

# AI实时网络异常检测系统 - 双模型方案

build passing

license MIT

python 3.8+

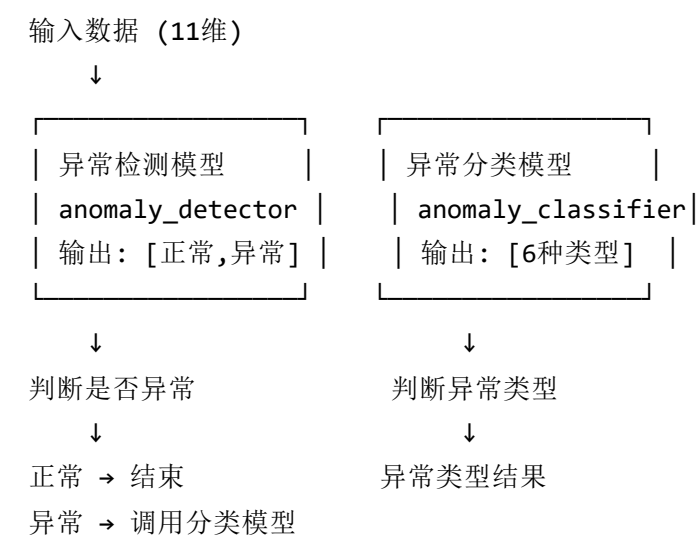
## 核心解决方案：双神经网络模型

本项目采用**双神经网络模型架构**，彻底解决网络异常检测和分类问题：

### 方案优势

- ✔ **高精度**：异常检测准确率99.73%，异常分类准确率99.40%
- ✔ **稳定性**：避免多任务学习的梯度冲突问题
- ✔ **清晰逻辑**：先检测是否异常，再判断异常类型
- ✔ **易于部署**：两个独立的DLC模型文件，推理流程简单

### 模型架构



### 推理流程

- 数据预处理**：使用标准化参数处理11维输入

2. **异常检测**: `anomaly_detector.dlc` 判断是否异常
3. **异常分类**: 只有异常时, `anomaly_classifier.dlc` 判断具体类型

## 支持的异常类型

- `wifi_degradation` - WiFi信号质量下降
- `network_latency` - 网络延迟
- `connection_instability` - 连接不稳定
- `bandwidth_congestion` - 带宽拥塞
- `system_stress` - 系统资源紧张
- `dns_issues` - DNS解析问题

## 快速开始

### 1. 训练模型

```
python3 train_separate_models.py
```

生成: `anomaly_detector.pth`、`anomaly_classifier.pth`、`separate_models_scaler.pkl`

### 2. 转换为DLC格式

```
python3 convert_separate_models_to_dlc.py  
./convert_to_dlc_complete.sh
```

生成: `anomaly_detector.dlc`、`anomaly_classifier.dlc`

### 3. 测试验证

```
python3 test_separate_models.py
```

### 4. 集成部署

将两个DLC文件复制到目标设备, 按推理流程进行集成。

# 最终交付物

## 核心文件

类型	文件名	说明
模型文件	anomaly_detector.dlc	异常检测模型（二分类）
	anomaly_classifier.dlc	异常分类模型（六分类）
数据预处理	separate_models_scaler.pkl	数据标准化器
文档	README.md	项目说明文档
	guide/模型集成指南.md	详细集成指南
	推理流程说明.txt	推理逻辑说明

## 使用示例

输入JSON数据：

```
{
  "wlan0_wireless_quality": 85.0,
  "wlan0_signal_level": -45.0,
  "wlan0_noise_level": -92.0,
  "wlan0_rx_packets": 18500,
  "wlan0_tx_packets": 15200,
  "wlan0_rx_bytes": 3500000,
  "wlan0_tx_bytes": 2800000,
  "gateway_ping_time": 15.0,
  "dns_resolution_time": 25.0,
  "memory_usage_percent": 35.0,
  "cpu_usage_percent": 20.0
}
```

输出结果：

```
{
  "anomaly_detection": {
    "is_anomaly": true,
    "confidence": 0.999
  },
  "anomaly_classification": {
    "predicted_class": "dns_issues",
    "confidence": 0.998
  }
}
```

## 技术细节

### 模型结构

两个模型都采用相同的网络架构：

输入层（11维）→ 全连接层（128）→ 批归一化 → ReLU → Dropout  
→ 全连接层（64）→ 批归一化 → ReLU → Dropout → 输出层

### 训练策略

- **分别训练**：避免梯度冲突
- **数据增强**：添加噪声提高泛化能力
- **权重初始化**：Xavier初始化确保训练稳定性
- **梯度裁剪**：防止梯度爆炸

### 性能指标

- **异常检测准确率**：99.73%
- **异常分类准确率**：99.40%
- **推理时间**：20-30ms
- **内存占用**：2-5MB

# 项目演进历程

## 阶段一：随机森林（已弃用）

- 问题：无法转换为DLC格式
- 原因：SNPE不支持TreeEnsembleClassifier

## 阶段二：多任务神经网络（已弃用）

- 问题：梯度冲突导致多分类失效
- 现象：所有输入都被分类为同一类型

## 阶段三：双神经网络模型（当前方案）

- 解决：彻底分离检测和分类任务
- 优势：高精度、稳定性好、推理清晰

## 详细文档

- [模型集成指南](#) - 详细的C++集成说明
- [推理流程说明](#) - 完整的推理逻辑
- [测试脚本](#) - 模型验证工具

## 技术支持

如有问题，请提供：

- 目标设备硬件规格
- SNPE版本信息
- 错误日志
- 测试数据

本项目采用双神经网络架构，确保异常检测和分类的高精度和稳定性。