基于深度信念网络的网络入侵检测策略研究

陈 虹[1] 任瑞仙[1]
CHEN Hong REN Ruixian

摘 要 针对传统网络入侵检测方法存在特征提取能力薄弱、入侵检测精度较低的问题，文章将深度信念网络（DBN）应用于网络入侵检测任务。研究首先设计了基于深度信念网络的入侵检测整体方案；其次深入分析了深度信念网络的特征提取过程及相关核心算法，利用训练集构建适用于入侵检测场景的 DBN 模型，并结合 softmax 分类器实现对网络攻击数据类型的精准识别；最后通过对比实验，验证了不同深度 DBN 模型的入侵检测性能差异，为提升网络安全态势感知能力提供了一种可行的解决方案。

关键词 网络入侵检测；深度信念网络；softmax 分类器

doi：10.3969/j.issn.1672-9528.2026.01.025

0 引言

随着云计算、物联网等技术的飞速发展，网络空间已深度融入社会生产生活的各个方面，其安全性与稳定性变得至关重要。然而，网络攻击行为正朝着规模化、复杂化和隐蔽化的方向演进，传统的基于特征匹配和固定规则的入侵检测系统在面对零日攻击、高级持续性威胁等新型网络威胁时，往往显得力不从心[1]。为了应对这一挑战，研究者们将目光投向了以机器学习为代表的人工智能技术。特别是基于深度学习[2]的入侵检测方法，因其能够从海量、高维的网络流量数据中自动学习深层次的、非线性的特征表示，展现出巨大的应用潜力，逐渐成为当前网络安全领域的研究热点。

深度信念网络（deep belief network）[3]凭借其强大的无监督特征学习能力和高效的分层训练机制，在网络入侵检测研究中受到了广泛关注。与需要大量标注数据的监督学习模型相比，DBN 模型能够首先通过无监督的逐层预训练来初始化网络权重，从而更有效地捕捉网络流量数据中潜在的正常与异常行为模式，构建出高效的分类器，在 KDD CUP 99、NSL-KDD 等标准数据集上取得了优于传统机器学习方法的检测精度，尤其在识别未知攻击变种方面展现出独特优势。因此，深度信念网络在网络入侵检测方面的应用，对于推动下一代智能安全防御技术的发展具有重要的理论价值与现实意义。

1. 山西工程技术学院大数据与智能工程系 山西阳泉 045000

[基金项目] 山西省教育厅山西省高等学校教学改革创新项目（J20241525）；2023 年度山西省教学改革创新项目（J2023171）；2023 年校级课程思政教学改革创新项目（2023KCSZ10）

1 深度信念网络

1.1 受限玻尔兹曼机

受限玻尔兹曼机是一个基于能量函数的生成模型，它能够学习到数据内在的表示规律[4]。受限玻尔兹曼机能够为未知分布的数据提供一个学习的模型，是由一个可视层神经元和一个隐含层神经元组成，两层神经元之间全连接，而层内无连接，图 1 为 RBM 结构图。

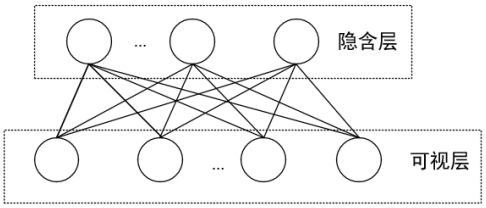

图 1 RBM 结构图

假设可视层单元的输入向量为 $\boldsymbol{v}=(v_1, v_2,\cdots, v_n)$，隐含层向量为 $\boldsymbol{h}=(h_1, h_2,\cdots, h_m)$，连接权重矩阵 $\boldsymbol{w}$，可视层神经元的阈值设为 a，隐含层神经元阈值设为 b，则受限玻尔兹曼机的能量函数可表示为：

$$E(\boldsymbol{v},\boldsymbol{h})=-\sum_{i=1}^{n}a_i\boldsymbol{v}_i-\sum_{j=1}^{m}b_j\boldsymbol{h}_j-\sum_{i=1}^{n}\sum_{j=1}^{m}\boldsymbol{w}_{ij}\boldsymbol{v}_i\boldsymbol{h}_j \quad (1)$$

式中：n 为可视层神经元数量；m 为隐含层神经元数量，根据受限玻尔兹曼机能量函数可求得可视层与隐含层神经元的联合概率分布：

$$p(\boldsymbol{v},\boldsymbol{h})=\frac{\mathrm{e}^{-E(\boldsymbol{v},\boldsymbol{h})}}{Z} \quad (2)$$

式中：Z 为标准化分数，由可视层神经元与隐含层神经元之间的能量值求和而得。由式（2）的联合概率分布，可以求出可视层神经元的独立分布公式：

$$p(\boldsymbol{v})=\sum_{\boldsymbol{h}}p(\boldsymbol{v},\boldsymbol{h})=\frac{\sum_{\boldsymbol{h}}\mathrm{e}^{-E(\boldsymbol{v},\boldsymbol{h})}}{\sum_{\boldsymbol{v}}\sum_{\boldsymbol{h}}\mathrm{e}^{-E(\boldsymbol{v},\boldsymbol{h})}} \tag{3}$$

对比散列算法[5]是将能得到受限玻尔兹曼机参数的估计值，为了使其学习过程稳定且收敛到最优，在上述算法的参数更新阶段加入动量因子，目的是将上一次迭代的梯度值引入本次参数更新中，用公式表示为：

$$\begin{aligned}[\Delta \boldsymbol{w}_{ij}]_n&=\delta[\Delta \boldsymbol{w}_{ij}]_{n-1}+\gamma(<\boldsymbol{v}_i\boldsymbol{h}_j>_0-<\boldsymbol{v}_i\boldsymbol{h}_j>_k)\\ [\Delta a_i]_n&=\delta[\Delta a_i]_{n-1}+\gamma(<\boldsymbol{v}_i>_0-<\boldsymbol{v}_i>_k)\\ [\Delta b_j]_n&=\delta[\Delta b_j]_{n-1}+\gamma(<\boldsymbol{h}_j>_0-<\boldsymbol{h}_j>_k)\end{aligned} \tag{4}$$

式中：$\delta\in[0,1]$为动量因子；n为受限玻尔兹曼机的迭代次数。

1.2 单个 RBM 特征学习

本文主要分析单个 RBM 的特征提取能力，DBN 网络结构顶层的分类器作用是分类和识别，模型分类能力取决于底层 RBM 的特征提取能力。由前文可知 RBM 学习能力的好坏直接影响到入侵检测性能的好坏。

RBN 网络学习过程中使用 sigmoid(x) = 1/(1+e^{-x}) 函数计算隐含层和可视层之间的激活概率，将 RBM 网络的输入数据设为 x，则其编码函数为 $f(x)$ = sigmoid(b+$\boldsymbol{w}x$)，解码函数为 $f(x)$ = sigmoid(a+$\boldsymbol{w}^{\mathrm{T}}x$)，其中 a、b 分别为可视层神经元和隐含层神经元的阈值，$\boldsymbol{w}$ 为连接权重矩阵，将数据输入到 RBM 模型中，通过解码函数便可获得原始数据的重构表达。

1.3 网络入侵模型训练

DBN 模型的训练可以分成两个阶段完成，首先，利用 RBM 结构进行无监督逐层训练来挖掘出隐含在数据内部的特征规律，采用的是贪婪学习算法，每次只单独训练一个 RBM，训练完毕后再接着训练下一个 RBM，并将其堆叠在上一个 RBM 之上，直到所有的 RBM 训练完毕。然后是有监督的微调阶段，使用反向传播算法并结合样本数据的标签来建立所提取出的特征和标签之间的非线性复杂对应关系，从而优化 DBN 模型的性能，提高其入侵检测能力。DBN 入侵检测模型的训练如图 2 所示。

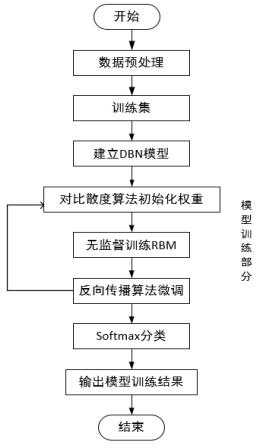

图 2 DBN 入侵检测模型训练流程图

模型训练的具体步骤如下：

步骤 1：使用训练集初始化 DBN 的可视层神经元，即将其作为第一个 RBM 进行数据的输入和训练；

步骤 2：使用对比散度算法训练第一个 RBM，重复训练直到达到最开始设置的最大训练次数为止，此时第一个 RBM 训练完毕；

步骤 3：确定第一个 RBM 的权重和偏执，根据 RBM 的激活函数计算出第一个 RBM 隐含层的输出，将此输出状态作为第二个 RBM 可视层的输入数据进行训练第二个 RBM；

步骤 4：同样使用对比散度算法训练第二个 RBM，重复训练直到完毕后，将第二个 RBM 堆叠到第一个 RBM 之上，同时将第二个 RBM 的输出作为第三个 RBM 的输入继续进行训练和堆叠；

步骤 5：按照步骤 3、4，一直执行训练算法，直到所有的 RBM 训练完毕；

步骤 6：在顶层添加标签层，使用反向传播算法对整个网络参数进行微调。该过程是利用已知标签，从 DBN 模型的最后一层开始，逐步向底层微调每一个 RBM 的参数，重复执行反向传播算法直到符合终止条件为止。

2 softmax 回归算法

数据进行特征提取之后将其输出作为 softmax 回归算法的输入，softmax 将输入的数据转换为 (0,1) 之间的概率值。由于后验概率之间存在竞争关系，即类别之间不可能同时符合同一样本，所以它能够很好地解决多分类问题，概率值越大表明其属于某种类别的可能性就越大。

softmax 回归算法的目的就是计算在给定样本数据的条件下其对应类别的概率，经过 softmax 层输出 k 维向量来表示类别概率向量。概率向量中的第 j 个元素表示属于第 j 个类别的概率，概率向量中元素的值之和为 1。用公式表示为：

$$\boldsymbol{h}_\theta(x^i)=\begin{bmatrix}p(y^i=1\mid x^i;\theta)\\ p(y^i=2\mid x^i;\theta)\\ \vdots\\ p(y^i=k\mid x^i;\theta)\end{bmatrix}=\frac{1}{\sum_{j=1}^{k}\mathrm{e}^{\theta_j^{\mathrm{T}}x^i}}\begin{bmatrix}\mathrm{e}^{\theta_1^{\mathrm{T}}x^i}\\ \mathrm{e}^{\theta_2^{\mathrm{T}}x^i}\\ \vdots\\ \mathrm{e}^{\theta_k^{\mathrm{T}}x^i}\end{bmatrix}=\begin{bmatrix}p_1\\ p_2\\ \vdots\\ p_n\end{bmatrix} \tag{5}$$

式中：$[\theta_1,\theta_2,\cdots,\theta_n]$ 是网络模型结构的参数；$\sum_{j=1}^{k}\mathrm{e}^{\theta_j^{\mathrm{T}}x^i}$ 的作用是对输出的概率值向量进行归一化，使得各类别概率值和为 1。

3 入侵检测系统模型

网络入侵检测的关键是对流量数据进行特征识别，深度信息网络能学习到复杂数据的内在规律、提取数据的关键特征，使用深度信息网络进行入侵检测的整体思路分为以下步骤。

（1）数据预处理：将原始数据集转换为可输入深度信念网络模型的数据；

（2）特征提取：使用深度信念网络对处理完毕的数据

集进行特征提取；

（3）入侵检测：将 DBN 模型的输出作为 softmax 回归分类器的输入，识别出输入数据对应的攻击类别。

DBN 模型的训练可以分成两个阶段完成，首先利用 RBM 结构进行无监督逐层训练来挖掘出隐含在数据内部的特征规律，采用的是贪婪学习算法，每次只单独训练一个 RBM，训练完毕后再接着训练下一个 RBM，并将其堆叠在上一个 RBM 之上，直到所有的 RBM 训练完毕。然后是有监督的微调阶段，使用反向传播算法并结合样本数据的标签来建立所提取出的特征和标签之间的非线性复杂对应关系，从而优化 DBN 模型的性能，提高其入侵检测能力。

模型训练的具体步骤如下：

步骤 1：使用训练集初始化 DBN 的可视层神经元，即将其作为第一个 RBM 进行数据的输入和训练。

步骤 2：使用对比散度算法训练第一个 RBM，重复训练直到达到最开始设置的最大训练次数为止，此时第一个 RBM 训练完毕。

步骤 3：确定第一个 RBM 的权重和偏执，根据 RBM 的激活函数计算出第一个 RBM 隐含层的输出，将此输出状态作为第二个 RBM 可视层的输入数据进行训练第二个 RBM。

步骤 4：同样使用对比散度算法训练第二个 RBM，重复训练直到完毕后，将第二个 RBM 堆叠到第一个 RBM 之上，同时将第二个 RBM 的输出作为第三个 RBM 的输入继续进行训练和堆叠。

步骤 5：按照步骤 3、4 一直执行训练算法，直到所有的 RBM 训练完毕。

步骤 6：在顶层添加标签层，使用反向传播算法对整个网络参数进行微调。

该过程是利用已知标签，从 DBN 模型的最后一层开始，逐步向底层微调每一个 RBM 的参数，重复执行反向传播算法直到符合终止条件为止。

4 性能指标评价

在入侵检测中，通常使用检测的准确率（Acc）、误报率（Fpr）、召回率（Re）作为评价入侵检测模型性能的指标。其中 TP 为真阳值，即实际为异常数据检测结果也是异常的数据量；TN 为真阴值，即数据实际的结果与检测结果都是正常值的数据量；FP 为假阳值，即实际数据是正常的，但检测结果为异常行为的数据量；FN 为假阴值，即实际数据是异常的但检测结果是正常的数据量。

检测的准确率（Acc）表示检测结果正确的样本数与总样本数之比，计算公式为：

$$\mathrm{Acc}=\frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FN}} \tag{6}$$

检测的误报率（Fpr）是指把正常的行为误判为入侵行为的数据量占总的正常行为的数据量的比例，其定义为：

$$\mathrm{Fpr}=\frac{\mathrm{FP}}{\mathrm{FP}+\mathrm{TN}} \tag{7}$$

召回率（Re）是指所有的异常数据中检测为异常数据，实际也为异常数据的样本所占的比例，又叫查全率、灵敏度，其计算公式为：

$$\mathrm{Re}=\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} \tag{8}$$

5 实验结果分析

5.1 数据集分析

实验中所用数据集对所提的入侵检测模型的性能有着重要作用，本文实验采用公开的网络安全数据集 NSL-KDD 是数据集 KDD CUP-1999 的改进版，没有重复数据，使得检测率更为准确，在训练集和测试集中记录数量设置比较合理。NSL-KDD 数据集共有 42 个属性，其中前 41 个是特征属性，最后 1 个是攻击类别属性。41 个特征属性按照在数据集中的序号可以分为以下类别，每一类别都有不同的含义。

研究人员在实验中选择 NSL-KDD 数据集的一部分，KDD Train20% 的训练集和 KDD test20% 的测试集。对数据统计表明，在 KDD Train20% 里 Normal 类型和 DoS 类型分别占比为 53.39% 和 36.65%，在 KDD test20% 里 Normal 类型和 DoS 类型分别占比为 43.08% 和 33.08%。由此可知，数据集存在类别严重不平衡的问题，会导致所建立的入侵检测不能反映出对未知攻击和小样本攻击的检测能力。因此，实验中根据 NSL-KDD 数据集中各类别数据的数量，所选用的训练集和测试集如表 1 所示。

表 1 数据集分布

标签	样本数	
	训练集	数据集
Normal	6 060	4 030
Dos	8 000	1 300
Probe	4 000	3 200
R2L	2 000	1 000
U2R	40	70

5.2 数据集预处理

5.2.1 数据特征转换

本文实验中采用概率质量函数（probability mass function, PMF）编码 [6] 进行转换，采用这种方法进行转换的好处是既能将其转化到 0~1 之间，又不会增加数据的维度。对于标签列数据，根据其各攻击小类对应的大类将其转化为数值型，具体如表 2 所示。

表 2 攻击类别对应表

攻击类别	对应的攻击子类型
Normal	Normal
Dos	back, land, Neptune, mailbomb, pod, smurf, teardrop, apache2, udpstorm, processtable, worm
Probe	satan, ipsweep, nmap, portsweep, mscan, saint guess_Password, ftp_write, imap, phf,multihop, spy
R2L	Warezmaster, warezclent, xlock, xsnoop, worm, Snmpguess, snmpgetattack, sendmail, named
U2R	buffer_overflow, loadmodule, rootkit, Perl, sqlattack, xterm, ps

将标签列也转换为数值型，具体如表 3 所示。

表 3 标签列转换表

标签	编码
Normal	0
Dos	1
Probe	2
R2L	3
U2R	4

5.2.2 归一化

由于数据来源于真实的网络环境，所以存在一些噪声数据和奇异样本，会导致模型训练时间长，收敛速度慢等问题。因此，需要对其进行归一化处理，使得数据的全部取值在一个固定的区间内，以减少过大或者过小数据对模型整体性能的影响，从而提高模型分类的准确率。本文采用离差归一化，离差归一化又称最大最小归一化，是一种线性的归一化方法，其具体的转换函数用公式表示为：

$$x^* = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{9}$$

式中：x 和 x^* 分别表示转换前与转换后的样本数据；$x_{\max}$ 和 $x_{\min}$ 分别表示样本数据中的最大值与最小值。

5.2.3 处理后数据集

采用上述的数据特征转换方法和归一化的方法，利用 Excel 工具和 Pandas 库对数据进行处理，结果如图 3~4 所示。

5.3 实验结果分析

在训练好 DBN 模型的基础上，将测试集输入模型进行入侵检测实验。本实验通过不同深度的 DBN 模型对比入侵检测性能结果。

实验设计了多个不同深度的 DBN 模型，已知预处理完成的数据集有 41 个特征属性和 1 个标签属性，而这 1 个标签属性对应五个类别。因此，基于数据的输入和输出需求，确定 DBN 模型的输入层神经元数量为 41，输出层为 5 个神经元，借鉴已有的研究，设置了如表 4 所示的不同深度和不同隐含层神经元数的五种 DBN 模型。

表 4 不同结构的 DBN 模型

模型	网络结构
DBN1	[41,30.5]
DBN2	[41,60,30,5]
DBN3	[41,60,40,20,5]
DBN4	[41,80,60,40,20,5]
DBN5	[41,100,80,60,40,20,5]

在这五种不同深度的 DBN 模型下分别进行入侵检测，设置每一个 RBM 的预训练都采用一次吉布斯采样算法对参数进行更新，在基于 RBM 的预训练过程中设置迭代次数为 30，微调阶段的迭代次数为 100 次，动量决定着

durat	proto	serviflag	src_bytes	dst_bytes	count	srv_csError	rv_srErro	sv_r_same	diff_srv_dst	h_dst	h_src	h_srv	h_dst_h	h_src_h	h_srv_h	dst_h_count	srv_count	label
0	tcp	ftp_dSF	491	0	2.00	2	0	0	0	1	0	0	150	25	0.17	0.03	0.17	normal
0	udp	otherSF	146	0	13.00	1	0	0	0	0.08	0.15	0	255	1	0	0.08	0.15	normal
0	tcp	privaS0	0	0	123	6	1	1	0	0.05	0.07	0	255	26	0.1	0.05	0	neptune
0	tcp	http SF	232	8153	5.00	5.00	0.2	0.2	0	1	0	0	30	255	1	0.03	0.04	normal
0	tcp	http SF	199	420	32.00	32.00	0	0	0	1	0	0.09	255	255	1	0	0	normal
0	tcp	privaREJ	0	0	19	19	0	0	1	0.16	0.06	0	255	19	0.07	0.07	0	neptune
0	tcp	privaS0	0	0	9	1	1	1	0	0.05	0.06	0	255	9	0.04	0.05	0	neptune
0	tcp	privaS0	0	0	16	1	1	1	0	0.14	0.06	0	255	15	0.06	0.07	0	neptune
0	tcp	remotS0	0	0	23	1	1	1	0	0.09	0.06	0	255	23	0.09	0.05	0	neptune
0	tcp	privaS0	0	0	8	1	1	1	0	0.06	0.05	0	255	13	0.05	0.06	0	neptune
0	tcp	privaREJ	0	0	12	0	0	0	1	0.06	0.06	0	255	12	0.07	0.07	0	neptune
0	tcp	privaS0	0	0	3	1	1	1	0	0.02	0.06	0	255	13	0.07	0.07	0	neptune
0	tcp	http SF	287	2251	7.00	7.00	0	0	0	1	0	0.43	8	219	1	0	0	normal

图 3 原数据集

dura	proto	servi	flag	src_byte	dst_byte	count	srv_count	serr	srv_serr	srv_r	same	diff	srv_c	dst_host_c	dst_host_s	r	dst_l	dst_l	dst_l	dst_l	dst_l	dst_l	label
0	0.815	0.05	0.05	0.595	3.56E-07	0	0.0039	0	0.00319389	0	0	0	0	1	0	0	0.588	0.00933	0.17	0	0.3	0.17	0
0	0.119	0.04	0	0.595	1.06E-07	0	0.0254	0	0.00195695	0	0	0	0	0.08	0.15	0	1	1	0.08031597	0.06	0.88	0	1
0	0.815	0.17	0	0.277	0	0	0.2407	0	0.01174168	1	1	0	0	0.05	0.07	0	1	1	0.10196078	0.1	0.05	0	0
0	0.815	0.32	0	0.595	1.68E-07	6.22E-06	0.0098	0	0.00978474	0.2	0.2	0	0	1	0	0	0.118	1	1	0.03	0.04	0.03	1
0	0.815	0.32	0	0.595	1.44E-07	3.21E-07	0.0587	0	0.06262321	0	0	1	0	1	0	0	1	1	0	0	0	0	1
0	0.815	0.17	0	0.089	0	0	0.3568	0	0.0237183	0	0	0	1	0.16	0.06	0	1	1	0.0745098	0.07	0.07	0	0
0	0.815	0.17	0	0.277	0	0	0.3249	0	0.01761252	1	1	0	0	0.05	0.06	0	1	1	0.03529412	0.04	0.04	0.05	0
0	0.815	0.17	0	0.277	0	0	0.229	0	0.03131116	1	1	0	0	0.14	0.06	0	1	1	0.05882353	0.06	0.07	0	1
0	0.815	0.17	0	0.277	0	0	0.5263	0	0.04509979	1	1	0	0	0.09	0.06	0	1	1	0.09019608	0.09	0.05	0	1
0	0.815	0	0	0.277	0	0	0.2804	0	0.01565558	1	1	0	0	0.06	0.05	0	1	1	0.05908309	0.05	0.06	0	1
0	0.815	0.17	0	0.089	0	0	0.4012	0	0.02348337	0	0	1	1	0.06	0.06	0	1	1	0.04705882	0.05	0.07	0	1
0	0.815	0.17	0	0.595	0	0	0.3894	0	0.00587064	1	1	0	0	0.02	0.06	0	1	1	0.05908309	0.05	0.07	0	1
0	0.815	0.32	0	2.277	2.08E-07	1.72E-06	0.0399	0	0.01369884	0	0	0	0.43	0.03137255	0.85882353	1	1	1	0.08031939	0.12	0.03	0	1
0	0.815	0.05	0	0.595	2.42E-07	0	0.0039	0	0.00319389	0	0	0	0	1	0	0	0.008	0.07843	1	1	0	1	3
0	0.815	0	0	0.277	0	0	0.456	0	0.00195695	1	1	0	0	0	0.06	0	1	1	0.00392157	0	0.07	0	1

图 4 处理后的数据集

DBN 模型的权重更新方向，与模型的梯度和上一次的权重值有关，模型训练开始取动量值为 0.5，然后随着训练轮次线性增加到 0.9。并分别就五个不同结构的 DBN 模型进行入侵检测，模型迭代训练过程中的 Acc 曲线和 Loss 曲线分别如图 5~6 所示。

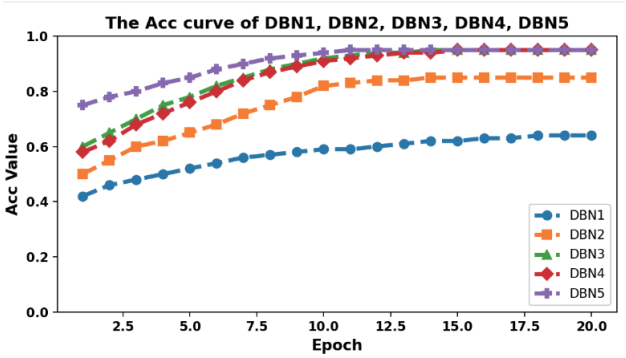

图 5 各模型的 Acc 曲线

由图 5 可知，随着模型的训练，Acc 曲线逐渐趋于平缓，在五种模型的 Acc 曲线中，DBN3 的 Acc 曲线明显在 DBN1、DBN2 的 Acc 曲线之上，相比于 DBN4、DBN5，DBN3 的 Acc 曲线略高。所以，可以得出 DBN3 模型的 Acc 曲线表现最好，模型的入侵检测性能最优。

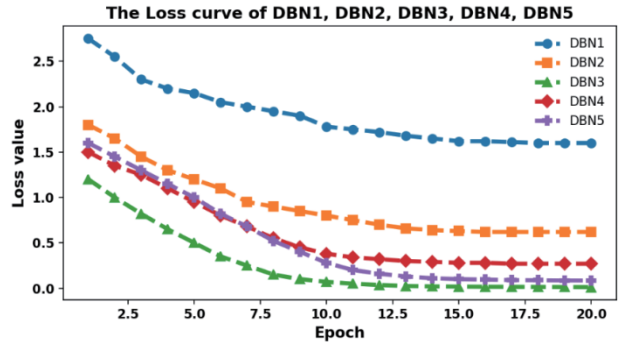

图 6 各模型 Loss 曲线

综上两方面的定性分析表明 DBN3 模型的入侵检测性能表现最优。表 5 为五种 DBN 模型进行入侵检测时在各评价指标上的具体结果。

表 5 五种 DBN 模型入侵检测结果

单位：%

模型	Acc	Fpr	Re
DBN1	59.14	13.77	61.27
DBN2	85.77	9.59	80.14
DBN3	94.26	3.85	89.78
DBN4	92.32	6.97	86.28
DBN5	93.21	7.61	85.44

由表 5 可知，DBN3 模型在 3 个评价指标方面都要优于其他的模型，进一步对 DBN3 模型在各样本类别进行入侵检测实验，结果如表 6 所示。

表 6 DBN3 模型入侵检测结果

单位：%

Label	Acc	Fpr	Re
Normal	97.66	1.63	93.24
Dos	96.11	2.04	91.32
Probe	96.84	3.77	89.21
R2L	94.15	5.29	87.39
U2R	69.15	8.31	52.47
Total	94.26	3.85	89.78

通过上述分析和实验结果证明，DBN 模型的隐含层数和隐含层神经元数确实会影响模型本身的入侵检测性能，最终通过借鉴已有研究和实验验证得出，DBN 模型的隐含层数为三层时，模型对数据的学习能力最好，入侵检测效果最佳。

6 结语

本文通过 DBN 模型进行网络入侵检测，通过模型进行特征提取，分析单个受限玻尔兹曼机对数据的学习分析能力；详细论述了 DBN 模型的建立和训练过程；分析了在 DBN 模型添加的 softmax 回归分类器对输出数据攻击类型判断过程；实验对数据集进行预处理后进行入侵检测试验，将不同深度 DBN 模型的运行结果进行对比得出结果。

参考文献：

[1] 金诗博，张立．基于对抗性双通道编码器的网络入侵检测算法 [J]. 火力与指挥控制，2024, 49(6):75-82.

[2] 邓淼磊，阚雨培，孙川川，等．基于深度学习的网络入侵检测系统综述 [J]. 计算机应用 ,2025, 45(2):453-466.

[3] 郭方洪，易新伟，徐博文，等．基于深度信念网络和迁移学习的隐匿 FDI 攻击入侵检测 [J]. 控制与决策，2022, 37(4): 913-921.

[4] 汪强龙，高晓光，吴必聪，等．受限玻尔兹曼机及其变体研究综述 [J]. 系统工程与电子技术，2024, 46(7):2323-2345.

[5] 陆鹏，肖晓强，王济瑾，等．基于散列函数加速的并行遗传算法 [J]. 计算机应用，2020,40(S1):124-127.

[6] 谭凯中，高琬晴，刘健．基于连续型贝叶斯概率估计器预测原子核质量 [J]. 核技术，2025,48(5):100-110.

【作者简介】

陈虹（1993—），女，山西晋中人，硕士，助教，研究方向：计算机网络、信息安全。

（收稿日期：2025-11-23 修回日期：2026-01-09）