Computer Science and Application 计算机科学与应用, 2025, 15(4), 406-415
Published Online April 2025 in Hans. https://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2025.154113

Hans汉斯

基于LSTM-DDPG的网络入侵检测方法研究

王国栋，姜　伟

哈尔滨师范大学计算机科学与信息工程学院，黑龙江　哈尔滨

收稿日期：2025年3月25日；录用日期：2025年4月23日；发布日期：2025年4月30日

摘　要

针对传统入侵检测系统在动态环境下时序特征捕捉不足、小样本攻击检测效果差的问题，本文提出基于LSTM-DDPG的入侵检测方法。通过将长短期记忆网络(LSTM)融入深度确定性策略梯度(DDPG)框架，构建具备时序建模与动态策略优化能力的检测模型。结合TON-IoT数据集进行实验验证。实验表明，融合模型较单一DDPG和LSTM在准确率(+13.07%/+21.58%)、精确率(+34.75%/+9.55%)、召回率(+29.43%/+99.13%)及F1值(+31.89%/+49.93%)上均显著提升，其中小样本攻击MITM的召回率提升3.29%。该方法验证了时序特征与强化学习融合的有效性，为动态网络安全防护提供新思路，未来将重点优化模型在小样本与大样本检测中的平衡性。

关键词

网络入侵检测，LSTM，DDPG，深度强化学习，时序数据处理，动态检测

Research on Network Intrusion Detection Method Based on LSTM-DDPG

Guodong Wang, Wei Jiang

College of Computer Science and Information Engineering, Harbin Normal University, Harbin Heilongjiang

Received: Mar. 25th, 2025; accepted: Apr. 23rd, 2025; published: Apr. 30th, 2025

Abstract

Aiming at the problems that the traditional intrusion detection system lacks time series feature

文章引用: 王国栋, 姜伟. 基于 LSTM-DDPG 的网络入侵检测方法研究[J]. 计算机科学与应用, 2025, 15(4): 406-415.
DOI: 10.12677/csa.2025.154113

capture and the detection effect of small sample attack is poor in dynamic environment, this paper proposes an intrusion detection method based on LSTM-DDPG. By integrating Long Short-Term Memory (LSTM) network into the Deep Deterministic Policy Gradient (DDPG) framework, a detection model with the ability of time series modeling and dynamic policy optimization was constructed. The TON-IoT dataset was used for experimental verification. The experimental results show that the fusion model significantly improves the accuracy (+13.07%/+21.58%), precision (+34.75%/+9.55%), recall (+29.43%/+99.13%) and F1 value (+31.89%/+49.93%) compared with single DDPG and LSTM. The recall rate of small sample attack MITM is increased by 3.29%. This method verifies the effectiveness of the fusion of time series features and reinforcement learning, and provides new ideas for dynamic network security protection. In the future, the balance between small sample and large sample detection of the model will be optimized.

Keywords

Network Intrusion Detection, LSTM, DDPG, Deep Reinforcement Learning, Temporal Data Processing, Dynamic Detection

1. 引言

入侵检测系统(Intrusion Detection System, IDS)作为网络安全的关键组件，通过监控网络或系统活动，识别潜在的攻击行为，从而及时采取措施防止或减轻安全威胁。然而，随着人工智能、大数据和云计算技术的快速发展，网络攻击手段日益复杂，传统的基于特定数据模式或行为规则的入侵检测方法已难以适应这种快速演变的攻击环境。

本文旨在探讨基于深度强化学习模型提升入侵检测系统(IDS)准确性与适应性的有效途径。为此，本文选用了深度确定性策略梯度(DDPG)网络，针对包含大量离散动作的离散－连续混合动作空间环境，并结合网络环境的时序特性，在 DDPG 智能体中融入长短期记忆网络(LSTM)以捕捉序列数据中的长期依赖关系。该方法融合了 LSTM 在时序数据处理中的优势与 DDPG 在策略优化中的卓越性能，从而显著提升了系统在网络入侵检测等时序数据驱动任务中的表现。

通过 LSTM 层，网络能够有效地捕捉数据的时间依赖性，而 DDPG 则能够基于这些信息学习到最优的入侵检测策略。与单独采用 DDPG 或 LSTM，LSTM 与 DDPG 的集成不仅充分发挥了二者各自的优势，还在动态网络环境中展现出更高效的入侵检测能力。该方法利用 LSTM 捕捉网络流量数据中的时序依赖关系，并借助 DDPG 在策略优化中的卓越性能，实现了对网络入侵行为的实时、精准识别，为网络安全防护提供了创新而有效的解决方案。

深度学习能够自动从原始网络数据中提取深层次特征，克服了传统方法中对手工特征设计的依赖，从而更准确地捕捉隐藏的异常模式。麻文刚[1]等人提出了一种基于 LSTM 和 ResNet 的入侵检测模型，该模型综合了 STM 和 ResNet 的优点，有效改善了深度网络中的过拟合问题，在 NSL-KDD 数据集上实现了约 90%的准确率。Kober J [2]等人将检测问题转化为预测马尔可夫奖励过程的价值函数问题，采用了线性基函数的时间差异算法进行值预测，从而能够准确地预测主机过程的异常时间行为。Xu X, Xie T [3]等人提出了一种分布式传感器和决策代理的体系结构，旨在改善在网络传感器代理的分层架构中分布式强化学习方法的模型精准率较差的问题。Di C [4]等人研究了分布式强化学习对入侵响应的适用性，然而，

他们发现系统无法仅通过考虑流量来区分合法流量和攻击流量。Servin A [5] [6]等人提出了一种基于神经网络的强化学习对抗性环境算法，首次将对抗强化学习应用于入侵检测，并将环境行为融入到改进的强化学习算法的学习过程中。他们将入侵检测中的术语与深度强化学习中的术语一一对应，该模型集成了强化和监督框架，产生的环境能够与通过网络特征和相关入侵标签形成的预先记录的样本数据集进行交互，并且选择具有优化策略的样本以实现最佳分类效果。Cao H [7]结合了 GAN 和变压器的优点，研究目标是建立智能检测系统。

2. 相关理论

2.1. 深度确定性策略梯度网络(DDPG)

DDPG (Deep Deterministic Policy Gradient)算法是将确定性策略与经验回放相结合的强化学习方法。其学习框架如图 1 所示。

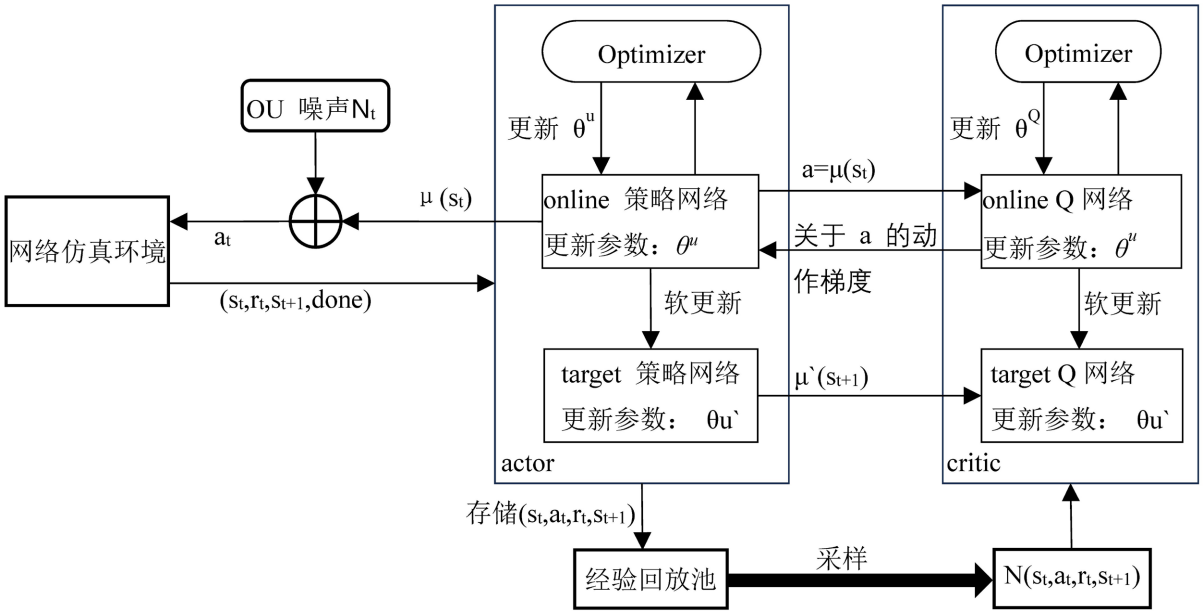

Figure 1. DDPG algorithm learning framework
图 1. DDPG 算法学习框架

在该框架中，*si* 表示状态 *i*。*ai* 是系统基于当前状态 *si* 作出的确定性决策动作，而非随机探索动作。done 是一个布尔变量，指示当前状态是否为终端状态。*ri* 表示状态 *i* 对应的奖励。在入侵检测的具体应用中，动作 *ai* 可能包括但不限于发出警报、特定的 IP 地址或调整防火墙规则等。

DDPG 算法是为连续动作空间设计的，其架构可以通过调整输出层来适应离散动作空间。对于入侵检测系统而言，DDPG 在处理复杂的决策过程时能够学习到一套策略，以应对网络入侵检测中涉及的大量动态变化因素。特别是，在需要实时适应新威胁和攻击手段的网络入侵检测系统中，DDPG 提供了一种能够在与环境交互的过程中不断更新策略的在线学习算法，这使得其具有很高的应用价值。

2.2. 长短期记忆网络(LSTM)

长短期记忆网络(LSTM, Long Short-Term Memory)是一种能够处理和预测时间序列数据中长期依赖关系的递归神经网络。其算法框架如图 2 所示。

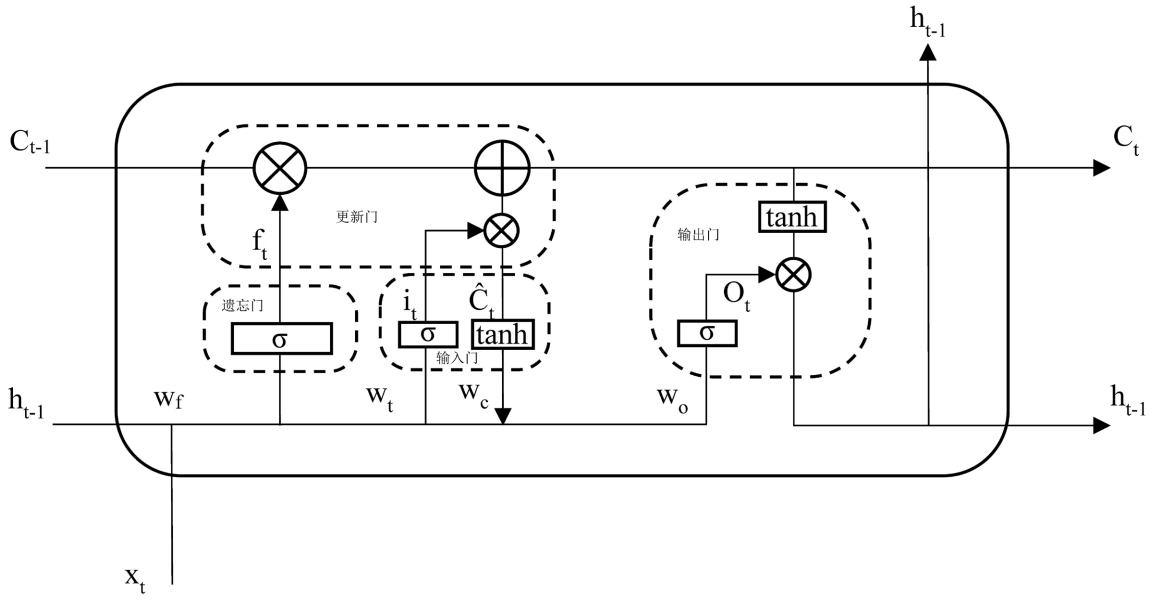

Figure 2. LSTM cell algorithm framework
图 2. LSTM 细胞算法框架

在 LSTM 中，x_t 表示输入，σ 表示 sigmod 函数，tanh 表示双曲正切函数，$f_t, i_t, \hat{C}_t, o_t$ 分别表示遗忘门、输入门、当前储存单元的候选者和输出门的运算结果，c_t 表示更新后的单元状态，h_t 是隐藏层的状态，w_f, w_t, w_c, w_o 分别表示对应部分的权重值。其逻辑表达式表示为：

$$\begin{cases} c_t = f_t * c_{t-1} + i_t * \hat{C}_t \\ h_t = o_t * \tanh(c_t) \end{cases} \tag{1}$$

其中，

$$\begin{cases} f_t = \sigma\left(w_t \cdot [h_{t-1}, x_t] + b_f\right) \\ i_t = \sigma\left(w_i \cdot [h_{t-1}, x_t] + b_i\right) \\ \hat{C}_t = \tanh\left(w_c \cdot [h_{t-1}, x_t] + b_c\right) \\ o_t = \sigma\left(w_o \cdot [h_{t-1}, x_t] + b_o\right) \end{cases} \tag{2}$$

LSTM 的核心为记忆细胞，即图 2 中 c_{t-1} 到 c_t 这一条贯穿顶部的水平线。LSTM 可以往记忆细胞之中添加或者移除信息，LSTM 把训练信息分成长期信息 c_t 和短期信息 h_t，c_t 可以不受影响地继续传递下去。或者移除信息是由遗忘门、输入门、输出门这三种门结构控制。遗忘门通过 sigmoid 激活函数来决定对长期信息的保留和删除。输入门通过 sigmoid 和 tanh 双曲正切函数决定该往长期信息中添加什么样的信息。输出门则通过 sigmoid 层决定输出信息，然后通过 tanh 层激活长期信息及点乘前者输出来得到当前时刻的预测值 h_t。

3. LSTM-DDPG 的算法设计

传统 DDPG 算法在策略更新过程中易遭遇不稳定性和收敛性问题，尤其在网络环境剧烈波动时，其在快速响应网络入侵检测任务中的效能受到明显制约。该算法通常依赖大量数据作为决策基础，而在复杂网络环境中这一条件往往难以满足；在平衡误报与漏报方面也存在不足，致使实际应用难以达到预期的安全标准。

为了克服这些挑战，本文将长短期记忆网络(LSTM)集成到 Actor 和 Critic 网络的智能体中。改进的算法网络结构如图 3 所示。LSTM 被用于 Actor 和 Critic 网络中。Actor 网络通过 LSTM 层提取时间序列特征，生成动作。Critic 网络则结合状态和动作，通过 LSTM 层评估动作的价值。增强了网络对时间序列数据的处理能力，使得 Actor 和 Critic 网络能够更好地捕捉网络流量中的长期依赖关系，帮助识别复杂的攻击模式。这种改进在鲁棒性方面通过 LSTM 能够捕捉长时间依赖关系，使得模型在面对长时间的攻击行为时仍能保持稳定的性能。DDPG 通过策略网络和价值网络的交替优化，能够在动态环境中保持较高的策略稳定性。通过经验回放机制，模型能够在不同的时间步中学习到更广泛的策略，提高对环境变化的适应能力。算法在网络入侵检测任务中更好地平衡误报和漏报。模型可行性在于模型通过分析 LSTM 的输出，可以识别哪些时间序列特征对决策有重要影响。通过 DDPG 的策略网络输出，可以观察到模型在不同状态下的决策变化。

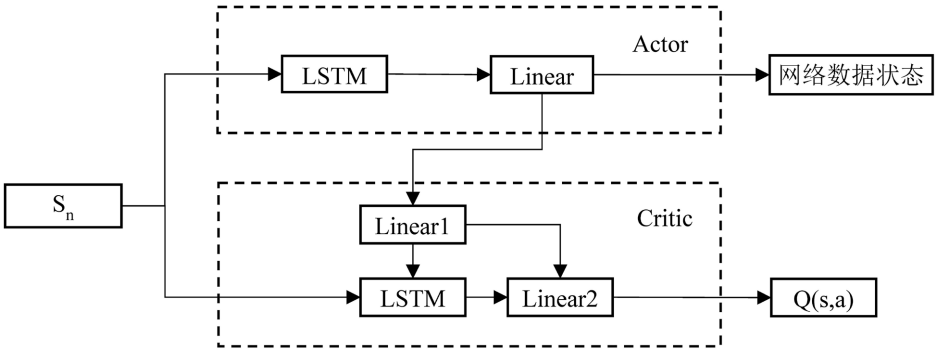

Figure 3. LSTM-DDPG algorithm improvement
图 3. LSTM-DDPG 算法改进部分

攻击行为在动态网络中呈现多尺度时序特性，双层 LSTM 的设计有助于捕获网络流量特征中的长期依赖关系和复杂的时序模式，首层 LSTM 提取基础时间模式，第二层捕获高阶关联特征。因此在图 3 所示的网络结构中，LSTM 由两层 LSTM 单元组成，旨在增强系统的实时性和提高网络的鲁棒性。Linear、Linear1 和 Linear2 分别代表一层全连接网络。Actor 和 Critic 网络的隐藏层如上图所示。隐藏层之间使用 ReLU 函数作为激活函数，以避免梯度消失问题。输出层使用 tanh 作为激活函数，将输出限制在一定范围内。

在学习阶段，Critic 网络在评分时，首先将 Actor 网络的输出通过 Linear1 处理，然后与当前状态合并作为输入源输入到 LSTM 网络中。最终，LSTM 网络的输出与 Linear1 处理结果合并传递给 Linear2，输出单一的结果，表示动作价值函数。

为有效地引导模型学习正确识别不同类型的网络流量 LSTM-DDPG 算法的奖励机制通过差异敏感的奖励计算和类别权重的设置，基于 Actor 网络输出的动作是否被判定为攻击类别流量。具体而言，对于正确检测出的动作结果，给予相应的系数，然后通过计算 Actor 网络输出与真实标签之间的差值与系数的乘积，作为奖励 r 的给予方式。整个 LSTM-DDPG 的伪代码如下所示：

LSTM-DDPG 的伪代码

θ^{Q} 和 θ^{μ} 随机初始化 Critic 网络 $Q(s, a|\theta^{Q})$和 Actor 网络 $\mu(s, a|\theta^{\mu})$

$\theta^{Q'} \leftarrow \theta^{Q}, \theta^{\mu'} \leftarrow \theta^{\mu}$ 初始化目标网络权重参数 Q'和 μ'

初始化经验回放区 R

续表

for episode = 1, M do:

行动探索，随机噪声 N 初始化

获得初始观察状态 s_1

for t = 1, T do:

$$a_t = \mu\left(s_t,\ a \mid \theta^{\mu}\right) + N_t$$

根据动作 a_t 获取奖励 r_t 和环境状态 s_{t+1} 将数据(s_t, a_t, r_t, s_{t+1})存入 R

从 R 中随机采样批量数目值 N 的多位数组(s_i, a_i, r_i, s_{i+1})

$$y_i = r_i + \gamma Q'\left(s_{i+1}, \mu'\left(s_{i+1} \mid \theta^{\mu'}\right) \mid \theta^{Q'}\right)$$

最小化损失函数 L 来更新 Critic 网络：

$$Loss = \frac{1}{N}\sum_i \left(y_i - Q\left(s_i,\ a_i \mid \theta^{Q}\right)\right)^2$$

采样策略梯度更新 Actor 策略网络：

$$\nabla_{\theta\mu} J\left(\theta\mu\right) \cong \frac{1}{N}\sum_i \nabla_{\theta\mu}\mu\left(s_t \mid \theta^{\mu}\right)\nabla_a QQ\left(s_i,\ a \mid \theta^{Q}\right)|_{a=\mu(s_i)}$$

更新目标网络：

$$\theta^{Q'} \leftarrow \tau\theta^{Q} + \left(1-\tau\right)\theta^{Q'}$$

$$\theta^{\mu'} \leftarrow \tau\theta^{\mu} + \left(1-\tau\right)\theta^{\mu'}$$

end for

end for

4. 实验与结果分析

实验的设计是使用 Python 语言实现的，具体实验环境如表 1 所示。

Table 1. Experimental environment configuration
表 1. 实验环境配置

环境	参数
操作系统	Windows 11 专业版
CPU	AMD Ryzen 9 7940H w/ Radeon 780M Graphics 4.00 GHz
GPU	NVIDIA GeForce RTX 4060 Laptop GPU
内存	DDR5 32.0 GB
Python	3.9.7
Pytorch	2.4.0 + cu121

4.1. 实验数据来源

本研究采用 TON-IoT 数据集作为实验基础，该数据集专为物联网(IoT)环境设计，旨在评估人工智能安全应用的性能。通过模拟真实 IoT 环境，它为研究人员提供了丰富的测试平台，推动了物联网安全研究。TON-IoT 涵盖 DDoS、DoS、侦察等常见网络攻击如表 2 所示，使研究人员能够深入分析 IoT 安全漏

洞，并探索有效防御机制，在物联网安全领域具有重要应用价值。

Table 2. Category and quantity distribution of training set and test set
表 2. 训练集与测试集的类别与数量分布

样本类型	训练集	训练集总数	测试集	测试集总数
backdoor	5755		12,956	
ddos	5903		14,090	
dos	5690		13,302	
injection	5913	44,450	14,051	
mitm	328		713	103,984
password	5972		13,889	
ransomware	4492		10,243	
scanning	5848		14,152	
xss	4549		10,588	
normal	12,692	12,692	29,348	29,348
样本总数	57,142	57,142	133,332	133,332

4.2. 数据预处理

数据集包含 10 种不同的攻击类别。本文将数据集随机打乱后按照 7:3 的比例划分为训练集和测试集并进行实验。针对数据集中存在的缺失值可能导致模型训练偏差的问题，本研究通过系统性数据清洗流程对不完整样本进行删除处理，为后续实验进行提供了有效性。基于详尽的攻击模式统计分布图谱，构建了标准化的分类编码框架。为确保研究方法的可溯源性并支持后续分析的复现性，最终形成的编码映射关系及各类别频次分布数据被完整记录于专用元数据存储库，该架构设计实现了分类体系的可视化追踪与样本分布特征的量化呈现。

为了处理这种数据分布不均匀的问题，本文采用 Z-score 标准化方法对数据进行预处理。Z-score 标准化方法的数学表达式如下所示：

$$z = \frac{x - \mu}{\sigma} \tag{3}$$

其中，x 是原始数据点，μ 是数据集的均值，σ 是数据集的标准差。Z-score 标准化方法通过将原始数据点减去均值后除以标准差，将数据转换为标准正态分布，从而使得不同特征之间的数值范围一致，有利于模型的训练和性能提升。

4.3. 评价指标

在本研究中，受样本类别不平衡的影响，准确率作为评价指标的可靠性受到质疑，模型可能在多数类别上表现优异，而在少数类别上效果欠佳。为全面评估模型性能，本文采用了多种评价指标，包括准确率(*ACC*)、精确率(*Precision*)、召回率(*Recall*)和 *F*1 分数，以提供更客观、公正的评估。这些指标的公式表达如下：

准确率(*ACC*)：

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

精确率(*Pre*)：

$$Pre = \frac{TP}{TP + FP} \tag{5}$$

召回率(*Recall*)：

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

*F*1 分数(*F*1)：

$$F1 = \frac{2 * Pre * Recall}{Pre + Recall} \tag{7}$$

其中，*TP* 表示真正例，*TN* 表示真负例，*FP* 表示假正例，*FN* 表示假负例。准确率反映了模型正确分类的总体比例，精确率反映了模型正确预测为正例的比例，召回率反映了模型正确识别出所有正例的比例，*F*1 分数是精确率和召回率的调和平均数。

4.4. 实验分析

对于提出的 LSTM-DDPG 模型，设定了奖励衰减系数为 0.9，软更新系数为 0.01。此外，动作和评论家的学习率分别被设置为 0.001 和 0.002。经验池的容量被设定为 3000，每次训练的样本数为 64，当经验池的容量达到 300 时，训练开始。隐含神经元的数量被设为 140。在优化算法的选择上，本文采用了 Adam 优化函数，它是一种基于梯度的一阶优化算法，相较于其他优化算法，具有更快的收敛速度和较低的计算复杂度。

为了验证改进后模型的有效性，本实验设计了 2 组消融实验，分别测试了基准模型 DDPG 和 LSTM。实验均在 100 个 Epoch 下进行训练，其中，我们选取了 50 次训练中的最终结果进行比较。基准模型与本文模型在实验集中的性能比较结果如表 3 所示，而在验证集中的性能比较结果如表 4 所示。

Table 3. Performance comparison between the benchmark model and the model presented in this paper in the experimental set (%)
表 3. 基准模型与本文模型在实验集中的性能比较(%)

模型	准确率	精确率	召回率	*F*1 值
LSTM	93.95	98.04	46.84	63.4
DDPG	74.28	79.54	87.91	83.52
LSTM-DDPG	94.93	98.55	93.53	95.97

Table 4. Performance comparison between the benchmark model and the model presented in this paper in the verification set (%)
表 4. 基准模型与本文模型在验证集中的性能比较(%)

模型	准确率	精确率	召回率	*F*1 值
LSTM	76.68	90.25	42.60	60.90
DDPG	82.45	73.37	65.54	69.23
LSTM-DDPG	93.23	98.87	84.83	91.31

在表 3 和表 4 所示的实验环境下，最优模型的性能进行了比较。结果显示，尽管 LSTM 在验证集上保持了良好的效果，但在准确率方面并不高。然而，将 LSTM 融合进 DDPG 网络后，模型的性能得到了

稳定的提升。与普通 DDPG 和 LSTM 相比，融合后的模型在准确率方面分别提升了 13.07%和 21.58%；精确率分别提升了 34.75%和 9.55%；召回值分别提升了 29.43%和 99.13%；*F*1 值分别提升了 31.89%和 49.93%。其训练过程准确率对比图如图 4 所示，测试过程准确率对比图如图 5 所示。这些结果表明，LSTM 与 DDPG 网络的融合在提高模型性能方面具有显著优势。

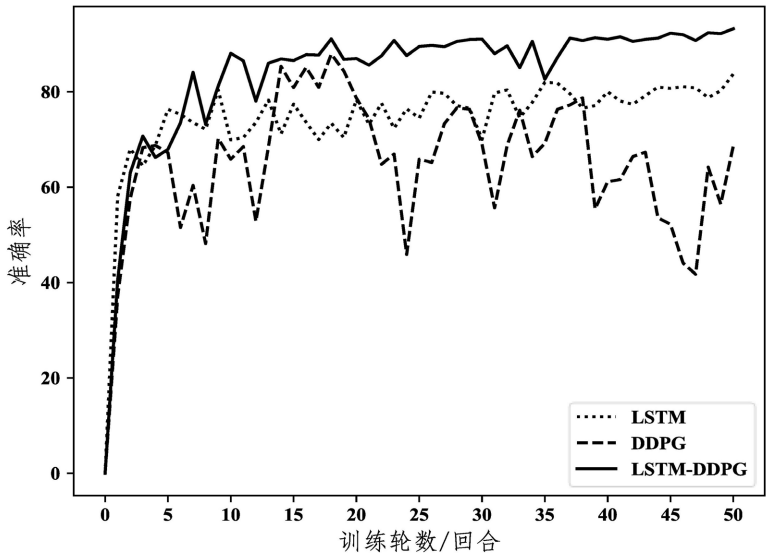

Figure 4. Comparison of accuracy of training process
图 4. 训练过程准确率对比图

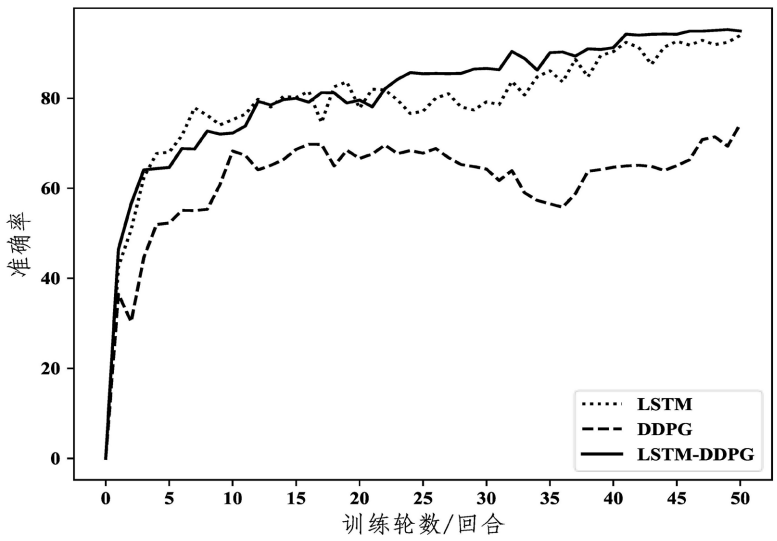

Figure 5. Comparison of accuracy of test process
图 5. 测试过程准确率对比图

表 5 展示了普通 DDPG 和 LSTM 与结合后的 LSTM-DDPG 网络在不同攻击类别上的精确率、召回率和 *F*1 值的检测结果对比。实验结果表明，融合改进后的网络在不同攻击类别的检测能力上均有所提升，尤其在少样本网络攻击模式的识别方面，相较于传统 DDPG 网络，融合模型展现出更强的检测能力。这一结果进一步验证了 LSTM 与 DDPG 融合在提升模型性能方面的有效性与可行性。

Table 5. Test results of other categories (%)
表 5. 其他类别检测结果(%)

模型	指标	类别								
		backdoor	ddos	dos	injection	mitm	password	ransomware	scanning	xss
LSTM	精确率	27.27	85.06	64.91	91.91	55.79	78.62	98.83	45.1	91.69
	召回率	0.27	30.94	35.15	26.25	1.24	27.09	22.46	47.59	14.78
	*F*1 值	0.53	45.38	45.61	40.84	2.43	40.3	36.6	46.31	25.46
DDPG	精确率	98.58	95.82	99.91	91.44	0	94.30	93.53	97.87	88.83
	召回率	50.83	42.22	49.15	53.33	0	53.05	45.67	51.41	43.41
	*F*1 值	67.08	58.61	65.89	67.37	0	67.90	61.37	67.41	58.32
LSTM-DDPG	精确率	99.64	95.90	99.90	94.01	32.98	96.36	96.36	97.49	82.40
	召回率	65.07	66.57	63.72	68.20	3.29	60.40	60.40	66.67	66.82
	*F*1 值	78.72	78.59	77.81	79.05	5.99	74.26	74.26	79.19	73.80

5. 结论

随着网络环境的快速演进，异常行为与恶意攻击的检测愈发关键。面对日益增长且形式多样的网络攻击，亟需一种能够适应动态变化的高效检测工具。本文提出了一种融合 DDPG 网络与 LSTM 优势的新方法，以实现对网络环境的动态监测。实验结果表明，该融合模型在检测少数类别的 MITM 攻击方面表现优异，并显著提升了整体的精确率、召回率和 F1 值。

本文提出的方法为动态检测网络环境提供了有益的探索与参考。未来将进一步拓展融合模型至更多网络数据集，以验证其可靠性与泛化能力。重点关注在提升对少量小样本检测能力的同时，确保对大样本的识别性能不受影响，从而进一步优化模型的实用性与稳定性。

参考文献

[1] 麻文刚, 张亚东, 郭进. 基于 LSTM 与改进残差网络优化的异常流量检测方法[J]. 通信学报, 2021, 42(5): 23-40.

[2] Kober, J., Bagnell, J.A. and Peters, J. (2013) Reinforcement Learning in Robotics: A Survey. *The International Journal of Robotics Research*, **32**, 1238-1274. https://doi.org/10.1177/0278364913495721

[3] Xu, X. and Xie, T. (2005) A Reinforcement Learning Approach for Host-Based Intrusion Detection Using Sequences of System Calls. In: Huang, D.S., Zhang, X.P. and Huang, G.B., Eds., *Advances in Intelligent Computing*, Springer, 995-1003. https://doi.org/10.1007/11538059_103

[4] Di, C., Su, Y., Han, Z. and Li, S. (2018) Learning Automata Based SVM for Intrusion Detection. In: Liang, Q., Mu, J., Jia, M., Wang, W., Feng, X. and Zhang, B., Eds., *Communications*, *Signal Processing*, *and Systems*, Springer, 2067-2074. https://doi.org/10.1007/978-981-10-6571-2_252

[5] Servin, A. and Kudenko, D. (2008) Multi-Agent Reinforcement Learning for Intrusion Detection. In: Tuyls, K., Nowe, A., Guessoum, Z. and Kudenko, D., Eds., *Adaptive Agents and Multi-Agent Systems III*, Springer, 211-223. https://doi.org/10.1007/978-3-540-77949-0_15

[6] Servin, A. and Kudenko, D. (2008) Multi-Agent Reinforcement Learning for Intrusion Detection: A Case Study and Evaluation. In: Bergmann, R., Lindemann, G., Kirn, S. and Pěchouček, M., Eds., *Multiagent System Technologies*, Springer, 159-170. https://doi.org/10.1007/978-3-540-87805-6_15

[7] Cao, H. (2024) The Detection of Abnormal Behavior by Artificial Intelligence Algorithms under Network Security. *IEEE Access*, **12**, 118605-118617. https://doi.org/10.1109/access.2024.3436541