$ 的流量特征 $\boldsymbol{y}^{(L'')}$。

3 基于 Softmax 的流量异常检测方法

在采用 SDA1、SDA2、SDA3 分别提取流量特征的基础上，联合 3 个 SDA 提取的流量特征并输入 Softmax 分类器进行训练，同时利用 dropout 技术对 3 个 SDA 的隐藏层神经元逐层进行变换(dropout 率分别为 p_1，p_2，p_3)，实现对 SDA 不同子网络输出特征所训练的 Softmax 分类器的综合集成，有效提高 Softmax 分类器的泛化能力，从而避免过拟合现象导致的 Softmax 分类器检测性能差的问题，训练过程如图 3 所示。

首先按前向传播过程，同时采用 SDA1、SDA2、SDA3 对有标签验证集中第 i 个流量样本 V_i 的特征逐层进行提取，并将经 dropout 变换后提取到的流量特征 $\boldsymbol{y}_{V_i}^{L}$，$\boldsymbol{y}_{V_i}^{L'}$，$\boldsymbol{y}_{V_i}^{L''}$ 进行联合后输入 Softmax 函数[14]，得到 V_i 的结果向量 $\boldsymbol{y}_{PV_i}$。

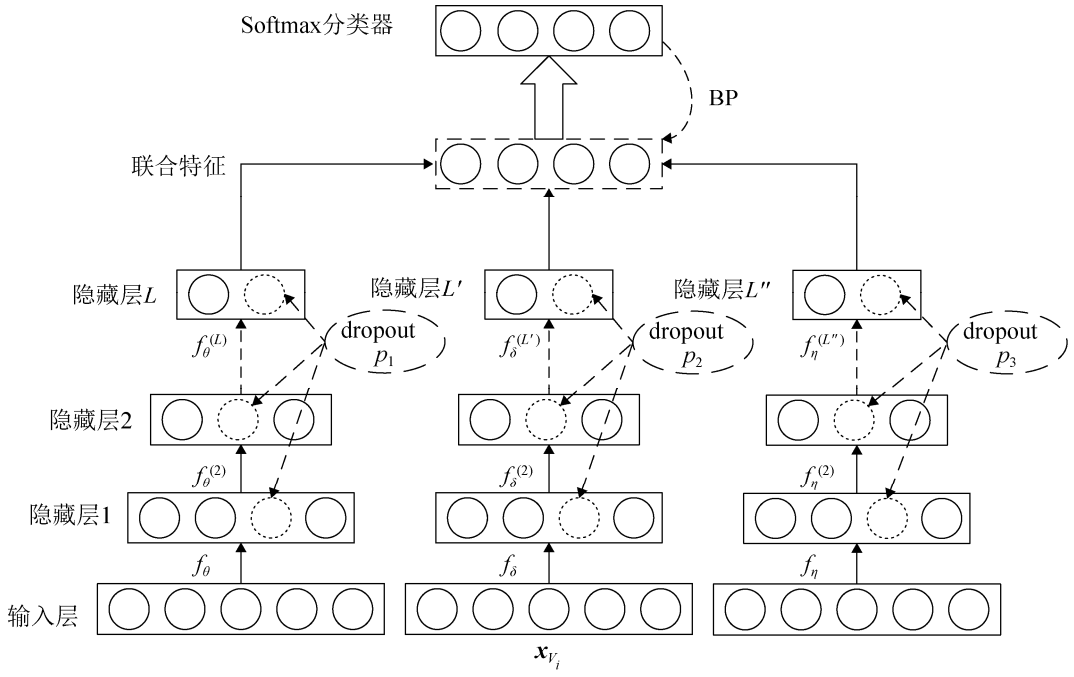


图 3 Softmax 训练过程
Fig. 3 Training process of Softmax

反向传播阶段，$\boldsymbol{W}_1,\boldsymbol{b}_1$ 同样采用小批量 Adam 算法进行迭代训练，其训练目标为最小化式(6)所示分类代价函数。

$$J(\boldsymbol{W}_1,\boldsymbol{b}_1)=-\frac{1}{m}[\sum_{i=1}^{m}\sum_{q=1}^{k+1}y_{TV_i}(q)\ln y_{PV_i}(q)], \tag{6}$$

式中：$y_{TV_i}(q)$ 为样本 V_i 的真实类别，当 V_i 的真实类别为 q 时，$y_{TV_i}(q)=1$，否则 $y_{TV_i}(q)=0$；$y_{PV_i}(q)$ 为 V_i 结果向量 $\boldsymbol{y}_{PV_i}$ 的第 q 个元素。

$\boldsymbol{W}_1,\boldsymbol{b}_1$ 更新规则如式(7)~(8)所示：

$$\begin{cases} g^{\boldsymbol{W}_1}=\dfrac{\partial J(\boldsymbol{W}_1,\boldsymbol{b}_1)}{\partial \boldsymbol{W}_1}\ , \\ \boldsymbol{r}^{\boldsymbol{W}_1}=\dfrac{\beta_1\boldsymbol{r}^{\boldsymbol{W}_1}+(1-\beta_1)g^{\boldsymbol{W}_1}}{1-\beta_1^t}\ , \\ \boldsymbol{v}^{\boldsymbol{W}_1}=\dfrac{\beta_2\boldsymbol{v}^{\boldsymbol{W}_1}+(1-\beta_2)g^{\boldsymbol{W}_1}*g^{\boldsymbol{W}_1}}{1-\beta_2^t}, \\ \boldsymbol{W}_1=\boldsymbol{W}_1-\varepsilon\dfrac{\boldsymbol{r}^{\boldsymbol{W}_1}}{\sqrt{\boldsymbol{v}^{\boldsymbol{W}_1}}+\xi}, \end{cases} \tag{7}$$

DOI: 10.16182/j.issn1004731x.joss.19-0573

$$\begin{cases} g^{\boldsymbol{b}_1}=\dfrac{\partial J(\boldsymbol{W}_1,\boldsymbol{b}_1)}{\partial \boldsymbol{b}_1}, \\ \boldsymbol{r}^{\boldsymbol{b}_1}=\dfrac{\beta_1 \boldsymbol{r}^{\boldsymbol{b}_1}+(1-\beta_1)g^{\boldsymbol{b}_1}}{1-\beta_1^t}, \\ \boldsymbol{v}^{\mathbf{b}_1}=\dfrac{\beta_2 \boldsymbol{v}^{\boldsymbol{b}_1}+(1-\beta_2)g^{\boldsymbol{b}_1}*g^{\boldsymbol{b}_1}}{1-\beta_2^t}, \\ \boldsymbol{b}_1=\boldsymbol{b}_1-\varepsilon\dfrac{\boldsymbol{r}^{\boldsymbol{b}_1}}{\sqrt{\boldsymbol{v}^{\boldsymbol{b}_1}}+\xi}, \end{cases} \tag{8}$$

式(7)~(8)中：$g^{\boldsymbol{W}_1}, g^{\boldsymbol{b}_1}$ 分别为 $\boldsymbol{W}_1, \boldsymbol{b}_1$ 的梯度；$\boldsymbol{r}^{\boldsymbol{W}_1}, \boldsymbol{r}^{\boldsymbol{b}_1}$ 分别为 $\boldsymbol{W}_1, \boldsymbol{b}_1$ 的一阶矩变量；$\boldsymbol{v}^{\boldsymbol{W}_1}, \boldsymbol{v}^{\boldsymbol{b}_1}$ 分别为 $\boldsymbol{W}_1, \boldsymbol{b}_1$ 的二阶矩变量，其初始值均为 0；$\boldsymbol{b}_1$ 的初始值为 0，$\boldsymbol{W}_1$ 采用 Xavier 方法进行初始化。

训练完成后，最终得到 Softmax 分类器参数 $\boldsymbol{W}_1, \boldsymbol{b}_1$ 的最优解 $\boldsymbol{W}_1^*, \boldsymbol{b}_1^*$，此后该 Softmax 可用于检测不平衡流量环境下的网络攻击。对于测试集中的第 i 个流量样本 T_i，其在采用 3 个 SDA 获取流量联合特征 $\boldsymbol{y}_{CT_i}$ 后，采用训练好的 Softmax 分类器进行检测，输出其预测类别 $P_{T_i}=\arg\max_q \text{Softmax}(\boldsymbol{W}_1^* \boldsymbol{y}_{CT_i}+\boldsymbol{b}_1^*)$。

4 实验

实验主要基于 Windows 平台和 NSL-KDD 数据集，采用 Python3.6.5 语言、Tensorflow1.8.0 框架和 Scikit-learn 0.21.3 学习库等对 TADMFL 方法的检测性能进行测试，分析该方法对不平衡流量环境下小流量攻击的检测率。

4.1 实验数据

NSL-KDD 作为广泛应用于流量异常检测领域的基准数据集，共包含 5 类流量：Normal，DoS，Probe，R2L 和 U2R，且不同类流量间存在较大的不平衡性，可用于对不平衡流量分类方法进行测试。实验中选择 NSL-KDD 数据集中的 KDDTrain+作为训练集，选择 KDDTrain+_20Percent 作为验证集，选择 KDDTest+作为测试集，实验数据集的流量分布情况如表 1 所示。

实验中只保留数据集与流量相关的 28 个特征，同时将符号型特征转变为数值型特征，即将属性特征“protocol type”的 3 个取值用 0~2 表示，将“service”的 70 个取值用 0~69 表示，将“flag”的 11 个取值用 0~10 表示。此外，为便于进行综合对比评价，全部采用式(9)所示方法对 28 维属性特征进行归一化处理，将每个属性特征的取值映射到[0,1]范围内。

$$x'=\frac{x-x_{\min}}{x_{\max}-x_{\min}}, \tag{9}$$

式中：x 为属性值，$x_{\max}$，$x_{\min}$ 分别为该属性的最大和最小值。

表 1 实验数据集流量分布情况
Tab. 1 Traffic distribution of the experimental datasets

流量类型	训练集	验证集	测试集
Normal	67 343	13 449	9 711
DoS	45 927	9 234	7 458
Probe	11 656	2 289	2 421
R2L	995	209	2 754
U2R	52	11	200

4.2 评价标准及参数设置

实验采用准确率(Accuracy，A)、查准率(Precision，P)、查全率(Recall，R)、**F**–measure 对检测方法进行测试。其中 A 反映了被正确分类的正类样本和负类样本占所有样本的比例，A 值越高，分类器的总体分类性能越好，其定义如式(10)所示；P 反映了被正确分类的正类样本占所有被分类为正类的样本个数的比例，P 的值越高，分类器的误报率越低，其定义如式(11)所示；R 反映了被正确分类的正类样本占所有正类样本的比例，R 的值越高，分类器的漏报率越低，其定义如式(12)所示；F–measure 反映了 P 和 R 的调和平均值，其值越大表明 P 和 R 的值越接近于 1，分类器对正类流量的检测性能越好，定义如式(13)所示。

$$A=\frac{TP+TN}{TP+TN+FP+FN}, \tag{10}$$

$$P=\frac{TP}{TP+FP}, \tag{11}$$

$$R=\frac{TP}{TP+FN}, \tag{12}$$

$$F-measure=\frac{2PR}{P+R}, \tag{13}$$

式(10)~(13)中：*TP* 为被正确判断为正类的正类样本数；*FP* 为被错误判断为正类的负类样本数；*TN* 为被正确判断为负类的负类样本数；*FN* 为被错误判断为负类的正类样本数。

实验中设置 3 个 SDA 训练过程中的噪声比例分别为 $\rho_1=0.05$，$\rho_2=0.1$，$\rho_3=0.15$；Softmax 训练过程中 3 个 SDA 的 dropout 率分别为 p_1=0.1，p_2=0.15，p_3=0.2；Adam 算法参数采用默认设置；小批量大小 m=50；SDA 训练阶段的迭代次数为 50，Softmax 训练阶段迭代次数为 10；3 个不同 SDA 的结构分别采用粒子群优化(Particle Swarm Optimization，PSO)算法进行优化[15]，优化目标为最大化 *F–measure* 的最小值，即 $fitness=1-\min$ (*F–measure*)。

4.3 检测性能分析

实验中，将所提 TADMFL 方法与基于随机森林的流量异常检测方法(Traffic Anomaly Detection Method Based on Random Forest，TADRF)、3 种基于单个 SDA 的流量异常检测方法(Traffic Anomaly Detection Method Based on SDA，TADSDA)、基于 Focal Loss 的流量异常检测方法(Traffic Anomaly Detection Method Based on Focal Loss，TADFL)，及基于双重自编码器特征[8]的流量异常检测方法(Traffic Anomaly Detection Method Based on Dual Autoencoders Features，TADDAF)进行对比，分别测试不同方法的流量异常检测性能。TADRF 方法的参数采用 Scikit-learn 0.21.3 学习库的默认设置，将决策树的数量设为 10；TADSDA 类方法中 SDA 的结构、激活函数、训练噪声比例和 dropout 概率等分别与 TADMFL 中的 SDA1、SDA2 和 SDA3 一致，依据其激活函数不同，将 3 种 TADSDA 类方法分别记为 TADSDA_S(采用 *sigmoid*(*x*)作为激活函数)、TADSDA_T(采用 *tanh*(*x*)作为激活函数)和 TADSDA_R(采用 *relu*(*x*)作为激活函数)；TADFL 采用与 TADSDA_S 同样的结构与参数，只是将其 Softmax 分类器的目标函数由交叉熵函数改为 focal loss 函数，focal loss 的参数依据文献[16]中方法设定；TADDAF 中 SAE 结构和激活函数与文献[8]一致，分类器采用 Softmax，训练算法和迭代次数与 TADMFL 方法一致。实验中 TADRF 同时采用带标签的训练集和验证集作为训练集；对于 TADMFL、TADSDA_S、TADSDA_T、TADSDA_R、TADFL 和 TADDAF 方法而言，其均采用无标签训练集对 SDA 或 SAE 进行训练，采用少量有标签验证集对分类器进行训练。每个实验独立重复 10 次并取平均值，实验结果如表 2 所示。

表 2 方法检测性能对比

Tab. 2 Detection performance comparison of different methods /%

流量类型		TADRF	TADSDA_S	TADSDA_T	TADSDA_R	TADFL	TADDAF	TADMFL
	A	76.38	75.48	76.59	75.14	77.59	78.23	**82.75**
	P	65.66	74.02	75.47	73.93	75.83	76.12	**79.54**
Normal	*R*	96.75	97.49	96.83	97.01	97.93	97.79	**98.95**
	F-measure	78.23	84.27	84.83	83.91	85.51	85.67	**88.61**
	P	**95.79**	95.51	95.03	95.18	95.69	95.15	95.72
DoS	*R*	78.78	73.01	76.15	72.44	77.12	77.39	**83.45**
	F-measure	86.44	82.76	84.55	82.27	85.71	85.76	**89.59**
	P	84.16	84.87	81.44	81.95	**87.35**	86.11	87.26
Probe	*R*	62.49	60.06	60.08	63.55	67.23	65.59	**67.32**
	F-measure	71.67	70.33	69.06	70.74	76.31	74.34	**76.36**
	P	98.94	95.99	93.75	94.03	99.12	99.14	**99.37**
R2L	*R*	10.57	10.15	14.07	6.91	18.45	15.78	**21.41**
	F-measure	19.09	18.35	24.41	12.86	32.11	27.25	**35.45**
	P	52.89	91.67	93.33	95.00	97.12	96.98	**98.13**
U2R	*R*	1.75	3.20	2.85	3.30	7.34	5.43	**8.64**
	F-measure	3.38	6.18	5.52	6.37	13.94	10.31	**16.15**

由表 2 可知，虽然 TADMFL 方法对 DoS 攻击流量的查准率略低于 TADRF 方法，对 Probe 攻击的查准率略低于 TADFL 方法，但其对所有流量分类的准确率、查全率和 F-measure 值较高，表明 TADMFL 方法具有较好的流量分类性能，且对正类流量的检测性能较好。特别是在 R2L 和 U2R 这两种小流量攻击的检测方面，TADMFL 方法具有较高的攻击流量查准率、查全率和 F-measure 值，相比 TADRF，TADSDA_S，TADSDA_T，TADSDA_R，TADFL 和 TADDAF 方法，TADMFL 方法对 R2L 攻击流量的查准率分别提高了 0.43%，3.52%，5.99%，5.68%，0.25%和 0.23%；查全率分别提高了 1.03 倍、1.11 倍、52.17%、2.10 倍、16.04%和 35.68%；F–measure 值分别提高了 85.70%，93.19%，45.23%，1.76 倍、10.40%和 30.09%；对 U2R 攻击流量的查准率分别提高了 85.54%，7.05%，5.14%，3.29%，1.04%和 1.19%；查全率分别提高了 3.94 倍、1.70 倍、2.03 倍、1.62 倍、17.71%和 59.12%；F-measure 值分别提高了 3.78 倍、1.61 倍、1.93 倍、1.54 倍、15.85%和 56.64%。

7 种检测方法流量分类准确率均值及其标准差如图 4 所示，在 5 种流量检测结果上的 F-measure 均值及其标准差如图 5 所示。

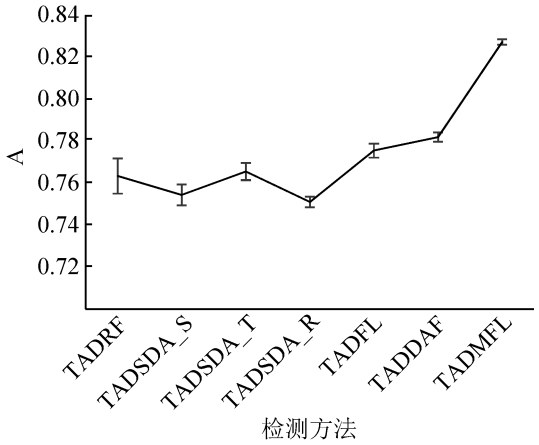
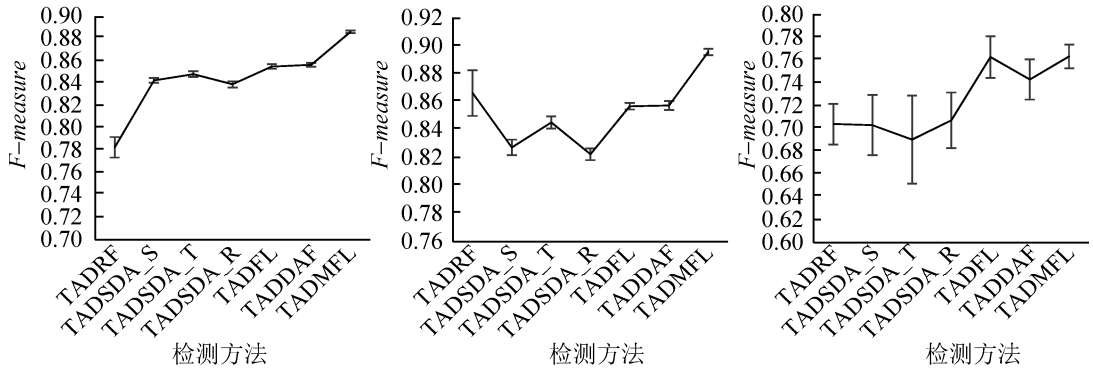
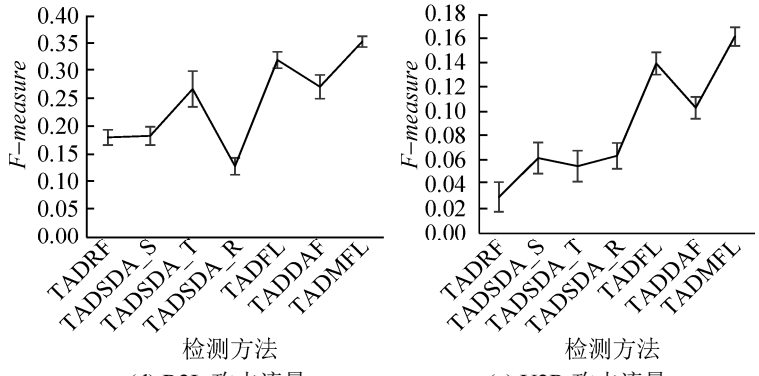


图 4 7 种方法分类准确率均值及其标准差比较

Fig. 4 Comparison of the mean and standard deviation of classification accuracy for the seven methods



(a) Normal 流量 (b) DoS 攻击流量 (c) Probe 攻击流量



(d) R2L 攻击流量 (e) U2R 攻击流量

图 5 7 种方法 F–measure 均值及其标准差比较

Fig. 5 Comparison of the mean and standard deviation of *F–measure* for the seven methods

由图 4~5 可知，相比 TADRF，TADSDA_S，TADSDA_T，TADSDA_R，TADFL 和 TADDAF 方法，TADMFL 方法分类准确率均值较大、标准差较小，表明 TADMFL 方法不仅具有较好的总体分类性能，且方法稳定性较好。TADMFL 方法在 5 种流量检测结果上具有较大的 *F-measure* 均值和较小的 *F-measure* 标准差，说明 TADMFL 方法既具有较好的正类流量检测性能，又具有较好的检测性能稳定性。

综上，TADMFL 方法能够显著提高总体流量分类性能及小流量攻击检测效果，且检测性能稳定。

5 结论

本文提出一种不平衡流量环境下基于多重特征学习的网络流量异常检测方法，该方法通过变换 SDA 激活函数、优化 SDA 结构、改变训练噪声比例等措施提高不同特征空间提取流量特征的准确性和鲁棒性，并采用 dropout 技术对 Softmax 分类器的泛化能力进行强化，从而有效提高了小流量攻击检测性能，且检测性能具有较好的稳定性。下一步将对 TADMFL 方法中采用的 SDA 数量进行寻优，并通过变换分类器类型，进一步提高其小流量攻击检测能力。

参考文献：

[1] Maurya C, Toshniwal D, Venkoparao G. Online Anomaly Detection via Class-imbalance Learning[C]// 8th International Conference on Contemporary Computing. Piscataway, New Jersey, USA: IEEE, 2015: 30-35.

[2] Junsomboon N, Phienthrakul T. Combining Over-sampling and Under-sampling Techniques for Imbalance Dataset[C]// 9th International Conference on Machine Learning and Computing. New York, USA: ACM, 2017: 243-247.

[3] 任家东, 刘新倩, 王倩, 等. 基于 KNN 离群点检测和随机森林的多层入侵检测方法[J]. 计算机研究与发展, 2019, 56(3): 566-575.
Ren Jiadong, Liu Xinqian, Wang Qian, et al. A Multi-level Intrusion Detection Method based on KNN Outlier Detection and Random Forests[J]. Journal of Computer Research and Development, 2019, 56(3): 566-575.

[4] Aburomman A, Reaz M. A Survey of Intrusion Detection Systems based on Ensemble and Hybrid Classifiers[J]. Computer & Security (S0167-4048), 2017, 65: 135-152.

[5] Kim J, Han Y, Lee J. Particle Swarm Optimization-deep Belief Network-based Rare Class Prediction Model for Highly Class Imbalance Problem[J]. Concurrency and Computation: Practice & Experience (S1532-0626), 2017, 29(11): 1-11.

[6] Kwon D, Kim H, Kim J, et al. A Survey of Deep Learning-based Network Anomaly Detection[J]. Cluster Computing (S1386-7857), 2019, 22(s1): 949-961.

[7] Yu Y, Long J, Cai Z. Session-based Network Intrusion Detection Using a Deep Learning Architecture [C]// 14th International Conference on Modeling Decisions for Artificial Intelligence. Berlin, German: Springer, 2017: 144-155.

[8] Ng W, Zeng G, Zhang J, et al. Dual Autoencoders Features for Imbalance Classification Problem[J]. Pattern Recognition (S0031-3203), 2016, 60: 875-889.

[9] Kingma D, Ba J. Adam: A method for Stochastic Optimization[C]// 3rd International Conference for Learning Representations. New York, USA: arXiv, 2015.

[10] Canadian Institute for Cybersecurity. The NSL-KDD dataset [EB/OL]. [2019-10-15]. https://www.unb.ca/cic/datasets/ nsl.html.

[11] Vincent P, Larochelle H, Lajoie I, et al. Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network With a Local Denoising Criterion[J]. Journal of Machine Learning Research (S1532-4435), 2010, 11: 3371-3408.

[12] Liu M, Wu W, Gu Z, et al. Deep Learning based on Batch Normalization for P300 Signal Detection[J]. Neurocomputing (S0925-2312), 2018, 275: 288-297.

[13] 陈建廷, 向阳. 深度神经网络训练中梯度不稳定现象研究综述[J]. 软件学报, 2018, 29(7): 2071-2091.
Chen Jianting, Xiang Yang. Survey of Unstable Gradients in Deep Neural Networks Training[J]. Journal of Software, 2018, 29(7): 2071-2091.

[14] 谷丛丛, 王艳, 严大虎, 等. 基于自编码组合特征提取的分类方法研究[J]. 系统仿真学报, 2018, 30(11):

DOI: 10.16182/j.issn1004731x.joss.19-0573

4132-4140.

Gu Congcong, Wang Yan, Yan Dahu, et al. Research on Classification based on Autoencoder Combination Features Extraction Method[J]. Journal of System Simulation, 2018, 30(11): 4132-4140.

[15] Qolomany B, Maabreh M, Al-Fuqaha A, et al. Parameters Optimization of deep Learning Models using Particle Swarm Optimization [C]// 13th International Wireless Communications and Mobile Computing Conference. Piscataway, New Jersey, USA: IEEE, 2017: 1285-1290.

[16] Li R, Xiao X, Ni S, et al. Byte Segment Neural Network for Network Traffic Classification [C]// IEEE/ACM 26th International Symposium on Quality of Service. Piscataway, New Jersey, USA: IEEE, 2018: 1-10.