基于EWC算法的DDoS攻击检测模型参数更新方法

张 斌 周奕涛*

(中国人民解放军战略支援部队信息工程大学 郑州 450001)

(河南省信息安全重点实验室 郑州 450001)

摘 要：针对现有基于多层线性感知器(Multi-Layer Perceptron, MLP)神经网络的DDoS攻击检测模型参数更新方法(MLP-UD)易遗忘模型训练原参数所用的DDoS攻击数据集(原数据集)知识、时间空间开销大的问题，该文提出一种基于弹性权重保持(Elastic Weight Consolidation, EWC)算法的模型参数更新方法(EWC-UD)。首先，使用K-Means算法计算原数据集聚类簇中心点作为费雪信息矩阵计算样本，有效提升计算样本均匀度与聚类覆盖率，大幅减少费雪信息矩阵计算量，提升参数更新效率。其次，基于费雪信息矩阵，对模型参数更新过程中的损失函数增加2次惩罚项，限制MLP神经网络中重要权重与偏置参数的变化，在保持对原数据集检测性能的基础上，提升对新DDoS攻击数据集的检测准确率。然后基于概率论对EWC-UD方法进行正确性证明，并分析时间复杂度。实验表明，针对构建的测试数据集，EWC-UD方法相较于MLP-UD仅训练新DDoS攻击数据集的更新方法，检测准确率提升37.05%，相较于MLP-UD同时训练新旧DDoS攻击数据集的更新方法，时间开销下降80.65%，内存开销降低33.18%。

关键词：分布式拒绝服务；模型参数更新；弹性权重保持算法；多层线性感知器

中图分类号：TN918.91; TP393 **文献标识码**：A **文章编号**：1009-5896(2021)10-2928-08

DOI: 10.11999/JEIT200682

DDoS Attack Detection Model Parameter Update Method Based on EWC Algorithm

ZHANG Bin ZHOU Yitao

(*PLA SSF Information Engineering University*, *Zhengzhou* 450001, *China*)

(*Henan Key Laboratory of Information Security*, *Zhengzhou* 450001, *China*)

Abstract: For the problem in the existing Multi-Layer Perceptron (MLP) based DDoS detection model parameter update method that the old model parameter training dataset knowledge is forgettable and the time and space complexity are enormous, a novel model parameter UpDate method EWC-UD based on Elastic Weight Consolidation (EWC) is proposed. Firstly, the cluster center points of the old dataset are calculated as the calculation samples of Fisher information matrix by the K-Means algorithm. The coverage rates of clusters and sampling uniformity are raised effectively, which significantly reduces the amount of Fisher Information Matrix calculation and improves the efficiency of the model parameter updates. Secondly, according to the calculated Fisher information matrix, a secondary penalty item is added to the loss function, limiting the important weight and bias parameter changes in the neural network. Maintaining the detection performance of the old DDoS attack dataset, EWC-UD improves the detection accuracy of the new DDoS attack datasets. Then based on probability theory, the correctness of EWC-UD is proved, and the time complexity is analyzed. Experiments show that for the constructed test dataset, the detection accuracy of EWC-UD is 37.05% higher than the MLP-UD that only trains the new DDoS attack dataset, and compared with the time MLP-UD

收稿日期：2020-08-04；改回日期：2021-07-21；网络出版：2021-09-06

*通信作者： 周奕涛 zyt1996715@163.com

基金项目：河南省基础与前沿技术研究计划基金(142300413201)，信息保障技术重点实验室开放基金项目(KJ-15-109)，信息工程大学科研项目(2019f3303)

Foundation Items: The Foundation and Frontier Technology Research Project of Henan Province (142300413201), The Open Fund Project of Information Assurance Technology Key Laboratory (KJ-15-109), The Research Project of Information Engineering University (2019f3303)

training both new and old DDoS attack datasets, the time and memory costs are reduced by 80.65% and 33.18 respectively.

Key words: Distributed Denail of Service (DDoS); Model parameter update; Elastic Weight Consolidation (EWC) algorithm; Multi-Layer Perceptron (MLP)

1　引言

DDoS攻击检测技术一直是网络安全领域的研究热点。MLP神经网络作为一类经典的机器学习算法，在DDoS攻击检测领域具有良好的检测效果[1,2]。但基于MLP神经网络的DDoS攻击检测模型经训练后，需适时对神经网络参数进行更新，否则神经网络参数将难以反映新型DDoS攻击特征，对新型DDoS攻击检测能力不足。

目前，基于MLP神经网络的DDoS攻击检测模型参数更新有3类方法：第1类为直接更新法，可分为仅训练新DDoS攻击数据集(简称新数据集)以及存储模型训练原参数所用的DDoS攻击数据集(简称原数据集)并与新数据集相结合一同重新训练检测模型两类更新方式，若仅对新数据集进行训练，检测模型对原数据集包含的DDoS攻击检测效果将明显下降，存在灾难性遗忘问题[3]，而存储原数据集，与新数据集一同重新训练模型会造成较大的时间与空间开销，更新效率有待提升[4]。第2类为新增网络结构法，该类方法新增网络结构学习新数据集知识，保存原有网络结构存储原数据集知识。如采用集成学习思想，新增弱分类器学习新数据集，结合投票机制输出分类结果的方法[5]，以及新增额外的神经元学习新数据知识的方法[6,7]。该类方法能有效解决灾难性遗忘问题，缺点是随着模型参数的更新，网络模型愈发复杂[8]。第3类为网络参数优化调整法，该类方法在训练新数据集时，对模型参数进行优化调整或合理分配，实现模型参数的更高效利用。如基于掩码思想，为不同的数据集分配不同的掩码组合[9]，利用使用遗忘函数优化更新过程[10]，以及通过抽取原数据集少量样本提炼原数据集知识的方法[11,12]。该类方法无需增加额外的网络结构，时间空间开销较小，但是更新性能较新增网络结构法较差[8]。

为有效解决灾难性遗忘问题并保持简单的神经网络结构，提升检测效率，针对第3类方法，本文提出一种基于EWC算法[11]的DDoS攻击检测模型参数更新方法(EWC-UD)，仅需保存原数据集少量样本以及原MLP神经网络参数，即可对检测模型进行更新，在解决灾难性遗忘问题的同时，具有较低的时间与空间复杂度。

2　EWC算法[11]

MLP神经网络在模型参数更新过程中，损失函数$L(\theta)$表示为

$$\begin{aligned} L(\theta) &= \text{CrossEntropy}(y_t(x), y_p(x)) \\ &= \sum_x y_t(x)\lg(y_p(x)) \end{aligned} \tag{1}$$

式中，θ为神经网络参数，$y_t(x)$为数据x的真实标签，$y_p(x)$为数据x输入神经网络后输出的预测标签，CrossEntropy为交叉熵函数。若采用式(1)作为参数更新过程中的损失函数，更新后模型参数将完全偏向新数据集，导致对原数据集的分类效果下降。而EWC算法通过在模型更新过程中的目标函数中增加惩罚项，为MLP神经网络中重要的神经元参数赋予较高弹性，使得其在更新过程中更难改变，以保持对原数据集较好的分类性能。EWC算法更新过程损失函数表示为

$$L_{\text{EWC}}(\theta) = L_B(\theta) + \frac{\lambda}{2}\sum_i \boldsymbol{F_i}\cdot(\theta_i - \theta^*_{A,i})^2 \tag{2}$$

式中，$L_B(\theta)$为新数据集的损失函数，λ为表达原数据集重要程度的常数，$\theta^*_{A,i}$为原模型参数，$\boldsymbol{F}$为对角费雪信息矩阵，费雪信息矩阵对角线元素包含神经网络输出关于神经网络参数的1阶导数，反映MLP神经网络参数对原数据集的重要程度。

3　EWC-UD模型参数更新方法

3.1　模型参数更新流程

EWC-UD方法模型参数更新流程如图1所示。其中，N_{Sample}为费雪信息矩阵计算样本数，$\theta_i(i\in\{1,2,\cdots,N\})$为神经网络参数，$N$为神经网络参数数目，$\boldsymbol{F}$为费雪信息矩阵，$L_B(\theta)$为新数据集的损失函数，$L_{\text{EWC-UD}}(\theta)$为EWC-UD方法训练过程中的损失函数，$\theta^*_{A,i}$为原模型参数，$y_p(x)$为神经网络预测输出。EWC-UD方法无需存储原数据集与改变神经网络结构，仅需保存原数据集的少量样本与MLP神经网络原参数，即可完成DDoS攻击检测模型的参数更新。

首先使用K-Means算法获得原数据集聚类簇，并以聚类簇中心点作为原数据集样本，有效提升样本覆盖率，增加样本信息量；然后，利用原数据集样本与原MLP神经网络参数计算MLP神经网络输出关于神经网络参数的1次导数，并构成费雪信息矩

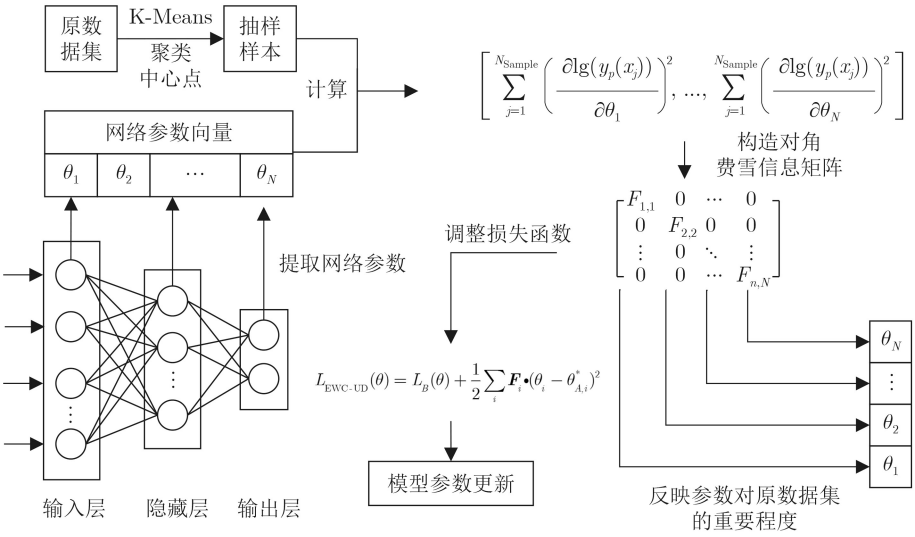


图 1 EWC-UD模型参数更新流程

阵，反映神经网络输出关于神经网络参数的变化率，体现神经网络参数对原数据集的重要程度；最后，根据不同参数对于原数据集的重要程度，为更新过程损失函数增加2次惩罚项，参数最优化过程表示为 $\theta^* = \arg\max_\theta \left(L_B(\theta) + \frac{\lambda}{2}\sum_i \boldsymbol{F_i} \cdot (\theta_i - \theta^*_{A,i})^2 \right)$，对原数据集越重要的参数，更新过程中给予的惩罚值越高，越不易改变，从而保持MLP神经网络对原数据集内DDoS攻击的检测能力。

EWC-UD模型参数更新算法如表1所示。

3.2 EWC-UD方法正确性证明

记原数据集A、新数据集B为D_A, D_B, θ_A, θ_B, $\theta_{A,B}$分别为MLP神经网络在D_A, D_B以及D_A与D_B下训练后的模型参数，设$\theta_{\text{EWC-UD}}$为经过EWC-UD更新后的模型参数。

因为$\theta_{A,B}$是在原数据集与新数据集下的最优参数，EWC-UD方法的正确性证明可转化为证明$\theta_{\text{EWC-UD}}$与$\theta_{A,B}$近似相等。基于概率论，将模型参数更新表示为条件概率的寻优过程，$\theta_{A,B}$可表示为

$$\theta_{A,B} = \arg\max_\theta (P(\theta|D_A, D_B)) \tag{3}$$

设D_A, D_B相互独立，利用条件概率公式、贝叶斯公式，对式(3)分解后，两边再同取lg函数，得到

$$\begin{aligned} \lg(P(\theta|D_A, D_B)) &= \lg\left(\frac{P(\theta, D_A, D_B)}{P(D_A, D_B)}\right) \\ &= \lg P(D_B|\theta) + \lg P(\theta|D_A) \\ &\quad - \lg P(D_B) \end{aligned} \tag{4}$$

式(4)中，右式第1项可表示为训练过程损失函数的负数，即$\lg P(D_B|\theta) = -L_B(\theta)$。同时利用拉普拉斯对角近似，近似计算式(4)右式第2项，在1阶极值点处进行泰勒展开，忽略3次方及以上项，得到

$$\begin{aligned} \lg(P(\theta|D_A)) \approx &\lg P(\theta_A|D_A) \\ &- \frac{1}{2}\frac{\partial^2 \lg P(\theta|D_A)}{\partial \theta^2}(\theta - \theta_A)^2 \end{aligned} \tag{5}$$

式(5)中，$\theta_A = \arg\max(P(\theta|D_A))$，表示原模型参数，$\frac{\partial^2 \lg P(\theta|D_A)}{\partial \theta^2}$为$\lg P(\theta|D_A)$的黑塞矩阵，表示为$H(\lg P(\theta|D_A))$，又可表示为费雪信息矩阵的累加和，表示为

$$H(\lg(\theta|D_A)) = \sum_{i=1}^{N} \boldsymbol{F}_i^A \tag{6}$$

结合式(5)、式(6)，将式(4)改写为

$$\begin{aligned} \lg(P(\theta|D_A, D_B)) \approx &- L_B(\theta) - \frac{1}{2}\sum_i \boldsymbol{F}_i^A \cdot (\theta - \theta_A)^2 \\ &+ \lg P(\theta_A|D_A) - \lg P(D_B) \end{aligned} \tag{7}$$

因为$\theta_{A,B} = \arg\max_\theta(\lg(P(\theta|D_A, D_B)))$，式(7)中，$\lg P(\theta_A|D_A)$, $\lg P(D_B)$为常数项，对于argmax函数没有影响，$\theta_{A,B}$可表示为

$$\theta_{A,B} \approx \arg\max_\theta \left(-L_B(\theta) - \frac{1}{2}\sum_i \boldsymbol{F}_i^A \cdot (\theta - \theta_A)^2 \right) \tag{8}$$

结合式(2)，设$\lambda = 1$，可得EWC-UD更新后模型参数$\theta_{\text{EWC-UD}}$为

$$\theta_{\text{EWC-UD}} = \arg\min_\theta \left(L_B(\theta) + \frac{1}{2}\sum_i F_i^A \cdot (\theta_i - \theta_A)^2 \right) \tag{9}$$

最后由式(8)、式(9)，得$\theta_{A,B} \approx \theta_{\text{EWC-UD}}$，EWC-UD方法正确性得证。

3.3 时间复杂度分析

MLP神经网络采用梯度下降法进行模型训练，

表 1　EWC-UD模型参数更新算法

输入：原数据集D_A、新数据集D_B
输出：神经网络参数θ^*

$y_t = \text{label}(D_A)$ //获得数据集标签
$x = \text{data}(D_A)$ //获得数据集数据
if Train_Time = 1 then//首次训练
　Var_list = $[W_1, b_1, W_2, b_2, W_3, b_3]$ // 3层MLP神经网络权重与偏置参数
　$N = \text{len}(D_A)$// 提取数据集长度
　$y_p(x) = \text{MLP}(x, \text{Var\_list})$// 神经网络输出预测结果
　$L(\theta) = \text{CrossEntropy}(y_t, y_p) = \sum_x y_t(x) \cdot \lg(y_p(x))$// 设置损失函数
　Var_list = Grandient Descent. minimize($L(\theta)$)//梯度下降法搜寻最优参数
　$\theta^* = \text{Var\_list}$
End if
else if Train_Time ≥ 2 then//模型参数更新
　Var_pre = Var_List//存储原模型参数
　$N_{\text{Sample}} = 30$//设置采样点数为30
　$F = \text{zeros}(\text{Var\_pre})$//费雪信息矩阵初始化
　$\text{Sample}_A = \text{K\_Means}(D_A, N_{\text{Sample}})$//利用K-Means算法获得抽样点
　　For i in range(len(Sample_A)): //计算费雪信息矩阵F
　　ders = gradients(ln($\text{Sample}_A[i]$), Var_pre)
　　For v in range(len(F)):
　　$F[v] += \text{square}(\text{ders}[v])$
　　End For
　End For
　$F = F/\text{Sample}_A$
End if
For i in range(len(Var_List)): // 修正损失函数
$L(\theta) = \text{Sum}(\text{CrossEntropy}(y_t, y_p), \text{Multiply}(F[v], (\theta[v] - \text{Var\_pre}[v])^2))$
End For
$\theta^* = \text{Grandient Descent.minimize}(L(\theta))$ //输出网络参数

梯度下降法的时间复杂度为$O\left(N_{\text{data}} \cdot C \cdot \lg\left(\frac{1}{\varepsilon}\right)\right)$。其中，$N_{\text{data}}$为数据量，$C$代表1次迭代的时间复杂度，$\varepsilon$为精度要求。为计算神经网络更新过程中的1次迭代时间复杂度C，设数据维度为$\text{Input}_{\text{dims}}$，神经网络参数维度为$\text{Para}_{\text{dims}}$，可得1次迭代的时间复杂度为

$$C = O(\text{Para}_{\text{dims}} \cdot N_{\text{data}} \cdot \text{Input}_{\text{dims}}) \tag{10}$$

由于$\text{Para}_{\text{dims}}$, $\text{Input}_{\text{dims}}$, ε均为常数，由式(10)可得神经网络训练过程时间复杂度为$O(N_{\text{data}}^2)$。

接下来计算EWC-UD的时间复杂度，设样本抽样数为N_{Sample}，得到计算费雪信息矩阵的时间复杂度为$O(N_{\text{Sample}}^2 \cdot \text{Input}_{\text{dims}} \cdot \text{Para}_{\text{dims}})$。由于$\text{Input}_{\text{dims}}$，$\text{Para}_{\text{dims}}$为常数，并且EWC-UD仅修改训练过程损失函数，未改变训练过程时间复杂度，时间复杂度为$O(N_{\text{data}}^2) + O(N_{\text{Sample}}^2)$。

设原数据集数据量为N_{old}，新数据集数据量为N_{new}，原数据集采样点数为N_{Sample}。按照传统的模型参数更新方式，需对原数据集与新数据集整合后一同训练，才可保证检测模型在原数据集与新数据集下均具有良好的检测效果，时间复杂度为$O((N_{\text{new}}+N_{\text{old}})^2)$，而EWC-UD方法时间复杂度为$O(N_{\text{new}}^2)+O(N_{\text{Sample}}^2)$。由于EWC-UD方法仅需新数据集与原数据集内的极少量样本即可实现模型的参数更新，N_{Sample}远小于N_{new}与N_{old}，证明EWC-UD方法的时间复杂度远低于传统结合新数据集与原数据集重新训练的模型参数更新方法。

4　实验验证

实验环境如下：Window10 x64操作系统，CPU Intel Core i7-8850H 2.6 GHz，32 GB RAM，1TB SSD存储空间。使用数据集包括CIC DoS Dataset 2016[13]，CIC IDS 2017[14]，CES-CIC-IDS2018-AWS[15]以及CIC DDoS 2019[16]。提取CIC IDS 2017，CES-CIC-IDS2018-AWS中的DDoS攻击部分，与CIC DoS Dataset 2016结合成为融合数据集(简称为$\text{Dataset}_{\text{old}}$)。使用CIC DDoS 2019作为新数据集(简称为$\text{Dataset}_{\text{new}}$)。

4.1　评价标准

采用精准率(Precision)、准确率(Accuracy)、召回率(Recall)、F1分数4个评价指标作为实验结果的评价指标，分别从攻击检测的覆盖率、误报率以及正确率等方面全面评价检测性能，计算公式如下：

$$\text{Recall} = \frac{\text{TP}}{\text{TP}+\text{FN}}, \quad \text{F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}},$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP}+\text{FP}},$$

$$\text{Accuracy} = \frac{\text{TP}+\text{TN}}{\text{TP}+\text{FP}+\text{TN}+\text{FN}}。$$

其中，TP表示正确归类的攻击样本，FP表示被错误归类的攻击样本，TN表示被正常分类的正常样本，FN表示被错误归类的正常样本。

4.2　实验数据预处理

由于使用多个实验数据集，并且各数据集之间在特征顺序、特征选取上存在一定差异。CIC IDS 2017, CES-CIC-IDS2018-AWS, CIC DoS Dataset 2016具有84类流量特征，而CIC DDoS 2019数据集则包括87类流量特征，并且特征顺序存在一定偏

差，特征格式不统一，为模型训练造成一定困难。因此首先需要对数据集特征进行对齐、删减等预处理操作。

(1) 特征删减：以CIC DDoS 2019数据集为基准，遍历所有特征，并在其他数据集中进行搜索，若特征在所有数据集中则进行保留，否则进行删除，之后再删除部分字符特征，最终剩下79类流量特征。

(2) 特征对齐：通过特征删减后，所有数据集具有相同特征，以CIC DDoS 2019数据集为基准，重新排列其余3个数据集特征次序，确保所有数据集特征排列一致，使得数据特征以相同顺序输入学习模型。

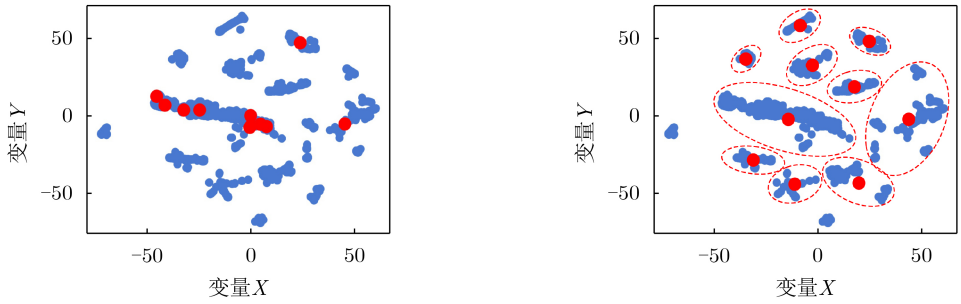
(3) 归一化：不同的流量特征通常具有不同的量纲与量纲单位，为了消除量纲对于训练过程的影响，需要对数据集进行标准化处理。归一化过程如下式所示：$x_i = \dfrac{x_i - x_{\min}}{x_{\max} - x_{\min}}$，其中$x_i$为原始值，$x_{\min}$, $x_{\max}$分别为数据集中该类流量特征的最小值与最大值，通过该式，将流量特征转换成[0,1]区间的值，消除量纲影响。

(4) 划分训练集与测试集：将数据集按照80%, 20%划分成训练集与测试集，训练集用以模型训练，测试集用以验证模型检测效果。

4.3 K-Means样本选取法性能验证

本节验证K-Means样本选取法性能，并与随机抽样法进行对比。利用t分布随机邻近嵌入算法(t-distributed Stochastic Neighbor Embedding, t-SNE)进行数据降维，画出原数据集中部分数据分布散点图。如图2(a)所示，数据集分布呈聚类状，由于数据分布不均匀，随机采样可能导致聚类覆盖率不足，导致抽样样本难以涵盖原数据集信息，使得近似计算的费雪信息矩阵与真实值偏差较大，造成模型更新失败。

因此，为获得均匀样本点，有效覆盖原数据集聚类，使用K-Means算法获得原数据集聚类簇中心点作为采样样本，如图2(b)所示，每个样本位于聚类中心，能有效反映聚类内数据信息，解决随机抽样带来的样本分布不均匀，聚类覆盖率不高的问题，减少所需计算样本，提升模型更新效率。



(a) 随机抽样法(采样点为10个)　(b) K-Means算法选取样本点(聚类中心点数为10)

图 2 随机抽样法与K-Means样本选取法图示

为对比K-Means样本选取法与随机抽样法性能，抽样点选取范围为[3,200]，每个抽样点分别使用K-Means样本选取法与随机抽样法进行10次模型参数更新，检验更新后检测模型对原数据集的检测性能。图3给出检测准确率均值关于抽样点数的变化曲线，并进行标准差填充。

如图3所示，阴影部分为10次实验下检测准确率标准差情况，反映数据稳定性，实线表示检测准确率均值情况，阴影上界曲线为Standard_UP，阴影下界为Standard_Down，计算式如下：

$$\text{Standard\_UP} = \frac{\text{Max} - \text{Mean}}{\text{Max} - \text{Min}} \cdot \text{Standard}$$

$$\text{Standard\_Down} = \frac{\text{Mean} - \text{Min}}{\text{Max} - \text{Min}} \cdot \text{Standard}$$

其中，Max, Min, Mean, Standard分别为10次实验中检测准确率最大值、最小值、均值以及标准差。

可以看出，当抽样点数大于20时，K-Means样本选取法的检测性能趋于稳定，且检测率均值较随机抽样法更高，随机抽样法只有在抽样点数大于200时，检测性能才趋于稳定。实验表明，相较于随

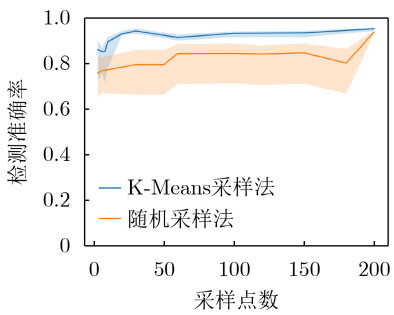


图 3 不同抽样点数的检测准确率均值标准差填充图

机抽样法，K-Means样本选取法均匀选取样本，有效提高聚类覆盖率与样本信息量，提升检测准确率。

表2给出抽样样本分别在3, 6, 20, 60, 100, 200个时，K-Means样本选取法与随机抽样法在10次试验下的检测平均准确率、最高准确率以及最低准确率。

表 2　K-Means样本选取法与随机抽样法性能验证(%)

抽样样本数	平均准确率		最高准确率		最低准确率	
	K-Means	Random	K-Means	Random	K-Means	Random
3	86.35	78.65	96.78	95.17	50.00	49.40
6	88.54	71.51	91.08	86.23	83.90	49.40
20	92.95	79.12	96.81	93.67	91.07	49.41
60	91.33	84.32	95.94	94.15	86.73	49.42
100	93.21	84.09	95.84	93.91	87.96	49.41
200	95.29	93.81	96.78	95.31	91.61	91.18

如表2所示，K-Means样本选取法仅需30个样本即可达到94.29%的平均检测准确率，仅占原数据集的0.0023‰，而随机抽样达到类似效果至少需要200个样本。实验表明，K-Means样本选取法可有效减少费雪信息矩阵的计算样本数，进一步提升更新效率。

4.4　更新性能验证

4.4.1　检测率验证

为验证EWC-UD方法的参数更新性能，针对MLP神经网络设置两组对照实验采用以下两种更新方式：

(1) MLP_New：对训练完成的模型进行保存，仅使用新数据集数据训练模型，更新参数。

(2) MLP_Whole：结合原数据集与新数据集对模型进行重新训练。

相较于EWC-UD方法，MLP_New方法除更新过程损失函数不同之外并无其他区别，而MLP_Whole除损失函数不同之外，还使用了原数据集。通过将所提EWC-UD方法与MLP_New, MLP_Whole方法对比，可较好地检验EWC-UD方法的针对于MLP神经网络模型参数更新过程的改进效果。除此之外，选用随机森林、决策树这两种具有一定增量更新能力的树状结构神经网络作为横向对照。

各类算法参数设置为：MLP采用4层网络结构，分为输入层、2个隐藏层以及输出层，输入层包括79个神经元节点，而隐藏层则分别包括16个、8个神经元节点，输出层包括2个输出节点，训练批次为500次，每批次采样数为2048个；随机森林算法训练批次为500次，每批次采样数为2048个，树的棵数为5棵，最大节点数为10个；决策树算法训练批次为500次，每批次采样数为2048个，最大节点数为50个。

同时，为了更加全面地反映现实网络环境，设计两种不同的模型参数更新场景：

(1) 场景1：使用$\text{Dataset}_{\text{old}}$作为原训练数据集，使用$\text{Dataset}_{\text{new}}$作为更新数据集。测试数据集为50%的原数据集与50%的新数据集。验证模型参数更新后，检测模型在测试数据集上的检测性能。该场景模拟正常的网络环境，在一定周期内进行正常更新的情况。

(2) 场景2：利用$\text{Dataset}_{\text{old}}$作为原训练数据集，在$\text{Dataset}_{\text{new}}$中选择单一的Portmap攻击数据作为新数据集。测试数据集为80%的原训练数据以及20%的更新数据集。该场景模拟出现新类型攻击，对于模型参数进行紧急更新的情况。

表3给出在场景1与场景2下，各类模型参数更新方法的模型参数更新性能：

如表3所示，场景1下，EWC-UD相较于MLP_New, RF-UD, DT-UD方法，在测试集上的准确率分别提升9.35%, 9.56%, 8.93%，F1分数分别提高0.15, 0.18, 0.17，在场景2下，在测试集上的准确率分别提升37.05%, 23%, 8.72%，F1分数分别提高0.33, 0.25, 0.11。实验表明，EWC-UD方法性能与MLP_Whole方法相近，但是MLP_Whole方法需保存原数据集，时间与空间开销高于EWC-UD方法。

为更加全面地评价5类检测模型参数更新方法，采用ROC曲线分析5类模型参数更新方法的检测性能。

如图4所示EWC-UD方法与MLP_Whole方法的ROC曲线几乎重合，且曲线下方面积较其他3类方法更大，进一步证明EWC-UD方法具有良好的模型参数更新性能，优于其他3类方法。

4.4.2　时间空间性能验证

表4中给出在场景1与场景2下，各类模型参数更新方法的时间与空间开销。

表 3 各类模型参数更新方法性能验证

方法	场景1				场景2			
	准确率(%)	精准率(%)	召回率(%)	F1分数	准确率(%)	精准率(%)	召回率(%)	F1分数
MLP_Whole	98.41	97.93	98.98	0.98	97.12	94.10	97.12	0.96
MLP_New	88.71	97.09	71.72	0.83	59.52	99.17	44.94	0.62
RF-UD	88.50	71.73	90.31	0.80	73.57	94.47	55.98	0.70
DT-UD	89.13	74.63	89.61	0.81	87.85	95.09	74.93	0.84
EWC-UD	98.06	98.02	98.23	0.98	96.57	94.96	94.67	0.95

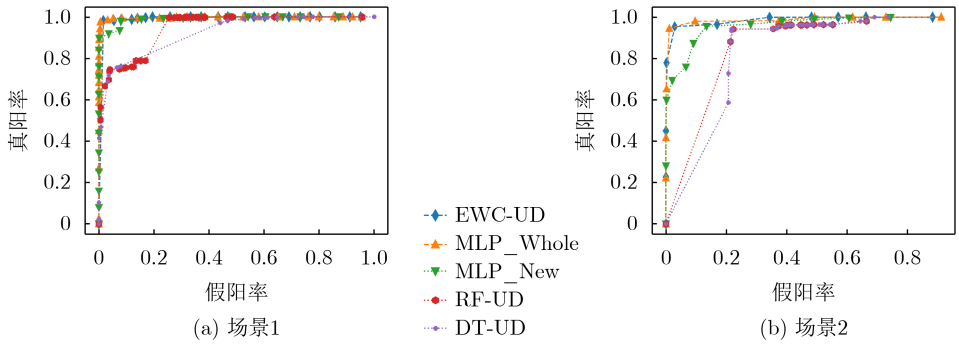


图 4 场景1与场景2下各类检测算法模型参数更新效果ROC曲线图

表 4 各类参数更新方法更新成本

方法	场景1		场景2	
	时间开销(s)	内存开销(MB)	时间开销(s)	内存开销(MB)
MLP_Whole	753.53	12783.82	322.03	11259.08
MLP_New	182.28	7467.50	89.52	3195.62
RF-UD	127.40	6278.46	34.46	3850.60
DT-UD	125.53	7099.88	48.02	3743.80
EWC-UD	145.80	8541.24	128.18	4318.04

由表4可知，MLP_Whole方法在场景1下较EWC-UD方法更新时间增加416.8%，内存开销增大49.67%，在场景2下，时间开销增加151.23%，内存开销增大160.75%。

实验表明MLP-UD方法存在灾难性遗忘现象，特别是在场景2下，对原数据集的检测准确率下降更为显著。尽管MLP_Whole方法可以在原数据集和新数据集下同时保持较好的检测性能，减少遗忘现象，但是时间与空间开销显著高于EWC-UD方法。而现实网络环境中，网络流量数据量巨大，若使用MLP_Whole方法，不仅延缓模型更新部署进度，还存在额外的存储与计算开销。EWC-UD方法在进行模型参数更新后，检测性能与MLP_Whole方法相近的同时，大幅减少时间与空间开销，更新效率得到显著提升。

5 结束语

本文提出一种基于EWC算法的DDoS攻击检测模型参数更新方法EWC-UD。该类方法仅需新数据集与原数据集的少量抽样即可完成模型更新，并且使用K-Means算法改进费雪信息矩阵计算样本选取过程，进一步提升模型参数更新效率。实验表明，EWC-UD方法有效改善灾难性遗忘问题，并且具有较低的时间与空间效率。在下一步工作中，考虑将所提方法应用于更加复杂的神经网络，提升方法普适性。

参 考 文 献

[1] WANG Meng, LU Yiqin, and QIN Jiancheng. A dynamic MLP-based DDoS attack detection method using feature selection and feedback[J]. *Computers & Security*, 2020, 88: 101645. doi: 10.1016/j.cose.2019.101645.

[2] 董书琴, 张斌. 基于深度特征学习的网络流量异常检测方法[J]. 电子与信息学报, 2020, 42(3): 695–703. doi: 10.11999/JEIT190266.
DONG Shuqin and ZHANG Bin. Network traffic anomaly detection method based on deep features learning[J].

Journal of Electronics & Information Technology, 2020, 42(3): 695–703. doi: 10.11999/JEIT190266.

[3] KEMKER R, MCCLURE M, ABITINO A, *et al*. Measuring catastrophic forgetting in neural networks[J]. arXiv preprint, arXiv: 1708.02072, 2017.

[4] KUMARAN D, HASSABIS D, and MCCLELLAND J L. What learning systems do intelligent agents need? Complementary learning systems theory updated[J]. *Trends in Cognitive Sciences*, 2016, 20(7): 512–534. doi: 10.1016/j.tics.2016.05.004.

[5] POLIKAR R, UPDA L, UPDA S S, *et al*. Learn++: An incremental learning algorithm for supervised neural networks[J]. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 2001, 31(4): 497–508. doi: 10.1109/5326.983933.

[6] ZRIBI M and BOUJELBENE Y. The neural networks with an incremental learning algorithm approach for mass classification in breast cancer[J]. *Biomedical Data Mining*, 2016, 5(118): 2. doi: 10.4172/2090-4924.1000118.

[7] SHIOTANI S, FUKUDA T, and SHIBATA T. A neural network architecture for incremental learning[J]. *Neurocomputing*, 1995, 9(2): 111–130. doi: 10.1016/0925-2312(94)00061-V.

[8] GEPPERTH A and HAMMER B. Incremental learning algorithms and applications[C]. The 24th European Symposium on Artificial Neural Networks, Bruges, Belgium, 2016: 357–368.

[9] MALLYA A, DAVIS D, and LAZEBNIK S. Piggyback: Adapting a single network to multiple tasks by learning to mask weights[C]. The 15th European Conference on Computer Vision, Munich, Germany, 2018: 72–88. doi: 10.1007/978-3-030-01225-0_5.

[10] PÉREZ-SÁNCHEZ B, FONTENLA-ROMERO O, GUIJARRO-BERDIÑAS B, *et al*. An online learning algorithm for adaptable topologies of neural networks[J]. *Expert Systems with Applications*, 2013, 40(18): 7294–7304. doi: 10.1016/j.eswa.2013.06.066.

[11] KIRKPATRICK J, PASCANU R, RABINOWITZ N, *et al*. Overcoming catastrophic forgetting in neural networks[J]. *Proceedings of the National Academy of Sciences of the United States of America*, 2017, 114(13): 3521–3526. doi: 10.1073/pnas.1611835114.

[12] CASTRO F M, MARÍN-JIMÉNEZ M J, GUIL N, *et al*. End-to-end incremental learning[C]. The 15th European Conference on Computer Vision, Munich, Germany, 2018: 241–257. doi: 10.1007/978-3-030-01258-8_15.

[13] Canadian Institute for Cybersecurity. CIC-DoS-2016[EB/OL]. https://www.unb.ca/cic/datasets/dos-dataset.html, 2020.

[14] Canadian Institute for Cybersecurity. CES-DDoS-2017[EB/OL]. https://www.unb.ca/cic/datasets/ids-2017.html, 2020.

[15] Canadian Institute for Cybersecurity. CES-CIC-IDS2018-AWS[EB/OL]. https://www.unb.ca/cic/datasets/ids-2018.html, 2020.

[16] SHARAFALDIN I, LASHKARI A H, HAKAK S, *et al*. Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy[C]. 2019 International Carnahan Conference on Security Technology, Chennai, India, 2019: 1–8. doi: 10.1109/CCST.2019.8888419.

张　斌：男，1969年生，教授，博士生导师，研究方向为信息系统安全.
周奕涛：男，1996年生，硕士生，研究方向为基于机器学习的DDoS攻击检测.

责任编辑：陈　倩