基于KELM选择性集成的复杂网络环境入侵检测

刘金平[1,2],何捷舟[1],马天雨[3],张五霞[1],唐朝晖[4],徐鹏飞[1]
(1.湖南师范大学智能计算与语言信息处理湖南省重点实验室,湖南长沙 410081;2.湖南师范大学计算与随机数学教育部重点实验室,湖南长沙 410081;3.湖南师范大学物理与电子科学学院,湖南长沙 410081;4.中南大学信息科学与工程学院,湖南长沙 410083)

摘 要: 为解决复杂网络环境网络入侵事件特征复杂多变、新型入侵检测度低、检测时间长、难以实现实时检测的问题,本文提出一种基于核极限学习机(Kernel Extreme Learning Machine,KELM)选择性集成的网络入侵检测方法(SEoKELM-NID).该方法采用Bagging策略独立快速训练出多个KELM子学习器;然后基于边缘距离最小化(Margin Distance Minimization,MDM)准则对KELM子学习器的集成增益进行度量,通过选择增益度高的部分KELM子学习器进行选择性集成,获得泛化能力强、效率高的选择性集成学习器;同时,引入一种基于批量样本增量学习的KELM子分类器在线更新策略,实现入侵检测模型的在线更新,使SEoKELM-NID能有效适应复杂网络环境的变化.在KDD99数据集和一个以太网和无线网络混合的复杂网络仿真实验平台上进行了仿真实验验证,结果表明,SEoKELM-NID相比基于单个学习器以及传统集成学习的网络入侵检测方法具有更好的识别准确性以及更快的识别速度,特别对于未知的网络入侵连接事件响应速度快、漏报率低.

关键词: 网络入侵检测;极限学习机(ELM);异常检测;选择性集成学习;边缘距离最小化
中图分类号: TP391 **文献标识码:** A **文章编号:** 0372-2112 (2019)05-1070-09
电子学报URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2019.05.014

Selective Ensemble of KELM-Based Complex Network Intrusion Detection

LIU Jin-ping[1,2], HE Jie-zhou[1], MA Tian-yu[3], ZHANG Wu-xia[1], TANG Zhao-hui[4], XU Peng-fei[1]
(1. *Hunan Provincial Key Laboratory of Intelligent Computing and Language Information Processing, Hunan Normal University, Changsha, Hunan* 410081, *China*;
2. *Key Laboratory of Computing and Stochastic Mathematics (Ministry of Education), Hunan Normal University, Changsha, Hunan* 410081, *China*;
3. *School of Physics and Electronics, Hunan Normal University, Changsha, Hunan* 410081, *China*;
4. *School of Information Science and Engineering, Central South University, Changsha, Hunan* 410083, *China*)

Abstract: To solve the problem of the low detection accuracy of new intrusions with long detection time due to the complex and changeable nature of network intrusions, this paper proposes a network intrusion detection method based on the selective learning of Kernel Extreme Learning Machines (KELMs). First, based on the high efficiency learning characteristics of the single KELM learner, multiple KELMs are trained independently by the Bagging strategy. Then, based on the margin distance minimization (MDM) guidelines, KELM learners are integrated by selecting a part of them with high gains based on the MDM-based gain measures. Extensive validation and comparative experiments on the the KDD99 data set and on a hybrid network simulation platform mixed with wireless networks and Ethernet networks demonstrate that the proposed method achieves better recognition accuracies with faster recognition speed than the network intrusion detection methods based on the single learner and the traditional ensemble learning, which can effectively detect the known and unknown network intrusion connection in real time.

Key words: network intrusion detection; extreme learning machine (ELM); anomaly detection; selective ensemble learning; margin distance minimization

收稿日期:2018-08-13;修回日期:2018-10-16;责任编辑:诸叶梅
基金项目:国家自然科学基金(No.61501183,No.61771492,No.61472134);国家自然科学基金-广东联合基金重点项目(No.U1701261);湖南省自然科学基金(No.2018JJ3349);湖南省研究生科研创新项目(No.CX2018B312)

1 引言

随着大数据时代的到来,网络信息面临着越来越多的安全威胁. 网络入侵检测作为一种主动的安全防护技术,希望在网络系统受到危害之前拦截和响应入侵,受到国内外广泛重视.

传统的入侵检测包括误用检测和异常检测. 误用检测通过建立入侵规则库来匹配发现网络中异常链接,虽然准确率高,但对新类型入侵以及旧病毒变种连接往往无能为力[1]. 异常检测通过总结正常网络连接特征用于网络异常分析,由于该方法对于新型攻击也有较好的检测效果,因而广受关注[2].

面对日益复杂的网络环境,无论是基于误用还是基于异常的入侵检测系统往往占用资源多、检测速度慢,需要人工干预的缺陷日益突出. 当异常访问或连接事件被检测处理之后,很可能已经产生严重的后果. 高效快速的入侵检测仍是一项极具有挑战性的任务[3]. 一个性能良好的入侵检测系统必须能进行自我学习、自我适应,最终能以较快的速度、较低的误报率和漏报率进行各种违规连接报警. 因此,近年来基于机器学习的网络入侵检测方法,因其自适应性强、智能度高受到国内外研究者广泛关注[4].

基于机器学习的网络入侵检测是将网络入侵检测转化为模式识别(分类)问题. 采用机器学习的方法训练分类器,比如支持向量机(SVM)[5]、极限学习器(ELM)[6]、决策树(DMT)[7]等,来对网络连接中的正常行为和异常行为进行鉴别. 基于机器学习的入侵检测方法因学习(分类)器选择的不同而各有优劣. 总体来说,复杂的分类器训练与检测时间相对较长,而简单的学习器虽然处理效率高,但对于多特征混合、入侵方式复杂多变的网络攻击事件,则可能很难获得有效的识别效果. 一些研究者希望通过综合多个学习器的优点以获得更好的检测性能. 因而,近年来基于集成学习的入侵检测方法受到广泛的关注[8,9].

集成学习算法通过集成多个子学习器来提升整体算法的泛化能力. 基于集成学习的入侵检测对于未知网络攻击的识别效果理论上要远优于基于单个学习器的入侵检测方法. 然而,对于集成学习来说,并不是每一个子学习器都是有利的,如果能选择出其中性能良好的部分子学习器进行选择性集成,将使集成学习器具有更好的性能,同时提高检测效率.

本文针对复杂网络环境中,网络连接复杂多变、网络攻击方式层出不穷、现有的网络入侵检测方法存在着识别速度慢、对于新型入侵模式识别率低等问题,基于 KELM 快速学习的优点,采用 Bagging 学习策略提出一种基于 KELM 选择性集成学习的网络入侵检测方法(SEoKELM-NID). SEoKELM-NID 通过细分每一个子 KELM 对集成检测器的增益度来对部分子学习器进行选择性集成;同时该方法能根据网络环境的变化对模型进行在线更新,以提高该方法对已知入侵类型的检测效率和降低对未知入侵类型的漏报率和误报率. 采用传统的 KDD99 数据集和一个手动搭建的包括以太网和无线网络混杂的复杂网络物理仿真平台验证了所提方法的有效性.

2 基于 KELM 选择性集成的入侵检测

网络入侵检测本质上是一个多变量模式分类问题[1]. 假设收集了 n 条网络连接数据集 $\boldsymbol{X}$,$\boldsymbol{X}=\{X_i \mid i=1,2,\cdots,n\}^{\mathrm{T}}\in\mathbb{R}^{n\times K}$,其中 $X_i\in\mathbb{R}^{K}$ 代表第 i 条网络连接数据记录,K 代表网络连接记录数据特征的维数,n 为收集的样本条数;这些记录对应的网络连接类型的标记为 $\boldsymbol{T}=\{T_i \mid i=1,\cdots,n\}^{\mathrm{T}}$. 那么基于单学习器的入侵检测模型为

$$G_j=\min_{G_j}\{\|\boldsymbol{T}-G_j(x_i)\|_2^2+\lambda\varphi(G_j)\} \tag{1}$$

式(1)中,$\varphi(\cdot)$为正则项,用来控制模型的复杂程度,λ 为相应的权重系数,如果 G_j 的结构是已经确定了的,那么式(1)相当于通过调整分类器 G_j 中的参数以获得一个最优的分类器[10].

本文提出的 SEoKELM-NID 方法基于 Bagging 学习策略,先并行训练一批具有一定互补功能的子学习器,然后采用边缘距离最小化准则(MDMC)对子学习器进行选择性学习. 该方法的流程示意图如图 1 所示.

2.1 KELM 子分类器

ELM[11,12]是南洋理工大学黄广斌教授提出的一种简单易用、有效的单隐层前馈神经网络. ELM 本质上可以归结为一个线性参数方程,通过对该线性系统求解可以获得一个闭式的(Close-form)理论上的全局最优解.

2.1.1 带正则项约束的 ELM

假设有 N 个网络连接数据样本$\{(X_i,T_i) \mid i=1,\cdots,N\}$,其中 $X_i\in\mathbb{R}^{n}$ 表示网络连接数据特征,$T_i\in\mathbb{R}^{m}$ 表示各网络连接数据记录对应的标签(其中只有一个变量为 1,其余的变量为 0,对应的非零数量的位置代表相应的网络连接类型).

一个有 L 个隐层节点的 ELM,其输出可以写成

$$O_j=\sum_{i=1}^{L}\beta_i h_i(X_j)=\boldsymbol{h}(X_j)\boldsymbol{\beta};\ j=1,\cdots,m \tag{2}$$

其中,$\boldsymbol{\beta}=(\beta_1,\cdots,\beta_L)^{\mathrm{T}}\in\mathbb{R}^{L\times m}$ 为输出层权值向量,$\boldsymbol{h}(X)=(h_1(X),\cdots,h_L(X))$为 X 的 ELM 非线性映射($h_i(X)$代表第 i 个隐含神经元的输出),

$$h_i(X)=g(W_i,b_i,X) \tag{3}$$

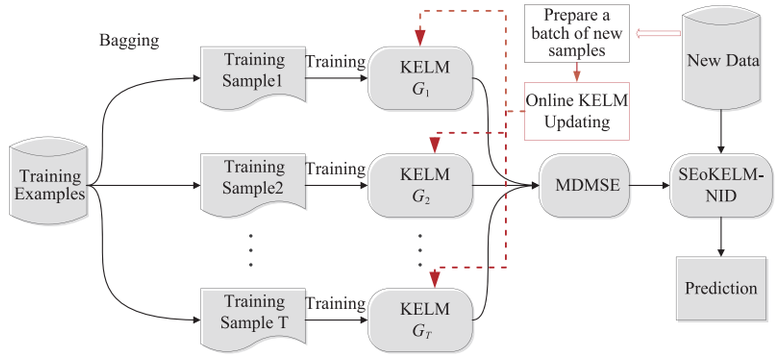

图1 SEoKELM-NID算法流程

其中 $g(\cdot)$ 代表隐含层的激活函数. 常用的激活函数有 sigmoid 函数,双曲正切函数,径向基函数等. $\boldsymbol{W}_i$ 和 $\boldsymbol{b}_i$ 分别代表第 i 个隐层单元的输入权重和偏置,且输入权重 $\boldsymbol{W}_i$ 和偏置 $\boldsymbol{b}_i$ 经随机初始化后不需要迭代修改.

基本的 ELM 学习器通过求解如下线性优化问题来获得输出层参数

$$\boldsymbol{\beta}_{\mathrm{BELM}}=\min_{\boldsymbol{\beta}}\|\boldsymbol{H}\boldsymbol{\beta}-\boldsymbol{T}\|_F^2 \tag{4}$$

式(4)中下标 F 代表 Frobenius 范数,$\boldsymbol{H}$ 是隐含层的输出矩阵,

$$\boldsymbol{H}=\begin{bmatrix}\boldsymbol{h}(X_1)\\ \vdots\\ \boldsymbol{h}(X_N)\end{bmatrix}=\begin{bmatrix}h_1(X_1) & \cdots & h_L(X_1)\\ \vdots & \ddots & \vdots\\ h_1(X_N) & \cdots & h_L(X_N)\end{bmatrix} \tag{5}$$

$\boldsymbol{T}=[T_1,T_2,\cdots,T_N]^{\mathrm{T}}$ 是训练数据集的目标(标签)矩阵,求解优化问题(4),可得

$$\boldsymbol{\beta}_{\mathrm{BELM}}=\boldsymbol{H}^{\dagger}\boldsymbol{T} \tag{6}$$

其中$\boldsymbol{H}^{\dagger}$ 代表矩阵 $\boldsymbol{H}$ 的 Moore-Penrose 广义逆.

为了提高 ELM 的泛化性能,通常在考虑逼近误差的同时,增加一个正则项,以获得更好的性能. 基于此原则,式(4)可以调整为一个通用的带正则项的 ELM 模型[13],

$$\boldsymbol{\beta}_{\mathrm{RELM}}=\min_{\boldsymbol{\beta}}\{\|\boldsymbol{H}\boldsymbol{\beta}-\boldsymbol{T}\|_q^{\sigma_1}+\lambda\|\boldsymbol{\beta}\|_p^{\sigma_2}\} \tag{7}$$

其中 $\sigma_1>0,\sigma_2>0,p$ 和 q 可以取 $0,1/2,1,2,\cdots,\infty$ 等各种范数. 如果 $p=2,q=F,\sigma_1=\sigma_2=2$,并且训练样本数 N 大于特征数 L 时,

$$\boldsymbol{\beta}_{\mathrm{RELM}}=\boldsymbol{H}^{\mathrm{T}}\left(\frac{\boldsymbol{I}}{\lambda}+\boldsymbol{H}\boldsymbol{H}^{\mathrm{T}}\right)^{-1}\boldsymbol{T} \tag{8}$$

获得 ELM 分类器模型参数 $\boldsymbol{\beta}_{\mathrm{RELM}}$ 后,对于多类别的网络入侵检测问题,如果给定一个测试样本 X_{test},其对应的网络入侵类型 j 为 $\boldsymbol{h}(X_{\mathrm{test}})\boldsymbol{\beta}$ 类型向量最大值对应的位置所对应的网络入侵类型,即,

$$j=\max_j\{\boldsymbol{h}(X_{\mathrm{test}})\boldsymbol{\beta}\} \tag{9}$$

2.1.2 KELM 分类器

式(6)或式(8)所描述的 ELM 输出层参数向量与隐含层的特征映射函数有直接的关联. 如果想避免隐含层特征映射函数对 ELM 模型的影响,那么 KELM 模型是一个更好的选择[12,14].

根据 $\boldsymbol{\beta}_{\mathrm{RELM}}$ 的计算结果式(8),给定一条新的网络连接记录 X_{test},其对应的连接类型向量可以表示为:

$$y_{\mathrm{test}}=\boldsymbol{h}(X_{\mathrm{test}})\boldsymbol{\beta}_{\mathrm{RELM}}=\boldsymbol{h}(X_{\mathrm{test}})\boldsymbol{H}^{\mathrm{T}}\left(\frac{\boldsymbol{I}}{\lambda}+\boldsymbol{H}\boldsymbol{H}^{\mathrm{T}}\right)^{-1}\boldsymbol{T} \tag{10}$$

其中 $\boldsymbol{H}\boldsymbol{H}^{\mathrm{T}}$ 可以用 ELM 核矩阵表示.

令 $\boldsymbol{\Omega}=\boldsymbol{H}\boldsymbol{H}^{\mathrm{T}}$ 代表 ELM 的核矩阵,其中$\boldsymbol{\Omega}_{ij}=\boldsymbol{h}(X_i)\boldsymbol{h}(X_h)^{\mathrm{T}}$ 可以用核函数 $K(X_i,X_j)$ 来表示. 将核矩阵代入式(10),可得

$$\begin{aligned}y_{\mathrm{test}}&=\boldsymbol{h}(X_{\mathrm{test}})(\boldsymbol{h}^{\mathrm{T}}(X_1),\cdots,\boldsymbol{h}^{\mathrm{T}}(X_N))^{\mathrm{T}}\left(\frac{\boldsymbol{I}}{\lambda}+\boldsymbol{\Omega}\right)^{-1}\boldsymbol{T}\\&=\begin{pmatrix}K(X_{\mathrm{test}},X_1)\\ \vdots\\ K(X_{\mathrm{test}},X)\end{pmatrix}\left(\frac{\boldsymbol{I}}{\lambda}+\underbrace{\begin{pmatrix}K(X_1,X_1) & \cdots & K(X_1,X_N)\\ \vdots & \ddots & \vdots\\ K(X_N,X_1) & \cdots & K(X_N,X_N)\end{pmatrix}}_{\boldsymbol{\Omega}}\right)^{-1}\boldsymbol{T}\end{aligned} \tag{11}$$

式(11)为 KELM 表达公式,其不需要显式提供隐含层的特征映射函数 $\boldsymbol{h}(\cdot)$,也不需要给定隐含层神经元的个数 L.

2.1.3 KELM 分类器在线增量式更新

由于网络入侵方式复杂多变,仅依据以往历史网络连接数据集获得的分类器模型,将难以适应新的网络入侵方式(类型)的变化. 因此,有必要对所获得的分类器模型(随时/实时)进行更新.

本文受 Liang 等[15]提出的基于递归最小二乘的在线顺序 ELM 更新算法(Online sequential ELM, OS-ELM)和杨乐[16]等人提出的基于核函数的在线序列 ELM 模型的启发,先基于历史数据集训练一批 KELM 子学习器模型;然后,在实际的网络安全监控中,当收集到一批新的具有比较明确标签的样本时(样本标签可以基于手工标记或者基于多分类器的共同决策结果),再对各个 KELM 子分类器模型进行在线更新.

KELM 子学习器模型在线更新的主要步骤如下:

(1)首先根据初始训练样本($\boldsymbol{\Pi}_0=\{X_i,T_i\}_{i=1}^{N_0}$,$N_0$ 为样本数目,并设这些初始样本的特征映射矩阵为$\boldsymbol{H}_0$,对应的标签矩阵为$\boldsymbol{T}_0$)获得初始 KELM 分类器模型$G^0(X)$.

根据式(11)可知 $G^0(x)=K_0\,\boldsymbol{\Omega}_0^{-1}\,\boldsymbol{T}_0$,其中 $\boldsymbol{K}_0=(K(X_1,x),K(X_2,x),\cdots,K(X_{N_0},x))^{\mathrm{T}}$,$\boldsymbol{\Omega}_0=\boldsymbol{I}_0/\lambda+\boldsymbol{\Omega}_{\mathrm{KELM}}^0$,$\boldsymbol{\Omega}_{\mathrm{KELM}}^0=\boldsymbol{H}_0\,\boldsymbol{H}_0^{\mathrm{T}}$.

(2)在网络入侵检测过程中,如果新获得了一批具有明确标签的样本 $\Delta\boldsymbol{\Pi}_1=\{X_i,T_i\}_{i=N_0+1}^{N_0+N_1}$($N_1$ 为新收集样本数目),此时综合考虑初始样本集$\boldsymbol{\Pi}_0$和新收集样本 $\Delta\boldsymbol{\Pi}_1$ 所构成的大样本集$\boldsymbol{\Pi}_1=\{\boldsymbol{\Pi}_0,\Delta\boldsymbol{\Pi}_1\}$,其对应的 KELM 分类器模型 $G^1(X)$可以表示为:

$$\begin{aligned}G^1(X)&=K_1\,\boldsymbol{\Omega}_1^{-1}\,\boldsymbol{T}_1\\&=\begin{pmatrix}K_0\\ \Delta K_1\end{pmatrix}\left(\frac{\boldsymbol{I}_1}{\lambda}+\begin{pmatrix}\boldsymbol{H}_0 & \Delta\boldsymbol{H}_{01}\\ \Delta\boldsymbol{H}_{10} & \Delta\boldsymbol{H}_1\end{pmatrix}\begin{pmatrix}\boldsymbol{H}_0^{\mathrm{T}} & \Delta\boldsymbol{H}_{10}^{\mathrm{T}}\\ \Delta\boldsymbol{H}_{01}^{\mathrm{T}} & \Delta\boldsymbol{H}_1^{\mathrm{T}}\end{pmatrix}\right)^{-1}\\&\quad\cdot\begin{pmatrix}\boldsymbol{T}_0\\ \Delta\boldsymbol{T}_1\end{pmatrix}\\&=\begin{pmatrix}K_0\\ \Delta K_1\end{pmatrix}\begin{pmatrix}\boldsymbol{I}_0/\lambda+\boldsymbol{\Omega}_{\mathrm{KELM}}^0 & \Delta\boldsymbol{\Omega}_{\mathrm{KELM}}^{01}\\ \Delta\boldsymbol{\Omega}_{\mathrm{KELM}}^{10} & \Delta\boldsymbol{I}_1/\lambda+\Delta\boldsymbol{\Omega}_{\mathrm{KELM}}^1\end{pmatrix}^{-1}\\&\quad\cdot\begin{pmatrix}\boldsymbol{T}_0\\ \Delta\boldsymbol{T}_1\end{pmatrix}\end{aligned}\tag{12}$$

其中 $\Delta\boldsymbol{K}_1=(K(X_{N_0+1},X),\cdots,K(X_{N_0+N_1},X))^{\mathrm{T}}$ 代表 X 在新训练样本的核映射,$\Delta\boldsymbol{\Omega}_{\mathrm{KELM}}^1=\Delta\boldsymbol{H}_1\,\boldsymbol{H}_1^{\mathrm{T}}\Delta\boldsymbol{T}_1=(T_{N_0+1},\cdots,T_{N_0+N_1})^{\mathrm{T}}$,$\Delta\boldsymbol{\Omega}_{\mathrm{KELM}}^{01}=\boldsymbol{H}_0\Delta\boldsymbol{H}_1^{\mathrm{T}}$,$\Delta\boldsymbol{\Omega}_{\mathrm{KELM}}^{10}=(\Delta\boldsymbol{\Omega}_{\mathrm{KELM}}^{01})^{\mathrm{T}}=\Delta\boldsymbol{H}_1\,\boldsymbol{H}_0^{\mathrm{T}}$,具体来讲

$$\Delta\boldsymbol{\Omega}_{\mathrm{KELM}}^{01}=\begin{pmatrix}K(X_1,X_{N_0+1}) & \cdots & K(X_1,X_{N_0+N_1})\\ \vdots & \ddots & \vdots\\ K(X_{N_0},X_{N_0+1}) & \cdots & K(X_{N_0},X_{N_0+N_1})\end{pmatrix}\in\mathbb{R}^{N_0\times N_1}$$

$$\Delta\boldsymbol{\Omega}_{\mathrm{KELM}}^{10}=\begin{pmatrix}K(X_{N_0+1},X_1) & \cdots & K(X_{N_0+1},X_{N_0})\\ \vdots & \ddots & \vdots\\ K(X_{N_0+N_1},X_1) & \cdots & K(X_{N_0+N_1},X_{N_0})\end{pmatrix}\in\mathbb{R}^{N_1\times N_0}$$

$$\Delta\boldsymbol{\Omega}_{\mathrm{KELM}}^{1}=\begin{pmatrix}K(X_{N_0+1},X_{N_0+1}) & \cdots & K(X_{N_0+1},X_{N_0+N_1})\\ \vdots & \ddots & \vdots\\ K(X_{N_0+N_1},X_{N_0+1}) & \cdots & K(X_{N_0+N_1},X_{N_0+N_1})\end{pmatrix}\in\mathbb{R}^{N_1\times N_1}$$

如果令 $\boldsymbol{L}_0=\boldsymbol{I}_0/\lambda+\boldsymbol{\Omega}_{\mathrm{KELM}}^0$,$\boldsymbol{L}_1=\Delta\boldsymbol{I}_1/\lambda+\Delta\boldsymbol{\Omega}_{\mathrm{KELM}}^1$,根据

$$\begin{pmatrix}\boldsymbol{A} & \boldsymbol{B}\\ \boldsymbol{C} & \boldsymbol{D}\end{pmatrix}^{-1}=\begin{pmatrix}\boldsymbol{A}^{-1}+\boldsymbol{A}^{-1}\boldsymbol{B}\boldsymbol{M}\boldsymbol{C}\boldsymbol{A}^{-1} & -\boldsymbol{A}^{-1}\boldsymbol{B}\boldsymbol{M}\\ -\boldsymbol{M}\boldsymbol{C}\boldsymbol{A}^{-1} & \boldsymbol{M}\end{pmatrix}$$

其中 $\boldsymbol{M}=(\boldsymbol{D}-\boldsymbol{C}\boldsymbol{A}^{-1}\boldsymbol{B})^{-1}$那么

$$\boldsymbol{\Omega}_1^{-1}=\begin{pmatrix}\boldsymbol{L}_0 & \Delta\boldsymbol{\Omega}_{\mathrm{KELM}}^{01}\\ \Delta\boldsymbol{\Omega}_{\mathrm{KELM}}^{10} & \boldsymbol{L}_1\end{pmatrix}^{-1}$$

$$=\begin{pmatrix}\boldsymbol{L}_0^{-1}+\boldsymbol{D}_1\,\boldsymbol{R}_{\mathrm{KELM}}\boldsymbol{D}_1^{\mathrm{T}} & -\boldsymbol{D}_1\,\boldsymbol{R}_{\mathrm{KELM}}\\ -\boldsymbol{R}_{\mathrm{KELM}}\boldsymbol{D}_1^{\mathrm{T}} & \boldsymbol{R}_{\mathrm{KELM}}\end{pmatrix}\tag{13}$$

其中 $\boldsymbol{R}_{\mathrm{KELM}}=(\boldsymbol{L}_1-\Delta\boldsymbol{\Omega}_{\mathrm{KELM}}^{10}\,\boldsymbol{L}_0^{-1}\Delta\boldsymbol{\Omega}_{\mathrm{KELM}}^{01})^{-1}$,$\boldsymbol{D}_1=\boldsymbol{L}_0^{-1}\cdot\Delta\boldsymbol{\Omega}_{\mathrm{KELM}}^{01}$.

将式(13)代入式(12)可得增量更新 KELM 模型 $G^1(X)$为

$$\begin{aligned}G^1(X)&=K_1(X)\boldsymbol{\Omega}_1^{-1}\,\boldsymbol{T}_1\\&=K_0\boldsymbol{L}_0^{\mathrm{T}}\,\boldsymbol{T}_0+K_0\,\boldsymbol{D}_1\,\boldsymbol{R}_{\mathrm{KELM}}\boldsymbol{D}_1^{\mathrm{T}}\,\boldsymbol{T}_0-\Delta K_1\,\boldsymbol{R}_{\mathrm{KELM}}\boldsymbol{D}_1^{\mathrm{T}}\,\boldsymbol{T}_0\\&\quad-K_0\,\boldsymbol{D}_1\,\boldsymbol{R}_{\mathrm{KELM}}\Delta\boldsymbol{T}_1+\Delta K_1\,\boldsymbol{R}_{\mathrm{KELM}}\Delta\boldsymbol{T}_1\\&=G^0(X)+(K_0\,\boldsymbol{D}_1-\Delta K_1)\boldsymbol{R}_{\mathrm{KELM}}(\boldsymbol{D}_1^{\mathrm{T}}\,\boldsymbol{T}_0-\Delta\boldsymbol{T}_1)\end{aligned}\tag{14}$$

(3)式(14)即为 KELM 的逐次批量更新公式. 假设前面已经实行了 k 次模型更新,获得模型参数为 $G^k(X)$,那么第 $k+1$ 次 KELM 分类器在线批量样本增量式更新准则为:

$$\begin{aligned}&G^{k+1}(X)=\\&G^k(X)+(\boldsymbol{K}_k\boldsymbol{D}_{k+1}-\Delta\boldsymbol{K}_{k+1})\boldsymbol{R}_{\mathrm{KELM}}(\boldsymbol{D}_{k+1}^{\mathrm{T}}\boldsymbol{T}_k-\Delta\boldsymbol{T}_{k+1})\end{aligned}\tag{15}$$

其中 $\boldsymbol{K}_k=(\boldsymbol{K}_1^{\mathrm{T}},\Delta\boldsymbol{K}_1^{\mathrm{T}},\cdots,\Delta\boldsymbol{K}_k^{\mathrm{T}})^{\mathrm{T}}$,$\boldsymbol{D}_{k+1}=\boldsymbol{L}_0^{-1}\Delta\boldsymbol{\Omega}_{\mathrm{KELM}}^{0k+1}$,$\Delta\boldsymbol{T}_{k+1}=(\boldsymbol{T}_{1+\sum_{i=1}^{k}N_i},\cdots,\boldsymbol{T}_{\sum_{i=1}^{k+1}N_i})$,$\boldsymbol{T}_k=(\boldsymbol{T}_0^{\mathrm{T}},\Delta\boldsymbol{T}_1^{\mathrm{T}},\cdots,\Delta\boldsymbol{T}_k^{\mathrm{T}})^{\mathrm{T}}$.

在实际应用中,随着批量更新的次数增多,式(15)中第二项中的矩阵(向量)会越来越大,为了保证计算的实效性,可以增加一个遗忘机制,用于舍弃一些时间比较久远的样本. 比如,规定最大的的样本数量为 $N_{\max}$,在第 j 次批量更新时,如果发现 $\sum_{i=1}^{j}N_i>N_{\max}$,那么根据样本的历史顺序,舍弃历史久远的 d 个样本,保证 $\sum_{i=1}^{j}N_i-d\leqslant N_{\max}$,从而既能有效更新 KELM 模型,也能保证计算的时效性.

2.2　基于 MDM 的 KELM 子学习器选择性集成(MDMSE)

集成学习将多个弱分类器通过一定的组合方式集成起来以形成新的强分类器算法,这类算法又称元算法(Meta - algorithm),最常见的诸如 Boosting[17] 和 Bagging[18]. Boosting 通过集中关注被已有分类器分类错误的样本对每个新创建的子学习器进行优化. Boosting 方法强调子学习器之间的强依赖关系,通过串行生成,代表算法有 Adaboost、XGBOOST、GBDT 等[19].

Bagging 即 Boostrap Aggregation,其中 Boostrap 是随机重采样,它提倡每一个子学习器都应尽可能的相互独立. 采用可同时生成的分布式并发计算方法进行建立,因此该方法能高效独立地并行训练一批子学习器,并使各个子学习器具有一定互补能力.

鉴于该 Bagging 策略具有并发学习的优点,本文采

用 Bagging 策略进行子分类器学习. 同时,为保证每一种异常入侵行为都能被最大化检测出来,本文提出一种基于边缘距离最小化的选择性集成(MDMSE)算法对所有子学习器进行增益度排序,通过选择增益度大的部分子学习器作为最终的集成结果,降低了弱学习器对最终检测结果的不利影响.

选择性集成学习是指从训练的所有子学习器中选择出差异性较大,泛化能力强的个体学习器加以集成,以获得更好的性能[20]. 研究表明,选择性集成算法优于单独的 Bagging 和 Boosting 集成算法.

论文[21]提出一种基于遗传算法的选择性学习算法 GASEN. 该方法首先采用 boostrap 方法对训练集进行采样,再对子学习器进行训练生成多个弱学习器;然后采用遗传算法求出各个学习器的最优权重向量. 该方法将子学习器的权重向量 w 作为遗传种群中的个体,并设置一个阈值对 w 的分量进行筛选,最后实现个体学习器的选择. 由于遗传算法以概率式方法求解,在寻优过程中,有一定的几率落入局部最优解,且庞大的运算量会增加计算与存储的开销.

为了尽可能避免陷入局部最优和提高计算效率,受论文[22]的启发,本文基于 MDM 原理提出一种新的选择性集成算法 MDMSE. 该方法基于 MDM 准则计算出每个子学习器对整体集成算法性能提升的增益度量,通过选择增益度高的 KELM 子学习器进行部分集成,以获得计算效率高、泛化能力强的强学习器. 下面从几个基本的定义出发来分析 MDMSE 的主要原理.

定义 1 分类器特征向量 $\boldsymbol{C}_t$

给定一个带标签的含有 n 个元素的数据集 D. 分类器 $\boldsymbol{G}_t$ 的特征向量 $\boldsymbol{C}_t$ 是一个 N 维的向量,其第 i 部分为

$$\boldsymbol{C}_t^i = 2 \oplus (h_t(x_i) == y_i) - 1, (x_i, y_i) \in \mathbb{Z} \quad (16)$$

其中 $h_t(x_i)$ 为分类器 t 在第 i 个样本上的判别结果,y_i 为第 i 个样本的标签,符号⊕表示两数之积. 当分类器 t 在数据集 D 的第 i 个样本上分类正确时 $\boldsymbol{C}_t^i$ 为 1,否则为 -1. 整体特征向量是所有分类器特征向量的和,这个特征向量的平均值为:

$$\overline{\boldsymbol{C}} = \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{C}_t \quad (17)$$

在二分类中,平均向量 $\overline{\boldsymbol{C}}$ 的第 i 部分是第 i 个例子的边界,定义为该示例接收到的正确和不正确的选票之间的差异,正则化在[-1,1]上. 对于多分类问题上,其等于(1 - 2edge(i)),其中 edge(i)对应于正确的类和其他类(异常类)的投票,其正规化范围为[-1,1][23].

定义 2 N 维空间优化点 $\boldsymbol{O}$

如果平均特征向量 $\overline{\boldsymbol{C}}$ 的第 i 部分是正值,则整体分类器在第 i 个例子上分类正确. 因此,如果一个部分集成分类器的特征向量在 N 维空间的第一象限(即所有分量都是正的),则在数据集 D 上分类都正确.

本文的目标就是选择出一个尽量小但其平均特征向量尽量接近第一象限中的某个参考位置的集成学习器. 选择任意目标位置 p 作为具有相同分量的点 $\boldsymbol{O}$,即:

$$O_i = p, \text{with } i = 1, \cdots\cdots, N \text{ and } 0 < p < 1 \quad (18)$$

定义 3 基于集成增益的子分类器迭代选择

每一次集成到集合中的分类器是那些从特征向量 $\overline{\boldsymbol{C}}$ 到目标点 $\boldsymbol{O}$ 的距离减少最多的分类器,即对集成增益最大的分类器. 因此,在第 u 次迭代中选择的分类器是:

$$S_u = \arg\min_k d\left(\boldsymbol{O}, \frac{1}{U}(\boldsymbol{C}_k + \sum_{t=1}^{u-1} \boldsymbol{C}_t)\right), k \in E_T \backslash S_{u-1} \quad (19)$$

其中 $d(v,u)$ 是点 v 与点 u 的欧几里得距离. S 可以看作当前子学习器对减少特征向量 $\overline{\boldsymbol{C}}$ 到目标点 $\boldsymbol{O}$ 的距离增益度,S 越小,增益度越大.

其中 p 设定为 $0 < p < 1$ 的任意常数. 理论上 p 应该足够小(比如 $P \sim 0.075$),以便简单的例子(那些被大多数子学习器正确分类的)能快速地接近 p 值. 随后,它们在选择下一个分类器时影响变小. 这允许算法逐渐聚焦于更难分类的样本. 相比之下,如果设定 p 值接近 1,对于整个选择过程中的所有实例都会有类似的吸引力,这将降低该方法的有效性.

综上所述,本文提出的 MDMSE 算法的主要步骤如下. (1)输入数据集 $D = \{(X_i, Y_i) \mid i = 1, \cdots, N\}$,设置子学习器个数 T,选择性学习集成个数 U. 最终选择性集成学习器集合 ST 初始为空值,N 维空间最优点位置 p. (2)基于 Bagging 机制对数据集 D 进行采样分为 $T+1$ 个子数据集. (3)分别用 T 个数据子集训练 T 个 KELM 子学习器 $\{G_t \mid t = 1, 2, \cdots, T\}$. (4)在 $T+1$ 数据集上根据式(19)计算每一个子学习器的增益度. (5)选择增益度最大的的子学习器加入集合 ST. (6)判断 ST 中学习器个数是否大于 U,满足条件则结束,否则重复执行 Step5.

3 实验验证

实验验证主要包含两大部分:(1)在 KDD99 数据集上,对所提出的 SEoKELM-NID 进行有效性验证,并分析不同参数对其性能的影响,同时对比 SEoKELM-NID 与相关入侵检测方法的性能;(2)在网络实验室搭建复杂网络物理仿真实验平台,检验 SEoKELM-NID 在复杂网络环境下对真实入侵类型检测的实时性与有效性.

3.1 KDD 数据集实验结果

3.1.1 数据集介绍及评价指标

KDD99 是由美国国防部高级规划署在 MIT 林肯实

验室模拟采集的网络连接数据集,大约包含五百万条网络连接数据,分为训练集与测试集,一共包含了 4 大类 39 小类异常入侵类型以及正常连接.

训练集中包含 22 种异常攻击类型,剩余 17 种作为未知类型存在测试集(KDD99 提供了一个 10% 测试集)中用于对未知入侵的测试判断,这也将作为验证本文方法泛化性的重要依据. 每一条连接数据连同标签一共有 42 维特征. 其中第 1、2、3 项以及最后的标签项是字符类型.

在本实验中,首先对 KDD99 数据集进行预处理将数据集的字符型特征转为数值型特征,并对特征数据进行特征归一化处理[24];然后,将处理好的训练数据集进行随机抽样,生成 $T+1$ 份子数据集,前 T 份用于学习器训练,第 $T+1$ 份用于选择性学习进行子学习器的选取.

实验环境为 Intel Core i5-3230 CPU @ 2.60GHz 8.0GB RAM 的硬件平台和 Ubuntu16.04 64 位操作系统运用 python3.6 和 pycharm 进行编程.

考虑到网络入侵的第二次人工审查,以及网络入侵造成的严重危害,网络入侵检测应尽可能检查出所有的异常连接. 本文采用准确率(AR)和漏报率(MR)这两个评价指标进行评估.

$$AR = (TP + TN)/(P + N) \tag{20}$$

$$MR = 1 - TN/(TN + FP) \tag{21}$$

其中 P 为正例个数,N 为负例个数;TP 为实际为正例且被分类器划分为正例的样本数,TN 为实际为负例且被分类器划分为负例的样本数,FP 为实际为负例但被分类器划分为正例的样本数. 准确率能客观的评估一个算法的综合性能,而漏报率能有效地评估算法的泛化性.

3.1.2　实验结果

本实验主要包含三个部分:(1)验证性实验:验证 SEoKELM-NID 的有效性,对比了其相对基于单个 KELM 以及基于传统 KELM 集成的入侵检测的性能优越性;(2)参数影响:分析不同参数设置对 SEoKELM-NID 性能的影响;(3)对比性实验:对比 SEoKELM-NID 与当前常见集成算法在 KDD99 上的分类正确率与漏报率.

(1)验证性实验　首先对比了 SEoKELM-NID 与 KELM 算法、KELM 集成算法在 KDD99 上的准确性以及检测时间,实验结果如表 1 所示. 实验中,选用径向基函数作为 KELM 的核函数,KELM 集成算法与 SEoKELM-NID 算法一样均采用 Bagging 机制,抽样次数与原维度一致,子学习器个数设为 100,MDMSE 进行子学习器选择时,最终入选的子学习器个数为 40. 表 1 中的结果为 30 次独立实验的平均值.

从表 1 结果可以看出,传统的 KELM 集成算法,虽然较单个 KELM 检测方法能提高 8 个百分点的 AR 并降低 0.6 个百分点的 MR,但同时也增加了十倍的检测时长. 而本文提出的 SEoKELM-NID,基于 MDM 准则选择部分性能良好的 KELM 子学习器进行集成,有效减少集成的子学习个数,从而消除了弱学习器对整体集成学习器的影响,在提升 AR 同时降低了 MR,同时也极大提高了检测效率(降低了检测时长).

表 1　SEoKELM-NID 与传统的 KELM 检测方法的性能比较

检测方法	AR	MR	检测时长(s)
SEoKELM-NID	0.983	0.125	0.105
KELM 集成	0.950	0.159	0.230
KELM	0.870	0.217	0.025

(2)参数设置对算法性能的影响　影响 SEoKELM-NID 性能的参数主要包括采用 Bagging 进行随机抽样时随机抽取的特征个数 F 和子学习器的集成个数 U.

输入层神经元个数即随机抽样的特征个数 F,不同的抽样率(对应不同的 F)对算法性能的影响如表 2 所示.

表 2　F 对入侵检测性能的影响

F	N	0.5N	0.01N	Log2(N)
正确率	0.9847	0.9845	0.9746	0.9744
漏报率	0.121	0.116	0.117	0.113

由表 2 可知,F 对实验结果的影响很小. 其主要原因是训练和测试所采用的 KDD99 样本集本身数量庞大,并且算法自身具有很好的泛化能力,仅通过对 F 的调节对算法的提升没有太大的帮助. 但 F 参数调节对小样本集所训练出来的学习器的算法性能有很大的提升.

子学器的个数决定了集成算法最终的泛化性能好坏,但过多的子学习器又会占用过多的资源. 本实验中,KELM 子学习器初始个数为 100,通过选择性学习来决定最终的集成个数. 最终的选择性集成子学习器个数对入侵检测性能影响如图 2 所示.

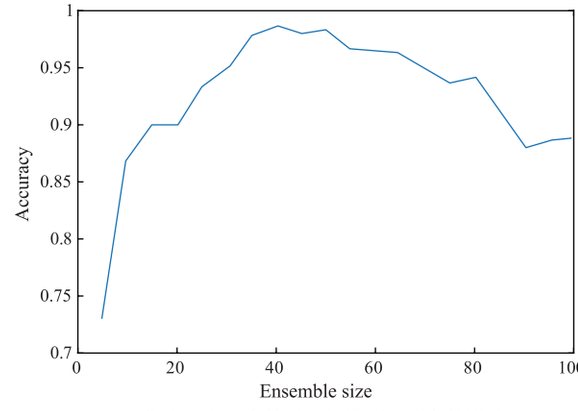

图2　集成子学习个数对入侵检测正确率的影响

从图 2 可以看出,随着子学习器的增加,入侵检测的准确率先逐渐增加然后又逐渐降低. 当子学习器在

35 ~ 40 之间时,网络入侵检测能取得较高的准确率,但随着子分类器数目的进一步增加(子分类器数目大于40)时,对入侵检测的性能不会有太大的帮助,甚至过多弱学习器的加入,反而会导致最终正确率的下降. 因此在后续的对比实验中,本文采用 40 作为选择性集成学习的最终集成个数.

(3)对比性结果 实验通过调用 scikit-learn 框架中的多个单学习器模型以及普通集成算法在 KDD99 数据集上进行训练,将其检测结果与本文提出的集成算法进行实验对比其准确率与漏报率如表 3 所示. 表 4 为不同算法的时间开销. 其中 SVM 与随机森林采用论文[25]中的参数设置. 而本文 SEoKELM-NID 采用上节最优参数设置. 为了避免因学习器单次实验稳定性不强造成的误差,表 3 和表 4 为 30 次独立重复实验的平均实验结果.

结果表明,本文提出的 SEoKELM-NID 能有效地提升学习器检测的准确率,同时大大降低了漏报率以及训练和检测时间. 从表 4 结果可以看出,普通随机森林以及梯度决策提升树在对入侵识别的正确率上有较好的表现,但训练时间较长,相比于本文算法在 0.592 的 CPU 时间可以达到 98% 的正确检测率,从时间上来看,本文算法要快上 6 倍. 而支持向量机由于其自身的泛化性强的特性,虽然正确率不高但漏报率较低.

SEoKELM-NID 虽然采用集成算法,但通过 Bagging 的分布式计算以及采用 KELM 作为子学习器,SEoKELM-NID 的训练时长上并不会比单学习器耗时更久. 同时 SEoKELM-NID 采用 MDMSE 进行了部分子学习器的集成. 在检测效率上更优于一般的全集成算法,同时检测时间更短. 当然无论是 Bagging 的分布式计算还是 KELM 和 MDMSE 的矩阵计算都意味着更大的空间消耗,这也是本文在未来将要进一步解决的问题.

KDD99 测试集中包含训练集中未包含的 17 类异常网络连接,本实验通过统计单个学习器算法和常用集成学习算法对未知类型的入侵连接的识别效果来进一步验证本文算法的泛化性,实验结果如表 5 所示.

表 3 SEoKELM-NID 与其它入侵检测方法性能对比

算法	评估指标	分类类别					
		PROBING	DOS	R2L	U2L	Normal	平均
SEoKELM-NID	准确率	0.981	0.985	0.990	0.977	0.991	0.985
	漏报率	0.115	0.132	0.110	0.145	0.108	0.125
随机森林[25]	准确率	0.935	0.930	0.947	0.932	0.958	0.940
	漏报率	0.606	0.614	0.602	0.638	0.559	0.604
朴素贝叶斯[26]	准确率	0.768	0.782	0.783	0.783	0.795	0.782
	漏报率	0.415	0.408	0.399	0.414	0.378	0.403
梯度决策提升树[27]	准确率	0.923	0.927	0.927	0.931	0.932	0.928
	漏报率	0.674	0.576	0.585	0.633	0.609	0.615
支持向量机[25]	准确率	0.912	0.918	0.922	0.916	0.933	0.920
	漏报率	0.243	0.222	0.195	0.194	0.213	0.213

表 4 SEoKELM-NID 与其他算法在运行时间上对比

	Train time(s)	Test time(s)
SEoKELM-NID	0.592	0.105
随机森林	13.589	2.261
朴素贝叶斯	5.812	0.531
梯度决策提升树	22.623	1.465
支持向量机	12.343	4.675

表 5 不同方法对未知入侵连接检测结果

算法	检测到的类型数	未知类型漏报率
SEoKELM-NID	17	0.14
随机森林	14	0.28
朴素贝叶斯	8	0.63
梯度决策提升树	15	0.24
支持向量机	13	0.31

由表 5 可知,本文所提出的基于 KELM 的选择性学习方法具有良好的泛化性能,并且在未知入侵类型的网络连接上,仍保持着较低的漏报率. 相比于泛化能力较好的支持向量机,以及普通的集成算法. 仍具有很好的优势.

3.2 复杂混合网络物理仿真实验

通过搭建网络仿真平台模拟现实生活中网络连接请求与异常入侵请求来对 SEoKELM-NID 算法的性能进行进一步验证.

3.2.1 仿真实验平台

本仿真实验采用 30 台物理终端设备、一台网络服务器、七个路由器、六个交换机,搭建网络物理仿真平台,其组网拓扑图如图 3 所示.

30 台物理终端设备分为 6 个子网,每个子网中包含三台以太网终端,两台无线网终端,一台子网路由,一台交换机. 三台以太网终端通过有线网络接入交换机再由交换机接入到子网路由,两台无线终端采用 wifi 的方式接入到子网路由中. 六个子网与攻击服务器由中央路由器连通. 在攻击服务器上使用 ettercap 软件(http://www.ettercap-project.org/ettercap/)对六个子网

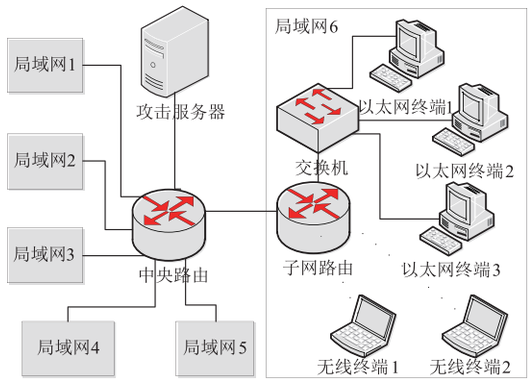

图3　网络仿真拓扑图

络发起网络仿真攻击,仿真攻击包含 PROBE、U2L、DOS、R2L 四大攻击类别. 在 30 台物理终端机上采用 TCPdump[28] 进行网络数据监听,通过抓包获取网络连接的数据,同时采集终端设备上的日志信息,将采集信息包括 41 个特征组成仿真数据集. 最后将处理好的数据集实时送往本文所训练的 SEoKELM-NID 检测器中进行判断,决定是否接收网络连接请求.

3.2.2　实验结果

通过一个星期的数据采集,每天经过过滤统计 30000 次网络请求,其中包含攻击的连接占百分之 30 即 9000 条,其中包含未知攻击 1000 条(属于四大攻击类型,但并不包含在检测器训练数据的 39 小类中如 DDoS 攻击,arp 攻击). 其检测结果如表 6 所示,其结果为 30 台 PC 机的平均值.

表 6　仿真实验结果

类型	AR	MR	平均响应时间(s)
DOS	0.983	0.13	0.10
R2L	0.983	0.15	0.07
U2R	0.990	0.09	0.09
PROBING	0.986	0.12	0.09
未知入侵	0.980	0.20	0.08

从表 6 可以看出,本文提出的 SEoKELM-NID 算法在网络入侵检测方面有很好的识别效果,尤其对于未知的网络入侵连接,其识别率依然保持在 98% 以上,平均响应时间控制在 0.1s 以内,满足网络入侵检测的有效性与实时性.

4　结论

本文提出的 SEoKELM-NID 采用 MDMSE 计算每一个 KELM 子学习器的集成增益度,通过选择增益度高的 KELM 子学习器进行部分选择性集成. SEoKELM-NID 同时具有 KELM 高效学习的特性以及 Bagging 集成算法的泛化能力,无论是在训练还是测试时都表现出良好的实时性,能及时发现异常的网络连接. 实验采用 Bagging 思想进行抽样集成再通过 Hadoop 的分布式数据处理与模型训练,可以并发地对每一个子 KELM 进行训练与检测,从而进一步提升算法效率. 实验结果表明,无论是在公共的 KDD99 数据集上还是手动搭建的复杂混合网络物理仿真平台上,SEoKELM-NID 都能有效检测出各种已知和未知的入侵类型,同时具有很低的误报率和漏报率.

参考文献

[1] 高妮,高岭,等. 基于自编码网络特征降维的轻量级入侵检测模型[J]. 电子学报,2017,45(3):730 – 739.
GAO Ni,GAO Ling,et al. A lightweight intrusion detection model based on autoencoder network with feature reduction [J]. Acta Electronica Sinica,2017,45(3):730 – 739. (in Chinese)

[2] HAMAMOTO A H,SAMPAIO L D H,ABR O T,et al. Network anomaly detection system using genetic algorithm and fuzzy logic [J]. Expert Systems with Applications,2018,92(1):309 – 402.

[3] 李立勋,张斌,董书琴,等. 基于脆弱性变换的网络动态防御有效性分析方法[J]. 电子学报,2018,46(12):3014 – 3020.
LI Li-xun,ZHANGBin,DONG Shu-qin,et al. Effectiveness analysis approach based on vulnerability mutation for network dynamic defense[J]. Acta Electronica Sincia,2018,46(12):3014 – 3020. (in Chinese)

[4] SULTANA N,CHILAMKURTI N,PENG W,et al. Survey on SDN based network intrusion detection system using machine learning approaches[J]. Peer-to-Peer Networking and Applications,2018,11(1 – 2):1 – 9.

[5] CHITRAKAR R,HUANG C. Selection of candidate support vectors in incremental SVM for network intrusion detection [J]. Computers & Security,2014,45(3):231 – 241.

[6] WANG C R,XU R F,LEE S J,et al. Network intrusion detection using equality constrained-optimization-based extreme learning machines[J]. Knowledge-Based Systems,2018,147(1):68 – 80.

[7] SONG Y,YAO S,et al. A new k-ary crisp decision tree induction with continuous valued attributes[J]. Chinese Journal of Electronics,2017,26(5):999 – 1007.

[8] SHEN Y,ZHENG K,WU C,et al. An ensemble method based on selection using bat algorithm for intrusion detection[J]. Computer Journal,2018,61(4):526 – 538.

[9] MA T,WANG F,CHENG J,et al. A hybrid spectral clustering and deep neural network ensemble algorithm for intrusion detection in sensor networks [J]. Sensors,2016,16(10):1701.

[10] LIU J,HE J,ZHANG W,et al. TCvBsISM:Texture clas-

sification via B-splines-based image statistical modeling [J]. IEEE Access,2018,6(1):44876 – 44893.
[11] LI S, SONG S, HUANG G, et al. Cross-domain extreme learning machines for domain adaptation[J]. IEEE Transactions on Systems Man & Cybernetics Systems,2018,PP(99):1 – 14.
[12] HUANG G B. What are extreme learning machines? filling the gap between Frank Rosenblatt's dream and John Von Neumann's puzzle [J]. Cognitive Computation, 2015,7(3):263 – 278.
[13] HUANG G, HUANG G B, SONG S, et al. Trends in extreme learning machines: a review[J]. Neural Networks, 2015,61(C):32 – 48.
[14] HUANG J, YU Z L, CAI Z, et al. Extreme learning machine with multi-scale local receptive fields for texture classification [J]. Multidimensional Systems and Signal Processing,2017,28(3):995 – 1011.
[15] LIANG N Y, HUANG G B, SARATCHANDRAN P, et al. A fast and accurate online sequential learning algorithm for feedforward networks [J]. IEEE Trans Neural Netw,2006,17(6):1411 – 1423.
[16] 杨乐,杨磊. 基于核函数的在线序列 ELM 模型[J]. 纺织高校基础科学学报,2013,26(4):516 – 520.
YANG Le, et al. Online sequence ELM model based on the kernel function[J]. Basic Science Journal of Textile Universities,2013,26(4):516 – 520. (in Chinese)
[17] MA G, WANG Y, WU L. Subspace ensemble learning via totally-corrective boosting for gait recognition[J]. Neurocomputing,2016,224(1):119 – 127.
[18] ERDAL H, KARAHANOÉLUB Í. Bagging ensemble models for bank profitability: an emprical research on Turkish development and investment banks[J]. Applied Soft Computing,2016,49(1):861 – 867.
[19] DRUCKER H, CORTES C, JACKEL L D, et al. Boosting and other ensemble methods [J]. Neural Computation, 1994,6(6):1289 – 1301.
[20] ZHOU Z, CHEN J, SONG Y, et al. RFSEN-ELM: Selective ensemble of extreme learning machines using rotation forest for image classification [J]. Neural Network World,2017,27(5):499 – 517.
[21] ZHOU Z H, WU J, TANG W. Ensembling neural networks: many could be better than all[J]. Artificial Intelligence,2002,137(1 – 2):239:263.
[22] MARTINEZMUOZ G, HERNANDEZLOBATO D, SUAREZ A. An Analysis of ensemble pruning techniques based on ordered aggregation[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2009, 31(2): 245 – 259.
[23] MART NEZ-MU OZ G, SU REZ A. Aggregation ordering in bagging[OL]. http://citeseerx. ist. psu. edu/viewdoc/summary? doi = 10. 1. 1. 146. 3650. 2004.
[24] DAVIS J J, CLARK A J. Data preprocessing for anomaly based network intrusion detection: A review[J]. Computers & Security,2011,30(6):353 – 375.
[25] HASAN M A M, NASSER M, PAL B, et al. Support vector machine and random forest modeling for intrusion detection system (IDS)[J]. Journal of Intelligent Learning Systems & Applications,2014,6(1):45 – 52.
[26] KOC L, MAZZUCHI T A, SARKANI S. A network intrusion detection system based on a Hidden Naïve Bayes multiclass classifier [J]. Expert Systems with Applications,2012,39(18):13492 – 13500.
[27] HU J, MIN J. Automated detection of driver fatigue based on EEG signals using gradient boosting decision tree model[J]. Cognitive Neurodynamics,2018,12(12):1 – 10.
[28] Duan Q, Al-Shaer E. Traffic-aware dynamic firewall policy management: techniques and applications [J]. IEEE Communications Magazine,2013,51(7):73 – 79.

作者简介

刘金平 男,1983 年生于湖南洞口. 博士,湖南师范大学信息科学与工程学院副教授. 研究方向为智能信息处理.
E-mail:ljp202518@163.com

何捷舟 男,1994 年生于湖南常德. 目前在湖南师范大学信息科学与工程学院攻读硕士学位. 研究方向为计算机视觉和模式识别.
E-mail:hdc@smail.hunnu.edu.cn

马天雨(通信作者) 男,1978 年生于甘肃白银,博士,湖南师范大学物理与电子学院讲师. 研究方向为复杂工业过程建模及优化控制.
E-mail:ljp@hunnu.edu.cn