基于生成对抗神经网络的流量生成方法研究

康　未，李维皓，刘桐菊
（华北计算机系统工程研究所，北京　100083）

摘　要：网络仿真中的流量生成对于确保仿真效果至关重要。目前常见的网络流量生成器通常基于某种随机模型，生成的流量只能服从指定的随机分布。实际网络中的随机模型往往难以确定，导致现有模型对真实网络流量的仿真有一定的偏差。为了解决这些问题，提出了基于生成对抗神经网络的时空相关流量生成模型；对网络流量数据改进了其编码方式，并使用 Z-score 处理流量数据，使数据趋于标准正态分布；提出了一种网络流量时空相关性的度量方法。实验结果表明，相较于现有的基线生成方式，所提出的方法在真实性和相关性的度量上平均提高了 9%。

关键词：网络仿真；网络流量生成；生成对抗神经网络；时空相关性
中图分类号：TP309　　**文献标识码：**A　　**DOI：**10.19358/j.issn.2097-1788.2024.06.005
引用格式：康未，李维皓，刘桐菊. 基于生成对抗神经网络的流量生成方法研究［J］. 网络安全与数据治理，2024，43(6)：33-41.

Traffic generation methods based on generative adversarial neural networks

Kang Wei，Li Weihao，Liu Tongju
(National Computer System Engineering Research Institute of China，Beijing 100083，China)

Abstract：Traffic generation in network simulation is crucial for ensuring simulation effectiveness. Currently，common network traffic generators are typically based on a certain random model，where the generated traffic adheres to a specified random distribution. However，determining a realistic random model for actual network traffic is often challenging，leading to biases in current models when simulating real network traffic. To address these issues，this paper proposes a spatiotemporal-correlated traffic generation model based on Generative Adversarial Neural Networks (GANs). The encoding method for network traffic data is improved，and Z-score is applied to process traffic data，making the data tend toward a standard normal distribution. Additionally，a measurement method for evaluating the spatiotemporal correlation of network traffic is introduced. Experimental results indicate that，compared to existing baseline generation methods，the proposed approach averages 9% improvement in measures of authenticity and correlation.

Key words：network simulation；network traffic generation；generative adversarial neural networks；spatiotemporal correlation

0　引言

随着计算机和网络技术的发展，网络环境变得日益复杂[1]，网络攻击事件频发，使得网络的安全性测试和评估尤为重要。在网络攻击方式不断演变之下，人工智能的发展使得网络的攻击和防御进入了一种新的态势[2]，导致传统的测试方法无法应对当前的需求。在此背景之下，亟需一种能够模拟真实场景的流量生成工具，以评估网络性能、检测网络潜在的风险和优化应用程序，防止网络攻击事件的发生。

现有网络流量生成器主要分为[3]：最大吞吐量生成器，以恒定或最大的速率生成网络流量，常用于网络带宽的测试，例如 Iperf2[4]；回放生成器，重放之前捕获的网络流量，如 TCPReplay[5]；随机模型生成器，利用随机模型来模拟网络流量的特征，如 Harpoon[6]；脚本生成器，允许用户编写复杂的逻辑，动态地修改数据包内容，可以生成任意类型的数据包，如 Moongen[7] 和 Scapy[8]；特定场景生成器，对特定应用程序实现的流量生成器，高度定制化，很难在其他环境继续使用。其中只有随机模型生成器和脚本生成器具有更多的灵活性，能够在不同的维度模拟网络流量，不过这需要用户首先对采集到的数据设定一个统计模型，或者利用统计方法估计一个模型。因此如果设定的随机分布不正确，或者流量数据

的模型是未知的，那么生成流量就会不准确甚至失败。

深度神经网络在拟合随机模型上有着天然的优势，可以利用在真实网络中采集的流量数据，拟合对应数据的统计分布。而生成对抗神经网络（Generative Adversarial Networks，GAN）[9] 已经在很多领域被广泛应用，如图像[10]、音频[11]、视频[12]等。通过对现有网络流量生成方法的研究，基于统计的流量生成方法如图 1 所示，首先在真实网络中采集一段数据，然后提取需要生成的网络流量的特征；将数据经过预处理，得到神经网络的输入数据，使用一种或多种统计模型进行拟合训练，利用拟合后的模型生成网络流量在另一个时间段的特征；最后再将生成特征组合成的流量，发送到模拟的网络环境中，用以测试网络环境，或者研究网络用户的行为。

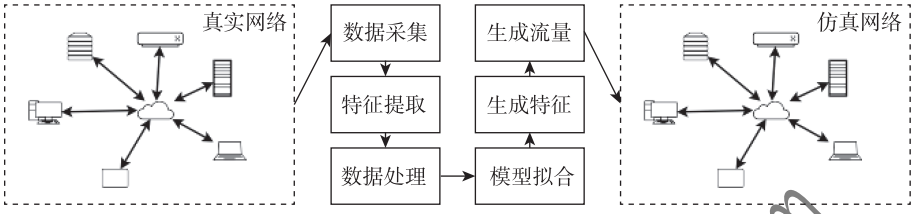

图 1　网络流量生成架构

现有的生成对抗神经网络流量生成方法[13-14]很少考虑流量的时空特征。但是在现实生活中，网络流量的产生依赖于网络用户的行为，使得网络流量在时空维度的特征呈现出多样性，而这种多样性造成数据流量服从的随机分布很难直接被描述，从而增加了拟合真实模型的难度。针对这一问题，本文分析流量在时空维度的多种特征，提出了一种融合时空特征的神经网络模型（Network Traffic GAN，NTGAN），利用条件机制向生成器传入时序数据，使得生成的流量能够在时空相关性上取得更好的效果，进而能够更好地模拟真实的网络环境。

另一方面，由于数据包的数量、长度、IP 地址等特征不均匀，网络流量会呈现出突发性。如在一段特定的网络流量中，绝大多数数据包指向很少的一部分 IP 地址，这导致网络流量的特征分布极不均匀。研究表明，在输入是正态分布的情况下，机器学习算法的训练结果往往会更好[15]。于是需要一种方法使输入的网络流量的分布通过一种可逆变换尽可能地接近正态分布，从而得到更好的训练效果。针对这一问题，本文在现有流量预处理的基础上[16]，改进了其编码方式，并使用 Z-score 对流量数据进行处理，使得网络流量能够更符合正态分布。

对生成的网络流量真实性的衡量[17]是一个新的挑战，现有评价方式不足以评价网络流量的时空关联性。为此本文提出了一种时空相关生成流量的度量方法，该度量方法在主成分分析的基础上，对生成的网络流量进行主成分分析降维处理，然后在低维对真实数据和生成数据进行比较。

为了验证所提出的模型，本文在空间和时间维度对仿真流量进行拟合实验。结果显示，与现有的生成方式相比，本文提出的方法的真实性和相关性指标平均提高了 9%。

1　相关工作

为了改进网络流量生成模型，研究者们提出了多种创新性方法。Ring 等[16]将生成对抗网络（GAN）引入网络流量生成领域，特别是处理网络数据包中的离散数据，例如 IP 地址和端口号。通过数值变换、二进制变换和嵌入变换等预处理方法，将离散数据转换为连续数据，并利用 GAN 网络在 CIDDS-001[18]数据集上生成网络数据，以供入侵检测系统测试。进一步的研究包括 Dowoo[19]等的 PcapGAN 模型，该模型通过编码器、生成器和解码器的协同工作，能够生成包括网络攻击数据和正常数据在内的 Pcap 数据，用于入侵检测系统。而 Rigaki[13]等通过修改恶意软件的代码，利用 GAN 网络生成网络流量，以模仿 FaceBook 聊天网络流量的行为，从而使得恶意软件的通信过程能够规避入侵防御系统的检测。

其他一些研究探索了不同的应用场景。比如，Lin[20]等提出的 DoppelGANger 模型可同时生成数据属性和序列特征，广泛应用于网络流量时间序列、地理分布宽带测量和集群使用测量等多个数据集。Cheng[14]使用 GAN 网络生成了 IP 层的网络数据包，包括 ICMP、DNS 和 HTTP 等协议的查询，实现了网络流的请求和响应。Hui[17]等通过使用知识增强生成对抗神经网络生成大规模物联网流量，引入语义知识并通过条件机制将设备类别合并到流量生成中。而 Shahid[21]等使用生成式深度神经网络生成了双向的网络流，以欺骗检测系统，使其认为生成的网络流是合法的网络数据包。胡勇进[22]等人提出一种采用 LetNet-5 深度神经网络来欺骗攻击者的流量模型，为流量混淆和欺骗提供了一种方法。这一系列的研究不仅在方法上取得了显著的进展，也为网络流量生成模型提供

了广泛的应用方向。

相较于之前的研究，本文在网络流量数据的预处理方面进行了创新性改进。通过对网络流量数据进行更为精细和有效的预处理，致力于提高生成模型的性能和适用性。在这一基础上，本文进一步引入了一个时空关联的网络流量生成模型，该模型不仅考虑了数据的时间序列特征，还考虑了数据之间的空间关联，从而更全面地捕捉网络流量的复杂结构和变化规律。除了模型的创新性，本文还提出了对生成数据的度量方法，为评估生成模型的性能提供了具体的标准和指标。

2 预备知识

2.1 生成对抗神经网络

生成对抗神经网络（Generative Adversarial Networks，GAN）[9]是一种非监督学习的框架，用来通过训练数据找到与之相似的分布。生成对抗神经网络由两部分组成：生成器和判别器，其中生成器 G 通过一个随机向量 z 得到 $G(z)$，判别器 D 用来验证传入的数据是不是真实的。GAN 的目标是利用训练数据生成尽可能真实的数据，以欺骗判别器。GAN 以如下形式表示：

$$\min_G \max_D V(D, G) \tag{1}$$

$$V(D, G) = \mathscr{E}_{x\sim p_d(x)}[\log(D(x))] + \mathscr{E}_{z\sim p_z(z)}[\log(1-D(G(z)))] \tag{2}$$

其中，$p_d(x)$ 表示训练数据的分布；$p_z(z)$ 表示随机向量，作为生成器 G 的输入；$G(z)$ 表示生成的数据分布；$D(x)$ 表示数据 x 真实性的概率。在训练过程中，希望生成器 G 能够最小化 $V(D,G)$，判别器 D 能够最大化 $V(D,G)$。经过多次迭代使模型收敛，达到一种纳什均衡。

2.2 JS 散度

JS 散度（Jsensen-Shannon Divergence，JSD）是一种度量不同随机分布之间距离的方法，定义如下：

$$\mathrm{JSD}(P\|Q) = \sqrt{\frac{\mathrm{KL}(P\|M) + \mathrm{KL}(Q\|M)}{2}} \tag{3}$$

其中，P 表示真实数据的分布；Q 表示生成数据的分布；$M = \frac{1}{2}(P+Q)$；KL 是 KullbackLeibler 散度，亦称相对熵或信息散度，可用于度量两个概率分布之间的差异。给定两个概率分布 P 和 Q，二者之间的 KL 散度定义为：

$$\mathrm{KL}(P\|Q) = \int_{-\infty}^{\infty} p(x) \log\frac{p(x)}{q(x)}\mathrm{d}x \tag{4}$$

其中，$p(x)$，$q(x)$ 分别为 P 和 Q 的概率密度函数。将 $\mathrm{KL}(P\|Q)$ 展开可得：

$$\mathrm{KL}(P\|Q) = \int_{-\infty}^{\infty} p(x)\log\frac{p(x)}{q(x)}\mathrm{d}x = H(P) - H(P, Q) \tag{5}$$

其中，$H(P)$ 为熵，$H(P, Q)$ 为 P 和 Q 的交叉熵。在信息论中，熵 $H(P)$ 表示对来自 P 的随机变量进行编码所需的最小字节数，而交叉熵 $H(P, Q)$ 则表示使用基于 Q 的编码对来自 P 的变量进行编码所需的字节数。因此，KL 散度可认为是使用基于 Q 的编码对来自 P 的变量进行编码所需的额外字节数，当且仅当 $P=Q$ 时，额外字节数为零。

本文采用 JS 散度刻画不同生成模型的生成流量数据与真实流量数据的距离，以此来衡量不同模型生成数据的真实性，距离越短表示生成的数据越真实。

2.3 主成分分析

数据流量通常是多维的，为了分析数据的时空相关性，通常使用主成分分析（Principal Component Analysis，PCA）[23]将数据流量降维到低维，同时使保留下来的数据具有最大方差。通过 PCA，可以找到数据中最重要的特征或主成分，从而减少数据的维度而不致损失太多的信息，进而可以使用可视化的方法来比较不同特征之间的相关性。

假设有一个包含 n 个样本和 p 个特征的数据集，可以将数据表示为一个 $n\times p$ 的矩阵 $\boldsymbol{X} = \{x_{ij},\ i=1, 2, \cdots, n,\ j=1, 2, \cdots, p\}$，其中每行对应一个样本，每列对应一个特征。主成分分析的目标是找到一个新的特征空间，将原始数据映射到这个新的空间，使得在新的特征空间中数据的方差最大。其基本步骤如下：

（1）中心化数据：对原始数据进行中心化，即将每个特征减去相应特征的均值 $\bar{x}_j = \frac{1}{n}\sum_{i=1}^{n} x_{ij}$，得到中心化后的数据：$x_{ij} \leftarrow x_{ij} - \bar{x}_j$。

（2）计算协方差矩阵 $\boldsymbol{C}$：协方差矩阵描述了不同特征之间的相关性和方差，使用中心化后的数据计算其协方差矩阵 $\boldsymbol{C} = \boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}$。

（3）特征值分解：对协方差矩阵 $\boldsymbol{C}$ 进行特征值分解，得到特征值 $\lambda_1, \lambda_2, \cdots, \lambda_p$ 和对应的特征向量 $\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_p$。特征值表示方差的大小，而特征向量则表示数据在相应方向上的分布。

（4）选择主成分：将特征值排序，选取前 k 个最大的特征值对应的特征向量，构成变换投影矩阵 $\boldsymbol{W}$。这些特征向量就是数据的主成分，而对应的特征值表示了在这些主成分方向上的方差。

（5）数据变换：将原始数据投影到所选择的主成分上，得到降维后的数据 $\boldsymbol{Z} = \boldsymbol{W}\boldsymbol{X}$。

3 NTGAN 流量生成方法

本文提出了基于生成对抗神经网络的时空相关流量

生成方法 NTGAN，主要包含三部分：网络流量预处理的方法，基于生成对抗神经网络的流量生成模型，网络流量时空相关性的评价标准，如图 2 所示。

3.1 网络流量数据预处理

本文用到的数据集来自网络中捕获到的流量，其数据格式通常是 .pcap 文件，这种格式的数据需要经过处理才能作为神经网络的输入。由于数据流量特征的数值通常是离散的，比如端口号字段，通常只有几个可用的数字，而其取值范围却是 0 到 65 535。在特定的网络中，所有的 IP 地址一般不会同时出现。所以如果直接使用数值转换，对于普通的网络流量，数值会有很大的间断，这很可能导致模型本身不可微，或者模型很难收敛。对此，本文对数据流量的关键特征进行分析，对其进行编码，使得数据流量的特征能够连续。依据正态分布中间多两边少的特征和流量特征出现的次数，将数据特征的编码映射到连续的整数，使其尽可能服从正态分布。再使用 Z-score 正则化方法，将特征向量映射到 N（0，1），得到标准化的向量数据。然后利用 MinMax 数据处理方法，将流量数据映射到（0，1），得到神经网络的输入数据。本文提取了网络流量中常见的特征，其中包括数据包发送的时间、数据包发送的间隔、数据包长度、数据包类型、源 IP 地址、目标 IP 地址、源端口号和目标端口号。对这些数据特征进行了预处理，具体步骤如下：

（1）过滤数据包：由于数据结构的一致性，本文只对 TCP 和 UDP 协议进行了处理，将无关协议的数据包丢弃，得到只包含 TCP 和 UDP 协议的数据包集合。

（2）提取数据包特征：通过读取 .pcap 数据包文件，提取流量数据特征，包括数据包发送的时间、数据包发送的间隔、数据包长度、数据包类型、源 IP 地址、目标 IP 地址、源端口号和目标端口号。

（3）特征编码：类似于 Word2Vec[24] 编码方式，对所有独立的特征进行编码，使得每个特征映射到一个整数。根据特征出现的次数中间多两边少，将特征依次编码到连续的整数序列。

（4）数据标准化：将特征对应的整数进行 Z-score 变换，使得数据映射到 N（0，1），再将特征对应的整数进行 MinMax 变换，使得数据映射到（0，1）。

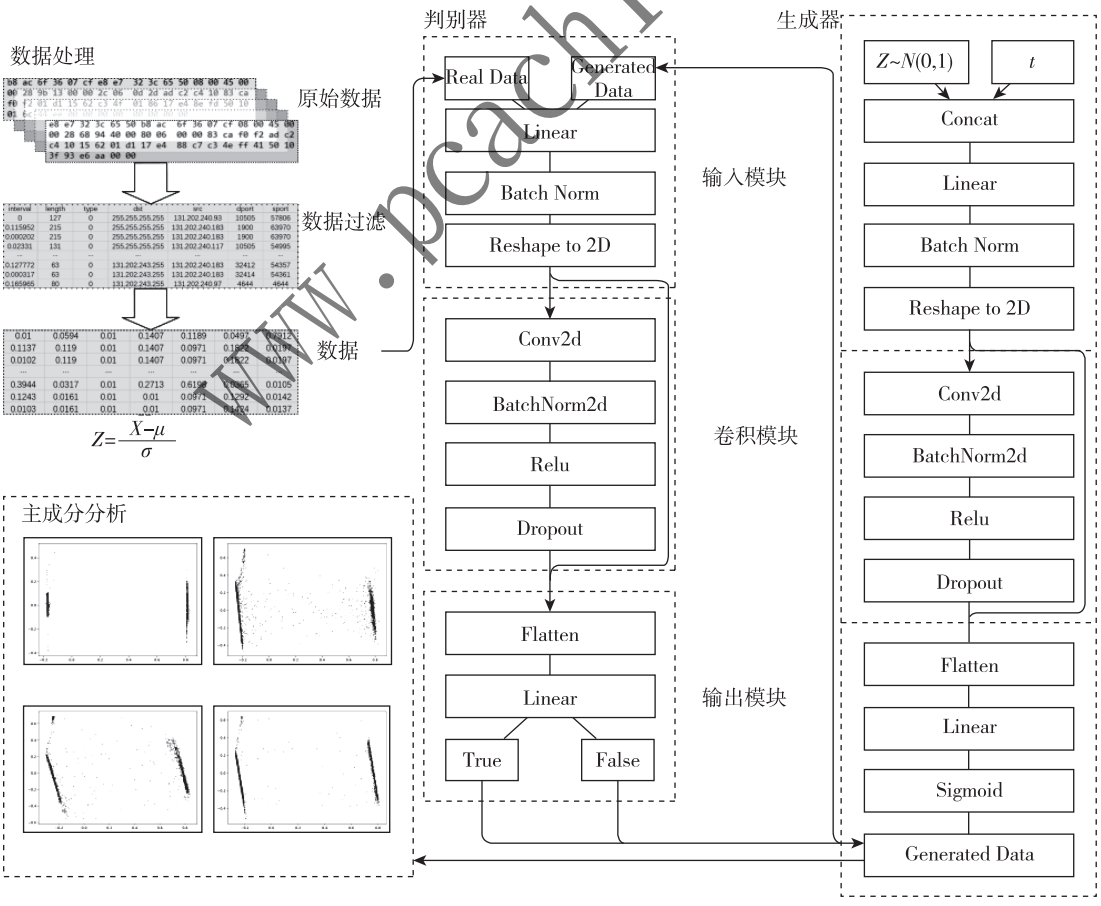

图 2　NTGAN 生成对抗网络流量模型架构图

其中，对于不同的特征 f_i，Z-score 标准化方法可表示为：

$$f'_i = \frac{v_i - \frac{1}{n}\sum_{j=1}^{n} f_j}{\sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(f_i - \sum_{j=1}^{n} f_j)}} \tag{6}$$

MinMax 归一化方法可以表示为：

$$f'_i = \frac{f_i - f_{\min}}{f_{\max} - f_{\min}} \tag{7}$$

其中，$f_{\min}$为特征 f 的最小值，$f_{\max}$为特征 f 的最大值。

3.2 网络流量生成模型

本文提出的流量生成模型分为生成器和判别器。生成器用于从噪声向量 $\boldsymbol{z}$ 和时间向量 $\boldsymbol{t}$ 生成与之相关的网络流量，而判别器用于判定生成流量与真实流量的真伪，进而利用反向传播算法[25]学习网络流量的特征，以改进生成器，让生成器能够生成更加真实的网络流量，以能够骗过判别器，具体如图 2 所示。

3.2.1 网络流量生成器

网络流量生成器由三部分组成，分别是输入模块、卷积模块和输出模块。对于输入模块，本文模型采用了条件机制[26]用于控制模型的时间变量，将随机向量 $\boldsymbol{z}$ 与时间向量 $\boldsymbol{t}$ 进行拼接，得到模型的输入向量 $\boldsymbol{h} = [\boldsymbol{z}; \boldsymbol{t}]$。输入模块由三部分组成：线性层将输入向量映射到规则化的向量；批正则化层用于提高训练的稳定性；Unflatten 层将向量转化为二维向量，得到卷积层的输入向量。对于输入向量 $\boldsymbol{h}_i$ 和输出向量 $\boldsymbol{h}_{i+1}$，其形式化表示如下：

$$\boldsymbol{h}_{i+1} = \text{Unflatten}(\text{BatchNorm}(\text{Linear}(\boldsymbol{h}_i))) \tag{8}$$

卷积模块采用与 DCGAN[27]（Deep Convolutional Generative Adversarial Networks）类似的机制，由五部分组成：卷积层、批正则化层、激活函数、Dropout 层和残差连接。卷积层能有效地捕获数据中的空间信息；批正则化层用于提高训练的稳定性；ReLU 作为激活函数；Dropout 层丢弃一部分权重，防止模型过拟合，再对整个卷积模块添加一个残差连接[28]，用于避免神经网络的梯度消失，其形式化表示如下：

$$\boldsymbol{h}_{i+1} = \text{Dropout}(\text{Relu}(\text{BatchNorm2d}(\text{Conv2d}(\boldsymbol{h}_i)))) + \boldsymbol{h}_i \tag{9}$$

其中 Relu 可表示为：

$$\text{Relu}(z) = \max(0, z) \tag{10}$$

输出模块同样由三部分构成：Flatten 将二维输出展开成一维向量；线性层将向量转化为与输出特征一致；Sigmoid 激活函数将向量映射到（0，1），得到生成的网络流量特征，其形式化表示如下：

$$\boldsymbol{h}_{i+1} = \text{Sigmoid}(\text{Linear}(\text{Flatten}(\boldsymbol{h}_i))) \tag{11}$$

其中 Sigmoid 函数可表示为：

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{12}$$

3.2.2 网络流量判别器

判别器的结构与生成器类似，同样由三部分构成，分别是输入模块、卷积模块和输出模块。区别在于判别器输出的是 Oneshot 编码的向量，只有两个值，分别表示流量真伪的概率。

3.2.3 模型损失函数

本文使用最小均方误差（Mean Squared Error，MSE）作为模型训练的损失函数。MSE 通常在回归问题中用于度量模型预测值与实际值之间的差异，通过计算平方误差的均值来衡量模型的性能，其优势在于对预测误差的敏感性较高。通过平方误差，较大的误差会受到更大的惩罚，这有助于更好地反映模型对异常值的处理能力。MSE 可以表示为：

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{13}$$

其中，n 是样本数量，y_i 是第 i 个样本的真实值，$\hat{y}_i$ 是模型对第 i 个样本的预测值。

3.3 时空相关性的评价

本文采用主成分分析的方法，将真实数据和生成数据通过主成分分析降维到 2 维。以可视化的方式展现真实数据和生成数据之间的差别，能够直观地表示数据特征在降维后的效果。主成分分析通过将数据降维，得到新的数据具有最大的方差，可以很好地体现不同特征经过降维的相关性。

除此以外，为了能够量化时空相关性，受主成分分析的启发，本文提出了网络流量时空相关性的量化方法——网络流量距离（Network Traffic Distance，NTD），将生成网络流量和真实网络流量的特征降维到 1 维，然后使用 NTD 比较二者的距离。NTD 可以表示为：

$$\text{NTD} = \alpha \log_2\left(1 + \frac{1}{n}\sum_{i=1}^{n}\sqrt{(y_i - \hat{y}_i)^2}\right) + (1-\alpha)\left(1 - \frac{\sum_{i=1}^{n} y_i \hat{y}_i}{\sqrt{\sum_{i=1}^{n} y_i^2} \cdot \sqrt{\sum_{i=1}^{n} \hat{y}_i^2}}\right) \tag{14}$$

其中，n 表示样本数量；y_i 表示真实流量降维后的第 i 个值；$\hat{y}_i$ 表示生成流量降维后的第 i 个值；α 表示平滑因子，用于平衡局部特征和全局特征，本文取平滑因子为 0.5。

4 实验分析

为了验证提出的生成模型，本文基于实验环境捕

获的流量数据和公开的流量数据集，对本文提出的模型与常用的三种基线模型进行比较实验。从数据真实性和时空关联性方面对数据进行了比较，结果显示本文的模型在真实性及时空关联性的度量上都具有更好的效果。

4.1 实验准备

本文实验环境如表 1 所示，基于 Python 语言和 PyTorch 深度学习框架实现，使用了 TorchGAN[29] 库，实验平台为 NVIDIA GeForce GTX1660 显卡、Linux Windows 5. 15. 123. 1-microsoftstandard-WSL2 操作系统。处理器为 Intel Core i5-9400，内存 24 GB。神经网络学习率设定为 0. 000 1，批量大小为 128，训练了 200 轮。使用 Adam 优化器，betas = （0. 5，0. 999）。

表 1 实验环境与参数

环境	参数
处理器	Intel Core i5-9400
内存	24 GB
显卡	NVIDIA GeForce GTX1660
操作系统	Linux Windows 5. 15. 123. 1 WSL2
Python	3. 10. 13
PyTorch	2. 0. 1 + cu117

4.1.1 数据集

本文充分验证了所提出的流量生成网络的有效性，分别在不同的网络流量数据集上对提出的模型进行测试。本文数据集部分来自实验模拟，通过模拟日常网络行为（如 DNS 查询、SSH 链接、网络访问等），抓取得到网络数据包，该数据集下文记为 Capture。除此以外，本文还使用了公开数据集 ISCXVPN2016[30]，对其中不同的 .pcap 文件进行了流量生成实验，其中包括：email1a（Email），vpn _ facebook _ audio2 （Facebook），vpn _ ftps（VPN），对应不同的网络应用场景。首先，对数据包依据 3.1 节进行了预处理，获取 10 000 条网络流量数据，然后取其中 7 000 条用于训练，3 000 条用于验证神经网络模型，神经网络经过 200 轮训练之后，利用训练得到的模型生成网络流量特征数据，最后使用生成的特征构造网络数据包。

4.1.2 基线模型分析

为了更好地说明本文模型的性能，本文计算 JSD 度量和 NTD 度量下的距离，并与以下三种典型的模型进行比较：

MLP[31]（Multiple Layer Perceptron）：与本文模型类似，区别在于使用多层感知机作为生成器和判别器；

LSTM[32]（Long Short-Term Memory）：以序列模型的方式，使用过去一段时间的历史数据来预测数据，使用了长短期记忆神经网络，以及一些线性层用于输出结果；

DCGAN[27]：DCGAN 在图像生成领域已取得了显著的成功，广泛应用于人脸生成、风格转换、图像超分辨率等任务，本文使用 DCGAN 作为一种基线模型，构造了与本文提出模型类似的回归模型。

4.2 真实性测试

4.2.1 训练过程

图 3 显示了一次训练过程的损失函数随着训练步数的变化。由于 Adam 优化器在开始时冲量较大，导致生成器的损失和判别器的损失在开始时不稳定。生成器的损失逐渐降低，说明生成器生成的数据逐渐趋于真实，而判别器的损失先降低而后升高，说明判别器一开始很容易就能分辨出数据的真伪，而随着生成器生成数据的改进，判别器出现错误的概率逐渐趋于一个稳定的值，经过 10 000 次以上的迭代，模型达到收敛状态。

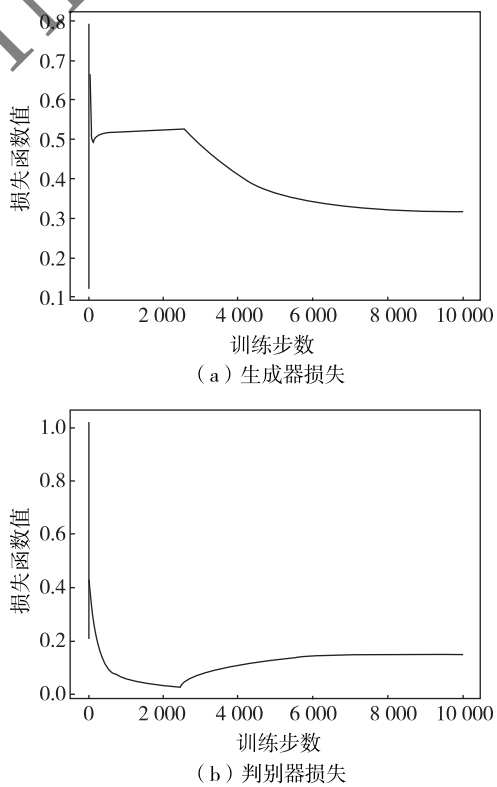

图 3 训练过程损失变化

表 2 展示了在 JSD 度量下不同模型生成网络流量的不同特征的值。实验显示，与基线模型相比，本文模型对多种网络流量特征的 JSD 的结果显著更低，而 JSD 更低表示与真实数据的分布更接近。

表 2　JSD 度量真实数据与生成数据之间的距离

模型	长度	类型	目标地址	源地址	目标端口	源端口
MLP	0.80	0.25	0.43	0.52	0.57	0.67
LSTM	0.80	0.44	0.58	0.54	0.68	0.71
DCGAN	0.74	0.21	0.19	0.21	0.43	0.63
NTGAN	0.72	0.17	0.15	0.17	0.36	0.58

通过主成分分析的方法，得到了生成数据和真实数据的主成分图像，如图 4 所示。实验结果显示，本文提出的方法 NTGAN 的主成分与真实分布有更高的相似性，而 MLP、LSTM、DCGAN 模型生成的流量特征虽然在整体上也有一定的相似性，但在细节上有着很多不足，出现了大量离群点，造成了生成流量特征的不稳定。

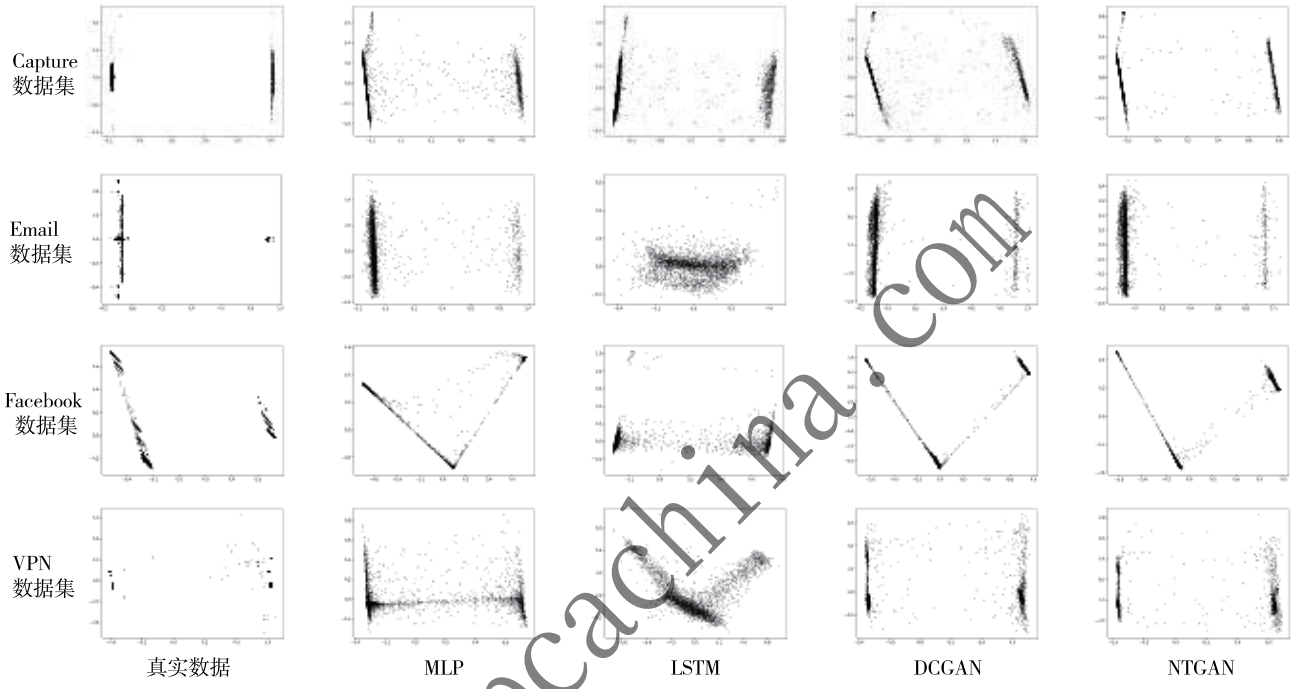

图 4　主成分分析示意图

4.2.2　消融实验

为了充分研究不同机制对流量生成模型带来的变化，本文针对关键的机制进行了组合实验。本文依据 3.3 节给出的方法，首先验证了两种不同的数据预处理方法。在不同模型下进行 3 次实验，如表 3 所示，与 Word2Vec 方法相比，本文的改进取得了更好的效果。

表 3　NTD 和 JSD 度量下不同数据预处理方式的分数

处理方式	NTD	JSD
Word2Vec	2.63	2.94
Word2Vec + 本文改进	2.31	2.65

其次，本文还验证了不同模型和不同数据集生成数据的真实性和相关性，如表 4 所示，实验验证了不同数据集和不同模型在 3 次实验中的平均结果。实验显示除了 LSTM 模型在 Facebook 数据集下 NTD 度量有优势以外，本文方法均取得了更好的效果。

表 4　NTD 和 JSD 度量下不同数据集不同模型的距离

数据集	模型	NTD	JSD
Capture	DCGAN	0.31	2.64
Capture	LSTM	0.35	3.54
Capture	MLP	0.31	3.34
Capture	NTGAN	**0.30**	**2.38**
Email	DCGAN	0.27	2.11
Email	LSTM	0.27	3.72
Email	MLP	0.27	3.15
Email	NTGAN	**0.26**	**1.94**
Facebook	DCGAN	0.39	2.86
Facebook	LSTM	**0.27**	3.66
Facebook	MLP	0.35	3.32
Facebook	NTGAN	0.33	**2.48**
VPN	DCGAN	0.28	2.03
VPN	LSTM	0.38	4.12
VPN	MLP	0.32	3.17
VPN	NTGAN	**0.25**	**1.8**

为了验证 NTGAN 模型的先进性，本文与现有多个网络流量生成模型在 JSD 和 NTD 上使用数据集 ISCX-VPN2016 进行了比较，结果如表 5 所示。可以看出相比于文献［16］和文献［17］中的方法，本文提出的模型在 JSD 和 NTD 度量上有一定的优势，能够更好地生成仿真流量。

表 5　NTD 和 JSD 度量下不同模型的分数

模型	NTD	JSD
文献［16］	2. 47	2. 71
文献［17］	2. 33	2. 67
NTGAN	2. 31	2. 65

5　结束语

针对网络流量生成中时空关联不足的问题，本文提出了一种基于生成式神经网络的网络流量生成模型 NTGAN，解决了生成网络流量时空相关性不足的问题。其次对网络流量数据的预处理做出了改进，使得输入网络流量分布趋于标准正态分布，得到了更好的训练效果。最后对网络流量提出了一种时空相关性的度量方法，可以有效地度量不同网络流量生成模型生成网络流量的相关性。此处降维到 1 维便于距离的计算，对于 2 维及以上的计算，由于计算方式会发生改变，故还有待进一步的研究。本文的研究不仅在网络流量数据预处理方面有所创新，而且通过引入时空关联的生成模型以及相关度量方法，为网络流量生成提供了新的思路和解决方案。

参考文献

［1］ Cisco. Cisco Annual Internet Report（2018 – 2023）［EB/OL］. ［2023 – 09 – 25］. https：//www. cisco. com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490. html.

［2］ SHAUKAT K, LUO S, VARADHARAJAN V, et al. A survey on machine learning techniques for cyber security in the last decade［J］. IEEE Access, 2020, 8：222310 – 222354.

［3］ MOLNÁR S, MEGYESI P, SZABÓ G. How to validate traffic generators? ［C］//Proceedings of 2013 IEEE International Conference on Communications Workshops（ICC）. IEEE, 2013：1340 – 1344.

［4］ Iperf 2 ［EB/OL］. ［2024 – 02 – 15］. https：//sourceforge. net/projects/iperf2/.

［5］ Tcpreplay-Pcap editing and replaying utilities ［EB/OL］. ［2024 – 02 – 07］. https：//tcpreplay. appneta. com/.

［6］ Harpoon：a flow-level traffic generator ［EB/OL］. ［2024 – 02 – 07］. https：//jsommers. github. io/harpoon/.

［7］ EMMERICH P, GALLENMÜLLER S, RAUMER D, et al. Moongen：a scriptable high-speed packet generator ［C］//Proceedings of the 2015 Internet Measurement Conference, 2015：275 – 287.

［8］ Scapy ［EB/OL］. ［2024 – 02 – 09］. https：//scapy. net/.

［9］ GOODFELLOW I J, POUGET-ABADIE J, MIRZA M, et al. Generative adversarial nets ［C］//Proceedings of the 27th International Conference on Neural Information Processing Systems. Cambridge, MA, USA：MIT Press, 2014：2672 – 2680.

［10］ KARRAS T, AILA T, LANINE S, et al. Progressive growing of GANs for improved quality, stability, and variation ［J］. arXiv：1710. 10196, 2017.

［11］ DONAHUE C, McAULEY J, PUCKETTE M. Adversarial audio synthesis ［J］. arXiv. 1802. 04208, 2018.

［12］ BROOKS T, HELLSTEN J, AITTALA M, et al. Generating long videos of dynamic scenes ［J］. Advances in Neural Information Processing Systems, 2022, 35：31769 – 31781.

［13］ RIGAKI M, GARCIA S. Bringing a GAN to a knife-fight：adapting malware communication to avoid detection ［C］//Proceedings of 2018 IEEE Security and Privacy Workshops（SPW）. IEEE, 2018：70 – 75.

［14］ CHENG A. PAC-GAN：packet generation of network traffic using generative adversarial networks ［C］//Proceedings of 2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference（IEMCON）. IEEE, 2019：728 – 734.

［15］ BROWNLEE J. Data preparation for machine learning：data cleaning, feature selection, and data transforms in Python ［M］. Machine Learning Mastery, 2020.

［16］ RING M, SCHLÖR D, LANDES D, et al. Flow-based network traffic generation using generative adversarial networks ［J］. Computers & Security, 2019, 82：156 – 172.

［17］ HUI S D, WANG H D, WANG Z H, et al. Knowledge enhanced GAN for IoT traffic generation ［C］//Proceedings of the ACM Web Conference, 2022：3336 – 3346.

［18］ RING M, WUNDERLICH S, GRÜDL D, et al. Flow-based benchmark data sets for intrusion detection ［C］//Proceedings of the 16th European Conference on Cyber Warfare and Security, 2017：361 – 369.

［19］ DOWOO B, JUNG Y, CHOI C. PcapGAN：packet capture file generator by style-based generative adversarial networks ［C］//Proceedings of 2019 18th IEEE International Conference on Machine Learning and Applications（ICMLA）. IEEE, 2019：1149 – 1154.

［20］ LIN Z, JAIN A, WANG C, et al. Using GANs for sharing networked time series data：challenges, initial promise, and open questions ［C］//Proceedings of the ACM Internet Measurement Conference. USA：ACM, 2020：464 – 483.

［21］ SHAHID M R, BLANC G, JMILA H, et al. Generative deep

learning for Internet of Things network traffic generation [C]// Proceedings of 2020 IEEE 25th Pacific Rim International Symposium on Dependable Computing (PRDC). IEEE, 2020: 70 - 79.

[22] 胡永进, 郭渊博, 马骏, 等. 基于对抗样本的网络欺骗流量生成方法 [J]. 通信学报, 2020, 41 (9): 59 - 70.

[23] JOLLIFFE I T. Principal component analysis for special types of data [M]. New York: Springer-Verlag, 2002.

[24] MIKOLOV T, CHEN K, CORRADO G, et al. Efficient estimation of word representations in vector space [J]. arXiv: 1301.3781, 2013.

[25] RUMELHART D E, HINTON G E, WILLIAMS R J. Learning representations by back-propagating errors [J]. Nature, 1986, 323 (6088): 533 - 536.

[26] MIRZA M, OSINDERO S. Conditional generative adversarial nets [J]. arXiv preprint arXiv: 1411.1784, 2014.

[27] RADFORD A, METZ L, CHINTALA S. Unsupervised representation learning with deep convolutional generative adversarial networks [J]. arXiv: 1511.06434, 2015.

[28] HE K M, ZHANG X Y, REN S Q, et al. Deep residual learning for image recognition [C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016: 770 - 778.

[29] PAL A, DAS A. TorchGAN: a flexible framework for GAN training and evaluation [J]. Journal of Open Source Software, 2021, 6 (66): 2606.

[30] LASHKARI A H, DRAPER-GIL G, MAMUN M S I, et al. Characterization of encrypted and VPN traffic using time-related features [C]//Proceedings of the 2nd International Conference on Information Systems Security and Privacy (ICISSP), 2016: 407 - 414.

[31] HAYKIN S. Neural networks: a comprehensive foundation [M]. Prentice Hall PTR, 1998.

[32] HOCHREITER S, SCHMIDHUBER J. Long short-term memory [J]. Neural Computation, 1997, 9 (8): 1735 - 1780.

(收稿日期: 2024 - 04 - 02)

作者简介:

康未 (1996 -), 男, 硕士研究生, 主要研究方向: 网络仿真、流量生成。

李维皓 (1990 -), 女, 博士, 高级工程师, 主要研究方向: 网络仿真、综合效能评估、隐私计算。

刘桐菊 (1978 -), 女, 硕士, 工程师, 主要研究方向: 网络仿真、态势感知、机器翻译。

版权声明

凡《网络安全与数据治理》录用的文章，如作者没有关于汇编权、翻译权、印刷权及电子版的复制权、信息网络传播权与发行权等版权的特殊声明，即视作该文章署名作者同意将该文章的汇编权、翻译权、印刷权及电子版的复制权、信息网络传播权与发行权授予本刊，本刊有权授权本刊合作数据库、合作媒体等合作伙伴使用。同时，本刊支付的稿酬已包含上述使用的费用，特此声明。

《网络安全与数据治理》编辑部