计算机工程与应用
Computer Engineering and Applications
ISSN 1002-8331,CN 11-2127/TP

《计算机工程与应用》网络首发论文

题目：　改进 ADASYN-SDA 的入侵检测模型研究
作者：　陈虹，赵建智，肖成龙，陈建虎，肖越
网络首发日期：　2019-01-23
引用格式：　陈虹，赵建智，肖成龙，陈建虎，肖越．改进 ADASYN-SDA 的入侵检测模型研究[J/OL]．计算机工程与应用．
http://kns.cnki.net/kcms/detail/11.2127.TP.20190122.1702.019.html

改进 ADASYN-SDA 的入侵检测模型研究

陈　虹，赵建智，肖成龙，陈建虎，肖　越

CHEN Hong, ZHAO Jianzhi, XIAO Chenglong, CHEN Jianhu, XIAO Yue

辽宁工程技术大学 软件学院，辽宁 葫芦岛 125105

School of Software, Liaoning Technical University, Huludao, Liaoning 125105, China

CHEN Hong, ZHAO Jianzhi, XIAO Chenglong, et al. Research on Improved Intrusion Detection Model of ADASYN-SDA. Computer Engineering and Applications

Abstract：Aiming at the poor detection performance of traditional intrusion detection models in high-dimensional data and data imbalance environment,an intrusion detection model combining Adaptive Synthetic Sampling Approach(ADASYN) with improved Stacked Denoising Autoencoder (SDA) is proposed.Firstly, the data oversampling process is performed using the ADASYN approach.Secondly, the Adam optimization approach and Dropout regularization are used to improve the SDA deep learning model, and the integration features of low dimensionality and high robustness are extracted.Finally, intrusion detection is performed in the softmax classifier.Experiments show that compared with SDA, AE-DNN and MSVM, the ADASYN-SDA model has a certain degree of improvements in average accuracy, detection rate and false positive rate.

Key words：stacked denoise autoencoder(SDA); Adaptive Synthetic Sampling Approach(ADASYN); deep learning; intrusion detection

摘　要：针对传统入侵检测模型在高维数据且数据不均衡环境下检测性能较差的问题，提出了一种自适应过采样算法(ADASYN)与改进堆叠式降噪自编码器(SDA)结合的入侵检测模型。首先使用 ADASYN 算法进行数据过采样处理。其次使用 Adam 优化算法，以及 Dropout 正则化对 SDA 深度学习模型进行改进，提取出低维数、高鲁棒性的集成特征。最后在 softmax 分类器中进行入侵检测识别。实验表明，ADASYN-SDA 模型相较于 SDA、AE-DNN 和 MSVM 模型，在平均准确率、检测率和误判率上均有一定程度的提高。

关键词：堆叠式降噪自编码器(SDA)；自适应过采样算法(ADASYN) ；深度学习；入侵检测

文献标志码：A　中图分类号：TP393.08　doi：10.3778/j.issn.1002-8331.1811-0013

基金项目：国家自然科学基金项目：自动识别自定义指令提高高层次综合效率的研究 (No.61404069)；辽宁省教育厅科学技术研究项目(No.LJ2017QL032)。

作者简介：陈虹(1967－),女,硕士, 副教授,CFF 会员,研究领域为信息安全,网络安全,E-mail: chh3188@163.com;赵建智(1995－),男,硕士研究生,研究领域为信息安全，网络安全;肖成龙(1984－),男，博士,副教授，研究领域为软硬件协同，高层次综合，可扩展处理器;陈建虎(1992－),男,甘肃兰州人,硕士,研究领域为入侵检测,机器学习;肖越(1993－),女,硕士研究生,研究领域为入侵检测,机器学习。

1 引言

进入 21 世纪，随着 IT 技术的发展与网络信息化的建设，各种各样的网络信息广泛传播，给人们生活带来快捷和便利的同时，其所造成的网络安全问题也日益严峻[1]，如何对入侵攻击类型进行有效检测，以及预警与保护系统的安全，成为网络安全问题的研究方向之一。

入侵检测系统(IDS,Intrusion Detection System)是一种对网络或主机传输网络流量进行监控，检测违反系统安全策略入侵行为的主动安全防护措施[2]。一般来说，入侵检测方法分为基于误用和异常的检测。基于误用的检测能够以低误报率检测已知类型网络攻击。基于异常的检测方法通过正常活动捕捉偏差识别攻击行为。与误用检测不同的是，基于异常的检测方法可以更好识别未知攻击行为。但该方法仍面临着高误报率的风险。入侵检测[3]系统一般使用误用与异常检测的混合方法来判别网络或系统中入侵攻击行为[4]。

网络环境中可能存在着数量极少破坏力却极大的攻击数据，即网络环境中存在着数据比例不平衡问题，如何在这些网络数据中取得较为理想的检测效果，以及在海量高维度的网络数据环境中，尽可能避免检测模型过拟合现象，成为入侵检测研究领域中不可避免的关键性问题。目前，基于浅层神经网络与简易的机器学习[5]的入侵检测方法，已经难以满足人们的需求，于是提出了深度学习[6]的入侵检测方法。

深度学习的概念于 2006 年由 Hinton 等人提出，深度神经网络具有逐层性，参数多等特点，具备拟合任意复杂函数的优势，因此，可以进行复杂的非线性映射，使用深度学习神经网络模型处理海量高维度数据[7]进行入侵检测，实现高维数据的低维度集成式特征，提取出更优质的数据特征，得出检测效果更高的分类结果。

近年来，入侵检测研究领域学者提出了多种基于深度学习的检测模型。Niyaz，Q[8]等人提出一种基于深度学习的自学习(STL)技术的网络入侵检测方法，该模型使用无监督学习的方式从未标记的数据中学习良好的特征表示，再进行分类任务。Fiore，U[9]等人提出使用判别受限的受限制玻尔兹曼机(DRBM)，用于半监督式的网络异常检测，提升神经网络泛化能力，并对提高分类精度方面有明显效果。Yin C[10]等人提出一种深度学习方法，使用递归神经网络(RNN-IDS)进行检测，提高入侵检测的准确性。

过高的数据维度会使模型出现过拟合的现象，训练集上表现良好，但模型对新数据的泛化学习能力下降，从而导致模型检测率的降低。自编码器对于高维度数据的集成式特征学习可以较好的解决该问题。

2006 年，Hinton 等人提出采用自编码器对高维数据进行降维处理，避免高维数据产生的维数灾难[11]。2007 年，Bengio 等人模仿 RBM 构建 DBN 的方法，使用自编码器实现堆叠式自编码[12]。2008 年，Vincent P 等人提出通过向传统自编码器中加入一定概率分布的噪声，修改了基本的自编码器，实现从破损不完整的数据集实现到对数据集的修复[13],从而得到鲁棒性更强的数据集。2010 年，Vincent P 等人又提出基于去噪自编码器的逐层相连的结构，将降噪自编码器隐藏层作为下一层降噪自编码器的输入，构成堆叠式降噪自编码器[14]，相比于普通自编码器，具备了更强的特征学习能力。2014 年，srivastava 等人提出了调节 Dropout 正则化中的超参数 p（在网络中保留某个神经元的概率，独立与其他神经元），即以概率 p 保留神经元，构造一种更加“稀疏”的网络来防止 DBN 网络出现过拟合问题[15]。2017 年，兰州大学陶亮亮等人同样提出了使用 Dropout 方法来解决数据不均衡造成的堆叠式自编码器深度网络过拟合问题，提升模型的泛化学习能力，同时将 Simgoid 替换为 Relu，加快网络的收敛速度[16]。

数据的不平衡问题导致少数类入侵数据的检测率下降，本文中所采用的 NSL-KDD 数据集[17,18]存在一定程度的数据比例不均衡问题，即存在某一种或几种少数类别类型的数据所占比例很小，导致少数类别的分类检测率低。解决数据比例的不平衡问题，数据层面的处理方法包括重采样和数据合

成。本文使用基于数据合成 ADASYN 算法[19]。该算法根据少数样本的分布特点自适应在难以分类的地方合成新样本。首先针对低频率攻击检测率低的问题，使用 ADASYN 算法自适应合成新样本添加到训练数据集中，提高少数类的检测率。其次针对入侵检测模型检测率低的问题，将 Dropout 正则化[20]与 Adam 优化算法[21]应用于堆叠式降噪自编码器进行模型改进。利用深度学习的逐层性，使数据空间中的高维特征表达映射为低维特征表达，在非监督学习过程中，实现海量高维度数据到低维度鲁棒性数据的特征重构。可以降低深度学习网络的收敛时间，提高入侵检测模型的检测性能。

2 改进模型中使用的算法

2.1 ADASYN 算法

ADASYN(Adaptive Synthetic Sampling Approach)算法是自适应综合过采样算法。该方法可以根据少数类样本的概率分布 r_i（r_i 为少数样本合成数目的判定准则）自适应地合成少数类样本，将合成的新样本添加到数据集中，在难以分类的类别中合成更多的样本，使样本比例达到相对均衡的效果，缓解数据不平衡的问题。

假设训练集样本 D 包含 m 个样本 $\{x_i, y_i\}$，$i=1,2,3,...,m$，其中 x_i 是 n 维特征空间 X 的一个样本，$y_i \in Y=\{0,1,2,3,4\}$ 是类标签，$y_i=3,4$ 时为少数样本，$y_i=0,1,2$ 为多数类样本。这里用 m_s 和 m_l 分别表示少类和多数类样本的数目。因此，有 $m_s \le m_l$，且 $m_s+m_l=m$。

算法流程如下：

(1) 计算不平衡度 $d=m_s/m_l$，式中 $d \in (0,1]$。

(2) 计算合成少数样本数：$G=(m_l-m_s)*\beta$，式中 $\beta \in [0,1]$ 表示加入合成样本后的不平衡度。$\beta=1$ 表示加入合成样本后多数类和少数类完全平衡，G 等于少数类和多数类的差值。

(3)对少数类的每个样本 x_i，找出它们在 n 维空间的 K 近邻，并计算其比率 $r_i=\Delta_i/K$，$i=1,2,...,m$，式中 Δ_i 是 x_i 的 K 近邻中多数类的数目。因此，$r_i \in (0,1]$。

(4)根据 $\hat{r}_i = r_i / \sum_{i=1}^{m_s} r_i$ 正则化 r_i，那么 r_i 为概率分布（$\sum \hat{r}_i = 1$），计算每个少数类样本周围多数类的情况。

(5)根据每个少数样本 x_i 计算合成的样本数目 g_i：$g_i=\hat{r}_i \times G$，式中 G 是合成样本的总数。

(6)在每个待合成的少数类样本周围 k 个邻居中选择 1 个少数类样本，根据下列等式进行合成：

$$s_j = x_i + (x_{zi} - x_i) \times \lambda$$

2.2 自编码器

自编码器(AE, AutoEncoder)是一种简单的神经网络判别模型属于无监督网络。自编码器主要用于捕捉可以代表输入数据的最重要因素，y 可以作为原始数据 x 的一种集成特征表达。自编码器在结构上根据隐层的数量可分为浅层自编码器和堆叠式自编码器。自编码器的基本原理如图 1 所示。

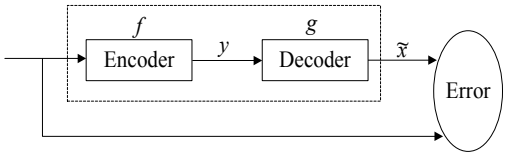


图 1 自编码器原理

自编码器基于以下流程：

(1)从输入层到隐藏层的计算过程，即原始数据输入 x 经过神经元之间加权与偏置 $\theta_1=(W_1,b_1)$ 处理后，进行编码器 Encoder 中的激活函数 f 编码，得到了编码数据 y。

$$y = f_{\theta_1}(W_1 x + b_1) \tag{1}$$

(2)从隐藏层到输出层的计算过程，即编码数据 y 经过 $\theta_2=(W_2,b_2)$ 后，在进行解码器 Decoder 的函数 g 解码后，得到编码重构 $\tilde{x}$ 。

$$\tilde{x}=g_{\theta_2}(W_2y+b_2) \tag{2}$$

(3)编码器的权值矩阵 W_1，通常与解码器的权值矩阵 W_2 互为转置，即 $W_1^T=W_2$ 。隐层神经元的激活函数通常使用 Sigmoid 函数，通过非线性变换找到输入数据的输入表示。通过对参数 θ 进行反复的迭代调优，使编码重构误差 L_{loss} 达到最小。

$$L_{loss}=\|x-\tilde{x}\|^2 \tag{3}$$

2.3 堆叠式降噪自编码器

自动编码(AE)对于完整保留的数据输入，通过简单重构可以学习到特征表达，在原始数据中加入数据噪声，对数据集进行破坏，可以实现破损数据集到正常数据集的还原，可以使用降噪噪编码器(DAE)进行噪声数据集的误差重构，从而得到鲁棒性更强的集成式特征表达。降噪自编码器原理如图 2 所示。

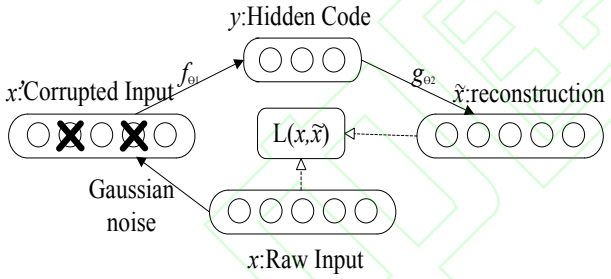


图 2　降噪自编码器

降噪自编码器(DAE)是在自动编码器的基础上，在原始输入(raw input)中加入概率分布为 q_D 数据高斯噪声(Gaussian noise) $x\sim q_D(x'|x)$ 的随机映射，将 x 实例中部分特征置为零，可以得到“破坏后”的噪声数据输入 x' (Corrupted Input)。定义联合分布函数：

$$q^0(X,X',Y)=q^0(X)q_D(X'|X)\delta_{f_\theta(X')}(Y) \tag{4}$$

Y 为 X' 的映射函数，以 θ 为参数。并使用梯度下降算法最小化目标函数：

$$\arg_{\theta_1\theta'}\min=E_{q^0(X,X')}[L_H(X,g_{\theta'}(f_\theta(X')))] \tag{5}$$

当处理大规模，高纬度的数据集时，浅层自编码器很难发挥出它的作用，堆叠式自编码器处理海量高维数据时，使用深层模型可以提取出更加有效的集成式特征。

堆叠式降噪自编码器与 2006 年 Hinton 提出的深度信念网络(DBN)以相同的方式来初始化神经网络。为了可以学习到有效的特征提取，噪声输入仅用于初始化单层的噪声数据训练。

降噪自编码器由三层神经网络构成，向第一层自编码器 DA_1 输入“噪声数据” x'，得到函数 f_θ，再使用 DA_1 中的 f_θ 函数学习原始数据 x，并使用 DA_1 的原始数据的输出作为第 2 层自编码器的输入，将第 2 层输入再次进行破坏，并重复训练第一层自编码器的过程。

形成逐层连接，实现堆叠式降噪自编码器结构如图 3 所示。噪声输入通过逐层自编码器的训练，实现特征变换的过程($x\rightarrow DA_1\rightarrow DA_2\rightarrow.....\rightarrow DA_n$)。

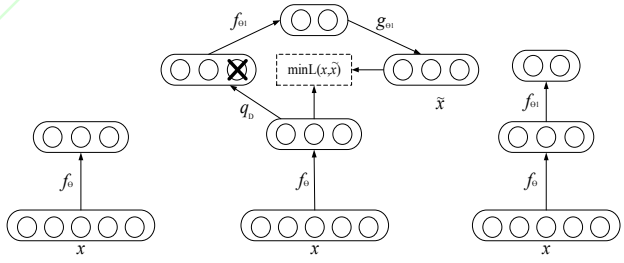


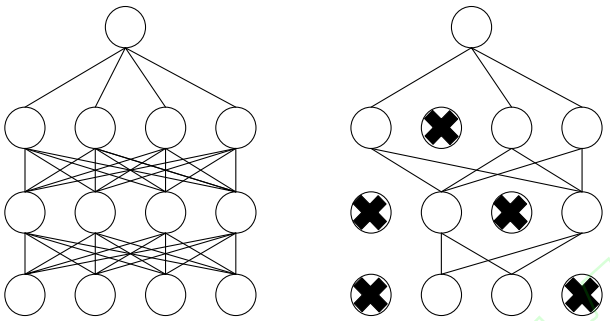
图 3　堆叠式去噪自编码器

在堆叠式降噪自编码器模型输出降噪降维的数据后，最后一层采用 softmax 层，使用 BP 算法进行权值微调，将神经网络的参数进行调整，并实现对数据集的分类功能。

2.4 Dropout 正则化

典型神经网络训练通过网络进行正向传导输入值，之后将网络生成的误差进行反向传播。

Dropout 针对该过程，随机删除隐藏层单元，由于删除隐藏单元是随机过程，并且随机采取一定批次训练集进行训练，保证最后每个模型用相同权重来融合，实现类似 boosting[22]算法的效果。保证每 2 个隐含节点不能每次同时出现，使全连接网络具有了一定稀疏化的性质，减轻不同特征的协同效应（即有效组织了某些特征在其他特征存在下才有效果的情况），增加了神经网络的鲁棒性。Dropout 可以有效减轻过拟合的发生，一定程度上达到了正则化的效果，并且减少神经网络收敛时间。Dropout 工作原理如图 4 所示。



（a）标准神经网络　　（b）执行dropout后

图 4　dropout 工作原理

2.5　Adam 优化算法

Adam（适应性矩阵估计）算法是随机梯度下降算法的扩展式，是一种可以替代传统随机梯度下降(SGD)[23]过程的一阶优化算法，近年来被广泛应用于深度学习中。它能够训练数据，并更新神经网络的权重。能够进行高效计算，适应性的改变学习率，所需内存少，减少神经网络收敛时间，适用于解决大规模数据和参数优化，非稳定目标以及高噪声和稀疏梯度问题。

随机梯度下降算法保持单一学习率调节自编码器中各层的权值。学习率在训练过程中不会改变。Adam 算法结合了 Adagrad 算法[24]善于处理稀疏梯度和 RMSprop 算法[25]善于处理非平稳目标的优点，为不同参数提供不同的自适应学习率，适用于高维空间和大数据集。

Adam 优化算法如下：

(1)计算参数的梯度值 g

Input: h_{n-1} 与 h_n 的损失函数，数据集的小批量 m

Output：计算参数的梯度值 g

For $i=1,2,3,...,n$

$$g=\frac{1}{m}\nabla_{\theta}\sum_{i}L(h_{n-1},h_n) \tag{6}$$

End for

(2)计算矩估计 m_t 和 v_t

Input:指数衰减率 $\beta_1, \beta_2\in[0,1]$

Output:修正后一阶、二阶矩阵估计 m_t 与 v_t

While$\theta_t$not converged do

m_t 和 v_t 分别为有偏差的一阶矩估计和有偏差的二阶矩估计。

$$m_t=\beta_1\cdot m_{t-1}+(1-\beta_1)\cdot g_t \tag{7}$$

$$v_t=\beta_2\cdot v_{t-1}+(1-\beta_2)\cdot g_t^2 \tag{8}$$

其中 $\beta_1, \beta_2\in[0,1]$为控制 m_t 与 v_t 的指数衰减率。

对一阶矩估计与二阶矩估计进行相应的偏差值修正，如公式(9)和(10)。

$$\hat{m}_t=\frac{m_t}{1-\beta_1^t} \tag{9}$$

$$\hat{v}_t=\frac{v_t}{1-\beta_2^t} \tag{10}$$

梯度 g 的矩估计经过偏置矫正后，每一次迭代

学习率都有一个固定范围，使参数相对平稳。

(3) 参数 θ 更新

通过以上计算实现模型参数 θ 的更新，其中 ε 为用于数值稳定的小常数，防止寻找最优梯度时发生除零现象。当参数 θ 没有收敛时，循环迭代更新各个部分。

Output:模型参数 θ_t

$$\theta_t = \theta_{t-1} - \frac{\alpha \cdot \hat{m}_t}{(\sqrt{\hat{v}_t} + \varepsilon)} \tag{11}$$

End while

Returnθ_t

3 改进 SDA 的入侵检测模型及算法

3.1 ADASYN-SDA 模型

基于改进堆叠式降噪自编码的入侵检测模型(ADASYN-SDA)如图 5 所示。其工作流程如下：

(1) 预处理(Pre-treatment)：第一步，对数据集单条实例进行数据与标签分离。第二步，将字符型特征进行数值化处理，实现数据高维空间映射。第三步，进行数据集最大最小归一化。第四步，进行标签 one-hot 编码。第五步，采用 ADASYN 算法对训练集进行数据过采样操作，增加少数样本 R2L，U2R 攻击类型数据的数量。

(2) 特征降维(Extract feature)：第一步，使用 Dropout 正则化与 Adam 优化算法对神经网络进行改进，针对其隐层层数、节点数进行设计。第二步，神经网络模型的预训练、权值微调。第三步，用堆叠式降噪自编码器对预处理数据集进行集成式特征提取。

(3) 入侵检测(Intrusion detection)：将深度网络模型生成的低维度、高鲁棒性数据集输入到 softmax 分类器[26]中，进行入侵检测。

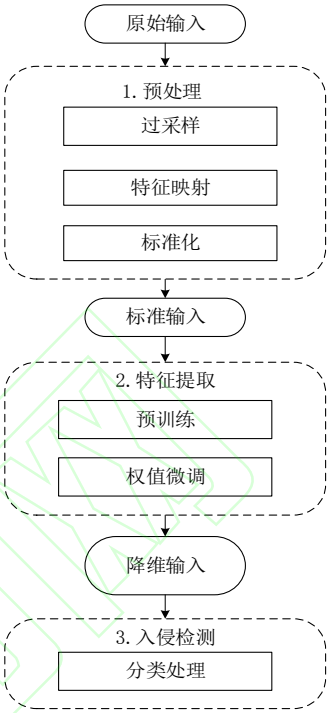


图 5 ADASYN-SDA 入侵检测模型

3.2 降噪自编码器参数更新

降噪自编码器模型参数初始化，其目的在于找出降噪自编码器模型最合适的参数，可按照以下步骤进行参数更新，具体过程如算法 1 所示。

算法 1：DAE

Input：原始输入 X，输入层节点数，隐藏层节点数，学习率 l_r，噪声破坏率 c_r，最大迭代次数 $T_{\max}$

Output：模型参数 $\theta = \{W, b\}$，隐藏层输出 Y

算法如下：

(1) X' = get_corrupted_input(X,c_r)

(2) For T from 1 to T_{max}

(3) Y = get_hidden(X',W_1,b_1)

(4) Z = get_reconstruction(Y,W_2,b_2)

(5) *cost* = get_cost(*X,Z*)

(6) for *param* in Adam(*W,b*)

(7) *gparam* = get_gradient(*cost,param*)

(8) *param* = *param* - *l_r***gparam*

(9) End for

(10) End for

3.3 SDA 模型训练

SDA 模型的预训练与权值微调如图 6 所示。

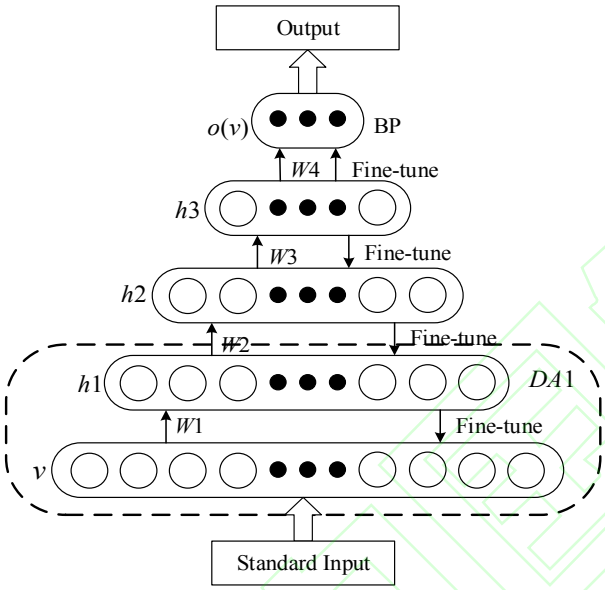


图 6 SDA 模型预训练与微调

堆叠式降噪自编码器预训练的过程中，采用无监督贪婪逐层预训练对每一层降噪自编码器进行预训练，进行深度网络模型的权值初始化，并将上一层自编码器的隐藏层作为下一编码器的输入层，逐层进行连接，预训练的过程如算法 2 所示。

算法 2：Pre-Train-DAE

Input：隐藏层输出 *Y_out*，隐藏层层数 *n_layers*，学习率 *l_r*，噪声破坏率 *c_r*，单层自编码器最大迭代次数 $T_{\max}$

Output：初始化模型参数 $\theta^h{}_i = \{W[i], b[i]\}$

算法如下：

(1)For *i* from 1 to *n_layers*

(2)*X'* = get_corrupted_input(*Y_out,c_r [i]*)

(3) For *T* from 1 to T_{max}

(4) *Y* = get_hidden(*X',W1[i],b1[i]*)

(5) *Z* = get_reconstruction(*Y,W2[i],b2[i]*)

(6) *Cost* = get_cost(*Y_out,Z*)

(7) For *param* in Adam(*W[i],b[i]*)

(8) *gparam* = get_gradient(*cost,param*)

(9) *param* = *param-l_r[i]*gparam*

(10) End for

(11) End for

(12) *Y_out* = *X*

(13) End for

堆叠降噪自编码器权值微调的过程中，使用反向传播算法对深度网络模型进行有监督的权值微调，将原始数据与重构数据之间的重构误差降低至最小。

算法 3：Fine-Tune-DAE

Input：预训练初始化参数 $\theta^h{}_i = \{W[i], b[i]\}$，训练批次 *m*，测试批次 *n*，微调迭代次数 T_{max}

Output：堆叠式降噪自编码器模型(SDA)

算法如下：

(1) While $T<T_{max}$ do

(2) For *batch* in *m*

(3) *Cost* = get_fine_cost(*X*)

(4) For *i* from 1 to *n_layers*

(5) For *param* in Adam*(W[i],b[i]*)

(6) *gparam* = get_gradient(*cost,param*)

(7) *param = param-l_r[i]*gparam*

(8) End for

(9) End for

(10) *valid_error* = get_valid_error(*batch*)

(11) if *valid_error* <*best_valid_error*

(12) update *best_vaild_error*

(13) *test_error* = get_test_error(*n*)

(14) if *test_error*<*best_test_error*

(15) update *best_test_error*

(16) End if

(17) End if

(18) End for

(19) End while

4 实验分析

4.1 实验环境与评价指标

(1) 实验环境

操作系统：Window 10

编程环境：python 3.5、matlab7.0

处理器：Inter(R) Core(TM) i5-4210U CPU@ 1.70GHz 2.40GHz

内存(RAM)：4.00GB

系统类型：64 位硬件环境

(2) 评价指标

评估样例采用二分类混淆矩阵，如表 1 所示。其中 TP 代表正确的正例，TN 代表正确的负例，FP 代表错误的正例，FN 代表错误的负例。

评价指标如下：

准确率(Accuracy)：被正确分类的样本除以所有样本，整体模型检测能力。检测率(Detection)：被正确识别的异常样本除以异常样本，模型对攻击类别的检测能力。误判率(Error)：正常样本被误报为异常的样本除去所有正常样本，对正常样本的检测能力。

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{12}$$

$$Detection = \frac{TN}{TN + FN} \tag{13}$$

$$Error = \frac{FP}{TP + FP} \tag{14}$$

表 1 二分类混淆矩阵

Total sample	predict positive	predict negative
Actual positive	TP	FN
Actual negative	FP	TN

4.2 实验数据

实验数据集采用 NSL-KDD 数据集作为入侵检测实验的训练集和测试集。NSL-KDD 数据集解决了 KDDCUP99 数据集存在的固有问题，不包含冗余记录，分类器不会偏向更频繁的记录，提高了分类器检测性能。实验数据分布如表 2 所示。

数据集中单条实例包括 22 种 TCP 连接的基本特征与内容特征，19 种基于主机和时间的网络流量统计特征。网络连接实例类型可以分为 5 大类，数据集正常类别数据标签为 Normal，以及 4 大攻击类型数据标签分别为 DOS（拒绝服务攻击），Probe

（端口监视或扫描），R2L（未授权的本地超级用户特权访问），U2R（来自远程主机未授权访问）。

表 2　实验数据分布

Data Label	Data type	Train	Test
0	Normal	67334	9711
1	DOS	45927	2421
2	R2L	925	2754
3	U2R	52	200
4	Probe	11656	4166

4.3　数据预处理

NSL-KDD 数据集单条实例的 41 个特征由 38 个数字型特征与 3 个字符型特征构成。数据预处理分为以下 3 个步骤：

(1)高维特征映射

本文将符号型的特征转化为二进制数字型特征，将 protocol_type_dict（协议类型）3 种字符类型 tcp-[1,0,0]，udp-[0,1,0]，icmp-[0,0,1]。server_type_dict（目标主机网络类型）70 种，flag_dict（连接正确或错误类型）11 种，字符特征转化为数字特征，将 41 维特征转化为 122 维特征，采用该方法处理数据，使归一化后数据范围为[0,1]，方便进行分类处理，减少与原数据间误差。

(2)One-Hot 编码

将各个小类型的字符型标签替换为五种大类型标签，五种字符型的数据标签进行 One-Hot 编码，过程如下：将 Normal 映射为 1,0,0,0,0；将 Probe 映射为 0,1,0,0,0；将 Dos 映射为 0,0,1,0,0；将 R2L 映射为 0,0,0,1,0；将 U2R 映射为 0,0,0,0,1。

(3)归一化

将原始数据归一化，转化到[0,1]范围的操作称为最大最小归一化。为了数据方便处理，防止大数据对小数据的覆盖，提高模型的检测性能。x 为单条数据实例的特征，$X_{\max}$ 为该特征最大值，$X_{\min}$ 为该特征最小值。

$$x^{*}=\frac{x-x_{\max}}{x_{\max}-x_{\min}} \tag{15}$$

4.4　模型参数设置

本文使用的堆叠式降噪自编码器为深度神经网络。参数设置如表 3 所示。

表 3　参数设置

SDA-softmax parameter	value
The number of layers of SDA	5
The number of nodes in input layer	122
The number of nodes in 1st hidden layer	90
The number of nodes in 2nd hidden layer	60
The number of nodes in 3rd hidden layer	30
The number of nodes in output layer	5
SDA pre-training Number of iterations	350
BP fine-tuning Number of iterations	100
Exponential decay rate of 1st moment estimation	0.9
Exponential decay rate of 2nd moment estimation	0.999
Constant stable value	1.00E-07
Dropout	0.15
Batch size	64
Noise layer	0.2 0.4 0.4 0.7

将自编码器网络总层数设置为 5 层，隐藏层数量为 3 层，其中输入层节点数为高维映射后数据维数 122 个节点，3 个隐藏层节点数量分别为 90-60-30，

输出层隐层节点数设置为 5。其中每层自编码器迭代次数设置为 70 次，网络权值 BP 反向传播算法微调迭代次数设置为 100 次。Adam 算法中，学习率设置为 0.001，一阶矩估计指数衰减率设置为 0.9，二阶矩估计指数衰减率设置为 0.999，常数稳定值为 10E-8，Dropout 正则化参数设置为 0.5，各个网络层节点的参数由逐层训练进行初始化，并且逐层加入一定概率的噪声值分别为 0.7, 0.4, 0.4, 0.2。

4.5 实验结果分析

4.5.1 模型参数分析

在 NSL-KDD 数据集的基础上进行数据抽样，使用该数据在 ADASYN-SDA 中进行对比实验，分析模型参数对模型性能的影响。数据抽样如表 4 所示。

表 4 抽样数据

抽样数据	Normal	Probe	Dos	R2L	U2R
训练	21873	8560	8012	883	52
测试	10137	5429	4331	196	10

(1) 网络层数以及隐层的节点数设置

确定最佳深层网络层数结构。为了确定 SDA 模型的最佳层数结构，进行了 5 种不同层数的对比实验。当网络层数由 2 层增长到 6 层时，训练时间有 78s 增加到 144s。而当网络层数确定在 5 层时，准确率达到了 99.39%。当层数为 6 层时，训练时间大幅增加，达到 144s。而准确率却下降了，通过综合对比，将深层网络层数设置为 5 层。层数对比如表 5 所示。

表 5 层数对比

层数	准确率	训练时间
122-5	97.97%	78s
122-60-5	98.44%	84s
122-90-60-5	98.44%	96s
122-90-60-30-5	99.39%	111s
122-90-60-30-15-5	99.23%	144s

(2) Dropout 的参数设置

确定最佳 Dropout 随机删除率。将 Dropout 随机删除率设置为从 0.1 置 0.5，当 Dropout 随机删除率设置为 0.15 时，准确率达到最高值，之后准确率开始逐渐下降，将 Dropout 随机删除率设置为 0.1 或 0.15 时，模型检测性能达到最佳值。Dropout 对比如表 6 所示。

表 6 Dropout 对比

Dropout	准确率
0.1	99.40%
0.15	99.41%
0.2	99.25%
0.3	98.95%
0.5	98.34%

(3)优化器对比

确定最佳优化器。选择 4 种常见的优化算法放入模型中，进行模型的检测性能对比，如表 7 所示。深度神经网络中常见的 SGD 优化器的准确率为 96.54%，检测效果一般，其中选用 Adam 优化器时，神经网络模型的检测效果最佳，准确率达到 99.53%。优化器对比如表 7 所示。

表 7 优化器对比

优化算法	准确率
SGD	96.54%
RMSprop	99.04%

Adagrad	98.51%
Adam	99.53%

4.5.2 与其他模型对比分析

(1) 数据抽样对比

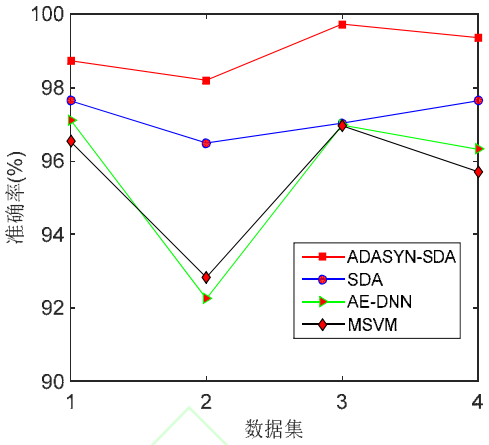
为了验证 ADASYN-SDA 模型的有效性，随机抽取 4 个实验数据集进行验证，并与 SDA、AE-DNN 和 MSVM 模型进行实验对比。4 个实验数据集数据分布如表 8 所示。

表 8 实验数据分配

序号	训练集			测试集		
	正常	异常	总数	正常	异常	总数
IDS1	10018	3069	13087	9481	2762	12243
IDS2	9369	2036	11405	7850	3206	10056
IDS3	8603	3186	11789	8169	2240	10409
IDS4	9332	2035	11367	8071	2160	10231

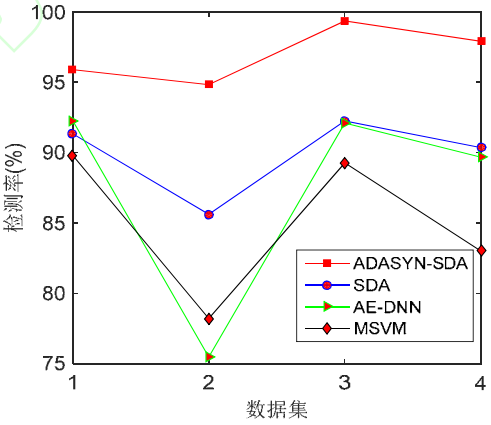
(2) 检测结果分析

将 4 个实验数据集 IDS1、IDS2、IDS3 和 IDS4 数据输入 ADASYN-SDA、SDA、AE-DNN 和 MSVM 模型中进行准确率、检测率、误报率等对比实验。实验结果如图 7~图 9 所示，4 种模型实验结果平均值如表 9 所示。



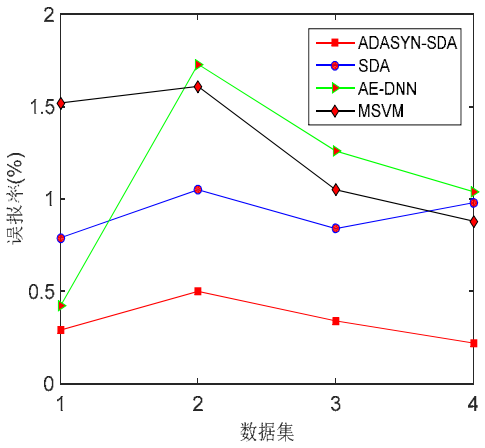
准确率(%)	IDS1	IDS2	IDS3	IDS4
ADASYN-SDA	98.73	98.20	99.73	99.36
SDA	97.64	96.49	97.03	97.64
AE-DNN	97.12	92.25	96.98	96.32
MSVM	96.53	92.83	96.96	95.71

图 7 准确率实验结果



检测率(%)	IDS1	IDS2	IDS3	IDS4
ADASYN-SDA	95.91	94.84	99.37	97.92
SDA	91.35	85.62	92.26	90.36
AE-DNN	92.28	75.48	92.11	89.68
MSVM	89.82	78.21	89.28	83.01

图 8 检测率实验结果



误报率(%)	IDS1	IDS2	IDS3	IDS4
ADASYN-SDA	0.29	0.5	0.34	0.22
SDA	0.79	1.05	0.84	0.98
AE-DNN	0.42	1.73	1.26	1.04
MSVM	1.52	1.61	1.05	0.88

图 9　误报率实验结果

表 9　四种模型实验结果平均值

平均结果(%)	ADASYN-SDA	SDA	AE-DNN	MSVM
平均准确率	99.01	97.62	94.32	94.19
平均检测率	97.01	90.27	87.39	85.08
平均误报率	0.34	0.92	1.11	1.27

由图 7~图 9 实验结果可知，ADASYN-SDA 模型在 4 种不同数据集中，ADASYN-SDA 模型的准确率、检测率和误报率三项指标均优于对比实验的 3 个模型。

稳定性分析：ADASYN-SDA 模型表现出相对平稳的检测性能，其中误报率最为平稳，4 个数据的误报率分别为 0.29%、0.5%、0.34%和 0.22%。而 AE-DNN 和 MSVM 模型在平均准确率、误报率和检测率上出现了较大波动。在稳定性上，不及 ADASYN-SDA 模型。SDA 模型，稳定性上与 ADASYN-SDA 模型相似。

性能分析：由表 9 可知 ADASYN-SDA 模型对比 SDA、AE-DNN、MSVM 模型，平均准确率分别提高了 1.39%、4.69%和 4.82%，平均检测率分别提高了 6.74%、9.62%和 11.93%，平均误报率分别降低了 0.58%、0.77%和 0.93%。

综上所述，ADASYN-SDA 模型在稳定性上优于 AE-DNN 和 MSVM 模型，在平均准确率、平均检测率和平均误报率三种评价指标上优于 SDA、AE-DNN 和 MSVM 模型。因此，ADASYN-SDA 模型在检测性能上拥有比较明显的优势。

(3) U2R，R2L 检测率

表 4 的抽样测试集中 R2L 攻击类型的数量为 196，U2R 攻击类型的数量为 10，相比其他类型实验数据样本比例严重失衡。使用 ADASYN 方法对训练集中 U2R，R2L 类型数据进行过采样处理。

同时对 ADASYN-SDA、SDA、AE-DNN 和 MSVM 四种模型。使用表 4 的抽样数据，进行检测率实验对比分析。实验表明，ADASYN-SDA 模型的分类检测率上略高于 SDA，AE-DNN，MSVM 模型，并提高了少数类别攻击类型数据(U2R，R2L)检测率，低频攻击检测率如表 10 所示。

表 10　低频攻击检测率

检测率	R2L	U2R
ADASYN-SDA	100%	97.50%
SDA	93.19%	67.50%
AE-DNN	91.87%	52.50%
MSVM	69.89%	0%

5　结束语

对于海量高维度、不平衡数据，传统入侵检测

模型效果差的问题，提出了自适应过采样算法(ADASYN)算法与堆叠式降噪自编码器(SDA) 结合算法。在 SDA 模型中加入一定概率的 Dropout 正则化，可以缓解深度网络模型容易产生过拟合现象。Adam 算法优化损失函数有效避免了局部最优问题，加快了网络收敛速度，大幅度提升了模型检测性能。针对 NSL-KDD 数据集存在数据比例不平衡问题，采用 ADASYN 算法对少数类别的数据 U2R 与 R2L 进行过采样处理，有效提升了低频攻击数据检测率。使用预处理后数据集，在 SDA 模型中进行集成式特征提取，进行入侵检测。实验表明，相较于对比模型，在准确率、检测率、误判率三个评价指标上都提升了一定的百分比，ADASYN-SDA 模型的平均准确率达到 99.01%，平均误报率达到 0.34%，平均检测率达到 97.01%。虽然该方法具有较为良好与稳定的检测性能，但仍然具有提升空间，如何提高该模型异常样本检测率，是下一步需要解决的关键问题。

参考文献：

[1] 李威,杨忠明.入侵检测系统的研究综述[J].吉林大学学报,2016,34(5),657-661

[2] 傅昊,罗守山.入侵检测系统的研究与设计[D],北京,北京邮电大学,2015:1-65

[3] 林伟宁，陈明志，詹云清，等. 一种基于 PCA 和随机森林分类的入侵检测算法研究[J]. 信息网络安全，2017，17(11)：50-54.

[4] 张新有,曾华燊,贾磊.入侵检测数据集 KDD CUP99 研究[J].计算机工程与设计.2010,31(22):4809-4816

[5] 和湘，刘晟，姜吉国. 基于机器学习的入侵检测方法对比研究[J]. 信息网络安全，2018，18(5)：1-11.

[6] 雷青，荆丽桦，赵德明，等. 基于深度学习的安卓 APP 视频枪支检测技术研究[J]. 信息网络安全，2016，16(9)：149-153.

[7] 陈虹,陈建虎,肖成龙,万广雪,肖振久. 深度学习模型下多分类器的入侵检测方法[J]. 计算机科学与探索, 2018.

[8] Javaid A, Niyaz Q, Sun W, et al. A deep learning approach for network intrusion detection system[C]//Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS). ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2016: 21-26.

[9] Fiore U, Palmieri F, Castiglione A, et al. Network anomaly detection with the restricted Boltzmann machine[J]. Neurocomputing, 2013, 122: 13-23.

[10] Yin C, Zhu Y, Fei J, et al. A deep learning approach for intrusion detection using recurrent neural networks[J]. IEEE Access, 2017, 5: 21954-21961.

[11]Hinton G E,Salakhutdinov R R.Reducing the dimensionality of data with neural Networks [J]. Science, 2006, 313(5786): 504-507

[12] Yoshua Bengio, Pascal Lamblin, Dan Popovici, Hugo Larochelle.Greedy Layer-Wise Training of Deep Networks[J]. International Conference on Neural Information Processing System, 2006, 19: 153-160

[13] Vincent P,Larochelle H,Bengio Y, et al.Extracting and Composing Robust Features with DenoisingAutoencoders[C]//Proc of the 25th Intenational Conference on Machine learning.2008:1096-1103.

[14] Vincent P,Larochelle H,Lajoie I,et al.Stacked Denoising Autoencoders: Learning Useful Representations ina Deep Network with a Local Denoising Criterion[J].Journal of Machine Learning Research,2010,11(6):3371-3408.

[15] Srivastava N,Hinton G,Krizhevsky A,et al.Dropout: A Simple Way to Prevent Neural Networks from Overtting[J]. Journal of Machine Learning Research,2014,15(1):1929-1958.

[16] 陶亮亮,林和.堆叠式降噪式自编码器深度网络的入侵

检测[D]. 甘肃：兰州大学，2017: 1-54.

[17] Dhanabal L, Shantharajah S P. A study on NSL-KDD dataset for intrusion detection system based on classification algorithms[J]. International Journal of Advanced Research in Computer and Communication Engineering, 2015, 4(6): 446-452.

[18] 陈虹, 万广雪, 肖振久. 基于优化数据处理的深度信念网络模型的入侵检测方法[J]. 计算机应用, 2017, 37(6): 1636-1643.

[19] He H, Bai Y, Garcia E A, et al. ADASYN: Adaptive synthetic sampling approach for imbalanced learning[C]//Neural Networks, 2008. IJCNN 2008.(IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on. IEEE, 2008: 1322-1328.

[20] Srivastava N, Hinton G, Krizhevsky A, et al. Dropout: a simple way to prevent neural networks from overfitting[J]. The Journal of Machine Learning Research, 2014, 15(1): 1929-1958.

[21] Kingma D P, Ba J. Adam: A method for stochastic optimization[J]. arXiv preprint arXiv:1412.6980, 2014.

[22] Abouelenien M, Yuan X. Boosting for learning frommulticlass data sets via a regularized loss function[C]//Proceedings of 2013 IEEE International Conference on Granular Computing(Gr C),Beijing, December13-15, 2013. IEEE, 2014: 4-9.

[23] Zinkevich M, Weimer M, Li L, et al. Parallelized stochastic gradient descent[C]//Advances in neural information processing systems, 2010: 2595-2603.

[24] Zeiler M D. ADADELTA: an adaptive learning rate method[J]. arXiv preprint arXiv:1212.5701, 2012.

[25] Hinton G, Srivastava N, Swersky K. Rmsprop: Divide the gradient by a running average of its recent magnitude[J]. Neural networks for machine learning, Coursera lecture 6e, 2012.

[26] Hinton G E, Salakhutdinov R R. Replicated softmax: an undirected topic model[C]//Advances in neural information processing systems. 2009: 1607-1614.