DOI:10.11918/202206023

基于特征缩减与自注意力机制的入侵检测方法

金志刚[1],周峻毅[1,2],武晓栋[1]

(1.天津大学 电气自动化与信息工程学院,天津 300072;2.天津大学 国际工程师学院,天津 300072)

摘 要:针对现代网络环境下流量数据特征高维化导致入侵检测时空复杂度较高,与传统入侵检测方法对流量数据之间相关性感知能力不足导致分类准确率不高的问题,以入侵检测高效性与准确性为目标,提出基于特征缩减和改进的自注意力机制的入侵检测方法。首先,针对数据高维化问题,使用具备非线性特征提取能力的自编码器进行特征抽取,降低数据冗余度的同时保证分类器的性能基本不变,确保入侵检测方法高效识别攻击行为。其次,针对传统入侵检测方法忽视流量数据相关性的问题,在入侵检测分类过程中引入自注意力机制学习一段时间内网络数据的相关性,并在原有的自注意力机制中引入因果卷积计算数据间的相关性分数,综合当前流量数据的局部位置信息和关注域内各流量数据之间的相关性综合分析当前流量行为并完成精准分类。在UNSW-NB15数据集上的实验表明,所提入侵检测方法在二分类任务中准确率达98.32%,在多分类任务中表现也同样优于传统入侵检测方法,在现代网络环境中具有较好的应用前景。

关键词:入侵检测;深度学习;自编码器;自注意力机制;因果卷积

中图分类号:TP393.08 **文献标志码**:A **文章编号**:0367-6234(2025)10-0112-11

An intrusion detection method based on feature reduction and self-attention mechanism

JIN Zhigang[1], ZHOU Junyi[1,2], WU Xiaodong[1]

(1. School of Electrical and Information Engineering, Tianjin University, Tianjin 300072, China;
2. International Engineering Institute, Tianjin University, Tianjin 300072, China)

Abstract: In view of the high spatial and temporal complexity of intrusion detection caused by high dimensionality of traffic data features in the modern network environment and low classification accuracy caused by the lack of sensitivity of traditional intrusion detection methods to the correlation between traffic data, an intrusion detection method based on feature reduction and improved self-attention mechanism is proposed to improve the efficiency and accuracy of intrusion detection. Firstly, aiming at the problem of high-dimensional data, an auto-encoder with nonlinear feature extraction capability is used to extract features, which reduces data redundancy and ensures classifier performance to be basically unchanged, so as to ensure that intrusion detection methods can effectively identify attacks. Secondly, aiming at the problem that traditional intrusion-detection methods ignore the correlation of traffic data, a self-attention mechanism is introduced in the intrusion detection classification process to learn the correlation of network data over a period of time. The causal convolution is introduced in original self-attention mechanism to calculate the correlation score between data, and integrate the local location information of current traffic data and the correlation between the traffic data in the concerned domain, which comprehensively analyzes current traffic behavior and complete accurate classification. Experimental results on UNSW-NB15 dataset show that the proposed intrusion detection method attains 98.32% accuracy on the binary classification tasks, and outperforms traditional methods on multi-classification tasks as well, indicating promising applicability in modern network environment.

Keywords: intrusion detection; deep learning; auto-encoder; self-attention mechanism; causal convolution

作为保障网络安全的重要技术之一,入侵检测相关技术一直是网络安全相关研究人员的关注焦点[1]。

入侵检测系统(intrusion detection system,IDS)可分为基于误用的IDS和基于异常的IDS。基于误用的IDS在面对现代网络环境时,因为其不能识别

收稿日期:2022-06-05;**录用日期**:2022-07-15;**网络首发日期**:2023-12-14
网络首发地址:https://link.cnki.net/urlid/23.1235.t.20231213.1922.002
基金项目:国家自然科学基金(52171337)
作者简介:金志刚(1972—),男,教授,博士生导师
通信作者:金志刚,zgjin@tju.edu.cn

未知攻击、规则库维护成本高等问题，不能很好适应当前对入侵检测准确性与实时性的需求[2]。因此，基于异常的 IDS 常被研究人员作为目前研究的主要方向。现代网络环境呈智能化、体系化态势，在设计基于异常的 IDS 时需要考虑以下几个方面。首先，现代网络产生的数据维度更高，数据特征更加复杂交迭[3]。其次，分析现代网络攻击行为时需要综合考虑多个报文才能分辨出攻击行为与其他流量行为的不同点。例如，针对 DDoS 攻击期间的单个数据包，如果仅考虑当前流量数据的数据特征而忽略一段时间内流量数据之间的相关性对该数据包进行评估，则该数据包可能被视为正常的客户端打开连接[4]；又如，Probing 利用 TCP 或 UDP 检索端口，在发现合适端口后发动 DoS 攻击使设备瘫痪；再如，DoS 攻击的频繁序列模式也存在内生联系[5]。综上，现代网络攻击手段内部以及攻击手段之间存在的丰富的相关性并没有在传统 IDS 中得到合理利用。

在如何保证入侵检测方法的高效与准确的研究领域中，相关人员已经展开了研究。Kasongo 等[6]针对现代无线流量数据之间存在时序相关性的问题，提出利用深度长短期记忆网络（deep long short term memory networks，DLSTM）来学习时序特征，同时应用基于信息增益的特征选择方法对数据降维。Kan 等[7]提出了一种基于自适应粒子群优化卷积神经网络的物联网入侵检测方法，该方法将一维卷积神经网络（convolutional neural network，CNN）的结构参数作为自适应粒子群优化的位置参数，学习流量数据空间信息的同时加速了整个模型的超参数优化速度。Fu 等[8]提出了一种融合注意力机制和双向长短期记忆网络的入侵检测模型。该模型首先通过 CNN 提取流量数据的序列特征，然后，利用通道注意力机制重新分配各通道的权重，最后，使用双向长短期记忆网络提取流量数据的时间特征。Andresini 等[9]提出了一种基于神经注意力的多输出可解释入侵检测模型。该模型通过注意力机制对网络流量数据进行准确的多类别分类，并利用注意力热图进行可解释性相关研究。刘月峰等[10]融合 CNN 和双向 LSTM 神经网络，通过 CNN 捕捉流量数据之间的平行局部特征，双向 LSTM 捕捉流量数据之间的长距离依赖关系。最后，通过注意力机制计算各属性特征的权重。宋勇等[11]通过逐层贪婪训练策略，改进稀疏自编码器的训练方式，在确保对流量数据特征高效提取的同时，利用支持向量机对降维后的数据进行分类，实验证明该方法自适性更强，分类精度也较高。郭志民等[12]利用自注意力机制提取流量数据的时间相关性，提出了基于 Transformer 网络的入侵检测方法，该方法在多个数据集上均取得了较好的检测结果。

上述方法虽取得了一定的效果，但存在以下几方面缺陷。首先，部分研究重点关注如何对数据特征进行提取和降维，忽略了降维后进一步捕捉流量数据相关性的重要性；其次，在捕捉流量数据相关性的方法上，应综合位置信息和时序信息对当前流量进行判断，单独使用 CNN 或循环神经网络（recurrent neural networks，RNN）难以捕捉数据的相关性信息；再次，由于网络流量数据往往数据量较大，如果使用循环神经网络及其变体神经网络（如 LSTM，GRU 等）难以避免梯度弥散、关键信息丢失等问题；最后，注意力机制的应用欠缺对流量数据时空信息的分析，不能很好地捕捉流量数据相关性。贴合现代网络安全环境的入侵检测方法应首先考虑对数据特征进行提取和降维，确保入侵检测方法便于应用和实时交互，其次，应寻找适当机制综合分析网络流量数据的时间信息与空间信息，以学习现代网络流量数据之间丰富的相关性，从而达到精准分类的目的。

综上，针对现代网络环境所面临的网络安全问题，提出了基于自编码器和改进的自注意力机制（AE-SATT）的入侵检测方法。首先，通过自编码器（auto-encoder，AE）学习数据的非线性特征，降低数据冗余度的同时保留原始流量数据的关键特征信息，使分类器可以高效利用自编码器处理后的数据进行训练；其次，考虑网络流量的时空性质，利用自注意力机制对流量数据之间的相关性进行建模，同时将因果卷积引入自注意力机制，解决原有自注意力机制考虑当前流量数据局部位置信息能力不足的问题，使模型综合分配关注度权重从而学习网络流量数据之间的相关性，并对当前流量数据进行精准分类；最后，对该入侵检测方法在现代网络中的部署方式进行模拟仿真，旨在利用现代网络环境中的关键技术以降低入侵检测系统的计算压力，验证了此种入侵检测方法在现代网络环境中的应用价值。

1　特征缩减与分类器

1.1　特征缩减模块

特征缩减旨在利用机器学习的相关方法减少原始数据集中冗余的特征，达到降低模型的计算量、提高模型性能的目的[13]。目前，应用于入侵检测系统的特征降维方法有很多，如主成分分析、支持向量机等。这些方法在面对特征呈线性相关的数据时具有良好的特征缩减效果，但在面对数据吞吐量大且数据特征复杂的应用背景时，无法有效地降低数据复

杂度[14]。

特征缩减不仅可以有效减少数据集中的冗余特征,还可以提高用于输入分类器的数据的鲁棒性,辅助分类器完成高效准确的分类任务[15]。自编码器作为一种无监督的算法在面对数据特征高维且复杂的情况时可以很好地拟合数据非线性特征,在压缩原始流量数据特征信息的同时完成将数据映射到低维空间的任务。同时,部分工作[1,11,16]利用自编码器及其变体对流量数据进行特征降维或缩减后,再对流量数据进行分类,均取得了较好的效果。

自编码器有编码和解码两个部分,设原始空间数据为 $\mathbf{R}^{m\times n}$,其中,m 为原始空间数据的样本数,n 为原始空间中数据样本的维度数。则式(1)为编码过程,式(2)为解码过程:

$$h = S(f(x)) = S(\boldsymbol{W}x + b_{\mathrm{h}}) \tag{1}$$

$$y = S(g(h)) = S(\boldsymbol{W}^{\mathrm{T}}h + b_{\mathrm{y}}) \tag{2}$$

式中:b_{h},b_{y} 分别为隐藏层和输出层的偏置项,$\boldsymbol{W}$,$\boldsymbol{W}^{\mathrm{T}}$ 分别为隐藏层和输出层的权重矩阵。自编码器通过构建以最小化均方误差为准则的目标函数,使输出层输出的结果尽可能与输入的数据样本相等。训练好的自编码器的中间隐藏层输出的低维度数据可以很好地表示出原有数据的关键信息,实现特征缩减。

1.2 改进的自注意力分类模块

自注意力机制最初由 Vaswani 等[17]提出,旨在解决自然语言处理领域中机器翻译等任务。该机制完全摒弃了传统的 RNN,通过完全依赖注意力机制对时序数据之间的相关性进行建模。这种机制不仅克服了传统 RNN 的梯度弥散问题,同时可以作到并行化输入,提高计算效率,这对于需要实时并准确分析现代网络攻击的入侵检测方法而言是一种很好的机制。原始的自注意力分类模块如图 1 所示。

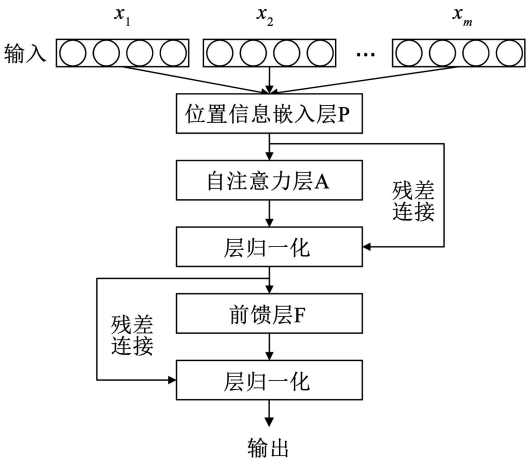


图 1 原始的自注意力分类模块

Fig. 1 Original self-attention classification module

设原始空间数据为 $\mathbf{R}^{m\times n}$,其中,m 为原始空间数据的样本数,n 为原始空间中数据样本的维度数。$x_i \in \mathbf{R}^m$,$i = 1,2,\cdots,m$。则位置信息嵌入层 P 的位置编码方式如式(3)所示,编码的具体过程与结果如图 2 所示。

$$\begin{aligned} f_{\mathrm{PE}(\mathrm{pos},2j)} &= \sin(\mathrm{pos}/10\ 000^{2j/n}) \\ f_{\mathrm{PE}(\mathrm{pos},2j+1)} &= \cos(\mathrm{pos}/10\ 000^{2j/n}) \end{aligned} \tag{3}$$

式中:pos 为当前样本在所有输入样本中的位置,j 为当前样本的特征位置。自注意力机制完全摒弃了 RNN,导致其无法像 RNN 一样学习网络流量数据的相对位置关系,需要在原始输入的基础上加上式(3)计算出的结果得到位置信息嵌入层 P 的输出,使流量数据样本融合了其特有的位置信息并保证模型学习到相对位置信息。

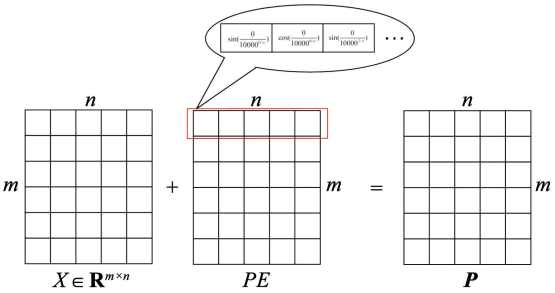


图 2 位置信息嵌入层的编码过程与结果

Fig. 2 Encoding process and results of location information embedding layer

自注意力层的自注意力分配函数如式(4)、(5)所示,数据在自注意力层中变化过程的具体示例如图 3 所示。

$$\begin{cases} \boldsymbol{Q} = \boldsymbol{P}\boldsymbol{W}^{Q} \\ \boldsymbol{K} = \boldsymbol{P}\boldsymbol{W}^{K} \\ \boldsymbol{V} = \boldsymbol{P}\boldsymbol{W}^{V} \end{cases} \tag{4}$$

$$f_{\mathrm{Attention}}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) = \mathrm{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^{\mathrm{T}}}{\sqrt{d_{\mathrm{k}}}}\right)\boldsymbol{V} \tag{5}$$

式中:$\boldsymbol{P}$ 为位置信息嵌入层输出;$\boldsymbol{W}^{Q}$,$\boldsymbol{W}^{K}$,$\boldsymbol{W}^{V}$ 分别为计算查询矩阵、键矩阵和值矩阵的权重矩阵,3 个矩阵的尺寸相同;d_{k} 为缩放因子。首先,通过权重矩阵将输入转换为查询矩阵、键矩阵和值矩阵;然后,通过查询矩阵点乘键矩阵转置的方式计算当前自注意力关注域内每个输入向量与各向量之间的相似度,并使用缩放因子保持训练时的梯度稳定;最后,使用 softmax 函数将计算结果转换为概率并与值矩阵相乘,得到当前向量对同一自注意力关注域内每个输入向量的注意力大小。为了自注意力层更好地捕捉流量数据之间相关性,引入多头自注意力机制代替原有的单一自注意力函数。其计算过程如下:

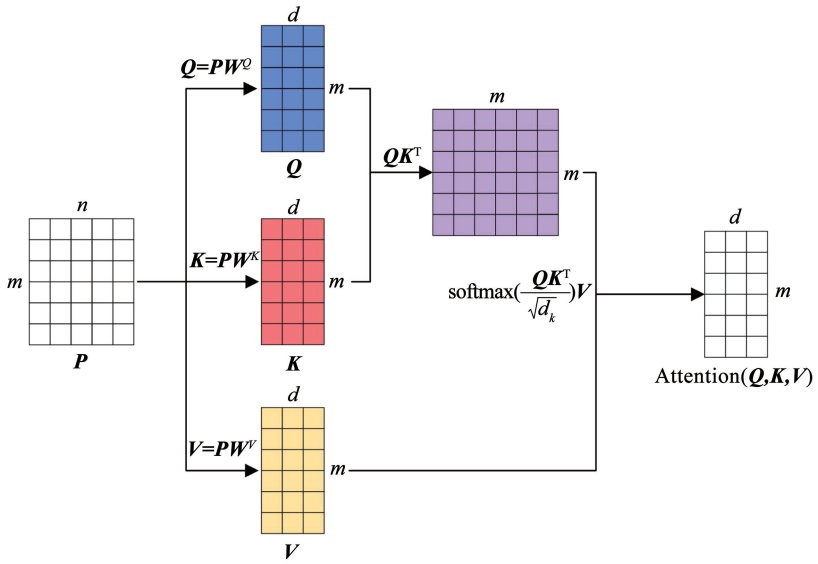


图 3　自注意力层数据变化过程示例

Fig. 3　Example of data change process from self-attention layer

$$f_{\text{MultiHead}}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) = \text{Concat}(f_{\text{head}_1},\cdots,f_{\text{head}_h})\,\boldsymbol{W}^O$$

$$f_{\text{head}_i} = f_{\text{Attention}}(\boldsymbol{P}\boldsymbol{W}_i^Q,\boldsymbol{P}\boldsymbol{W}_i^K,\boldsymbol{P}\boldsymbol{W}_i^V) \tag{6}$$

式中：$\boldsymbol{W}_i^Q,\boldsymbol{W}_i^K,\boldsymbol{W}_i^V$ 为第 i 头的权重矩阵，Concat(·)为拼接函数，$\boldsymbol{W}^O$ 为用于计算自注意力层输出的权重矩阵。

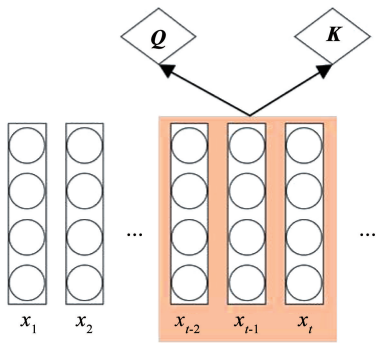
层归一化操作将自注意力层 A 的输出与位置信息嵌入层 P 的输出求和以后再进行层归一化。求和的目的是利用残差结构，以保证模型的训练效果不会因为神经网络结构的复杂而出现退化现象。层归一化操作在同一批次输入数据的每一个特征维度上进行归一化操作，以减小数据偏差，避免出现梯度消失或梯度爆炸的问题。

前馈层 F 的计算过程如下：

$$F(L) = \text{ReLU}(L\boldsymbol{W}_1 + b_1)\,\boldsymbol{W}_2 + b_2 \tag{7}$$

式中：L 为层归一化输出，$\boldsymbol{W}_1,\boldsymbol{W}_2$ 为权重矩阵，b_1,b_2 为偏置项。前馈层 F 的引入可以提高整个模型的分类性能，弥补自注意力机制对样本内部特征的学习能力。最后，再次通过残差连接与层归一化，得到整个自注意力分类模块的输出。

对于入侵检测任务而言，流量数据是否被归类为异常在一定程度上与该流量数据附近几个流量数据有关。原始的自注意力模块在计算注意力分配函数时，使用的都是线性点积，这就会导致模型对于每一个当前输入样本，在分配注意力时会丢失当前输入的局部信息。当同一自注意力关注域内的流量数据中有因网络波动的异常数据时，原始的自注意力机制就有可能将异常数据也视为一种恶意攻击的前兆，导致对当前网络状态的误判。因此，为了提高模型对输入数据局部信息关注程度，使用因果卷积代替线性点积的方式来计算查询矩阵和键矩阵，以保证在计算每个输入向量与各向量之间的相似度时引入局部位置信息，而计算值矩阵时仍采用线性点积。因果卷积可以保证在计算过程中模型看不到未来输入数据的信息，计算示例如图 4 所示。



注：步长为1，卷积核大小为3。

图 4　因果卷积计算 $\boldsymbol{Q},\boldsymbol{K}$

Fig. 4　Causal convolution calculation $\boldsymbol{Q},\boldsymbol{K}$

对于 t 时刻而言，使用步长为 1，卷积核大小为 3 的一维卷积来计算查询矩阵 $\boldsymbol{Q}$ 和键矩阵 $\boldsymbol{K}$，值矩阵 $\boldsymbol{V}$ 的计算仍和式(4)的计算过程相同。这样模型在进行自注意力分配时就可以综合考虑每个输入数据的局部位置信息，从而达到提高模型的整体分类性能的目的。将上述方法替换掉原始的自注意力分类模型中计算查询矩阵和键矩阵的方法，保留其他模块不变，就完成了改进的自注意力分类模块的构建。

2　AE-SATT 的构建

2.1　AE-SATT 入侵检测方法的设计

AE-SATT 的完整结构见图 5。

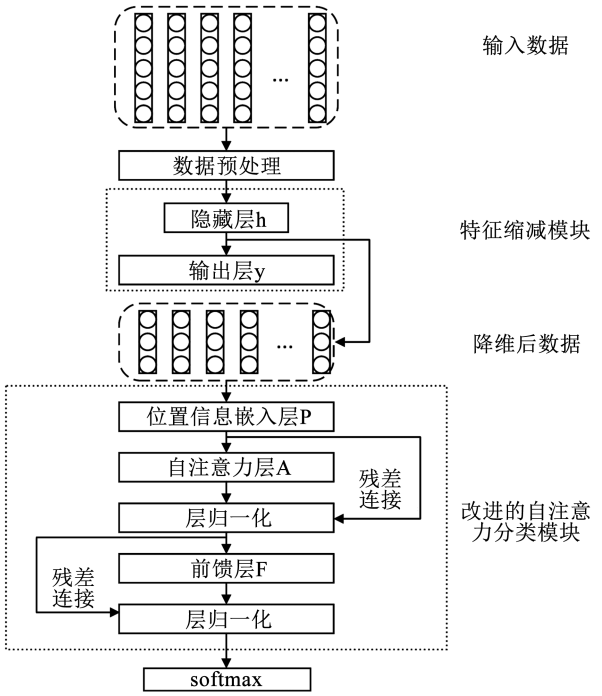


图 5 AE-SATT 完整结构

Fig. 5 AE-SATT complete structure

设计的具体步骤如下：

1)对原始的入侵检测数据集进行数据预处理的操作。首先对数据集中的类似于传输层协议(如 TCP,UDP)等字符类特征属性进行数值化操作,具体方法为利用独热编码的方式,对该字符类特征属性进行编码,用编码结果代替原来的字符类属性。其次,对所有数值类特征属性进行归一化操作以防止模型在训练过程中出现收敛过慢的问题。目前的主流方法是最大 - 最小归一化方法,具体如下：

$$x^* = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{8}$$

式中：x^* 为经过归一化之后的结果,$x_{\max}$ 代表该特征属性中的最大数值,$x_{\min}$ 代表该特征属性中的最小数值。通过上述计算,特征属性的值会被映射到[0,1]。

2)将经过预处理后的数据输入到特征缩减模块中的自编码器进行训练,待模型训练好以后将模型保存下来,用于后续分类模块的输入。

3)调用训练好的特征缩减模块,将模型隐藏层 h 的输出作为改进的自注意力分类模块的输入。改进的自注意力分类模块的具体设计细节同 1.2 节。将分类模块的输出利用 softmax 函数进行概率转换,并使用交叉熵损失函数对分类模块进行训练。

4)在分类模块中,在自注意力层 A 和最后的层归一化的后面使用 Dropout 方法使一定比例的神经元失活,目的是防止分类模块的过拟合现象,提高模型的泛化能力。

5)在以上步骤的基础上,对模型中的重要超参数进行调整,通过实验找到最优的超参数以保证模型的分类性能。

2.2 AE-SATT 算法流程

1)利用步骤 1 中的方法对输入数据进行预处理

2)For i in epoch_1

3)初始化自编码器所有权重矩阵 $\boldsymbol{W}$ 和偏置项 b

4)将预处理后的数据输入到自编码器中

5)训练模型

6)return $\boldsymbol{W}, b$

7)End For

8)保存自编码器模型

9)For i in epoch_2

10)初始化分类模块所有权重矩阵 $\boldsymbol{W}$ 和偏置项 b

11)调用保存好的自编码器文件

12)将自编码器隐藏层输出作为分类模块输入

13)训练模型

14)return $\boldsymbol{W}, b$

15)End For

16)End

2.3 面向现代网络环境的入侵检测方法的模拟部署

模拟现代网络环境中的关键技术,将入侵检测方法进行合理部署可以有效降低终端设备的计算压力,提高入侵检测方法的可行性。现代网络环境中所需要的关键技术之一是基于云计算的技术[18]。雾计算层是介于云计算层和网络终端设备之间的计算层,且雾计算可以解决云计算中的传输延迟、移动性差等问题[19]。通过以上两个关键技术对入侵检测方法进行模拟部署的细节如图 6 所示。

入侵检测数据处理模块被部署在雾计算层,该模块的主要任务是完成数据预处理和特征缩减。其主要原因在于:雾计算层相较于云计算层更靠近网络终端设备,可以在第一时间将收集到的数据进行处理并准备发送给云计算层;雾计算层的能力可以很好地完成数据预处理和特征缩减这些对计算能力要求不高的工作。入侵检测攻击识别模块部署在云计算层,利用云计算层的强大计算能力训练模型并对收集到的流量数据进行实时分类。通过以上模拟部署可以分摊终端设备上的计算压力,并保证入侵检测方法的性能。

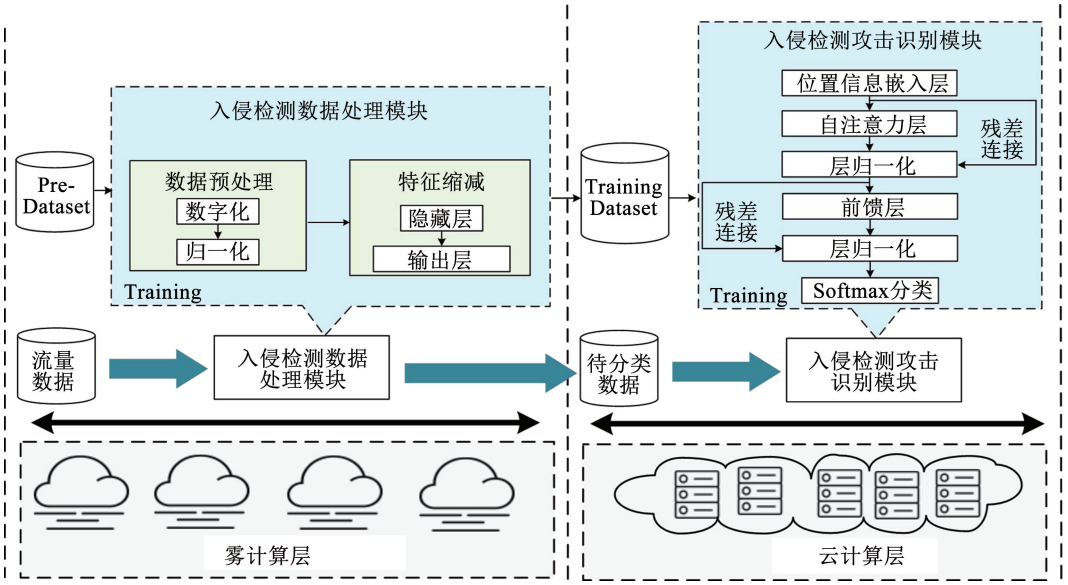


图 6　入侵检测方法的模拟部署

Fig. 6　Simulation deployment of intrusion detection methods

3　实　验

实验的硬件环境为 CPU：AMD Ryzen 7 5900H 3.2 GHz，RAM：16 GB(3 200 MHz)。实验的软件环境为 Windows11 操作系统、Python 3.6.5、Tensorflow2.0.0、Keras2.3.1。特别说明，为了方便后续实验的对比，本文实验均不采用 GPU 进行运算，而是采用 CPU 进行运算。

3.1　实验数据集

UNSW-NB15 是由 Moustafa 等[20]于澳大利亚网络安全中心收集并整理的入侵检测数据集，相较于 KDD CUP 99 数据集，可以更好地模拟现代网络攻击，更适合本文的研究背景。UNSW-NB15 数据集中包含 9 种不同类型的攻击，每一条实例数据有 49 个特征维度。其中，最后两个特征维度是攻击类型和标签(0 代表正常，1 代表攻击)。原始的 UNSW-NB15 数据集有 2 540 044 条实例数据，其中包含大量的正常实例，这种数据的不平衡会导致模型很容易过拟合[21]。因此，本文采用的是 UNSW-NB15-training 和 UNSW-NB15-testing 数据集。这两个数据集是原 UNSW-NB15 的子集，该数据集是从原 UNSW-NB15 数据集中截取而来的，保留大部分攻击示例的同时删掉部分正常示例，并保证训练集和测试集具有相似的分布[22]。数据集中舍弃掉‘scrip’、‘sport’、‘dstip’、‘stime’和‘ltime’5 个特征，加入了‘id’这一特征。因此，本文使用的数据集中的实例数据有 45 个维度。UNSW-NB15-training 和 UNSW-NB15-testing 数据集的各类型实例分布如表 1 所示。

表 1　UNSW-NB15 训练集和测试集数据分布

Tab. 1　UNSW-NB15 training set and test set data distribution

类别	训练集实例数	测试集实例数
Normal	56 000	37 000
Analysis	2 000	677
Backdoor	1 746	583
Dos	12 264	4 089
Exploits	33 393	11 132
Fuzzers	18 184	6 062
Generic	40 000	18 871
Reconnaissance	10 491	3 496
Shellcode	1 133	378
Worm	130	44

3.2　数据预处理

UNSW-NB15-training 和 UNSW-NB15-testing 数据集中共包含 45 维特征属性，其中，‘state’、‘service’和‘proto’这 3 个特征属性为字符类特征属性。因此，要用独热编码对这 3 个特征进行数值化，具体的数值化结果如图 7 所示。

数值化后的数据，还要进一步将数据集中的‘id’特征作为流量数据的行索引，再将‘attack_cat’和‘label’特征剔除，最后，将‘label’特征另进行存储用于后续监督训练。接下来对数值化的特征进行归一化处理，至此就完成了数据预处理的操作。

3.3　评价指标

本文实验部分主要参考的评价指标有准确率(accuracy，A)、精确率(precision，R_P)、召回率(recall，R_E)、F_1 值、马修斯相关系数(matthews correlation coefficient，C_{MC})。具体计算过程如下：

$$A=\frac{T_P+T_N}{T_P+T_N+F_P+F_N} \tag{9}$$

$$R_{\mathrm{P}}=\frac{T_{\mathrm{P}}}{T_{\mathrm{P}}+F_{\mathrm{P}}} \tag{10}$$

$$R_{\mathrm{E}}=\frac{T_{\mathrm{P}}}{T_{\mathrm{P}}+F_{\mathrm{N}}} \tag{11}$$

$$F_1=\frac{2T_{\mathrm{P}}}{2T_{\mathrm{P}}+F_{\mathrm{P}}+F_{\mathrm{N}}} \tag{12}$$

$$C_{\mathrm{MC}}=\frac{T_{\mathrm{P}}\times T_{\mathrm{N}}-F_{\mathrm{P}}\times F_{\mathrm{N}}}{\sqrt{(T_{\mathrm{P}}+F_{\mathrm{P}})(T_{\mathrm{P}}+F_{\mathrm{N}})(T_{\mathrm{N}}+F_{\mathrm{P}})(T_{\mathrm{N}}+F_{\mathrm{N}})}} \tag{13}$$

式中的具体参数含义见表 2。

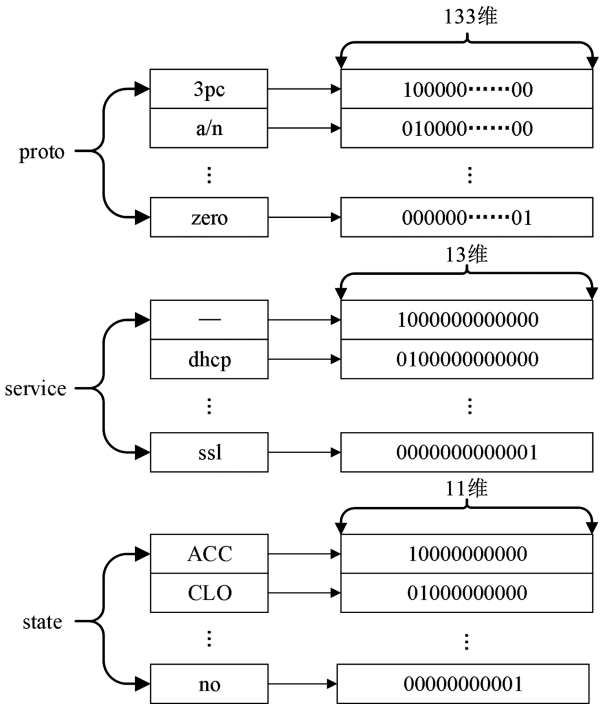


图 7 字符类特征数值化结果示例

Fig. 7 Example of numerical results for character class features method

表 2 评价指标中各参数含义

Tab. 2 Meanings of parameters in evaluation metrics

参数	具体含义
T_{P}	正实例被判为正实例
T_{N}	负实例被判为负实例
F_{P}	负实例被判为正实例
F_{N}	正实例被判为负实例

3.4 调参实验

调参实验可以确定模型训练的最优超参数,提高模型的整体性能。调参实验选择一次训练选取的样本数(Batch-size,Bz),学习率(Learning-rate,Lr)和自注意力关注域(ATT-span)进行调整,具体实验结果见表 3。

表 3 调参实验数据

Tab. 3 Parameter adjustment experimental data

Bz	Lr	ATT-span	训练集准确率/%
32	0.001	8	96.62
32	0.001	16	96.73
32	0.001	32	96.59
32	0.001	64	96.65
32	0.1	16	68.07
64	0.1	16	68.07
128	0.1	16	92.28
32	0.01	16	93.48
64	0.01	16	95.06
128	0.01	16	96.29
64	0.001	16	96.22
128	0.001	16	95.95

实验首先在固定 Bz 和 Lr 的情况下改变 ATT-span 来观察训练集上的准确率变化。根据 1.2 节的介绍,ATT-span 的改变对准确率的影响应该较大,但是在实验过程中发现 ATT-span 的变化对准确率的影响相对较小。通过观察数据集得知,UNSW-NB15-training 和 UNSW-NB15-testing 数据集是从原 UNSW-NB15 数据集当中截取而来的,正常实例和攻击实例并不完全按真实情况分布。数据集中虽然保留了原 UNSW-NB15 数据集中各攻击类别内部的时序,但删去了大量正常实例和整体数据集各类别在数据集内的时序变化,导致 ATT-span 的变化对模型学习流量数据之间的相关性影响不大。

在得出上述结论后,设置 ATT-span 为 16,对 Bz 和 Lr 进行调整,最后,确定超参数 Bz 为 32,Lr 为 0.001,ATT-span 为 16。模型其他超参数根据经验以及实验确定,详见表 4。

表 4 其他超参数设置

Tab. 4 Other hyper parameter settings

超参数名称	训练轮数	Dropout	特征缩减后数据维度	自注意力层头数	卷积核大小	卷积步长
超参数数值	30	0.2	16	4	3	1

3.5 二分类实验

3.5.1 模型性能指标分析

本实验将传统的入侵检测方法以及部分基于时序的入侵检测方法与本文提出的入侵检测方法进行对比来分析模型捕捉数据相关性,从而实现精准分类的程度。由于部分比对模型使用的数据集与本文实验采用的数据集不同,该环节实验中用于比对的其他模型数据均为复现的结果以方便结论分析。具体实验结果见表 5。

表 5　AE-SATT 模型与其他模型性能比较

Tab. 5　Performance comparison of AE-SATT model with other models　%

模型	A	R_P	R_E	F_1	C_{MC}
SVM	81.60	75.10	99.63	85.64	66.30
RF	86.91	81.49	98.62	89.24	75.08
LSTM[23]	94.24	96.07	87.13	91.38	87.14
RNN[24]	89.41	96.38	78.35	86.43	80.90
DAL[25]	92.65	94.97	88.33	91.53	85.23
AE-SATT	98.32	97.55	92.15	94.77	92.04

分析表 5 中的数据可知，本文提出的模型在准确率、精确率上远优于其他模型。而在召回率上，本文提出的模型低于 SVM 和 RF。召回率的偏低说明 AE-SATT 模型被正确判定的正例占总正例的比重较低。进一步分析，SVM 和 RF 模型主要通过数据集中每条实例自身的特征属性来对该条实例进行分类，并不像 AE-SATT 模型去关注同一自注意力关注域内的所有数据实例。类似于 Generic 攻击，在数据集中往往都是多条实例连续出现的，AE-SATT 通过训练学习数据之间的相关性，可能错误地将多条连续出现的正常实例判定为攻击实例，导致召回率相对偏低。但是从 F_1 值和 C_{MC} 上看，AE-SATT 模型的综合性能是优于其他模型的。并且，AE-SATT 模型的各项评价指标均优于 LSTM、RNN 以及 DAL，这说明传统的循环神经网络及其变体所学习到的数据时序性相较于 AE-SATT 学习到的数据相关性对分类影响小。综合一段时间的流量数据对当前数据实例进行分类在现代网络环境中更加合理。

3.5.2　消融实验

消融实验主要分为两部分：保持特征缩减模块不变，比对改进的自注意力分类模块（使用因果卷积计算 $\boldsymbol{Q}$，$\boldsymbol{K}$）和原始的自注意力分类模块（使用点击计算 $\boldsymbol{Q}$，$\boldsymbol{K}$）；保持改进的自注意力分类模块不变，比对使用特征缩减的 AE-SATT 模型和去掉特征缩减的 SATT 模型。

首先保持特征缩减模块不变，比对改进的自注意力分类模块和原始的自注意力分类模块，各项超参数设置同 3.2 节，具体实验结果见表 6。

表 6　消融实验 1

Tab. 6　Ablation study 1　%

模型	A	R_P	R_E	F_1	C_{MC}
改进	98.32	97.55	92.15	94.77	92.04
原始	97.93	97.66	91.69	94.58	91.80

通过消融实验 1 可以得知改进的自注意力分类模块相较于原始的自注意力分类模块在多项评价指标上有小幅度提升。

进一步，为了比对原始的自注意力分类模块与改进的自注意力分类模块对流量数据之间相关性的捕捉能力。在模型训练完毕后，选取自注意力层中第一头的权重，对测试集中某一自注意力关注域内的 16 条流量数据中的一条异常数据与域内的其他数据进行相关性分析，结果如图 8、9 所示。

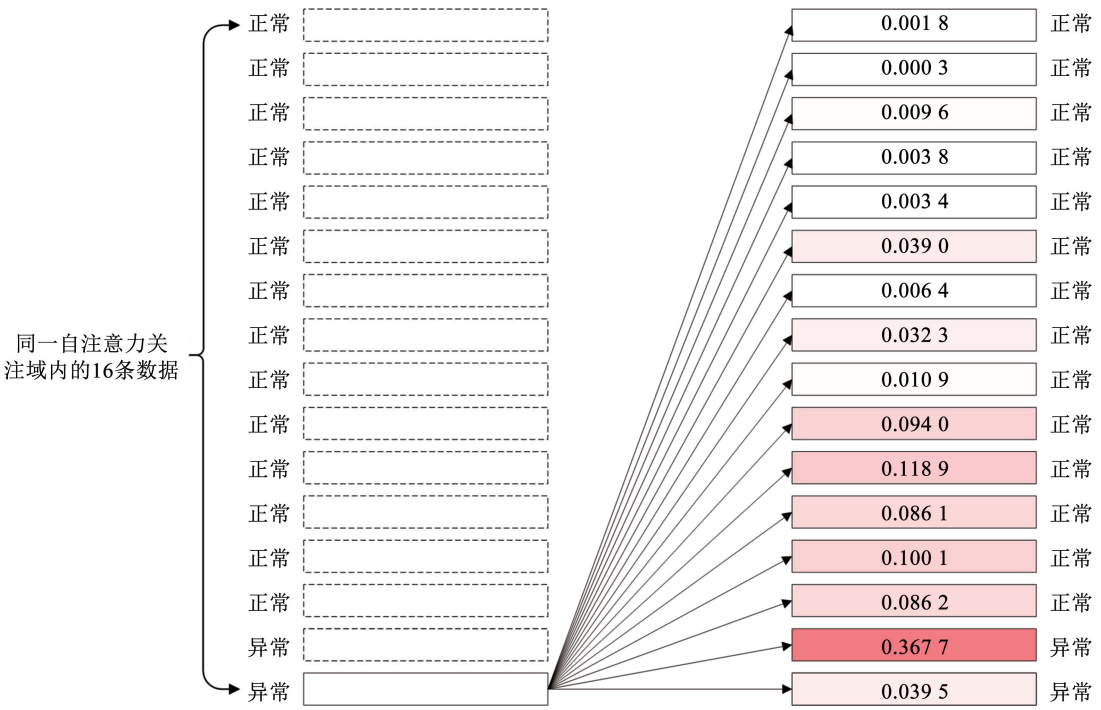


图 8　原始自注意力模块对流量数据相关性的分析

Fig. 8　Original self-attentive module analysis of traffic data correlation

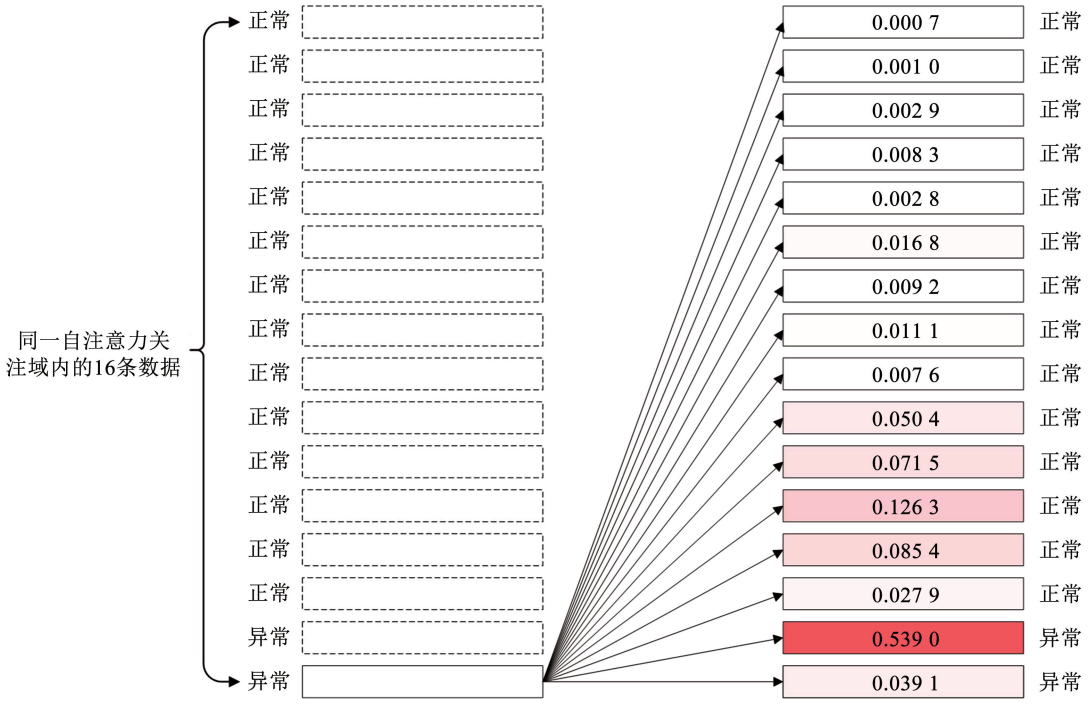


图 9 改进自注意模块对流量数据相关性的分析

Fig. 9 Improved self-attentive module analysis of traffic data correlation

通过对比图 8 和图 9 可知,融合了因果卷积以后,在分析最后一条数据与其他数据相关性时,改进的自注意力模块相较于之前更加关注倒数第二条数据,即模型通过训练,学习到了局部位置信息,判断倒数第二条流量对最后一条流量的分类影响更大。综合上述实验结果可知,通过引入因果卷积的操作,模型可以更加关注流量数据的局部位置信息,综合自注意力关注域内的数据信息和当前数据的局部信息对当前数据进行精准分类。而各项评价指标提升效果不显著的原因大概率仍是因为 UNSW-NB15-training 和 UNSW-NB15-testing 数据集数据的分布问题,局部信息的重要性不是非常显著。

为比对特征缩减模块是否可以在保证分类器性能的前提下提高入侵检测分类器的实时性能,降低分类器的计算开销,开展消融实验。本文入侵检测方法分类器的输入是直接调用已经训练好的特征缩减模块获取降维后的数据,从更加直观的角度考虑,选用准确率、模型总参数、训练时间和测试时间 4 个指标来比对特征缩减对分类器的影响并分析实验结果。保持改进的自注意力分类模块不变,比对使用特征缩减的 AE-SATT 模型和去掉特征缩减的 SATT 模型。具体实验结果见表 7。

分析消融实验 2 的数据,模型在去掉特征缩减模块后准确率小幅度下降,并且模型总参数、训练时间和测试时间都显著高于 AE-SATT 模型。特征缩减模块很好地提取出了原数据集中的重要特征,帮助模型更好地学习流量数据之间的相关性以进行分类;模型总参数与输入数据维度和网络复杂度呈正相关,带有特征缩减模块的模型总参数的大幅减少说明特征缩减可以降低分类器的计算量,训练和测试速度明显提高,说明特征缩减可以帮助分类器高效实时地完成任务。

表 7 消融实验 2

Tab. 7 Ablation study 2

模型	准确率/%	模型总参数	训练时间/min	测试时间/μs
AE-SATT	98.32	4 416	0:42.87	37
SATT	97.86	547 476	2:53.40	186

3.6 多分类实验

在多分类问题中,首先生成多分类的混淆矩阵,并根据混淆矩阵计算各类别的具体指标和模型在进行多分类任务时的模型总性能。混淆矩阵的结果见表 8,根据混淆矩阵计算而来的各类别性能指标见表 9。

通过分析多分类结果,可以看到模型对于识别 Normal、Exploits、Generic 类的表现较好。这是因为在 UNSW-NB15-training 和 UNSW-NB15-testing 中。这些类别都会在一段时间内连续出现,模型在学习流量数据相关性时可以很好地学习到这些类别的相关性规律,并在测试集上给出准确的分类结果,而对于类似 Worm 类、Shellcode 类等在训练集和测试集当中的数量都较少且这些类别的攻击不会在一段时间内连续出现,及训练集和测试集是通过截取原始 UNSW-NB15 数据集得到的,导致模型无法很好地学习到 Worm 类、Shellcode 类等攻击出现时与其他流量数据的相关特性,从而无法准确判断出其类别。

表 8　多分类混淆矩阵

Tab. 8　Multi-classification confusion matrix

实际类别	预测类别									
	Normal	Analysis	Backdoor	Dos	Expliots	Fuzzers	Generic	Reconnaissance	Shellcode	Worm
Normal	35 635	1	0	3	522	694	0	144	1	0
Analysis	9	0	0	0	666	2	0	0	0	0
Backdoor	2	0	0	0	546	24	0	11	0	0
Dos	359	0	0	217	3 155	257	5	96	0	0
Expliots	276	19	0	0	9 589	1 126	0	122	0	0
Fuzzers	575	0	0	0	1 487	3 808	2	190	0	0
Generic	15	0	0	0	442	228	18 155	31	0	0
Reconnaissance	114	1	0	1	770	920	5	1 685	0	0
Shellcode	9	0	0	0	23	120	0	202	24	0
Worm	0	0	0	0	34	9	0	1	0	0

表 9　各类别评价指标

Tab. 9　Evaluation indicators of each category

类别	R_P	R_E
Normal	96.33	96.31
Analysis	0	0
Backdoor	0	0
Dos	98.19	5.31
Expliots	55.64	86.14
Fuzzers	52.98	62.82
Generic	99.93	96.21
Reconnaissance	67.89	48.20
Shellcode	96.00	6.35
Worm	0	0

将多分类模型的性能与其他入侵检测方法进行比较，结果如图 10 所示。

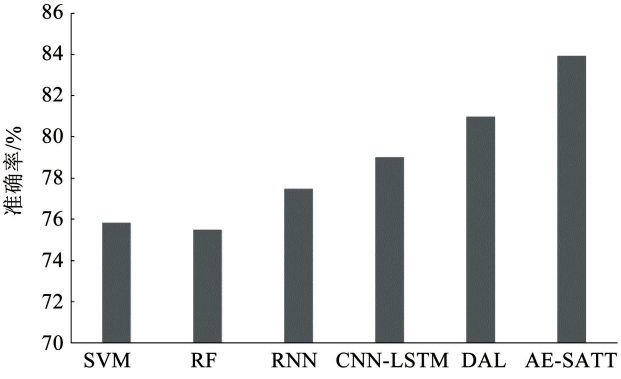


图 10　多分类模型准确率

Fig. 10　Accuracy of multi-classification model

分析实验数据可知，AE-SATT 模型在多分类任务上的性能相对于二分类有所下滑，但与 SVM，RF，RNN[24]，CNN-LSTM[26]，DAL[25] 模型的对比可知，AE-SATT 模型在分类准确率上还是优于其他模型的。这说明在更复杂的多分类任务中，本文的模型能够通过学习到的流量数据之间的相关性来对当前数据的具体类别进行相对准确的判断。综上，综合分析一段时间内流量数据之间的相关性可以更好地完成对流量数据的分类。

3.7　模拟入侵检测方法部署实验

为了模拟本文提出的面向现代网络环境下的入侵检测方法，利用 Tensorflow 和 Keras 下的工具来模拟云、雾计算层的数据传递。首先，在一个 python 文件中编写数据预处理和自编码器的功能。在主函数中导入自编码器模型并训练，将训练好的自编码器模型用 Keras 框架下的回调函数保存为. hdf5 文件。在训练分类模块时直接加载. hdf5 文件，并将隐藏层的输出数据作为分类模块的输入，通过以上过程模拟云、雾计算层的数据传递。实验主要比对这种部署方式与模拟不使用云、雾计算技术，直接将两个模型一起训练时部署方式的系统参数详见表 10。

表 10　两种部署方式的性能对比

Tab. 10　Performance comparison of two deployment methods

构建方式	CPU 平均利用率/%
模拟融合云、雾计算部署	27.3
传统部署方式	31.0

分析实验数据可知，模拟融合云、雾计算部署入侵检测方式的 CPU 平均利用率相比传统部署方式偏低，能够减轻终端设备上的计算压力。

4　结论与展望

本文利用自编码器完成特征缩减的工作，有效降低系统计算量并帮助分类模型更有效地进行训练。在分类模块，本文采用改进的自注意力机制，通过自注意力机制捕捉一段时间内流量数据的相关性，并在此基础上利用因果卷积操作学习流量数据的局部位置信息，综合学习当前流量数据与自注意力关注域内各流量数据的相关程度。在此基础上，模拟融合云、雾计算技术的入侵检测方法来缓解终端设备的计算压力。实验证明本文提出的入侵检测方法可以有效解决现代网络环境在入侵检测方面面

临的问题,为现代网络环境入侵检测系统的设计提供了有效方案。

在未来的工作中,希望可以利用真实的现代网络数据,如物联网设备产生的数据来进行相关问题的研究,同时,希望可以将该入侵检测系统的设计方案运用到真实的网络环境下进行测试,以证明该系统的实时、准确和可靠。考虑到现有部分攻击采用流量加密的形式使攻击更具有隐蔽性,希望下一步的工作可以解决这方面的问题。

参考文献

[1] LI Xukui, CHEN Wei, ZHANG Qianru, et al. Building auto-encoder intrusion detection system based on random forest feature selection[J]. Computers & Security, 2020, 95: 101851. DOI: 10.1016/j.cose.2020.101851

[2] SHONE N, NGOC T N, PHAI V D, et al. A deep learning approach to network intrusion detection[J]. IEEE Transactions on Emerging Topics in Computational Intelligence, 2018, 2(1): 41. DOI: 10.1109/TETCI.2017.2772792

[3] WANG Wei, LIU Jiqiang, PITSILIS G, et al. Abstracting massive data for lightweight intrusion detection in computer networks[J]. Information Sciences, 2018, 433: 417. DOI: 10.1016/j.ins.2016.10.023

[4] VIEGAS E, SANTIN A O, ABREU V. Machine learning intrusion detection in big data era: a multi-objective approach for longer model lifespans [J]. IEEE Transactions on Network Science and Engineering, 2020, 8(1): 366. DOI: 10.1109/TNSE.2020.3038618

[5] MAHONEY, MATTHEW V C, PHILIP K. Learning models of network traffic for detecting novel attacks[R/OL]. (2002-05-24)[2022-03-25]

[6] KASONGO S M, SUN Yanxia. A deep long short-term memory based classifier for wireless intrusion detection system[J]. ICT Express, 2020, 6(2): 98. DOI: 10.1016/j.icte.2019.08.004

[7] KAN Xiu, FAN Yixuan, FANG Zhijun, et al. A novel IoT network intrusion detection approach based on adaptive particle swarm optimization convolutional neural network[J]. Information Sciences, 2021, 568: 147. DOI: 10.1016/j.ins.2021.03.060

[8] FU Yanfang, DU Yishuai, CAO Zijian, et al. A deep learning model for network intrusion detection with imbalanced data[J]. Electronics, 2022, 11(6): 898. DOI: 10.3390/electronics11060898

[9] ANDRESINI G, APPICE A, CAFORIO F P, et al. ROULETTE: a neural attention multi-output model for explainable network intrusion detection [J]. Expert Systems with Applications, 2022, 201: 117144. DOI: 10.1016/j.eswa.2022.117144

[10] 刘月峰, 蔡爽, 杨涵晰, 等. 融合 CNN 与 BiLSTM 的网络入侵检测方法[J]. 计算机工程, 2019, 45(12): 127
LIU Yuefeng, CAI Shuang, YANG Xihan, et al. Network intrusion detection method integrating CNN and BiLSTM [J]. Computer Engineering, 2019, 45(12): 127. DOI:10.19678/j.issn.1000-3428.0053263

[11] 宋勇, 侯冰楠, 蔡志平. 基于深度学习特征提取的网络入侵检测方法[J]. 华中科技大学学报(自然科学版), 2021, 49(2): 115
SONG Yong, HOU Bingnan, CAI Zhiping. Network intrusion detection method based on deep learning feature extraction[J]. Journal of Huazhong University of Science and Technology(Natural Science Edition), 2021, 49(2): 115. DOI:10.13245/j.hust.210219

[12] 郭志民, 周劼英, 王丹, 等. 基于 Transformer 神经网络模型的网络入侵检测方法[J]. 重庆大学学报, 2021, 44(11): 81
GUO Zhimin, ZHOU Jieying, WANG Dan, et al. Network intrusion detection method based on transformer neural network model[J]. Journal of Chongqing University, 2021, 44(11): 81

[13] DUA M. Machine learning approach to IDS: a comprehensive review[C]//2019 3rd International Conference on Electronics, Communication and Aerospace Technology (ICECA). Piscataway, NJ: IEEE, 2019: 117

[14] 丁红卫, 万良, 龙廷艳. 深度自编码网络在入侵检测中的应用研究[J]. 哈尔滨工业大学学报, 2019, 51(5): 185
DING Hongwei, WAN Liang, LONG Tingyan. Research on the application of deep auto-encoder network in intrusion detection[J]. Journal of Harbin Institute of Technology, 2019, 51(5): 185

[15] WANG Wenjuan, DU Xuehui, SHAN Dibin, et al. Cloud intrusion detection method based on stacked contractive auto-encoder and support vector machine[J]. IEEE Transactions on Cloud Computing, 2020, 10(3): 1634. DOI: 10.1109/TCC.2020.3001017

[16] YANG Yanqing, ZHENG Kangfeng, WU Chunhua, et al. Improving the classification effectiveness of intrusion detection by using improved conditional variational autoencoder and deep neural network[J]. Sensors, 2019, 19(11): 2528. DOI: 10.3390/s19112528

[17] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[C]//Neural Information Processing Systems. La Jolla, California: MIT Press, 2017: 30

[18] WIECLAW L, PASICHNYK V, KUNANETS N, et al. Cloud computing technologies in "smart city" projects[C]//2017 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS). Piscataway, NJ: IEEE, 2017: 339

[19] Al Y M, SCHAEFER D. Fog computing as a complementary approach to cloud computing[C]//2019 International Conference on Computer and Information Sciences (ICCIS). Piscataway, NJ: IEEE, 2019: 1

[20] 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 (MilCIS). Piscataway, NJ: IEEE, 2015: 1

[21] 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. DOI:10.1016/j.ins.2019.10.069

[22] DIVEKAR A, PAREKH M, SAVLA V, et al. Benchmarking datasets for anomaly-based network intrusion detection: KDD CUP 99 alternatives[C]//2018 IEEE 3rd International Conference on Computing, Communication and Security (ICCCS). Piscataway, NJ: IEEE, 2018: 1

[23] KIM J, KIM J, THU H L T, et al. Long short term memory recurrent neural network classifier for intrusion detection[C]//2016 International Conference on Platform Technology and Service (PlatCon). Piscataway, NJ: IEEE, 2016: 1

[24] YIN Chuanlong, ZHU Yuefei, FEI Jinlong, et al. A deep learning approach for intrusion detection using recurrent neural networks[J]. IEEE Access, 2017, 5: 21954. DOI:10.1109/ACCESS.2017.2762418

[25] CAO Ke, FENG Weiiia, MA Chunmei, et al. Network intrusion detection based on dense dilated convolutions and attention mechanism[C]//2021 International Wireless Communications and Mobile Computing (IWCMC). Piscataway, NJ: IEEE, 2021: 463

[26] KARANAM L, PATTANAIK K K, ALDMOUR R. Intrusion detection mechanism for large scale networks using CNN-LSTM [C]//2020 13th International Conference on Developments in eSystems Engineering (DeSE). Piscataway, NJ: IEEE, 2020: 323

(编辑 刘 彤)