第 48 卷 第 12 期 自 动 化 学 报 Vol. 48, No. 12
2022 年 12 月 ACTA AUTOMATICA SINICA December, 2022

基于天牛群优化与改进正则化极限学习机的网络入侵检测

王振东 [1] 刘尧迪 [1] 杨书新 [1] 王俊岭 [1] 李大海 [1]

摘 要 正则化极限学习机 (Regularized extreme learning machine, RELM) 因其极易于实现、训练速度快等优点在诸多领域均取得了成功应用. 对此, 本文将 RELM 引入到入侵检测中, 设计了天牛群优化算法 (Beetle swarm optimization, BSO), 并针对 RELM 由于随机初始化参数带来的潜在缺陷, 提出基于天牛群优化与改进正则化极限学习机 (BSO-IRELM) 的网络入侵检测算法. 使用 LU 分解求解 RELM 的输出权值矩阵, 进一步缩短了 RELM 的训练时间, 同时利用 BSO 对 RELM 的权值和阈值进行联合优化. 为避免 BSO 算法陷入局部最优, 引入 Tent 映射反向学习、莱维飞行的群体学习与动态变异策略提升优化性能. 实验结果表明, 在机器学习 UCI 数据集上, 相比于 RELM、IRELM、GA-IRELM、PSO-IRELM 等算法, BSO-IRELM 的数据分类性能提升明显. 最后, 将 BSO-IRELM 应用于网络入侵检测数据集 NSL-KDD, 并与 BP (Back propagation)、LR (Logistics regression)、RBF (Radial basis function)、AB (AdaBoost)、SVM (Support vector machine)、RELM、IRELM 等算法进行了对比, 结果证明 BSO-IRELM 算法在准确率、精确率、真正率和假正率等指标上均具有明显优势.

关键词 入侵检测, 正则化极限学习机, LU 分解, 天牛群优化算法

引用格式 王振东, 刘尧迪, 杨书新, 王俊岭, 李大海. 基于天牛群优化与改进正则化极限学习机的网络入侵检测. 自动化学报, 2022, **48**(12): 3024−3041

DOI 10.16383/j.aas.c190851

Network Intrusion Detection Based BSO and Improved RELM

WANG Zhen-Dong[1] LIU Yao-Di[1] YANG Shu-Xin[1] WANG Jun-Ling[1] LI Da-Hai[1]

Abstract Regularized extreme learning machine (RELM) has been successfully applied in many fields due to its advantages of easy realization and fast training speed. In this regard, RELM is introduced into intrusion detection. Potential flaws caused by random initialization parameters, and a network intrusion detection algorithm based on the beetle swarm optimization (BSO) and improved RELM (BSO-IRELM) was proposed. LU decomposition was used to solve the output of weighting matrix of RELM, which further shortened the training of RELM. At the same time, BSO algorithm was designed to jointly optimize the weights and thresholds of RELM. To prevent the BSO algorithm from falling into a local optimum, Tent mapping inverse learning, Levy flight group learning and dynamic mutation strategies were introduced to improve optimization performance. Experimental results show that compared to algorithms such as RELM, IRELM, GA-IRELM, and PSO-IRELM on machine learning UCI datasets, BSO-IRELM's data classification performance improves significantly. Finally, the BSO-IRELM was applied to the network intrusion detection dataset NSL-KDD, and compared with BP (back propagation), LR (logistics regression), RBF (radial basis function), AB (AdaBoost), SVM (support vector machine), RELM, IRELM and other algorithms. It is clear that the BSO-IRELM algorithm has obvious advantages in terms of accuracy, precision, true positive rate, and false positive rate.

Key words Intrusion detection, regularized extreme learning machine (RELM), LU decomposition, beetle swarm optimization (BSO)

Citation Wang Zhen-Dong, Liu Yao-Di, Yang Shu-Xin, Wang Jun-Ling, Li Da-Hai. Network intrusion detection based BSO and improved RELM. *Acta Automatica Sinica*, 2022, **48**(12): 3024−3041

随着网络技术的快速发展, 网络结构趋于复杂, 由此发生网络入侵的风险也越来越大, 如何辨识各种网络入侵成为人们高度关注的问题. 入侵检测 (Intrusion detection, ID) 技术作为一种能够动态监控、预防和抵御入侵行为的新型安全机制, 已经逐渐发展成为保障网络系统安全的关键技术. 然而网络规模、网络速率以及入侵类型的持续增大、增多, 使得入侵检测技术面临越来越多的挑战[1]. 因

收稿日期 2019-12-16 录用日期 2020-04-07
Manuscript received December 16, 2019; accepted April 7, 2020
国家自然科学基金 (61562037, 61562038, 61563019, 61763017), 江西省自然科学基金 (20171BAB202026, 20181BBE58018) 资助
Supported by National Natural Science Foundation of China (61562037, 61562038, 61563019, 61763017) and Natural Science Foundation of Jiangxi Province (20171BAB202026, 20181BBE58018)
本文责任编委 段书凯
Recommended by Associate Editor DUAN Shu-Kai
1. 江西理工大学信息工程学院 赣州 341000
1. School of Information Engineering, Jiangxi University of Science and Technology, Ganzhou 341000

此, 如何设计面向当前及未来网络环境的新型入侵检测机制, 提高入侵检测的检测速度、降低漏报率和误报率, 提升检测性能成为相关领域研究人员关注的核心问题.

在已有研究中, 通常认为数据挖掘、机器学习以及神经网络在内的多种方法是有效的入侵检测方法[2−5]. 但大部分数据挖掘算法对噪声较为敏感, 若数据集包含噪声数据较多, 则算法极易出现过拟合现象; 机器学习算法因其自身比较复杂, 数据集过大会导致模型训练时间过长, 计算成本较高; 神经网络通过模拟人类大脑的思维方式来处理信息, 具有自组织、自学习和自适应的特点, 将其应用于入侵检测可以很好地解决数据挖掘和机器学习存在的问题, 提升检测性能, 使得基于神经网络的入侵检测成为研究热点[6]. 传统神经网络如 BP (Back propagation) 神经网络在诸多文献中已应用于网络入侵检测, 并取得一定效果[7−8], 但需多次迭代确定网络输出权值, 严重影响网络的学习能力.

作为一种新型神经网络, 极限学习机 (Extreme learning machine, ELM)[9] 的出现引起了研究人员广泛的关注. ELM 是一种单隐层前馈型神经网络, 最大特点是输入层和隐含层之间的连接权值, 以及隐含层节点的阈值只需通过最小二乘法计算一次即可得到最优初始权值和阈值, 而不需通过反向传播算法进行更新. 因易于实现、训练速度快, ELM 在数据分类[10−11]、故障识别[12]、能耗预测[13]、入侵检测[14] 等许多领域取得了成功. 但 ELM 并未充分考虑结构化风险可能导致的过拟合问题. Deng 等[15] 提出了正则化极限学习机 (Regularized extreme learning machine, RELM) 的概念, 并将其应用于 SinC 函数的近似、现实世界回归以及 UCI 数据集的分类. 结果表明, RELM 不仅能够保留 ELM 的所有优点, 对离群点还具有一定的抗干扰能力, 能够获得极小的训练误差, 模型的泛化性能也得到显著提高.

鉴于 RELM 在分类问题中表现出的优越性能, 本文将 RELM 引入到入侵检测领域. 但直接使用 RELM 会存在如下问题: 1) RELM 的输出权值矩阵通过逆矩阵求得, 逆矩阵在求解过程中接近奇异值, 会降低算法的求解精度, 影响分类准确率; 2) 随机初始化权值矩阵以及隐含层阈值, 会导致原始数据随机映射到 RELM 特征空间时出现难以预测的非线性分布, 并对算法的分类准确率造成影响. 对此, 本文尝试使用天牛群优化 (Beetle swarm optimization, BSO) 算法优化 RELM 的初始权值矩阵, 并将 LU 分解 (LU decomposition) 引入求解 RELM 的输出权值矩阵, 提出 BSO-IRELM 算法, 将其应用于入侵检测. 实验结果表明, BSO-IRELM 算法具有优秀的数据分类能力, 能够有效实现 Normal、Probe、DoS、R2L、U2R 等各类攻击的检测.

1 BSO-IRELM 算法

RELM 随机初始化输入层和隐含层之间的权值以及隐含层节点的阈值, 通过特征映射使数据分布呈现某种非线性的几何结构, 影响分类性能. 而使用逆矩阵求解输出权值矩阵, 会导致求解过程中出现奇异矩阵的情形. 对此, 使用具有良好寻优能力的 BSO 算法对 RELM 的初始权值和阈值进行初始化, 并引入 LU 分解求解 RELM 的输出权值矩阵, 规避求解过程的奇异矩阵.

1.1 基于 LU 分解的 IRELM

由传统统计学原理可知, 实际风险包括经验风险和结构风险两种[16]. 一种具有好的泛化能力的模型应该能够平衡这两种风险. 因此增加正则项以调节系数 β, 提高模型的泛化能力. RELM 的目标函数为

$$\min \frac{1}{2}\|\beta\|^2 + \frac{\lambda}{2}\sum_{i=1}^{N}\|e_i\|^2 \tag{1}$$

$$\text{s.t.}\quad h(x_i)\beta = t_i - e_i,\quad i = 1, \cdots, N \tag{2}$$

其中, e_i 为训练误差, $\|\beta\|^2$ 和 $\|e_i\|^2$ 分别为结构风险和经验风险, λ 为惩罚因子.

根据式 (1) 和式 (2) 建立拉格朗日方程, 得

$$L(\alpha, e, \beta) = \frac{\lambda}{2}\|e\|^2 + \frac{1}{2}\|\beta\|^2 - \alpha(H\beta - T - e) \tag{3}$$

式中, $\alpha_i \in \mathrm{R}$, $i = 1, \cdots, N$ 为拉格朗日算子. 对式 (3) 的变量 (α, e, β) 分别求偏导并令其等于零, 得

$$\begin{cases} \dfrac{\partial L}{\partial \beta} = 0 \to \beta^{\mathrm{T}} = \alpha H \\ \dfrac{\partial L}{\partial e} = 0 \to \lambda e^{\mathrm{T}} + \alpha = 0 \\ \dfrac{\partial L}{\partial \alpha} = 0 \to H\beta - T - e = 0 \end{cases} \tag{4}$$

对式 (4) 进行最小二乘法计算, 得到输出权值矩阵

$$\beta = \left(\frac{I}{\lambda} + H^{\mathrm{T}}H\right)^{-1} H^{\mathrm{T}}T \tag{5}$$

其中, I 为单位矩阵.

式 (5) 计算输出权值矩阵 β 涉及到矩阵求逆的运算, 若输入样本过大, 会导致矩阵求逆复杂度增大, 从而降低 RELM 的训练效率. 为降低 RELM

的计算复杂度, 本文提出一种基于 LU 分解的 IRELM 算法, 改变 RELM 输出权值矩阵的求解方法, 降低算法复杂度, 提高入侵检测分类精确度.

由式 (5) 得

$$\left(\frac{I}{\lambda}+H^{\mathrm{T}}H\right)\beta=H^{\mathrm{T}}T \tag{6}$$

令 $A=(I/\lambda+H^{\mathrm{T}}H)$, $b=H^{\mathrm{T}}T$, 则式 (6) 转化为

$$A\beta=b \tag{7}$$

LU 分解求解 RELM 输出权值矩阵的具体步骤如下:

矩阵 A 可进行唯一的 LU 分解, 设

$$A=\begin{bmatrix}1 & & & \\ l_{21} & 1 & & \\ \vdots & \vdots & \ddots & \\ l_{n1} & l_{n2} & \cdots & 1\end{bmatrix}\begin{bmatrix}u_{11} & u_{12} & \cdots & u_{1n} \\ & u_{22} & \cdots & u_{2n} \\ & & \ddots & \vdots \\ & & & u_{nn}\end{bmatrix}$$

由矩阵乘法并令两边矩阵的 (i, j) 元素相等, 得上下三角矩阵中的元素为

$$\begin{cases} u_{ij}=a_{ij}-\sum\limits_{k=1}^{i-1}l_{ik}u_{kj}, \\ \qquad\qquad j=i, i+1, \cdots, n \\ l_{ij}=\dfrac{a_{ij}-\sum\limits_{k=1}^{j-1}l_{ik}u_{kj}}{u_{jj}}, \\ \qquad\qquad i=j+1, j+2, \cdots, n \end{cases} \tag{8}$$

当矩阵 A 进行 LU 分解后, 解线性方程组 $A\beta=b$ 等价于求解下面两个三角形方程组

$$\begin{cases} Ly=b \\ U\beta=y \end{cases}$$

求解 $Ly=b$ 的递推公式为

$$\begin{cases} y_1=b_1 \\ y_i=b_i-\sum\limits_{k=1}^{i-1}l_{ik}y_k, \quad i=2, 3, \cdots, n \end{cases} \tag{9}$$

求解 $U\beta=y$ 的递推公式为

$$x_i=\frac{y_i-\sum\limits_{k=i+1}^{n}u_{ik}x_k}{u_{ii}}, \quad i=n, n-1, \cdots, 2, 1 \tag{10}$$

从上面的求解过程可以看出, 输出权值矩阵不需使用式 (5) 逆矩阵的方法来计算, 只需通过式 (8) ~ (10) 的迭代递推公式进行简单的加减运算即可求出 RELM 的输出权值, 能够大大降低算法的复杂度. 同时, 使用 LU 分解求解输出权值矩阵可以很好地避免计算过程出现奇异矩阵的情况, 提高算法的分类准确率.

1.2 BSO 算法设计

天牛须搜索算法 (Beetle antennae search, BAS) 是 Jiang 等[17–18] 在 2017 年模拟天牛觅食原理时提出的启发式算法, 用于解决压力容器和 Himmelblau 等非线性优化问题. BAS 算法具有求解速度快和精度高的特点, 已成功应用于信号定位[19] 和数据分类[20] 等领域. 但 BAS 算法在高维空间搜索时只能收敛到局部极值, 且在多维函数优化中, 只依赖于单个天牛个体进行搜索会增加算法陷入局部最优的可能性. 对此, 本文提出了一种带莱维飞行群体学习策略与动态变异策略的混沌天牛群算法. 将天牛个体搜索扩展为群体搜索, 并使用 Tent 映射反向学习初始化种群, 促使初始群体信息分布均匀, 提高搜索效率. 此外, 利用莱维飞行群体学习策略, 使得天牛个体既能学习自身经验又可以学习群体经验, 让各天牛个体有目的和指导性地移动, 提高算法的收敛性能. 最后引入动态变异策略, 增加迭代后期种群多样性, 避免算法陷入局部最优.

1.2.1 Tent 映射反向学习初始化种群

研究表明[21], 在群智能搜索中, 算法的收敛性能会受到初始种群的影响. 种群的数量越多、分布越均匀, 算法越能够在更短的时间内收敛到最优解; 反之, 则会影响算法的收敛性能. 使用混沌映射初始化种群具有随机性、遍历性以及有界性的特点, 能够有效提高算法的搜索效率. 而 Tent 映射产生初始序列比 Logistic 映射产生初始序列更加均匀, 所以本文采用 Tent 映射对天牛群体初始化, 并使用反向学习策略优化初始种群, 通过反向个体与现有个体一起竞争, 让更优秀的个体被选进下一代学习, 可以扩大种群的搜索范围, 减少无效搜索, 从而提高算法的收敛速度. Tent 映射的数学表达式为

$$x_{n+1}=\begin{cases} 2x_n, & 0<x_n\le\dfrac{1}{2} \\ 2\left(1-x_n\right), & \dfrac{1}{2}<x_n\le 1 \end{cases} \tag{11}$$

反向解的定义为: 在 D 维空间中的一个可行解为 $\boldsymbol{x}=(x_1, x_2, \cdots, x_D)$, $\boldsymbol{x}=[a, b]$, 则其反向解为 $\boldsymbol{x}'=(x_1', x_2', \cdots, x_D')$, 其中, $x_i'=a+b-x_i$.

综上所述, 采用 Tent 映射反向学习初始化种群的具体步骤如下:

步骤 1. 在搜索空间中使用 Tent 映射产生 N 个天牛种群的位置 x_{ij}, $i=1, 2, \cdots, D; j=1, 2, \cdots, N$ 作为初始种群 OB;

步骤 2. 根据反向解的定义, 产生初始种群 OB 中的每个天牛群体 x_{ij} 的反向群体 x_{ij}' 作为反向种

群 FB;

步骤 3. 合并种群 OB 和 FB, 使用升序将这 $2N$ 个天牛群体的适应度值排序, 选取其中适应度值前 N 的天牛群体作为初始种群.

1.2.2　莱维飞行的群体学习策略

标准 BAS 算法中, 天牛个体的搜索范围有限, 搜索位置从全局最优向局部最优转移比较困难. 虽然将个体搜索改为群体搜索能够扩大种群的搜索范围, 但由于天牛个体之间没有信息交流和反馈, 会影响算法的收敛精度. 根据粒子群算法可知, 群体中个体在移动过程中, 需不断学习历史群体经验, 即个体最优应具有向历史最优移动的趋势, 这种移动趋势能够对算法收敛速度的提升起到决定性作用. 为此, 在粒子群算法框架下, 引入具有莱维飞行的指导性学习策略.

莱维分布是 20 世纪 30 年代法国数学家莱维 (Levy) 提出的一种概率分布, Mandelbrotb 对其进行了详细描述[22]. 莱维飞行作为一种服从莱维分布的随机搜索方法, 可以增加种群的多样性, 扩大搜索范围, 避免算法陷入局部最优, 可有效增强算法的寻优能力. 其中莱维分布满足

$$Levy \sim u = t^{-\theta},\quad 1 < \theta \le 3$$

莱维飞行模型较为复杂, 目前使用 Mantegna 算法进行模拟, 数学表达式如下:

步长 t 的计算公式为

$$t = \frac{\mu}{|v|^{\frac{1}{\theta}}}$$

其中, μ, v 服从正态分布

$$\mu \sim \mathrm{N}\left(0, \sigma_\mu^2\right)$$
$$v \sim \mathrm{N}\left(0, \sigma_v^2\right)$$

$$\sigma_\mu = \left[\frac{\Gamma(1+\theta) \times \sin\dfrac{\pi\theta}{2}}{\Gamma\left(\dfrac{1+\theta}{2}\right) \times \theta \times 2^{\frac{\theta-1}{2}}}\right]^{\frac{1}{\theta}}$$

其中, Γ 是标准的伽马分布, 为节约计算时间取 $\theta = 1.5$.

图 1 是莱维飞行的轨迹示意图. 由于莱维飞行是二阶矩发散的, 所以其在运动过程中的跳跃性很大.

指导性学习策略中天牛朝向的公式通过莱维飞行策略进行更新:

$$\begin{aligned} d(t+1) = {} & \omega \times d(t) + C_1 \times Levy(\theta) \times \\ & (gbest(t) - X(t)) + C_2 \times Levy(\theta) \times \\ & (zbest - X(t)) \end{aligned} \tag{12}$$

最终个体位置更新式为

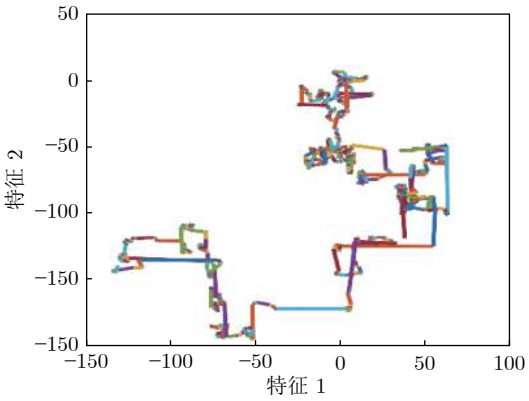


图 1　莱维飞行轨迹示意图

Fig.1　Levy flight path diagram

$$\begin{aligned} X(t+1) = {} & X(t) + k_1 \times step \times d(t) \times \\ & \mathrm{sign}\left(f(X_r(t)) - f(X_l(t))\right) + k_2 d(t) \end{aligned} \tag{13}$$

其中, $d(t)$ 表示第 t 代天牛的朝向, $X(t)$ 表示第 t 代天牛的位置, $gbest(t)$ 表示第 t 代天牛的个体极值, $zbest$ 表示迄今为止的全局极值, ω 为惯性权重, C_1, C_2 为学习因子, $Levy(\theta)$ 为莱维随机数. $f(X_l(t))$ 表示第 t 代天牛左须的适应度函数值, $f(X_r(t))$ 表示第 t 代天牛右须的适应度函数值, $step$ 为天牛步长, k_1, k_2 为比例系数, sign 为符号函数. 天牛朝向公式中, 第 2 部分是自学习部分, 表示天牛个体对自身历史的记忆, 有向自身最优位置移动的趋势; 第 3 部分为社会学习部分, 表示天牛个体之间的学习以及群体的历史经验, 有向群体最优位置移动的趋势.

1.2.3　动态变异策略

天牛群算法在迭代后期种群的多样性会越来越低, 使算法的搜索能力下降. 为避免迭代后期出现早熟现象, 引入动态变异策略, 增加天牛种群在迭代后期的多样性, 提高算法的收敛精度. 目前, 相关学者提出了多种变异算法, 典型的变异算法有高斯变异 (Gaussian mutation)[23] 和柯西变异 (Cauchy mutation)[24]. 柯西算子相比于高斯算子具有较长的两翼, 可以产生大范围的随机数, 使算法有更大的机会跳出局部最优, 同时, 当峰值较低时柯西变异只需要花费更少的时间来搜索附近区域. 柯西变异概率分布如图 2 所示.

因此, 选择柯西变异对天牛群体进行二次寻优, 对 X 进行变异操作, 即

$$\begin{aligned} & X^*(t) = X(t) + \eta * C(0, 1) \\ & \eta = \mathrm{e}^{-\lambda \frac{t}{T}} \end{aligned} \tag{14}$$

其中, η 是变异权重, 其值随着迭代次数的增加而减小, T 为最大迭代次数, $\lambda = 10$ 为常数, $C(0, 1)$ 是比例参数为 1 的柯西算子产生的一个随机数.

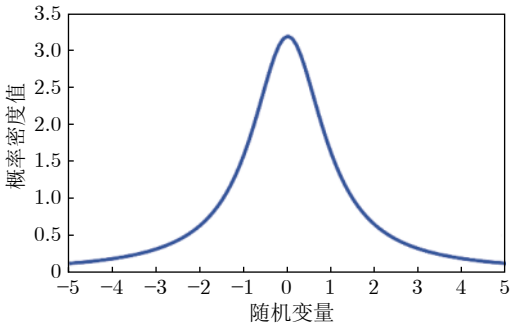


图 2 柯西变异概率分布图

Fig. 2 Cauchy variation probability distribution map

1.2.4 天牛群算法步骤

天牛群算法的步骤如下, 具体流程图如图 3 所示.

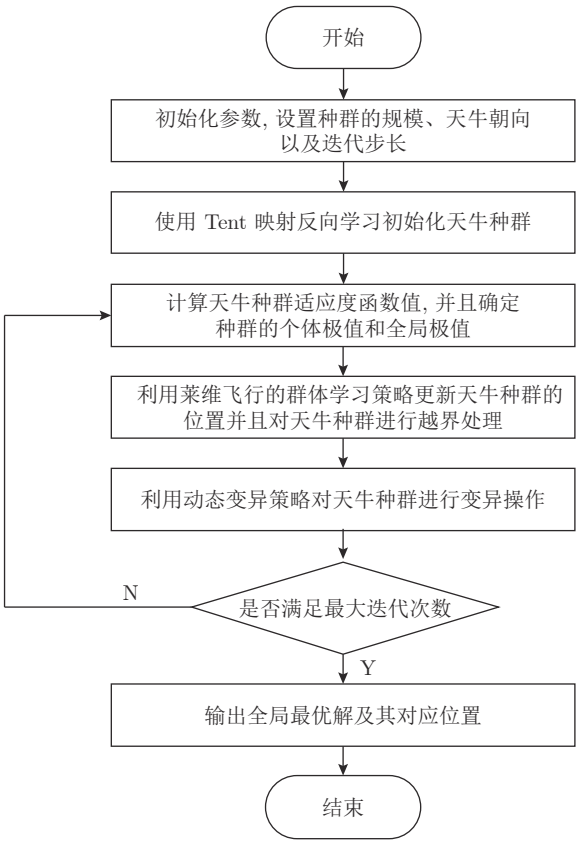


图 3 天牛群算法流程图

Fig. 3 Flow chart of BSO algorithm

步骤 1. 初始化天牛群算法参数: 设置天牛规模、迭代步长、最大迭代次数, 使用 Tent 映射反向学习策略初始化天牛群体, 初始化天牛朝向.

步骤 2. 计算群体中天牛个体相应的适应度函数值, 根据适应度函数值确定种群的个体极值和全局极值.

步骤 3. 利用式 (12) 和式 (13) 更新天牛个体的朝向以及位置, 对天牛种群进行越界处理.

步骤 4. 利用式 (14) 对天牛种群进行变异操作.

步骤 5. 判断算法是否满足迭代终止条件, 若满足则输出全局最优解及其对应的位置, 否则返回步骤 2.

1.2.5 天牛群算法伪代码

天牛群算法伪代码如下所示:

Algorithm 1. BSO algorithm

Input: population size N, step size $Step$, maximum number of iteration T, dimension D, inertia weight W, learning factor C_1, C_2, ratio k_1, k_2, constant λ, the distance between the two whisks and the center of mass l

Output: best BSO $zbest$

1. Initialize the BSO population X with Tent map reverse-learning strategy, initialize the BSO toward d with random solutions, initialize Levy flight factor $Levy(\theta)$
2. Calculate fitness of X, find global best BSO as $zbest$, find local best BSO as $gbest$
3. **for** $i = 1$ to T **do**
4. $\quad d = d/norm(d)$
5. $\quad \eta = \mathrm{e}^{-\lambda \frac{i}{T}}$
6. $\quad$ **for** $j = 1$ to N **do**
7. $\quad\quad X_l(j) = X(j) + d(j) \times l$
8. $\quad\quad$ Calculate fitness of X_l
9. $\quad\quad X_r(j) = X(j) - d(j) \times l$
10. $\quad\quad$ Calculate fitness of X_r
11. $\quad\quad$ BSO toward: $d(t+1) = \omega \times d(t) + C_1 \times Levy(\theta) \times (gbest(t) - X(t)) + C_2 \times Levy(\theta) \times (zbest - X(t))$
12. $\quad\quad$ population: $X(t+1) = X(t) + k_1 \times step \times d(t) \times \mathrm{sign}(f(X_r(t)) - f(X_l(t))) + k_2 d(t)$
13. $\quad\quad$ mutation: $X^*(t) = X(t) + \eta \times C(0, 1)$
14. $\quad$ **end for**
15. $\quad$ Calculate fitness of new BSO, if new BSO are better, update it in the population $zbest$ and $gbest$
16. **end for**

1.2.6 天牛群算法性能分析

为验证 BSO 算法的性能, 选取 $F(x): \min f(x) = \sum_{i=1}^{n} x_i^2$ 单峰函数对算法进行函数寻优以及收敛性测试, $F(x)$ 搜索范围设置为$[-10, 10]$, 并且与 GA (Genetic algorithm)、PSO (Particle swam optimization)、DE (Differemtial evolution) 和 BAS 算法进行对比, 算法迭代次数设置为 5 000 次并运行 20 次. 图 4 描述了 5 种算法在函数 $F(x)$ 上的测试结果. 由图 4 可知, 相比于 GA、PSO、DE、BAS 四种优化算法, BSO 的收敛速度和收敛精度都有明显

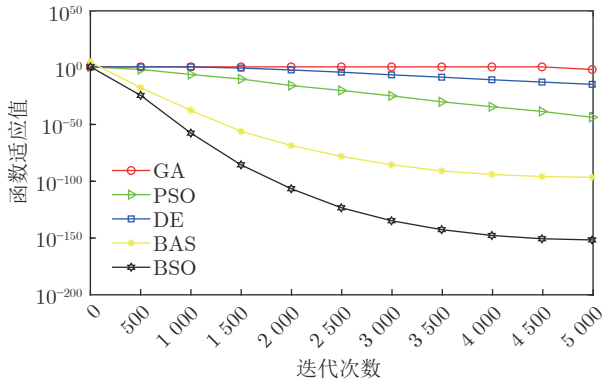


图 4　算法测试结果图

Fig. 4　Test result graph of algorithm

优势. 一方面引入莱维飞行的群体学习策略, 平衡算法的全局搜索和局部搜索能力, 加快算法的收敛; 另一方面加入动态变异策略, 帮助算法跳出局部最优, 使寻优速率加快. 故相比于传统算法, BSO 算法的性能具有明显的提升.

2　BSO-IRELM 的入侵检测算法

基于上述分析和推导, 将 LU 分解和 BSO 算法引入 RELM 算法, 并建立基于 BSO-IRELM 的入侵检测模型.

2.1　适应度函数

适应度函数是判断个体适应环境能力大小的标准. 因此, 适应度函数的选择直接影响算法的收敛精度以及能否找到最优解. 文献 [25] 将入侵检测准确率、误报率以及特征数的比例权重作为适应度函数, 既增加了计算量又会因计数不当导致收敛精度较低; 文献 [26] 将群智能算法函数值的分段函数作为适应度函数, 需通过函数值确定适应度函数表达式, 增加了计算复杂度.

将入侵检测误差和函数作为适应度函数, 预测结果可由神经网络直接得到, 计算方便, 无需反复确定适应度函数表达式, 也不会因计数不当造成误差. 其数学表达式为

$$f = \sum_{k=1}^{M} |y_k - y'_k| \tag{15}$$

其中, y_k 表示网络的实际输出, y'_k 表示网络的训练输出, M 表示输入神经元的个数.

2.2　算法模型描述

使用 BSO 优化 IRELM 的基本思路是求出适应度函数最好的一组天牛位置, 在迭代结束时把该位置作为 IRELM 的最优初始权值和阈值建立入侵检测模型, 模型描述如图 5 所示.

入侵检测框架的步骤如下:

步骤 1. 对原始的 NSL-KDD 数据集进行预处理. 预处理过程包括 2 个子步骤:

步骤 1.1. 高维数据特征映射. 本文使用高维特征映射, 将离散型特征转化为数字型特征.

步骤 1.2. 数据归一化. 由于同种属性的数据之间差异较大, 影响神经网络的训练, 因此将数据归一化为 [−1, 1] 的实数.

步骤 2. 标准数据集划分. 将标准数据集划分为训练集和测试集.

步骤 3. 模型训练. 对 IRLEM 进行训练和参数调优. 本过程包括 4 个子步骤:

步骤 3.1. 初始化 IRELM 模型参数. $innum$ 个输入层节点、$midnum$ 个隐藏层节点和 $outnum$ 个输出层节点以及网络初始权值和阈值.

步骤 3.2. 初始化天牛群体. 天牛种群大小 N. 所求问题维度 $D = (innum + 1) \times midnum(midnum + 1) \times outnum$ 和最大迭代次数 T 及天牛种群位置 x_i.

步骤 3.3. 根据训练样本和适应度函数计算天牛群适应度函数值, 对适应度函数值升序排列并寻找天牛群的最优位置和最优适应度函数值. 若满足迭代终止条件, 则迭代结束转到步骤 3.4; 否则转到步骤 3.3.

步骤 3.4. 输出 BSO 全局最优位置对应的权值和阈值, 即 IRELM 的最优初始权值和阈值.

步骤 4. 将测试数据输入到训练好的 BSO-IRELM 入侵检测模型中, 进而得到每条数据的分类结果.

2.3　BSO-IRELM 伪代码

BSO-IRELM 伪代码如下所示:

Algorithm 2. BSO-IRELM intrusion detection algorithm

Module 1: dataset preprocessing

1. High dimensional data feature mapping
2. Data normalization
3. Divide training dataset and testing dataset

Module 2: population preprocessing

4. Set the parameters, initialize the position and orientation of the BSO

Module 3: BSO-IRELM intrusion detection

5. Initialize the weights and thresholds of IRELM
6. **if** the maximum number of iterations is not reached
7. According to the training dataset, the fitness function value of BSO were calculated, the global optimum and it corresponding position
8. Update the position of the BSO, transgress and mutate

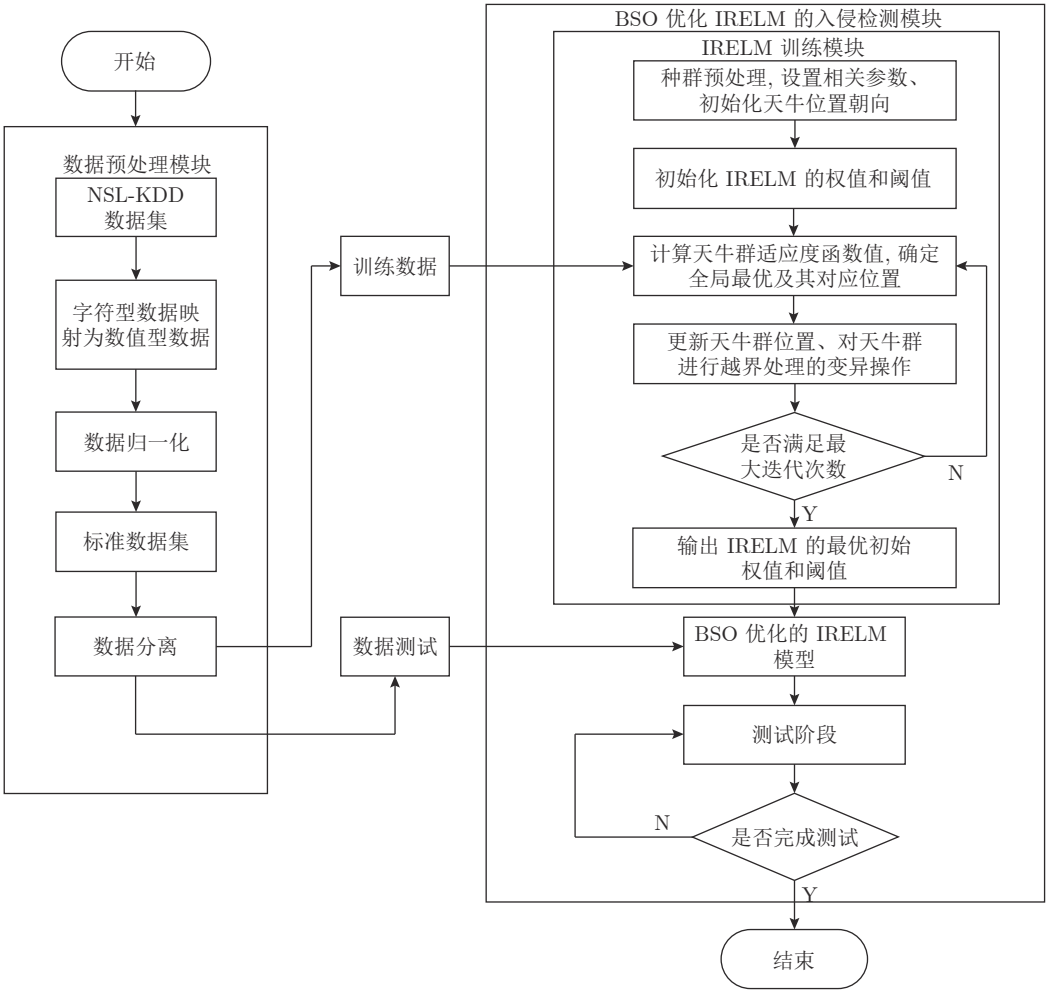


图 5 BSO-IRELM 算法入侵检测框架图

Fig. 5 BSO-IRELM Algorithm intrusion detection framework

the BSO

9. **else**

10. Output the optimal initial weights and thresholds

11. **end**

12. Establish BSO-IRELM model

13. **if** testing phase is not complete

14. Enter the testing dataset for testing

15. **else**

16. Complete the testing

17. **end**

18. Output classification result

2.4 算法复杂度分析

2.4.1 LU 分解复杂度

线性方程组的求解方法较多, 直接使用矩阵求逆计算不仅对硬件资源的占有量有影响, 还影响权值的更新速度. 因此, 对于硬件平台来说, 由于物理资源有限, 需要找到一种低能耗且快速的求解方法. 矩阵求逆的计算步骤繁多, 运算量大, 时间复杂度高, 所以在使用硬件实现时考虑采用 LU 分解法. 为进一步说明 LU 分解的优势, 对矩阵求逆和 LU 分解作如下分析: 对于 n 阶的方阵, 矩阵求逆的算法复杂度为 $\mathrm{O}(n^3)$, LU 分解的算法复杂度为 $\mathrm{O}(2/3\times n^3)$, 虽然二者数量级相同, 但因系数存在差异, 因此在运行时间上存在显著差异. 为更加直观地说明两种算法的复杂度, 对一个 10 阶矩阵进行实验, 结果表明, 矩阵求逆的运行时间为 0.4491 s, LU 分解的运行时间为 0.0479 s, 运行时间大大缩短. 本文后续使用数据集的维度远远大于 10 阶, 故使用 LU 分解的优势将更加明显.

2.4.2 BSO 算法复杂度

假设算法最大迭代次数为 $t_{\max}$, 维度为 D. 在 BAS 算法中, 初始化天牛个体, 其算法复杂度为 $\mathrm{O}(D)$. 在每一次迭代中需要完成以下步骤, 先计算

天牛个体的适应度值并找出当前最优位置, 其时间复杂度为 $O(1)$, 之后天牛个体更新位置, 其时间复杂度为 $O(1)$, 故一次迭代的算法复杂度为 $O(1+1)$. 总的时间复杂度为 $O(t_{\max}(1+1)+D)$, 即为 $O(t_{\max})$. 相比于 BAS, BSO 使用群体, 假设种群规模为 N, 初始化种群, 其算法复杂度为 $O(ND)$. 在每一次迭代中需要完成以下步骤, 先计算天牛群的适应度值并找出种群中个体最优和全局最优, 其时间复杂度为 $O(N)$, 之后天牛群位置更新, 其时间复杂度为 $O(N)$, 故一次迭代的算法复杂度为 $O(N+N)$. 总的时间复杂度为 $O(t_{\max}(N+N)+ND)$, 即为 $O(t_{\max}N)$. 故 BSO 总的时间复杂度大于 BAS, 但 BSO 克服了 BAS 极易陷入局部最优以及搜索范围有限的缺点, 提高了 BSO 的收敛精度.

2.4.3　BSO-IRELM 算法复杂度

假设算法最大迭代次数为 $t_{\max}$, 种群规模为 N, 所求问题维度 $D=(innum+1)\times midnum+(midnum+1)\times outnum$. 在 BSO-RELM 模型中, BSO 使用 Tent 映射反向学习初始化种群, 其算法复杂度为 $O(ND)$. 计算种群的适应度函数值并找出种群中个体最优和全局最优, 其时间复杂度为 $O(N)$. 在每一次迭代中需要完成以下步骤, 根据训练集通过矩阵求逆计算天牛群的适应度函数值, 其时间复杂度为 $O(midnum^3\times N)$, 之后找出种群的个体最优和全局最优, 并对天牛群位置更新, 其时间复杂度为 $O(N)$, 故一次迭代的算法复杂度为 $O(N+midnum^3\times N)$. 最后, 使用测试集对 BSO-RELM模型进行性能测试, 其时间复杂度为 $O(midnum^3)$. 总的时间复杂度为 $O(t_{\max}(N+midnum^3\times N)+ND+N+midnum^3)$. 相比于 BSO-RELM 模型, BSO-IRELM 中通过 LU 分解求解天牛群适应度函数值, 其时间复杂度为 $O(2/3\times midnum^3\times N)$, 总的时间复杂度为 $O(t_{\max}(N+2/3\times midnum^3\times N)+ND+N+2/3\times midnum^3)$. 故 BSO-IRELM 的时间复杂度远小于 BSO-RELM, 且其检测精度远优于 BSO-RELM.

3　实验结果

3.1　实验参数设置

模型中参数设置如下.

1) 群智能算法参数[27] 如表 1 所示.

表 1　群智能算法参数

Table 1　Swarm intelligence algorithm parameters

参数	值	参数	值
自学习因子	2.4	群体学习因子	1.6
交叉概率	0.7	遗传概率	0.5
惯性权重	0.8	种群数量	50
比例系数	0.4, 0.6	惩罚因子	0.5

2) RELM 神经网络模型参数为: 100-50-1, 迭代次数为 100.

RELM 模型参数通过 GA-RELM 二分类实验确定. 迭代次数设置为 100, 隐含层单元分别为 20, 30, 40, 50, 60. 实验结果表明, 包含 50 个隐含层单元的模型具有最优的检测准确率. 当把隐含层的个数从 50 增加到 60 时, 入侵检测的准确率有所下降. 保持隐含层单元数量不变, 增加迭代次数, 模型的检测性能也只会由于过度拟合而产生波动.

3.2　实验数据集

模型中使用到的数据集如下.

1) UCI 数据集[28] 如表 2 所示.

表 2　UCI 数据集

Table 2　UCI dataset

数据集	维度	类别数	样本总数	测试样本数
Iris	4	3	150	30
Wine	13	3	178	30

2) 入侵检测数据集[29]: NSL-KDD 数据集, 训练样本 9 850 条数据, 测试样本 2 000 条数据.

3.3　UCI 数据集实验结果

为了验证算法的有效性, 在 UCI 数据集上将 BSO-IRELM 与 IRELM、GA-IRELM、PSO-IRELM、BSO-RELM 以及传统 RELM 进行对比.

在表 3 ~ 5 中列出了各算法的性能评价指标, 并在图 6 中给出了部分算法的预测结果. 从图 6 可以直观地看出, BSO-IRELM 算法具有较好的预测结果, 在 Iris 数据集中, BSO-IRELM 的真实值和预测值完全重合, 可以实现完美分类; 在 Wine 数据集中, BSO-IRELM 仅存在一个错误分类. 从表 3 ~ 5 中可以看出, 各算法的性能评价指标均较优, 这

表 3　各算法在 UCI 数据集上的准确率 (%)

Table 3　Accuracy of each algorithm on UCI dataset (%)

数据集	RELM	IRELM	GA-IRELM	PSO-IRELM	BSO-RELM	BSO-IRELM
Iris	76.6667 (23/30)	90 (27/30)	96.6667 (29/30)	100 (30/30)	100 (30/30)	100 (30/30)
Wine	80 (24/30)	90 (27/30)	93.3333 (28/30)	93.3333 (28/30)	93.3333 (28/30)	96.6666 (29/30)

表 4 各算法在 Iris 数据集上的性能评价指标

Table 4 Performance evaluation index of each algorithm on Iris dataset

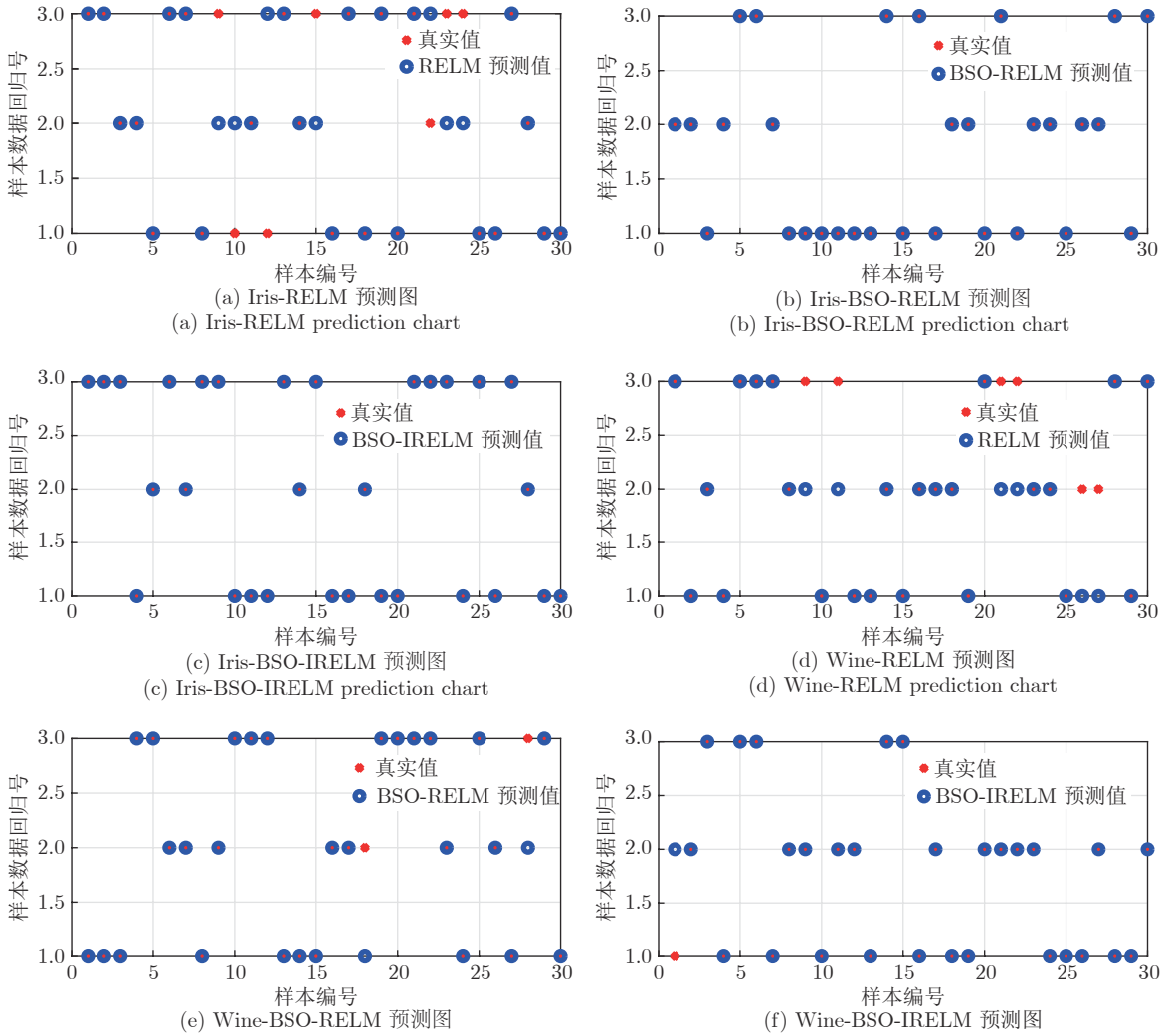
算法	类别	精确率 (%)	TPR (%)	FPR (%)	F值 (%)	AUC
RELM	1	100 (9/9)	81.8182 (9/1)	0 (0/14)	90	0.9091
	2	50 (5/10)	83.3333 (5/6)	21.7391 (5/23)	62.5	0.8125
	3	81.8182 (9/11)	69.2308 (9/13)	12.5 (2/16)	75	0.7873
IRELM	1	100 (8/8)	100 (8/8)	0 (0/19)	100	1
	2	75 (9/12)	100 (9/9)	14.2857 (3/21)	85.7143	0.9286
	3	100 (10/10)	76.9231 (10/13)	0 (0/17)	86.9565	0.8846
GA-IRELM	1	100 (13/13)	100 (13/13)	0 (0/16)	100	1
	2	100 (7/7)	87.5 (7/8)	0 (0/22)	93.3333	0.9375
	3	90 (9/10)	100 (9/9)	4.7619 (1/21)	94.7368	0.9762
PSO-IRELM	1	100 (8/8)	100 (8/8)	0 (0/22)	100	1
	2	100 (12/12)	100 (12/12)	0 (0/28)	100	1
	3	100 (10/10)	100 (10/10)	0 (0/20)	100	1
BSO-RELM	1	100 (13/13)	100 (13/13)	0 (0/17)	100	1
	2	100 (10/10)	100 (10/10)	0 (0/20)	100	1
	3	100 (7/7)	100 (7/7)	0 (0/23)	100	1
BSO-IRELM	1	100 (12/12)	100 (12/12)	0 (0/18)	100	1
	2	100 (5/5)	100 (5/5)	0 (0/25)	100	1
	3	100 (13/13)	100 (13/13)	0 (0/17)	100	1

表 5 各算法在 Wine 数据集上的性能评价指标

Table 5 Performance evaluation index of each algorithm on Wine dataset

算法	类别	精确率 (%)	TPR (%)	FPR (%)	F值 (%)	AUC
RELM	1	81.8182 (9/11)	100 (9/9)	11. 7647 (2/17)	90	0.9524
	2	66.6667 (8/12)	80 (8/10)	20 (4/20)	72.7273	0.8000
	3	100 (7/7)	63.6364 (7/11)	0 (0/17)	77.7778	0.8182
IRELM	1	88.8889 (8/9)	88.8889 (8/9)	5 (1/20)	88.8889	0.9206
	2	90.9091 (10/11)	83.3333 (10/12)	5.5556 (1/18)	86.9565	0.8889
	3	90 (9/10)	100 (9/9)	5.2632 (1/19)	94.7368	0.9762
GA-IRELM	1	87.5 (7/8)	100 (7/7)	4. 5455 (1/22)	93. 3333	0.9783
	2	100 (16/16)	88. 8889 (16/18)	0 (0/12)	94.1176	0.9444
	3	83.3333 (5/6)	100 (5/5)	4.1667 (1/24)	90.9091	0.9800
PSO-IRELM	1	90 (9/10)	100 (9/9)	5 (1/20)	94.7368	0.9762
	2	90. 9091 (10/11)	90.9091 (10/11)	5.2632 (1/19)	90.9091	0.9282
	3	100 (9/9)	90 (9/10)	0 (0/19)	94.7368	0.9500
BSO-RELM	1	90.9091 (10/11)	100 (10/10)	5.2632 (1/19)	95.2381	0.9750
	2	87.5 (7/8)	87.5 (7/8)	4.5455 (1/22)	87.5	0.9148
	3	100 (11/11)	91.6667 (11/12)	0 (0/17)	95.6522	0.9583
BSO-IRELM	1	100 (12/12)	92.3077 (12/13)	0 (0/17)	96	0.9615
	2	92.3077 (12/13)	100 (12/12)	5.5556 (1/18)	96	0.9722
	3	100 (5/5)	100 (5/5)	0 (0/24)	100	1

与本文估计基本一致. 因为两个数据集的大小相当, 且是平衡数据集, 所以检测结果较理想. 相比于 PSO-IRELM 和 GA-IRELM, BSO-IRELM 的性能评价指标均有所提高, 这充分说明 BSO 算法较 PSO 算法和 GA 算法具有更优的寻优能力. 此外, 在所有测试条件下, BSO-IRELM 的各项性能也优于 BSO-RELM、IRELM 和传统的 RELM 算法, 这说明了引入 LU 分解法以及 BSO 算法的必要性, 同时



(a) Iris-RELM 预测图
(a) Iris-RELM prediction chart

(b) Iris-BSO-RELM 预测图
(b) Iris-BSO-RELM prediction chart

(c) Iris-BSO-IRELM 预测图
(c) Iris-BSO-IRELM prediction chart

(d) Wine-RELM 预测图
(d) Wine-RELM prediction chart

(e) Wine-BSO-RELM 预测图
(e) Wine-BSO-RELM prediction chart

(f) Wine-BSO-IRELM 预测图
(f) Wine-BSO-IRELM prediction

图 6　部分算法在 UCI 数据集上的检测结果

Fig. 6　The detection results of part algorithm on UCI dataset

也验证了之前关于 RELM 存在潜在问题的假设. 从而验证了 BSO-IRELM 算法具有较优的分类性能, 因此, 进一步将 BSO-IRELM 算法应用到网络入侵检测中验证它的可行性.

3.4　NSL-KDD 数据集 2 元分类实验结果

将 4 类攻击合并为 Abnormal (非正常), 标记为 2, 正常数据 (Normal) 标记为 1, 实验转化为 2 元分类问题. 表 6 和表 7 给出了各算法的实验结果. 图 7 和图 8 分别给出了 2 元分类混淆矩阵和 ROC (Receiver operating characteristic) 曲线对比图. 从表中可以看出, BSO-IRELM 在检测正常数据和攻击类型数据方面效果较好. 由于 2 元分类数据集是非平衡数据集, 2 种数据数量相差较大, 所以准确率无法反映非平衡数据集的真实情况, 故除准确率外, 还从精确率、真正率 (True positive rate, TPR)、假正率 (False positive rate, FPR)、F 值和 AUC (Area under curve) 对 2 元分类进行评价, 上述指标的计

表 6　各算法的准确率 (%)

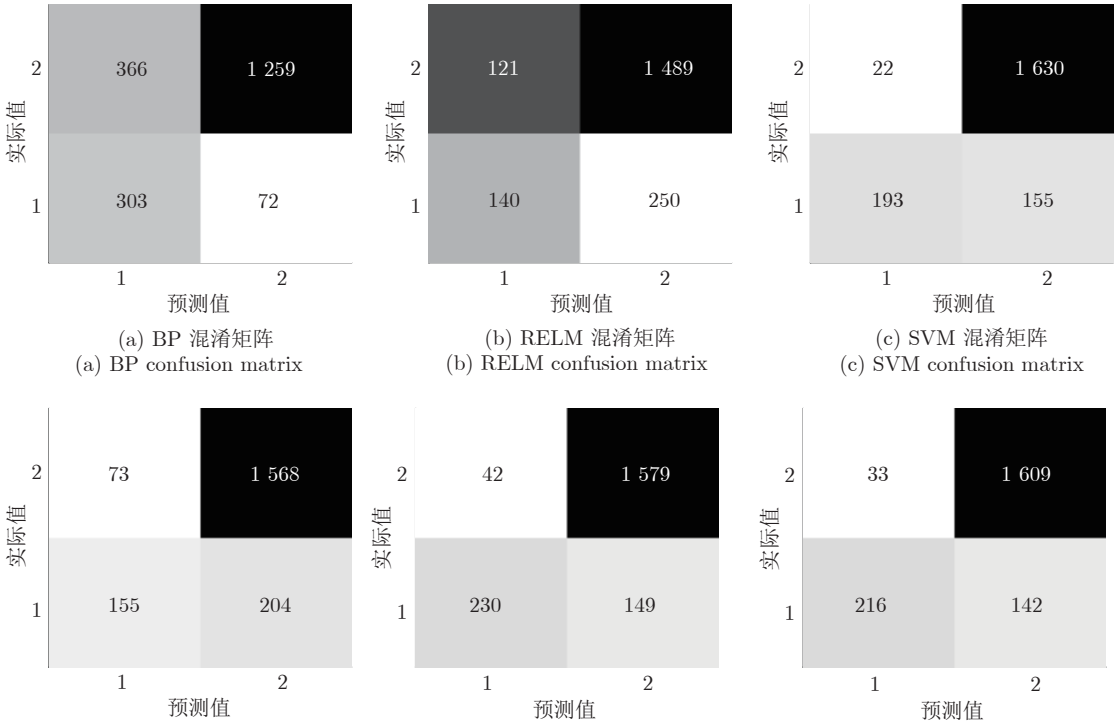
Table 6　Accuracy of each algorithm (%)

算法	准确率
BP	78.1 (1562/2000)
LR	81.3 (1626/2000)
RBF	88.9 (1778/2000)
AB	86.15 (1723/2000)
SVM	91.15 (1823/2000)
RELM	81.45 (1629/2000)
IRELM	83.9 (1678/2000)
GA-IRELM	89.5 (1790/2000)
PSO-IRELM	90.45 (1809/2000)
BSO-IRELM	91.25 (1825/2000)

表 7 各算法的性能评价指标

Table 7 Performance evaluation index of each algorithm

算法	类别	精确率 (%)	TPR (%)	FPR (%)	F值 (%)	AUC
BP	1	45.2915 (303/669)	80.8 (303/375)	22.5231 (366/1625)	58.046	0.7914
	2	94.5905 (1259/1331)	77.4769 (1259/1625)	19.2 (72/375)	85.1827	0.7914
LR	1	85.7143 (6/7)	1.5831 (6/379)	0.06169 (1/1621)	3.1088	0.5076
	2	81.2845 (1620/1993)	99.9383 (1620/1621)	98.4169 (373/379)	89.6514	0.5076
RBF	1	67.8663 (264/389)	73.1302 (264/361)	7.6266 (125/1639)	70.4	0.8275
	2	93.9789 (1514/1611)	92.3734 (1514/1639)	26.8698 (97/361)	93.1692	0.8275
AB	1	67.9825 (155/228)	43.1755 (155/359)	4.4485 (73/1641)	52.8109	0.6936
	2	88.4876 (1568/1772)	95.5515 (1568/1641)	56.8245 (204/359)	91.884	0.6936
SVM	1	89.7674 (193/215)	55.4598 (193/348)	1.3317 (22/1652)	68.5613	0.7706
	2	91.3165 (1630/1785)	98.6683 (1630/1652)	44.5402 (155/348)	94.8502	0.7706
RELM	1	53.6339 (140/261)	35.8974 (140/390)	7.5155 (121/1610)	43.0088	0.6711
	2	85.6239 (1489/1739)	92.4844 (1489/1610)	64.1025 (250/390)	88.9220	0.7076
IRELM	1	55.5556 (145/261)	41.3105 (145/351)	7.0346 (116/1649)	47.3856	0.6714
	2	88.1541 (1533/1739)	92.9654 (1533/1649)	58.6894 (206/351)	90.4958	0.7280
GA-IRELM	1	88.8412 (207/233)	52.9412 (20/391)	1.6159 (26/1609)	66.3462	0.7566
	2	89.5868 (1583/1767)	98.3840 (1583/1609)	47.0588 (184/391)	93.7792	0.7955
PSO-IRELM	1	84.5588 (230/272)	60.686 (230/379)	2.5910 (42/1621)	70.6605	0.7905
	2	91.3773 (1579/1728)	97.4090 (1579/1621)	39.3139 (149/379)	94.2967	0.8066
BSO-IRELM	1	86.747 (216/249)	60.3352 (216/358)	2.0097 (33/1642)	71.1697	0.7916
	2	91.9428 (1609/1751)	97.9902 (1609/1642)	39.6648 (142/358)	94.8702	0.8416



(a) BP 混淆矩阵
(a) BP confusion matrix

(b) RELM 混淆矩阵
(b) RELM confusion matrix

(c) SVM 混淆矩阵
(c) SVM confusion matrix

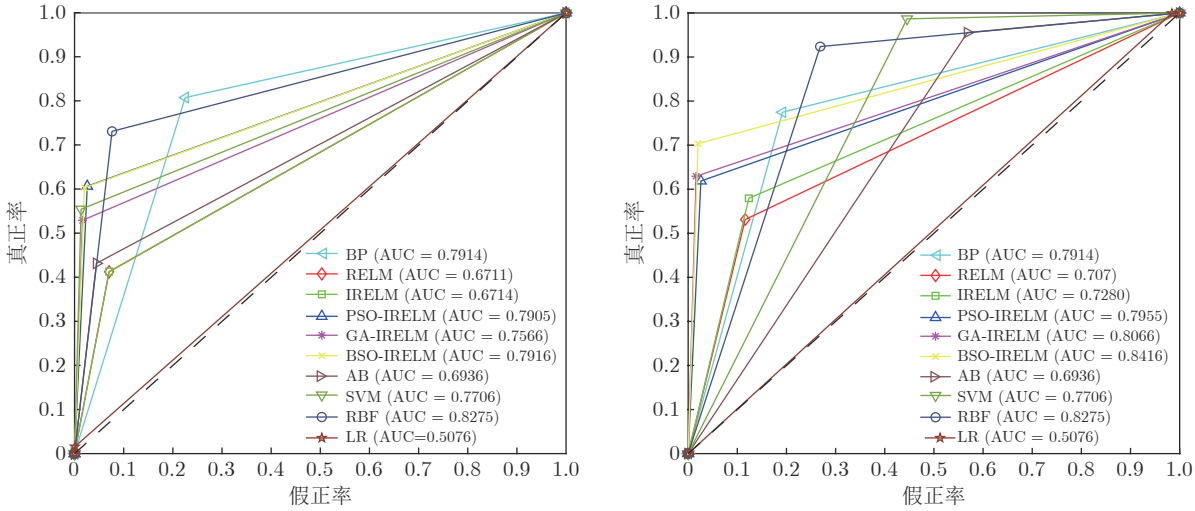
(d) AdaBoost 混淆矩阵
(d) AdaBoost confusion matrix

(e) PSO-IRELM 混淆矩阵
(e) PSO-IRELM confusion matrix

(f) BSO-IRELM 混淆矩阵
(f) BSO-IRELM confusion matrix

图 7 2 元分类混淆矩阵

Fig. 7 Binary classification confusion matrix



(a) 正常样本 ROC 曲线图
(a) Normal sample ROC curves

(b) 攻击样本 ROC 曲线图
(b) Attack sample ROC curves

图 8　2 元分类 ROC 曲线对比图

Fig. 8　Binary classification of ROC curve comparison diagram

算方法参照文献 [30]. 在大多数测试条件下, BSO-IRELM 的性能优于 BP、LR (Logistics regression)、RBF (Radial basis function)、AB (AdaBoost), 可以取得与 PSO-IRELM、GA-IRELM 和 SVM (Support vector machine) 相近的分类性能, 总体上优于 PSO-IRELM、GA-IRELM 和 SVM. 且性能远优于 IRELM 和传统 RELM. 特别地, BSO-IRELM 的 F 值和 AUC 均最优, 但在精确率方面, 比 BP 差 2.6477%, 在真正率方面, 比 LR 差 1.94814%, 而在假正率方面, 比 PSO-IRELM 差 0.3509%. 由于测试集是由随机选取的 2 000 条数据组成, BSO-IRELM 中攻击类型数据所占比重较大, 故精确率、真正率和假正率略差. 混淆矩阵将数据集中的记录按照实际结果和预测结果进行汇总, 实现可视化. 从图 7 可以看出, BSO-IRELM的分类模型最准确, 因为此时一、三象限对应位置的数值最大, 而二、四象限对应位置的数值最小, 且 BP 用于入侵检测的能力最差, 相比于各算法, BP模型将大量攻击类型数据预测为正常类型, 会给网络安全带来很大的威胁. 此时 BP 神经网络可能因为训练过程中, 权值收敛到局部极小点导致网络训练失败. ROC 曲线图是反映真正率和假正率之间关系的曲线. 曲线将整个图划分为两个部分, 曲线下部分的面积即为 AUC, 用来表示预测准确性, AUC 越高, 说明预测准确率越高. 从图 8 可以看出, 在各算法中, 无论检测正常数据还是攻击类型数据, BSO-IRELM 的 AUC 均较优, 但相比于 RBF 的正常数据检测差0.0359. 在大多数情况下, BP、SVM 的 AUC 优于 IRELM 和传统 RELM, 但相比于 PSO-IRELM、GA-IRELM 以及 BSO-IRELM, AUC 均较差. 这充分说明, RELM 潜在的参数问题对于分类性能的影响, 突出了引入 LU 分解法以及 BSO 算法的必要性, 验证了 BSO-IRELM 相比于 BP 和传统 RELM对于 2 元分类的入侵检测具有较好的检测性能.

3.5　NSL-KDD 数据集多元分类实验结果

Normal、Probe、DoS、R2L、U2R 各为一类, 分别记为 1, 2, 3, 4, 5, 实验变成多分类. 在表 8 ~ 13 中列出了对比结果, 并在图 9 中给出了多元分类混淆矩阵, 在图 10 中给出了多元分类 ROC 曲线图.

从表 8 可以看出, BSO-IRELM 的准确率最高, 为 88.7%, 其他算法, 如 BP、LR、RBF、AB、SVM、RELM、IRELM、GA-IRELM 和 PSO-IRELM 的准确率分别为 73.1%, 47.2%, 81.95%, 76.05%,

表 8　不同算法检测准确率 (%)

Table 8　Accuracy of different algorithms (%)

算法	准确率
BP	73.1 (1 462/2 000)
LR	47.2 (944/2 000)
RBF	81.95 (1 639/2 000)
AB	76.05 (1 521/2 000)
SVM	83.15 (1 663/2 000)
RELM	62.7 (1 254/2 000)
IRELM	71.9 (1 438/2 000)
GA-IRELM	86.35 (1 727/2 000)
PSO-IRELM	86.15 (1 723/2 000)
BSO-IRELM	88.7 (1 774/2 000)

表 9 各算法在 Normal 上的性能评价指标

Table 9 Performance evaluation index of each algorithm on Normal

算法	精准率 (%)	TPR (%)	FPR (%)	F值 (%)	AUC
BP	77.7778 (14/18)	3.8674 (14/362)	0.27548 (4/1452)	7.3684	0.5181
LR	76.4706 (13/17)	3.6723 (13/354)	0.42781 (4/935)	7.0081	0.5172
RBF	66.9377 (247/369)	73.5119 (247/336)	8.0581 (122/1514)	70.0709	0.8301
AB	52.3517 (256/489)	66.8407 (256/383)	15.5541 (233/1498)	58.7156	0.7622
SVM	91.2863 (220/241)	63.0372 (220/349)	1.4344 (21/1464)	74.5763	0.8088
RELM	89.5238 (188/210)	53.4091 (188/352)	2.0221 (22/1088)	66.9039	0.7604
IRELM	49.7619 (209/420)	58.3799 (209/358)	14.6528 (211/1440)	53.7275	0.7277
GA-IRELM	88.9706 (242/272)	64.191 (242/377)	1.9802 (30/1515)	74.5763	0.8117
PSO-IRELM	80.3571 (225/280)	61.3079 (225/367)	3.5415 (55/1553)	69.5518	0.7897
BSO-IRELM	83.9552 (225/268)	66.1765 (225/340)	2.701 (43/1592)	74.0132	0.8179

表 10 各算法在 Probe 上的性能评价指标

Table 10 Performance evaluation index of each algorithm on Probe

算法	精准率 (%)	TPR (%)	FPR (%)	F值 (%)	AUC
BP	82.9787 (390/470)	97.5 (390/400)	6.9444 (80/1152)	89.6552	0.9625
LR	62.6039 (226/361)	56.7839 (226/398)	15.8265 (135/853)	59.552	0.7418
RBF	99.5902 (243/244)	59.7052 (243/407)	0.071582 (1/1397)	74.6544	0.7982
AB	89.6359 (320/357)	82.4742 (320/388)	2.9887 (37/1238)	85.906	0.9009
SVM	73.3728 (372/507)	88.7828 (372/419)	9.467 (135/1426)	80.3456	0.9012
RELM	51.7661 (425/821)	97.7011 (425/435)	32.3265 (396/1225)	67.6752	0.8620
IRELM	79.9065 (342/428)	91.4439 (342/374)	7.2758 (86/1182)	85.2868	0.9308
GA-IRELM	89.8851 (391/435)	92.435 (391/423)	3.1884 (44/1380)	91.1422	0.9482
PSO-IRELM	89.7638 (3422/381)	94.4751 (342/362)	2.7465 (39/1420)	92.0592	0.9605
BSO-IRELM	89.6629 (399/445)	93.8824 (399/425)	3.2372 (396/1225)	91.7241	0.9548

表 11 各算法在 DoS 上的性能评价指标

Table 11 Performance evaluation index of each algorithm on DoS

算法	精准率 (%)	TPR (%)	FPR (%)	F值 (%)	AUC
BP	93.0233 (600/645)	81.5217 (600/736)	4.9614 (45/907)	86.8936	0.8898
LR	42.1687 (665/1577)	87.5 (665/760)	76.5743 (912/1191)	56.9106	0.5698
RBF	99.4074 (671/675)	90.9214 (671/738)	0.41152 (4/972)	94.9752	0.9530
AB	90.3509 (515/570)	70.6447 (515/729)	5.1838 (55/1061)	79.2918	0.8316
SVM	84.7118 (676/798)	90.1333 (676/750)	11.0009 (122/1109)	87.3385	0.9019
RELM	96.7391 (178/184)	25.0704 (178/710)	0.55453 (6/1082)	39.821	0.6230
IRELM	96.7059 (411/425)	54.0079 (411/761)	1.3449 (14/1041)	69.3086	0.7644
GA-IRELM	90.7539 (638/703)	89.4811 (638/713)	5.6326 (65/1154)	90.113	0.9222
PSO-IRELM	93.5302 (665/711)	89.502 (665/742)	4.1667 (16/1104)	91.4718	0.9292
BSO-IRELM	96.3636 (689/715)	93.6141 (689/736)	2.3402 (26/1111)	94.969	0.9578

83.15%, 62.7%, 71.9%, 86.35% 和 86.15%. 但由于 NSL-KDD 多分类数据集仍是非平衡数据集, 故从精确率、真正率、假正率、F 值和 AUC 等方面进一步分析各算法的入侵检测性能.

从表 9 可以看出, 对于 Normal 类型数据, BP 和 LR 的综合检测性能最差, 真正率仅为 3.8647% 和 3.6723%, 较 BSO-IRELM 差 62.3091% 和 62.5042%. 大多数情况下, BSO-IRLEM 的分类性能与 SVM、PSO-IRELM 以及 GA-IRELM 相近, 但较 AB、RBF、IRELM 和传统 RELM 性能均有所提升. 由于 BSO-IRELM 将大量正常数据误判为攻击类型, 因而会导致精确率和假正率略差.

表 12　各算法在 R2L 上的性能评价指标

Table 12　Performance evaluation index of each algorithm on R2L

算法	精准率 (%)	TPR (%)	FPR (%)	F值 (%)	AUC
BP	87.5 (14/16)	31.1111 (14/45)	0.13793 (2/1450)	45.9016	0.5150
LR	NaN (0/0)	0 (0/32)	0 (0/944)	NaN	0
RBF	68 (17/25)	56.6667 (17/30)	0.4908 (8/1630)	61.8182	0.7813
AB	NaN (0/0)	0 (0/36)	0 (0/1521)	NaN	0
SVM	NaN (0/0)	0 (0/36)	0 (0/1663)	NaN	0
RELM	50 (1/2)	3.4483 (1/29)	0.079745 (1/1254)	6.4516	0.5147
IRELM	81.8182 (9/11)	39.1304 (9/23)	0.13976 (2/1431)	52.9412	0.6951
GA-IRELM	90.9091 (20/22)	57.1429 (20/35)	0.11703 (2/1709)	70.1754	0.7852
PSO-IRELM	90.9091 (20/22)	54.0541 (20/37)	0.1173 (2/1705)	67.7966	0.7698
BSO-IRELM	92.3077 (24/26)	82.7586 (24/29)	0.39728 (7/1762)	69.0909	0.8258

表 13　各算法在 U2R 上的性能评价指标

Table 13　Performance evaluation index of each algorithm on U2R

算法	精准率 (%)	TPR (%)	FPR (%)	F值 (%)	AUC
BP	52.1739 (44/851)	97.1554 (444/457)	28.5614 (407/1425)	67.8899	0.8539
LR	88.8889 (40/45)	8.7719 (40/456)	0.55006 (5/909)	15.9681	0.5422
RBF	67.1033 (461/687)	94.274 (461/489)	16.0969 (226/1404)	78.4014	0.8966
AB	73.6301 (430/584)	92.6724 (430/464)	12.3695 (154/1245)	82.0611	0.9132
SVM	87.0044 (395/454)	88.565 (395/446)	4.4461 (59/1327)	87.7778	0.9238
RELM	59.0038 (462/783)	97.4684 (462/474)	28.841 (321/1113)	73.5084	0.8822
IRELM	65.2235 (467/716)	96.4876 (467/484)	20.4098 (249/1220)	77.8333	0.9003
GA-IRELM	76.7606 (436/568)	96.4602 (436/452)	9.2762 (132/1423)	85.4902	0.9397
PSO-IRELM	77.7228 (471/606)	95.9267 (471/491)	9.7332 (135/1387)	85.8706	0.9349
BSO-IRELM	80.9524 (442/546)	94.0426 (442/470)	7.2423 (104/1436)	87.0079	0.9362

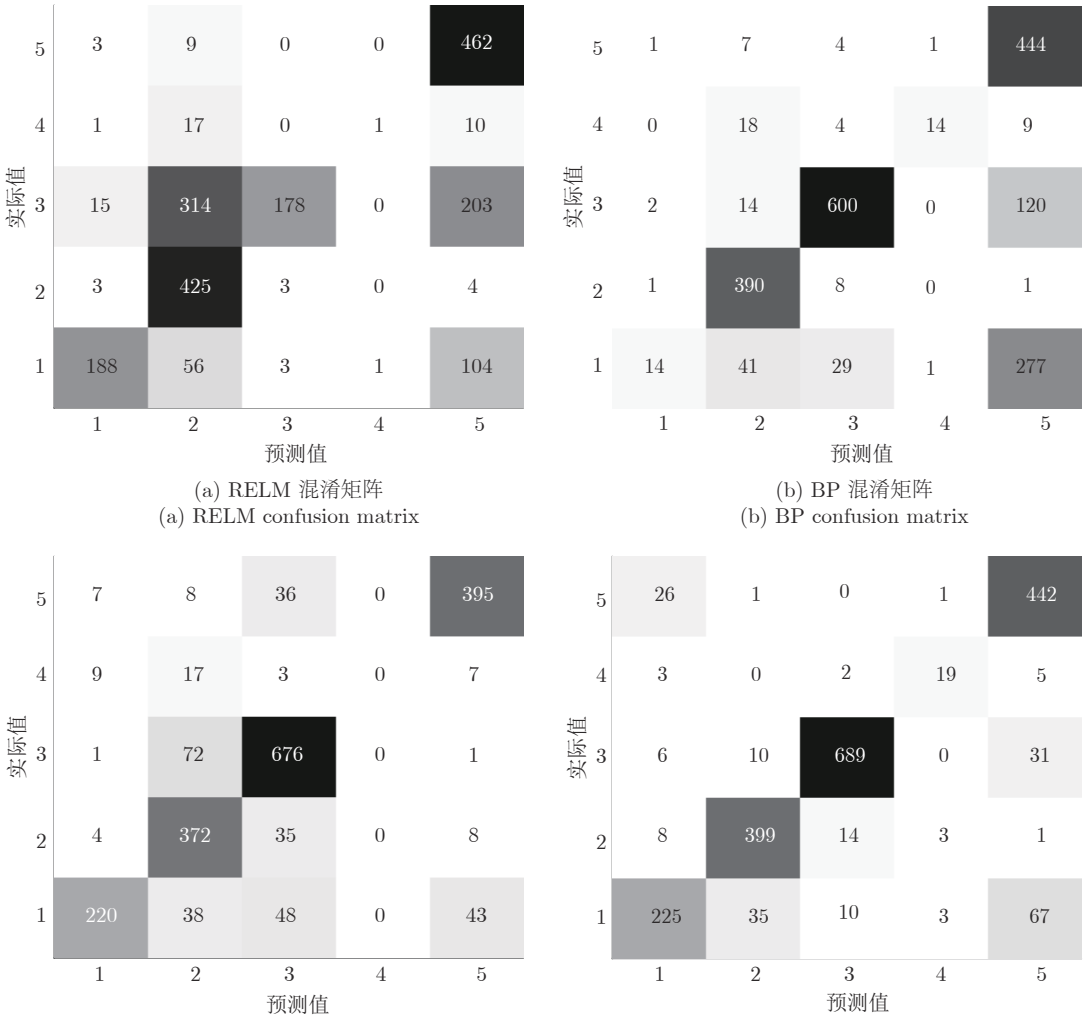
从表 10 可以看出, 对于 Probe 类型攻击, BP、AB、IRELM、GA-IRELM、PSO-IRELM 和 BSO-IRELM 的性能相近, 远优于 LR、SVM、RBF、RELM 的性能, 但总体上 BSO-IRELM 的性能最优. 此时, 除了 LR 和 RELM, 其余各算法对于 Probe 类型攻击均具有较强的识别能力.

从表 11 可以看出, 对于 DoS 类型攻击, BSO-IRELM 的检测性能最优, LR 检测性能最差, 除传统 RELM 和 IRELM 的真正率较差, 仅为 25.0704% 和 54.0079%, 其余各算法的性能评价指标均较优. 由于 DoS 的攻击数目最多, 如果使用聚类大小进行判断需要单独处理, 否则检测结果不理想. 但本文判定是根据入侵数据与正常数据的差异, 所以对于攻击数目较多的 DoS 攻击仍具有较好的检测结果.

从表 12 可以看出, 对于 R2L 攻击类型, 在大多数情况下, BP、RBF、RELM 和 IRELM 的性能相近, 效果均较差, LR、AB 和 SVM 的检测性能最差, 基本上无法正确识别 R2L 攻击类型, 而 BSO-IRELM 的性能较优, 相比于 RELM, 精确率提高了 42.3077%, 效果显著, 大大超出预想. R2L 攻击总数很少, 而且许多 R2L 入侵是伪装成合法用户身份进行攻击, 这就使得其特征与正常数据包类似, 造成 R2L 攻击检测困难. 但 BSO-IRELM 相比于 BP、LR、AB、SVM 和传统 RELM, 很好地学习了 R2L 的特征, 并将其正确分类.

从表 13 可以看出, 对于 U2R 类型攻击, SVM、AB、GA-IRELM、PSO-IRELM 和 BSO-IRELM 的性能相近, 效果较优, BP、LR、RBF、RELM 和 IRELM 的性能相近, 效果较差, 且 BSO-IRELM 相比于 BP、LR、RBF、AB、SVM 和传统 RELM, 分类性能均有所提高, AUC 分别提高 0.0823、0.3940、0.0396、0.0230、0.0124 和 0.054. 本文对 NSL-KDD 数据集进行去重并随机产生测试集和训练集, 可以在一定程度上降低不同数据类型数量之间的差距, 使模型学习到更多的 U2R 特征, 虽然在某些方面检测性能未最优, 但是整体检测性能较好地符合预期.

当精确率和召回率发生冲突时, 很难对模型进行比较. 而 F 值同时兼顾了精确率和召回率, 可以看作是精确率和召回率的一种调和平均, 能够更好地评价模型. BSO-IRELM 的 F 值均较优, 对 Nor-



(a) RELM 混淆矩阵
(a) RELM confusion matrix

(b) BP 混淆矩阵
(b) BP confusion matrix

(c) SVM 混淆矩阵
(c) SVM confusion matrix

(d) BSO-IRELM 混淆矩阵
(d) BSO-IRELM confusion matrix

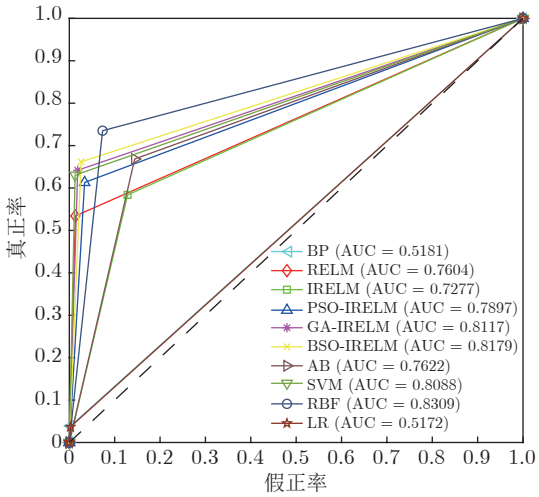
图 9 多元分类混淆矩阵

Fig. 9 Multiple classification confusion matrix

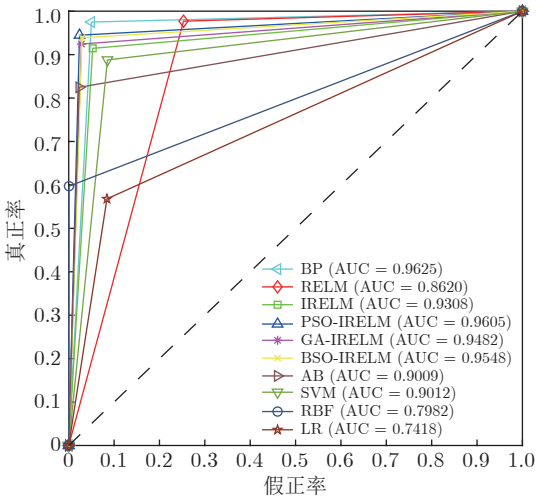
mal 类型数据, 比 LR 增加 67.0051%; 对 Probe 类型攻击, 比 LR 增加 32.1721%; 对 R2L 类型攻击, 比 RELM 增加 62.6369%; 对 DoS 和 U2R 类型攻击, BSO-IRELM 的 F 值为各算法中最优的. 由于 DoS 和 U2R 攻击类型数目较多, BSO-IRELM 的特征学习比较充分, 所以对 DoS 和 U2R 攻击类型检测的 F 值最优. 表明特征库中的特征越多、越丰富, 模型的分类效果越好.

混淆矩阵使用热度图, 通过色差、亮度来展示数据的差异, 易于理解. 深色表示预测值和真实值重合较多的区域, 浅色表示预测值和真实值重合较少的区域. 从图 9 可以看出, BSO-IRELM 的深色区域集中出现在混淆矩阵副对角线上, 且副对角线之和为 1 774, 其中 BSO-IRELM 的分类准确率最优, 符合预期. 从混淆矩阵也可以看出实际分类情况, 如 RELM 将大量为 DoS 类型攻击和 Normal 数据预测为 U2R 类型攻击, BP 将大量的 Normal 数据和 DoS 类型攻击预测为 U2R 攻击类型. SVM 基本上无法正确识别 R2L 攻击类型. ROC 曲线存在一个巨大优势, 当正负样本的分布发生变化时, 其形状能够保持基本不变, 因此 ROC 曲线能够降低不同测试集带来的干扰, 更加客观地衡量模型本身的性能. 从图 10 可以看出, BSO-IRELM 对 DoS 和 R2L 攻击类型具有最优的 AUC, 对 Normal 类型, LR 的 AUC 较 BSO-IRELM 差 0.3007, 对 Probe 类型攻击, RELM 的 AUC 较 BSO-IRELM 差 0.0928, 对 U2R 攻击类型, BP 的 AUC 较 BSO-IRELM 差 0.0823. 同时, LR 对各种类型的 AUC 均较差, AB、SVM、LR 对 R2L 攻击类型的 AUC 均为 0, BP 对 Normal 类型和 R2L 攻击类型的 AUC 均较差.

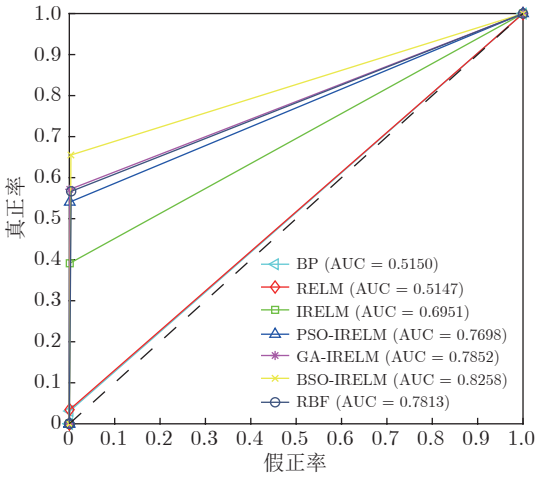
综上所述, 在 UCI 数据集上, BSO-IRELM 的



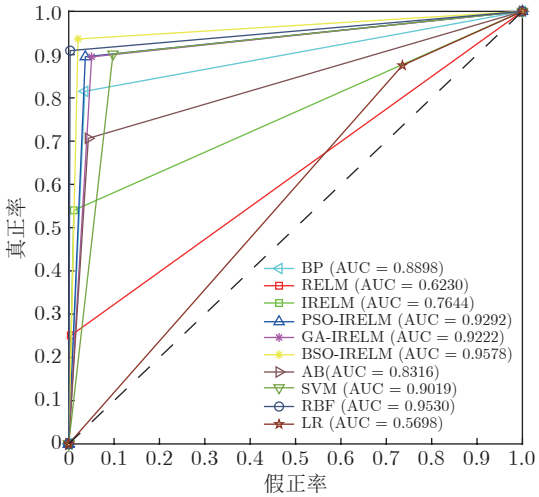
(a) Normal-ROC 曲线图
(a) Normal-ROC curves



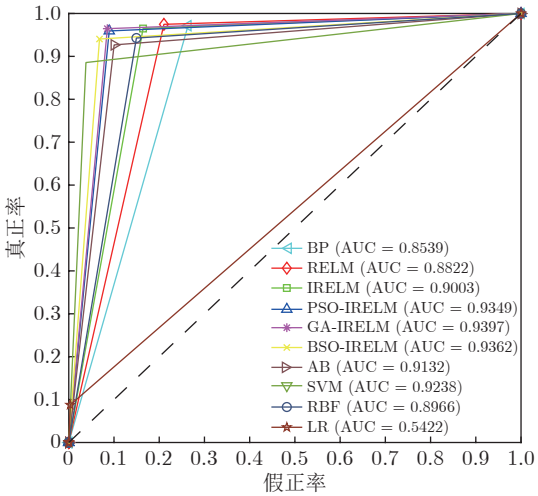
(b) Probe-ROC 曲线图
(b) Probe-ROC curves



(c) R2L-ROC 曲线图
(c) R2L-ROC curves



(d) DoS-ROC 曲线图
(d) DoS-ROC curves



(e) R2L-ROC 曲线图
(e) R2L-ROC curves

图 10　多元分类 ROC 曲线对比图

Fig. 10　Multiple classification ROC curve comparison diagram

各项性能均优于 IRELM 和传统的 RELM 算法, 这说明引入 LU 分解法以及 BSO 算法的必要性, 同时也验证了之前关于 RELM 存在潜在问题的假设. 从而验证了 BSO-IRELM 算法具有较优的分类性能. 在 NSL-KDD 数据集上进一步进行 2 元与多元分类入侵检测, 在大多数情况下, BSO-IRELM 的性能优于 BP、LR、RBF、AB、SVM、IRELM、GA-IRELM、PSO-IRELM 和传统 RELM 算法.

4 结束语

本文提出了一种基于天牛群优化与改进正则化极限学习机 (BSO-IRELM) 的网络入侵检测算法, 有效解决了现有方法存在的准确率、精确率、真正率、假正率等偏低的问题. 算法使用了 LU 分解以求解 RELM 的输出权值矩阵, 并设计了天牛群优化算法 BSO, 实现了对 RELM 的权值和阈值的联合优化. 实验结果表明, 无论在机器学习数据集 UCI 或网络入侵检测数据集 NSL-KDD 上, 与已有方法相比, BSO-IRELM 算法在各种评价指标上都具有明显优势. 下一步研究的重点是扩展 BSO-IRELM 的检测应用领域, 检验其在网络安全威胁检测 (如病毒检测、漏洞检测等) 中的使用效果.

References

1 Tsai C F, Hsu Y F, Lin C Y, Lin W Y. Intrusion detection by machine learning: A review. *Expert Systems With Applications*, 2009, **36**(10): 11994−12000

2 Ren Jia-Dong, Liu Xin-Qian, Wang Qian, He Hai-Tao, Zhao Xiao-Lin. An multi-level intrusion detection method based on KNN outlier detection and random forests. *Journal of Computer Research and Development*, 2019, **56**(3): 566−575
(任家东, 刘新倩, 王倩, 何海涛, 赵小林. 基于KNN离群点检测和随机森林的多层入侵检测方法. 计算机研究与发展, 2019, **56**(3): 566−575)

3 Gao Ni, Gao Ling, He Yi-Yue, Wang Hai. A lightweight intrusion detection model based on autoencoder network with feature reduction. *Acta Electronica Sinica*, 2017, **45**(3): 730−739
(高妮, 高岭, 贺毅岳, 王海. 基于自编码网络特征降维的轻量级入侵检测模型. 电子学报, 2017, **45**(3): 730−739)

4 Ahmad I, Basheri M, Iqbal M J, Rahim A. Performance comparison of support vector machine, random forest, and extreme learning machine for intrusion detection. *IEEE Access*, 2018, **6**: 33789−33795

5 Mabu S, Gotoh S, Obayashi M, Kuremoto T. A random-forests-based classifier using class association rules and its application to an intrusion detection system. *Artificial Life and Robotics*, 2016, **21**(3): 371−377

6 Shenfield A, Day D, Ayesh A. Intelligent intrusion detection systems using artificial neural networks. *ICT Express*, 2018, **4**(2): 95−99

7 Ding H W, Wan L. Research on intrusion detection based on KPCA-BP neural network. In: Proceedings of the 18th IEEE International Conference on Communication Technology (ICCT). Chongqing, China: IEEE, 2018. 911−915

8 Wang T, Wei L H, Ai J Q. Improved BP Neural Network for Intrusion Detection Based on AFSA. In: Proceedings of the 2015 International Symposium on Computers and Informatics (ISCI). Beijing, China: Atlantis Press, 2015. 373−380

9 Huang G, Song S J, Gupta J N D, Wu C. Semi-supervised and unsupervised extreme learning machines. *IEEE Transactions on Cybernetics*, 2014, **44**(12): 2405−2417

10 Lu Hui-Juan, An Chun-Lin, Ma Xiao-Ping, Zheng En-Hui, Yang Xiao-Bing. Disagreement measure based ensemble of extreme learning machine for gene expression data classification. *Chinese Journal of Computers*, 2013, **36**(2): 341−348
(陆慧娟, 安春霖, 马小平, 郑恩辉, 杨小兵. 基于输出不一致测度的极限学习机集成的基因表达数据分类. 计算机学报, 2013, **36**(2): 341−348)

11 Chen Xiao-Yun, Liao Meng-Zhen. Dimensionality reduction with extreme learning machine based on sparsity and neighborhood preserving. *Acta Automatica Sinica*, 2019, **45**(2): 325−333
(陈晓云, 廖梦真. 基于稀疏和近邻保持的极限学习机降维. 自动化学报, 2019, **45**(2): 325−333)

12 Yang Z X, Wang X B, Wong P K, Zhong J H. ELM based representational learning for fault diagnosis of wind turbine equipment. *Proceedings of ELM-2015 Volume 2: Theory, Algorithms and Applications (II)*. Cham: Springer, 2016. 169−178

13 Zou Wei-Dong, Xia Yuan-Qing. Virtual machine power prediction using incremental extreme learning machine based on compression driving amount. *Acta Automatica Sinica*, 2019, **45**(7): 1290−1297
(邹伟东, 夏元清. 基于压缩动量项的增量型ELM虚拟机能耗预测. 自动化学报, 2019, **45**(7): 1290−1297)

14 Ku J H, Zheng B. Intrusion detection based on self-adaptive differential evolution extreme learning machine with gaussian kernel. In: Proceedings of the 8th International Symposium on Parallel Architecture, Algorithm and Programming. Haikou, China: Springer, 2017. 13−24

15 Deng W Y, Zheng Q H, Chen L. Regularized extreme learning machine. In: Proceedings of the 2009 IEEE Symposium on Computational Intelligence and Data Mining. Nashville, USA: IEEE, 2009. 389−395

16 Huang G B, Wang D H, Lan Y. Extreme learning machines: A survey. *International Journal of Machine Learning and Cybernetics*, 2011, **2**(2): 107−122

17 Jiang X Y, Li S. BAS: Beetle antennae search algorithm for optimization problems [Online], available: https://arxiv.org/pdf/1710.10724.pdf, March 27, 2019

18 Jiang X Y. Li S. Beetle antennae search without parameter tuning (BAS-WPT) for multi-objective optimization. *Filomat*, 2020, **34**(15): 5113−5119

19 Liu Ying, Qian Zhi-Hong, Jia Di. Universal localization algorithm based on beetle antennae search in indoor environment. *Journal of Electronics and Information Technology*, 2019, **41**(7): 1565−1571
(刘影, 钱志鸿, 贾迪. 室内环境中基于天牛须寻优的普适定位方法. 电子与信息学报, 2019, **41**(7): 1565−1571)

20 Wu Q, Ma Z P, Xu G, Li S, Chen D C. A novel neural network classifier using beetle antennae search algorithm for pattern classification. *IEEE Access*, 2019, **7**: 64686−64696

21 Kaur G, Arora S. Chaotic whale optimization algorithm. *Journal of Computational Design and Engineering*, 2018, **5**(3): 275−284

22 Ling Y, Zhou Y Q, Luo Q F. Lévy flight trajectory-based whale optimization algorithm for global optimization. *IEEE Access*, 2017, **5**: 6168−6186

23 Sarangi A, Samal S, Sarangi S K. Analysis of Gaussian and Cauchy mutations in modified particle swarm optimization algorithm. In: Proceedings of the 5th International Conference on Advanced Computing and Communication Systems (ICACCS). Coimbatore, India: IEEE, 2019. 463−467

24 Rudolph G. Local convergence rates of simple evolutionary algorithms with Cauchy mutations. *IEEE Transactions on Evolu-*

tionary Computation, 1997, **1**(4): 249−258

25 Gauthama Raman M R, Somu N, Kirthivasan K, Liscano R, Shankar Sriram V S. An efficient intrusion detection system based on hypergraph-Genetic algorithm for parameter optimization and feature selection in support vector machine. *Knowledge-Based Systems*, 2017, **134**: 1−12

26 Mazini M, Shirazi B, Mahdavi I. Anomaly network-based intrusion detection system using a reliable hybrid artificial bee colony and AdaBoost algorithms. *Journal of King Saud University-Computer and Information Sciences*, 2019, **31**(4): 541−553

27 Wang Dong-Feng, Meng Li. Performance analysis and parameter selection of PSO algorithms. *Acta Automatica Sinica*, 2016, **42**(10): 1552−1561
(王东风, 孟丽. 粒子群优化算法的性能分析和参数选择. 自动化学报, 2016, **42**(10): 1552−1561)

28 UCI. UCI dataset [Online], available: https://archive.ics.uci.edu/ml/index.php, June 27, 2019

29 NSL-KDD dataset [Online], available: https://www.unb.ca/cic/datasets/nsl.html, June 27, 2019

30 Vinayakumar R, Alazab M, Soman K P, Poornachandran P, Al-Nemrat A, Venkatraman S. Deep learning approach for intelligent intrusion detection system. *IEEE Access*, 2019, **7**: 41525−41550

王振东 博士, 江西理工大学信息工程学院副教授. 主要研究方向为无线传感器网络, 智能物联网, 认知计算, 大数据和信息安全.
E-mail: wangzhendong@hrbeu.edu.cn
(**WANG Zhen-Dong** Ph.D., associate professor at the School of Information Engineering, Jiangxi University of Science and Technology. His research interest covers wireless sensor networks, smart internet of things, cognitive computing, and big data and information security.)

刘尧迪 江西理工大学信息工程学院硕士研究生. 主要研究方向为网络安全, 入侵检测, 群智能优化算法, 机器学习与深度学习. 本文通信作者.
E-mail: liuyaodi@yeah.net
(**LIU Yao-Di** Master student at the School of Information Engineering, Jiangxi University of Science and Technology. Her research interest covers network security, intrusion detection, swarm intelligence optimization algorithm, machine learning, and deep learning. Corresponding author of this paper.)

杨书新 博士, 江西理工大学信息工程学院副教授. 主要研究方向为数据管理, 信息检索和生物信息学.
E-mail: yimuyunlang@sina.com
(**YANG Shu-Xin** Ph.D., associate professor at the School of Information Engineering, Jiangxi University of Science and Technology. His research interest covers data management, information retrieval, and bioinformatics.)

王俊岭 博士, 江西理工大学信息工程学院副教授. 主要研究方向为分布式计算, 容错, 计算机视觉.
E-mail: wangjunling@jxust.edu.cn
(**WANG Jun-Ling** Ph.D., associate professor at the School of Information Engineering, Jiangxi University of Science and Technology. His research interest covers distributed computing, fault tolerance, and computer vision.)

李大海 博士, 江西理工大学信息工程学院副教授. 主要研究方向为分布式系统服务质量控制, 分布式系统自学资源调度控制.
E-mail: dlai6535@aliyun.com
(**LI Da-Hai** Ph.D., associate professor at the School of Information Engineering, Jiangxi University of Science and Technology. His research interest covers distributed system quality of service (QoS) control, and distributed system self-learning resource scheduling control.)