可变融合的随机注意力胶囊网络入侵检测模型

张兴兰，尹晟霖
（北京工业大学信息学部，北京 100022）

摘 要：为了增强检测模型的准确率与泛化性，提出了一种可变融合的随机注意力胶囊网络的入侵检测模型，通过特征动态融合，模型能够更好地捕捉数据特征；同时使用随机注意力机制，减少了对训练数据的依赖，使模型更具有泛化能力。所提模型在 NSL-KDD 和 UNSW-NB15 数据集上进行验证，实验表明，模型在 2 种测试集上的准确率分别达到了 99.49%和 98.60%。

关键词：深度学习；入侵检测；网络空间安全；胶囊网络；随机注意力
中图分类号：TP309
文献标识码：A
doi: 10.11959/j.issn.1000−436x.2020220

Intrusion detection model of random attention capsule network based on variable fusion

ZHANG Xinglan, YIN Shenglin
The Information Department of Beijing University of Technology, Beijing 100022, China

Abstract: In order to enhance the accuracy and generalization of the detection model, an intrusion detection model of random attention capsule network with variable fusion was proposed. Through dynamic feature fusion, the model could better capture data features. At the same time, random attention mechanism was used to reduce the dependence on training data and make the model more generalization. The model was validated on NSL-KDD and UNSW-NB15 datasets. The experimental results show that the accuracy of the model on the two test sets is 99.49% and 98.60% respectively.

Key words: deep learning, intrusion detection, cyberspace security, capsule network, random attention

1 引言

随着网络技术的不断发展，互联网成了人们日常生活中的重要工具，社会的发展也越来越离不开网络。但是，随着人们在网络上的互动增多，网络安全问题层出不穷，网络数据流量与日俱增，使入侵行为特征更加多样化[1]，网络安全已成为影响网络发展的重要因素。入侵检测技术是网络空间安全中最重要的防御手段之一。入侵检测的本质是一种数据的分类任务，对于分类任务来说，在神经网络算法中，已经有了非常明显的效果[2]。文献[3-6]提出了基于纯深度神经网络的模型，通过线性层来提取特征，取得了不错的效果，但没有考虑模型的泛化能力。文献[7]提出了使用卷积神经网络（CNN, convolutional neural network）的方式来进行特征提取，将数据处理成 one-hot 的形式，通过数组重组形成 CNN 可以处理的数据结构，最后通过 Softmax 函数来进行数据的分类操作。文献[8]通过使用一维的卷积神经网络配合数据归一化的方式，解决了数据不平衡的问题，并取得了不错的效果。文献[9]提出了循环神经网络（RNN, recurrent neural network）的方式来进行特征提取，也取得了不错的效果。虽然 CNN 可以通过简单的方式来计算大量的数据，并且有着很强的特征提取能力，但是在池化

收稿日期：2020−06−18；**修回日期**：2020−08−18
基金项目：国家自然科学基金资助项目（No.61801008）
Foundation Item: The National Natural Science Foundation of China (No.61801008)

操作时，会舍弃一些信息，在图像处理方面，可能不会产生重要的影响，但是对于入侵数据而言，这些信息可能是至关重要的[10]。RNN 主要是处理具有时序关系的序列，然而入侵检测数据的特征之间并没有特定的先后顺序，使用 RNN 进行特征提取时会融入不必要的信息。

为了解决当前神经网络模型中存在的一些缺陷，本文研究了 2 种新的机制，即特征动态融合机制和随机注意力机制，并在胶囊网络的基础上，提出了可变融合的随机注意力胶囊网络入侵检测模型。在全局特征提取过程中，入侵检测模型使用注意力机制来提取，但是传统的注意力机制会依赖训练数据本身，导致模型的泛化能力降低。基于此，本文提出了随机注意力机制来减少模型对训练数据的依赖。另外，胶囊网络中的特征提取是通过卷积实现的，为了补充在卷积操作中可能丢失的信息，通过可变的特征融合机制，建立了多特征提取通道，将通过随机注意力机制提取到的特征与胶囊网络卷积层提取到的特征进行动态融合，补充了卷积操作中丢失的信息，并且在最后优化了胶囊网络中的压缩函数，使其能够更好地捕捉全局特征和局部特征的关系，减少噪声的干扰，具有比原来的胶囊网络更强的特征提取能力和检测能力，并具有同时针对入侵检测数据的特点，可以更好地检测出数据细节的变化。

最后，本文将模型用在 NSL-KDD 和 UNSW-NB15 入侵检测数据的分类上，并与非神经网络模型和传统神经网络模型进行对比。实验结果表明，本文的模型在泛化能力方面高于其他模型，在 UNSW-NB15 测试集上的准确率达到了 98.60%；在 NSL-KDD 训练集的准确率可达 99.49%，效率方面也有了提升。

2　相关理论

2.1　胶囊网络

2017 年 Hinton[11]首次提出了胶囊网络模型，并且该模型被认为可能成为下一代重要的神经网络模型。胶囊网络是由胶囊组成的。胶囊是一组神经元的集合，与传统的神经网络不同，在胶囊中，神经元的集合是向量或者矩阵。每个神经元表示了图像中出现的特定实体的各种属性，比如图片中物体的方向、所在的位置以及形态颜色，通过使用胶囊向量的模长来表示实体所存在的可能性大小。胶囊网络是低级胶囊通过动态路由机制来向高级胶囊传递信息的[11-12]，因此与传统的神经网络相比，胶囊是具有更强的特征提取能力的网络，特别是对细节的提取。到目前为止，还没有人使用胶囊网络来处理入侵检测数据类型的结构化数据。将入侵检测与胶囊网络结合，也是一个新的研究点。

2.2　注意力机制

注意力机制被广泛运用在深度学习中的各种任务中，其目标是从杂乱的信息中选取对当前任务有关的信息，减少噪声对结果的影响。在传统的注意力机制中，基本上都是通过单词之间的交互来确定最终的注意力分数，比如在自注意力机制中，序列中的每个单词参与注意力的运算，能够对较远距离的依赖关系进行建模。自注意力机制提供了强大的建模能力，但是需要进行所有单词之间的两两交互。虽然可以获得更多的交互信息，但所得结果会非常依赖所给训练集的内容，使模型缺少了泛化的能力。文献[13]提出了合成注意力机制，减少了对训练数据的依赖和模型的训练时间，并取得了与自注意力机制相差不多的效果，在特定任务中甚至要优于自注意力机制。

3　模型分析

可变融合的随机注意力胶囊网络的框架结构如图 1 所示。模型首先将数据处理成矩阵的形式，然后进行特征提取，将得到的特征矩阵进行融合，融合后的特征矩阵被包裹成胶囊的形式送到初级胶囊层中，通过动态路由机制后输出结果，送到输出层中。本文的目的是对流量数据进行分类。

3.1　随机注意力的特征提取

注意力机制对减少数据噪声有着十分重要的作用，但是自注意力在某些时候并不能取得理想的效果，反而会增加运算成本，降低效率。并且，自注意力机制会过多地依赖训练集本身，导致模型的泛化能力下降。为此，本文提出了一种全新的注意力模型，此模型不依赖序列中的特征，而是利用随机注意力的方式，通过模型的最终目的来自动地进行调整，减少了模型对训练集的过度依赖，并且随机注意力矩阵会根据预测值与真实值的误差，通过反向传播进行调整，以达到最好的分类效果。随机注意力省去了特征之间的交互，

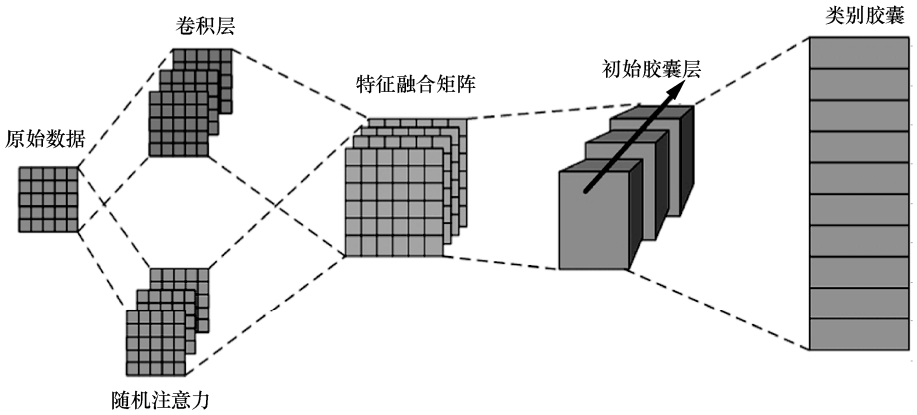

图 1 模型框架

对于包含多个特征的序列来讲，极大地缩减了模型的运算时间。

通过随机注意力机制，获得数据的注意力矩阵 $\boldsymbol{A}_{\text{Attention}}$。初始矩阵是使用 2 个随机初始化矩阵的乘积来生成随机矩阵 $\boldsymbol{R}$，如图 2 所示。

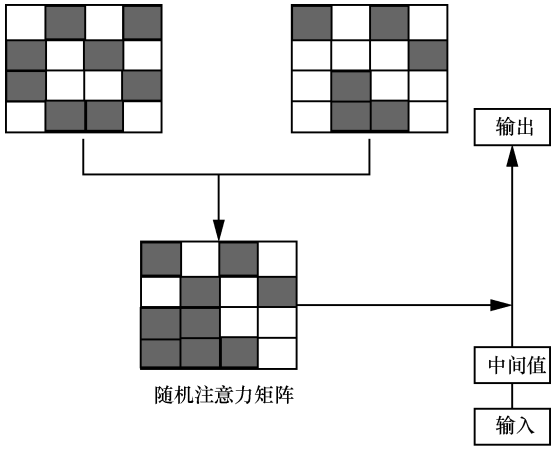

图 2 随机注意力机制

通过接收输入 $\boldsymbol{X}\in\mathbb{R}^{l\times d}$ 并输出特征矩阵 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$。其中，$l$ 是序列的长度，d 是特征的维度，n 是模型的维度。首先通过参数化矩阵 $\boldsymbol{W}$ 将输入从 d 维映射到 n 维的 $\boldsymbol{B}$。随机初始化 2 个可学习的矩阵 $\boldsymbol{R}_1,\boldsymbol{R}_2\in\mathbb{R}^{n\times n}$，并相乘，得到矩阵 $\boldsymbol{R}$，矩阵 $\boldsymbol{R}$ 用于注意力分数的计算。对于 $\boldsymbol{R}=\begin{bmatrix} r_{11} & \cdots & r_{n1} \\ \vdots & \ddots & \vdots \\ r_{n1} & \cdots & r_{nn} \end{bmatrix}$，经过 Softmax 函数后得到分数矩阵 $\boldsymbol{G}=\begin{bmatrix} g_{11} & \cdots & g_{n1} \\ \vdots & \ddots & \vdots \\ g_{n1} & \cdots & g_{nn} \end{bmatrix}$，并与 $\boldsymbol{B}$ 相乘，得到最后的特征矩阵 $\boldsymbol{A}_{\text{Attention}}\in\mathbb{R}^{n\times n}$。具体如式(1)~式(4)所示。

$$\boldsymbol{B}=\boldsymbol{X}\boldsymbol{W} \tag{1}$$

$$\boldsymbol{R}=\boldsymbol{R}_1\boldsymbol{R}_2^{\mathrm{T}} \tag{2}$$

$$g_i=\frac{e^{r_i}}{\sum_{j=1}^{n}e^{r_j}} \tag{3}$$

$$\boldsymbol{A}_{\text{Attention}}=\text{Softmax}(\boldsymbol{R})\boldsymbol{B} \tag{4}$$

将获得的 $\boldsymbol{A}_{\text{Attention}}$ 复制 M 次，形成 M 个 $n\times n$ 的矩阵 $\boldsymbol{A}_{\text{Attention}}$，并与胶囊网络提取到的局部特征进行融合。在胶囊网络中，原始数据通过卷积层进行卷积操作，产生 M 个 $n\times n$ 的矩阵。

3.2 可变融合

一些特征融合过程会直接将全局特征和局部特征相结合，但是这种操作在某些情况下会降低模型的准确率。在入侵检测的任务下，因为某些攻击类型是由某几个特定的特征来决定的，这时如果盲目地增加全局特征的信息，会产生许多噪声，使模型对此类攻击检测的准确率下降。同样，有些攻击是被所有特征控制的，这时局部特征就不能很好地给模型充分的信息，使模型检测出这一类的攻击。所以，寻找局部特征和全局特征融合的界限，对模型的检测准确率有着重要的影响。基于此，本文提出了可变融合机制。

通过特征提取得到了 2 个特征矩阵，为了使模型能够根据任务的目的来动态地融合全局特征和局部特征，本文提出了可变融合机制的方式，通过设定一个可学习的参数，将 2 个部分的特征矩阵按可变的比例进行融合得到最终的特征矩阵 $\boldsymbol{H}\in\mathbb{R}^{n\times n}$。

$$\boldsymbol{H}=\left[\left[1-f(\alpha)\right]\boldsymbol{A}_{\text{Attention}} \oplus f(\alpha)\boldsymbol{A}_{\text{ReluCon}}\right] \tag{5}$$

其中，⊕ 是元素连接符，即两组特征矩阵进行堆叠，对应位置的元素对齐，最终形成 2×M 个 n×n 的特征矩阵 $\boldsymbol{H}$；α 是一个可学习的参数，数值被初始化为 0.5；$f(x)$ 是一个范围函数，保证每次 α 更新后的值始终在[0,1]，如式(6)所示。

$$f(x)=\begin{cases}1, x>1\\ 0, x<0\\ x, \text{其他}\end{cases} \tag{6}$$

3.3　动态路由机制及输出层

特征矩阵被传到初级胶囊层中。下层胶囊需要将该层胶囊存储的计算结果传递给上层胶囊，其中传递过程是通过动态路由机制来实现的。文献[7]中的动态路由机制如式(7)~式(11)所示。

$$\boldsymbol{v}_j=\frac{\|\boldsymbol{s}_j\|^2}{1+\|\boldsymbol{s}_j\|^2}\frac{\boldsymbol{s}_j}{\|\boldsymbol{s}_j\|} \tag{7}$$

$$u_{j|i}=W_{ij}u_i \tag{8}$$

$$s_j=\sum_i c_{ij}u_{j|i} \tag{9}$$

$$c_{ij}=\frac{\exp(b_{ij})}{\sum_j \exp(b_{ij})} \tag{10}$$

$$b_{ij}\leftarrow b_{ij}+u_{j|i}v_j \tag{11}$$

一个胶囊的输出向量的长度表示实体出现的概率，因此需要一个非线性函数 Squashing 确保输出在[0,1]，$\boldsymbol{v}_j$ 是胶囊 j 的输出，$\boldsymbol{s}_j$ 是胶囊 j 的总输入。输入向量 $\boldsymbol{u}$ 与权重矩阵 $\boldsymbol{W}$ 相乘，通过这一步实现了低级特征与高级特征之间关系的编码。然后通过动态路由机制，来动态更新 L 层胶囊 i 到 L+1 层胶囊 j 的概率。b_{ij} 的初始值为 0，c_{ij} 为耦合系数，是低级胶囊到高级胶囊的权重、两层胶囊之间的相关性，值越大表示相关性越强。根据动态路由机制更新 b_{ij}，以达到更新 c_{ij} 的目的。迭代完成后，上层胶囊中的所有向量 $\boldsymbol{u}$ 乘以对应的权重 c_{ij} 得到 $\boldsymbol{s}_j$，最后通过激活函数 Squashing 得到最后的输出 $\boldsymbol{v}_j$，激活函数 Squashing 在保留向量方向的同时，将向量模的大小压缩到[0,1]。$\boldsymbol{v}_j$ 的模长就代表对应类别的概率。

本文中的动态路由机制的与文献[11]中相似，但是为了使动态过程更加接近入侵检测的数据，本文对压缩函数 Squashing 进行了修改，如图 3 所示。使用 x 和 y 分别代表压缩函数中的 $\boldsymbol{s}_j$ 和 $\boldsymbol{v}_j$，x 和 y 都是标量，在二维坐标系下研究函数的性质。从图 3 中可以发现，原始的压缩函数在处理模长较短的胶囊时，会把数值压缩到 0 附近，这样的全局压缩会导致在迭代更新时丢失部分胶囊的重要信息，同时函数增长速率过缓，对于模长比较短和模长比较长的胶囊会有明显的区分，但却不能很好地区分中间长度的胶囊，并且影响迭代速度。

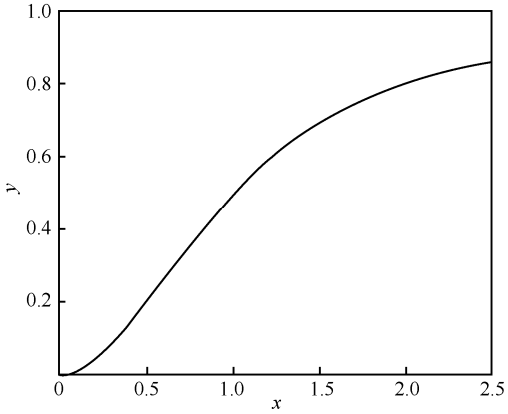

图 3　原始 Squashing 函数图像

为了解决这个问题，本文对原来的压缩方法进行了调整，改进的压缩方法如式(12)所示，函数图像如图 4 所示。此压缩函数的特点是在模长接近 0 时起到了放大作用，不像原来函数一样进行全局压缩，导致部分信息被忽略。

$$\boldsymbol{v}_j=\frac{\|\boldsymbol{s}_j\|^2}{0.25+\|\boldsymbol{s}_j\|^2}\frac{\boldsymbol{s}_j}{\|\boldsymbol{s}_j\|} \tag{12}$$

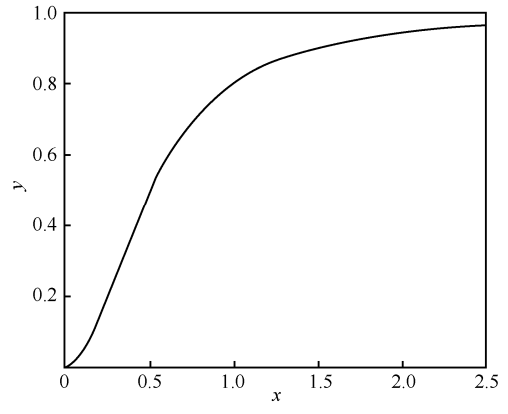

图 4　改进的压缩函数图像

在输出层部分，本文并没有采用原始胶囊网络中的重构操作，因为在特征提取过程中融合了全局特征，重构后将会带来一定的误差，分类的预测结果依旧采用 $\boldsymbol{v}_j$ 模长的形式来表示，在损失函数部分，

本文只使用了如式(13)所示的 Margin Loss 函数。

$$L_c = T_c \max(0, m^+ - \| v_c \|)^2 + \lambda(1 - T_c)\max(0, \| v_c \| - m^-)^2 \tag{13}$$

当预测的类 c 出现时，T_c=1，并且 m^+=0.9，m^-=0.1，λ=0.5，最后的损失是所有胶囊损失的总和。

4 实验及分析

4.1 数据集与数据处理

公开的网络入侵检测数据并不多，本文采用的是 NSL-KDD 入侵检测数据集[14-15]和 UNSW-NB15 入侵检测数据集[16]。

NSL-KDD 是对 KDD CUP99 数据集的优化。NSL-KDD 数据集解决了原数据集中的一些问题，但是 NSL-KDD 数据集仍然存在着另一些问题，同时，虽然该数据集也不是目前真实网络环境下入侵数据的代表，但它仍然可以作为有效的基准数据集来检测模型的能力。NSL-KDD 数据集包括 4 个子数据集：KDDTrain+、KDDTrain+_20Precent、KDDTest+、KDDTest+21。本文使用 KDDTrain+来进行训练，KDDTest+来进行测试，数据集中含有 4 种异常类型，被细分为 39 种攻击类型，其中有 17 种未知攻击类型出现在测试集中。每一条记录包括 41 个特征和 1 个类别标识。其中 41 个特征中是由 TCP（transmission control protocol）连接基本特征（9 种）、TCP 连接内容特征（13 种）、基于时间的网络流量统计特征（9 种）和基于主机的网络流量统计特征（10 种）组成。NSL-KDD 数据集标签数量如表 1 所示。

表 1 NSL-KDD 数据集标签数量

数据集	Normal/条	DoS/条	Probe/条	U2R/条	R2L/条	Total/条
KDDTrain+	67 345	45 926	11 655	52	995	125 973
KDDTest+	9 711	7 458	2421	200	2 754	22 544

UNSW-NB15 数据集中包含 6 个文件，其中 UNSWNB15_1.csv、UNSW-NB15_2.csv、UNSW-NB15_3.csv 和 UNSW-NB15_4.csv 包含了数据集中的所有记录，每个文件中含有正常数据和攻击数据。本文使用的是 UNSW_NB15_training-set.csv 和 UNSW_NB15_testing-set.csv，它将数据集分为了测试集和训练集，共有 9 种类型的攻击：Namely、Fuzzers、Analysis、Backdoors、DoS、Exploits、Generic、Reconnaissance、Shellcode 和 Worms。数据集中的每条数据包含 49 个特征，其中包括 Flow Features(1～5)、Base Features(6～18)、Content Features(19～26)、Time Features(27～35)。1～35 的特征是从数据包中搜集的综合信息，大多数特征是从报头中生成的。在此基础上，数据又增加了 General Purpose Features(36～40) 和 Connection Features(41～49)。

训练集中一共有 175 431 条记录，测试集中一共有 82 332 条记录。UNSW-NB15 数据集中标签数据如表 2 所示。

首先要进行数据处理，数据处理包括字符类型数字化、数据归一化 2 个步骤。

步骤 1 字符类型数字化

以 NSL-KDD 数据集为例：在数据集的特征中，有 3 个特征和类别标识是字符类型的。在字符类型数字化的过程中，一共采取了 2 种处理方式，分别是 one-hot 方式和标签编码的方式。协议类型的值有 3 种，分别是 TCP、UDP（user datagram protocol）和 ICMP（Internet control message protocol），目标主机的网络服务类型有 70 种，连接正常或者错误的状态有 11 种。在 one-hot 方式下，协议类型被处理为[1,0,0]、[0,1,0]、[0,0,1]的形式，其他特征处理过程类似，最终每条数据的长度为 121 维。在标签编码方式下，协议类型分别被处理为 0、1、2，其他特征处理过程类似，最终数据被处理成为每条长度为 41 维。

步骤 2 归一化处理

进行数值化之后，由于数值之间的量纲不同，会产生较大的差异。通过归一化处理后，可以消除不同特征之间的差异，对于离散型特征，采用最大最小归一化的方法，对于连续型特征，采用 Z-Score 的方式，将数值固定在[0,1]，如式(14)～式(15)所示。

表 2 UNSW-NB15 数据集标签数量

数据集	Normal/条	Fuzzers/条	Analysis/条	Backdoors/条	DoS/条	Exploits/条	Generic/条	Reconnaissance/条	Shellcode/条	Worms/条	Total/条
训练集	56 000	18 184	2 000	1 746	12 264	33 393	40 000	10 491	1 133	130	175 341
测试集	37 000	6 062	677	583	4 089	11 132	18 871	3 496	378	44	82 332

$$x_{\text{norm}}=\frac{x-x_{\min}}{x_{\max}-x_{\min}} \tag{14}$$

$$x^{*}=\frac{x-\mu}{\sigma} \tag{15}$$

其中，x 是原始数据，$x_{\min}$ 是同一特征中的最小值，$x_{\max}$ 是同一特征中的最大值，σ 是特征中的标准差，μ 是样本中的均值，x_{norm} 和 x^{*} 是原始数据标准归一化后的结果。

4.2　实验结果及分析

本文使用准确率 AC、精确率 P、召回率 R 和 F1-score（如式(16)~式(19)所示）作为实验效果优劣的评价标准[17]，其中 TN 是数据为正常且预测也为正常的数量，TP 是数据为异常且预测也为异常的数量，FN 是数据为异常但预测为正常的数量，FP 是数据为正常但预测为异常的数量。

$$\text{AC}=\frac{\text{TN+TP}}{\text{TN+TP+FN+FP}} \tag{16}$$

$$P=\frac{\text{TP}}{\text{TP+FP}} \tag{17}$$

$$R=\frac{\text{TP}}{\text{TP+FN}} \tag{18}$$

$$\text{F1-score}=\frac{2PR}{P+R} \tag{19}$$

在实验中本文发现在对字符特征进行处理的时候，采取 one-hot 编码和标签编码对实验结果影响不大。因此，本文使用标签编码的方式。同时在实验过程中为了避免出现过拟合的现象，进行 5 折交叉验证。将训练集等分为 5 份，其中 4 份用于训练，1 份用于验证，验证集的最终结果取平均值。数据集划分结果如图 5 所示。

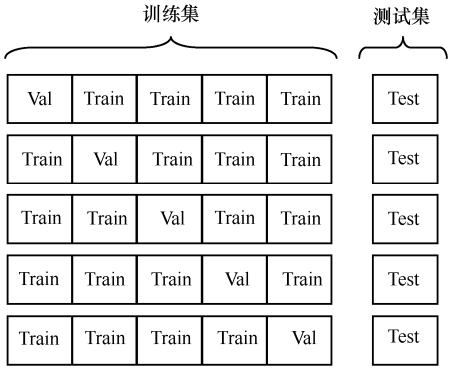

图 5　数据集划分结果

1) 在 NSL-KDD 数据集上的实验

在处理数据的同时删除数值全为 0 的 num_outbound_cmds 特征项，产生每条数据长度为 40 的特征数组，由于数据存在着严重的类别不平衡现象，对数据进行类别不平衡处理，对新生成的数据集重新划分，其中训练集和测试集中的数据没有重叠。同时使用了非神经网络算法：KPCA（kernel principal components analysis）+SVM（support vector machine）、KPCA+KNN（k-nearest neighbor）、GBT（gradient boosting tree）及神经网络算法 Vanilla Capsule、CNN、CNN+LSTM 来进行对比，各个模型的评价指标如表 3 所示。

在验证集中，所有的模型都达到了不错的效果，本文模型甚至达到了 99.80%的准确率。但是在测试集中，能看出明显的差距。在非神经网络算法中，效果最好的模型准确率达到了 99.29%，本文模型则达到了 99.49%。在神经网络算法中，CNN 和 CNN-LSTM 的混合模型在测试集中的准

表 3　　**NSL-KDD 中不同模型的评价指标**

模型	验证集				测试集			
	AC	P	R	F1-score	AC	P	R	F1-score
KPCA + SVM	98.96%	99.19%	99.22%	99.20%	95.89%	96.23%	96.06%	96.14%
KPCA + KNN	99.33%	99.34%	99.36%	99.35%	97.81%	97.38%	97.89%	97.63%
GBT	99.40%	99.46%	99.47%	99.47%	99.29%	99.23%	99.25%	99.24%
CNN	97.80%	97.82%	97.82%	97.82%	94.69%	94.70%	94.72%	94.71%
CNN+ LSTM	97.88%	97.99%	98.03%	98.01%	95.01%	95.98%	95.99%	95.98%
Vanilla Capsule	99.18%	98.89%	99.19%	99.04%	97.29%	97.30%	97.31%	97.30%
本文模型	99.80%	99.82%	99.82%	99.82%	99.49%	99.47%	99.46%	99.46%

确率有了明显的下降，原始胶囊网络的准确率也下降了 1.89%。测试集中存在训练集不曾出现过的攻击特征，经过处理后的训练集和测试集的数据分布仍存在一定的差异。这说明 CNN 和 CNN-LSTM 只能够很好地拟合训练集中的数据，并没有很好的泛化能力。而胶囊网络具有不错的泛化能力，并且通过改进，本文模型在测试集上的准确率比原始胶囊网络高了 2.20%，有了明显的提升。

本文研究了动态路由机制中的迭代次数、特征融合率和准确率之间的关系，从图 6 迭代次数与准确率的关系中可以得知，一开始准确率与迭代次数呈增长趋势，但当迭代超过 4 次后，模型在测试集的准确率开始下降，这说明模型出现了过拟合。迭代次数的选择，对于模型检测的效果也至关重要，过多的迭代次数会导致模型效率下降并降低准确率，迭代次数不够会出现欠拟合。

图 7 是在迭代次数为 4 时融合率和准确率之间的关系。从图 7 中可以看出，随着准确率的提高，融合率在不断缩小，由式(5)可以得出，模型的关注内容不断趋向于全局特征。

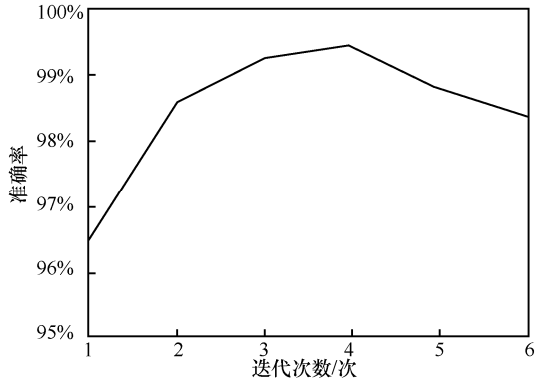

图 6 迭代次数与准确率的关系

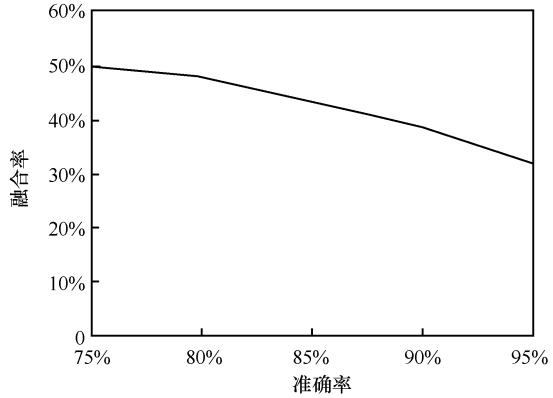

图 7 融合率与准确率的关系

为了进一步验证动态融合机制的有效性，本文与直接将局部特征和全局特征拼接的模型 Random + Capsule 进行了对比，2 个模型在测试集上的正确率如表 4 所示。从表 4 中可以看出，动态融合机制的模型在测试集中有着更高的准确率。

表 4 模型对比的准确率

模型	AC
Random+Capsule	98.18%
本文模型	99.49%

本文还与传统的 Transformer 中的自注意力机制结合的胶囊网络进行比较，通过对比可以发现，使用自注意力机制的模型过度关注了训练集的数据内容，导致泛化能力明显减弱。

表 5 本文模型与胶囊网络的对比

模型	验证集	测试集
Capsule+Self-Attention	99.31%	97.13%
本文模型	99.80%	99.49%

为了验证对压缩函数猜想，本文对比了原压缩函数和本文改进的压缩函数，其准确率和迭代次数关系如图 8 所示。

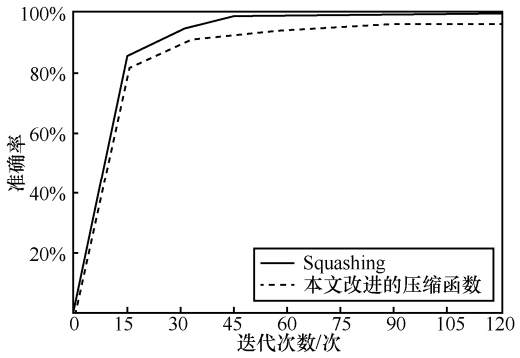

图 8 2 种函数迭代次数与准确率

由图 8 可知，本文改进的压缩函数具有更高的准确率和更快的收敛速度。

2) 在 UNSW-NB15 数据集上的实验

数据处理的过程与 NLS-KDD 数据集类似，各模型的评价指标如表 6 所示。

UNSW-NB15 数据集弥补了 NSL-KDD 数据集中的不足。从结果中可知，与 NLS-KDD 数据集中的结果相比较，非神经网络算法的准确率有所下降，效果最好的模型达到了 97.99%的准确率；神经网络算法中，CNN 和 CNN+LSTM 的准

表 6　　UNSW-NB15 中不同模型的评价指标

模型	验证集				测试集			
	AC	*P*	*R*	F1-score	AC	*P*	*R*	F1-score
KPCA + SVM	98.42%	90.32%	96.33%	93.23%	95.37%	94.99%	95.02%	95.00%
KPCA + KNN	98.79%	98.77%	98.79%	98.78%	96.22%	96.19%	96.17%	96.18%
GBT	99.21%	99.11%	99.10%	99.10%	97.99%	97.98%	97.79%	97.88%
CNN	96.89%	96.90%	96.89%	96.89%	94.20%	94.31%	94.33%	94.32%
CNN + LSTM	97.73%	97.70%	97.71%	97.70%	95.65%	95.34%	95.34%	95.34%
Vanilla Capsule	97.89%	97.83%	97.86%	97.84%	97.47%	97.43%	97.45%	97.44%
本文模型	99.24%	99.21%	99.19%	99.20%	98.60%	98.59%	98.61%	98.60%

确率仍然有所不足。本文模型在测试集中的准确率，比原始的胶囊网络高了 1.13%，仍保持了较高的泛化能力。

本文分析了融合率在 UNSW-NB15 数据集中的变化，结果如图 9 所示。

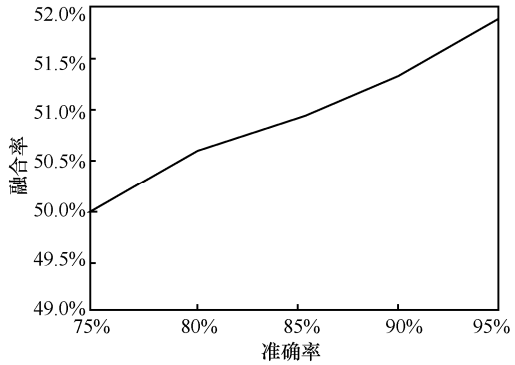

图 9　UNSW-NB15 中融合率与准确率的关系

从图 9 中可以看到，在 UNSW-NB15 数据集中融合率有了轻微的上升，说明模型提高了关注局部特征的比重。

5　结束语

本文提出了可变融合的随机注意力胶囊网络，对 NSL-KDD 和 UNSW-NB15 数据集进行训练和测试。使用随机注意力机制，获得了全局特征，减少了模型的训练时间，通过动态融合机制，将全局特征与局部特征按合适的比例进行融合，能够更好地把握全局与局部的关系，最后通过胶囊网络进行预测。结果表明，本文模型的泛化能力明显增强，提高了入侵检测的正确率，但与非神经网络算法相比，在运行时间上有待提高。胶囊网络的动态路由机制仍是一个研究热点，今后可以通过对路由机制的改进，在进一步增强模型对入侵检测数据的泛化能力同时减少运行时间以提高效率。

参考文献：

[1] LIAO H J, LIN C H R, LIN Y C, et al. Intrusion detection system: a comprehensive review[J]. Journal of Network & Computer Applications, 2013, 36(1):16-24.

[2] KIM K, AMINANTO M E. Deep learning in intrusion detection perspective: Overview and further challenges[C]// International Workshop on Big Data & Information Security. Piscataway: IEEE Press, 2018: 5-10.

[3] TANG T A, MHAMDI L, MCLERNON D, et al. Deep learning approach for network intrusion detection in software defined networking[C]// International Conference on Wireless Networks & Mobile Communications. Piscataway: IEEE Press, 2016, doi:10.1109/WINCOM.2016.7777224.

[4] GU G X, CHEN C T, BUEHLER M J. De novo composite design based on machine learning algorithm[J]. Extreme Mechanics Letters, 2017, 18:19-28.

[5] VINAYAKUMAR R, SOMAN K P, POORNACHANDRAN P. Applying convolutional neural network for network intrusion detection[C]// 2017 International Conference on Advances in Computing, Communications and Informatics . Piscataway: IEEE Press, 2017, doi:10.1109/ICACCI.2017.8126009.

[6] AL-ZEWAIRI M , ALMAJALI S , AWAJAN A . Experimental evaluation of a multi-layer feed-forward artificial neural network classifier for network intrusion detection system[C]// The 2017 International Conference on New Trends in Computing Sciences. Piscataway: IEEE Press, 2018: 167-172.

[7] VINAYAKUMAR R, ALAZAB M, KP S, et al. Deep learning approach for intelligent intrusion detection system[J]. IEEE Access, 2019, PP(99):1-1.

[8] AZIZJON M, JUMABEK A, KIM W. 1D CNN based network intrusion detection with normalization on imbalanced data[C]// 2020 International Conference on Artificial Intelligence in Information and Communication . Piscataway: IEEE Press, 2020, doi: 10.1109/ICAIIC48513.2020.9064976.

[9] KIM J , KIM J , THU H L T , et al. Long short term memory recurrent neural network classifier for intrusion detection[C]// International Conference on Platform Technology & Service. Piscataway: IEEE

Press, 2016, doi: 10.1109/PlatCon.2016.7456805.
[10] CHEN Y, ABRAHAM A, YANG J. Feature selection and intrusion detection using hybrid flexible neural tree[C]// International Symposium on Neural Networks. Berlin: Springer, 2005: 439-444.
[11] SABOUR S, FROSST N, HINTON G E. Dynamic routing between capsules[C]//Proceeding of the Neural Information Processing Systems. New York: ACM Press, 2017: 3856-3866.
[12] HINTON G E，SABOUR S, FROSST N． Matrix capsules with EM routing[C]//Sixth International Conference on Learning Representations. 2018: 1-9.
[13] TAY Y, BAHRI D, METZLER D, et al. Synthesizer: rethinking self-attention in transformer models[J]. arXiv Preprint, arXiv: 2005.00743v1, 2020.
[14] 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. Piscataway: IEEE Press, 2009: 1-6.
[15] DHANABAL L, SHANTHARAJAH S P. A study on NSL-KDD dataset for intrusion detection system based on classification algorithms[J]. International Journal of Advanced Research in Computer and Communication Engineering, 2015, 4(6): 446-452.
[16] MOUSTAFA N, SLAY J. UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)[C]// Military Communications and Information Systems Conference. Piscataway: IEEE Press, 2015: 16.
[17] DONG B, WANG X. Comparison deep learning method to traditional methods using for network intrusion detection[C]// IEEE International Conference on Communication Software & Networks. Piscataway: IEEE Press, 2016: 581-585.

[作者简介]

张兴兰（1970− ），女，山西吕梁人，博士，北京工业大学教授，主要研究方向为密码学、信息安全等。

尹晟霖（1996− ），男，山东淄博人，北京工业大学硕士生，主要研究方向为深度学习、信息安全等。