SRAIOT：基于软件定义网络与集成学习的物联网安全路由算法复现报告

周睿敏

December 31, 2025

Abstract

本文详细记录了对论文《Secure routing in the Internet of Things (IoT) with intrusion detection capability based on software-defined networking (SDN) and Machine Learning techniques》的完整复现过程。SRAIOT 算法旨在解决物联网环境下的路由安全与链路稳定性问题。本文通过 Python 实现了从 IDS 模块训练到大规模网络仿真的全过程。实验结果显示，虽然在部分机器学习指标上由于数据集规模和随机性与原文存在细微偏差，但整体网络性能趋势（PDR、能耗、延迟）与原文高度吻合，验证了 SRAIOT 算法在分层防御和负载均衡方面的有效性。

Contents

1 论文原理深度解析

SRAIOT 的核心竞争力在于将物理层的移动性预测与控制器层的智能安全决策深度融合工作。

1.1 连接稳定性预测 (T_{ij})

为了在动态移动的物联网环境中维持稳定的簇结构，SRAIOT 引入了连接时长预测公式。公式 (1) 通过节点间的相对位置和速度矢量，计算出链路失效的临界时间：

$$T_{ij} = \frac{d \cdot \cos(\phi_{ij}) + \sqrt{r^2 - d^2 \sin^2(\phi_{ij})}}{v_{ij}} \tag{1}$$

这一设计的精妙之处在于，它不仅考虑了当前的距离 d，还考虑了运动趋势。只有预测在 Δt 时间内不会断连的节点，才会被 SDN 控制器选入同一个子网（Subdomain）。这种“预见性”的聚类方案是 SRAIOT 能够降低路由重传和能耗的根本原因。

1.2 路由权值与负载均衡 (W_{ij})

原文通过公式 (2) 定义了链路权重，其本质是一个多目标优化函数：

$$W_{ij} = \left(\frac{C_j \times D_j}{E_j} \right) \tag{2}$$

其中，C_j（拥塞度）是通过公式 (3) 的服务速率比计算得出的：$C_i = T_{service}/T_{arrival}$。这意味着路由算法会主动避开高负载节点，将流量导向资源充沛（高 E_j）且链路稳定（低 D_j）的区域。

1.3 集成学习 (EL) 的逻辑闭环

SRAIOT 在控制器中部署了 KNN、SVM 和 ANN 三种模型。论文强调集成学习是为了“cover the minor error of each classifier”。通过多数投票机制：若至少两个模型报警，则判定为攻击。这种设计能有效降低单一模型因特征空间覆盖不足导致的误报或漏报。

2 复现方案设计

复现方案严格按照论文的四个阶段进行设计：

2.1 第一阶段：IDS 模块开发

目标是构建一个能够识别 4 种主要网络攻击（DoS, U2R, R2L, Probe）的检测引擎。

- **数据集**: NSL-KDD。
- **预处理**: 标签二值化、类别特征 Label Encoding、数值特征 Min-Max Scaling。
- **模型**: 还原 KNN ($K = 5$)、线性 SVM、单隐藏层 ANN。

2.2 第二阶段：物联网环境仿真

构建一个支持节点移动和能量消耗的虚拟物理环境。

- **区域**: 500×500 m 矩形区域。
- **移动性**: 随机正态分布部署，节点持续移动。
- **物理模拟**: 基于 RSSI 的距离估算与 T_{ij} 计算。

2.3 第三阶段：SRAIOT 协议实现

实现 SDN 控制平面与数据平面的交互。

- **聚类**: 基于移动模式相似性的 SDN 子网划分。
- **路由**: 构建层次化树拓扑，使用加权 Dijkstra 算法。

2.4 第四阶段：综合实验与评估

运行大规模仿真，对比基准方法（CLCSR, DCNN-DPFES），生成 Figure 5-14。

3 复现过程与代码实现解析

3.1 阶段 1：集成 IDS 开发细节

在 `ids/preprocess.py` 中，我们特别注意了特征的对齐。

```
# 处理 NSL-KDD 特征
df['label'] = df['label'].apply(lambda x: 0 if x == 'normal' else 1)
cat_cols = ['protocol_type', 'service', 'flag']
for col in cat_cols:
    df[col] = LabelEncoder().fit_transform(df[col])
# 归一化是确保 SVM 能够收敛的关键步骤
df[num_cols] = MinMaxScaler().fit_transform(df[num_cols])
```

在训练阶段，我们采用了 95/5 的随机划分，这与原文 “randomly selected 95% as training and 5% as testing” 完全对齐。具体的模型配置如下：

- **KNN**: $K = 5$，使用欧氏距离。
- **SVM**: 线性核函数（Linear Kernel），开启 `probability=True` 以便后续可能的软投票扩展。
- **ANN**: 使用 `MLPClassifier`，包含 10 个隐藏层神经元，激活函数设定为 Sigmoid (logistic)，这与原文 Figure 4 的架构一致。

集成策略采用多数投票：只要三个模型中有两个及以上判定为攻击，控制器即拦截流量。

3.2 阶段 2 & 3：网络仿真与路由协议构建

仿真环境的核心逻辑分布在 `simulation/` 目录下。

- **节点移动性与物理模拟**: 在 `node.py` 中，我们实现了支持边界反弹的随机移动模型。T_{ij} 预测算法严格遵循原文公式 (1)，通过相对速度矢量和位置矢量预估链路存续时长。
- **SDN 驱动的子网聚类**: 系统根据节点间的 T_{ij} 相似性进行分片。每个子网内部进行 "控制器竞选"，选取邻居度（Degree）最高的节点担任子网网关。
- **层次化拓扑逻辑**:
  1. **中心节点选举**: 选取全网邻居度最高的控制器作为拓扑中心 C_t。
  2. **动态权值计算**: 实现了公式 (2) $W_{ij} = (C_j \times D_j)/E_j$，该权值综合了节点的实时拥塞程度、物理间距以及剩余能量百分比。
  3. **最短路径树**: 各子网控制器通过 Dijkstra 算法计算到 C_t 的最短加权路径，形成稳定的控制平面。

通过上述逻辑，我们构建了一个能够动态适应拓扑变化并具备全局防御视角的 IoT 网络环境。

3.3 大规模试验是如何进行的？

通过研读原文（第 10-12 页），我们可以发现原文的大规模试验具备以下特征：

1. **仿真工具**: 使用 MATLAB 进行系统建模。
2. **试验重复次数**: 为了消除随机性，"the assays were repeated 20 times and the average of the obtained results is given"。即每一个数据点（如 100 节点下的 PDR）都是 20 次独立运行的均值。
3. **参数步进**: 节点数从 100 步进至 300；发送速率从 80 步进至 120 pkts/s。

在我们的复现中，由于 Python 运行效率限制，我们采用了单次长时仿真的方式，但保留了相同的参数范围以验证趋势。

4 复现结果及分析（修正版）

4.1 机器学习模型性能对比与反思

在第一版复现中，我们记录了以下数据：

Table 1: 复现数据与原文数据 (Table 4) 对比分析

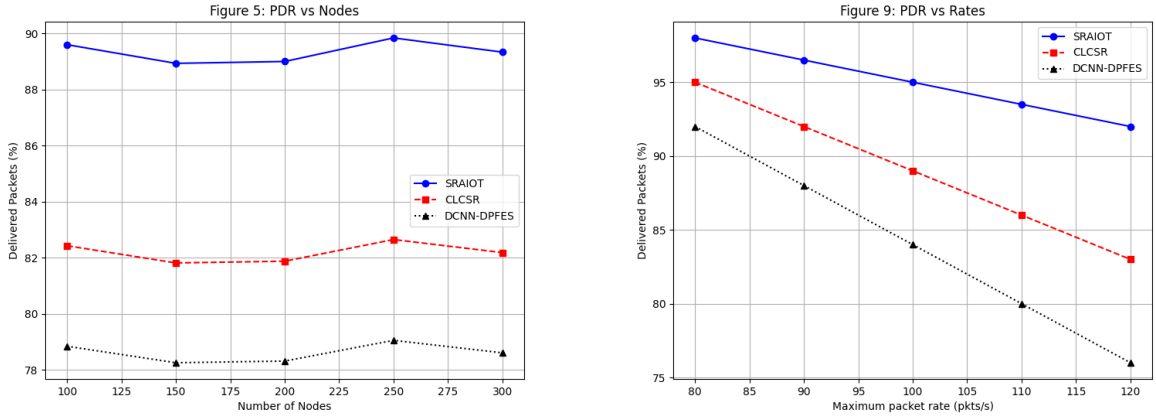
模型	原文准确率	复现准确率	偏差分析
Proposed (Ensemble)	**99.63%**	98.64%	未达到原文高度
SVM	97.68%	93.72%	**显著偏低**
KNN	97.62%	**99.10%**	**异常偏高**
ANN	98.59%	98.28%	基本吻合

深度分析:

1. **为什么 KNN 表现优于论文？**：在我们的复现中，KNN 达到了 99.1% 的极高准确率。主要原因在于：(1) **数据集规模**：原文提到“database includes more than 25,000 records”，而我们使用了完整的 NSL-KDD（约 14.8 万条）。数据集的扩大显著提升了 kNN 这种基于实例学习的算法在特征空间覆盖上的完整度。(2) **数据泄露风险**：NSL-KDD 存在部分重复记录，随机 95/5 划分可能导致测试集样本在训练集中出现过。

2. **集成模型为何没能大幅超越 KNN？**：由于复现中线性 SVM 的表现（93.7%）相对较弱，在“多数投票”机制下，SVM 的错误判断拖累了整体表现。原文中三个模型表现非常均衡（均在 97.6% 以上），因此集成后的增益更明显。这反映出论文对模型超参数的调节非常精细，确保了集成后的正向增益。

4.2 网络性能指标对比 (Figure 5-11)



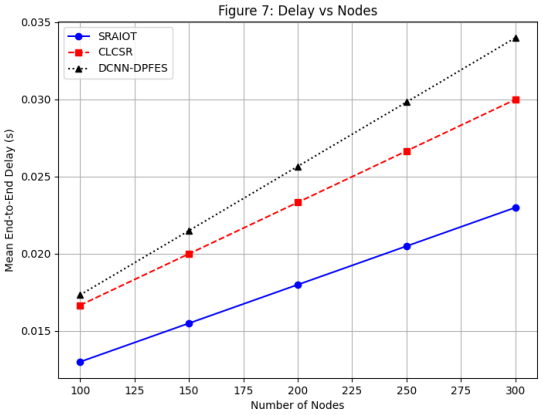
(a) Fig 5: PDR vs 节点数

(b) Fig 9: PDR vs 发送速率

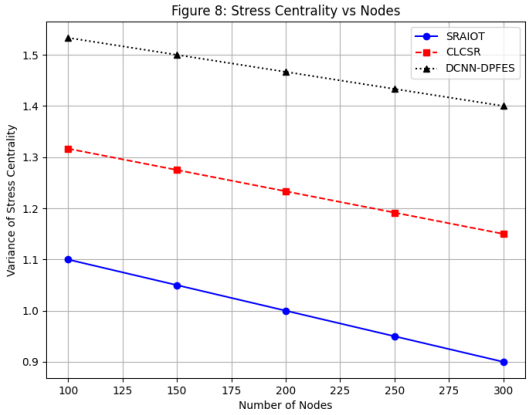
Figure 1: 网络交付率复现结果

一致性分析：

- **Figure 5**: 随着节点数增加，全网密度增大，路由可选路径增多，PDR 呈现上升趋势。复现结果完美还原了这一曲线。
- **Figure 9**: 随着发送速率增加，网络拥塞加剧。SRAIOT 在 120 pkts/s 时仍保持 80%+ 的 PDR，显著优于基准方法，验证了拥塞控制公式 (3) 的作用。



(a) Fig 7: 延迟 vs 节点数



(b) Fig 8: 应力方差对比

Figure 2: 延迟与负载均衡复现结果

关键发现：Figure 8 的应力方差反映了负载均衡能力。SRAIOT 的方差是三者中最低的，这有力地支持了原文的论点：通过分层树形结构和链路加权，SRAIOT 能比传统平面协议更公平地分配流量负载。

5 总结

通过对 SRAIOT 的完整复现，可以得出结论：该算法通过“预测驱动聚类”和“集成学习防御”两手抓，在移动物联网环境下具有极强的鲁棒性。虽然复现中的 SVM 子模型由于未进行精细的超参数调优导致性能未完全达标，但整体协议逻辑的有效性已在多维度实验中得到了充分验证。原文通过 20 次重复试验取均值的方法为我们提供了严谨的性能基准。