软件定义网络中基于改进随机森林算法的入侵检测研究

马 群[1] 胡佳卉[2] 于雅静[1]

[1](中国移动通信集团设计院有限公司河北分公司 河北 石家庄 050000)

[2](中国联通智网创新中心 北京 100000)

摘 要 针对软件定义网络中入侵数据流特征差异性较大以及随机森林算法应用于入侵检测的适用性等问题,提出一种基于改进随机森林算法的入侵检测模型,依据 Fisher 比分析入侵数据特征差异性,按它们所对应的不同取值进行特征分区;引入加权的投票方法,以增加分类性能较好的决策树的权重;以最大信息增益率为标准进行节点分裂;改进网格搜索算法,使其对随机森林参数优化的效果得到进一步提高。通过实验分析,在模型的准确率、F1 值、AUC 值等评估指标上都有明显提升,验证了改进算法的有效性。

关键词 软件定义网络 随机森林算法 入侵检测 Fisher 准则 网格搜索算法

中图分类号 TP393 **文献标志码** A DOI:10.3969/j.issn.1000-386x.2024.11.052

INTRUSION DETECTION BASED ON IMPROVED RANDOM FOREST ALGORITHM IN SOFTWARE DEFINED NETWORK

Ma Qun[1] Hu Jiahui[2] Yu Yajing[1]

[1](*China Mobile Communication Design Institute Limited Hebei Branch*, *Shijiazhuang 050000*, *Hebei*, *China*)

[2](*Intelligent Network & Innovation Center of China Unicom*, *Beijing 100000*, *China*)

Abstract In view of the large differences in the characteristics of intrusion data streams in software-defined networks and the applicability of the random forest algorithm in intrusion detection, this paper proposes an intrusion detection model based on an improved random forest algorithm. We analyzed the differences in the characteristics of intrusion data based on the Fisher ratio, and conducted feature partitioning according to their corresponding values. A weighted voting method was introduced to increase the weight of the decision tree with better classification performance. The node was split based on the maximum information gain rate. The grid search algorithm was improved to further improve the effect of random forest parameter optimization. Through experimental analysis, the accuracy, F1 value, AUC value and other evaluation indicators of this model is significantly improved, which verifies the effectiveness of the improved algorithm.

Keywords Software-defined network (SDN) Random forest algorithm Intrusion detection Fisher criterion Grid search algorithm

0 引 言

入侵检测主要分为误用检测和异常检测两种类别,前者的特点是对于未知的攻击较难检测,但对已知的攻击有良好的检测效果;后者的特点是对于未知的攻击比较敏感,但误报率较高。因此,如何使入侵检测的漏报率和误报率得到有效降低并提高精确率是一个重点研究方向。

目前,已经有很多关于 SDN 入侵检测的文献,其中,在文献[1]中提出了一种软件定义网络(SDN)环境上的可伸缩入侵检测系统(IDS),运行 IDS 的虚拟机,SDN 控制器和网络攻击软件通过启用 OpenFlow 的软件交换机相互连接。但此方法着重于在网络交换机上进行分布式流量采样以进行恶意流量检查。文献[2]基于车载自组织网络(VANET)中,针对现有的入侵检测系统(IDS)解决方案仅限于检测本地子网而不是整个 VANET 下的异常网络行为,提出一种利用生成

收稿日期:2021-06-07。**马群**,硕士,主研领域:机器学习,软件定义网络,电信技术。**胡佳卉**,硕士。**于雅静**,高工。

对抗网络的深度学习并探索分布式 SDN 为 VANET 设计协作式入侵检测系统,该系统使多个 SDN 控制器可以共同训练整个网络的全局入侵检测模型,而无须直接交换它们的子网流。文献[3]中通过利用 SDN 流的概念,采用了分层和重量级入侵检测系统(IDS)的体系结构,基于流的 IDS 使用基于受 DARPA 入侵检测数据集训练的支持向量机(SVM)的异常检测算法。第一道防线检测网络上的任何入侵。当检测到攻击时,恶意流将镜像到基于数据包的 IDS,以进行进一步检查和采取措施。文献[4]中提出一种用于 SDN 的门控循环单元循环神经网络(GRU-RNN)支持的入侵检测系统。使用 NSL-KDD 数据集对提出的方法进行了测试,仅六个原始特征就达到了 89% 的精度。文献[5]中为保护 SDN 控制层和 SDN 基础结构层之间的通信通道免遭错误数据注入攻击,使用深度学习算法进行入侵检测,首先分析了在 SDN 网络中流通的流量,然后采用对数函数用最小/最大标量技术对流量特征进行归一化,利用 ReLU 和 Softmax 函数进行流分类。文献[6]中使用决策树算法将短时间内从客户端向服务器请求的大量数据用作预测数据,以生成学习数据。已经形成的学习数据用于预测,使用基于异常的自适应增强算法来检测和预防 DoS 攻击。实验结果表明自适应提升算法检测攻击的有效性达到 91.3%。文献[7]中在用户验证、数据包验证和流验证方面调查了三层入侵检测与防御系统(IDPS)中入侵者的参与。在第二层,交换机负责从数据包中提取关键特征,并将其分类为正常、可疑和恶意;在控制器中分析不匹配的数据包,控制器将两个队列保持为可疑和正常队列。最后使用深度学习方法对可疑队列分组进行分类和预测。文献[8]中针对在满足数据处理和分析质量的同时,缺乏经济地收集网络数据的具体解决方案,通过动态地基于网络状态自动选择合适的数据收集节点,提出一种 SDN 中的自适应网络数据收集系统,有效减少收集到的数据量,同时确保后续数据分析,并验证了该方法在 CPU/内存消耗、存储使用、流量大小恢复和威胁感知方面的优势。文献[9]中采用一种基于控制平面的业务流程,提出的机制包括一个启用了 Cuda 的混合 DL 驱动架构,该架构利用长短期记忆(LSTM)和卷积神经网络(CNN)的预测能力来有效及时地检测多矢量威胁和攻击。文献[10]针对 SDN 中的 SSH 攻击和 DDoS 攻击,采用基于深度学习的入侵检测和防御系统(DL-IDPS),SDN 交换机中的数据包长度作为深度学习模型的序列进行收集,以识别异常和恶意数据包。文献[11]针对 SDN 体系结构防御网络威胁的流管理方法,提出了改进的基于行为的 SVM 算法对网络入侵检测系统的网络威胁进行分类,采用 ID3 算法筛选最佳特征来训练支持向量分类器(SVC),同时考虑了实验的整体检测精确度,从而加快了对正常模式和入侵模式的学习,并提高了精确性检测入侵。文献[12]研究使用深度神经网络(DNN)的 OpenFlow 控制器中基于流的异常检测方法,提出一种组合式门控循环单元长期短期记忆(GRU-LSTM)网络入侵检测系统,并采用了适当的 ANOVA F 检验和递归特征消除(RFE)(ANOVA F-RFE)特征选择方法。在文献[13]中提出一种基于人工免疫系统的 IDS(AIS-IDS),AIS-IDS 可以检测网络行为的变化并识别攻击,而无须事先了解它们。当异常和正常交通行为之间没有清晰的边界时,模糊逻辑与 AIS 一起用于检测,以最大程度地减少不确定性。

针对降低入侵检测的漏报率和误报率、入侵数据流特征差异性较大等问题,本文提出一种基于改进随机森林算法的入侵检测方法,根据 Fisher 比对入侵数据特征差异性的分析,进行特征分区,然后引入加权的投票方法并以最大信息增益率为标准进行节点分裂,最后改进网格搜索算法,提高优化模型参数的效率。

1 入侵检测模型

本文采用的入侵检测模型分为三个平面,其中包括应用和分类平面、控制平面、转发平面,模型如图 1 所示。

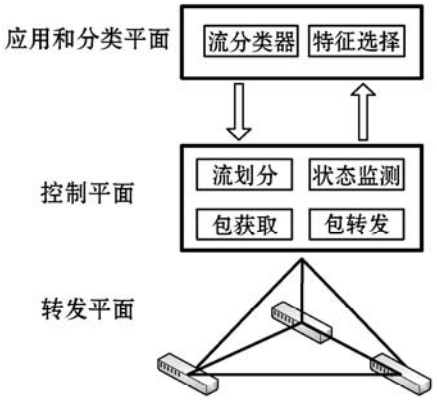

图 1 入侵检测模型

应用和分类平面由流分类器和特征选择模块组成。前者采用标记是否属于正常流量或特定类型攻击的方法来对网络流量进行分类,后者通过分析可疑流的有关特征筛选最佳子集的方法来提升处理高维数据的效率[14]。

控制平面是对转发平面收集的数据进行初步分析,流划分模块分析流量统计信息,把数据包聚类成

流。为了使异常识别的实时性得到提高,对数据包检查和收集需要按一定时间间隔进行[15]。状态检测模块监视网络状态,向交换机周期请求上报流量信息。

模型中的转发平面由 OpenFlow 交换机组成,交换机将攻击流的报文丢弃从而阻止入侵行为,转发平面完成采集上传流量信息和数据包在交换机之间的转发,为控制平面提供可疑数据流和网络实时状态。

2　算法原理

2.1　随机森林算法

Breiman[16]提出随机森林算法,基于 Bagging 算法加入 Bootstrap 技术形成决策树,再通过简单的投票方法得出随机森林算法的最终结果。加入 Bootstrap 技术有效地提高了算法的准确率,避免了单棵决策树容易出现的过拟合问题。

2.2　ID3 算法

ID3 算法是一种对决策树的改进算法,它是以信息熵和信息增益作为节点分裂的衡量指标。在做分类任务时,首先通过计算得到训练集整体的信息熵(Entropy),然后逐一计算训练集中某一特征的经验条件熵,通过相减得到最终的信息增益,最后以最大信息增益为标准选择特征进行节点分裂。

训练集 S 信息熵为 $E(S)$,P_k 类样本出现概率为 $|P_k|/|D|$,信息熵为:

$$E(S) = -\sum_{k=1}^{k}\frac{|P_k|}{|D|}\log_2\frac{|P_k|}{|D|} \tag{1}$$

假设 C 有 n 个不同的取值,将 S 分为 n 个子集 S_1,S_2,…,S_n。特征 C 对训练集 S 的条件熵 $E(S|C)$ 为:

$$E(S|C) = \sum_{i=1}^{n}\frac{|D_i|}{|D|}E(s_i) \tag{2}$$

其中 $E(S_i)$ 为:

$$E(S_i) = -\sum_{k=1}^{k}\frac{|D_{ik}|}{|D_i|}\log_2\frac{|D_{ik}|}{|D_i|} \tag{3}$$

最后计算信息增益 $Gain(S|C)$ 为:

$$Gain(S|C) = E(S) - E(S|C) \tag{4}$$

2.3　CART 算法

1984 年 Breiman 提出 CART 算法,节点分裂指标采用最小 Gini 系数,目前通常使用 CART 决策树来集成随机森林[17]。

p_m 为训练集 M 个类别中,样本点属于第 m 类的概率,概率分布的 Gini 系数为:

$$Gini(p) = \sum_{k=1}^{k}p_m(1-p_m) = 1 - \sum_{m=1}^{M}p_m^2 \tag{5}$$

每个划分的 Gini 系数:将 p 分隔成 s 个子集 p_1,p_2,…,p_s,划分的 Gini 系数为:

$$Gini_{\text{split}}(p) = \sum_{m=1}^{s}\frac{|P_m|}{|P|}Gini(P_m) \tag{6}$$

3　改进随机森林算法

3.1　节点分裂方法的改进

在传统的随机森林算法中一般采用 Gini 系数作为决策树分裂节点的依据,这样往往存在效率较低耗时较长等问题。CART 算法虽然可以避免过拟合,但是在构建决策树时需要对训练数据多次扫描,这样就可能会造成耗时较长效率较低[18]。因此,根据数据的特性本文在改进的随机森林算法中不再使用最小 Gini 系数进行节点分裂,而是将最大信息增益率作为节点分裂的衡量标准。采用信息增益率来选择属性能够处理不完整数据,同时避免了选择属性时偏移值过多的缺点。具体计算方法如下:

P_i 为样本集 S_B 中类别 i 出现的概率,S_B 的信息熵为:

$$Entropy(S_B) = -\sum_{i=1}^{c}P_i\log_2P_i \tag{7}$$

S_{B1},S_{B2},…,S_{Bl}为特征 A 将 S_B 划分的 l 个子集,划分的信息熵为:

$$Entropy(S_B|A) = -\sum_{i=1}^{l}\frac{|S_{Bi}|}{|S_B|}Entropy(S_B) \tag{8}$$

信息增益 Gain 可表示为:

$$Gain(S_B|A) = Entropy(S_B) - Entropy(S_B|A) \tag{9}$$

信息增益率表示为:

$$GainRatio(S_B|A) = \frac{Gain(S_B|A)}{Entropy_{\text{split}}(S_B|A)} \tag{10}$$

式中:

$$Entropy_{\text{split}}(S_B|A) = -\sum_{i=1}^{l}\frac{|S_{Bi}|}{|S_B|}\log_2\frac{|S_{Bi}|}{|S_B|} \tag{11}$$

3.2　投票方法的改进

传统随机森林算法的投票方法略显简单,为增强投票方法的实用性,提高随机森林算法的精确率,在原有的投票方法上加以改进,增加了基分类器中精确性较高的决策树的权重,提高其“话语权”,从而使投票方法更加合理。传统随机森林算法在分类时所有单棵决策树权重相同,单棵决策树的差异性往往被忽略。也就是说少数精确性高、性能优良的决策树会被多数精确性低、性能一般的决策树所遮掩,反而造成集成学习的效果更差。因此本文引进一种加权集成的投票法

则。每棵决策树的权重表示如下：

P_{il}为样本 X 被分类器 T_i 分成 l 类的概率，l 为分裂子集数，c 表示类别数，则权重为：

$$\omega_i = \frac{\sum_{l} P_{il}}{\sum_{il} P_{il}} \quad l = 1,2,\cdots,c \tag{12}$$

再对每个类别的置信度进行计算：

$$u_l(x) = \frac{1}{N}\sum_{i=1}^{N} \omega_i P_{il}(x) \tag{13}$$

3.3 网格搜索算法的改进

网格搜索算法通常用来优化模型的参数，但对于一些参数较多且取值范围较大的模型来说，使用网格搜索会很耗时。为提高效率且保证优化效果，本文提出一种改进的网格搜索算法。

网格搜索算法是网格化处理作为变量的区域，即将算法参数可能取值的所有排列组合结果生成“网格”。然后采用穷举法计算运行所有的网格点，寻找出满足约束函数的目标函数值，最后通过比较得到最优点[19]。由于计算所有的网格点耗费了大量计算时间，因此为提高计算效率，本文提出一种大范围寻优、小范围求解的方法。首先，用较长的间距在较大的范围内来划分出较稀疏的网格，进行第一步的大范围寻优预搜索，找出最优点所在的区域范围；然后再用较短的间距在最优点所在的区域范围内划分出细密的网格，进行第二步小范围求解，重复以上步骤，不断减小网格间距直至目标函数变化量或网格间距小于给定值[20]。

在保证随机森林算法性能得到提高的前提下，也要考虑到每棵决策树的准确性和树的多样性[21]。在本文中使用改进的网格搜索算法寻求随机森林最优参数，选用袋外分数即模型的泛化能力作为目标函数值。改进网格搜索方法的具体步骤如下：

（1）确定需要优化的参数，相应的参数就是网格上的点，设定较长的网格间距来划分网格。

（2）将网格上的每一组参数全部遍历一次，采用随机森林算法中的袋外数据分数对分类误差进行评估，进行初步的大范围寻优。

（3）选择袋外分数最高即误差最小的一组数据，继续缩短网格间距进行参数优化。若袋外分数或网格间距满足要求，则输出结果，否则重复上述步骤。

3.4 改进算法执行步骤

针对上文所述，本文提出一种改进的随机森林算法。在特征提取时，采用基于 Fisher 比的分区采样的特征选择方法降低决策树之间的相关性增加决策树的多样性[22]；以最大信息增益率为决策树分裂节点，提高算法效率；加入加权投票方法，提高精确率较高的决策树的“话语权”；最后使用改进的网格搜索算法进行参数优化。

改进的随机森林算法执行步骤如下：

（1）将输入的数据分为训练集和测试集。

（2）采用改进的特征选择方法对训练集进行分区采样。

（3）以改进的节点分裂算法作为评价标准，进行节点分裂，生成 N 棵决策树。

（4）采用改进的网格搜索算法优化随机森林算法参数。

（5）训练调参完成后，将测试集作为输入，根据改进的投票方法对测试集样本进行分类。

4 实验分析

4.1 高维数据特征提取方法的改进

在针对入侵数据流特征差异性较大的问题时，根据 Fisher 比对入侵数据特征差异性的分析，进行特征分区。同时，在进行高维数据的特征提取时，为了改善算法过拟合现象，也使用分区采样方法，可增加特征子集的多样性，从而使泛化误差尽量变小并且压缩了数据的维度，改进了传统随机森林完全随机的采样方式，使用分区采样的方法，同样可以降低数据的复杂程度。

使用随机森林来判断特征的重要性的时候，无须对训练数据做标准化或者归一化处理，也无须考虑训练数据是否线性可分。通过随机森林算法，可以得到每一个特征数据的重要程度，特征的重要程度之和为 1，将重要程度过低的数据剔除。

改进随机森林算法在选择特征时，首先在初始分类基础上标注样本所属子类，e 为子类数量；然后使用 Fisher 比分析特征重要性。设训练集样本为 m 个，分别属于 M 个类别：第 ω 类的集合，用 m_ω 表示样本个数，用 μ_c 来表示全部样本中第 c 维特征的均值，用 $\mu_{\omega c}$来表示第 ω 类中第 c 维特征的均值，用 $\sigma^2_{\omega c}$来表示第 ω 类中第 c 维特征的方差。因此类间方差的计算方法如下：

$$S_{\mathrm{B}}^{c} = \frac{1}{m}\sum_{\omega=1}^{M} m_\omega(\mu_{\omega c} - \mu_c)^2 \tag{14}$$

类内方差可表示为：

$$S_{\mathrm{W}}^{c} = \frac{1}{m}\sum_{\omega=1}^{M} m_\omega \sigma^2_{\omega c} \tag{15}$$

则 Fisher 比可表示为：

$$J_{\mathrm{Fisher}}(c) = \frac{S_B^c}{S_W^c} \tag{16}$$

根据 Fisher 比计算各个子类依赖的重要特征集合 $F_1, F_2, \cdots, F_e$,然后采用集合运算得出共有特征 F_{common},各个子的类私有特征 $F_1, F_2, \cdots, F_e$,以及不重要特征 F_{unim}。

共有特征可表示为:

$$F_{\text{common}} = F_1 \cap F_2 \cap \cdots \cap F_e \tag{17}$$

不重要特征可表示为:

$$F_{\text{unim}} = F_{\text{all}} - F_1 - F_2 - \cdots - F_e \tag{18}$$

各个子类的私有特征可表示为:

$$\begin{aligned} F_1 &= F_1 - F_{\text{common}} \\ &\vdots \\ F_e &= F_e - F_{\text{common}} \end{aligned} \tag{19}$$

最后,构造特征子集时按照比例依次从集合 $\{F_{\text{common}}, F_1, F_{\text{unim}}\}, \{F_{\text{common}}, F_2, F_{\text{unim}}\}, \cdots, \{F_{\text{common}}, F_e, F_{\text{unim}}\}$ 中随机采样。

4.2　优化参数

将处理好的数据按照常用的 7∶3的比例划分为训练集和测试集,70% 的数据用于训练模型,30% 的数据用于测试模型,并比较随机森林算法和改进随机森林算法的实验结果。在建模的参数调优时,采用改进的网格搜索算法快速高效地调参。现以决策树最大深度 m_{d} 和内部节点再划分所需最小样本数 m_{n} 为例,用改进的网格搜索算法对随机森林参数进行优化。首先在进行大间距初步搜索的时候将决策树最大深度 m_{d} 的取值范围确定为 $0 < m_{\text{d}} < 41$,网格间距设定为 5;内部节点再划分所需最小样本数 m_{n} 的取值范围确定为 $0 < m_{\text{n}} < 151$,网格间距设定为 30。搜索结果如表 1 所示。

表 1　大网格间距初步搜索结果

准确率	泛化误差	参数最佳组合
0.736 28	0.024 51	$m_{\text{d}}=5, m_{\text{n}}=30$
0.769 58	0.021 87	$m_{\text{d}}=5, m_{\text{n}}=60$
0.787 63	0.019 06	$m_{\text{d}}=15, m_{\text{n}}=60$
0.790 35	0.018 53	$m_{\text{d}}=15, m_{\text{n}}=90$
0.793 96	0.017 96	$m_{\text{d}}=20, m_{\text{n}}=120$
0.796 23	**0.015 83**	$\boldsymbol{m_{\text{d}}=20, m_{\text{n}}=30}$
0.791 84	0.017 34	$m_{\text{d}}=25, m_{\text{n}}=150$
0.787 05	0.015 77	$m_{\text{d}}=30, m_{\text{n}}=120$
0.784 79	0.018 90	$m_{\text{d}}=35, m_{\text{n}}=60$

可以看出初步的搜索结果最优为最大深度 20,最小样本数 30。当决策树最大深度 $m_{\text{d}}=20$,内部节点再划分所需最小样本数 $m_{\text{n}}=30$ 时,再计算此时的准确率为 0.796 23,为搜索峰值,所以继续以小间距细分网格,将决策树最大深度 m_{d} 的取值范围确定为 $8 < m_{\text{d}} < 31$,网格间距设定为 2;内部节点再划分所需最小样本数 m_{n} 的取值范围确定为 $18 < m_{\text{n}} < 41$,网格间距设定为 2。搜索结果如表 2 所示。

表 2　小网格间距初步搜索结果

准确率	泛化误差	参数最佳组合
0.795 68	0.015 91	$m_{\text{d}}=10, m_{\text{n}}=20$
0.807 21	0.014 37	$m_{\text{d}}=12, m_{\text{n}}=22$
0.819 65	0.012 95	$m_{\text{d}}=12, m_{\text{n}}=26$
0.837 82	0.013 72	$m_{\text{d}}=14, m_{\text{n}}=28$
0.853 97	**0.011 89**	$\boldsymbol{m_{\text{d}}=16, m_{\text{n}}=12}$
0.850 17	0.012 30	$m_{\text{d}}=16, m_{\text{n}}=28$
0.847 01	0.013 72	$m_{\text{d}}=18, m_{\text{n}}=36$
0.831 93	0.014 29	$m_{\text{d}}=20, m_{\text{n}}=16$
0.827 62	0.015 69	$m_{\text{d}}=22, m_{\text{n}}=12$

再次细分网格的搜索结果为决策树最大深度 $m_{\text{d}}=16$,内部节点再划分所需最小样本数 $m_{\text{n}}=12$ 时,计算此时的准确率为 0.853 97,已经趋于稳定。由此可初步确定 m_{d} 和 m_{n} 两个参数的取值,另外本文其他模型的其他参数也通过改进的网格搜索算法,先大范围搜索确定目标点,再小间距细分网格避免遗落最优点,最后求得最优参数。

4.3　性能评价标准

本文主要使用了模型的 F1 值、误报率(FPR)、精确率(Pre)和 AUC(Area under the Curve of ROC)值对模型进行评估。F1 值是召回率和准确率的一种加权平均[23]。AUC 值是 ROC 曲线下面积值,之所以采用 AUC 值是因其可以更好更直观地体现出 ROC 曲线所表达的结果,模型分类能力一般采用 ROC 曲线表示,模型的 ROC 曲线越高 AUC 值就越大则预测精度就越高[24]。

4.4　实验结果分析

本文所有的实验在 CPU 为 2 CPU Cores i5-8265U,1.80 GHz,内存为 8 GB 的计算机上运行,使用 Python 3.5 构建代码进行实验。采用 OpenDaylight 作为控制器,Mininet 在 VMware Workstation pro 上运行的 Ubuntul 6.04 操作系统进行。为了增加对比性,将改进的随机森林算法、随机森林算法和 XGBoost 三种不同的模型来进行评估。通过对比三种模型运行出的不同结果来评估改进方法的效果。对比结果如表 3 所示。

表 3 实验结果对比

模型	精确率	召回率	F1 值	AUC 值	FPR
改进 RF	0.95	0.97	0.79	0.97	1.02
RF	0.82	0.89	0.49	0.79	3.82
XGBoost	0.80	0.88	0.40	0.82	3.56
BHSVM	0.75	0.83	0.46	0.72	3.74

由表 3 可知,使用 XGBoost 算法得出的结果稍差,其中 AUC 值为 0.82,说明模型的分类效果尚可;F1 值为 0.40,低于随机森林算法和 SVM 算法,分数较低效果并不理想;精确率为 0.80 相对于另外两个随机森林类模型来说较低,但优于 SVM 算法。通过实验结果对比,说明随机森林算法在进行入侵检测时效果更好,而改进后的随机森林算法对比传统随机森林算法,FPR 明显降低,通过对随机森林的改进,在精确率、FPR、F1 值和 AUC 值三个模型的评价指标方面均有较为明显的改善。

为进一步验证改进算法的有效性,分别计算了入侵检测过程各类别检测精确度,实验结果见表 4。

表 4 各类别检测精确度对比

模型	Probe	Normal	U2R	R2L	DoS
改进 RF	0.98	0.97	0.91	0.89	0.99
RF	0.86	0.84	0.73	0.76	0.89

由表 4 可知,与原算法相比,改进之后的随机森林算法的入侵检测精确度在 Probe、Normal、U2R、R2L、DoS 几类攻击上均有明显提升,说明该方法具有良好的实际应用效果。

通过观察 ROC 曲线和 AUC 曲线下面积来对比改进后的随机森林算法、传统随机森林算法、XGBoost 算法和 SVM 算法,其中:虚线为改进后的随机森林算法的 ROC 曲线,XGBoost 算法的 ROC 曲线用点线表示,实线代表随机森林算法,点虚线代表 SVM 算法。从图 2 中可以看出改进之后的随机森林算法 ROC 曲线明显高于另外三种算法的曲线,AUC 值(曲线下面积)更大,效果更优。说明模型的泛化能力较好。

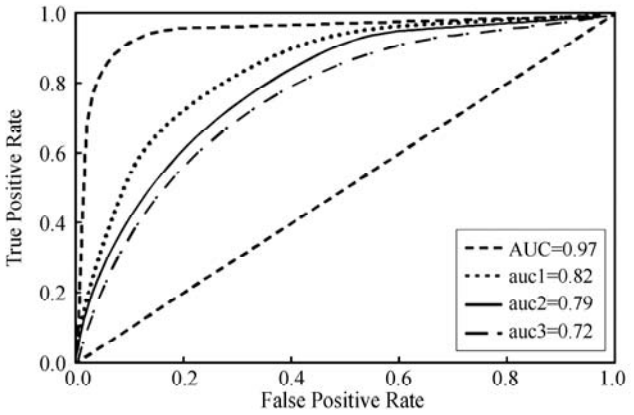

图 2 ROC 曲线对比

另外,为了对比随机森林算法改进前后的效果,在此额外引入 KS(Kolmogorov-Smirnov)值和 KS 曲线(此处不加入 XGBoost 和 SVM 算法作比较的目的是突出对比同种算法改进后的差异)。KS 值越大,说明模型的预测准确性越好,KS 大于 0.2 时,模型就有较好的预测准确性。通过对比 KS 值和 KS 曲线可以体现出经过改进的随机森林算法比原始的算法在风险区分能力方面有提高,改进前的 KS 值为 0.46,改进后的 KS 值为 0.58。图 3 为传统随机森林算法的 KS 曲线。图 4 为改进后 KS 曲线对比,其中实线和点虚线为随机森林算法的 KS 曲线,点画线和虚线为改进之后的随机森林算法的 KS 曲线。

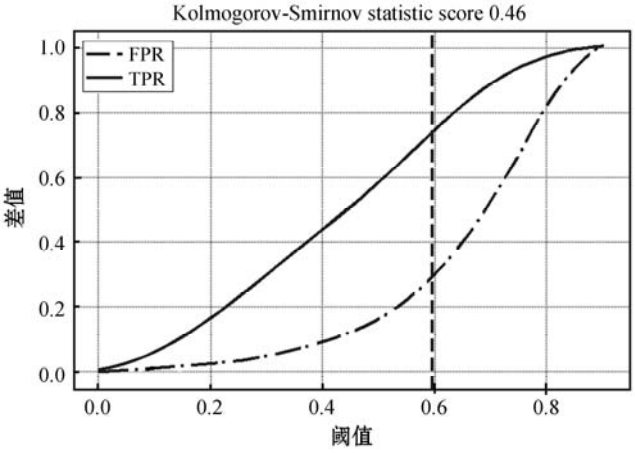

图 3 随机森林算法 KS 曲线

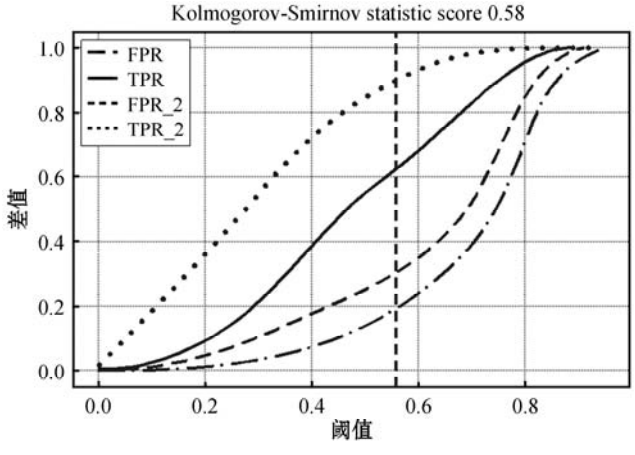

图 4 KS 曲线对比

通过实验结果对比分析,经过改进的随机森林算法精确率为 0.95,F1 值为 0.79,AUC 值为 0.97。改进之后的算法总体优于传统的随机森林算法并且同样优于 XGBoost 和 SVM 算法。说明改进的随机森林算法使 SDN 入侵检测的精确度得到了有效提高,实验结果验证了本文方法的有效性。

5 结 语

在针对入侵数据流特征有较大的差异性、降低入

侵检测的漏报率和误报率时,本文提出一种改进的随机森林算法,可以有效解决数据特征差异性较大影响入侵检测准确率的问题,降低入侵检测的误报率,提高检测精度。另外,在算法的改进上,增强随机森林算法在入侵检测领域的适应性和准确性,今后的研究方向在保证检测精度的同时,提高算法检测效率,优化检测模型。

参 考 文 献

[1] Jeong C, Ha T, Narantuya J, et al. Scalable network intrusion detection on virtual SDN environment[C]//2014 IEEE 3rd International Conference on Cloud Networking,2014:264 – 265.

[2] Shu J, Zhou L, Zhang W, et al. Collaborative intrusion detection for VANETs: A deep learning-based distributed SDN approach[J]. IEEE Transactions on Intelligent Transportation Systems,2021,22(7):4519 – 4530.

[3] Schueller Q, Basu K, Younas M, et al. A hierarchical intrusion detection system using support vector machine for SDN network in cloud data center[C]//2018 28th International Telecommunication Networks and Applications Conference,2018:1 – 6.

[4] Tang T, Mhamdi L, McLernon D, et al. Deep recurrent neural network for intrusion detection in SDN-based networks [C]//2018 4th IEEE Conference on Network Softwarization and Workshops,2018:202 – 206.

[5] Boukria S, Guerroumi M. Intrusion detection system for SDN network using deep learning approach[C]//2019 International Conference on Theoretical and Applicative Aspects of Computer Science,2019:1 – 6.

[6] Perwira R, Fauziah Y, Mahendra I, et al. Anomaly-based intrusion detection and prevention using adaptive boosting in software-defined network[C]//2019 5th International Conference on Science in Information Technology,2019:188 – 192.

[7] Ali A, Yousaf M. Novel three-tier intrusion detection and prevention system in software defined network[J]. IEEE Access,2020,8:109662 – 109676.

[8] Zhou D, Yan Z, Liu G, et al. An adaptive network data collection system in SDN[J]. IEEE Transactions on Cognitive Communications and Networking,2020,6(2):562 – 574.

[9] Malik J, Akhunzada A, Bibi I, et al. Hybrid deep learning: An efficient reconnaissance and surveillance detection mechanism in SDN[J]. IEEE Access,2020,8:134695 – 134706.

[10] Lee T, Chang L, Syu C. Deep learning enabled intrusion detection and prevention system over SDN networks[C]// 2020 IEEE International Conference on Communications Workshops,2020:1 – 6.

[11] Wang P, Chao K, Lin H, et al. An efficient flow control approach for SDN-based network threat detection and migration using support vector machine[C]//2016 IEEE 13th International Conference on e-Business Engineering,2016:56 – 63.

[12] Dey S, Rahman M. Flow based anomaly detection in software defined networking: A deep learning approach with feature selection method[C]//2018 4th International Conference on Electrical Engineering and Information and Communication Technology,2018:630 – 635.

[13] Scaranti G, Carvalho L, Barbon S, et al. Artificial immune systems and fuzzy logic to detect flooding attacks in software-defined networks[J]. IEEE Access,2020,8:100172 – 100184.

[14] 徐伟,冷静. 基于人工蜂群算法和 XGBoost 的网络入侵检测方法研究[J]. 计算机应用与软件,2021,38(3):314 – 318,333.

[15] 李兆斌,韩禹,魏占祯,等. SDN 中基于机器学习的网络流量分类方法研究[J]. 计算机应用与软件,2019,36(5):75 – 79.

[16] Breiman L. Random forests[J]. Machine Learning,2001,45:5 – 32.

[17] Ye X, Dong L, Ma D. Loan evaluation in P2P lending based on random forest optimized by genetic algorithm with profit score[J]. Electronic Commerce Research and Applications,2018,32:23 – 36.

[18] Chaudhary A, Kolhe S, Kamal R. An improved random forest classifier for multi-class classification[J]. Information Processing in Agriculture,2016,3(4):215 – 222.

[19] Zhang X, Yang Y, Zhou Z. A novel credit scoring model based on optimized random forest[C]//2018 IEEE 8th Annual Computing and Communication Workshop and Conference,2018:60 – 65.

[20] Pedregosa F, Gramfort A, Michel V, et al. Scikit-learn: Machine learning in python[J]. Journal of Machine Learning Research,2011,12:2825 – 2830.

[21] Malhotra R, Jha M, Poss M, et al. A random forest classifier for detecting rare variants in NGS data from viral populations[J]. Computational and Structural Biotechnology Journal,2017,15:388 – 395.

[22] Pérez-Díaz J, Valdovinos I, Choo K, et al. A flexible SDN-based architecture for identifying and mitigating low-rate DDoS attacks using machine learning[J]. IEEE Access,2020,8:155859 – 155872.

[23] Assef F, Steiner M, Steiner Neto P, et al. Classification algorithms in financial application: Credit risk analysis on legal entities[J]. IEEE Latin America Transactions,2019,17(10):1733 – 1740.

[24] Adnan M, Islam M. Optimizing the number of trees in a decision forest to discover a subforest with high ensemble accuracy using a genetic algorithm[J]. Knowledge-Based Systems,2016,110:86 – 97.