计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2023,32(6):42−50 [doi: 10.15888/j.cnki.csa.009114]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于多特征融合自动编码器的增量式入侵检测①

张碧洪[1], 夏海霞[2], 张　宇[1], 高志刚[3]

[1](浙江理工大学 计算机科学与技术学院, 杭州 310018)
[2](浙江理工大学 信息科学与工程学院, 杭州 310018)
[3](杭州电子科技大学 计算机学院, 杭州 310018)
通信作者: 夏海霞, E-mail: xiahx@zstu.edu.cn

摘　要: 针对增量式入侵检测算法由于对旧知识产生灾难性遗忘而导致对旧类别数据分类准确率不高的问题, 本文提出了一种基于非对称式多特征融合自动编码器 (asymmetric multi-feature fusion auto-encoder, AMAE) 和全连接分类神经网络 (classification deep neural network, C-DNN) 的增量式入侵检测算法 (ImFace). 在增量学习阶段, ImFace 会为每一批新的数据集训练一个 AMAE 模型和 C-DNN 模型. 同时, 本文通过使用变分自动编码器 (variational auto-encoder, VAE) 对数据进行过采样的方式来改善由于数据集不平衡而导致 C-DNN 对某些类别数据的检测能力不足的问题. 在检测阶段, ImFace 将输入数据经过所有 AMAE 和 C-DNN, 然后将 AMAE 的结果作为置信度来选择某一个 C-DNN 的输出结果作为最终结果. 本文使用 CICIDS2017 数据集来检验 ImFace 算法的有效性. 实验结果表明, ImFace 算法不仅能够保留对旧类别的分类能力, 同时对新类别的数据也有很高的检测准确率.

关键词: 入侵检测; 非对称式多特征融合自动编码器; 灾难性遗忘; 增量学习; 变分自动编码器; 深度学习; 目标检测

引用格式: 张碧洪,夏海霞,张宇,高志刚.基于多特征融合自动编码器的增量式入侵检测.计算机系统应用,2023,32(6):42–50. http://www.c-s-a.org.cn/1003-3254/9114.html

Incremental Intrusion Detection Based on Multi-feature Fusion Auto-encoder

ZHANG Bi-Hong[1], XIA Hai-Xia[2], ZHANG Yu[1], GAO Zhi-Gang[3]

[1](School of Computer Science and Technology, Zhejiang Sci-Tech University, Hangzhou 310018, China)
[2](School of Informatics Science and Engineering, Zhejiang Sci-Tech University, Hangzhou 310018, China)
[3](School of Computer Science, Hangzhou Dianzi University, Hangzhou 310018, China)

Abstract: To address the problem that incremental intrusion detection algorithms do not classify old category data with high accuracy due to catastrophic forgetting of old knowledge, this study proposes an incremental intrusion detection algorithm (ImFace) based on asymmetric multi-feature fusion auto-encoder (AMAE) and fully connected classification deep neural network (C-DNN). In the incremental learning phase, ImFace trains an AMAE model and a C-DNN model for each new batch of the dataset. At the same time, this study solves the problem of C-DNN's insufficient ability to detect certain categories of data due to unbalanced datasets by oversampling the data through a variational auto-encoder (VAE). In the detection phase, ImFace makes the input data pass through all AMAEs and C-DNNs and then uses the result of AMAEs as the confidence level to select the output result of a C-DNN as the final result. In this study, the CICIDS2017 dataset is used to test the effectiveness of the ImFace algorithm. The experimental results show that the ImFace algorithm not only retains the ability to classify old categories but also has a high detection accuracy for new categories of data.

Key words: intrusion detection; asymmetric multi-feature fusion auto-encoder (AMAE); catastrophic forgetting; incremental learning; variational auto-encoder (VAE); deep learning; target detection

① 基金项目: 国家重点研发计划 (2021YFC3320301); 国家自然科学基金 (61877015); 浙江省自然科学基金 (LY21F020028)
收稿时间: 2022-11-17; 修改时间: 2022-12-23; 采用时间: 2023-01-06; csa 在线出版时间: 2023-03-24
CNKI 网络首发时间: 2023-03-27

1 引言

随着计算机技术的发展, 网络在人们的生活和工作中发挥着越来越重要的作用. 然而, 在如今愈加复杂的网络环境中, 针对主机的恶意活动和网络威胁的数量也在不断地增加. 因此主机一旦遭受网络入侵将造成信息泄露, 设备瘫痪和财产损失等重大问题. 因此, 入侵检测系统 (intrusion detection systems, IDSs) 成为网络安全领域中所关注的重点. IDSs 是计算机网络安全中的关键组件, 其旨在检测网络流量中针对计算机的恶意活动, 例如分布式拒绝服务, 注入攻击等, 并对网络入侵行为采取相应的措施. 当 IDSs 检测到针对计算机的网络攻击时, 它将向安全管理员发出警告. 所以, 拥有可靠的 IDSs 是保护计算机网络免受网络攻击, 减少经济损失的重要保障.

IDSs 在 1980 年被首次提出, 经过几十年的发展取得了一定的成果. IDSs 根据检测数据的来源可以分为基于主机的 IDSs 和基于网络的 IDSs. 近些年, 研究人员逐渐将机器学习算法应用于基于网络的 IDSs 中, 以进一步改进 IDSs 的高准确率和低误报率. 早些年, 应用于 IDSs 的机器学习算法主要以传统的机器学习算法为主, 例如决策树, 支持向量机, K-最近邻分类算法, 逻辑回归算法等等. 传统的机器学习算法具有计算速度快, 内存占用少等优点. 如今, 随着机器学习的不断发展, 研究人员将深度学习算法与 IDSs 结合起来, 例如 Xiao 等人[1] 提出了一种基于卷积神经网络的 IDSs (CNN-IDS), 研究人员将输入数据通过主成分分析的方式进行降维后转换成灰度图像, 然后将图像输入卷积神经网络中进行分类学习. Yang 等人[2] 使用长短期记忆网络 (long short-term memory, LSTM) 来提取网络流量中的时序特征, 并且使用注意力机制保存数据之间的依赖关系. 这些机器学习算法可以通过模型训练等方式提取数据集中的知识. 然而, 在实际应用中如何获取包含所有入侵检测类型的数据集成为一个重要的问题. 因为, 网络攻击的类型并不能一次收集完成, 而是随着时间的推移不断出现新的攻击方式. 虽然, 当新数据到达时我们可以通过使用全部数据集创建一个新的模型. 但是, 随着数据集的不断增大, 模型训练所需要的时间和资源也在不断增加. 此外, 我们也很难修改之前的入侵检测模型来适应新的入侵变体. 这是因为机器学习中的大部分模型的训练方式都是静态的, 即模型的训练过程仅执行一次. 如果在旧数据集上训练完成的模型在新任务上再次进行训练就会对模型提取到的旧知识产生灾难性遗忘. 为了克服灾难性的问题, 研究人员提出了增量式学习的方法.

增量式学习是指一个模型能够不断地从新样本中学习新知识的同时, 并能保存大部分以前已经学习到的旧知识. 增量学习是多任务的, 并且允许模型在学习下一个任务之前多次处理当前任务的数据. 根据模型提取新知识的方式, 可以将增量式学习分为基于模型结构的方法, 基于回放的方法和基于正则化的方法. 目前, 已经有很多研究人员将增量学习应用与图片分类的任务中. 例如 LwF 模型[3] 通过知识蒸馏和冻结网络层来给新任务的损失函数施加约束的正则化方法来减少模型对旧知识的遗忘. iCaRL 模型[4] 在此的基础上通过保留具有代表性的旧数据, 并将旧数据和新数据进行一起训练的回放方法来应对模型的灾难性遗忘的问题. 通过将增量式学习应用于 IDSs 的方法, 能够使 IDSs 在不断发展的网络攻击类型具有检测能力.

因此, 本文针对增量式入侵检测算法中对旧类别数据的分类准确率较低的问题提出了一种基于非对称式多特征融合自动编码器 (asymmetric multi-feature fusion auto-encoder, AMAE) 和多个全连接分类神经网络 (classification deep neural network, C-DNN) 的增量式入侵检测算法 (ImFace). 本文的主要贡献如下.

1) 本文提出了一种非对称式多特征融合自动编码器 (AMAE), AMAE 通过循环神经网络, 卷积神经网络和全连接神经网络来提取旧类别数据中的时序特征, 空间特征和全局特征. 多特征的融合提取能够增加 AMAE 对新旧类别数据的分类能力.

2) 本文提出了一种基于 AMAE 和 C-DNN 进行联合决策的增量式入侵检测算法 (ImFace). ImFace 在增量学习过程中仅需要新数据集而无需保存旧数据集. ImFace 决策算法在保留算法对旧类别分类能力的同时能够很好地学习新知识.

3) 本文为了解决由于不同类别数据之间数量极度不平衡而导致 C-DNN 对稀疏样本分类能力不佳的问题, 在 C-DNN 训练阶段使用 VAE 对稀疏样本进行过采样. 同时本文在 ImFace 算法检测阶段对稀疏样本设置了高优先级来提高 ImFace 算法对稀疏样本的检测能力.

2 相关工作

入侵检测算法根据检测方式的不同可以分为基于

签名和基于异常两种方式. 目前, 机器学习算法被广泛地应用于基于异常的入侵检测中, 其中包括传统机器学习和深度学习两种方式. 在传统的机器学习算法中, 研究人员使用逻辑回归, 决策树, 支持向量机, K-最近邻算法等经典的算法[5–9] 来识别各种网络攻击. 随着深度学习的发展, 研究人员开始将神经网络应用于入侵检测中. 例如: Wu 等人[10] 首先使用循环神经网络在正常的时间序列上建立一个模型, 然后使用朴素贝叶斯算法的预测误差来检测异常值. Duan 等人[11] 则使用主成分分析算法对神经网络的输入数据进行降维. 然后将降维后的数据转换为灰度图像送入卷积神经网络内进行分类. 这种方法能够减少数据的冗余度的同时能够提取数据中的空间特征. 通过实验结果表明, 该方法可以达到 94% 的准确率. Hassan 等人[12] 提出了一种基于 CNN 和减重长短期记忆 (WDLSTM) 的混合神经网络模型. 该方法首先使用 CNN 提取数据中的特征, 然后使用循环神经网络对提取到的特征进行处理.

然而, 大部分的入侵检测算法都是静态的即模型都是在包含所有入侵检测类型的数据集上进行训练. 静态模型一旦训练完成后无法随着不断发展的网络入侵变体的增加而学习新的知识. 因此, 增量学习开始与入侵检测算法相结合. Yi 等人[13] 开发了一种增量式的 SVM 分类器用于网络入侵检测, 研究人员将特征属性的均值和均方差加入 RBF 核函数中用以减少学习过程中出现的震荡问题. Constantinides 等人[14] 和 Xu 等人[15] 则分别将 SVM 和 KNN 或者自组织增量神经网络 (SOINN) 结合起来用以提高增量学习中的准确率. 由于 SVM 算法本身支持增量学习的特性, 因此很多研究人员将其应用于入侵检测算法中. 然而 SVM 的训练复杂度与数据集的规模相关, 所以该算法在大规模的数据集中表现并不理想.

随着深度学习的发展, 许多研究人员开始使用神经网络来处理大规模的数据. 但是, 深度学习在增量式入侵检测算法中的研究还比较少. 相反的是, 深度学习与增量式学习的结合在图像分类中成为研究热点. 例如: Roy 等人[16] 提出了一种自适应深度卷积神经网络进行图像的增量式学习, 该算法通过树状的方式增长来保存对旧知识的记忆和对新知识的学习. Tao 等人[17] 则更加关注小样本的深度增量学习, 研究人员将神经气体网络和增量学习结合起来. 该算法旨在从小样本数据中学习新的类别, 并由此提出了 TOPIC 框架. Zhu 等人[18] 使用旧类别的数据生成原型, 并在原型中加入噪声后作为伪标签样本用于对模型的旧知识回放. 同时, 研究人员将自监督损失, 知识蒸馏损失和伪标签样本的损失之和作为新的损失函数. 如今, 将深度学习应用于增量式入侵检测算法开始受到研究人员的关注. 例如, Data 等人[19] 提出了将多个深度前馈神经网络进行树状结构连接的增量式入侵算法 (T-DFNN). T-DFNN 通过生成在神经网络输出层生成神经元节点的方式来学习新任务中的知识. 通过在 CICIDS2017 数据集上的实验证明, T-DFNN 在每个增量学习过程中的 *F*1-*score* 宏观平均值均在 0.85 以上. 李珊珊等人[20] 提出了基于可能性理论和自组织神经网络的无监督增量式入侵检测算法 (P-SOINN). P-SOINN 在自组织神经网络增加新节点过程中将可能性隶属度作为新类别的判别标准.

3 ImFace 算法介绍

3.1 多特征融合自动编码器 (AMAE)

如图 1 所示, 本文提出了一种多特征融合自动编码器的神经网络结构 (AMAE). AMAE 是一种非对称式的自动编码器结构, 其中共包含 3 个编码器和 1 个解码器.

编码器部分主要包含 LSTM_encoder, Linear_encoder 和 Conv_encoder 三个模块. LSTM_encoder 主要由 LSTM 层组成. LSTM 是专门用来处理序列数据的循环神经网络 (recurrent neural network, RNN) 的一种变体, 其中所包含的细胞状态能够将序列数据中较早时间步长所包含的信息传递到较晚时间步长中, 这克服了 RNN 短时记忆的问题. 同时, LSTM 通过门结构例如遗忘门, 输入门和输出门来控制序列数据中信息的移除和添加. 本文在 LSTM_encoder 模块共包含了 3 层 LSTM, 其中的输出维度分别为 [128, 64, 18]. 同时, 本文在 LSTM 层的前面连接了 batch normalization (BN) 层用于加快模型训练速度, 防止模型训练时梯度爆炸, 并在 LSTM 层后面连接 tanh 激活层用于激活函数. ImFace 算法主要通过 LSTM_encoder 模块来提取网络流量数据中的时序特征. Linear_encoder 主要由 3 个全连接层组成, 全连接层的每个神经元与下一层的神经元全部连接, 从而实现数据的线性变换. Linear_encoder 所包含全连接层的神经元个数分别为 128, 64, 18. Linear_encoder 同样增加了 BN 层, 但是全连接层后面连接 Leaky_ReLU 作为激活层. 因为全连接层之

间的每个神经元都进行了互相连接, 所以本文使用 Linear_encoder 模块来提取数据中的全局特征. Conv_encoder 模块主要使用一维卷积神经元 (Conv1D) 的卷积层来处理数据. Conv1D 仅在一个维度上进行卷积操作, 因此常被用来处理一维的序列数据. Conv1D 主要包含输入通道数, 卷积产生的通道数以及卷积核的尺寸, 步长等. 本文在 Conv_encoder 共设置两个卷积层, 其中的每个卷积层使用的卷积核尺寸为 3, 步长为 2, 输入通道数和卷积产生的通道数都为 1. 卷积层的前后也同样连接 BN 层和 Leaky_ReLU 激活层. 由于卷积核以相同的步长对数据进行卷积操作, 因此本文主要使用 Conv_encoder 提取网络流量数据中的局部空间特征.

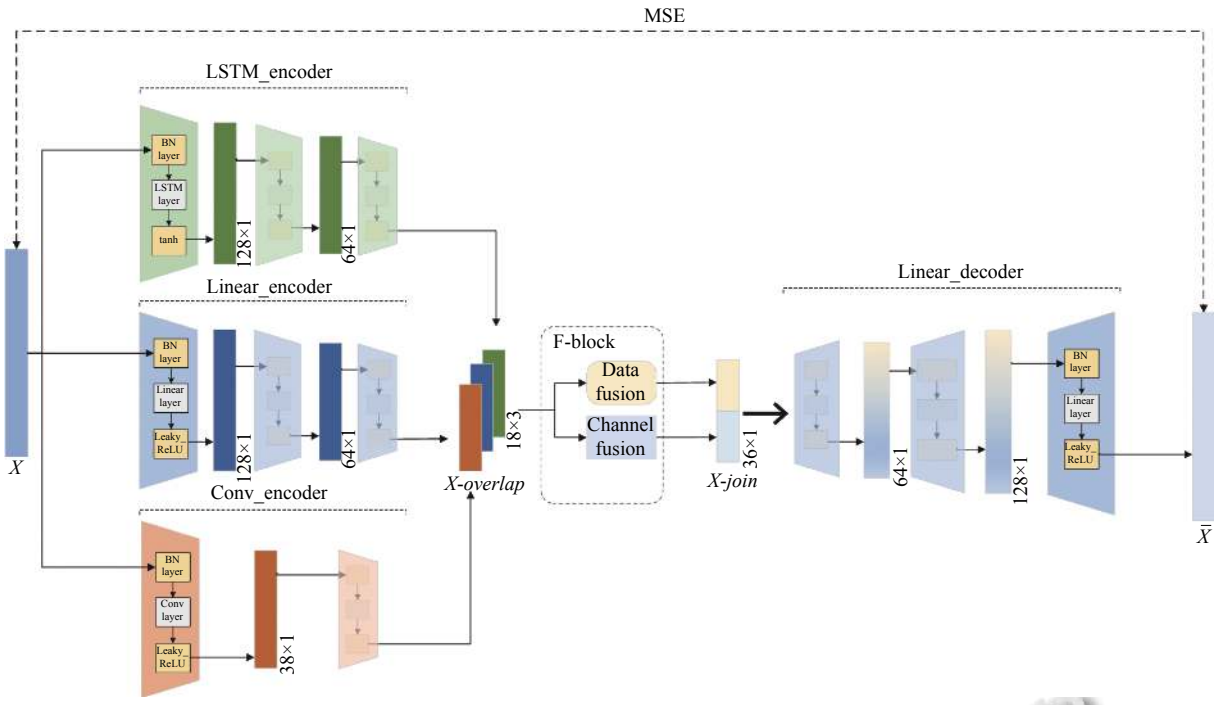

图 1 AMAE 神经网络结构图

输入数据 x 经过编码器部分处理后生成维度为 [18, 3] 的 *X-overlap*, 然后将 *X-overlap* 送入 F-block 部分进行融合处理. F-block 的融合计算主要包括两部分, 分别为 Data fusion 和 Channel fusion. 其中 Data fusion 是对 *X-overlap* 在第 1 维度进行求和计算, 计算后的数据维度为 [18, 1]. Channel fusion 则使用一个 Conv1D 卷积层对 *X-overlap* 进行多通道卷积计算. 该卷积层所使用的 Conv1D 的输入通道数为 3, 卷积后的通道数为 1, 卷积核的尺寸和步长都为 1. 然后将 Data fusion 和 Channel fusion 输出的数据进行拼接后形成维度为 [36, 1] 的数据 *X-join*. 然后将 *X-join* 送入 Linear_dencoder 模块中进行解码操作.

Linear_decoder 模块同样主要使用 3 个全连接层对数据进行处理, 并在全连接层的前面和后面分别连接了 BN 层和 Leaky_ReLU 层. Linear_decoder 中全连接层的输出维度分别为 [36, 64, 128].

本文认为对多特征数据等进行融合后, 对称的自动编码器结构无法还原融合特征中的空间特征, 时序特征等. 同时由于全连接层具有对数据进行综合计算的能力. 因此, 本文采用了非对称式的自动编码器结构即在解码阶段仅采用 Linear_decoder 一个模块进行解码.

3.2 算法决策流程

ImFace 算法在训练阶段中会为每一部分知识训练一个 C-DNN 和 AMAE. 图 2 中 C-DNN (old) 和 AMAE (old) 表示模型是使用旧数据集训练. C-DNN(new), C-DNN (new) 和 AMAE (new) 表示模型是仅用新数据集训练的.

如图 2 所示, 在检测阶段 ImFace 算法会首先查询当前模型中所包含 C-DNN 的数量. 当 ImFace 中仅包含一个 C-DNN 时, 数据类别仅以 C-DNN 的分类结果为准确结果. 当 ImFace 中包含多个 C-DNN 时即存在

C-DNN (old) 和 C-DNN (new), 输入数据 X 将被输入所有的 C-DNN 中, 并将所有 C-DNN 输出的分类结果保存到数组 *C-List*. 同时将输入数据 X 输入所有的 AMAE 内, 并计算输入数据 X 和输出数据 $\bar{X}$ 之间的均方误差 (MSE), 并将所有的 MSE 保存到数组 *R-List*. 最后将 *R-List* 中的 MSE 作为置信度来选择 *C-List* 内的某一个分类结果作为最终结果即选择 *C-List* 内对应 MSE 值最小的分类结果作为最终结果.

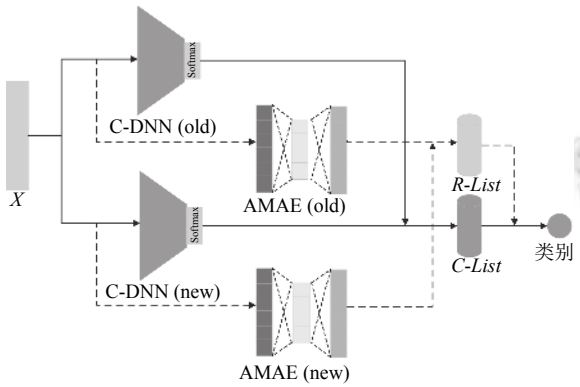

图 2 ImFace 检测阶段流程图

除此以外, 如果 *C-List* 内的分类结果存在某些稀疏样本例如 Heartbleed , Wen Attack-Sql Injection 等则会进行优先选择.

3.3 稀疏样本增强

网络流量数据普遍具有数量不平衡的问题即不同类别数据之间的样本数量差距过大. 这个问题将导致 C-DNN 对于某些稀疏样本的分类准确率差. 因此, 本文使用变分自动编码器 (VAE) 对稀疏样本进行过采样来加强 C-DNN 对稀疏样本的分类能力.

VAE 是由 Kingma 等人[21] 在 2013 年提出的一种概率生成模型. VAE 将编码器作为识别模型, 解码器作为生成模型. VAE 的主要目的是对解码器即生成模型 $P_\theta(x|z)$ 做参数估计, 其中 θ 表示神经网络参数, z 表示隐含变量, x 表示要生成的数据. 由于无法直接通过贝叶斯公式得到 $P_\theta(x|z)$, 因此它通过训练编码器即识别模型 $Q_\Phi(x|z)$ 来逼近真实的后验概率 $P_\theta(x|z)$. VAE 一般使用 KL 散度来衡量两个分布的相似程度.

本文为每种稀疏样本单独训练一个变分自动编码器, 然后从符合正态分布的数据中随机采样一些数据后通过 VAE 生成新的样本数据用于 C-DNN 的训练.

4 数据集介绍及预处理

4.1 CICIDS2017 数据集

本文在实验阶段使用 CICIDS2017 入侵检测数据集对 ImFace 算法进行训练和测试. CICIDS2017 是一种由 Sharafaldin 等人[22] 在 2018 年发布的公开入侵检测数据集, 该数据集中是由研究人员使用 CICFlowMeter 软件从原始的网络数据包中获取. CICIDS2017 数据集中的数据包含 6 个基本特征和 78 个功能特征共 84 个网络流量特征. 同时, 该数据集一共包含 15 种不同类别的数据, 其中 1 种类型为正常流量, 另外 14 种为不同的网络攻击类型例如 DDoS, Heartbleed 等.

4.2 数据集预处理

1) 由于数据集中的基本特征如 Flow ID, Protocol, Timestamp, Source IP, Destination IP 和 Source Port 具有网络标识的特点. 因此, 为了防止这些基本特征导致模型产生过拟合, 本文从数据集中删除这 6 个基本特征. 同时, 本文删除了数据集中 288 602 行无标签的数据. 此时, 数据集共由 2 830 743 行数据和 78 个特征组成.

2) 本文分别使用每种类别的平均值, 最大值和最小值替换其特征内的缺失值, 无限值和负值.

3) 本文将数据集中的所有数据进行了标准化处理. 标准化是指将数据按比例缩放, 使数据统一映射到 [0, 1] 区间上. 数据标准化更加有利于模型收敛到最优解, 并且提高模型的精度. 数据标准化的具体表达如式 (1) 所示, 其中 X_{norm} 表示标准化后的值, $X_{\min}$ 表示该特征中的最小值, $X_{\max}$ 表示该特征中的最大值.

$$X_{\text{norm}} = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \tag{1}$$

4) 本文从数据集内随机抽取 12 个类别的数据, 并且将其分为 3 组用来模拟增量学习的环境. 同时, 将抽到的数据集划分为训练集和测试集. 具体的划分情况如表 1 所示.

5 实验与分析

5.1 实验指标

本文在测试阶段对不同算法的精度 (*precision*), 召回率 (*recall*), 准确率 (*accuracy*), *F*1 分数 (*F*1-*score*) 等指标进行计算和对比. 具体的数据公式以及指标说明如下所示.

$$precision = \frac{TP}{TP+FP} \tag{2}$$

$$recall = \frac{TP}{TP+FN} \tag{3}$$

$$accuracy = \frac{TP+FN}{TP+TN+FP+FN} \tag{4}$$

$$F1\text{-}score = 2\times\frac{precision\times recall}{precision+recall} \tag{5}$$

其中, true positive (*TP*) 表示预测为正例, 实际为正例的样本数量; false positive (*FP*) 表示预测为正例, 实际为负例的样本数量; true negative (*TN*) 表示预测为负例, 实际为负例的样本数量; false negative (*FN*) 表示预测为负例, 实际为正例的样本数量; *precision* 表示在所有被模型判断为正例中, 实际正例所占的比例; *recall* 表示被模型正确判为正例占数据集中所有正例的比例; *accuracy* 表示模型判断正确的数据占所有数据的比例; *F*1-*score* 则是结合 *precision* 和 *recall* 的结果计算的综合性能指标.

表 1 不同批次下的数据集划分表

批次	类别	类别编码	数据集		
			训练集	测试集	总计
1	DoS Hulk	1	184 858	46 215	231 073
	FTP-Patator	5	6 350	1 588	7 938
	Bot	9	1 573	393	1 966
	Web Attack-XSS	11	522	130	652
2	PortScan	2	127 144	31 786	158 930
	SSH-Patator	6	4 718	1 179	5 897
	Web Attack-Brute Force	10	1 206	301	1 507
	Heartbleed	14	9	2	11
3	DDoS	3	102 421	25 606	128 027
	DoS Goldeneye	4	8 234	2 059	10 293
	DoS Slowhttptest	8	4 399	1 100	5 499
	Web Attack-Sql Injection	13	17	4	21
Total			441 451	110 363	551 814

5.2 ImFace 算法训练参数

本文所使用的 C-DNN 神经网络一共由神经元个数分别为 1 024, 512, 256, 128, 64, 4 的 6 个全连接层所组成. 同时, 本文在全连接层的前后分别连接 BN 层和 Leaky_ReLU 激活函数, 并在最后一层使用 Softmax 函数对输出数据进行归一化处理. AMAE 和 C-DNN 具体的训练参数如表 2 所示.

表 2 不同模型的神经网络训练参数表

模型	训练轮数	学习率	批尺寸
C-DNN	300	1E–4	1 024
AMAE	900	1E–5	1 024

5.3 基准算法介绍

DNN-batch 是一种基于回放的增量学习算法. 对样本数量较多的数据类别, DNN-batch 通过训练好的 VAE 来计算每条数据的 MSE, 然后根据 MSE 将数据进行从小到大的排序后选取前 5 000 条数据后与所有的稀疏样本用于在 DNN-batch 的增量阶段进行旧类别数据的回顾.

Hoeffding Tree 是一种基于决策树的增量学习算法. 该算法对每条数据进行检查一次后逐步生成一个决策树, 而这些样本在更新决策树以后不需要进行存储.

5.4 实验结果对比与分析

本文对比了 DNN-batch, Hoeffding Tree 和 ImFace 的评估指标, 例如精度, 召回率和 *F*1 分数等. 本文将每批次的数据种类数量分别划分为 4, 8, 12. 在实验过程中, 本文统计了每个批次下不同种类数据中的精度, 召回率和 *F*1 分数. 如表 3 所示, 对于 DNN-batch 算法来说对旧类别数据的回顾能够有效防止模型的灾难性遗忘, 例如 DoS Hulk, FTP-Patator. 然而, DNN-batch 算法随着数据集批次的增加对数据量较少的样本的分类精度开始降低, 例如 Bot. 对于数据量极少的样本如 Web Attack-Xss, Heartbleed, Wen Attack-Sql Injection 等, DNN-batch 则完全丧失了分类能力. 因此, DNN-batch 对于数量多的样本有效, 对于数据量少的样本则会随着批次的增加而逐渐丧失分类能力. Hoeffding Tree 算法则对数据量大的样本具有较强的分类能力, 对于数据量少的样本学习能力较弱弱.

DNN-batch 和 Hoeffding Tree 都是在原有模型的基础上进行增量学习. 因此, 随着数据种类和数量的增多, 模型会对旧知识进行不同程度的遗忘. 通过实验分析 DNN-batch 和 Hoeffding Tree 发现, 这两种算法对数据量少的数据类型分类能力均不佳. 相反, 由于 ImFace 算法使用每批训练集训练一个 C-DNN 模型后不再进行改变, 所以 C-DNN 本质上对数据的分类能力不再降低. 同时, ImFace 算法使用 AMAE 对每一批数据集的特征进行提取, 因此 AMAE 具有分别旧类别数据和新类别数据的能力. 因此, 将 AMAE 的结果作为置信度来选择 C-DNN 的某一个结果作为最终结果不仅能够保留对数据量大的数据的分类能力, 同时保留了对数据量少的数据类型的分类能力.

本文绘制了不同批次中的宏平均 (macro avg) 变化图和加权平均 (weighted avg) 变化图用来直观的表示模型在增量过程中的 *accuracy* 变化. 图 3 中 macro

avg 表示对每个类别的精度, 召回率和 $F1$ 分数进行加和求平均, weighted avg 则根据每个类别样本数量在总样本中所占的比例进行的加权求和. 图 3 表明, ImFace 算法对旧知识的和对新知识的学习能力都比其余两种算法更强, 同时 ImFace 能够很好地保留对旧类别中数量少的样本的分类能力.

表 3　不同算法在不同批次中的实验结果对照表

批次	类别	精确率			召回率			$F1$分数		
		DNN-batch	Hoeffding Tree	ImFace	DNN-batch	Hoeffding Tree	ImFace	DNN-batch	Hoeffding Tree	ImFace
1	Dos Hulk	0.9996	0.9998	0.9998	0.9994	0.9969	1.0000	0.9995	0.9984	0.9999
	FTP-Patator	1.0000	1.0000	0.9975	0.9969	0.9962	0.9981	0.9984	0.9981	0.9978
	Bot	0.9149	0.9947	1.0000	0.9847	0.9618	1.0000	0.9485	0.9780	1.0000
	Web Attack-Xss	1.0000	0.6368	1.0000	0.9000	0.9308	0.9154	0.9474	0.7562	0.9958
2	Dos Hulk	0.9993	0.9919	0.9997	0.9968	0.9992	0.9998	0.9981	0.9955	0.9997
	FTP-Patator	0.9975	1.0000	0.9826	0.9956	0.9931	0.9969	0.9965	0.9965	0.9897
	Bot	0.9792	1.0000	1.0000	0.7176	0.0560	0.9949	0.8282	0.1060	0.9974
	Web Attack-Xss	1.0000	0.7500	0.8815	0.0308	0.0231	0.8981	0.0597	0.0448	0.8981
	PortScan	0.9962	0.9997	0.9995	0.9994	0.9973	0.9997	0.9978	0.9985	0.9996
	SSH-Patator	0.8678	0.9806	0.9727	0.9907	0.9873	0.9686	0.9251	0.9839	0.9707
	Web Attack-Brute Force	0.6997	1.0000	0.9963	0.8904	0.1661	0.8904	0.7836	0.2849	0.9404
	Heartbleed	0.5000	1.0000	1.0000	0.5000	0.5000	1.0000	0.5000	0.6667	1.0000
3	Dos Hulk	0.9999	0.9782	0.9988	0.8930	0.3526	0.9992	0.9434	0.5183	0.9990
	FTP-Patato	0.9900	1.0000	0.9838	0.9969	0.9950	0.9969	0.9934	0.9975	0.9903
	Bot	0.9427	0.9702	0.9287	0.7532	0.9949	0.9949	0.8373	0.9824	0.9607
	Web Attack-Xss	0.0000	0.6000	0.9835	0.0000	0.0231	0.9154	0.0000	0.0444	0.9482
	PortScan	0.9994	0.9987	0.9988	0.9968	0.9986	0.9991	0.9981	0.9987	0.9989
	SSH-Patator	0.8668	0.9615	0.9722	0.9881	0.9949	0.9500	0.9235	0.9779	0.9610
	Web Attack-Brute Force	0.6853	0.9412	0.9962	0.8538	0.1063	0.8738	0.7604	0.1910	0.9310
	Heartbleed	0.0000	1.0000	1.0000	0.0000	0.5000	1.0000	0.0000	0.6667	1.0000
	DDoS	0.8426	0.9998	0.9987	0.9995	0.9981	0.9970	0.9144	0.9989	0.9978
	DoS Goldeneye	0.9197	0.9776	0.9831	0.9908	0.9956	0.9913	0.9539	0.9865	0.9872
	DoS Slowhttptest	0.9354	0.9973	0.9829	0.9873	0.9891	0.9927	0.9606	0.9932	0.9878
	Wen Attack-Sql Injection	0.0000	0.0000	0.7500	0.0000	0.0000	0.5000	0.0000	0.0000	0.6000

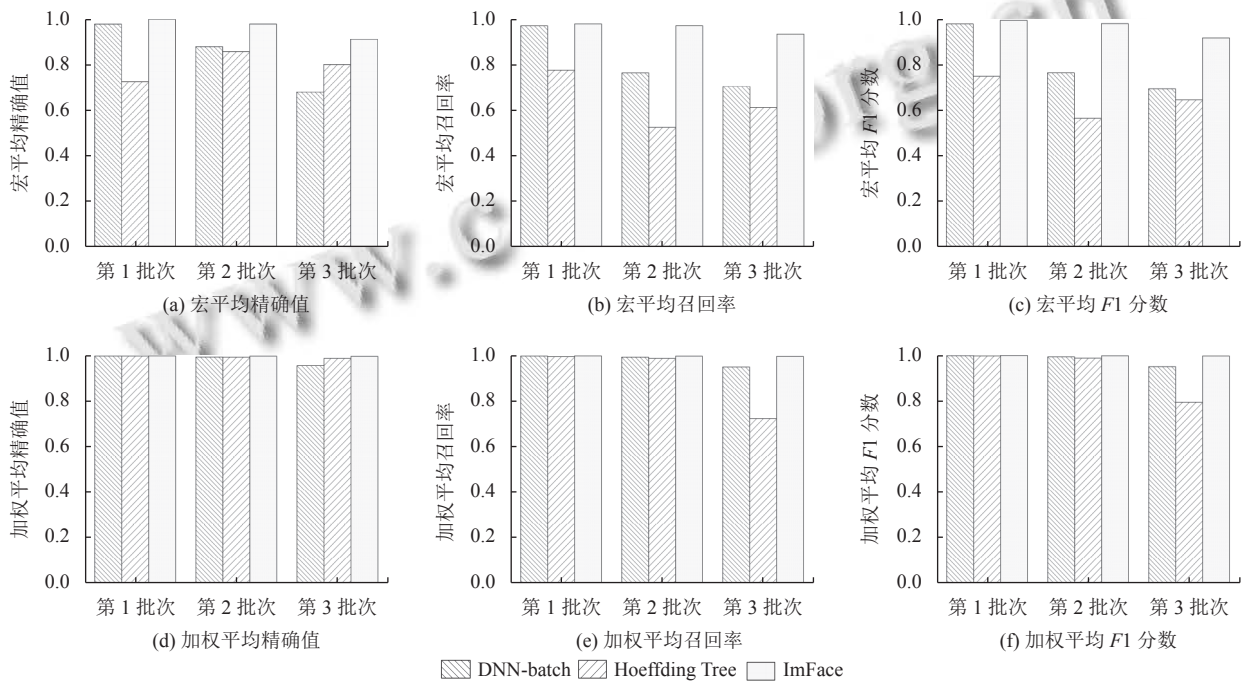

图 3　不同算法在宏观情况和加权情况中的实验结果对照图

5.5 检测时间对比

如表 4 所示, 本文记录了不同算法在测试阶段每批次使用的数据量以及不同算法之间的平均处理时间. 由于 DNN-batch 在增量学习的过程中仅仅改变输出层的神经元, 因此 DNN-batch 对测试集的处理时间是 3 种算法中最短的. 第 1 批次中, 因为 ImFace 仅使用 C-DNN 对测试数据进行分类, 因此 ImFace 的处理时间要比 Hoeffding Tree 更快. 在剩余的批次中, 每增加一部分新的类别数据, Hoeffding Tree 就会生成一颗新的决策树, ImFace 则会新增加一个 C-DNN 和一个 AMAE. 因此, 两种算法在增量学习阶段对数据的处理时间基本相同.

表 4 不同算法对测试集的平均处理时间对照表

批次	种类数量	测试集			平均处理时间 (s)		
		DNN-batch	Hoeffding Tree	ImFace	DNN-batch	Hoeffding Tree	ImFace
1	4	48326	48326	48326	0.39	8.09	4.20
2	8	81594	81594	81594	0.69	20.15	18.66
3	12	110363	110363	110363	0.97	36.58	34.08

6 总结与展望

增量学习主要面临的问题是模型在学习新知识时会对旧知识产生灾难性遗忘. 针对这个问题, 本文提出的基于 AMAE 和 C-DNN 的 ImFace 算法. 通过实验证明, ImFace 算法能够在增量学习过程中保留对旧知识的分类能力. 并且, 本文通过在训练阶段通过 VAE 对数据进行过采样的方法来减少由于数据集不平衡而对 C-DNN 的影响.

ImFace 在未来仍有许多改进的空间. 其中由于每次 ImFace 算法每次增加新类型的数据都会增加一个 AMAE 和一个 C-DNN, 因此如何减少算法的空间占用成为 ImFace 算法未来的改进方向之一. 除此以外, 通过实验结果证明随着新类型数据的增加, ImFace 算法对数据的分类时间也在增加. 所以, 如何减少 ImFace 算法的分类时间也是未来需要解决的一个问题.

参考文献

1 Xiao YH, Xing C, Zhang TN, *et al*. An intrusion detection model based on feature reduction and convolutional neural networks. IEEE Access, 2019, 7: 42210–42219. [doi: 10.1109/ACCESS.2019.2904620]

2 Yang SC, Tan MS, Xia SY, *et al*. A method of intrusion detection based on Attention-LSTM neural network. Proceedings of the 5th International Conference on Machine Learning Technologies. Beijing: ACM, 2020. 46–50. [doi: 10.1145/3409073.3409096]

3 Li ZZ, Hoiem D. Learning without forgetting. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2018, 40(12): 2935–2947. [doi: 10.1109/TPAMI.2017.2773081]

4 Rebuffi SA, Kolesnikov A, Sperl G, *et al*. iCaRL: Incremental classifier and representation learning. Proceedings of the 2017 IEEE conference on Computer Vision and Pattern Recognition. Honolulu: IEEE, 2017. 5533–5542.

5 欧元芳, 缪祥华. 改进灰狼算法优化支持向量机的入侵检测研究. 化工自动化及仪表, 2022, 49(2): 219–226. [doi: 10.3969/j.issn.1000-3932.2022.02.017]

6 田桂丰, 单志龙, 廖祝华, 等. 基于空间降维和多核支持向量机的网络入侵检测. 济南大学学报 (自然科学版), 2021, 35(4): 365–369, 275. [doi: 10.13349/j.cnki.jdxbn.20210201.005]

7 王国华, 伍忠东, 丁龙斌. 改进多层分类策略的随机森林网络入侵检测方法. 兰州交通大学学报, 2022, 41(2): 55–62. [doi: 10.3969/j.issn.1001-4373.2022.02.008]

8 Miah O, Khan SS, Shatabda S, *et al*. Improving detection accuracy for imbalanced network intrusion classification using cluster-based under-sampling with random forests. Proceedings of the 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT). Dhaka: IEEE, 2019. 1–5. [doi: 10.1109/ICASERT.2019.8934495]

9 沈焱萍, 伍淳华, 罗捷, 等. 基于元优化的 KNN 入侵检测模型. 北京工业大学学报, 2020, 46(1): 24–32. [doi: 10.11936/bjutxb2018100005]

10 Wu D, Jiang ZK, Xie XF, *et al*. LSTM learning with Bayesian and Gaussian processing for anomaly detection in industrial IoT. IEEE Transactions on Industrial Informatics, 2020, 16(8): 5244–5253. [doi: 10.1109/TII.2019.2952917]

11 Duan T, Tian YH, Zhang HR, *et al*. Intelligent processing of intrusion detection data. IEEE Access, 2020, 8: 78330–78342. [doi: 10.1109/ACCESS.2020.2989498]

12 Hassan MM, Gumaei A, Alsanad A, *et al*. A hybrid deep

learning model for efficient intrusion detection in big data environment. Information Sciences, 2020, 513: 386–396. [doi: 10.1016/j.ins.2019.10.069]

13 Yi Y, Wu JS, Xu W. Incremental SVM based on reserved set for network intrusion detection. Expert Systems with Applications, 2011, 38(6): 7698–7707. [doi: 10.1016/j.eswa.2010.12.141]

14 Constantinides C, Shiaeles S, Ghita B, *et al*. A novel online incremental learning intrusion prevention system. Proceedings of the 10th IFIP International Conference on New Technologies, Mobility and Security (NTMS). Canary Islands: IEEE, 2019. 1–6. [doi: 10.1109/NTMS.2019.8763842]

15 Xu BH, Chen SY, Zhang HC, *et al*. Incremental k-NN SVM method in intrusion detection. Proceedings of the 8th IEEE International Conference on Software Engineering and Service Science (ICSESS). Beijing: IEEE, 2017. 712–717. [doi: 10.1109/ICSESS.2017.8343013]

16 Roy D, Panda P, Roy K. Tree-CNN: A hierarchical deep convolutional neural network for incremental learning. Neural Networks, 2020, 121: 148–160. [doi: 10.1016/j.neunet.2019.09.010]

17 Tao XY, Hong XP, Chang XY, *et al*. Few-shot class-incremental learning. Proceedings of 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Seattle: IEEE, 2020. 12180–12189.

18 Zhu F, Zhang XY, Wang C, *et al*. Prototype augmentation and self-supervision for incremental learning. Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Nashville: IEEE, 2021. 5867–5876.

19 Data M, Aritsugi M. T-DFNN: An incremental learning algorithm for intrusion detection systems. IEEE Access, 2021, 9: 154156–154171. [doi: 10.1109/ACCESS.2021.3127985]

20 李珊珊, 李兆玉, 赖雪梅, 等. 基于概率神经网络的增量式入侵检测方法. 计算机仿真, 2022, 39(9): 476–482.

21 Kingma DP, Welling M. An introduction to variational auto-encoders. Foundations and Trends® in Machine Learning, 2019, 12(4): 307–392. [doi: 10.1561/2200000056]

22 Sharafaldin I, Lashkari AH, Ghorbani AA. Toward generating a new intrusion detection dataset and intrusion traffic characterization. Proceedings of the 4th International Conference on Information Systems Security and Privacy. Funchal: SciTePress, 2018. 108–116. [doi: 10.5220/0006639801080116]

(校对责编: 孙君艳)