Wk5项目总结

房颤预测项目流程梳理

1. 项目概述

这个项目旨在利用单导联心电图（ECG）信号预测短期内的房颤风险。核心思路是：分析目前未显示房颤的ECG数据，预测患者未来发生房颤的可能性，将患者分为高风险或低风险两类。

2. 数据收集与处理

2.1 数据来源

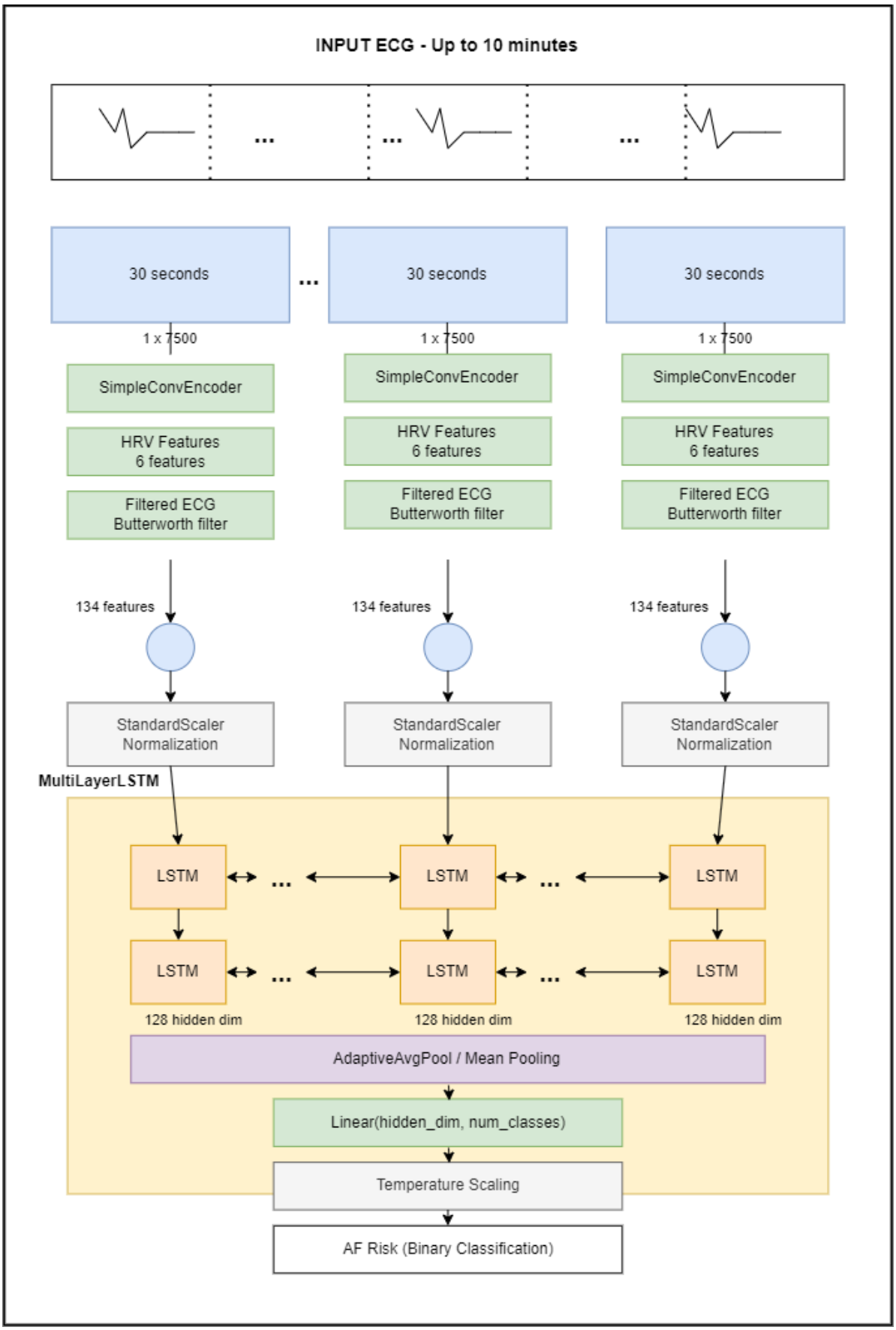
- MIT-BIH房颤数据库
- PAF预测挑战数据库

2.2 数据预处理

- **信号读取与分段**:
  - 使用wfdb库读取ECG信号和注释文件(.atr, .qrs)
  - 通过滑动窗口技术识别前10分钟无房颤，后20分钟有/无房颤的数据片段
  - 使用glob模块遍历文件目录，提取信号采样点和标签
- **信号降噪**:
  - Butterworth带通滤波器(0.5-40Hz)去除高低频噪声
  - IIR陷波滤波器消除50Hz工频干扰
  - 基于文件命名规则分段并保存为.npy文件
- **数据增强**:
  - 添加高斯噪声、拉普拉斯噪声和有色噪声
  - 时间拉伸、振幅缩放和基线漂移模拟
  - 片段交换和信号反转
  - 生成重叠片段扩充数据量
- **信号遮掩自监督学习**:
  - 随机将20%采样点置零
  - 使用卷积自编码器重构缺失部分
  - 预训练编码器提取鲁棒特征

3. 模型架构

项目采用了混合特征提取和序列分类的架构：


INPUT ECG - Up to 10 minutes
30 seconds
30 seconds
30 seconds
1 x 7500
1 x 7500
1 x 7500
SimpleConvEncoder
SimpleConvEncoder
SimpleConvEncoder
HRV Features
6 features
HRV Features
6 features
HRV Features
6 features
Filtered ECG
Butterworth filter
Filtered ECG
Butterworth filter
Filtered ECG
Butterworth filter
134 features
134 features
134 features
StandardScaler
Normalization
StandardScaler
Normalization
StandardScaler
Normalization
MultiLayerLSTM
LSTM
LSTM
LSTM
LSTM
LSTM
LSTM
128 hidden dim
128 hidden dim
128 hidden dim
AdaptiveAvgPool / Mean Pooling
Linear(hidden_dim, num_classes)
Temperature Scaling
AF Risk (Binary Classification)

3.1 特征提取组件

3.1.1 CNN特征提取器（SimpleConvEncoder）

```
# 从原始ECG信号(batch, 1, 7500)提取特征到(batch, 128)
self.conv1 = nn.Conv1d(input_channels, 32, kernel_size=3, stride=1,
padding=1)
self.conv2 = nn.Conv1d(32, 64, kernel_size=3, stride=2, padding=1)
self.conv3 = nn.Conv1d(64, feature_dim, kernel_size=3, stride=2, padding=1)
self.global_pool = nn.AdaptiveAvgPool1d(1)
```

3.1.2 HRV特征计算

```
# 提取6个HRV特征指标
def compute_hrv_features(ecg, fs=250):
    # R峰检测和标准HRV指标：mean_rr, sdnn, rmssd, nn50, pnn50
    # 频域HRV指标：lf_hf比值
    return [mean_rr, sdnn, rmssd, nn50, pnn50, lf_hf]
```

3.2 分类模型

多层LSTM序列分类器

```
# 接收组合特征(134维 = 6维HRV + 128维CNN)
self.lstm = nn.LSTM(input_dim, hidden_dim, num_layers=num_layers,
batch_first=True)
self.pool = nn.AdaptiveAvgPool2d((1, hidden_dim))
self.fc = nn.Linear(hidden_dim, num_classes)
self.temperature = nn.Parameter(torch.ones(1, dtype=torch.float)) # 校准参数
```

4. 训练策略

4.1 数据集构建

```
# 处理不同采样率的数据源，重采样到统一采样率
class ECGDataset(Dataset):
    def __init__(self, root_dirs={...}, target_fs=250, seq_len=20):
        # 支持从多个数据源加载数据
        # 将所有数据重采样到target_fs
        # 每个样本包含seq_len个30秒片段
```

4.2 处理类别不平衡

```python
# Focal Loss实现
class FocalLoss(nn.Module):
    def __init__(self, alpha=0.25, gamma=2.0, weight=None,
reduction='mean'):
        # 增加对困难样本和少数类的关注

# 类别权重计算
self.class_weights = torch.tensor([weight_low, weight_high],
dtype=torch.float)

# 加权采样
sampler = torch.utils.data.WeightedRandomSampler(samples_weight,
len(samples_weight))
```

4.3 特征提取过程

```python
def extract_features(dataloader, cnn_model, device, fs=250):
    # 从ECG片段提取HRV和CNN特征
    # 对信号进行增强和标准化
    # 错误处理和异常值检测
```

4.4 两阶段训练

4.4.1 阶段1：训练LSTM，冻结CNN

```python
# 冻结CNN参数
for param in cnn_model.parameters():
    param.requires_grad = False

# LSTM训练参数
optimizer = torch.optim.AdamW(lstm_model.parameters(), lr=5e-4,
weight_decay=1e-5)
scheduler = ReduceLROnPlateau(optimizer, mode='max', factor=0.7, patience=7)
```

4.4.2 阶段2：微调整个模型

```python
# 解冻CNN
for param in cnn_model.parameters():
    param.requires_grad = True

# 差异化学习率
optimizer = torch.optim.AdamW([
    {'params': lstm_model.parameters(), 'lr': 2e-4},
    {'params': cnn_model.parameters(), 'lr': 5e-6}
], weight_decay=2e-5)

# 循环学习率
scheduler = CyclicLR(
    optimizer,
    base_lr=[5e-5, 1e-6],
    max_lr=[3e-4, 5e-6],
    step_size_up=5,
    mode='triangular2'
)
```

4.5 评估与模型选择

```python
# 5折交叉验证
kfold = StratifiedKFold(n_splits=5, shuffle=True, random_state=42)

# 综合评估指标
results = {
    'accuracy': accuracy,
    'precision': precision,
    'recall': recall,
    'f1': f1,
    'confusion_matrix': conf_matrix
}

# 保存最佳模型
torch.save({
    'lstm_state_dict': best_model['lstm_state_dict'],
    'cnn_state_dict': best_model['cnn_state_dict'],
    ...
}, 'best_model_final.pth')
```

5. 技术特点总结

1. **信号处理技术**:
    - 带通滤波和陷波滤波降噪
    - 信号标准化与基线漂移校正
    - R峰检测算法优化
    - 不同采样率信号的重采样处理
2. **特征工程**:
    - CNN自动特征提取
    - HRV多维度指标计算(时域、频域)
    - 特征标准化和异常值处理
    - 多源特征融合策略
3. **深度学习技术**:
    - 卷积神经网络特征提取
    - LSTM序列建模
    - 温度缩放模型校准
    - 迁移学习(预训练权重)
    - 自监督学习(信号遮掩)
4. **训练策略创新**:
    - 两阶段训练流程
    - 差异化学习率设计
    - 循环学习率调度
    - 梯度裁剪防止梯度爆炸
    - Focal Loss处理类别不平衡
    - 早停策略避免过拟合
5. **验证与评估**:
    - 多指标综合评估
    - K折交叉验证
    - 模型集成选择最优模型