# 基于深度监督离散哈希神经网络的网络入侵检测方法

薛 胤, 魏松杰

南京理工大学计算机科学与工程学院 南京 中国 210000

**摘要** 近年来, 网络应用规模迅速扩张, 网络异常流量和攻击行为严重威胁网络空间安全, 有效检测网络中的攻击行为成为重要研究课题。目前基于人工智能的网络入侵检测方法, 已经成为网络安全领域的研究热点。现有方法大多基于深度学习方法, 局限于两个问题: 一是网络流量数据的维度高, 特征提取难度大; 二是检测模型的泛化能力较差、误报率较高。为了解决这些问题, 提出了一种基于深度监督离散哈希神经网络的网络入侵检测模型, 通过学习目标的哈希表示用于入侵检测。该模型包含一个轻量的多层神经网络和一个基于监督离散哈希的机器学习框架, 采用交替最小化损失函数的方式加速模型收敛, 学习一组可以很好保留同类网络数据相似性、反映不同类型流量之间的差异的定长哈希码, 并可以通过哈希码间的汉明距离来检测网络入侵, 以减少冗余特征及数据降维方法导致的信息损失对最终检测结果的影响。在入侵检测上, 使用多分段索引哈希的方法查询最近邻哈希码以判别流量类型, 实现快速准确的入侵检测。提出的模型在 CIC-IDS2017、NSL-KDD、UNSW-NB15 数据集上进行实验验证, 并在准确率、误报率等度量指标上对模型的性能进行分析评价, 体现了良好的检测准确性和泛化能力。学习到的二进制哈希编码可以有效反映不同类型流量之间的差异。在网络入侵检测上的准确率达到 97%以上, 误报率较其他检测方法有显著提升。

**关键词** 网络安全; 流量建模; 网络入侵检测; 哈希神经网络

**中图法分类号** TN92 DOI 号 10.19363/J.cnki.cn10-1380/tn.2025.07.05

# Network Intrusion Detection with Deep Neural Network for Supervised Learning of Discrete Hash

XUE Yin, WEI Songjie

School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210000, China

**Abstract** In recent years, internet applications are expanding rapidly. Anomalous network traffic and operations seriously threaten the security of cyberspace. Detecting attacks effectively in network has become an important research topic nowadays. Applying AI in network intrusion detection has become a promising research direction. However, there are two main challenges with existing deep learning-based methods: high dimensionality of massive traffic data and poor generalization ability with high false positive rates. To address these issues, we propose a network intrusion detection model based on Deep Supervised Discrete Hash Neural Network, where the learning objective is to obtain hash representations for intrusion detection. This model consists of a lightweight multi-layer neural network and a supervised discrete hash learning framework. The model adopts an alternating minimization approach to accelerate convergence by minimizing the loss function. It learns a set of fixed-length hash codes that effectively preserve the similarity among similar network data and reflect the differences between different types of traffic. The Hamming distance between hash codes is used for intrusion detection, in order to reduce the impact of redundant features and information loss caused by data dimensionality reduction methods on the final detection results. For intrusion detection, a method called Multi-Segment Index Hashing is used to query the nearest neighbor hash codes and determine the traffic type, enabling fast and accurate intrusion detection. The proposed model is experimentally validated on the CIC-IDS2017, NSL-KDD, and UNSW-NB15 datasets, and its performance is analyzed for accuracy and false rate. The results demonstrate that the model processes good generalization and detection capabilities, and the learned binary hash codes can effectively reflect the differences between various types of traffic. The accuracy of the intrusion detection achieved by the model surpassed 97% on the tested datasets, and the false positive rate showed a significant improvement compared to the other benchmark methods.

**Key words** network security; traffic modeling, network intrusion detection; hash neural network

---

**通讯作者:** 魏松杰, 博士, 副教授, Email: swei@njust.edu.cn。

本课题得到国家重点研发计划子课题“内生安全交换机关键技术研究”(No. 2020YFB1804604)、工业互联网创新发展工程项目“工业企业网络安全综合防护平台”(No. TC200H01V)资助。

收稿日期: 2023-11-23; 修改日期: 2024-01-19; 定稿日期: 2025-06-12

## 1 引言

近年来, 网络规模迅速扩张, 网络信息的安全受到日益严峻的挑战, Heartbleed、Web 攻击、渗透、僵尸网络和 DDoS 等多种类型的网络攻击严重威胁网络空间安全。这些攻击行为对于依赖互联网的当今社会已经造成巨大的经济损失, 因此, 有效防范网络中的攻击行为成为时下的一个重要研究课题。现有的网络安全的防御技术主要有数据加密、防火墙和入侵检测系统 (Intrusion detection system, IDS) 等。IDS 按数据来源区分有基于主机和基于网络的入侵检测。基于网络的 IDS (Network intrusion detection system, NIDS) 作为网络安全防护中的关键一环, 旨在通过对网络的实时监测, 从网络上的流量数据中识别出潜在的攻击行为, 为安全管理人员提供响应方案。NIDS 广泛适用于交通和物联网安全等多个应用领域, 并且在这些场景中表现很好, 但是面临当今海量的网络数据, 一些早期的方法不再适用。机器学习相关技术的研究在最近几年进一步深入, 基于机器学习方法的入侵检测受到此领域工作者广泛关注和研究。

传统的网络入侵检测方法可以分为基于误用的检测方法和基于异常的检测方法。前者通过人为描述各种攻击样本的特征和模式并以此作为规则来检测, 应用较为广泛。该方法拥有较高的查准率, 但是需要维护一个昂贵的攻击模式库, 且只能检测已知的攻击。异常检测方法的提出主要是为了应对误用检测方法的不足, 较常用的一些技术有基于隐马尔可夫等随机过程模型的方法、支持向量机、主成分分析法和基于深度学习的方法等。目前入侵检测一个重要的研究方向是采用深度学习的相关方法, 利用已知数据集建立有效的检测模型以寻求对未知数据的异常判定。基于深度学习的方法在网络入侵检测领域取得了不错的成效, 但是这些方法误报率仍然较高, 且泛化能力较差, 同时海量高维数据的处理也是难点之一[1], 有效的特征提取以减轻信息冗余和损失带来的影响是一项具有挑战性的任务。

针对这些方法面临的问题, 本文提出一种基于深度监督离散哈希神经网络的网络入侵检测模型。主要贡献如下:

1) 提出一种哈希神经网络算法, 学习原始网络流量数据的哈希表示, 保留同类数据相似性的同时扩大不同类数据间的差异, 并可以用于精确的网络入侵检测。该方法包含一个轻量的多层神经网络和一个基于监督离散哈希的机器学习框架, 最终将神经网络的输出层限制为一组二进制哈希编码, 学习网络流量数据的哈希表示。模型输出的哈希码可以学习网络流量数据的特征信息并映射到相应的二进制位, 冗余的信息不影响流量数据的区分, 因此不需要和常见的入侵检测模型一样采取降维或者聚类等数据处理方法对数据集进行大量的预处理工作, 降低了对原始数据的处理要求。

2) 在模型训练上, 由于哈希编码的离散特性, 本文采用交替最小化损失函数的方法优化网络权重, 加快模型收敛。

3) 哈希神经网络学习得到一组可以很好保留同类网络数据相似性、反映不同类型流量之间差异的二进制哈希码, 并可以通过计算哈希码间的汉明距离来检测网络入侵, 在此基础上引入多索引哈希算法进行快速、准确的哈希检索。

实验表明, 本文所提模型学习到的二进制哈希编码可以有效反映不同类型流量之间的差异, 在入侵检测上的准确率与误报率相近或优于最新的检测方法。本文的其余部分安排如下: 第 1 节介绍了国内外相关工作; 第 2 节详细介绍了本文提出的入侵检测方法; 第 3 节介绍了实验并对结果进行分析讨论; 第 4 节主要阐述了工作结论。

## 2 相关工作

网络入侵检测系统作为网络安全防护中的关键一环, 旨在通过对网络的实时监测, 从网络上的流量数据中识别出潜在的攻击行为, 为安全管理人员提供响应方案。该系统的优点在于使用单个系统监视整个网络, 从而减少了在各个主机上安装监控软件所需的时间和成本等资源投入。然而, 这种方法存在一个问题, 即由于网络上的监视难以获取到重要的系统内部状态信息, 因此对于入侵检测而言更加困难。

机器学习在 IDS 中的应用相当广泛。贝叶斯网络逻辑简单, 推理能力强, 在该领域有诸多应用。隐马尔可夫模型 (Hidden Markov model, HMM) 是一种经典的机器学习模型, 可以很好的处理基于序列的例如时间或状态序列的相关问题。Xu 等人[2]将 HTTP 有效载荷表示为字节序列, 采用 HMM 分析, 对跨站脚本攻击和 SQL 注入有很好的检测效果。支持向量机 (Support vector machines, SVM) 可以有效解决小样本问题, Shah 等人[3]通过改进 k-means 算法构建高质量的训练数据集, 并结合极限学习机算法和 SVM, 其构建的入侵检测模型能够有效识别拒绝服务攻击。入侵检测数据集通常包含网络流量的基

本特征、流量特征和内容特征。这些特征可以包括 TCP 连接的持续时间、源主机数目以及登陆失败次数等。数据集包含连续和离散数据, 存在冗余和不相关数据, 并且特征之间的数量级差异大。为了减少计算开销, 提升分类器的泛化能力, 层次聚类和主成分分析法 (Principal component analysis, PCA) 被用于大量入侵检测工作中。PCA 是较为常用的数据降维方法, 被大量用于入侵检测工作中, 以降低数据特征维数、减少计算开销, 提升分类器的泛化能力。文献[4]首先使用相关性作为特征选择方法来消除数据集中的冗余和不相关属性, 然后使用主成分分析的降维方法来提高解释性并最小化信息损失, 使得 IDS 模型准确性有所提高。

传统机器学习方法是较为浅层的学习方法, 通常需要人为选取特征, 且需要大量领域内专业知识。Hinton 教授于 2006 年提出深度学习理论[5], 与机器学习不同的是, 深度学习方法学习样本数据的内在规律和表示层次, 是一个复杂的机器学习算法。目前, 深度学习在机器翻译、自然语言处理、多媒体学习等领域取得了重要成果。CNN 与普通神经网络类似, 都由一定数量的具有可学习的权重和偏置常量的神经元构成, 是一种典型的前馈神经网络, 具有深度结构, 但其包含卷积计算, 可以准确且高效地提取数据特征。Pingale 等人[6]利用 CNN 提取有效的特征, 并将特征转换成向量形式使用 RV 系数进行特征选择, 最终提出的深度混合模型可以有效检测网络入侵。RNN 在序列数据中具有很好的表达能力, 可以有效挖掘时序和语义信息。因此, 一些研究尝试使用 RNN 来进行序列相关的入侵检测, 并取得了良好的效果。针对 RNN 存在的梯度消失的问题, LSTM 通过门控制将长短期记忆结合起来, 一定程度上解决了梯度消失的问题。Hassan 等人[7]提出了一种结合了 CNN 和 LSTM 的入侵检测模型。该模型利用 CNN 来提取数据的空间特征, 并通过 LSTM 保留特征之间的依赖关系, 从而有效提取数据的时空特征以提高模型准确率。生成式对抗网络是当前最有研究价值的模型之一, 是由 Goodfellow 等人[8-9]提出的一种新颖的深度学习方法。生成式对抗网络 (Generative adversarial networks, GAN) 包含生成器和鉴别器两个部分。生成器通过学习真实样本数据的概率分布, 生成全新的样本数据。鉴别器则负责判断输入的数据是真实数据还是由生成器生成的假数据。生成式对抗网络可以在数据集数据较少的情况下通过学习少量数据产生新的样本, 从而解决数据集不平衡问题, 提高入侵检测模型的性能。

目前的哈希算法一般可以认为有以下两种类别: 数据无关算法和数据相关算法。各种哈希算法中, 局部敏感哈希[10]是最具代表性的数据无关算法, 它通过随机线性投影来构造散列函数, 将数据映射成二进制代码。然而不可避免的是, 数据无关算法有其特有的局限性——算法不使用训练数据, 因此学习效率较低且要获得较高的准确率和召回率需要相当长的哈希编码。在这种情况下, 机器学习在学习基于给定数据集的高效散列函数方面得到了推广。

数据相关的哈希学习算法旨在使用给定数据集生成短二进制码, 从而可以有效并且高效地检索海量数据。这些方法可以进一步分为有监督方法和无监督方法。无监督方法中相当一部分属于线性散列算法, 其中较为代表性的包括主成分分析散列[11]、迭代量化[12]等。迭代量化中投影矩阵根据给定数据集通过迭代投影和阈值处理进行优化, 最终学习一组超平面作为线性散列函数。有监督方法中有最小损失散列[13]、线性判别分析散列[14]等, 最具代表性的是带有内核的监督哈希[15], 它利用最小化相似数据对哈希码之间的汉明距离, 且最大化不同对之间的汉明距离来学习数据的哈希表示。基于机器学习的哈希算法的主要难点在于处理对所追求的哈希码的具体约束, 监督离散哈希[16]通过引用辅助变量, 采用离散循环坐标下降法得出更优的哈希编码。

近年来, 深度学习神经网络逐渐被推广到哈希算法的学习上来, 并且证明了与常规机器学习相比具有更好的性能。卷积神经网络[17]是较早将深度神经网络用于哈希学习的研究之一。深度语义排名散列[18]利用深度学习模型来发现更深层次的语义相似性, 并且可以在大型训练集上很好地扩展。虽然基于深度学习的方法在图像检索方面取得了不错的成果, 但在语义信息的完整性等方面仍有一些不足。最新的研究尝试将学习过程分为两个流: 哈希流和分类流。其中, 哈希流用于学习哈希函数, 而分类流则用于挖掘语义信息。Li 等人[19]利用 CNN 同步学习哈希函数和图像表示, 在输出层根据成对的分类和标签信息直接输出二进制编码。

深度学习在入侵检测领域中表现出比传统机器学习更加优异的性能, 但仍存在泛化能力差、误报率高等问题, 且海量高维网络流量数据的处理也面临挑战。

## 3 网络入侵检测方法

本节对提出的网络入侵检测方法进行介绍, 第一部分为网络入侵检测方法描述, 第二部分对基于

深度监督离散哈希(Deep supervised discrete hashing, DSDH)神经网络的网络入侵检测模型的具体流程作进一步说明, 第三部分阐述监督离散哈希框架下的DSDH 算法。

### 3.1 方法描述

本文提出基于 DSDH 的网络入侵检测方法, 其包含一个轻量的多层神经网络和一个基于监督离散哈希的机器学习框架。该模型的整体流程包括数据收集及预处理、哈希神经网络构建、入侵检测模型训练与优化、网络流量数据异常检测等模块(见图 1)。具体步骤如下: 首先, 通过在线实时数据收集模块获取网络流量数据, 提取数据的相关统计特征并对其进行预处理工作; 其次, 基于监督离散哈希的机器学习框架构建哈希神经网络模型; 接着, 在获取网络流量特征数据后, 训练优化所构建的哈希神经网络, 更新网络权重参数, 通过哈希神经网络映射出网络流量数据的哈希编码; 最后, 将学习到的哈希编码用于网络流量数据的异常检测, 获取检测结果以判别网络入侵。

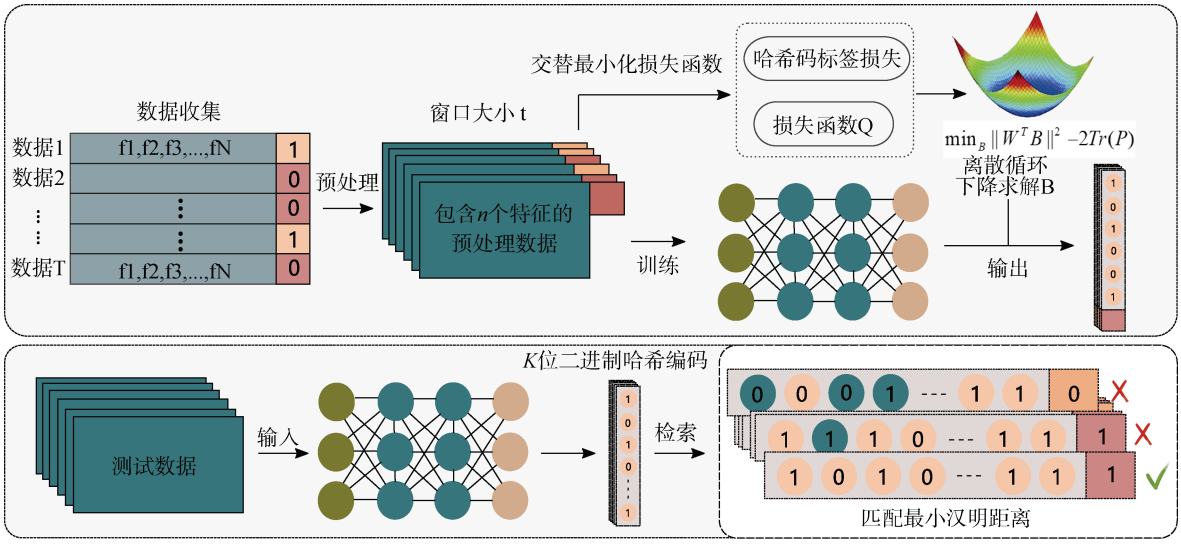


**图 1 基于 DSDH 的网络入侵检测流程图**

**Figure 1 Flow chart of network intrusion detection based on DSDH**

### 3.2 模型流程

#### 3.2.1 数据收集

为了验证本文入侵检测模型在原始网络流量数据上的通用性, 本文的实验数据选用 CIC-IDS2017[20]、NSL-KDD[21]、UNSW-NB15[22]三种数据集。其中, 实验结果以 CIC-IDS2017 数据集为主展开讨论。该数据集捕获共计 5 天的网络流量数据, 多种网络攻击被成功实现且分别于周二、周三、周四和周五上午和下午被执行。其包含了基于 HTTP、HTTPS、FTP、SSH 等网络协议在内的大量真实网络流量数据, 包含了良性和最新的常见攻击, 采集的攻击包括 Web 攻击、渗透、僵尸网络和 DDoS 等 8 种类型, 通过对其进行特征提取, 最终生成了 80 余条特征, 并以标签形式标明了所属类型。

#### 3.2.2 数据预处理

**1) 数据标准化**

数据标准化常被用于深度学习方法中, 实验表明模型训练时使用相同标度的数据可以有效提高训练精度, 在一定程度上加快模型的收敛。实验中, 由于选取的数据集中存在 Nan 和 Infiniti 脏数据, 因此首先需要对数据集进行整合以及脏数据的剔除。接着, 由于网络流量数据的数值跨度较大, 所以不同于其他例如图像数据使用的等比例缩放方法, 本文依据式 1 对网络流量数据进行缩放, 最终使经过处理之后的数据都处于同一数量级, 且有效降低大数值数据之间的差异性。

$$f(x)=1-\frac{a}{b\log(1+x)+1} \tag{1}$$

其中, $a$、$b$ 为权重参数, 实验中设为 100、10.

**2) 特征选择与特征编码**

对于常见的例如基于 CNN、基于 AE 的入侵检测模型, 特征选择可以清除一些不必要的特征, 从而减少数据集数据的维数, 降低模型后续的训练难度。对于本文提出的基于哈希神经网络的入侵检测模型而言, 神经网络最后一层输出被严格限制为二进制哈希码, 因此网络流量数据经过训练可以由原始数据散列为二进制特征的状态位。一般认为对于重要的特征不同类型网络流量对应的神经网络输出应当是不同的, 而不重要的特征不同类型网络流量对应的网络输出可以是相同的, 并且可以用汉明距

离来量化不同类型网络流量最终的输出之间的差异, 因此可以降低对原始数据的处理要求。实验中最终分别从 CIC-IDS2017 和 NSL-KDD 数据集中选取了 78 和 36 个特征。

CIC-IDS2017 数据集的标签信息为不能参与训练的非数字特征符号, 因此需要将非数字的特征符号转换为数字类型。处理数据集中的这些特殊特征值, 一般采用标签编码和 One-hot 编码两种编码方式来解决, 其中 One-hot 编码方式更加适合深度学习算法。One-hot 编码方案采用多位状态位的形式对多种状态编码, 本文通过 One-hot 编码处理将 CIC-IDS2017 数据集中的标签特征映射成一个多维向量, 使数据集符合模型的输入要求。

#### 3.2.3 模型训练与优化

本文设计包括两层隐藏层以及输入输出层的轻量神经网络模型, 并将网络的输入节点数为网络流量数据特征数, 输出节点数设为哈希码长度。神经网络各层采用全连接, 应用非线性激活函数拓宽最终输出函数, 并作为前向传播函数向前迭代神经网络中每一层各节点的值。

基于机器学习的哈希算法的主要难点在于处理对所追求的哈希码的具体约束, 本文引入一种基于监督离散哈希的机器学习框架, 该方法通过引用辅助变量, 重新描述设想的学习目标, 使其可以使用离散循环坐标下降法求解一个关键的正则化子问题(式 14), 从而直接优化二进制哈希码, 最终获得高质量的网络流量数据二进制哈希编码。在本文哈希神经网络的训练上, 各神经节点权重数据具有连续性, 而模型输出节点限制为离散编码, 这使模型的收敛难度较大。不同于一般神经网络的节点权重更新过程, 本文在模型的训练上, 分别计算所得二进制编码与标签的损失(式 3)以及神经网络每次迭代预测与标签的差值(式 5), 交替最小化两个损失函数以修正各节点权值, 从而使模型的整个优化过程非常高效。

#### 3.2.4 网络流量数据异常检测

在网络入侵检测的基本框架中, 哈希编码大幅度提升了入侵检测的速度。基于哈希码的线性扫描检索性能很好, 但是相对于检索哈希表的常量级检索时间, 其代价仍然很高, 为了进一步加快检测速度, 本文在异常检测中使用多索引哈希算法 (Multi-index hashing, MIH)。步骤如下:

1) 将训练集得到的一组二进制哈希码中每一条分割为多个连续不重合的子串, 建立索引, 为每一段哈希码建立一个哈希表;

2) 检索时, 同样将测试数据的哈希码划分为多个子串, 在对应的哈希表中进行查找, 得到候选结果;

3) 依据测试哈希码与候选结果两者的汉明距离排序, 得到最近邻。

基于该算法, 对于一个 $b$ 位哈希码 $h$, 将其分为 $m$ 段, 在查询与哈希码 $h$ 的汉明距离为 $r$ 的所有哈希码时, 分别对这 $m$ 段哈希码进行查找, 找出在每一段哈希码中汉明距离小于$\lfloor r/m \rfloor$的查询结果, 将 $m$ 段的查询结果合并之后即为最终的候选集。相对于线性搜索需要查找 $L(b, r)$ 个哈希桶的情况, 该方法只需要搜索 $m*L(b/m, \lfloor r/m \rfloor)$ 个桶, 极大提高了检索效率。

本文的异常检测模块将测试的网络流量数据通过哈希神经网络映射为一组二进制码, 在训练得到的带有标签数据的哈希码集合中检索最近邻, 并通过其标签判别网络流量数据类别, 实现高效的入侵检测(如图 1 所示)。

### 3.3 哈希神经网络

#### 3.3.1 哈希目标

给定N个网络数据流量样本 $X=\{x_i\}_{i=1}^{N}$, $x_i \in \mathbb{R}^{\tau \times 1}$, $\tau$ 表示样本特征数, 通过哈希神经网络学习一个 $K$ 位二进制编码集合 $B \in \{-1,1\}^{k \times N}$, 其中第 $i$ 个二进制编码 $b_i \in \{-1,1\}^{K}$ 表示第 $i$ 个样本的哈希码。这些哈希码通过哈希函数 $h(\cdot)$ 生成, 可以表示成 $[h_1(\cdot),\cdots, \ h_K(\cdot)]$。对于样本 $x_i$, 其哈希码可以表示成 $b_i = h(x_i) = \ [h_1(x_i),\cdots,h_K(x_i)]$。总的来说, 样本数据将通过哈希神经网络投射为一组哈希编码, 该哈希码可以有效反映不同类样本数据之间的差异性, 并通过排序哈希码间的汉明距离来进行网络流量的异常判别, 识别网络入侵。汉明距离为不匹配的数量, 即要把字符串 $A$ 转换成字符串 $B$ 所需的最小替换操作次数。对于任意两个等长的二进制编码 $a,b \in \{-1,1\}^{j}$, 它们两者之间的汉明距离可以通过一个 $OR$ 运算来计算, 即: $d_h(a,b)=\sum_{i=1}^{\mathrm{j}} xor(a_i,b_i)$。

#### 3.3.2 深度监督离散哈希

本文设想学习的二进制哈希编码应该使两个同类网络流量数据间的汉明距离尽量小, 同时使两个不同类数据间的汉明距离尽量大。首先用一个常见的线性分类器来建模二进制编码和数据标签两者的关系:

$$Y = W^T B \tag{2}$$

其中 $w_k \in R^{K \times N}, k=1,\cdots,d$ 为分类器权重, $d$ 为样本

类别数, $Y=[y_1,y_2,\cdots,y_N]$ 为真实的标签向量。损失函数 $Q$ 可以计算为:

$$Q=\sum_{i=1}^{N}L(y_i,W^Tb_i)+c\|W\|_F^2 \tag{3}$$

其中 $L(\cdot)$ 为损失函数, $c$ 为正则化参数, $\|\cdot\|_F$ 为矩阵的 Frobenius 范数。对于神经网络, 样本标签信息可以表示为 $Y=\{y_i\}_{i=1}^N\in R^{d\times N}$。给定语义标签信息, 成对标签信息间的相似性可以表示为: $M=\{m_{ij}\}$, $m_{ij}\in\{0,1\}$。当 $x_i$ 与 $x_j$ 语义相近时, $m_{ij}=1$, 反之, $m_{ij}=0$。对于两个二进制码 $b_i,b_j$, 可以使用汉明距离来衡量它们的相似性, 并可以使用它们的内积来具体的量化, 计算公式为:

$$DistaH(b_i,b_j)=(K-<b_i,b_j>)/2 \tag{4}$$

接下来使用负对数似然函数作为神经网络的损失函数, 损失函数可以表示为:

$$P=-\sum_{m_{ij}\in M}(m_{ij}D_{ij}-\log(1+e^{D_{ij}})) \tag{5}$$

其中 $D_{ij}=<b_i,b_j>/2$。结合式 3 和式 5, 可以得到:

$$\begin{aligned}L&=P+uQ\\&=-\sum_{m_{ij}\in M}(m_{ij}D_{ij}-\log(1+e^{D_{ij}}))\\&\quad+u\sum_{i=1}^{N}L(y_i,W^Tb_i)+v\|W\|_F^2\end{aligned} \tag{6}$$

其中 $u$ 为权重参数, $v=cu$。接下来选择线性分类器的 $l_2$ 损失, 将公式 6 改写为:

**算法 1** 基于深度监督离散哈希的神经网络算法

**输入**: 训练数据 $\{x_i,y_i\}_{i=1}^n$; 哈希码长度 $L$; 训练次数 $T$; *Batch_size* $M$; 超参数 $u,c,t$

**输出**: 哈希函数 $H(x)=sgn(F(x))$ (神经网络参数)

1. 从训练数据中选取 $m$ 个训练样本 $\{s_j\}_{j=1}^m$
2. FOR *epoch* = 1, 2, …, $T$ DO
3. 随机排序 $m$ 个样本;
4. FOR *iter* = 1, 2, …, *n/M* DO
5. 从排序好的训练数据中选取 $M$ 个样本作为输入进行训练;
6. 根据公式 9 计算梯度;
7. 根据公式 10 更新神经网络参数 $G, n$ 和 $g$;
8. 根据公式 12 更新神经网络权重 $W$;
9. 根据公式 15 更新哈希码;

$$\begin{aligned}L&=-\sum_{m_{ij}\in M}(m_{ij}D_{ij}-\log(1+e^{D_{ij}}))\\&\quad+u\sum_{i=1}^{N}\|y_i-W^Tb_i\|_2^2+v\|W\|_F^2\end{aligned} \tag{7}$$

公式中 $\|\cdot\|_2$ 是向量的 $l_2$ 范数。文献[16-17]通过引入辅助变量将公式进一步放宽, 将公式 7 分解为两个子优化问题。这里 $I_{ij}=h_i^Th_j/2, h_i(i=1,\cdots,N)$ 为最终全连接层的输出:

$$\begin{aligned}F&=-\sum_{m_{ij}\in M}(m_{ij}I_{ij}-\log(1+e^{I_{ij}}))\\&\quad+u\sum_{i=1}^{N}\|y_i-W^Tb_i\|_2^2+v\|W\|_F^2\\&\quad+t\sum_{i=1}^{N}\|b_i-h_i\|_2^2\\&\quad s.t.\ b_i\in\{-1,1\}^K,(i=1,\cdots,N)\end{aligned} \tag{8}$$

$$h_i=G^T\,\mathrm{z}(x_i;g)+n \tag{9}$$

其中 $g$ 为最终全连接层前的各层参数, $G\in R^{s\times K}$ 表示权重矩阵, $n\in R^{K\times 1}$ 是偏差项。接下来使用交替最小化方法迭代求解。首先, 当确定 $b_i,W$ 时, 更新参数 $G, n$ 和 $z$:

$$\begin{aligned}\frac{\partial F}{\partial G}&=z(x_i;g)\left(\frac{\partial F}{\partial h_i}\right)^T\frac{\partial F}{\partial n}\\&=\frac{\partial F}{\partial h_i}\frac{\partial F}{\partial z(x_i;g)}\\&=G\frac{\partial F}{\partial h_i}\end{aligned} \tag{10}$$

梯度将通过反向传播(BP)算法传播到前一层。然后, 确定 $G,n,g$ 和 $b_i$, 计算 $W$:

$$F=u\sum_{i=1}^{N}\|y_i-W^Tb_i\|_2^2+v\|W\|_F^2 \tag{11}$$

方程 11 是一个最小二乘问题, 它有一个封闭形式的解:

$$W=\left(BB^T+\frac{v}{\mu}I\right)^{-1}B^TY \tag{12}$$

其中 $B=\{b_i\}_{i=1}^N\in\{-1,1\}^{K\times N}, Y=\{y_i\}_{i=1}^N\in R^{C\times N}$. 最后, 确定 $G,n,\mathrm{z}$ 和 $W$, 式 11 可以写成:

$$\begin{aligned}F&=u\sum_{i=1}^{N}\|y_i-W^Tb_i\|_2^2\\&\quad+t\sum_{i=1}^{N}\|b_i-h_i\|_2^2\\&\quad s.t.\ b_i\in\{-1,1\}^K,(i=1,\cdots,N)\end{aligned} \tag{13}$$

本文使用离散循环坐标下降法逐行迭代求解 $B$:

$$\min_B \| W^T B \|^2 - 2Tr(P), s.t. B \in \{-1,1\}^{K \times N} \quad (14)$$

其中 $P = WY + \frac{t}{u} H$ .可以推导出:

$$x = sgn(p - B_1^T W_1 w) \quad (15)$$

哈希码中各编码位都是通过迭代计算得到的, 依次根据预先学习的 $K$–1 位计算。然后不断更新每一位的值, 直到最终收敛。

## 4 实验

### 4.1 实验数据

本文的实验数据选用 NSL-KDD、UNSW-NB15 以及 CIC-IDS2017 数据集, 其中实验结果主要基于 CIC-IDS2017 进行讨论。以下对本文实验用到的部分网络攻击数据作简要说明:

1) 拒绝服务攻击(DoS):指通过向目标发送大量无用的数据包, 从而占用其大部分系统资源, 使系统无法正常运行的攻击行为。

2) 分布式拒绝服务攻击(DDoS):指利用多个位于不同位置的攻击者或单个攻击者控制的多台机器, 同时发送大量的请求或数据包给目标服务器, 使其过载、崩溃或无法正常提供服务。

3) 端口扫描攻击(PortScan):指通过与目标主机建立连接的方式, 查探目标主机上提供的服务以及处于激活状态的端口, 通过扫描目标主机的开放端口, 获取系统信息, 识别潜在的漏洞或弱点。

4) User-to-Root(U2R):指利用操作系统或应用程序中的漏洞, 从普通用户权限提升为具有超级用户(root)权限的攻击行为。

5) Remote-to-Local(R2L):指通过从远程系统向本地系统发起攻击, 试图获取本地系统的访问权限或执行未经授权的操作。

### 4.2 模型构建与训练

实验中神经网络输入节点数设为 78, 两层隐藏层神经元数量设为 64, 48, 输出节点数设为哈希码长度 (12、24、48、96)。设定训练过程中 *Batch_size* 为 8, *Learning_rate* 为 1*e*-5, 训练迭代次数为 120 *epoch*。首先将预处理好的网络流量数据作为输入通过哈希神经网络进行训练, 调整学习率和超参数使网络更好地收敛, 并保存训练好的模型。然后将测试数据以及所有数据集中的数据通过训练好的模型计算得到相应的哈希编码。最后, 对于每一条测试数据, 从输出的训练集数据的哈希编码中匹配汉明距离最小的网络流量数据, 对比测试数据与匹配数据的标签是否相同, 并以此计算准确率、误报率和 $F1$ 分数。

本文主要使用以下三种指标来评估入侵检测模型的性能, 即准确率、误报率、$F1$ 分数, 有关定义如表 1 所示。计算公式如下:

**表 1 混淆矩阵**

**Table 1 Confusion matrix**

| 预测\实际 | 异常 | 正常 |
|---|---|---|
| 异常 | TP | FN |
| 正常 | FP | TN |

$$AC = \frac{TP + TN}{TP + FN + FP + TN}$$

$$FPR = \frac{FP}{FP + TN}$$

$$F1 = 2\frac{PR * RE}{PR + RE}$$

### 4.3 实验结果和分析

本文实验通过哈希神经网络将网络流量数据的多维特征映射至定长哈希编码, 设定训练过程中 *Batch_size* 为 8, *Learning_rate* 为 1*e*-5, 训练迭代次数为 120 *epoch*。对于多分类问题, 随机从目标数据集中选取一定比例各类型网络流量数据, 以 3:1 的比例划分训练集和测试集。二分类问题中的两类数据分别选择 3000 条数据进行组合, 仍以 3:1 的比例划分。以 CIC-IDS2017 为例, 实验数据的实例数如表 2 所示, 除较少数据的个别类别外, 多数网络流量数据选取 8000 条。除图 2 的训练过程由测试集测试结果计算得来以外, 其余实验结果均由占全部类型数据 20% 的验证集计算得出。

**表 2 CIC-IDS2017 实验数据集中的实例数**

**Table 2 The number of instances in the CIC-IDS2017 experimental dataset**

| 攻击类型 | Total | Train | Test |
|---|---|---|---|
| Benign | 2271320 | 6000 | 2000 |
| DDoS | 128025 | 6000 | 2000 |
| DoS | 230124 | 6000 | 2000 |
| Web Attack | 652 | 450 | 150 |
| Port Scan | 158804 | 6000 | 2000 |
| Botnet | 1956 | 1200 | 400 |
| Infiltration | 36 | 30 | 6 |

#### 4.3.1 模型可行性及性能

**多分类** 各数据集不同位数哈希码 (12, 24, 48, 96) 在模型训练过程中的 *Accuracy* 如图 2 所示, 实验结果选取前 80 *epoch*。从图 2 中可以看出, 模型在各数据集上训练迭代次数达到 40 *epoch* 左右时即收敛, 最终准确率都在 97%以上。另外, 不同哈希码长度对

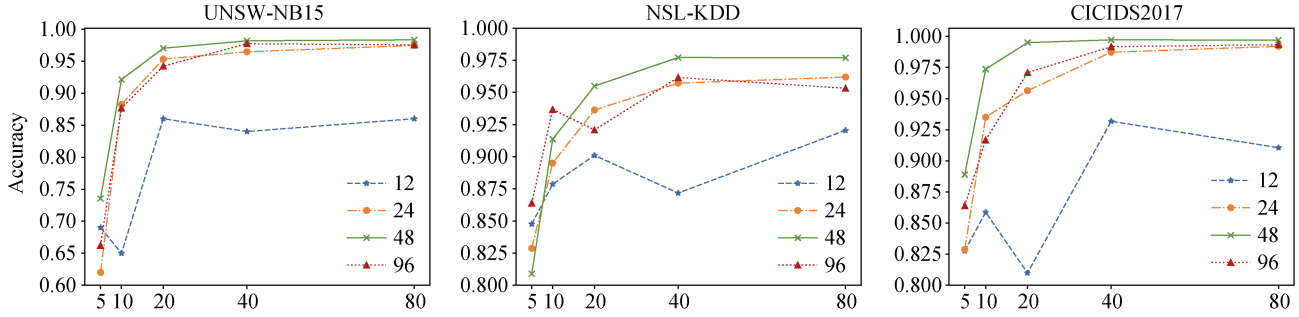


**图 2　不同位长哈希码的 Accuracy 指标**

**Figure 2　Accuracy metrics for hash codes of different digit lengths**

训练时模型精度的影响程度有差异，相比于其他位数的哈希码，12 位的哈希码在训练过程中的表现较为不稳定。哈希编码由流量数据的全局信息得出，因此在编码长度较小时，一位哈希编码的改变会很大程度上影响整个哈希码对于流量数据特征信息的反映，从而影响模型训练的精度。而当哈希码位数多于数据特征维数，流量数据的一个特征信息会映射到多位哈希编码上，导致提取特征的稀释，弱化哈希码的特征表达能力。在图 2 展示的训练过程中，很显然的是较低和较高位数的哈希码都不能够稳定表达网络数据的特征。24 位、48 位、96 位哈希码的训练过程比较平稳。在最终的精度表现上，48 位显然优于 12 位、24 位和 96 位哈希码，表明一定程度上适当长度的哈希码更能够反映出网络流量数据的特征信息，对应的神经网络模型拥有相对更好的检测性能。从训练结果看，48 位哈希码达到了最高的训练精度且训练过程平稳，表现出最佳的性能。为了进一步评价基于 DSDH 的网络入侵检测算法的可行性及性能，接下来选取 48 位编码的哈希神经网络对不同网络攻击类型的流量数据进行评测。从表 3 可以看出，本文提出的入侵检测模型在对不同类型网络流量数据的异常检测上均取得了很好的检测效果，对各类常见的网络攻击类型的检测准确率均达到了 97%以上。模型在多数据集上都能保持高精度，且只包含部分数据的测试集在训练中达到了很高的准确率，在对超两倍于测试集数据量的验证集进行检测时，依然保持了较高的准确率，验证了入侵检测模型良好的泛化能力。

**表 3　不同类型网络攻击的检测结果**

**Table 3　Detection results of different types of cyber-attacks**

| 攻击类型 | *Accuracy* | *Precision* | *F1-score* | *FPR* |
|---|---|---|---|---|
| DDoS | 99.13 | 98.41 | 97.72 | 0.83 |
| DoS | 99.45 | 97.55 | 98.88 | 0.96 |
| Web Attack: XSS | 76.62 | 65.93 | 68.13 | 36.41 |
| PortScan | 99.93 | 99.19 | 99.25 | 1.02 |
| Botnet | 99.22 | 98.94 | 98.7 | 0.99 |
| Infiltration | 100 | 100 | 100 | 0 |
| **AVG** | **99.43** | **98.74** | **99.15** | **1.12** |

此外，本文对不同类型网络流量数据的哈希码间的差异作出了评估。实验采用 48 位长哈希码的 DSDH 模型，分别量化以 DDoS、DoS、PortScan 等网络攻击为主要异常检测目标的 48 位入侵检测模型学习得到的哈希码的一般分布(计数为 1 的编码位)，为了便于观察，其中均为 1 或者均为 0 的编码位计数为 0。具体量化差异性的实验结果如图 3 所示，不同类别网络流量数据学习得到的哈希码间存在明显差异。其中 BENIGN 类型数据哈希码的分布较为分散，而 DoS、DDoS、PortScan 类型较为集中，显示了正

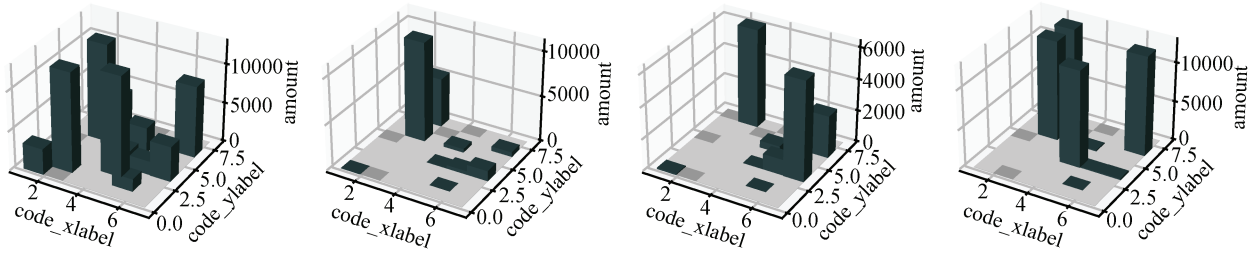


**图 3　不同类型网络流量数据哈希码的一般分布**

**Figure 3　The general distribution of hash codes for different types of network traffic data**

常流量数据特征分布分散, 攻击类型数据特征较为一致的数据特性。实验表明哈希神经网络可以使得相近数据特征的网络流量学习到的哈希编码尽可能相似, 反之则尽量不同。

**二分类** 在二分类问题上, 本文对不同模型学习的哈希码间的差异性作出了评估。实验采用 BENIGN+任一网络攻击类型数据集构建训练集训练的 48 位入侵检测模型, 以 BENIGN 正常流量的平均哈希码(取各编码位众数)为基准, 分别量化以 DDoS、DoS、PortScan 为异常检测目标的 48 位检测模型对不同网络攻击学习得到的哈希码间的平均汉明距离。其中所提及的基准哈希码仅代表正常流量哈希码的一般分布, 各类型较于此的汉明距离可以体现相互之间的差异。实验表明各类网络攻击流量学习得到的哈希码与正常流量的哈希码汉明距离很大, 都在 27 以上, 对比正常流量哈希码有 56%以上的差异, 因此模型可以有效地检测各类网络攻击。图 4 为 BENIGN+DDoS 数据集原数据与哈希码 PCA 降维至三维的 3D 图像, 其中两类数据各占 50%, 总共 6000 条数据。图像显示, 原数据降维后同一类型数据的映射点较为分散, 不能作出一定的区分, 而训练得到的哈希编码降维后出现一定程度的聚类效果, 意味着原始数据中属于同一类别或相似的数据点可能在哈希结果中靠近彼此。

基于此, 接下来实验仍以上述的实验条件进行, 总结了部分实验数据最终的哈希码与基准的汉明距离分布, 结果如图 5 所示, 可以看出:

1) 攻击流量数据与基准的汉明距离分布较为集中, 且与 BENIGN 类型有明显差异, 从汉明距离的差异上可以很准确的在两者之间作出区分。

2) DDoS、DoS 攻击流量的哈希码与正常流量哈希码相比两者的汉明距离接近, 和 PortScan 类型有明显差异。PortScan 类型由于其相似的攻击模式, 在哈希码上有着极为相似的特征表达, 因此在与基准的汉明距离分布上尤为集中。

实验结果反映出相似网络攻击哈希码的汉明距离接近, 而攻击模式差异较大的网络攻击间哈希码差异会很大, 证明了模型对网络流量数据有很好的特征表达能力。

### 4.3.2 同类工作比较

基于上一小节的分析, 本文提出的入侵检测模型可以有效学习网络流量数据的哈希表示, 并用于网络入侵检测。本节在 *AC*、*FPR*、*F*1 等指标方面进一步评估模型在多分类问题上的表现并与部分最近的工作[23-26]比较(见表 4), 其使用的数据集包含了 CIC-IDS2017 和 NSL-KDD。数据质量和训练算法是影响入侵检测能力的两个关键因素, 文献[23]使用朴素贝叶斯特征转换技术构建高质量数据集, 训练后的 SVM 分类器在 CIC-IDS2017 上可以达到 98.84%的分类准确率; HELAD 通过有机整合深度学习技术设计了一种异构集成算法, 首先用少量数据训练自编码器以对数据进行异常评分标记, 然后使用带有异常评分的标签数据训练 LSTM, 提高算法的适应性和准确性。本文提出学习网络流量数据的有效哈希表示, 减少冗余特征对检测结果的影响, 提高模型的泛化能力和检测性能, 并引入多分段索引哈希算法在汉明空间中检索哈希码以进行快速准确的入侵检测。在 CIC-IDS2017 数据集的表现上, 就 *AC* 性能而言, 本文的方法对比 NB-SVM 算法提高 0.6%, 低于 HELAD 算法 0.43%; 在 *F*1 方面, 比 NB-SVM

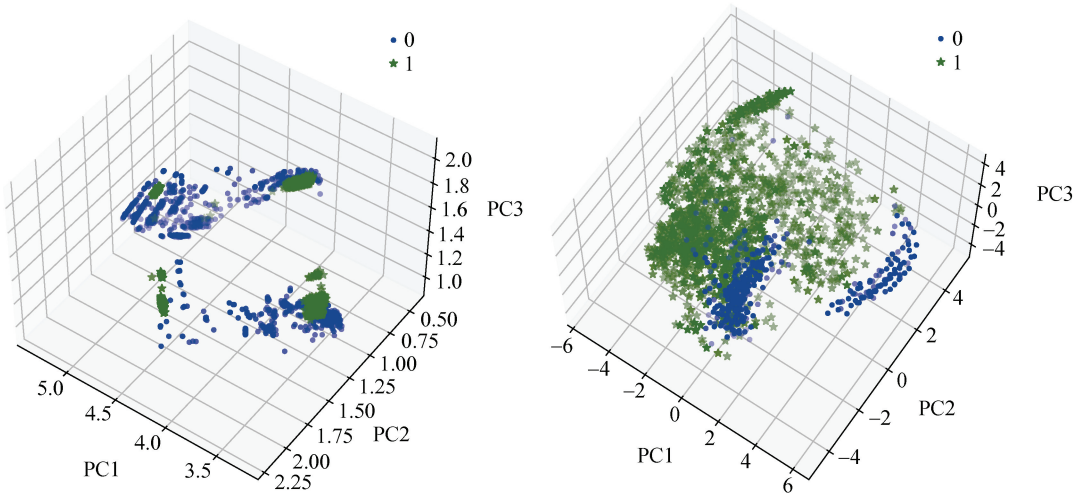


**图 4 BENIGN-DDoS 原数据(左)与哈希码 PCA(右)降维**

**Figure 4 BENIGN-DDoS raw data (left) and hash code (right) PCA dimensionality reduction**

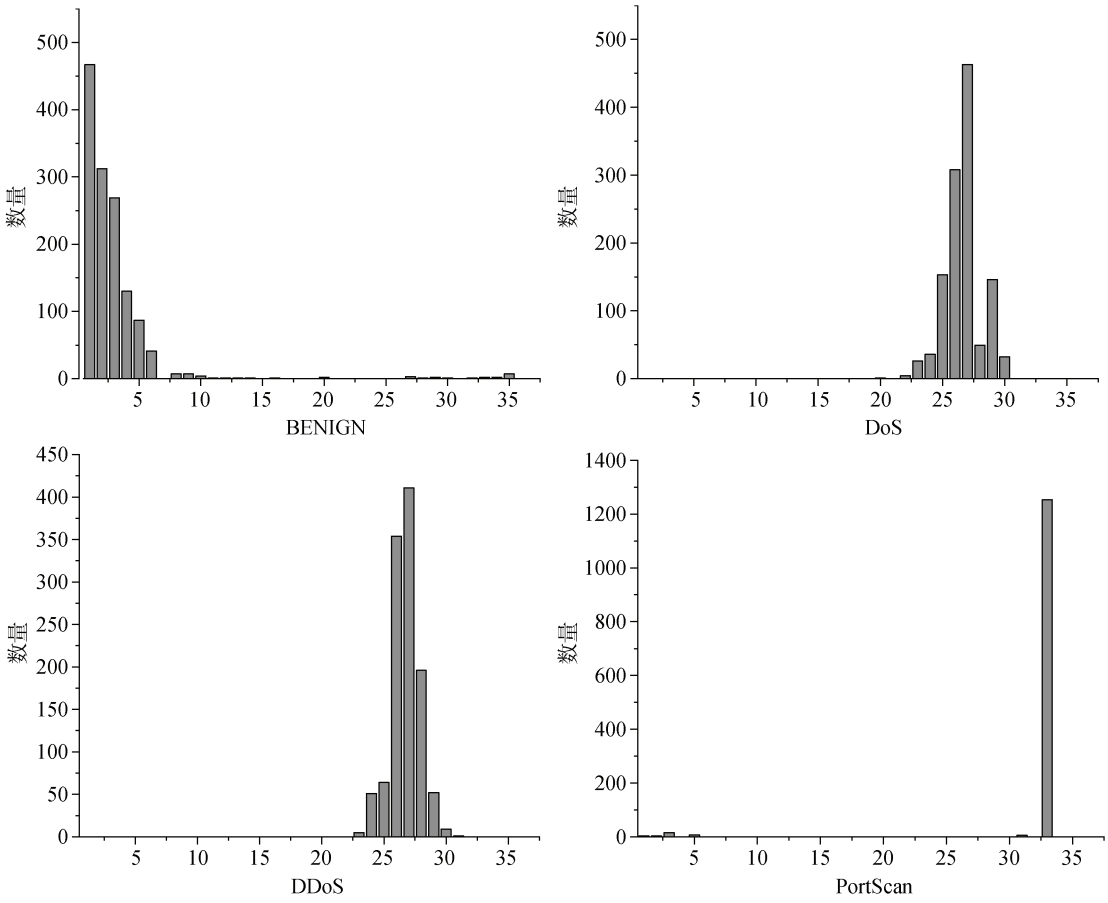


**图 5　不同网络攻击哈希码与基准的汉明距离分布**

**Figure 5　Distribution of Hamming distances between hash codes of different network attacks and the baseline.**

**表 4　实验结果和比较**

**Table 4　Experimental results and comparisons**

| 数据集 | 模型 | *AC*(%) | *FPR*(%) | *F*1(%) |
|---|---|---|---|---|
| CIC-IDS2017 | NB-SVM[23] | 98.84 | 3.55 | 95.04 |
| | HELAD[24] | 99.86 | 2.15 | 99.58 |
| | **OURS** | **99.43** | **1.12** | **99.15** |
| NSL-KDD | RNN[25] | 83.28 | 3.06 | - |
| | A-DQN[26] | 97.2 | 1.24 | 97.8 |
| | **OURS** | **97.62** | **0.87** | **98.51** |

提升 4.32%, 比 HELAD 低 0.43%; 在 *FPR* 指标上, 对比 NB-SVM 和 HELAD 算法分别有 68.4%、47.9%的提升。在使用了 NSL-KDD 数据集的部分工作中, RNN 在入侵检测方面有较强的建模能力, 可以有效捕捉时间序列信息, 在 *AC* 和 *FPR* 性能表现上达到了 83.28%和 3.06%; A-DQN 在多个分布式代理中采用了 Deep Q-Network 逻辑, 设计了一个分布式攻击检测平台, 从而降低检测误报率, 并在基准数据集上进行了广泛的实验验证。就 *AC* 性能而言, 本文方法对比 RNN 提升 17.2%, 对比 A-DQN 提升 0.43%; 在 *F*1 方面, 对比 A-DQN 提升 0.72%; 在 *FPR* 指标上对比 RNN、A-DQN 分别有 71.6%和 29.8%的提升。综上, 对比结果表明, 本文的方法在网络入侵检测上的表现很好, 在入侵检测上对比现有的检测方法, 准确率相近或更优, 且误报率更低。

## 5　结语

传统的基于机器学习的网络入侵检测方法依赖大规模的样本数据, 在数据的预处理上一般采用降维或者聚类等一些数据处理的方法对大规模样本数据进行处理, 未来网络中流量数据的规模将会成倍增长, 有效的特征提取以减轻信息冗余和损失带来的影响极具挑战。对此本文提出了将深度神经网络和哈希学习结合起来, 建立一种基于深度监督离散哈希神经网络的网络入侵检测模型, 该模型学习网络流量数据的有效哈希表示, 减轻冗余特征对检测结果的影响, 大大减小了前期数据的预处理难度。模型通过神经网络学习得到定长的离散二进制哈希编码, 并可以用于网络入侵检测。在模型的训练上, 由于哈希码的离散特性, 本文模型通过交替最小化损失函数的方法, 可以更好地优化网络权重, 加快模

型收敛。实验结果表明, 本文所提模型具有良好的泛化能力, 学习到的二进制哈希编码可以很好保留同类网络数据相似性、反映不同类型流量之间的差异, 在入侵检测上的准确率和误报率上相近或优于最新的检测方法。

## 参考文献


[1] Jian S J, Lu Z G, Du D, et al. Overview of Network Intrusion Detection Technology[J]. *Journal of Cyber Security*, 2020, 5(4): 96-122.
(蹇诗婕, 卢志刚, 杜丹, 等. 网络入侵检测技术综述[J]. *信息安全学报*, 2020, 5(4): 96-122.)

[2] Xu J, Shelton C R. Intrusion Detection Using Continuous Time Bayesian Networks[J]. *Journal of Artificial Intelligence Research*, 2010, 39: 745-774.

[3] Ali Shah R, Qian Y T, Kumar D, et al. Network Intrusion Detection through Discriminative Feature Selection by Using Sparse Logistic Regression[J]. *Future Internet*, 2017, 9(4): 81.

[4] Khaoula R, Mohamed M. Improving Intrusion Detection Using PCA and K-Means Clustering Algorithm[C]. *2022 9th International Conference on Wireless Networks and Mobile Communications*, 2022: 1-5.

[5] LeCun Y, Bengio Y, Hinton G. Deep Learning[J]. *Nature*, 2015, 521(7553): 436-444.

[6] Pingale S V, Sutar S R. Remora Whale Optimization-Based Hybrid Deep Learning for Network Intrusion Detection Using CNN Features[J]. *Expert Systems with Applications*, 2022, 210: 118476.

[7] Hassan M M, Gumaei A, Alsanad A, et al. A Hybrid Deep Learning Model for Efficient Intrusion Detection in Big Data Environment[J]. *Information Sciences*, 2020, 513: 386-396.

[8] Lee J, Park K. GAN-Based Imbalanced Data Intrusion Detection System[J]. *Personal and Ubiquitous Computing*, 2021, 25(1): 121-128.

[9] Ferdowsi A, Saad W. Generative Adversarial Networks for Distributed Intrusion Detection in the Internet of Things[C]. *2019 IEEE Global Communications Conference*, 2019: 1-6.

[10] Gionis A, Indyk P, Motwani R, et al. Similarity search in high dimensions via hashing[C]. *International Conference on Very Large Data Bases*, 1999: 518-529.

[11] Wang J, Kumar S, Chang S F. Semi-Supervised Hashing for Large-Scale Search[C]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012: 2393-2406.

[12] Gong Y C, Lazebnik S, Gordo A, et al. Iterative Quantization: A Procrustean Approach to Learning Binary Codes for Large-Scale Image Retrieval[J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013, 35(12): 2916-2929.

[13] Norouzi M, Blei D M. Minimal loss hashing for compact binary codes[C]. *Proceedings of the 28th international conference on machine learning*, 2011: 353-360.

[14] Norouzi M, Fleet D J, Salakhutdinov R R. Hamming distance metric learning[J]. *Advances in neural information processing systems*, 2012: 1070-1078.

[15] Liu W, Wang J, Ji R R, et al. Supervised Hashing with Kernels[C]. *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012: 2074-2081.

[16] Shen F M, Shen C H, Liu W, et al. Supervised Discrete Hashing[C]. *2015 IEEE Conference on Computer Vision and Pattern Recognition*, 2015: 37-45.

[17] Xia R K, Pan Y, Lai H J, et al. Supervised Hashing for Image Retrieval *via* Image Representation Learning[J]. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2014, 28(1).

[18] Zhao F, Huang Y Z, Wang L, et al. Deep Semantic Ranking Based Hashing for Multi-Label Image Retrieval[C]. *2015 IEEE Conference on Computer Vision and Pattern Recognition*, 2015: 1556-1564.

[19] Li Q, Sun Z A, He R, et al. Deep Supervised Discrete Hashing[C]. *Conference on Neural Information Processing Systems*, 2017: 2482-2491.

[20] Sharafaldin I, Habibi Lashkari A, Ghorbani A A. Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization[C]. *The 4th International Conference on Information Systems Security and Privacy*, 2018: 108-116.

[21] Tavallaee M, Bagheri E, Lu W, et al. A Detailed Analysis of the KDD CUP 99 Data Set[C]. *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, 2009: 1-6.

[22] Moustafa N, Slay J. UNSW-NB15: A Comprehensive Data Set for Network Intrusion Detection Systems (UNSW-NB15 Network Data Set)[C]. *2015 Military Communications and Information Systems Conference*, 2015: 1-6.

[23] Gu J, Lu S. An Effective Intrusion Detection Approach Using SVM with Naïve Bayes Feature Embedding[J]. *Computers & Security*, 2021, 103: 102158.

[24] Zhong Y, Chen W Q, Wang Z L, et al. HELAD: A Novel Network Anomaly Detection Model Based on Heterogeneous Ensemble Learning[J]. *Computer Networks*, 2020, 169: 107049.

[25] Yin C L, Zhu Y F, Fei J L, et al. A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks[J]. *IEEE Access*, 2017, 5: 21954-21961.

[26] Sethi K, Madhav Y V, Kumar R, et al. Attention Based Multi-Agent Intrusion Detection Systems Using Reinforcement Learning[J]. *Journal of Information Security and Applications*, 2021, 61: 102923.


**薛胤** CCF 学生会员, 南京理工大学软件工程专业硕士生。本科毕业于南京理工大学软件工程专业。研究领域为网络安全态势感知、入侵检测等。Email: 1048277151@njust.edu.cn

**魏松杰** 博士, 副教授, CCF 高级会员。目前就职于南京理工大学计算机科学与工程学院。研究领域为网络体系结构, 网络与信息安全, 流量分析与建模等。Email: swei@njust.edu.cn