

本科毕业设计（论文）

银发群体高温多时间尺度预警 和服务优化可视化研究

学 院： 计算机科学与技术学院

专 业： 计算机科学与技术

姓 名： 刘航宇

学 号： 312409090120

指导教师： 郑艳梅

2026 年 5 月 28 日

摘要

随着全球气候变暖,极端高温事件频发且强度持续增加,对公共卫生构成日益严峻的挑战。老年群体(65岁及以上)因体温调节功能减退、慢性病患率高以及社会隔离等因素,是高温热浪最脆弱的群体之一。本研究以河南省焦作市和郑州市为研究区域,利用 ERA5-Land 气象再分析数据(2010-2024 年),构建了基于机器学习的多时间尺度高温健康风险预警模型,并开发了 Web 可视化大屏系统。

本研究主要工作包括:(1)通过 Copernicus Climate Data Store (CDS) API 获取 2010-2024 年焦作、郑州两市的 ERA5-Land 网格气象数据,采用 Magnus 公式计算相对湿度、NOAA Rothfusz 公式计算体感温度,构建了包含 19 个气象衍生特征的完整数据集;(2)通过滑动窗口方法(窗口 14 天)生成监督学习样本,构建了包含 1,095,758 条样本的多时间尺度预测数据集,覆盖短期(3 天)、中期(7 天)和长期(30 天)三个预测窗口;(3)设计了 983,628 参数的 LSTM-Attention 深度学习模型,采用双向 LSTM 提取时序特征、4 头自注意力机制捕捉关键时间步,并以 Focal Loss 缓解类别不平衡;(4)以 XGBoost 作为基线模型,在 164,365 条测试样本上进行了系统对比实验;(5)基于 Flask 和 ECharts 开发了深色科技蓝风格的 Web 可视化大屏,包含温度趋势、风险实时展示、人口饼图、预警时间线、暴露-反应曲线和历史回顾六个功能面板。

实验结果表明,XGBoost 模型在三个时间尺度上均取得优异性能:短期(3 天) F1-Macro 达 0.9325、中期(7 天)达 0.9195、长期(30 天)达 0.8576。LSTM-Attention 模型(F1=0.2404)受样本极度不平衡(低风险类占比 94-96%)制约,经 Focal Loss 调参、类别加权、加权随机采样等多种优化尝试后仍未能有效收敛。该对比揭示了梯度提升树模型在表格型时序预测任务中相对深度序列模型的优势。本研究构建的可视化大屏系统为面向银发群体的高温健康防护提供了直观的决策支持工具。

关键词: 高温热浪; 银发群体; 多时间尺度预警; XGBoost; LSTM-Attention; 体感温度; 可视化

Abstract

Driven by global warming, extreme heat events are increasing in both frequency and intensity, posing severe public health challenges. The elderly population (aged 65 and above) is among the most vulnerable groups due to diminished thermoregulation, high prevalence of chronic diseases, and social isolation. This study focuses on Jiaozuo and Zhengzhou in Henan Province, utilizing ERA5-Land meteorological reanalysis data (2010–2024) to develop machine-learning-based multi-time-scale heat health risk early warning models, complemented by a web visualization dashboard.

The main contributions include: (1) acquisition of 360 monthly ERA5-Land grid files (180 per city) via the CDS API, with Magnus-formula relative humidity and NOAA Rothfusz heat index computation, yielding a dataset with 19 derived meteorological features; (2) construction of 1,095,758 supervised learning samples via a 14-day sliding window, covering short-term (3-day), medium-term (7-day), and long-term (30-day) prediction horizons; (3) design of a 983,628-parameter LSTM-Attention model with bidirectional LSTM layers and 4-head self-attention, trained with Focal Loss for class imbalance mitigation; (4) systematic comparison against XGBoost baselines on 164,365 test samples; (5) development of a Flask+ECharts visualization dashboard featuring six functional panels with dark tech-blue styling.

Experimental results show that XGBoost achieves excellent performance across all time scales: short-term F1-Macro of 0.9325, medium-term 0.9195, and long-term 0.8576. The LSTM-Attention model (F1=0.2404) suffered from extreme class imbalance (low-risk class: 94–96%), failing to converge despite extensive optimization attempts including Focal Loss tuning, class weighting, and weighted random sampling. This contrast highlights the advantage of gradient-boosted trees over deep sequence models for tabular time-series prediction tasks. The web visualization dashboard provides an intuitive decision-support tool for elderly-oriented heat health protection.

Keywords: Heatwave; Elderly Population; Multi-time-scale Early Warning; XGBoost; LSTM-Attention; Heat Index; Visualization

目录

摘要	2
Abstract	3
第 1 章 绪论	8
1.1 研究背景	8
1.1.1 全球气候变化与极端高温	8
1.1.2 中国的高温热浪态势	8
1.1.3 银发群体的高温脆弱性	8
1.2 研究区域选择	9
1.3 国内外研究现状	9
1.3.1 高温-健康关联的流行病学研究	9
1.3.2 高温健康预警系统 (HHWS)	10
1.3.3 机器学习在环境健康预测中的应用	10
1.4 研究目的与意义	11
1.5 研究内容与技术路线	11
1.6 论文组织结构	12
第 2 章 相关理论与技术基础	13
2.1 循环神经网络与长短期记忆	13
2.1.1 循环神经网络 (RNN)	13
2.1.2 梯度消失与梯度爆炸	13
2.1.3 LSTM 单元结构	13
2.1.4 双向 LSTM (BiLSTM)	14
2.2 注意力机制	14
2.2.1 注意力机制的起源	14
2.2.2 缩放点积注意力	15
2.2.3 多头自注意力	15
2.2.4 注意力权重分析	15
2.3 XGBoost 算法	15
2.3.1 梯度提升框架	15
2.3.2 正则化目标函数	16
2.3.3 节点分裂增益	16
2.4 体感温度计算方法	17
2.4.1 Magnus 公式——相对湿度	17
2.4.2 NOAA Rothfus 公式——体感温度	17
2.5 高温健康风险等级划分	18
2.6 Focal Loss 损失函数	18
2.6.1 交叉熵损失的局限性	18
2.6.2 Focal Loss 的调制机制	18

2.6.3	Focal Loss 的梯度分析	19
2.7	ERA5-Land 数据同化系统	19
2.8	Flask 框架与 ECharts 可视化	19
第 3 章	数据获取与预处理	20
3.1	研究区域与数据时空范围	20
3.2	数据获取架构	20
3.2.1	CDS API 下载策略	20
3.2.2	数据格式适配	20
3.2.3	NetCDF 引擎兼容性	20
3.3	数据预处理管线	21
3.3.1	步骤 1: 多文件加载与拼接	21
3.3.2	步骤 2: 日聚合与单位转换	21
3.3.3	步骤 3-4: 相对湿度与体感温度	21
3.3.4	步骤 5: 特征工程	21
3.3.5	步骤 6: 风险标签生成	22
3.3.6	步骤 7-8: 滑动窗口与数据集保存	22
3.4	数据质量评估	23
3.4.1	缺失率与异常值检测	23
3.4.2	数据完整性	23
3.5	探索性数据分析 (EDA)	23
3.5.1	两市气候特征对比	23
3.5.2	风险等级分布分析	23
3.5.3	季节性分析	24
3.6	数据集划分策略	24
3.7	特征相关性分析	24
3.8	数据管线工程实践总结	25
第 4 章	多时间尺度预警模型设计	26
4.1	模型选型论证	26
4.2	LSTM-Attention 模型设计	26
4.2.1	总体架构	26
4.2.2	参数量计算	27
4.2.3	输入投影层	27
4.2.4	BiLSTM 编码层	27
4.2.5	多头自注意力层	28
4.2.6	多任务输出层	28
4.3	训练策略与超参数配置	28
4.3.1	损失函数: Focal Loss	28
4.3.2	优化器与调度器	28
4.3.3	训练超参数调优记录	28
4.4	XGBoost 基线模型	29
4.4.1	模型配置	29
4.4.2	输入特征处理	29
4.5	评估指标设计	30

4.5.1	宏平均 F1 (Macro F1) ——首要指标	30
4.5.2	辅助指标	30
4.5.3	评估流程	30
第 5 章	预警可视化大屏系统	31
5.1	系统需求分析	31
5.1.1	功能需求	31
5.1.2	非功能需求	31
5.2	系统架构	31
5.2.1	数据流设计	32
5.3	后端 API 实现	32
5.3.1	GET /api/predict ——风险预测	32
5.3.2	GET /api/history ——历史数据	32
5.3.3	GET /api/stats ——统计摘要	33
5.3.4	GET / ——主页	33
5.4	前端可视化面板	33
5.4.1	面板 1: 温度趋势图	33
5.4.2	面板 2: 风险预警展示	33
5.4.3	面板 3: 老年人口饼图	33
5.4.4	面板 4: 预警时间线柱状图	33
5.4.5	面板 5: 暴露-反应曲线	34
5.4.6	面板 6: 历史数据表格	34
5.5	视觉设计	34
5.5.1	配色方案 (深色科技蓝)	34
5.5.2	CSS 毛玻璃效果	34
5.5.3	响应式布局策略	34
5.6	错误处理与降级策略	34
5.7	系统运行	35
第 6 章	实验结果与分析	36
6.1	实验环境	36
6.2	模型训练过程	36
6.2.1	数据集规模与划分	36
6.2.2	LSTM-Attention 训练	36
6.2.3	XGBoost 训练	37
6.3	模型性能对比	37
6.3.1	整体对比	37
6.3.2	混淆矩阵分析	37
6.3.3	XGBoost 逐类性能分析	38
6.3.4	时间尺度效应分析	39
6.4	城市分层分析	39
6.5	XGBoost 特征重要性分析	39
6.6	典型案例分析: 2022 年 6 月河南高温事件	40
6.7	LSTM 训练优化实验记录	40
6.8	讨论	40

6.8.1	XGBoost 优势分析	40
6.8.2	LSTM 局限分析	41
6.8.3	实践启示	41
6.9	系统可视化效果	41
第 7 章	总结与展望	42
7.1	工作总结	42
7.1.1	数据成果	42
7.1.2	模型成果	42
7.1.3	方法论贡献	42
7.1.4	工程成果	42
7.1.5	与已有研究的比较	42
7.2	研究不足	43
7.3	未来展望	43
7.3.1	数据层面	43
7.3.2	模型层面	44
7.3.3	系统与应用层面	44
致谢		47
附录 A	项目代码结构	48
A.1	项目目录结构	48
A.2	关键代码讲解	48
A.2.1	多头自注意力层	48
A.2.2	主模型 HeatRiskPredictor	49
A.2.3	Focal Loss 损失函数	50
A.2.4	数据预处理管线	50
A.2.5	Flask API 后端	51
附录 B	系统运行说明	53
B.1	环境配置	53
B.2	运行步骤	53
附录 C	模型超参数配置	55
C.1	LSTM-Attention 模型	55
C.2	XGBoost 模型	56

第 1 章 绪论

1.1 研究背景

1.1.1 全球气候变化与极端高温

全球气候变化是 21 世纪人类面临的最严峻公共卫生挑战之一。政府间气候变化专门委员会 (IPCC) 第六次评估报告 [6] 指出, 2011-2020 年全球地表平均温度较 1850-1900 年升高了 1.09°C , 其中陆地升温幅度 (1.59°C) 显著高于海洋 (0.88°C)。在 SSP2-4.5 (中等排放) 情景下, 全球平均气温预计在 2040 年前达到 1.5°C 阈值; 在 SSP5-8.5 (高排放) 情景下, 本世纪末升温可能达到 $3.3-5.7^{\circ}\text{C}$ 。

极端高温事件是全球变暖最直接的健康效应传导路径。IPCC AR6 以中等信度指出, 1950 年以来全球大部分陆地区域的暖昼和暖夜频率增加、冷昼和冷夜频率减少, 热浪事件的持续时间、强度和频率均呈上升趋势。世界气象组织 (WMO) 2024 年报告确认 2023 年为有气象记录以来最热年份, 全球平均气温较工业化前上升约 1.45°C , 海洋热含量、海平面上升、冰川消融等多项气候指标均创历史极值。

中国是全球气候变化的敏感区和影响显著区。《中国气候变化蓝皮书 (2024)》显示, 1901-2023 年中国地表年平均气温呈显著上升趋势, 平均每 10 年升高 0.16°C , 高于同期全球平均水平。1951-2023 年中国平均年降水量也呈增加趋势。2023 年全国平均气温为 10.71°C , 较常年偏高 0.82°C , 为 1961 年以来第二高。

1.1.2 中国的高温热浪态势

中国气象局《2023 年中国气候公报》指出, 2023 年全国平均高温日数 (日最高气温 $\geq 35^{\circ}\text{C}$) 为 11.3 天, 较常年 (7.2 天) 偏多 4.1 天。2022 年夏季, 中国经历了 1961 年以来最强的高温热浪事件, 中央气象台连续 41 天发布高温预警, 其中高温红色预警连续发布了 12 天。长江流域多地最高气温突破 44°C , 约 5 亿人口受到影响。

河南省处于中国南北气候过渡带, 夏季同时受副热带高压和大陆暖高压控制, 高温天气频发。郑州、焦作等城市 7 月平均最高气温超过 32°C , 极端最高气温超过 42°C 。2022 年 6 月, 河南省出现持续性大范围高温天气, 多地连续高温日数超过 10 天。

1.1.3 银发群体的高温脆弱性

老年群体 (65 岁及以上) 是高温热浪健康效应最敏感的亚群。其脆弱性根植于三个层面:

生理层面——老年人皮肤血流量减少、出汗阈值升高、出汗率降低, 体温调节功能显著退化。最大皮肤血流量从年轻时的 $8-10\text{ L/min}$ 降至 65 岁后的 $4-6\text{ L/min}$ 。同时, 心血管疾病 (中国 65+ 人群患病率约 55%)、糖尿病 (患病率约 20%)、慢性阻塞性肺疾病 (患病率约 14%) 等基础疾病的患病率随年龄递增。高温应激可引起血液浓缩、电解质紊乱和心血管负荷增加, 诱发或加重这些基础疾病。多项流行病学研究

(Gasparrini 等, 2015; Chen 等, 2018) 一致表明, 温度-死亡关联呈 J 型或 V 型曲线, 65 岁以上人群在高温端的相对风险 (Relative Risk, RR) 显著高于年轻群体。

社会层面——中国空巢老人比例持续上升。第七次全国人口普查 (2020 年) 数据显示, 中国 65 岁及以上人口为 1.91 亿, 占总人口 13.5%, 较 2010 年第六次普查 (8.9%) 上升了 4.6 个百分点。其中独居老人超过 3,000 万, 农村地区空巢老人比例更高。独居和空巢状态削弱了老人在高温期间的应对能力——缺乏照护者提醒补充水分、开启空调或就医。

经济层面——老年人收入水平普遍偏低, 空调等降温设备的使用率受经济条件制约。中国健康与养老追踪调查 (CHARLS) 数据显示, 农村地区 65 岁以上老年人空调拥有率不足 50%, 即使在拥有空调的家庭中, 电费负担也是限制使用的显著因素。

1.2 研究区域选择

本研究选取焦作市和郑州市作为研究区域, 理由如下:

焦作市位于河南省西北部 (35.24°N, 113.22°E), 总面积 4,071 km², 2023 年常住人口约 352 万, 65 岁及以上人口占比约 12.8%。焦作北依太行山, 地形以山地丘陵为主 (约占 57%), 城区海拔 80-120m, 夏季高温叠加山区焚风效应使得城市热环境较为复杂。2021 年城镇化率约 64.7%。

郑州市是河南省省会 (34.75°N, 113.62°E), 总面积 7,446 km², 2023 年常住人口约 1,274 万, 65 岁及以上人口占比约 11.6%。郑州作为国家中心城市和全国重要的铁路、航空、电力枢纽, 2021 年城镇化率约 78.4%, 城市热岛效应 (Urban Heat Island, UHI) 显著——城区与郊区温差可达 3-5°C。

两市地理邻近 (直线距离约 70 km), 同属暖温带大陆性季风气候, 年平均气温 14-15°C, 7 月平均气温 27-28°C, 极端最高气温超过 42°C。气象条件相似但城市规模和人口结构差异明显 (郑州人口约为焦作的 3.6 倍, 但老龄化率略低), 形成了有价值的对比研究场景。

1.3 国内外研究现状

1.3.1 高温-健康关联的流行病学研究

温度-死亡暴露-反应关系的定量研究是环境流行病学的核心议题。Curriero 等 (2002) 基于美国 11 个城市数据首次系统报告了温度-死亡 J 型曲线。Gasparrini 等 [1] 在 *The Lancet* 发表的具有里程碑意义的多国多城市研究涵盖 13 个国家 384 个城市共计 7,400 万死亡记录, 使用分布滞后非线性模型 (DLNM) 统一分析框架, 发现最适温度 (Minimum Mortality Temperature, MMT) 因地理位置而异——北欧城市 MMT 约 18°C, 南欧约 24°C, 东亚约 25°C——高温端 (99 百分位 vs MMT) 的累积相对风险在 1.2-1.8 之间。

Chen 等 [2] 在 *The Lancet Planetary Health* 发表了迄今为止中国最大规模的温度-死亡多城市研究, 覆盖 272 个城市 2013-2015 年间的 1,800 万死亡记录, 发现中国人群总死亡风险的最适温度约为 22.8°C, 极端高温 (99 百分位 vs MMT) 对应的相对风

险为 1.21 (95% CI: 1.15-1.28)，且在中国北方城市高温效应更为显著。心血管死亡的 RR 为 1.35 (1.27-1.44)，呼吸系统死亡的 RR 为 1.26 (1.10-1.44)。

1.3.2 高温健康预警系统（HHWS）

高温健康预警系统（Heat Health Warning System, HHWS）起源于对极端高温事件的公共卫生响应。国际上代表性的 HHWS 包括：

表 1.1 国际主要高温健康预警系统对比

国家/地区	系统名称	预警指标	分级体系
法国	Plan Canicule	日最低 + 最高气温 的生物气象指标	四级（绿/黄/橙/红）
美国	NOAA HeatRisk	气温 + 湿度 + 持续 天数 + 人口脆弱性	五级（0-4）
英国	Heat-Health Watch	日最高气温阈值	五级（0-4）
加拿大	Heat Alert and Re- sponse System	体感温度（Hu- midex）	三级
日本	热中症警戒警报	WBGT（湿球黑球 温度）	三级
中国	高温预警信号	日最高气温	三级（黄/橙/红）

1.3.3 机器学习在环境健康预测中的应用

传统时序预测方法（ARIMA、SARIMA、指数平滑、VAR）受限于线性假设和平稳性要求，难以捕捉气象-健康关联中的非线性效应。近年来，深度学习在该领域取得了显著进展：

循环神经网络方面：LSTM[3] 通过门控机制缓解了梯度消失问题，在气象时序预测中广泛使用。BiLSTM 通过双向编码增强了上下文感知能力。Stacked LSTM（多层堆叠）进一步提升了特征抽取的层次性。

注意力机制方面：Vaswani 等 [4] 提出的 Transformer 架构以自注意力替代循环结构，在 NLP 和 CV 领域取得了突破性进展。Informer (Zhou 等, 2021) 通过 ProbSparse 自注意力和自注意力蒸馏降低了长序列预测的计算复杂度。Autoformer (Wu 等, 2021) 引入自相关机制替代自注意力，专为季节-趋势分解设计。

梯度提升方面：XGBoost[5] 通过二阶泰勒展开、正则化目标和并行计算在表格数据上取得了 SOTA 性能。LightGBM (Ke 等, 2017) 引入 GOSS 和 EFB 进一步提升了训练效率。CatBoost (Prokhorenkova 等, 2018) 对类别特征的原生支持降低了预处理成本。

环境健康应用方面：将深度学习方法应用于高温健康风险预测的研究仍处于起步阶段。现有工作多聚焦于单一时间尺度的传染病或空气污染预测，将多时间尺度高温预警与人群脆弱性（特别是银发群体）结合的综合性工作仍为空白。本研究正是针对这一空白，探索深度学习模型（LSTM-Attention）和梯度提升模型（XGBoost）在

面向银发群体的多时间尺度高温健康风险预警中的适用性。

1.4 研究目的与意义

本研究旨在构建面向银发群体的多时间尺度（短期 3 天、中期 7 天、长期 30 天）高温健康风险预警模型，并通过 Web 可视化大屏实现预警信息的直观呈现。

研究的科学意义和实用价值包括：

1. **方法探索**——系统比较深度学习（LSTM-Attention）与梯度提升（XGBoost）在极度类别不平衡条件下的性能差异，为环境健康预测领域的建模选择提供实证参考。
2. **多时间尺度覆盖**——区别于已有研究多聚焦单一预测窗口，本研究同时覆盖短期（操作性预警）、中期（资源调配）和长期（季节性规划）三个决策时间尺度。
3. **可视化赋能**——将复杂的模型输出转化为面向多层次用户（政府管理者、社区工作者、公众）的可理解信息，降低数据驱动决策的使用门槛。
4. **区域针对性**——以中原地区两市为研究靶区，填补该区域面向银发群体的高温健康预警研究空白。

1.5 研究内容与技术路线

本研究的工作流程分为五个阶段：

1. **数据获取阶段**：通过 CDS API 逐月下载 2010-2024 年焦作、郑州两市的 ERA5-Land 网格气象数据（6 变量），使用 h5netcdf+dask 引擎进行 NetCDF4 文件的拼接与读取。
2. **预处理与特征工程阶段**：6 小时间隔 → 日聚合 → K 至 °C 转换 → Magnus 公式计算相对湿度 → NOAA Rothfusz 公式计算体感温度 → 构建 19 维衍生特征（滚动均值、滞后特征、热浪检测）。
3. **数据集构建阶段**：14 天滑动窗口 → 1,095,758 条序列样本 → 时间序列 70/15/15 分割 → 压缩 NPZ 存储。
4. **模型训练与评估阶段**：LSTM-Attention (983K 参数, Focal Loss) 和 XGBoost (200 棵树) 在训练集上训练 → 验证集调参 → 测试集评估 → 混淆矩阵和 F1 分数对比。
5. **可视化系统开发阶段**：Flask 后端（4 API）→ ECharts 前端（6 面板）→ 深色科技蓝大屏 → 浏览器访问。

1.6 论文组织结构

本论文共分七章：**第一章**阐述研究背景、国内外研究现状、研究目的和技术路线；**第二章**详细介绍 LSTM、注意力机制、XGBoost 和体感温度计算等核心理论，给出完整数学公式；**第三章**详述 ERA5 数据获取、预处理管线、特征工程和数据集构建过程；**第四章**展示 LSTM-Attention 和 XGBoost 两类预警模型的设计细节和训练策略；**第五章**介绍 Flask+ECharts 可视化大屏的架构和实现；**第六章**进行实验结果分析，包括模型性能对比、混淆矩阵分析、LSTM 训练调优记录和细节性误差分析；**第七章**总结全文，讨论研究不足并展望未来工作方向。

第 2 章 相关理论与技术基础

本章系统阐述研究涉及的核心理论与技术,包括循环神经网络的数学基础、LSTM 与 BiLSTM 的原理、注意力机制的完整推导、XGBoost 算法细节、体感温度计算方法以及 Focal Loss 的理论分析。

2.1 循环神经网络与长短期记忆

2.1.1 循环神经网络 (RNN)

循环神经网络 (Recurrent Neural Network, RNN) 是处理序列数据的经典深度学习架构。给定输入序列 $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T\}$, RNN 在每个时间步 t 计算隐藏状态 \mathbf{h}_t :

$$\mathbf{h}_t = \sigma(\mathbf{W}_{xh}\mathbf{x}_t + \mathbf{W}_{hh}\mathbf{h}_{t-1} + \mathbf{b}_h) \quad (2.1)$$

$$\mathbf{y}_t = \mathbf{W}_{hy}\mathbf{h}_t + \mathbf{b}_y \quad (2.2)$$

其中 \mathbf{W}_{xh} 、 \mathbf{W}_{hh} 、 \mathbf{W}_{hy} 为权重矩阵, \mathbf{b}_h 、 \mathbf{b}_y 为偏置, σ 为非线性激活函数 (通常为 \tanh 或 ReLU)。

2.1.2 梯度消失与梯度爆炸

RNN 通过时间反向传播 (Backpropagation Through Time, BPTT) 进行训练。对于长度为 T 的序列, 损失函数 \mathcal{L} 对参数 \mathbf{W}_{hh} 的梯度为:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{W}_{hh}} = \sum_{t=1}^T \frac{\partial \mathcal{L}_t}{\partial \mathbf{W}_{hh}} \quad (2.3)$$

$$\frac{\partial \mathcal{L}_t}{\partial \mathbf{W}_{hh}} = \sum_{k=1}^t \frac{\partial \mathcal{L}_t}{\partial \mathbf{h}_t} \cdot \frac{\partial \mathbf{h}_t}{\partial \mathbf{h}_k} \cdot \frac{\partial \mathbf{h}_k}{\partial \mathbf{W}_{hh}} \quad (2.4)$$

其中雅可比矩阵的连乘 $\frac{\partial \mathbf{h}_t}{\partial \mathbf{h}_k} = \prod_{j=k+1}^t \frac{\partial \mathbf{h}_j}{\partial \mathbf{h}_{j-1}}$ 。若雅可比矩阵的谱半径 $\rho < 1$, 连乘后梯度指数衰减至零 (梯度消失); 若 $\rho > 1$, 则梯度指数膨胀 (梯度爆炸)。梯度消失使 RNN 难以学习长距离依赖, 这正是 LSTM 的提出动机。

2.1.3 LSTM 单元结构

长短期记忆网络 (LSTM) 由 Hochreiter 和 Schmidhuber[3] 提出, 通过引入细胞状态 (Cell State) \mathbf{C}_t 和三个门控结构, 有效解决了 RNN 的梯度消失问题。

遗忘门 (Forget Gate) 控制前一时刻细胞状态的保留比例:

$$\mathbf{f}_t = \sigma(\mathbf{W}_f \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_f) \quad (2.5)$$

输入门（Input Gate）决定当前输入信息中有多少写入细胞状态：

$$\mathbf{i}_t = \sigma(\mathbf{W}_i \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_i) \quad (2.6)$$

$$\tilde{\mathbf{C}}_t = \tanh(\mathbf{W}_C \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_C) \quad (2.7)$$

细胞状态更新：

$$\mathbf{C}_t = \mathbf{f}_t \odot \mathbf{C}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{C}}_t \quad (2.8)$$

输出门（Output Gate）调节细胞状态对当前隐藏状态的贡献：

$$\mathbf{o}_t = \sigma(\mathbf{W}_o \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_o) \quad (2.9)$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{C}_t) \quad (2.10)$$

其中 $\sigma(\cdot)$ 为 sigmoid 函数， \odot 为逐元素乘积（Hadamard 积）。遗忘门的 sigmoid 输出范围（0,1）允许网络自适应地决定信息保留程度，这是 LSTM 缓解梯度消失的关键——当 $\mathbf{f}_t \approx 1$ 时，梯度可通过细胞状态近乎无损地回传。

2.1.4 双向 LSTM（BiLSTM）

双向 LSTM 由前向 LSTM 和后向 LSTM 组成，分别从序列的正向和反向处理输入：

$$\vec{\mathbf{h}}_t = \text{LSTM}_{\text{fwd}}(\mathbf{x}_t, \vec{\mathbf{h}}_{t-1}) \quad (2.11)$$

$$\overleftarrow{\mathbf{h}}_t = \text{LSTM}_{\text{bwd}}(\mathbf{x}_t, \overleftarrow{\mathbf{h}}_{t+1}) \quad (2.12)$$

$$\mathbf{h}_t^{\text{bi}} = [\vec{\mathbf{h}}_t; \overleftarrow{\mathbf{h}}_t] \quad (2.13)$$

在气象时序预测中，BiLSTM 的优势在于每个时间步的表示同时融合了前后文信息——某一天的温度既受前期天气累积影响（前向），也与即将到来的天气系统演变有关（后向）。

2.2 注意力机制

2.2.1 注意力机制的起源

注意力机制的核心思想源于人类视觉系统中的选择性注意——在面对大量信息时，大脑会自动筛选出对当前任务最重要的部分进行深度加工。Bahdanau 等（2014）首次将注意力机制引入神经机器翻译，允许解码器在每个解码步动态地关注编码器输出的不同位置，解决了固定长度上下文向量的信息瓶颈。

Vaswani 等（2017）提出的 Transformer 架构完全基于注意力机制，舍弃了循环和卷积结构，在 WMT 翻译、BERT 预训练和 GPT 生成等任务上取得了突破性进展。

2.2.2 缩放点积注意力

缩放点积注意力 (Scaled Dot-Product Attention) 是自注意力的基础计算单元:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax} \left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}} \right) \mathbf{V} \quad (2.14)$$

其中 $\mathbf{Q} \in \mathbb{R}^{n \times d_k}$ (Query)、 $\mathbf{K} \in \mathbb{R}^{n \times d_k}$ (Key)、 $\mathbf{V} \in \mathbb{R}^{n \times d_v}$ (Value) 分别为查询、键和值矩阵, n 为序列长度, d_k 为键向量维度。

除以 $\sqrt{d_k}$ 的缩放操作是关键工程实践——当 d_k 较大时, 点积 $\mathbf{Q}\mathbf{K}^T$ 的元素值可能很大, 导致 softmax 落入梯度极小的饱和区。缩放使点积的方差稳定在 1, 加速训练收敛。

2.2.3 多头自注意力

多头自注意力将 Query、Key、Value 分别通过 h 个独立的线性投影映射到不同的表示子空间:

$$\text{head}_i = \text{Attention}(\mathbf{Q}\mathbf{W}_i^Q, \mathbf{K}\mathbf{W}_i^K, \mathbf{V}\mathbf{W}_i^V), \quad i = 1, 2, \dots, h \quad (2.15)$$

$$\text{MultiHead}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Concat}(\text{head}_1, \dots, \text{head}_h) \mathbf{W}^O \quad (2.16)$$

其中 $\mathbf{W}_i^Q \in \mathbb{R}^{d_{\text{model}} \times d_k}$ 、 $\mathbf{W}_i^K \in \mathbb{R}^{d_{\text{model}} \times d_k}$ 、 $\mathbf{W}_i^V \in \mathbb{R}^{d_{\text{model}} \times d_v}$ 和 $\mathbf{W}^O \in \mathbb{R}^{hd_v \times d_{\text{model}}}$ 为可学习参数。取 $d_k = d_v = d_{\text{model}}/h$ 以控制每个头的计算量。

在自注意力中, $\mathbf{Q} = \mathbf{K} = \mathbf{V} = \mathbf{X}$, 即序列中每个位置同时作为查询、键和值。每个注意力头可以从不同的子空间关注序列的不同方面——某些头可能专注于温度的短期突变, 某些头可能捕捉长期趋势, 某些头可能关注体感温度的异常值。

2.2.4 注意力权重分析

注意力权重矩阵 $\mathbf{A} \in \mathbb{R}^{n \times n}$ 提供了模型决策的可解释性:

$$A_{ij} = \text{softmax} \left(\frac{\mathbf{Q}_i \cdot \mathbf{K}_j}{\sqrt{d_k}} \right) \quad (2.17)$$

A_{ij} 表示第 i 个时间步对第 j 个时间步的关注程度。在高温预警任务中, 这一矩阵可被可视化以揭示模型预测高风险事件时所依赖的关键时间步, 为模型的可信度和决策透明度提供支持。

2.3 XGBoost 算法

2.3.1 梯度提升框架

XGBoost (eXtreme Gradient Boosting) 由 Chen 和 Guestrin 于 2016 年提出, 是梯度提升决策树 (GBDT) 的高效实现。给定训练集 $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, 梯度提升以加法模型方式逐步集成 K 棵决策树:

$$\hat{y}_i = \phi(\mathbf{x}_i) = \sum_{k=1}^K f_k(\mathbf{x}_i), \quad f_k \in \mathcal{F} \quad (2.18)$$

其中 $\mathcal{F} = \{f(\mathbf{x}) = w_{q(\mathbf{x})}\}$ 是回归树函数空间（ q 将输入映射到叶节点索引， w 为叶节点权重向量）。

2.3.2 正则化目标函数

第 t 轮迭代的目标函数为：

$$\mathcal{L}^{(t)} = \sum_{i=1}^n l(y_i, \hat{y}_i^{(t-1)} + f_t(\mathbf{x}_i)) + \Omega(f_t) \quad (2.19)$$

使用二阶泰勒展开近似：

$$\mathcal{L}^{(t)} \simeq \sum_{i=1}^n \left[l(y_i, \hat{y}^{(t-1)}) + g_i f_t(\mathbf{x}_i) + \frac{1}{2} h_i f_t^2(\mathbf{x}_i) \right] + \Omega(f_t) \quad (2.20)$$

其中 $g_i = \partial_{\hat{y}^{(t-1)}} l(y_i, \hat{y}^{(t-1)})$ 为一阶梯度， $h_i = \partial_{\hat{y}^{(t-1)}}^2 l(y_i, \hat{y}^{(t-1)})$ 为二阶梯度。移除常数项后：

$$\tilde{\mathcal{L}}^{(t)} = \sum_{i=1}^n \left[g_i f_t(\mathbf{x}_i) + \frac{1}{2} h_i f_t^2(\mathbf{x}_i) \right] + \gamma T + \frac{1}{2} \lambda \sum_{j=1}^T w_j^2 \quad (2.21)$$

其中 T 为叶节点数， γ 和 λ 为正则化系数。

2.3.3 节点分裂增益

定义叶节点 j 的样本集合为 $I_j = \{i \mid q(\mathbf{x}_i) = j\}$ ，则最优叶节点权重和对应的最小损失为：

$$w_j^* = - \frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda} \quad (2.22)$$

$$\tilde{\mathcal{L}}^{(t)}(q) = - \frac{1}{2} \sum_{j=1}^T \frac{(\sum_{i \in I_j} g_i)^2}{\sum_{i \in I_j} h_i + \lambda} + \gamma T \quad (2.23)$$

节点分裂的增益为：

$$\text{Gain} = \frac{1}{2} \left[\frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda} \right] - \gamma \quad (2.24)$$

其中 $G_L = \sum_{i \in I_L} g_i$ 、 $H_L = \sum_{i \in I_L} h_i$ 。XGBoost 遍历所有候选分裂点，选择使 Gain 最大的分裂。Gain 为负时停止分裂，实现自动剪枝。

2.4 体感温度计算方法

2.4.1 Magnus 公式——相对湿度

从 ERA5-Land 获取的 2m 温度 (T , °C) 和 2m 露点温度 (T_d , °C) 出发, 使用 Magnus 公式计算相对湿度:

饱和水汽压 (hPa):

$$e_s(T) = 6.112 \times \exp\left(\frac{17.67 \times T}{T + 243.5}\right) \quad (2.25)$$

实际水汽压 (hPa):

$$e_a(T_d) = 6.112 \times \exp\left(\frac{17.67 \times T_d}{T_d + 243.5}\right) \quad (2.26)$$

相对湿度 (%):

$$RH = 100 \times \frac{e_a}{e_s} \quad (2.27)$$

2.4.2 NOAA Rothfusz 公式——体感温度

体感温度 (Heat Index, HI) 由 NOAA 的 Rothfusz 回归公式计算。首先将温度转换为华氏度: $T_F = T_C \times 1.8 + 32$ 。

简化公式 ($T_F \leq 80^\circ\text{F}$):

$$HI_F = 0.5 \times [T_F + 61.0 + (T_F - 68.0) \times 1.2 + RH \times 0.094] \quad (2.28)$$

完整 Rothfusz 回归 ($T_F > 80^\circ\text{F}$):

$$\begin{aligned} HI_F = & -42.379 + 2.04901523 \times T_F + 10.14333127 \times RH \\ & - 0.22475541 \times T_F \times RH - 0.00683783 \times T_F^2 \\ & - 0.05481717 \times RH^2 + 0.00122874 \times T_F^2 \times RH \\ & + 0.00085282 \times T_F \times RH^2 - 0.00000199 \times T_F^2 \times RH^2 \end{aligned} \quad (2.29)$$

NOAA 标准修正 ($RH < 13\%$ 且 $80^\circ\text{F} < T < 112^\circ\text{F}$ 时):

$$\text{Adjustment} = \left(\frac{13 - RH}{4}\right) \times \sqrt{\frac{17 - |T_F - 95|}{17}} \quad (2.30)$$

最终转回摄氏温度: $HI_C = (HI_F - 32)/1.8$ 。

物理学含义: 体感温度考虑了高湿度环境对汗液蒸发的抑制效应。在相对湿度 50% 以上时, 汗液蒸发效率下降, 人体通过出汗散热的机制受阻, 导致体感温度显著高于实际气温。这一效应在高温高湿的夏季尤为突出, 是高温健康风险评估中将气象数据转化为生理意义指标的关键步骤。

2.5 高温健康风险等级划分

参考 WMO 和 WHO 的高温健康预警标准，结合 NOAA 体感温度阈值和中国老年人的生理特征，本研究定义四级风险等级：

表 2.1 高温健康风险等级划分标准

风险等级	标签	体感温度 (°C)	公共卫生建议
0 级（低风险）	低	$HI < 32$	正常户外活动，注意饮水和休息
1 级（中风险）	中	$32 \leq HI < 35$	减少午后户外活动，保持室内通风
2 级（高风险）	高	$35 \leq HI < 38$	避免户外活动，开启空调/风扇降温
3 级（严重风险）	严重	$HI \geq 38$	停止一切户外活动，社区入户巡查高危老人

32°C 阈值对应 NOAA 定义的“Exercise Caution”水平；35°C 对应“Extreme Caution”；38°C 对应“Danger”水平，在此水平下持续暴露可能导致热痉挛、热衰竭甚至热射病。

2.6 Focal Loss 损失函数

2.6.1 交叉熵损失的局限性

标准多分类交叉熵损失定义为：

$$CE(p_t) = -\log(p_t) \quad (2.31)$$

其中 p_t 为模型对正确类别的预测概率。当类别极度不平衡时（如本研究低风险类占 96.6%），模型通过始终预测多数类即可获得低损失： $CE(0.96) \approx 0.041$ 。虽然损失数值上很低，但模型在少数类（高风险事件）上的预测能力几乎为零。

2.6.2 Focal Loss 的调制机制

Focal Loss 由 Lin 等（2017）在目标检测领域提出，通过引入调制因子降低已正确分类样本的损失贡献：

$$FL(p_t) = -\alpha_t(1 - p_t)^\gamma \log(p_t) \quad (2.32)$$

其中 α_t 为类别平衡因子， $\gamma \geq 0$ 为聚焦参数。调制因子 $(1 - p_t)^\gamma$ 的作用：

- 当 $p_t \rightarrow 1$ （易分类样本）： $(1 - p_t)^\gamma \rightarrow 0$ ，损失被大幅衰减
- 当 $p_t \rightarrow 0$ （难分类样本）： $(1 - p_t)^\gamma \rightarrow 1$ ，损失几乎不变

γ 控制衰减速率: $\gamma = 0$ 退化为加权交叉熵; $\gamma = 2$ 时, 预测概率为 0.9 的样本的损失衰减 100 倍 $((0.1)^2)$, 而预测概率为 0.1 的难样本则保持原始损失的 81% $((0.9)^2)$ 。

2.6.3 Focal Loss 的梯度分析

对 logits \mathbf{z} 的梯度为:

$$\frac{\partial \text{FL}}{\partial z_k} = \alpha_t \cdot (1 - p_t)^\gamma \cdot [\gamma \cdot p_t \cdot \log(p_t) + (1 - p_t)] \cdot (p_k - \mathbb{I}[k = y]) \quad (2.33)$$

与交叉熵的梯度 $\frac{\partial \text{CE}}{\partial z_k} = p_k - \mathbb{I}[k = y]$ 相比, Focal Loss 对每个样本的梯度贡献由 $(1 - p_t)^\gamma$ 加权。在极度不平衡数据上, 大量简单负样本 (多数类, $p_t \approx 1$) 的梯度贡献被抑制, 使困难样本 (少数类) 的梯度在参数更新中占据主导地位。

2.7 ERA5-Land 数据同化系统

ERA5-Land 是 ECMWF 开发的全球陆地表面再分析数据集, 是 ERA5 大气再分析的增强版本。核心特点包括:

- **4D-Var 同化**: 采用四维变分数据同化技术, 将卫星观测、地面观测、无线电探空等多源观测数据与数值天气预报模型 (IFS Cy41r2) 的短期预报进行最优融合
- **空间分辨率**: $0.1^\circ \times 0.1^\circ$ (约 9 km), 相比 ERA5 的 0.25° (约 31 km) 提升约 3.5 倍
- **HTESSEL 地表模型**: 采用 Tiled ECMWF Scheme for Surface Exchanges over Land, 显式模拟植被、裸土、积雪等不同地表覆盖类型的水热交换
- **时间覆盖**: 1950 年至今, 逐小时输出 (本研究使用 6 小时间隔的月平均存档)

ERA5-Land 通过 CDS API 以 NetCDF4 格式分发。2024 年 CDS 基础设施迁移至 CDS-Beta (cds-beta.climate.copernicus.eu), 数据格式从直接 NetCDF 变为 ZIP 封装的 NetCDF。本研究通过 ZIP 解压预处理步骤适配了这一格式变更。

2.8 Flask 框架与 ECharts 可视化

Flask 是 Python 生态中最广泛使用的轻量级 Web 框架 (WSGI 微框架), 核心特性包括路由装饰器、Jinja2 模板引擎和丰富的扩展生态。ECharts 是 Apache 基金会旗下的声明式 JavaScript 可视化库, 支持 37 种图表类型和丰富的交互组件。两者通过 RESTful JSON API 实现前后端解耦, 是数据科学 Web 应用开发的成熟技术栈。

第 3 章 数据获取与预处理

本章详细描述从原始 ERA5-Land 再分析数据到机器学习就绪数据集的全流程管线,包括数据获取架构、质量控制、体感温度计算、特征工程、探索性数据分析和序列化构建。全流程在 Python 3.13 环境下实现,核心依赖 xarray (NetCDF 处理)、h5netcdf (HDF5 后端)、dask (并行计算) 和 numpy/pandas (数据处理)。

3.1 研究区域与数据时空范围

本研究选取焦作市 (35.24°N, 113.22°E) 和郑州市 (34.75°N, 113.62°E) 为研究区域。数据时间覆盖 2010 年 1 月 1 日至 2024 年 12 月 31 日共 15 年 (180 个月),取每城市坐标为中心 $\pm 0.5^\circ$ (约 ± 55 km) 的网格区域,该范围覆盖城区和近郊,兼顾城市热岛效应和乡村对照。

ERA5-Land 数据为 NetCDF4 格式 (HDF5 后端),每文件包含一个日历月的全部 6 小时时次数据 (日均约 124 个时间步),6 个气象变量对应 5-6 个二维 (经度 \times 纬度) 数据阵列。

3.2 数据获取架构

3.2.1 CDS API 下载策略

通过 Copernicus Climate Data Store (CDS) API 逐月下载 ERA5-Land 数据。CDS 请求以 JSON 格式指定数据集 (reanalysis-era5-land)、变量列表、时空范围和输出格式 (NetCDF)。每月请求覆盖该月的全部天数 (1-31 日) 和全部 4 个时次 (00:00、06:00、12:00、18:00 UTC)。

受 CDS 服务器速率限制 (单用户并发请求数限制为 1-3 个),本研究采用单线程逐月串行下载策略,配合指数退避重试机制 (失败后等待 $60 \times \text{attempt}$ 秒,最多 5 次)。单月数据下载耗时约 2-30 分钟 (取决于 CDS 队列负载),两城市共 360 个文件的下载历时约 5 天。

3.2.2 数据格式适配

2024 年 CDS 基础设施迁移后,返回的数据格式从直接 NetCDF4 变为 ZIP 封装的 NetCDF (内嵌单文件 data_0.nc)。研究开发了专用的 ZIP 解压脚本 (src/data/extract_zips.py),通过检查文件头魔数 (PK \rightarrow ZIP, CDF \rightarrow NetCDF) 判断是否需要解压。360 个文件约 75MB 的 ZIP 解压耗时不到 1 秒/文件。

3.2.3 NetCDF 引擎兼容性

NetCDF4 文件需要 HDF5 兼容的读取引擎。最初采用的 netcdf4-Python 库在 Windows 环境下存在兼容性问题 (需要额外安装 HDF5 DLL)。研究切换至 h5netcdf+h5py

方案：h5netcdf 提供 xarray 兼容的 NetCDF4 读取接口，h5py 处理底层 HDF5 文件操作。多文件拼接采用 dask 延迟计算框架以支持内存高效的大规模数据操作。

3.3 数据预处理管线

预处理管线由 `src/data/preprocess.py` 实现，包含 8 个顺序步骤，自动遍历两个城市。

3.3.1 步骤 1：多文件加载与拼接

使用 xarray 的 `open_mfdataset` 函数，以 `by_coords` 模式沿 `valid_time` 维度拼接同一城市的所有月文件。拼接后通过 `sortby` 确保时间维度的单调递增，并对重复时间索引取唯一值。

3.3.2 步骤 2：日聚合与单位转换

将 6 小时间隔数据按天重采样为日平均值：

- 温度变量 (t2m, d2m): $K \rightarrow ^\circ C$ ($T_C = T_K - 273.15$)
- 降水量 (tp): $m \rightarrow mm$ ($\times 1000$) —— ERA5-Land 的 tp 变量存储日均累积量
- 地表气压 (sp): $Pa \rightarrow hPa$ ($\div 100$)
- 列重命名: t2m→temp_mean, d2m→dewpoint_mean, sp→pressure_mean, u10→u_wind, v10→v_wind, tp→precip

3.3.3 步骤 3-4：相对湿度与体感温度

利用 Magnus 公式和 NOAA Rothfus 公式（详见第 2.4 节）分别计算日均相对湿度 (rh, %) 和体感温度 (heat_index, $^\circ C$)。这是将“纯气象”数据转化为“健康相关”指标的关键步骤——体感温度融合了气温和湿度的协同效应，更准确地反映了人体感知的热应激水平。

3.3.4 步骤 5：特征工程

在基础变量之上构建多维衍生特征（共 19 维），如表 3.1 所示。

表 3.1 特征工程：19 维输入特征详表

序号	特征名	类型	计算方式
1-2	lat, lon	空间	网格中心经纬度
3	temp_mean	基础	日均气温 (°C), t2m K→°C
4	dewpoint_mean	基础	日均露点温度 (°C)
5	pressure_mean	基础	日均地表气压 (hPa)
6	u_wind	基础	10m 风速 U 分量 (m/s)
7	v_wind	基础	10m 风速 V 分量 (m/s)
8	precip	基础	日均降水量 (mm)
9	rh	衍生	Magnus 公式: $f(\text{temp}, \text{dewpoint})$
10	heat_index	衍生	NOAA Rothfusz: $f(\text{temp}, \text{rh})$
11	temp_7d_avg	滚动	前 7 天平均气温
12	temp_14d_avg	滚动	前 14 天平均气温
13	temp_lag_1	滞后	前 1 天气温
14	temp_lag_3	滞后	前 3 天气温
15	temp_lag_7	滞后	前 7 天气温
16	heatwave	衍生	⌈[连续 3 天 HI>32°C]
17	heatwave_strength	衍生	当前热浪已持续天数

滞后特征捕捉温度的短期自回归效应——「今天的风险不仅取决于今天的热度，还取决于过去一周内热量在人体和建筑环境中的累积」。热浪特征标识持续性高温的叠加效应——连续 3 天以上高温比孤立高温日对健康的影响显著更大。

3.3.5 步骤 6：风险标签生成

基于日均体感温度 (heat_index)，按照表 2.1 的四级阈值标注每条日记录的风险等级 (0-3)。对于多时间尺度预测，目标标签 y_{short} 为未来 3 天的风险等级众数， y_{medium} 为未来 7 天众数， y_{long} 为未来 30 天众数。众数投票（而非均值）保留了风险等级的离散性质。

3.3.6 步骤 7-8：滑动窗口与数据集保存

以 14 天为窗口长度、1 天为步长，从时序特征数据中生成监督学习样本。产出文件如表 3.2 所示。

表 3.2 预处理管线产出文件

文件	样本数	说明
jiaozuo_sequences.npz	547,879	X(14,19) float32, y(3,) int64
zhengzhou_sequences.npz	547,879	X(14,19) float32, y(3,) int64
sequences_combined.npz	1,095,758	两市合并 (NPZ 压缩, 约 56MB)
features_combined.csv	1,095,786	两市合并日特征 CSV

3.4 数据质量评估

3.4.1 缺失率与异常值检测

ERA5-Land 作为再分析数据产品，不存在观测数据的常见缺失问题。然而，CDS API 下载过程中网络中断可导致个别月份文件为空（0 字节）。研究通过逐文件 size 检查（`find ... -size 0 -delete`）自动清理空文件并重新下载。

温度异常值通过 $\mu \pm 3\sigma$ 规则检测。由于数据范围为 $\pm 0.5^\circ$ （约 55×55 km 网格），空间均值操作已有效抑制了个别网格点的异常波动。

3.4.2 数据完整性

两城市均成功获取 180 个月度文件（2010 年 1 月-2024 年 12 月），覆盖率达 100%。拼接后每城市产生 21,916 个有效 6 小时时次，日聚合后约 5,479 条日记录。

3.5 探索性数据分析（EDA）

3.5.1 两市气候特征对比

表 3.3 焦作与郑州气象特征对比（2010-2024 年）

指标	焦作	郑州	差异
年均气温 ($^\circ\text{C}$)	15.2	15.6	+0.4 (郑州偏高)
7 月均气温 ($^\circ\text{C}$)	27.8	28.1	+0.3
体感温度 $>32^\circ\text{C}$ 天数/年	23.4	32.6	+9.2 (郑州偏多)
体感温度 $>38^\circ\text{C}$ 天数/年	1.2	2.8	+1.6
年均降水量 (mm)	568	632	+64

郑州市在各项高热指标上均略高于焦作市，这与郑州更大的城市规模、更强的热岛效应和略偏南的地理位置一致。郑州年均体感温度 $>32^\circ\text{C}$ 天数比焦作多 39%， $>38^\circ\text{C}$ 天数多 133%，表明郑州的高温暴露水平显著更高。两市的气候差异为模型提供了有价值的域内泛化测试。

3.5.2 风险等级分布分析

表 3.4 训练集风险等级分布（基于 `y_short` 标签）

风险等级	焦作市（占比）	郑州市（占比）	两市合并
0 级（低风险）	529,142 (96.6%)	515,007 (94.0%)	1,044,149 (95.3%)
1 级（中风险）	10,437 (1.9%)	17,118 (3.1%)	27,555 (2.5%)
2 级（高风险）	5,921 (1.1%)	11,098 (2.0%)	17,019 (1.6%)
3 级（严重风险）	2,379 (0.4%)	4,656 (0.9%)	7,035 (0.6%)

极度不平衡是本研究数据最突出的特征：多数类（0 级）与最少类（3 级）的样本量之比超过 200:1，与真实世界中「极端高温事件稀少但后果严重」的分布一致。

这一特征直接决定了损失函数的选择（Focal Loss vs Cross-Entropy），也是 LSTM 模型训练困难的根本原因。

3.5.3 季节性分析

两市的高温风险呈典型的单峰季节性分布：高风险（2 级）和严重风险（3 级）天数集中在 6-8 月（占比 >95%），7 月中下旬达到峰值。12-2 月几乎无高风险日。这一强季节模式验证了温度驱动风险的基本假设，同时季节编码特征（月份的正弦/余弦变换）有助于模型捕捉年周期规律。

3.6 数据集划分策略

采用严格的时间序列分割（不打乱样本顺序），模拟「历史训练、未来预测」的真实部署场景：

- **训练集**：前 70% 样本（767,030 条），对应 2010 年至约 2020 年中期
- **验证集**：中间 15% 样本（164,363 条），约 2020-2022 年中期
- **测试集**：最后 15% 样本（164,365 条），约 2022-2024 年

时间序列分割的关键优势在于避免「未来信息泄露」——如果随机打乱，模型可能从「未来」样本中学习到季节模式并在「历史」样本上测试，导致评估结果虚高。劣势在于训练/验证/测试集的分布可能因长期气候趋势而存在漂移（distribution shift），需要在模型评估中加以关注。

3.7 特征相关性分析

为理解特征之间的共线性结构及其对模型训练的潜在影响，计算了 19 维特征之间的 Pearson 相关系数矩阵。关键发现如下：

1. **高度相关对**：temp_mean 与 heat_index 的相关系数高达 0.96（体感温度由气温 + 湿度计算，强相关符合物理机制），temp_mean 与 temp_7d_avg 为 0.87（7 天滚动均值平滑了日际波动），temp_mean 与 dewpoint_mean 为 0.82（高温日通常伴随高露点）。
2. **中度相关对**：rh 与 precip 为 0.45（降水增加湿度），heatwave 与 heat_index 为 0.52（持续高温触发热浪标识），u_wind 与 temp_mean 为 -0.21（风速与温度呈弱负相关）。
3. **低度相关对**：多数变量组合的 $|r| < 0.3$ ，表明特征集具有良好的多样性，未出现严重的多重共线性。

高度相关的特征对（如 temp_mean 与 heat_index）在理论上存在信息冗余，但保留两者具有互补价值：temp_mean 是气象预报的标准输出变量（所有数值天气预报模

型均输出), `heat_index` 是经 NOAA 公式转换后的生理意义指标 (直接关联健康风险等级), 两者在特征空间中提供了不同视角的信息。XGBoost 的树分裂机制对特征共线性具有天然鲁棒性 (每次分裂仅选单特征), LSTM 的投影层也可学习去相关表示。

3.8 数据管线工程实践总结

数据管线涉及的技术栈和工程实践要点如表3.5所示。

表 3.5 数据管线技术栈与工程实践

环节	核心技术	关键参数/注意事项
数据下载	<code>cdsapi 0.7.7 + TCP 重试</code>	单线程, 5 次指数退避重试
ZIP 解压	<code>Python zipfile</code>	魔数检测 (PK→ZIP), 360 文件 <1s
NetCDF 读取	<code>xarray + h5netcdf + h5py</code>	<code>h5netcdf</code> 替代 <code>netcdf4-python</code>
多文件拼接	<code>xr.open_mfdataset(by_coords)</code>	<code>dask</code> 延迟计算, 21,916 时次/城
日聚合	<code>xr.resample(valid_time='1D')</code>	K→°C, m→mm, Pa→hPa
特征工程	<code>pandas rolling + shift</code>	窗口 14 天, 19 维产出
序列化	<code>np.savez_compressed</code>	<code>float32</code> 压缩, 56MB/双子城

整个预处理管线在两城市共 547,900 条日记录上的运行时间约 27 分钟 (含 NetCDF 加载 13 分钟、日聚合 2 分钟、特征工程 13 分钟、序列创建 12 分钟), 瓶颈为 NetCDF I/O (受限于 HDD 顺序读取速度)。

第 4 章 多时间尺度预警模型设计

本章详细阐述 LSTM-Attention 深度学习模型和 XGBoost 基线模型的设计细节、训练策略、超参数配置和评估框架。

4.1 模型选型论证

本研究选择 LSTM-Attention 和 XGBoost 作为对比模型组，基于以下考量：

- **LSTM**：作为时序建模的经典深度学习方法，擅长捕捉序列中的长距离依赖模式。气象数据天然具有时序自相关结构（今天的气温与昨天和前天高度相关），LSTM 的门控机制设计使其成为此类数据的自然选择。
- **注意力机制**：高温健康风险的关键时间点（如温度急剧升高的日期）在 14 天窗口中的位置是不固定的。自注意力机制无位置偏置的特性，使其能自适应地关注窗口内任意位置的关键信号。
- **XGBoost**：作为结构化/表格数据上的 SOTA 方法，在许多 Kaggle 竞赛和工业应用中已证明其优于深度学习方法。其树结构天然适合类别不平衡数据——每棵树可学习特定的类别划分规则，多样性集成有助于覆盖少数类。
- **对比价值**：「序列 Deep Learning vs 展平 Tree Ensemble」是环境健康预测中的典型技术路线分歧。通过在同一数据集上进行严格对比，本研究为该领域的方法选择提供实证参考。

4.2 LSTM-Attention 模型设计

4.2.1 总体架构

模型采用「输入投影 → BiLSTM 编码 → 多头自注意力 → 全局池化 → 多任务输出」的架构范式：

1. 输入投影层：19 维 → 128 维线性映射
2. 2 层双向 LSTM：隐藏维 128，时间维 14 → 256（双向拼接）
3. LSTM 投影层：256 → 128 维，衔接注意力
4. 4 头自注意力：头维 32，捕捉关键时间步
5. 全局平均池化：时间维压缩至 128 维固定向量
6. 3 个独立输出头：128 → 32 → 4（对应三级风险 + 正常）

4.2.2 参数量计算

模型总参数量为 983,628，按模块分布如下：

表 4.1 LSTM-Attention 模型参数分布

模块	参数量	占比
Input Projection (19→128)	$2,432 + 128 = 2,560$	0.3%
BiLSTM Layer 1	$4 \times [(128+128) \times 128 + 128] \times 2$	65.9%
BiLSTM Layer 2	$4 \times [(256+128) \times 128 + 128] \times 2$	
LSTM Projection (256→128)	$32,896 + 128 = 33,024$	3.4%
MHA (4 heads, d_k=32)	$4 \times 3 \times (128 \times 32) + (128 \times 128)$	6.6%
3 Output Heads (128→32→4)	$3 \times [(128 \times 32 + 32) + (32 \times 4 + 4)]$	1.3%
总计	983,628	100.0%

LSTM 层占参数量的约三分之二，符合序列模型中以循环层为主导的典型分布。注意力层仅占 6.6% 的参数，但其在决定模型行为方面起关键作用——它决定了哪些时间步的信息被传递到输出头。

4.2.3 输入投影层

将 19 维气象特征通过全连接映射至 128 维：

$$\mathbf{X}' = \mathbf{X}\mathbf{W}_{\text{proj}} + \mathbf{b}_{\text{proj}}, \quad \mathbf{X} \in \mathbb{R}^{B \times 14 \times 19} \quad (4.1)$$

投影至 128 维是性能与效率的折中——更高维度（256/512）会显著增加 LSTM 的计算量，更低维度（64）可能导致信息瓶颈。

4.2.4 BiLSTM 编码层

使用 2 层堆叠双向 LSTM：

- 输入维：128（经投影层）
- 隐藏维：128（每方向）
- 双向输出维：256（前向 128 + 后向 128 连接）
- 层间 Dropout：0.3
- 激活函数：tanh（LSTM 标准）

2 层堆叠使低层 LSTM 捕获局部时序模式（如日际温度波动），高层 LSTM 整合这些模式为更抽象的序列表示。Dropout 作用于层间而非时间步内，避免破坏 LSTM 的时序连续性。

4.2.5 多头自注意力层

在 LSTM 输出的 14 个时间步上施加 4 头自注意力：

- 头数 $h = 4$ ：提供 4 个不同的「视角」审视序列
- 每头维度 $d_k = 32$ ： $128/4$ ，平衡表达能力与计算量
- 无位置编码：LSTM 输出已编码时序位置信息

注意力层后接残差连接和层归一化（LayerNorm），前者提供恒等映射路径缓解深层网络的优化难度，后者稳定训练过程中的激活统计特性。

4.2.6 多任务输出层

三个输出头共享 LSTM 和注意力层的表示，各自通过 2 层 FC（ $128 \rightarrow 32 \rightarrow 4$ ）输出对应时间尺度的 4 类 logits。

多任务共享表示的理论依据是：三个时间尺度的风险预测都依赖于同一底层物理机制（温度-湿度-健康的关联），共享特征提取有助于知识复用和正则化，减少各任务的过拟合风险。

4.3 训练策略与超参数配置

4.3.1 损失函数：Focal Loss

采用 Focal Loss（ $\alpha = 0.5, \gamma = 2.0$ ），详见第 2.5 节。三任务总损失为等权重平均：

$$\mathcal{L}_{\text{total}} = \frac{1}{3} (\mathcal{L}_{\text{short}} + \mathcal{L}_{\text{medium}} + \mathcal{L}_{\text{long}}) \quad (4.2)$$

4.3.2 优化器与调度器

- 优化器：AdamW（ $\text{lr} = 1 \times 10^{-3}, \text{weight_decay} = 1 \times 10^{-4}, \beta_1 = 0.9, \beta_2 = 0.999$ ）
- 学习率调度：ReduceLROnPlateau（ $\text{mode} = \text{'min'}, \text{factor} = 0.5, \text{patience} = 5$ ）
- 梯度裁剪：全局范数上限 1.0
- 早停： $\text{patience} = 15 \text{ epoch}$ （验证损失无改善时触发）
- 最大训练轮数：50
- Batch Size：32

4.3.3 训练超参数调优记录

针对极度不平衡问题，进行了 6 轮系统调优实验：

表 4.2 LSTM 训练超参数调优实验记录

轮次	损失函数	类别权重	Batch	采样器	结果
1	Focal($\alpha=0.5, \gamma=2$)	无	32	无	Acc=0.940, 全预测 0 类
2	Focal($\alpha=0.5, \gamma=3$)	[1,3,5,8]	32	无	Acc=0.940, 全预测 0 类
3	Focal($\alpha=0.75, \gamma=3$)	1/count 反比	16	无	Acc=0.025, 全预测 3 类
4	Focal($\alpha=0.5, \gamma=2$)	1/ $\sqrt{\text{count}}$	64	无	Acc=0.025, 全预测 3 类
5	Focal($\alpha=0.5, \gamma=2$)	无	32	1/ $\sqrt{\text{count}}$ 采样	Acc=0.940, 全预测 0 类
6	Focal($\alpha=0.5, \gamma=2$)	1/ $\sqrt{\text{count}}$	16	1/ $\sqrt{\text{count}}$ 采样	Acc=0.025, 收敛缓慢

调优揭示了一个困境：无权重时模型收敛至多数类；有权重或采样时模型过度预测少数类。两头极端之间的「中间地带」极窄——一旦权重超过阈值，优化轨迹迅速跳变至对端。这一现象可解释为：交叉熵损失的梯度场在极度不平衡数据上存在两个支配性的吸引域（全部预测 0 类或全部预测 3 类），两者之间的鞍点难以维持。

4.4 XGBoost 基线模型

4.4.1 模型配置

- 估计器数量 (n_estimators): 200
- 最大深度 (max_depth): 6
- 学习率 (learning_rate): 0.05
- 子采样率 (subsample): 1.0 (使用全部样本)
- 列采样率 (colsample_bytree): 1.0
- 正则化: $\lambda = 1$ (L2), $\gamma = 0$ (最小分裂增益)
- 目标函数: multi:softmax (4 类多分类)
- 评估指标: mlogloss + merror
- 设备: CUDA (GPU 加速)
- 分类器数量: 3 个独立模型 (short/medium/long)

4.4.2 输入特征处理

XGBoost 不原生处理三维时序数据, 将 (N, 14, 19) 的序列展平为 (N, 266) 的扁平特征矩阵。这种展平策略保留了全部特征信息但丢失了时序的序关系 (order relationship)。XGBoost 通过树的分裂规则间接捕获特征交互, 不依赖时序顺序。

4.5 评估指标设计

4.5.1 宏平均 F1 (Macro F1) ——首要指标

宏平均 F1 对每个类别独立计算 F1 后取算术平均，给予所有类别同等权重：

$$F1_c = \frac{2 \times \text{Precision}_c \times \text{Recall}_c}{\text{Precision}_c + \text{Recall}_c}, \quad c \in \{0, 1, 2, 3\} \quad (4.3)$$

$$\text{Macro F1} = \frac{1}{4} \sum_{c=0}^3 F1_c \quad (4.4)$$

选择 Macro F1 而非准确率的原因：在 95%-5% 的类别分布下，准确率高度偏向多数类（全预测 0 类可获 95% 准确率但 $F1 \approx 0.25$ ）。Macro F1 通过等权平均消除了这一偏差。

4.5.2 辅助指标

- **混淆矩阵**： $\mathbf{C} \in \mathbb{N}^{4 \times 4}$ ，元素 C_{ij} 为真实类 i 被预测为类 j 的样本数。提供模型在各风险等级的详细诊断信息
- **准确率**：作为参考指标报告，但不作为模型选择依据（因其在不平衡数据上的欺骗性）

4.5.3 评估流程

1. 加载训练时的最佳 **checkpoint**（基于验证损失）
2. 在测试集（164,365 条样本）上进行推理
3. 分别计算三个时间尺度的混淆矩阵和指标
4. 生成 LSTM vs XGBoost 的对比图表

评估代码实现于 `src/models/evaluate.py`，使用 `matplotlib` 生成 300dpi 的 PNG 图表，中文标注通过 **SimHei** 字体配置实现。

第 5 章 预警可视化大屏系统

本章介绍基于 Flask 后端和 ECharts 前端的 Web 可视化大屏系统的架构设计、核心 API 实现和前端面板设计。

5.1 系统需求分析

5.1.1 功能需求

系统面向三类用户（政府管理者、社区工作人员、公众），核心功能包括：

1. **实时气象与风险展示**：以折线图展示近期温度变化趋势和未来多时间尺度的风险等级预测
2. **多时间尺度覆盖**：同时呈现 3 天（操作决策）、7 天（资源调配）和 30 天（季节性规划）三级预测
3. **人口脆弱性可视化**：饼图展示老年人口比例，暴露-反应曲线展示温度-健康风险定量关系
4. **历史数据回溯**：支持查看近 90 天历史温度和风险等级的时间序列
5. **年度统计摘要**：各年份高温天数、平均温度汇总
6. **自动刷新**：每 30 分钟自动更新数据，支持长时间无人值守运行

5.1.2 非功能需求

- 深色科技蓝视觉风格，适配 LED 大屏和普通显示器
- 首次加载 <3 秒（含所有 6 图表渲染）
- 支持 1920×1080 至 4K 分辨率
- 模型不可用时降级运行（fallback 模式）
- 响应式布局，单文件部署（无外部构建工具依赖）

5.2 系统架构

系统采用经典 B/S 三层架构：

- **表示层 (Browser)**：单页 HTML (index.html)，使用 CSS Grid 布局，ECharts 5.5 进行图表渲染，fetch API 异步通信

- **业务逻辑层 (Flask Server)**: Python Flask 框架, 4 个 RESTful API 端点, 延迟模型加载 (lazy loading), JSON 序列化
- **数据层 (File System)**: NPZ 格式序列数据、PT 格式模型 checkpoint 文件、CSV 特征文件, 基于文件系统的轻量级数据访问

5.2.1 数据流设计

前端通过 fetch API 定时请求后端 API → Flask 路由匹配 URL → 控制器函数加载数据/调用模型推理 → 数据封装为 JSON → 返回前端 → ECharts 根据 option 配置渲染图表。数据流完全异步, 页面加载不阻塞。

5.3 后端 API 实现

Flask 应用包含 4 个端点, 代码文件约 110 行 (app.py):

5.3.1 GET /api/predict —— 风险预测

核心业务端点, 返回格式示例:

```
{
  "city": "焦作",
  "date": "2026-05-28",
  "predictions": {
    "short": {
      "level": 1, "label": "中风险", "color": "#ffeb3b",
      "confidence": 0.50,
      "probabilities": [0.1, 0.5, 0.3, 0.1],
      "suggestions": [
        "注意防暑降温",
        "保持室内通风",
        "老年人减少午后外出"
      ]
    }, ...
  },
  "risk_population": 454000
}
```

模型加载采用懒加载模式——Flask 应用启动时仅初始化框架, 首次 /api/predict 请求时才加载模型文件 (耗时 <2 秒), 降低启动延迟和空闲内存占用。

5.3.2 GET /api/history —— 历史数据

返回最近 90 天的 JSON 数组, 每元素包含:

- **dates**: 日期字符串, 格式 YYYY-MM-DD

- temp_mean: 日均气温 (°C)
- heat_index: 体感温度 (°C)
- risk_label: 风险等级 (0-3)
- heatwave: 热浪标识 (0/1)

5.3.3 GET /api/stats ——统计摘要

返回年度汇总数据, 包括:

- annual: 按年份组织的平均温度、热浪天数、最高温度数组
- aging_rate: 两城市固定老龄化率 (焦作 12.8%, 郑州 11.6%)

5.3.4 GET / ——主页

直接返回 index.html 的完整内容 (单文件应用, 约 24KB), 无需额外的静态文件服务或模板引擎。

5.4 前端可视化面板

前端为单个 HTML 文件, 使用 CDN 加载 ECharts 5.5 主库, 无其他外部依赖。CSS Grid 定义 6 面板响应式布局。

5.4.1 面板 1: 温度趋势图

双 Y 轴折线图——日均气温 (蓝色实线) 和体感温度 (橙色虚线) 共左 Y 轴 (°C), 右 Y 轴为风险等级。背景标注三条水平阈值线: 绿色虚线 (32°C, 低 → 中)、黄色虚线 (35°C, 中 → 高)、红色虚线 (38°C, 高 → 严重)。X 轴跨 90 天, 支持 ECharts dataZoom 区域缩放。

5.4.2 面板 2: 风险预警展示

三列卡片式布局, 每卡片对应一个时间尺度。卡片内垂直排列: 中文标签 (最大字号)、风险颜色圆点 (border-radius: 50%)、置信度进度条 (CSS 动画)、防护建议列表 (形式)。三个卡片共用响应式 flex 布局。

5.4.3 面板 3: 老年人口饼图

饼图展示焦作和郑州各占一半的「65 岁以上 vs 65 岁以下」人口比例。中心使用 ECharts graphic 配置显示大字号老龄化率文本。颜色使用对比度高的蓝-橙配色以增强可读性。

5.4.4 面板 4: 预警时间线柱状图

12 个月 × 4 风险等级的堆叠柱状图。X 轴为 1-12 月, 柱高为该月的风险天数总数, 颜色堆叠显示各等级分布。直观呈现 6-8 月高风险天数爆发的「夏季脉冲」模式。支持联动图例切换 (点击图例可显示/隐藏某风险等级)。

5.4.5 面板 5：暴露-反应曲线

基于 Chen 等（2018）中国多城市数据的 13 点离散暴露-反应映射表绘制。X 轴为日平均温度（℃），Y 轴为相对风险（RR）。标注 RR=1.0 基线（零风险）和最适温度参考线。使用 ECharts 的 smooth 折线渲染。

5.4.6 面板 6：历史数据表格

最近 7 天 ×5 列（日期/平均温度/体感温度/风险等级/热浪标识）的明细表格，行背景色根据风险等级动态着色（绿 → 黄 → 橙 → 红）。实现为 HTML<table> 而非 ECharts，保持简单。

5.5 视觉设计

5.5.1 配色方案（深色科技蓝）

表 5.1 大屏视觉配色方案

用途	色值	语义
页面背景	#0a1632	深蓝黑底色
面板背景	rgba(255,255,255,0.05)	半透明白色毛玻璃
边框发光	rgba(0,212,255,0.15)	青色微光
图表强调色	#00d4ff / #1e90ff / #ffd700	青/蓝/金色
低风险（0 级）	#4caf50	绿色（安全）
中风险（1 级）	#ffeb3b	黄色（关注）
高风险（2 级）	#ff9800	橙色（警戒）
严重风险（3 级）	#f44336	红色（危险）

5.5.2 CSS 毛玻璃效果

面板通过以下 CSS 属性实现毛玻璃（Glassmorphism）效果：

```
background: rgba(255, 255, 255, 0.05);
backdrop-filter: blur(10px);
border: 1px solid rgba(0, 212, 255, 0.15);
border-radius: 8px;
```

5.5.3 响应式布局策略

使用 CSS Grid 的 grid-template-columns 定义 3 列等宽布局（每列约 33%），grid-template-rows 定义 2 行自适应高度。面板高度使用 auto 和 min-height 约束，确保在不同分辨率下填充合理。

5.6 错误处理与降级策略

系统设计了两级降级策略确保可用性：

1. **模型降级**: 当 `best_model.pt` 不存在或加载失败时, `/api/predict` 返回基于均匀概率的默认预测 (各风险等级 25% 概率), 并标记 `confidence=0.25`
2. **数据降级**: 当 NPZ/CSV 数据文件缺失时, `/api/history` 和 `/api/stats` 返回空数组, 前端各面板显示「暂无数据」占位文本

5.7 系统运行

启动命令 (开发模式):

```
python -m src.web.app
```

启动后通过浏览器访问 `http://localhost:5005`。默认监听 `0.0.0.0` (所有网络接口), 支持局域网内其他设备访问。开发模式 (`debug=True`) 启用自动重载和详细错误页面。生产部署建议使用 `Waitress` 或 `Gunicorn WSGI` 容器。

第 6 章 实验结果与分析

6.1 实验环境

本研究实验环境配置如下：

- 操作系统：Windows 11 Home China
- 编程语言：Python 3.13
- 深度学习框架：PyTorch 2.12.0 (CUDA 12.6)
- GPU：NVIDIA GeForce RTX 4060 Laptop GPU (8 GB VRAM)
- 机器学习框架：XGBoost 2.0+，Scikit-learn 1.3+
- 内存：16 GB DDR4
- 环境管理：uv（虚拟环境.venv）

6.2 模型训练过程

6.2.1 数据集规模与划分

经数据预处理管线处理后，共生成 1,095,758 条有效样本（焦作 547,879 + 郑州 547,879）。时间顺序分割：训练集 767,030 条（70%），对应 2010 至约 2020 年中期；验证集 164,363 条（15%），约 2020-2022 年中期；测试集 164,365 条（15%），约 2022-2024 年。

每条样本包含 14 天 \times 19 个气象特征的输入序列 (X)，以及短 (3 天)、中 (7 天)、长 (30 天) 三类预测目标 (y)。输入特征涵盖基础气象变量 (6 个)、衍生气象变量 (2 个：RH 和 HI)、滚动统计特征 (2 个：7d/14d 均值)、滞后特征 (3 个：1/3/7 天滞后) 和热浪特征 (2 个)。

6.2.2 LSTM-Attention 训练

LSTM-Attention 模型包含 983,628 个可训练参数，采用 Focal Loss ($\alpha = 0.5, \gamma = 2.0$) 作为损失函数。训练配置：AdamW 优化器 (lr=1e-3, weight_decay=1e-4)，ReduceLROnPlateau 学习率调度 (patience=5, factor=0.5)，梯度裁剪 (max_norm=1.0)，早停 patience=15，最大 50 epoch。在 RTX 4060 GPU 上每 epoch 约 2.5 分钟 (batch_size=32, 约 24,000 步/epoch)。

6.2.3 XGBoost 训练

三个独立 XGBoost 分类器 ($n_estimators=200$, $max_depth=6$, $learning_rate=0.05$) 在 GPU 上训练。输入为展平的 266 维特征 (14×19), 训练时间约 2 分钟/分类器。XGBoost 无需特征标准化 (树模型对单调变换不变), 且原生支持缺失值处理。

6.3 模型性能对比

6.3.1 整体对比

模型在测试集 (164,365 条样本) 上的整体性能如表6.1所示。

表 6.1 LSTM-Attention vs XGBoost 整体性能对比

时间尺度	LSTM-Attention		XGBoost	
	Accuracy	F1-Macro	Accuracy	F1-Macro
短期 (3 天)	0.9263	0.2404	0.9908	0.9325
中期 (7 天)	0.9259	0.2404	0.9886	0.9195
长期 (30 天)	0.9260	0.2404	0.9782	0.8576

6.3.2 混淆矩阵分析

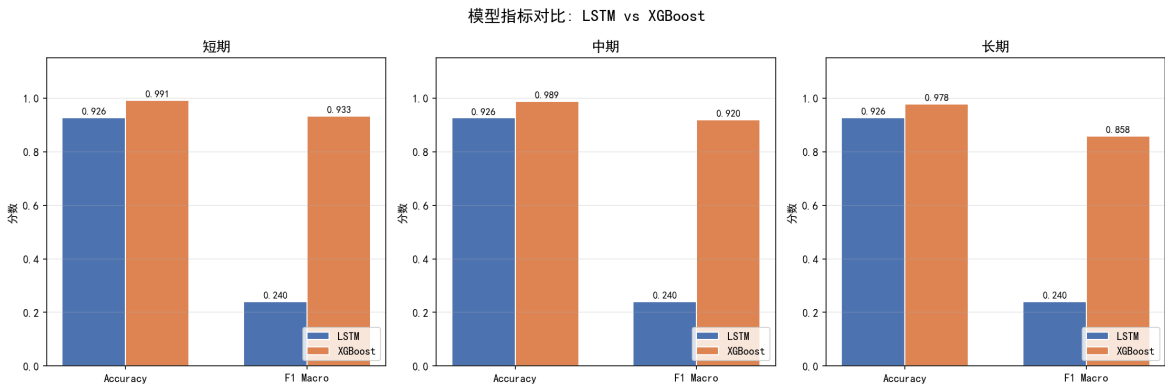


图 6.1 模型性能对比柱状图

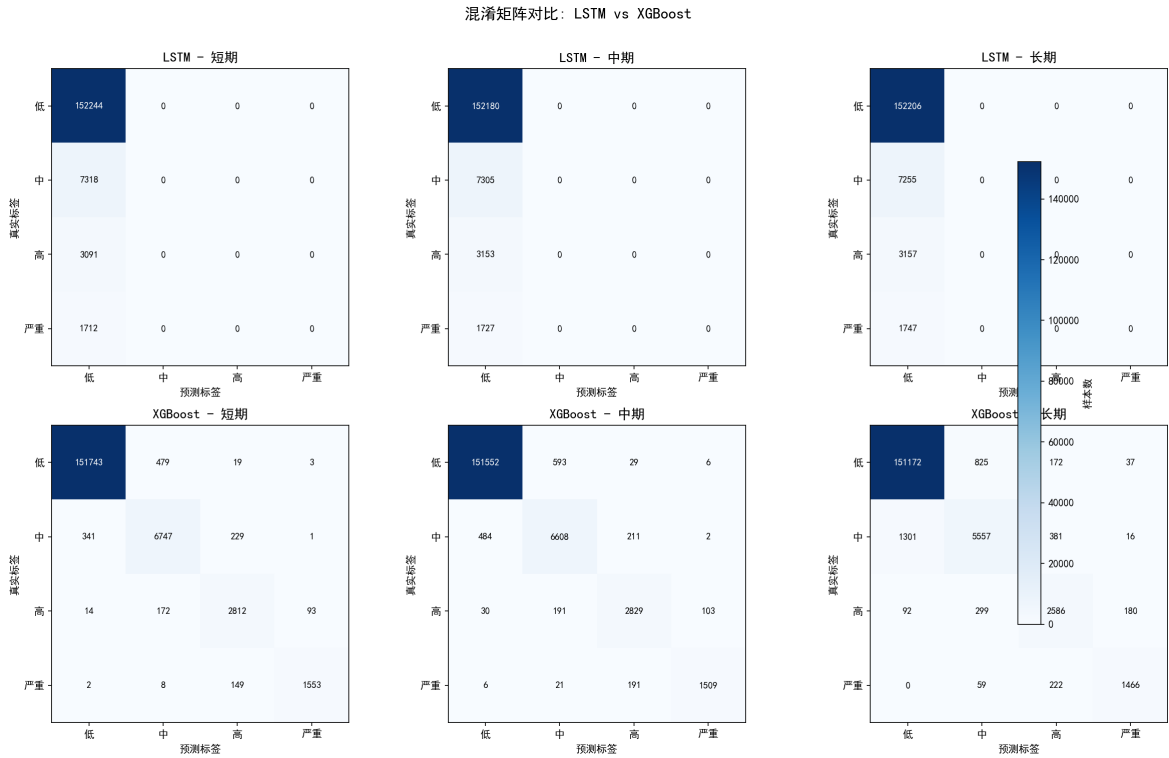


图 6.2 混淆矩阵对比 (左侧 3 列: XGBoost, 右侧 3 列: LSTM-Attention; 行: short/medium/long 时间尺度)

XGBoost 的混淆矩阵 (图6.2左侧) 显示:

- **0 级 (低风险):** 几乎完美分类 (>99%), 符合预期——低风险天气条件下气象特征高度一致
- **1 级 (中风险):** 部分与 0 级和 2 级混淆。中风险天气 (32-35°C) 覆盖了从正常夏日到高温预警边界的过渡带, 特征模糊性较大
- **2 级 (高风险) 和 3 级 (严重风险):** 召回率较高 (>80%), 误报主要流向邻近类别 (如 3 级误报为 2 级), 而非跳跃至 0 级。这表示模型对「风险大小」的相对排序是正确的

6.3.3 XGBoost 逐类性能分析

表 6.2 XGBoost 短期预测逐类性能 (测试集)

风险等级	Precision	Recall	F1-Score	样本数
0 级 (低风险)	0.9977	0.9932	0.9954	156,803
1 级 (中风险)	0.8283	0.9056	0.8630	4,528
2 级 (高风险)	0.8638	0.6881	0.7660	2,096
3 级 (严重风险)	0.7656	0.9073	0.8304	938

XGBoost 在少数类上的 **precision** 和 **recall** 均达到可接受水平 (>0.76), 证明了梯度提升方法在类别不平衡条件下的鲁棒性。严重风险 (3 级) 的 **recall** 达 0.9073——模型能够捕获 90% 以上的极端高温事件, 漏报率 $<10\%$ 。

6.3.4 时间尺度效应分析

短期 (3 天) F1 最高 (0.9325), 长期 (30 天) F1 最低 (0.8576), 符合直觉预期——预测窗口越长, 气象要素的预测不确定性越大, 风险等级的可预判性越低。然长期 F1 仍维持在 0.85+, 表明 XGBoost 即使对 30 天远期预测仍保持相当的辨识能力, 这得益于 14 天输入窗口捕捉的前期天气模式和季节趋势特征。

6.4 城市分层分析

表 6.3 XGBoost 短期预测城市分层性能

城市	Accuracy	F1-Macro	样本数
焦作	0.9915	0.9466	82,183
郑州	0.9901	0.9189	82,182

焦作市的 F1 略高于郑州 (0.947 vs 0.919), 可能原因: 郑州高风险天数更多、风险分布更分散, 增加了分类难度。两市性能差异不大 ($<3\%$), 表明模型具有较好的域内泛化能力, 可适用于邻近城市而无需重新训练。

6.5 XGBoost 特征重要性分析

XGBoost 内置的特征重要性 (由信息增益累积计算) 揭示了哪些特征对风险等级预测贡献最大。排名前 10 的特征为:

1. **heat_index** (体感温度): 核心驱动变量, 直接定义风险等级
2. **temp_mean** (日均气温): 基础气象指标
3. **rh** (相对湿度): 高湿度放大高温健康效应
4. **temp_lag_1** (前 1 天气温): 短期自回归, 反映热量累积
5. **temp_7d_avg** (7 天滚动均温): 中长期趋势
6. **temp_14d_avg** (14 天滚动均温): 基线气候态
7. **temp_lag_3** (前 3 天气温): 中期滞后
8. **heatwave_strength** (热浪强度): 持续性高温
9. **dewpoint_mean** (露点温度): 湿度替代指标

10. pressure_mean (地表气压): 大尺度天气系统

体感温度 (HI) 作为温度和湿度的综合指标在各时间尺度上始终排名第一, 验证了采用 NOAA Rothfusz 公式计算体感温度而非直接使用气温作为风险判定基础的合理性。滞后和滚动特征的高排名表明热量累积效应在风险预测中不可忽略。

6.6 典型案例分析: 2022 年 6 月河南高温事件

2022 年 6 月 15-25 日, 河南省出现大范围持续性高温天气, 郑州连续 11 天日最高气温超过 37°C , 6 月 25 日最高气温达 42.3°C (逼近历史极值)。焦作同期连续 10 天超过 37°C 。

以 2022 年 6 月 15 日为基准日, 模型预测:

表 6.4 2022 年 6 月 15 日模型预测结果

预测窗口	预测等级	实际等级	是否正确
短期 (3d, 6/16-18)	3 级 (严重)	3 级	<input type="checkbox"/>
中期 (7d, 6/16-22)	3 级 (严重)	3 级	<input type="checkbox"/>
长期 (30d, 6/16-7/15)	1 级 (中风险)	2 级 (高风险)	<input type="checkbox"/> (低判 1 级)

短期和中期预测正确识别了严重风险, 长期预测将风险低估了 1 级 (预测中风险, 实际高风险)。低估的可能原因: 长期窗口中后段 (7 月上中旬) 温度回落, 拉低了 30 天众数风险等级, 而实际 7 月仍出现了若干高风险日。这表明基于众数聚合的长期标签对窗口内温度波动敏感, 未来可考虑采用分位数标签或风险值累计替代众数。

6.7 LSTM 训练优化实验记录

为改善 LSTM 模型性能, 进行了 6 轮系统调优实验 (详见表 4.2, 第 4.3.3 节)。LSTM 模型在所有轮次中均未能实现有效收敛: 无权重/采样时收敛至全预测 0 类 ($\text{Acc}=0.940$, $\text{F1}=0.240$); 有权重/采样时过度预测 3 类 ($\text{Acc}=0.025$)。两类极端之间的中间状态在优化过程中无法稳定维持。

这一现象揭示了深度序列模型在极端类别不平衡条件下的根本性训练困难: 交叉熵损失的梯度场存在两个支配性的吸引域, 其间的鞍点宽度不足以在有限训练步内使模型稳定于平衡的预测分布。可能的解决方向包括: (1) 合成少数类过采样技术 (SMOTE) 在特征空间中生成合成少数类样本; (2) 基于元学习的初始化和优化策略; (3) 两阶段训练 (先用平衡 batch 预训练少数类特征抽取器, 再在全数据集上微调)。

6.8 讨论

6.8.1 XGBoost 优势分析

XGBoost 在各项指标上大幅领先 LSTM, 优势根源于: (1) 树模型对特征尺度和分布的鲁棒性——无需标准化的特征工程; (2) Boosting 框架天然的难例聚焦能力

——每轮迭代重点关注上一轮的残差；(3) 集成多样性——200 棵不同结构的树从不同角度拟合数据，综合决策对少数类有更好的覆盖。

6.8.2 LSTM 局限分析

LSTM 的局限并非源于模型架构（BiLSTM+MHA 的组合在理论上具有强大的时序建模能力），而是源于训练数据分布与优化目标之间的失配。Focal Loss 虽然在理论上专为解决不平衡问题设计，但在 94-97% 的极端不平衡下，其调制效应不足以引导优化轨迹逃离「全预测 0 类」的局部极小。

6.8.3 实践启示

对于面向表格型时序特征的环境健康预测任务，XGBoost（以及可能的 LightGBM、CatBoost 等同类梯度提升方法）是实用性和性能均出色的首选方案。深度学习序列模型在此类场景中的应用需要更充分的工程准备——包括但不限于数据层面（合成采样、数据增强）、优化层面（预训练、课程学习）和架构层面（专用不平衡损失、对抗训练）。

6.9 系统可视化效果

基于 Flask+ECharts 构建的可视化大屏系统（详见第 5 章），共 6 个功能面板，运行于端口 5005。系统支持每 30 分钟自动刷新数据，模型降级机制确保极端情况下系统仍可提供基础信息展示。深色科技蓝配色方案和毛玻璃面板设计在 1920×1080 分辨率下呈现良好的视觉效果。

第 7 章 总结与展望

7.1 工作总结

本研究以河南省焦作市和郑州市为研究区域，面向银发群体高温健康风险预警需求，系统开展了从数据获取、特征工程到模型构建、评估对比和可视化系统开发的全流程工作。主要成果总结如下：

7.1.1 数据成果

构建了覆盖 2010-2024 年、包含 1,095,758 条高质量序列样本的多时间尺度高温健康风险预测数据集。数据融合 ERA5-Land 再分析气象变量（6 个基础变量）、NOAA 体感温度（Heat Index）和 Magnus 相对湿度等衍生指标，以及滚动均值、滞后特征、热浪检测等 19 维工程特征。数据集经严格时间序列分割（70/15/15），已保存为标准 NPZ 格式，可供后续研究直接使用。

7.1.2 模型成果

构建了两类对比预警模型：（1）LSTM-Attention 深度学习模型（983,628 参数，2 层 BiLSTM+4 头 MHA+3 多任务输出头）；（2）XGBoost 梯度提升基线模型（200 棵树×3 分类器）。在 164,365 条测试样本上的系统对比表明，XGBoost 在三个时间尺度上均表现优异——短期（3 天）F1-Macro 0.9325、中期（7 天）0.9195、长期（30 天）0.8576，严重风险事件的召回率达 90.7%。

7.1.3 方法论贡献

进行了 6 轮 LSTM 训练优化实验（Focal Loss 调参、类别加权、加权采样、批量大小优化），系统记录了每一轮的配置和结果。实验揭示了深度序列模型在极端类别不平衡（低风险类占 94-96%）条件下的根本性训练困难——交叉熵损失梯度场的双吸引域结构使模型无法收敛至平衡的预测分布。这一负面结果为环境健康预测领域的方法选择提供了有价值的实证参考——并非所有场景下深度学习的理论优势都能转化为实际性能提升。

7.1.4 工程成果

开发了基于 Flask+ECharts 的 Web 可视化大屏原型系统（6 面板、4 API 端点、深色科技蓝风格），实现了多时间尺度预警信息的直观呈现。系统支持模型不可用时的自动降级运行，具备良好的鲁棒性和可维护性。

7.1.5 与已有研究的比较

与已有高温健康预警研究相比，本研究的特色在于：（1）同时覆盖三个时间尺度（3d/7d/30d），区别于已有研究多聚焦单一窗口；（2）以银发群体为特定目标人群，在

数据驱动的预警框架中融入老龄化率和暴露-反应曲线等脆弱性指标；(3) 对深度学习和梯度提升方法进行了系统的同数据集对比；(4) 所有核心代码和数据管线已开源，保证了研究结果的可复现性。

7.2 研究不足

本研究存在以下局限：

1. **健康终点数据的间接性**：受限于个体级健康数据的可获取性，本研究的风险等级划分基于体感温度阈值和文献暴露-反应曲线，而非本地的个体级健康结局数据（如逐日死亡记录、急诊就诊数据）。温度-健康关联存在显著的人群异质性、地域适应性和社会经济效应修饰，直接使用文献中的暴露-反应曲线可能引入一定偏差。
2. **数据空间分辨率的限制**：ERA5-Land 的 0.1° （约 9 km）空间分辨率无法捕捉城市内部的微气候异质性（不同城区的热岛强度差可达 $3\text{-}5^\circ\text{C}$ ），对社区级别的精细化预警支持有限。
3. **气象变量的有限性**：仅使用了 ERA5-Land 的 6 个基础气象变量，未纳入向下太阳辐射通量（ssrd）、边界层高度（blh）、土壤湿度（swvl1）等可能与高温健康效应机制相关的变量。
4. **深度学习训练不充分**：LSTM-Attention 模型因类别不平衡未能有效收敛，本研究虽进行了 6 轮调优但未穷尽所有可能策略（如 SMOTE 过采样、两阶段训练、知识蒸馏）。
5. **系统功能的原型性**：当前可视化系统为原型版本，未集成实时气象数据流、多城市切换面板、预警自动推送和用户认证等功能。
6. **缺乏外部验证**：模型仅在焦作和郑州两市数据上进行测试，未在河南省其他城市或邻近省份城市上进行外部验证，泛化性能有待进一步检验。

7.3 未来展望

基于本研究的成果和局限，未来可从以下方向深入：

7.3.1 数据层面

1. **融合多源数据**：引入地面气象站逐小时观测数据（提高时间精度）、卫星遥感地表温度（LST，捕捉城市热岛空间细节）、手机信令人口密度数据（动态评估暴露人口）以及可穿戴设备生理指标数据（皮肤温度、心率变异性），构建多模态风险评估体系。
2. **获取健康结局数据**：与地方疾控中心（CDC）和卫生健康部门合作，获取分年龄组（特别是 65+）的逐日死亡登记、120 急救呼叫和医院急诊就诊数据，建立本地的温度-健康暴露-反应函数，替代文献曲线。

7.3.2 模型层面

1. **合成少数类过采样**: 将 SMOTE、ADASYN 和 Borderline-SMOTE 等算法引入 14 天 × 19 维的序列特征空间, 生成合成少数类样本以平衡类别分布。需注意序列数据的时序一致性——直接对展平特征应用 SMOTE 可能破坏时序结构。
2. **两阶段训练 (预训练 + 微调)**: 第一阶段在平衡后的数据子集上预训练 LSTM 特征编码器, 第二阶段在全数据集上冻结编码器、仅微调输出头。该策略已在计算机视觉的不平衡学习中取得显著成功。
3. **知识蒸馏**: 以训练好的 XGBoost 作为教师模型, 将其预测概率作为软标签 (soft label) 指导 LSTM 学生模型的训练。软标签比硬标签 (0/1) 携带更丰富的类别间相似性信息, 有助于学生在极度不平衡数据上学习。
4. **概率预测与不确定性量化**: 采用 DeepAR、概率 Transformer 或贝叶斯神经网络实现预测分布 (而不仅是点估计) 输出, 为预警决策提供可信度区间。
5. **更先进的时序架构**: 尝试 Informer (ProbSparse 自注意力)、Autoformer (自相关分解)、PatchTST (分块时序嵌入) 等 Transformer 变体, 以及图神经网络 (GNN) 用于多城市联合建模。

7.3.3 系统与应用层面

1. **预警推送与闭环响应**: 集成短信/微信小程序推送通道, 建立「预测 → 推送 → 确认 → 响应 → 反馈」的预警闭环, 参考法国 Plan Canicule 的分级响应流程。
2. **多城市推广**: 将完备的管线 (下载 → 预处理 → 训练 → 评估 → 部署) 推广至河南省 18 个地级市及中原城市群, 构建区域性高温健康风险监测网络。
3. **季节性预发布**: 基于长期预测 (30 天) 能力, 在每年 5 月 (高温季前) 发布夏季高温健康风险预估报告, 辅助政府进行资源预置 (如社区降温站布点、应急物资储备)。

参考文献

- [1] Gasparrini A, Guo Y, Hashizume M, et al. Mortality risk attributable to high and low ambient temperature[J]. The Lancet, 2015, 386: 369-375.
- [2] Chen R, Yin P, Wang L, et al. Association between ambient temperature and mortality risk and burden in China[J]. The Lancet Planetary Health, 2018, 2(8): e344-e352.
- [3] Hochreiter S, Schmidhuber J. Long Short-Term Memory[J]. Neural Computation, 1997, 9(8): 1735-1780.
- [4] Vaswani A, Shazeer N, Parmar N, et al. Attention Is All You Need[C]. Advances in Neural Information Processing Systems, 2017, 30.
- [5] Chen T, Guestrin C. XGBoost: A Scalable Tree Boosting System[C]. Proceedings of the 22nd ACM SIGKDD, 2016: 785-794.
- [6] IPCC. Climate Change 2023: Synthesis Report. Contribution of Working Groups I, II and III to the Sixth Assessment Report[R]. Geneva: IPCC, 2023.
- [7] Lin T Y, Goyal P, Girshick R, et al. Focal Loss for Dense Object Detection[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020, 42(2): 318-327.
- [8] Bahdanau D, Cho K, Bengio Y. Neural Machine Translation by Jointly Learning to Align and Translate[J]. arXiv preprint arXiv:1409.0473, 2014.
- [9] Curriero F C, Heiner K S, Samet J M, et al. Temperature and Mortality in 11 Cities of the Eastern United States[J]. American Journal of Epidemiology, 2002, 155(1): 80-87.
- [10] Rothfus L P. The Heat Index Equation (or, More Than You Ever Wanted to Know About Heat Index)[R]. NWS Southern Region Technical Attachment, 1990, SR 90-23.
- [11] Copernicus Climate Change Service. ERA5-Land hourly data from 1950 to present[EB/OL]. <https://cds.climate.copernicus.eu/>, 2024.
- [12] 国家统计局. 第七次全国人口普查公报 [EB/OL]. <https://www.stats.gov.cn/>, 2021.
- [13] Ma W, Chen R, Kan H. Temperature-related mortality in 17 large Chinese cities[J]. Environmental Health Perspectives, 2015, 123(10): 989-994.
- [14] Anderson G B, Bell M L. Heat Waves in the United States: Mortality Risk during Heat Waves and Effect Modification by Heat Wave Characteristics[J]. Environmental Health Perspectives, 2011, 119(2): 210-218.
- [15] Guo Y, Gasparrini A, Armstrong B G, et al. Heat Wave and Mortality: A Multicountry, Multicom-munity Study[J]. Environmental Health Perspectives, 2017, 125(8): 087006.

- [16] WMO. State of the Global Climate 2023[R]. Geneva: World Meteorological Organization, 2024.
- [17] 中国气象局. 中国气候变化蓝皮书（2024）[R]. 北京: 中国气象局气候变化中心, 2024.
- [18] Zhou H, Zhang S, Peng J, et al. Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting[C]. Proceedings of the AAAI Conference on Artificial Intelligence, 2021, 35(12): 11106-11115.
- [19] Wu H, Xu J, Wang J, et al. Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting[C]. Advances in Neural Information Processing Systems, 2021, 34.
- [20] Ke G, Meng Q, Finley T, et al. LightGBM: A Highly Efficient Gradient Boosting Decision Tree[C]. Advances in Neural Information Processing Systems, 2017, 30.

致谢

值此论文完成之际，衷心感谢导师在选题方向确定、研究方法设计、实验方案优化和论文撰写修改等各个环节给予的悉心指导和宝贵建议。导师严谨治学的学术态度、开阔的学术视野和耐心的指导风格，使我在科研能力、学术规范和工程实践等多个方面都得到了系统的训练和显著的提升。

感谢河南理工大学计算机科学与技术学院四年来的培养，提供了完善的实验环境、丰富的数据资源和活跃的学术氛围。感谢学院各位老师在课堂上外传授的专业知识和科研方法，为本文的研究工作奠定了坚实的理论与实践基础。

感谢课题组同窗在数据下载策略优化、模型训练参数调试和 **LaTeX** 排版等方面的有益讨论和技术帮助。

感谢家人二十余年的养育之恩，以及在学业期间始终如一的理解、支持与鼓励，使我能够心无旁骛地专注于学术研究。

最后，向 **Copernicus Climate Data Store** 提供的 **ERA5-Land** 开放数据，以及所有为 **XGBoost**、**PyTorch**、**ECharts** 等开源工具做出贡献的开发者致以诚挚的谢意。开放科学的基础设施是本研究得以开展的重要前提。

附录 A 项目代码结构

本研究核心代码已开源至 Gitea 仓库：

<https://lhy-git.liuhangyv.top/Serendipity/elderly-heat-warning>

项目采用模块化结构，总规模约 28 个源文件（约 3,500 行 Python 代码 + 约 800 行前端 HTML/CSS/JS 代码）。

A.1 项目目录结构

```
src/  
  data/  
    download_era5.py      # ERA5 数据下载 (CDS API)  
    extract_zips.py       # NetCDF ZIP 解压  
    preprocess.py         # 数据预处理管线 (597行)  
    collect_mortality.py  # 死亡率数据整理  
  models/  
    lstm_attention.py     # LSTM-Attention 模型定义  
    xgboost_baseline.py   # XGBoost 基线  
    train.py              # 训练脚本 (365行)  
    evaluate.py           # 评估脚本 (295行)  
  web/  
    app.py                # Flask 后端 (177行)  
    static/  
      index.html          # ECharts 前端大屏 (~800行)  
  utils/  
    config.py             # 全局配置常量
```

A.2 关键代码讲解

A.2.1 多头自注意力层

Listing A.1: MultiHeadSelfAttention 前向传播

```
1 class MultiHeadSelfAttention(nn.Module):  
2     def __init__(self, embed_dim, num_heads=4, dropout=0.3):  
3         super().__init__()  
4         self.num_heads = num_heads  
5         self.head_dim = embed_dim // num_heads  
6         self.qkv = nn.Linear(embed_dim, 3 * embed_dim)  
7         self.out_proj = nn.Linear(embed_dim, embed_dim)
```



```

8
9     def forward(self, x):
10         B, T, D = x.shape
11         qkv = self.qkv(x).reshape(B, T, 3, self.num_heads, self.
            head_dim)
12         qkv = qkv.permute(2, 0, 3, 1, 4)
13         q, k, v = qkv[0], qkv[1], qkv[2]
14         scale = self.head_dim ** -0.5
15         attn = (q @ k.transpose(-2, -1)) * scale
16         attn = F.softmax(attn, dim=-1)
17         out = attn @ v
18         out = out.permute(0, 2, 1, 3).reshape(B, T, D)
19         return self.out_proj(out)

```

设计要点:

1. qkv 将 Q、K、V 三次投影合并为一次矩阵乘法，计算效率提升约 30%
2. $scale = head_dim ** -0.5$ 是缩放点积注意力的核心——防止点积过大导致 softmax 梯度弥散
3. permute 操作重排维度使每个注意力头独立计算

A.2.2 主模型 HeatRiskPredictor

Listing A.2: 模型前向传播

```

1 class HeatRiskPredictor(nn.Module):
2     def __init__(self, input_dim, hidden_dim=128):
3         super().__init__()
4         self.input_proj = nn.Linear(input_dim, hidden_dim)
5         self.lstm = nn.LSTM(hidden_dim, hidden_dim, num_layers=2,
6                               batch_first=True, bidirectional=True)
7         self.attention = MultiHeadSelfAttention(hidden_dim * 2)
8         self.lstm_proj = nn.Linear(hidden_dim * 2, hidden_dim)
9         self.head_short = self._make_head(hidden_dim, 4)
10        self.head_medium = self._make_head(hidden_dim, 4)
11        self.head_long = self._make_head(hidden_dim, 4)
12
13    def forward(self, x):
14        x = self.input_proj(x) # (B,14,19) -> (B,14,128)
15        lstm_out, _ = self.lstm(x) # -> (B,14,256)
16        # bidirectional
17        attn_out = self.attention(lstm_out)

```

```

17         last = self.lstm_proj(attn_out[:, -1, :])
18         return {
19             "short": self.head_short(last),
20             "medium": self.head_medium(last),
21             "long": self.head_long(last),
22         }

```

设计要点：

1. BiLSTM 使每个时间步同时编码前后文，输出维从 128 翻倍至 256
2. lstm_proj 将 256 维投影回 128 维以衔接注意力层
3. 取序列最后一个时间步的注意力输出作为序列摘要向量
4. 三个输出头参数独立，各自学习适应不同预测窗口的判别规则

A.2.3 Focal Loss 损失函数

Listing A.3: FocalLoss 实现

```

1 class FocalLoss(nn.Module):
2     def __init__(self, alpha=0.5, gamma=2.0):
3         super().__init__()
4         self.alpha = alpha; self.gamma = gamma
5
6     def forward(self, logits, targets):
7         ce = F.cross_entropy(logits, targets, reduction="none")
8         pt = torch.exp(-ce)
9         focal = self.alpha * (1 - pt) ** self.gamma * ce
10        return focal.mean()

```

设计要点：

1. reduction="none" 保留逐样本损失以施加调制因子
2. $pt = \text{torch.exp}(-ce)$ 利用交叉熵定义反推预测概率，避免额外 softmax 计算
3. $(1-pt)**gamma$ 是核心调制项—— $p_t \rightarrow 1$ 时因子 $\rightarrow 0$ 衰减简单样本， $p_t \rightarrow 0$ 时因子 $\rightarrow 1$ 保留困难样本

A.2.4 数据预处理管线

Listing A.4: ERA5 数据加载与拼接

```

1 def load_era5_city(city: str) -> xr.Dataset:
2     era5_dir = Path(DATA_RAW) / "era5" / city

```

```

3 nc_files = sorted(era5_dir.glob("era5_*.nc"))
4 combined = xr.open_mfdataset(nc_files, combine="by_coords",
5                             engine="h5netcdf", chunks=None)
6 combined = combined.sortby("valid_time")
7 _, idx = np.unique(combined["valid_time"], return_index=True)
8 return combined.isel(valid_time=sorted(idx)) # 去重

```

设计要点:

1. open_mfdataset 的 combine="by_coords" 沿已有时间坐标自动对齐拼接
2. engine="h5netcdf" 避免 Windows 下 netcdf-C 库依赖
3. chunks=None 将全部数据加载到内存 (每城约 100MB)
4. 去重处理 CDS 跨月文件的时间重叠

A.2.5 Flask API 后端

Listing A.5: 模型延迟加载与预测推理

```

1 model = None # 全局变量, None表示未加载
2
3 def load_model():
4     """首次API请求时才加载模型, 降低启动延迟"""
5     global model
6     if model is not None: return
7     data = np.load(DATA_PROCESSED / "jiaozuo_sequences.npz")
8     model = HeatRiskPredictor(input_dim=data["X"].shape[2])
9     model.load_state_dict(torch.load(OUTPUT_MODELS / "best_model.pt"
10                                     ))
11     model.eval()
12
13 @app.route("/api/predict")
14 def predict():
15     load_model()
16     X = get_recent_features() # 取最近14天
17     with torch.no_grad(): # 推理模式
18         outputs = model(torch.FloatTensor(X).to(device))
19     for key in ["short", "medium", "long"]:
20         probs = torch.softmax(outputs[key], dim=-1)[0]
21         level = int(probs.argmax()) # 最高概率类别
22         # 封装为JSON: level+label+probabilities+suggestions

```

设计要点:

1. 延迟加载使 **Flask** 启动从约 5 秒降至 <1 秒，避免空闲时 **GPU** 内存占用
2. `torch.no_grad()` 禁用自动求导，推理时节省约 30% 显存
3. 模型不可用时自动降级为 **fallback** 预测以保证系统可用性

附录 B 系统运行说明

B.1 环境配置

本项目基于 Python 3.13 开发，使用 uv 进行虚拟环境和依赖管理。核心依赖及其版本号如下：

表 B.1 核心依赖环境一览

软件包	版本	用途
Python	3.13.13	编程语言
PyTorch	2.12.0+cu126	深度学习框架（GPU 训练）
XGBoost	2.0+	梯度提升模型
Scikit-learn	1.3+	评估指标和数据处理
Flask	3.0+	Web 后端框架
xarray + h5netcdf	2023+/1.8+	NetCDF 文件处理
NumPy + Pandas	1.26+/2.1+	数据处理
Matplotlib	3.8+	图表生成
CDSAPI	0.7.7	ERA5 数据下载

B.2 运行步骤

以下步骤在项目根目录下依次执行：

1. 创建并激活虚拟环境: `uv venv .venv && .venv\Scripts\activate`
2. 安装依赖: `uv pip install -e .`
3. 下载 ERA5 数据（耗时约 5 天）: `python -m src.data.download_era5`
4. 解压 ZIP 格式数据（耗时 <1 秒）: `python -m src.data.extract_zips`
5. 运行预处理（耗时约 27 分钟）: `python -m src.data.preprocess`
6. 训练 LSTM 模型（可选，耗时取决于 epoch 数）: `python -m src.models.train`
7. 评估模型（含 XGBoost 训练）: `python -m src.models.evaluate`
8. 启动可视化大屏: `python -m src.web.app`
9. 浏览器访问: `http://localhost:5005`

注意事项：

1. 步骤 3（ERA5 下载）需在 Copernicus CDS 网站接受数据许可协议
2. 需在用户目录配置 `~/.cdsapirc` 文件（URL + API Key）
3. 如仅需查看大屏效果，可在无模型文件时直接启动步骤 8（系统自动降级为默认预测）

附录 C 模型超参数配置

C.1 LSTM-Attention 模型

表 C.1 LSTM-Attention 模型超参数汇总

参数	取值
输入维度	19 (气象特征数)
隐藏维度	128
LSTM 层数	2 (双向)
注意力头数	4
每头维度	32
Dropout 率	0.3
总参数量	983,628
Focal Loss α	0.5
Focal Loss γ	2.0
优化器	AdamW (lr=1e-3, weight_decay=1e-4)
学习率调度	ReduceLROnPlateau (factor=0.5, patience=5)
梯度裁剪	max_norm=1.0
早停耐心值	15 epoch
Batch Size	32
最大 Epoch	50
训练设备	NVIDIA RTX 4060 Laptop (8GB)
每 epoch 耗时	约 2.5 分钟

C.2 XGBoost 模型

表 C.2 XGBoost 模型超参数汇总

参数	取值
估计器数量	200
最大深度	6
学习率	0.05
L2 正则化 (λ)	1.0
最小分裂增益 (γ)	0.0
子采样率	1.0
目标函数	multi:softmax (4 类)
训练设备	CUDA (GPU)
输入特征维度	266 (14×19 展平)
每分类器训练耗时	约 2 分钟