



中国研究生创新实践系列大赛  
“华为杯”第二十一届中国研究生  
数学建模竞赛

学 校 中山大学

---

参赛队号 24105580031

---

1.谭苟

---

队员姓名 2.李民

---

3.唐锡涛

---

# 中国研究生创新实践系列大赛

## “华为杯”第二十一届中国研究生

### 数学建模竞赛

#### 题 目：WLAN 组网中网络吞吐量建模

#### 摘 要

无线局域网（Wireless Local Area Network, WLAN）使用无线信道作为传输介质连接两个或多个设备。WLAN 基于 IEEE 802.11 标准不断演进，其部署场景也从家庭场景迅速覆盖到无线化办公、教育、医疗等场景，下一代 Wi-Fi 7 标准支持峰值速率 30Gbps，但在高密部署场景下，节点密集度增加，相邻小区覆盖范围重叠使得干扰、碰撞等问题突出，实际部署带宽和数据传输速率大幅下降，可支持系统吞吐量仍然有限，因此精准和快速的吞吐量预测具有十分重要的价值和意义，并且能够为设计鲁棒性和高性能 WLAN 系统提供很大的技术提升空间。

针对问题一，首先进行**数据清洗**，剔除发送流量类型和标签集存在缺失值的部分。然后，对输入集进行 **Z-score 标准化**、**0-1 归一化**，并剔除异常数据。对清洗后的数据，采用 **PCA** 和 **XGBoost** 预测贡献度分析影响重要性。将得到的 4 类重要变量，作为模型输入。模型设计部分，按照同频 AP 个数进行分类建模。我们对比了**线性回归**、**XGBoost**、**随机森林**、**多层感知机**、**自编码器**和**变分自编码器**多个模型在数据集上预测发送数据帧序列总时长（seq\_time）的表现，并通过**均方误差（MSE）**寻找最优的模型。针对模型训练时数据集规模较小的问题，我们采用了**数据增强**的方法丰富数据集，使得模型精度有了较大提升。多次实验表明，收敛后的 XGBoost 模型的均方根误差为 **0.446（2AP 模型）**，**8.77（3AP 模型）**。

针对问题二，为了同时满足预测空间流数和调制编码方案（MCS, NSS）的要求，本文提出了**联合编码构建标签集**方法。我们对比了**逻辑回归分类**、**XGBoost**、**随机森林**、**多层感知机**设计的分类模型，在训练集上分类（MCS, NSS）的表现，并通过模型预测的**准确率（accuracy）**和**F1-Score**寻找最优的模型。多次实验表明，收敛后的 XGBoost 模型的准确率为 **100%（2AP 模型）**，**98%（3AP 模型）**。

针对问题三，为了预测网络吞吐量，在问题一、二的基础上，在输入向量添加实测中统计的数据帧真实空间流数和调制编码方案（MCS, NSS）。我们沿用了问题一中的回归模型进行测试，并通过**均方误差（MSE）**寻找最优的模型。多次实验表明，收敛后的 XGBoost 模型的均方根误差为 **2.665（2AP 模型）**，**4.005（3AP 模型）**。

**关键词：**相关性分析 影响度排序 机器学习 WLAN 吞吐量预测

## 目 录

1	问题重述	3
1.1	问题背景	3
1.2	问题描述	3
2	基本假设与符号说明	4
2.1	基本假设	4
2.2	符号说明	4
3	数据预处理	5
3.1	数据集基本信息分析	5
3.2	缺失值处理	5
3.3	异常值处理	6
3.4	指标归一化	6
3.5	数据集增强	6
4	问题一的模型建立与求解	8
4.1	问题分析	8
4.2	影响因子相关性分析	8
4.3	影响性强弱排序模型	13
4.4	特征选择	15
4.5	发送机会预测模型	15
4.6	问题一的预测结果	26
5	问题二的模型建立与求解	27
5.1	问题分析	27
5.2	特征选择	27
5.3	模型参数设置	28
5.4	模型训练	29
5.5	问题二的预测结果	31
6	问题三的模型建立与求解	32
6.1	问题分析	32
6.2	特征选择	34
6.3	模型训练	34
6.4	问题三的预测结果	36
7	模型评价与展望	38
7.1	创新点	38
7.2	优点	38
7.3	缺点	38
7.4	展望	38
	参考文献	39
	附录	40

# 1 问题重述

## 1.1 问题背景

无线局域网（Wireless Local Area Network, WLAN）是一种无线计算机网络，使用无线信道作为传输介质连接两个或多个设备。WLAN 基于 IEEE 802.11 标准不断演进，从语音/邮件到网页和视频以及更加复杂的虚拟现实/增强现实，WLAN 的部署场景也从家庭场景迅速覆盖到无线化办公、教育、医疗、工业制造、仓储等场景。下一代 Wi-Fi 7 标准支持峰值速率 30Gbps，但在高密部署场景下，节点密集度增加，相邻小区覆盖范围重叠使得干扰、碰撞等问题突出，实际部署带宽和数据传输速率大幅下降，因此可支持系统吞吐量仍然有限，需要对 WLAN 系统进一步优化。

WLAN 优化问题的基础问题是吞吐量预测，精准和快速的吞吐量预测能够为设计鲁棒性和高性能 WLAN 系统提供很大的技术提升空间。最近一些研究利用机器学习的方法，通过特征提取，如信道、带宽、流量、发送功率、信道接入机制等基本信息、各个节点之间的接收信号强度（Received Signal Strength Indication, RSSI）、信干噪比（Signal to Interference and Noise Ratio, SINR）、传输时间等架构信息、以及节点动态位置、动态干扰等临时信息，进行训练和建模，从而预测系统的吞吐量。这些研究大多通过与仿真结果对比，来验证模型的预测精度。然而实际部署中，WLAN 系统的通信行为受到信道环境、干扰的快速变化，业务流量复杂多样的影响，仿真结果并不精确，因此依靠仿真所训练的模型无法真正商用。

综上，基于 WLAN 实测数据，分析 WLAN 网络拓扑、节点间 RSSI、信道接入机制、干扰等因素对 WLAN 数据发送、速率的影响，并进一步对 WLAN 系统吞吐量进行精确预测和优化，具有十分重要的价值和意义。

## 1.2 问题描述

在本次题目的研究中，我们根据赛题所给出的三个子题目对 WLAN 的优化问题进行拆解，具体描述如下：

### 问题一：基于测试基本信息的 AP 发送机会影响分析及预测

针对 WLAN 网络实测数据集中的测试基本信息与 AP 发送机会的相关性进行分析，根据各个因素的影响性强弱进行排序。同时，根据不同的 AP 测试场景进行分类讨论，建立 AP 发送机会的预测模型，并讨论不同情况下建立通用预测模型的可行性。最终，对测试集中的模型输入信息进行预测。

### 问题二：基于 RSSI 与门限信息的（MCS, NSS）参数预测

根据问题一中 AP 发送机会分析的结果，筛选最多次数的（MCS, NSS）参数。基于实测数据中的基本测试信息，主要为 RSSI 与门限信息，建立（MCS, NSS）参数的预测模型，最终，对测试集中的模型输入信息进行预测。

### 问题三：基于测试基本信息与（MCS, NSS）参数的网络吞吐量预测

根据问题一与问题二的分析结果，讨论网络吞吐量受基本测试信息以及（MCS, NSS）参数的影响。基于以上的关键参数建立网络吞吐量预测模型，同时，考虑同频 AP 个数对模型的影响情况。最终，结合测试集中的模型输入与统计的数据帧真实（MCS, NSS）预测网络吞吐量。

## 2 基本假设与符号说明

### 2.1 基本假设

我们根据题目的描述，对以下的关键词和基本假设进行了总结。

**WLAN 吞吐量的定义：**节点单位时间内成功发送的比特数。

**BBS 与 AP 的绑定：**假设一个 BSS 场景中仅包含唯一的主要 AP 节点，并且每个 AP 节点和 STA 节点发送和接收数据不能同时发生。

**WLAN 传输机制：**假设本场景中同步传输和异步传输在稳定功率递增的情况下，不会同时发生。

**环境底噪忽略问题：**分析时，假设环境底噪可忽略，则 STA 的 SINR 为关联 AP 到该 STA 的 RSSI 与邻区 AP 到该 STA 的 RSSI 之差。

### 2.2 符号说明

缩写	英文释义	中文释义
AMC	Adaptive Modulation and Coding	自适应调制编码
AP	Access Point	无线接入点
ACK	Acknowledgement	确认
ACKTimeout	Acknowledgement Timeout	确认超时
BSS	Basic Service Set	基本服务集
CCA	Clear Channel Assessment	信道可用评估
CDF	Cumulative Distributed Function	累积分布函数
CTS	Clear to Send	允许发送
CW	Contention Window	竞争窗口
DCF	Distributed Coordination Function	分布式协调功能
DIFS	DCF Inter-Frame Space	DCF 帧间距
ED	Energy Detection	能量检测
MAC	Medium Access Control	媒体控制
MCS	Modulation and Coding Scheme	调制编码方案
NAV	Network Allocator Vector	网络分配矢量
NSS	Number of Spatial Stream	空间流数
PD	Packet Detection	包检测
PER	Packet Error Rate	丢包率
PHY	Physical	物理层
RSSI	Received Signal Strength Indication	接收信号能量强度
RTS	Request to Send	请求发送
SIFS	Short Inter-Frame Space	短帧间距
SINR	Signal To Interference And Noise Ratio	信干噪比
STA	Station	站点
TCP	Transmission Control Protocol	传输控制协议
UDP	User Datagram Protocol	用户数据报协议
WLAN	Wireless Local Area Network	无线局域网

### 3 数据预处理

我们首先对 WLAN 的实测数据集进行了初步分析，并且将实测数据集分类按 2AP、3AP 场景下的训练集（training）、测试集（testing）进行分类，2AP 场景下的训练数据集共 5 个文件。3AP 场景下的训练数据集共 8 个文件。2AP 场景下的测试数据集共 2 个文件，3AP 场景下的测试数据集共 2 个文件。本题目的结果预测文件分类为 test\_set\_1\_\*用于问题一和问题三，test\_set\_2\_\*用于问题二。

#### 3.1 数据集基本信息分析

我们对所有数据文件所包含的维度信息进行分析，将各个维度的数据进行了大致分类，便于后续特征的选取，情况如下表 1 所示，结果表明，常量指标无法再同一个文件中提供有效的分类信息，而测量统计指标由于题目输入的限制，主要作为模型训练和输出结果精度的分析特征。最后，变量指标与 RSSI 指标可以提供有效的分类信息，可以进一步观察，并通过数据清洗作为本题目模型训练的关键输入特征。

表 1 数据集维度信息分析结果

特征类别	特征名称
常量类	test_dur, pkt_len, pd, ed, seq_time
测量统计值	nss, mcs, per, num_ampdu, ppdu_dur, other_air_time, seq_time, throughput
变量类	loc_id, protocol, nav, rssi
RSSI-AP 相关	ap_from_ap_*_sum_*, ap_from_ap_*_max_*, ap_from_ap_*_mean_*, sta_to_ap_*_sum_*, sta_to_ap_*_max_*, sta_to_ap_*_mean_*, sta_from_ap_*_sum_*, sta_from_ap_*_max_*, sta_from_ap_*_mean_*,
RSSI-STA 相关	sta_from_sta

对于 RSSI 的相关特征，每种均有 sum、max、mean 三种类别，且每类指标是时序数据，并且存在数据长度不固定的问题。另一方面，我们对测试集进行了观察，确认测试数据集中测量统计指标的情况，例如，test\_set\_1\_\*的文件中含有 nss, mcs, per 的数据，可以用于问题三的求解。

根据以上的分析，我们将训练数据集文件分为 2AP、3AP 两类情况进行建模和讨论。并且每个问题的模型求解设置如下：

问题一：protocol, nav, rssi 作为模型输入，seq\_time 为预测标签。

问题二：protocol, nav, rssi 作为模型输入，(nss, mcs)的组合为预测标签。

问题三：protocol, nav, rssi, phy 作为模型输入，throughput 为预测标签。需要注意的是，此处 phy 是通过（MCS, NSS）与 PHY Rate 对应表进行映射和提取。

最后，模型在训练数据集上的划分为，训练集（80%）：测试集（20%）。

#### 3.2 缺失值处理

我们首先使用 python 的数据分析库 pandas.info()查看各文件中数据的缺失值情况，如空值和缺失值。实验结果发现训练集和测试集中均有缺失值的出现，因此，对该列中包

含具体数据的情况，我们采用均值函数（mean）对缺失部分进行填充，而对于该列全部为空值的数据，我们使用 0 值进行填充。其中，对于流量类型（protocol）的缺失，因无法判定属于类别，我们将这条数据进行过滤和舍弃。此外，对于训练集中预测标签缺失的情况，同样舍弃该类数据，例如，训练集：training\_set\_2ap\_loc2\_nav82.csv 文件中 test\_id=40 的点缺失了预测标签，故舍弃此条数据。

### 3.3 异常值处理

我们使用 Z-Score 的方法进行异常值的筛选，在本文的实验中，我们选取  $3\sigma$  为阈值，筛选异常值。

$$Z = \frac{X - \mu}{\sigma}$$

通过利用以上的异常值查找方法，我们发现对于输入集：未发现异常值。对于标签集，在 training\_set\_2ap\_loc2\_nav82.csv 文件中 test\_id=39 的 seq\_time 为 0.4，被视为异常点，故舍弃此数据。

### 3.4 指标归一化

由于不同模型训练的需要，如神经网络模型，我们考虑将数据进行统一的归一化。对于输入集，我们采用 MinMax 方法将所以数值类型的数据归一化到 [0,1]，消除特征量纲的影响，反之，预测的标签集则无需归一化。同时，针对 RSSI、流量类型（Protocol）、（NSS，MCS）的编码进行了特殊处理：

（1）**RSSI 时间序列单值化**：由于 RSSI 类型为时间序列，长度不固定，且某些列为单个值。本文求均值将此指标单值化。

（2）**流量类型（Protocol）的编码**：由于流量类型为两个 TCP 与 UDP，我们通过独热（One-Hot）编码转为 [1,0]、[0,1] 的形式，加入到模型输入的向量中。

（3）**NSS，MCS 的编码**：通过观察，我们发现训练数据集中，两者的组合能够枚举，因此，我们将 NSS 和 MCS 的组合统一合并为一个标签 NSS\_MCS，最后，我们使用 LabelEncoder 编码转为整数，方便模型训练预测。在模型预测结束后，再统一进行解码，并拆分为 NSS 和 MCS。

### 3.5 数据集增强

我们针对 2AP 场景和 3AP 场景进行分类建模，在进行数据集划分后，我们发现对应场景的数据集较小，并且在分类场景中呈现出较大的样本不平衡性。针对样本过少与不平衡的问题，我们采用了数据增强的方法丰富原始数据集，并且减缓样本不平衡带来的问题。数据增强的方法具体如下：

**增强数据的选择**：根据算法需要进行预测的标签类进行提取，如本题目中需要进行预测的发送序列时长（seq\_time）、网络吞吐量（Throughput）、NSS 和 MCS 的组合。

**数据增强方法**：针对数值标签（seq\_time，Throughput），我们统计了两类数据在训练数据集中的分布情况，并计算数据之间的最小差值。由于这个差值决定了类别发生改变的最小值域  $\Delta\delta$ 。因此，在考虑到不改变原始数值类别发生变化的情况下，我们采用  $\Delta\delta * 0.1$  作为原始数据的增量幅值扩充数据集大小。我们依据如下公式将一行数据经过增

幅后，转变为多行数据。新增行数的大小与增幅的次数有关，通过多次的实验测试，最终我们选择了效果较优时相比原始数据集 5 倍的扩展大小。

$$line_{\{new\}} = line_{\{old\}} \pm \Delta\delta * 0.1$$

此外，通过对 NSS 和 MCS 分类模型的调试，我们发现数据集存在较大的不平衡性，例如 (NSS, MCS) = (2,11) 在数据集中频繁出现，考虑到数据增幅将会扩大该不平衡性，于是通过减少该类数据的增幅次数以达到相对的样本集平衡。



## 4 问题一的模型建立与求解

### 4.1 问题分析

本赛题提供的 WLAN 组网场景的拓扑图如下图 1 所示，同时，数据集采集的同频 AP 数量为 2 或 3。根据题目描述，数据集字段的定义如下：

**测试基本信息（模型输入）：**网络拓扑、业务流量、门限、节点间 RSSI

**数据帧统计信息（模型输出）：**各节点发送时长、（MCS, NSS）、PER、吞吐量

针对问题一，我们将其分为两个阶段进行处理：第一阶段，解决影响性强弱的排序问题，对实测数据集进行观察和分析，对数据集中参数的影响方式进行分类，并讨论其影响的模式，从而建立影响性分析模型，实现影响性强弱的排序；第二阶段，根据提取出的关键影响参数，针对 2 个 AP 的场景和 3 个 AP 的场景分别建立预测模型，实现 AP 发送机会的预测。

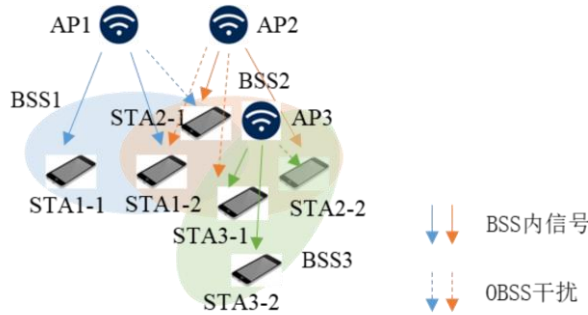


图 1 WLAN 组网场景的测试拓扑

### 4.2 影响因子相关性分析

我们首先对实测数据集中的测试基本信息进行分析，包括网络拓扑信息，业务流量信息，门限信息以及节点间 RSSI 信息。通过观察数据的趋势和形态，我们初步得到了如表 2 的影响因子与 AP 发送机会的相关性分析结果，由于对原始数据集的观察较为丰富，详细的分类依据见附件 1，后文仅对本文选取的关键影响因子进行详细讨论。

表 2 影响因子与 AP 发送机会的相关性分析结果

影响因子 (测试基本信息)	相关性	名称	相关性分析
eirp	高	发送功率	AP 发送功率影响 AP 间发生干扰的阈值
loc_id	高	部署位置	部署位置对 AP 发送机会影响较高
nav	高	NAV 门限	NAV 门限会影响 AP 间发生干扰的阈值
protocol	中	流量类型	UDP 或 TCP 的 AP 发送机会会有明显差异
RSSI_ap_from_ap	低	AP 间 RSSI	AP 互相监听的 RSSI, AP 位置相关
RSSI_sta_to_ap	低	AP 监听 STA	AP 监听 STA 的 RSSI, AP 位置相关
RSSI_sta_from_ap	低	STA 监听 AP	STA 监听 AP 的 RSSI, AP 位置相关
RSSI_sta_from_sta	低	STA 监听 STA	STA 监听 STA 的 RSSI, STA 位置相关
test_id	无	测试编号	用于标识测试

bss_id	无	BSS 编号	用于标识 BSS 场景，与 AP 绑定
ap_name, ap_id	无	AP 标识	用于标识 AP
sta_mac, sta_id	无	STA 标识	用于标识 STA，接收设备不影响发送
test_dur	无	测试时间	固定常数
pkt_len	无	数据长度	固定常数
pd	无	包门限	固定常数
ed	无	能量门限	固定常数

在上表的分析结果中，用于标识数据唯一性与固定常数的影响因子在实测数据集中无法提供足够的相关性信息，因此我们在影响性排序的过程中将会进行排除。其次，部署位置（loc\_id）、NAV 门限、发送功率（eirp）、流量类型（protocol）与 AP 的发送机会具有较强的相关性，同时，经过我们的分析，RSSI 可以作为 AP 与 AP，AP 与 STA 间信号强度或相对位置的判断依据。下面我们将会对这些影响因子进行详细的影响类型和程度的分析。

**发送功率（eirp）：**通过观察实测数据集，我们发现每个测试记录文件中发送功率随着实验次数递增，于是我们将实验按照测试顺序进行组合，同时，固定流量类型和两个 AP 与三个 AP 交互的场景，对发送功率与 AP 发送机会进行可视化分析，分析结果如下图所示 2 和图 3 所示。横坐标为随实验次数递增的序号，纵坐标分别表示 AP 发送机会与发送功率，这里我们将两个数值进行了归一化便于分析。结果表明：

（1）随着发送功率的增大，AP 发送机会在达到 NAV 门限的阈值后，将会出现 AP 间互相干扰的情况，因此在发送功率达到一定的阈值时，AP 的发送机会出现明显下降。

（2）TCP 模式相比于 UDP 模式其发送机会的波动性更大。

因此，发送功率与 AP 发送机会的相关性较高，作为我们主要关注的影响因子。

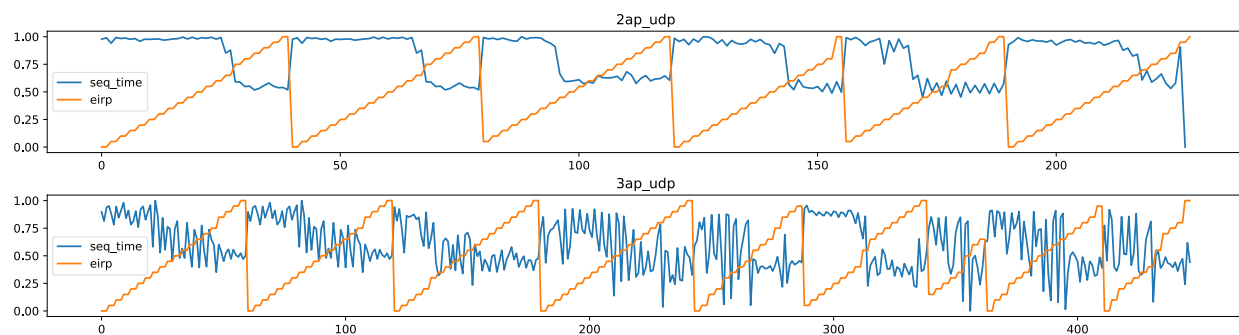


图 2 UDP 模式下发送功率与 AP 发送机会的可视化结果

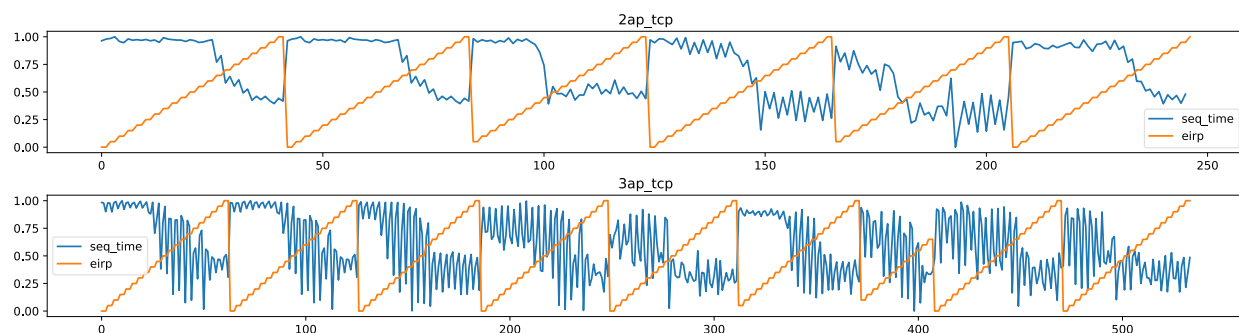


图 3 TCP 模式下发送功率与 AP 发送机会的可视化结果

**部署位置 (loc\_id)**：针对不同的部署位置，我们固定了 NAV 门限、流量类型、以及 AP 个数，对每个 AP 在固定场景下受到部署位置的影响进行了可视化分析，分析结果如下图 4 和图 5 所示。横坐标为逐渐递增的发送功率，纵坐标为 AP 发送机会。结果表明：

(1) 在两个 AP 交互的场景下，部署位置对 AP 发送机会的影响较小，并且可以观察到 loc1 的位置发送机会明显较 loc0，loc2 偏低。

(2) 在三个 AP 交互的场景下，部署位置对 AP 发送机会的影响较大，使其产生了更复杂的波动情况。这在 loc33，loc31 表现的更加明显。

(3) TCP 模式相比 UDP 模式受到的影响更小，具有更强的稳定性。

因此，部署位置与 AP 发送机会的相关性较高，作为我们主要关注的影响因子。

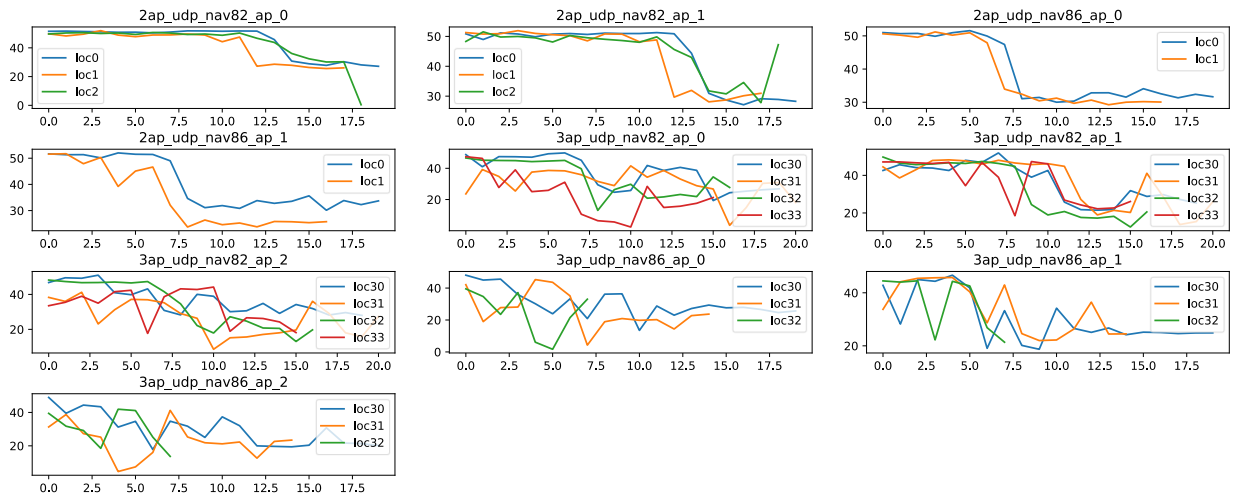


图 4 UDP 模式下不同部署位置对 AP 发送机会的影响

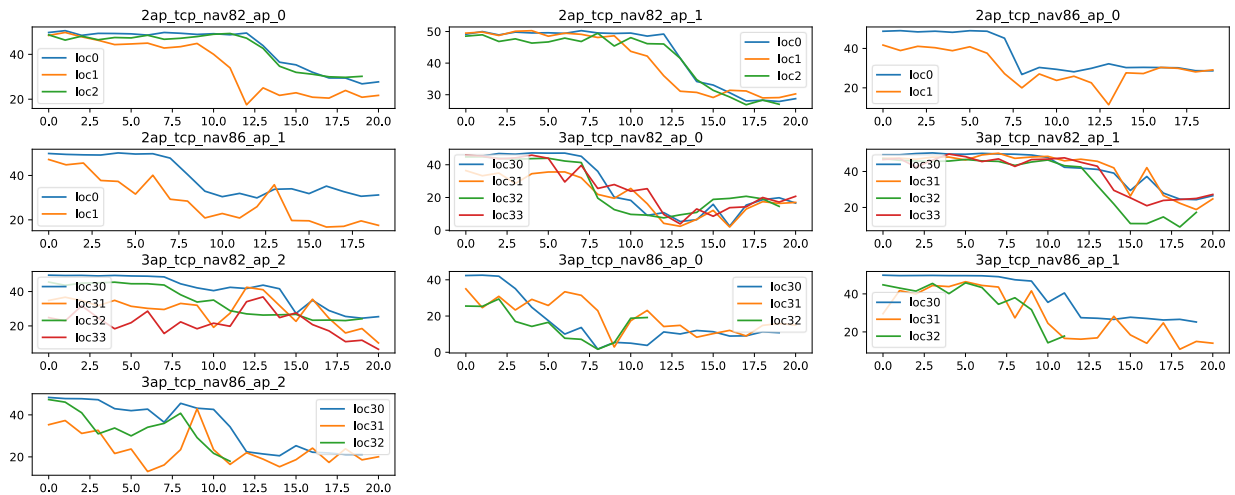


图 5 TCP 模式下不同部署位置对 AP 发送机会的影响

**NAV 门限 (nav)**：针对不同的 NAV 门限，我们固定了部署位置与流量类型，对每个 AP 在固定场景下受到 NAV 门限的影响进行了可视化分析，分析结果如下图 6 和图 7 所示，横坐标为逐渐递增的发送功率，纵坐标为 AP 发送机会。结果表明：

(1) NAV 门限是 NAV 机制触发的关键因素，当 AP 接收到相邻 BSS 的数据帧时，将会使用 RSSI 与 NAV 门限判断是否接收数据。在实测数据集中，随着发送功率的升高，AP 间发生数据传输和干扰的概率将会变大。而较低的 NAV 门限（数据集中为-82dBm 与-86dBm），即 nav86 所标注的 AP 发送机会曲线会在更低的功率时触发 AP 间的 NAV 机制，从而对 AP 发送机会产生影响。

(2) TCP 模式相比 UDP 模式呈现出更加稳定的趋势。

因此，NAV 门限与 AP 发送机会的相关性较高，作为我们主要关注的影响因子。

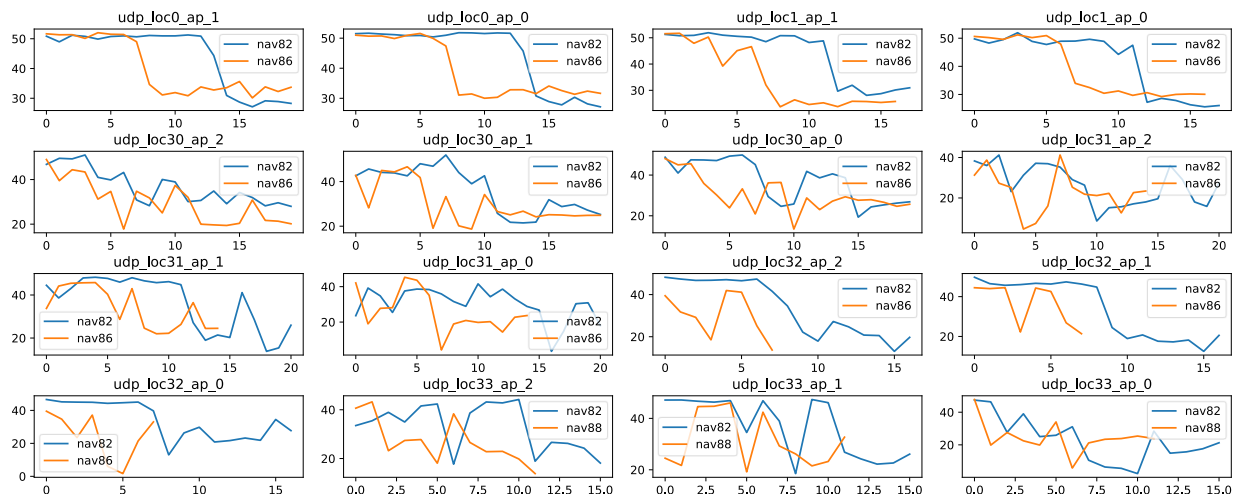


图 6 UDP 模式下不同 NAV 门限对 AP 发送机会的影响

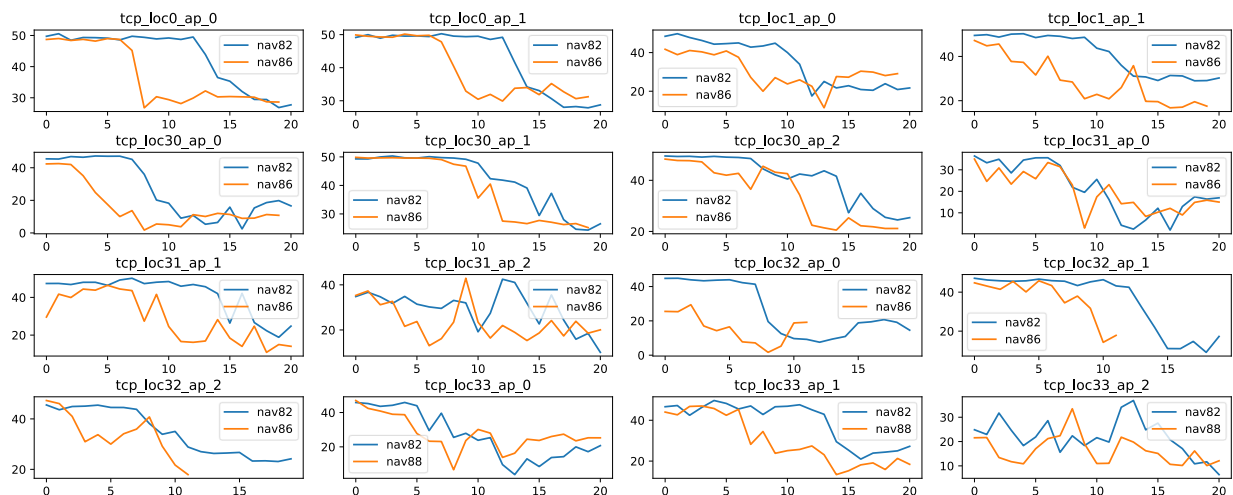


图 7 TCP 模式下不同 NAV 门限对 AP 发送机会的影响

**流量类型 (protocol)：**通过以上的分析，可以明显的观测到 TCP 模式与 UDP 模式在 AP 发送机会的稳定性上具有明显的差异。针对不同的流量类型，我们固定了部署位置与 NAV 门限，对不同流量类型的 AP 发送机会随发送功率的变化进行了可视化分析，结果如下图 8 所示。结果表明：

(1) UDP 模式相比于 TCP 模式更容易受到其他影响因素的影响，呈现出更明显的波动性。

(2) UDP 和 TCP 在发送功率达到一定阈值时，会出现明显的下降，这说明功率达到 AP 的 NAV 门限后，AP 间出现明显的互相干扰。

因此，流量类型与 AP 发送机会的相关性较高，作为我们主要关注的影响因子。

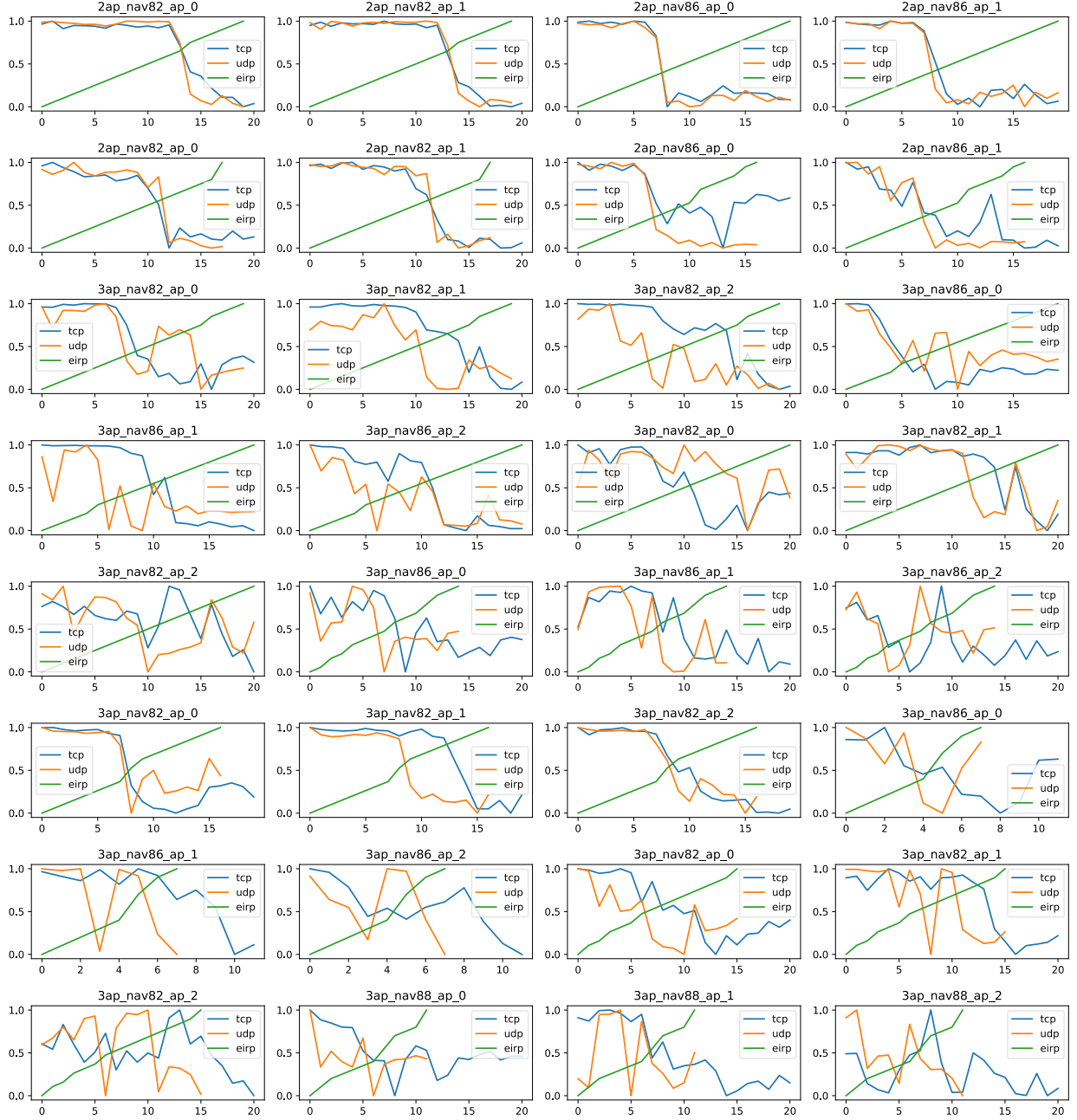


图 8 不同的流量类型对 AP 发送机会的影响

**RSSI 信号：**我们对数据集中的 RSSI 相关列进行分析，发现 RSSI 表达了两个节点（设备）间的信号强度，因此，我们将其视为设备间的距离因素。如下图 9 所示，基于数据集中的设备拓扑提取，主要分为两种交互场景：

（1）当有两个 AP 与两个 STA 设备时，如图 9 左侧，以 AP\_0 为主观测视角时的 RSSI 交互拓扑。结果表明，AP\_0 的主要与 STA\_0 发生交互，同时会受到 AP\_1 的信号干扰，同时，考虑 STA\_1 的交互信息，可以将该场景视为 AP\_0 的空间位置信息，因此我们选用了图中的几条边的 RSSI 数据作为主要输入特征。



(2) 当有三个 AP 和三个 STA 设备时，如图 9 右侧，以 AP\_0 为主观测视角时的 RSSI 交互拓扑。结果与两个 AP 的场景类似，我们选用了与之相邻的 AP 节点，以及 STA\_0 相邻的设备的 RSSI 数据作为主要输入特征。

因此 RSSI 信号与 AP 发送机会具有一定的相关性，作为我们主要关注的影响因子。

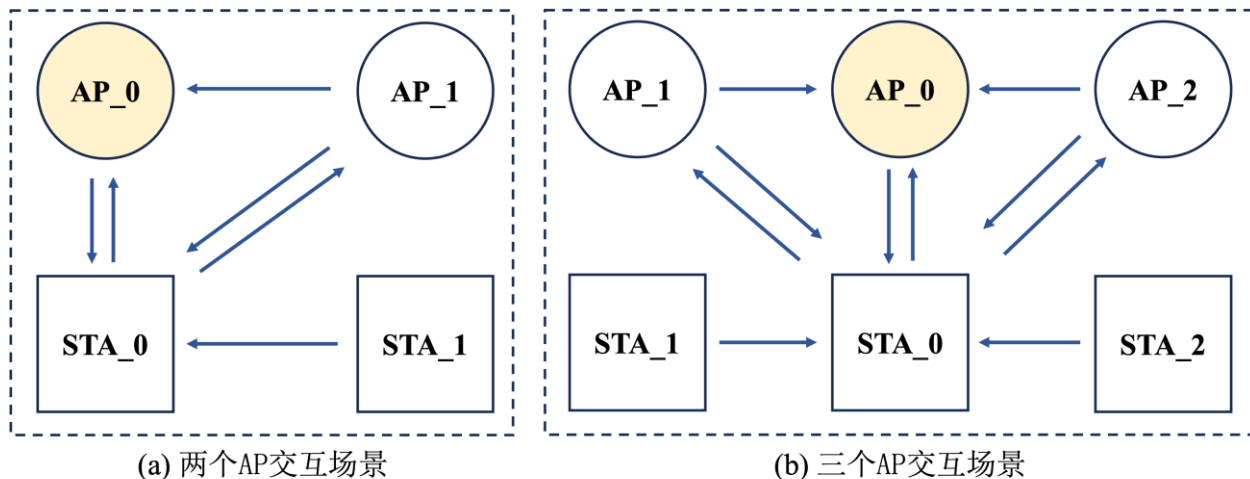


图 9 RSSI 信号数据场景建模

### 4.3 影响性强弱排序模型

根据前面小节的分析，我们选取了与 AP 发送机会相关性最大的影响因子进行影响性强弱的排序，分别是部署位置 (loc\_id)，NAV 门限 (nav)，发送功率 (eirp)，流量类型 (protocol)，RSSI。

**影响性强弱的定义：**根据对数据变化趋势的观察，我们发现 AP 的传输方式反映了 AP 间交互和干扰的情况。因此，我们将影响性强弱定义为影响因子使 AP 传输方式发生改变的临界点。

由于每个独立测试中，发送功率 (eirp) 为单调递增的变量，而发送功率的增加代表着 AP 间信号的增加，越高的功率将会使得 AP 间发生交互的概率增加。因此，我们通过寻找实测数据集中 AP 发送机会的明显转折点，统计 AP 发送机会变化的临界功率值，并记录部署位置 (loc\_id)、NAV 门限 (nav)、发送功率 (eirp)、流量类型 (protocol) 和 RSSI，表 3 展示了部分的统计数据，其中 RSSI 选用 ap\_from\_ap\_x\_mean 的均值作为输入特征，其他特征则选用测试场景标注的信息。

表 3 发送功率临界点统计信息

protocol	loc_id	NAV	RSSI	erip
tcp	loc0	82	85.02702703	21
udp	loc0	82	85.02702703	21
tcp	loc0	86	87.75	15
udp	loc0	86	87.75	15
tcp	loc1	82	83.54651163	18
udp	loc1	82	80.59756098	21
tcp	loc1	86	86.71428571	15
udp	loc1	86	86.71428571	15

tcp	loc2	82	84.30588235	21
udp	loc2	82	84.45348837	20
tcp	loc30	82	82.85	17
udp	loc30	82	87.21568627	16
tcp	loc30	86	81.27692308	18
udp	loc30	86	88.85245902	15
tcp	loc31	82	81.27692308	19
udp	loc31	82	84.01639344	20
tcp	loc31	86	78.95	18
udp	loc31	86	86.09090909	16
tcp	loc32	82	81.04038005	21
udp	loc32	82	86.54545455	16
tcp	loc32	86	84.19469027	17
udp	loc32	86	85.34939759	16
tcp	loc33	82	79.15776699	22
udp	loc33	82	83.9516129	19
tcp	loc33	88	85.07334963	15
udp	loc33	88	88.45514223	15

**影响性强弱的排序：**根据上表的统计信息，我们使用主成分分析法（PCA）对这些影响因子进行了分析。表 4 展示了这些影响因子在每个主成分中的贡献情况，表 5 展示了每个主成分解释的方差比例。

**表 4 影响因子主成分分析的贡献度统计**

影响因子	PC1	PC2	PC3	PC4	PC5
protocol	0.2829	0.5022	-0.6761	-0.4407	0.1276
loc_id	0.0345	0.6331	0.7027	-0.3003	0.1177
NAV	-0.4954	0.4538	-0.1911	0.6035	0.3844
RSSI	-0.5601	-0.3195	-0.0102	-0.5564	0.5238
eirp	0.5995	-0.1969	0.1112	0.2042	0.7401

**表 5 影响因子主成分解释的方差比例**

主成分	解释方差比例
PC1	0.4706
PC2	0.2458
PC3	0.1922
PC4	0.0614
PC5	0.0298

通过以上分析，结合我们在 AP 发送机会的预测模型测试结果（即 RSSI 与 AP 发送机会的预测具有更强的相关性这一结论，我们将在下一小节进行解释）。我们认为 RSSI

是影响最大的因素，其次，eirp 在主成分分析中占比最大，因此排在第二。NAV 在 PC1 与 PC2 中的加权贡献值大于 loc\_id，排在第三。Loc\_id 在第二个成分中排名第一，排在第四。Protocol 则排在末尾。因此，我们认为最终的影响性强弱的关系为：

$$\text{RSSI} > \text{eirp} > \text{NAV} > \text{loc\_id} > \text{protocol}$$

#### 4.4 特征选择

本文依据特征影响强弱排序选择 RSSI、eirp、NAV、protocol 作为问题输入集，由于 RSSI 种类较多，不同的 RSSI 均含 sum、max、mean 三类。mean\_RSSI 为所有天线均值，较为稳定，且可以衡量 WLAN 网络拓扑的相对位置，考虑为特征。sum\_RSSI 和 max\_RSSI 分别为所有天线总和与最大值，不确定能否考虑为特征。本文针对 2AP、3AP 均选用 XGBoost 为基准，分别对特征做消融实验分析，结果如下。最终只选用 mean\_RSSI 单类特征，在特征维度和拟合效果上均较优。

表 6 2AP RSSI 消融实验均方误差对比

特征	特征数量	均方误差（MSE）
mean_RSSI	6	6.37
mean_RSSI +sum_RSSI	11	8.59
mean_RSSI +max_RSSI	11	13.85
mean_RSSI +sum_RSSI + max_RSSI	16	7.42

表 7 3AP RSSI 消融实验均方误差对比

特征	特征数量	均方误差（MSE）
mean_RSSI	10	39.52
mean_RSSI +sum_RSSI	18	43.87
mean_RSSI +max_RSSI	18	51.36
mean_RSSI +sum_RSSI + max_RSSI	26	41.33

#### 4.5 发送机会预测模型

根据前两个小节分析，我们选用了具有相关性的特征进行预处理，作为输入特征建立预测模型。同时，由于观测到测试数据集（test\_set\_1\_\*, test\_set\_2\_\*）中的部署位置与训练数据集中不一致，所以排除了部署位置这个输入特征。因此，我们最终选用了 NAV 门限（nav），发送功率（eirp），流量类型（protocol），RSSI 作为模型的输入特征。AP 发送的序列时间（seq\_time）作为模型预测的输出。

**特征预处理：**我们首先对选用的特征进行了预处理，其中 NAV 门限为每个实验的固定值，流量类型为 TCP 和 UDP 的二分类数据，进行 0/1 编码，发送功率则采用原始数值。针对 RSSI，我们将其认为是节点间距离的表达，因此我们选用了 AP 与 AP 间，AP 与 STA 间的 RSSI 均值作为输入特征。

**模型建立：**由于选择的输入特征为多维数据，是典型的数据预测问题。因此，我们选择了几个通用的基于机器学习和逻辑回归的预测模型进行对比。分别是线性回归、XGBoost、随机森林、支持向量回归、多层感知机。同时，由于观测到两个 AP 与三个 AP 交互场景下数据的差异较大，我们分别设计了 2AP 和 3AP 的预测模型。



#### 4.5.1 线性回归

线性回归的目标是通过寻找最优的线性函数，使得自变量对因变量的预测最为准确。线性回归通过最小化残差平方和（Sum of Squared Errors, SSE）来求解模型的系数 $\beta_0$ 和 $\beta_1$ 。残差是指预测值和真实值之间的差距：

$$\begin{aligned}\hat{y}_i &= \beta_0 + \beta_1 x_i \\ error &= y_i - \hat{y}_i\end{aligned}$$

目标是最小化以下目标函数：

$$SSE = \sum_{i=1}^n (y_i - \hat{y}_i)^2 = \sum_{i=1}^n (y_i - (\beta_0 + \beta_1 x_i))^2$$

通过对 $\beta_0$ 和 $\beta_1$ 求导并令导数为零，得到线性回归的解析解：

$$\beta_1 = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2}$$

其中 $\bar{x}$ 和 $\bar{y}$ 分别是自变量 $x$ 和因变量 $y$ 的平均值。

$$\beta_0 = \bar{y} - \beta_1 \bar{x}$$

多元线性回归的形式则如下

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p + \epsilon$$

或者更为简洁地表示为：

$$y = \mathbf{X}\boldsymbol{\beta} + \epsilon$$

最小化平方误差的解析解是通过矩阵形式给出：

$$\boldsymbol{\beta} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}$$

#### 4.5.2 XGBoost

XGBoost（Extreme Gradient Boosting）是一种高效的梯度提升树（Gradient Boosting Tree, GBDT）算法，被广泛应用于分类和回归任务。在每一轮迭代中，XGBoost会构建一个新的决策树作为弱学习器。每个树的输出被加权后与之前的预测结果相加，以得到新的预测值，而优化目标则是通过最小化损失函数来评估模型的性能。作为梯度提升树模型的一种，XGBoost基本原理结合多个弱学习器，也就是决策树，可以表示为：

$$F(x) = \sum_{i=1}^t f_i(x)$$

其中 $f(x_i)$ 是第 $i$ 棵树。损失函数加入了正则化项，减少模型复杂度，防止过拟合

$$L(y, F(x)) = \sum_{i=1}^N l(y_i, F(x_i)) + \sum_{i=1}^t \Omega(f_i)$$

XGBoost与其他模型不同的优化是，XGBoost使用损失函数的一阶和二阶导数：

$$\begin{aligned}g_i &= \frac{\partial l(y_i, F(x_i))}{\partial F(x)} \\ h_i &= \frac{\partial^2 l(y_i, F(x_i))}{\partial F(x)^2}\end{aligned}$$

在构建决策树时，XGBoost 通过最小化每个叶子节点的损失和正则化项的总和，选择最优的分裂点（其中  $\lambda$  是正则化参数）：

$$\text{Gain} = \frac{1}{2} \left( \frac{g^2}{h + \lambda} - \frac{G^2}{H + \lambda} \right)$$

#### 4.5.3 随机森林

随机森林（Random Forest）是一种结合了多个决策树的模型，决策树是一种树形结构的模型，其中每个内部节点代表特征的测试，分支代表测试结果，叶节点代表类别或数值预测。决策树通过选择最佳特征进行分裂来减少数据集的不纯度，通常使用基尼系数或者信息增益来评估特征的选择。随机森林基于 Bagging（Bootstrap Aggregating）集成学习方法，通过并行训练多个模型并对其结果进行平均结合。

随机森林首先从原始数据集中随机抽取多个样本集，每个样本集用于训练一棵决策树。对于每棵树的每个节点，随机森林从所有特征中随机选择一部分特征进行分裂。对于选取的特征，随机森林使用基尼系数或信息增益来选择最佳特征进行节点分裂。基尼系数定义如下：

$$\text{Gini}(D) = 1 - \sum_{i=1}^C p_i^2$$

其中  $p_i$  指的是第  $i$  类在数据集  $D$  中的比例。信息增益定义如下：

$$\text{IG}(D, A) = \text{Entropy}(D) - \sum_{v \in A} \frac{|D_v|}{|D|} \text{Entropy}(D_v)$$

$$\text{Entropy}(D) = - \sum_{i=1}^C p_i \log_2(p_i)$$

随机森林会重复上述操作，直到达到参数预设的树的数量。集成时，对每棵树的预测结果进行集成。对于分类任务，随机森林采用多数投票定最终类别。对于回归任务，随机森林则采取所有树的预测值的平均。

#### 4.5.4 支持向量回归

支持向量回归（Support Vector Regression, SVR）是支持向量机（SVM）的回归版本，基本原理是找到一个超平面，使预测误差在一定的容忍范围内，同时保持模型的复杂度尽可能低。同时，SVR 在高维空间中通过核函数的作用可以处理非线性问题。具体来说，SVR 试图找到一个函数满足以下条件：

$$|y_i - f(x_i)| \leq \epsilon$$

$f(x)$  一般为线性函数， $w$ 、 $x$ 、 $b$  分别代表权重向量，输入向量和偏置：

$$f(x) = w \cdot x + b$$

为了使 SVR 模型尽可能简单，SVR 要在约束条件下最小化  $w$  的大小，通过最小化正则化项来实现：

$$\min \frac{1}{2} \|w\|^2$$

在满足约束条件下：

$$\begin{aligned} y_i - (w \cdot x_i + b) &\leq \epsilon \\ (w \cdot x_i + b) - y_i &\leq \epsilon \end{aligned}$$

如果某些数据点的误差超出了 $\epsilon$ ，我们引入松弛变量 $\xi_i$ 和 $\xi_i^*$ 来表示超出的误差，这时优化问题变为：

$$\begin{aligned} \min \quad & \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n (\xi_i + \xi_i^*) \\ \text{subject to: } & y_i - (w \cdot x_i + b) \leq \epsilon + \xi_i \\ & (w \cdot x_i + b) - y_i \leq \epsilon + \xi_i^* \\ & \xi_i, \xi_i^* \geq 0 \end{aligned}$$

$C$  是惩罚系数，控制误差和模型复杂度的权衡； $\xi_i$ 和 $\xi_i^*$ 表示数据点超过 $\epsilon$ 的误差。为求解该优化问题，SVR 引入拉格朗日乘子并构建拉格朗日函数。拉格朗日函数表示为：

$$L(w, b, \xi, \xi^*, \alpha, \alpha^*, \eta, \eta^*) = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n (\xi_i + \xi_i^*) - \sum_{i=1}^n \alpha_i (\epsilon + \xi_i - y_i + w \cdot x_i + b) - \sum_{i=1}^n \alpha_i^* (\epsilon + \xi_i^* + y_i - w \cdot x_i - b)$$

通过求解这个拉格朗日函数的对偶问题，可以得到最终的模型。在实际应用中，输入数据往往是非线性的，此时通过核函数将数据映射到高维空间中，使问题在高维空间中变为线性可分。核函数用于计算输入数据在高维空间中的内积，从而避免了直接计算映射后的高维特征向量。因此，SVR 的函数形式为：

$$f(x) = \sum_{i=1}^n (\alpha_i - \alpha_i^*) K(x_i, x) + b$$

其中， $\alpha_i$  和  $\alpha_i^*$  是从拉格朗日对偶问题中得到的拉格朗日乘子。最终，SVR 的预测函数为：

$$f(x) = \sum_{i=1}^n (\alpha_i - \alpha_i^*) K(x_i, x) + b$$

支持向量是那些 $\alpha_i$  不为 0 对应的数据点，即这些数据点决定了模型的超平面。

#### 4.5.5 多层感知机

多层感知机(MLP)的原理是寻找类别间最合理、最具有鲁棒性的超平面，最具代表的感知机是 SVM 支持向量机算法。神经网络同时借鉴了感知机和仿生学，通常来说，动物神经接受一个信号后会发送各个神经元，各个神经元接受输入后根据自身判断，激活产生输出信号后汇总从而实现对信息源实现识别、分类，一个典型的神经网络如下图所示：

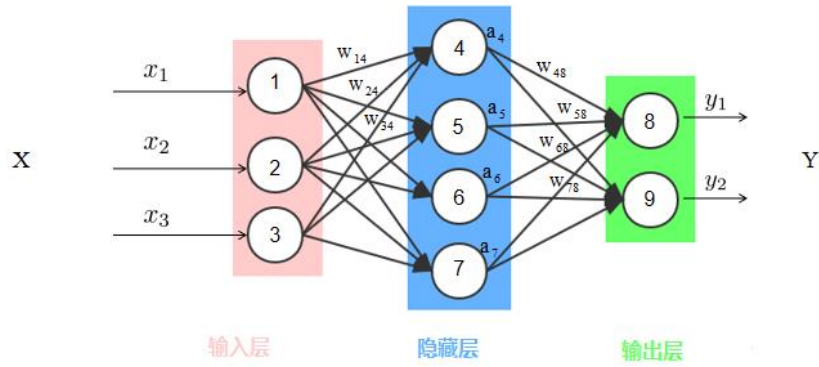


图 10 典型的多层感知机传播示意图

神经元的计算过程可以表示为：

$$z = \sum_{i=1}^n w_i x_i + b$$

$$a = f(z)$$

激活函数引入非线性，使得神经网络能够表示复杂的模式。常见的激活函数包括 ReLU、Sigmoid、SoftPlus、tanh 函数等。之后，再经过前向传播，从输入层传递信号到输出层的过程，模型通过前向传播计算预测值。为了衡量模型的预测值和真实值之间的差距，神经网络引入损失函数。常见的损失函数有均方差误差或者交叉熵。

神经网络最有特色的是反向传播。反向传播是训练神经网络的重要步骤，通过计算梯度并使用梯度下降法更新权重，来最小化损失函数。首先，计算输出层的误差：

$$\delta^{(2)} = \hat{y} - y$$

反向传播从输出层逐层向回传播误差，计算隐藏层的梯度：

$$\delta^{(1)} = (\delta^{(2)} W^{(2)}) \odot f'(z^{(1)})$$

权重更新利用梯度下降法更新每一层的权重和偏置。对于权重  $W$  的更新：

$$W^{(l)} = W^{(l)} - \eta \frac{\partial L}{\partial W^{(l)}}$$

其中  $\eta$  是学习率，求导项是损失函数对权重的梯度。类似地，更新偏置  $b$ ：

$$b^{(l)} = b^{(l)} - \eta \frac{\partial L}{\partial b^{(l)}}$$

#### 4.5.6 自编码器

自编码器（Autoencoder, AE）是一种无监督学习的神经网络模型，旨在学习数据的紧凑表示（编码）以及从这种表示重建数据的能力（解码）。自编码器的核心思想是通过压缩（编码）和重构（解码）的过程，逼近输入数据的本质结构。

自编码器的基本结构是一个对称的神经网络。其中编码器负责将输入数据转换为低维的表示，而解码器负责从低维的表示重建输入数据。自编码器示意图如图 11 所示

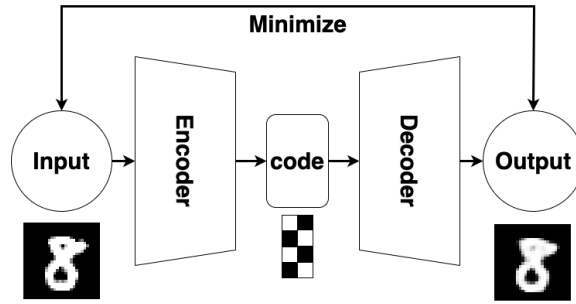


图 11 自编码器示意图

编码器的作用可以用公式表示为：

$$\mathbf{z} = f_{\text{enc}}(\mathbf{x}) = \sigma(W_{\text{enc}}\mathbf{x} + \mathbf{b}_{\text{enc}})$$

解码器的作用可以用公式表示为：

$$\hat{\mathbf{x}} = f_{\text{dec}}(\mathbf{z}) = \sigma(W_{\text{dec}}\mathbf{z} + \mathbf{b}_{\text{dec}})$$

自编码器的优化函数为重构与输入的误差，可以表示为：

$$L(\mathbf{x}, \hat{\mathbf{x}}) = \|\mathbf{x} - \hat{\mathbf{x}}\|_2^2$$

通过最小化该损失函数，自编码器可以学习到使得输入数据能够有效重建的表示。

与神经网络一样。自编码器的训练使用梯度下降或其变体（如 Adam）来最小化损失函数，权重和偏置通过反向传播算法更新。

#### 4.5.7 变分自编码器

变分自编码器（Variational Autoencoder, VAE）是一种自编码器，它结合了概率生成模型，其目标是学习复杂数据的潜在分布，并从中进行采样生成新数据。VAE 引入了概率论的思想，通过学习数据的潜在变量（latent variables）的分布，能够实现更为灵活和鲁棒的生成模型。变分自编码器的示意图如图 12 所示

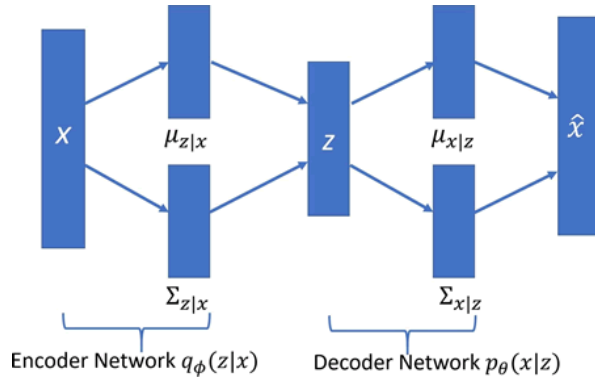


图 12 变分自编码器示意图

与传统自编码器相似，VAE 的结构仍然包含编码器和解码器两个部分。不同的是，编码器的输出不是固定的隐含表示，而是一个概率分布的参数。具体来说，编码器会输出潜在变量的均值和方差，并从该分布中进行采样，解码器则根据采样的潜在变量来重建数据。

VAE 假设输入数据  $\mathbf{x}$  由潜在变量  $\mathbf{z}$  生成，并且这些潜在变量来自某个简单的先验分布  $p(\mathbf{z})$ （通常假设为标准正态分布  $\mathcal{N}(\mathbf{0}, \mathbf{I})$ ）。解码器则学习从潜在变量  $\mathbf{z}$  生成观测数据  $\mathbf{x}$  的条件分布  $p(\mathbf{x} | \mathbf{z})$ 。这一原理可以表示成：

$$\mathbf{z} \sim p(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$$

$$\mathbf{x} \sim p(\mathbf{x}|\mathbf{z})$$

由于无法直接计算真实的后验分布，变分自编码器引入变分推断方法，用一个参数化的分布  $q(\mathbf{z}|\mathbf{x})$  来近似后验分布。这个  $q(\mathbf{z}|\mathbf{x})$  就是我们通过编码器  $f_{\text{enc}}$  计算出来的。

假设  $q(\mathbf{z}|\mathbf{x})$  是正态分布  $\mathcal{N}(\mu(\mathbf{x}), \sigma^2(\mathbf{x}))$ ，编码器输出均值  $\mu(\mathbf{x})$  和对数方差  $\log(\sigma^2(\mathbf{x}))$ ，即：

$$q(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}; \mu(\mathbf{x}), \sigma^2(\mathbf{x})\mathbf{I})$$

VAE 的训练目标是最大化数据的对数似然  $\log p(\mathbf{x})$ 。由于直接优化这个对数似然较难，我们采用变分推断的方法，最大化证据下界（Evidence Lower Bound, ELBO），该下界定义为：

$$\log p(\mathbf{x}) \geq \mathbb{E}_{q(\mathbf{z}|\mathbf{x})} [\log p(\mathbf{x}|\mathbf{z})] - D_{\text{KL}}(q(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))$$

公式中第一项  $\mathbb{E}_{q(\mathbf{z}|\mathbf{x})} [\log p(\mathbf{x}|\mathbf{z})]$  是重构误差项，表示从潜在变量  $\mathbf{z}$  重构数据  $\mathbf{x}$  的对数似然期望，鼓励模型生成与真实数据相似的重构数据。第二项  $D_{\text{KL}}(q(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))$  是 KL 散度项，表示编码器输出的后验分布  $q(\mathbf{z}|\mathbf{x})$  与先验分布  $p(\mathbf{z})$  之间的差异，鼓励编码器生成接近先验分布的潜在变量。

VAE 的损失函数可以表示为：

$$L_{\text{VAE}} = \mathbb{E}_{q(\mathbf{z}|\mathbf{x})} [-\log p(\mathbf{x}|\mathbf{z})] + D_{\text{KL}}(q(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))$$

也就是通过最小化重构误差和 KL 散度来训练模型。

在 VAE 的具体实现中，VAE 使用了重参数化技巧（Reparameterization Trick）。具体来说，我们不直接从  $q(\mathbf{z}|\mathbf{x})$  中采样，而是通过均值和标准差来重参数化采样过程：

$$\mathbf{z} = \mu(\mathbf{x}) + \sigma(\mathbf{x}) \odot \epsilon$$

其中  $\epsilon \sim \mathcal{N}(0, \mathbf{I})$  是标准正态分布中的噪声， $\mu(\mathbf{x})$  和  $\sigma(\mathbf{x})$  分别为编码器的输出均值和标准差。这使得采样过程变得可微，可以通过梯度下降进行优化。

#### 4.5.8 模型参数设置

针对本文使用的模型，我们主要参考了模型的默认参数，具体数值如下表 8 所示。

表 8 预测模型的默认参数

模型名称	参数名称	参数含义	参数值
XGBRegressor	n_estimators	决策树的个数	100
	learning_rate	学习率	0.3
	max_depth	决策树最大深度	6
	booster	基学习器	gbtree
	objective	损失函数	reg:squarederror
RandomForestRegressor	n_estimators	决策树的个数	100
	criterion	评判函数	squared_error
	max_depth	最大深度	None
	min_samples_split	叶节点上最小样本数	3

	max_features	分裂考虑的最大特征数	sqrt
SVR	kernel	核函数	rbf
	C	正则化参数	1.0
	epsilon	容许的误差范围	0.1
	gamma	核函数中的系数	scale
MLPRegressor	hidden_layer_sizes	隐藏层信息	(64, 32, 16, 32)
	max_iter	最大迭代参数	1000
	alpha	L2 正则化参数	0.001
	activation	激活函数	relu
	solver	梯度下降法	adam
LinearRegression	fit_intercept	是否计算截距	True
	normalize	是否归一化	False
EncoderDecoderRegressor	hidden_size	隐藏层大小	128
	epoch	训练轮数	1000
	lr	学习率	0.001
VAERegressor	batch_size	批大小	8
	hidden_size	隐藏层大小	64
	latent_size	隐变量大小	4
	epoch	训练轮数	1000
	lr	学习率	0.001

#### 4.5.9 模型训练

我们将训练集按照 8:2 的比例进行划分，其中训练集为 80%，测试集为 20%，并使用均方误差（MSE）评估模型在测试数据集上的效果。但由于训练数据集的规模较小，机器学习拟合不充分，且神经网络模型容易出现过拟合现象，本文同时增强数据集进行预测，发现模型性能更好，且泛化能力强。最终原始数据集和增强数据集的训练精度如下表 9 和表 10 所示。

**表 9 2AP 发送机会预测模型均方误差对比**

数据集	模型名称	均方误差（MSE）
原始数据集	线性回归（LinearRegression）	29.23
	XGBoost 回归（XGBoostRegression）	10.87
	随机森林回归（RandomForestRegression）	15.28
	支持向量回归（SVR）	33.12
	多层感知机（MLP）	13.18
	自编码器（AE）	3.84
	变分自编码器（VAE）	6.37
增强数据集	线性回归（LinearRegression）	22.52
	XGBoost 回归（XGBoostRegression）	0.618
	随机森林回归（RandomForestRegression）	0.845
	支持向量回归（SVR）	17.08

多层感知机（MLP）	2.167
自编码器（AE）	2.114
变分自编码器（VAE）	23.18

表 10 3AP 发送机会预测模型均方误差对比

数据集	模型名称	均方误差（MSE）
原始数据集	线性回归（LinearRegression）	80.11
	XGBoost 回归（XGBoostRegression）	39.52
	随机森林回归（RandomForestRegression）	50.42
	支持向量回归（SVR）	73.81
	多层感知机（MLP）	39.69
	自编码器（AE）	31.14
	变分自编码器（VAE）	41.13
增强数据集	线性回归（LinearRegression）	66.35
	XGBoost 回归（XGBoostRegression）	8.778
	随机森林回归（RandomForestRegression）	9.598
	支持向量回归（SVR）	42.81
	多层感知机（MLP）	11.43
	自编码器（AE）	19.46
	变分自编码器（VAE）	47.71

根据上表中均方误差的对比，我们选用了 XGBoost、随机森林、自编码器、变分自编码器作为主要优化的模型。经过模型调优我们发现，由于训练数据集较小，多层感知机和神经网络（AE、VAE 模型）容易出现过拟合，因此在该数据尺度下，我们选择参数调试后效果更优的 XGBoost 模型作为最终的预测模型。

在 2AP 场景下，我们训练的 XGBoost 模型在训练数据集上的预测结果与真实值的均方误差（MSE）为 2.618。预测结果如下图 13 所示，可以看到预测值与真实值的误差较小。同时，我们输出了各个特征对分类的贡献情况，结果如下图 14 所示，可以看出 RSSI 对预测结果的贡献最高，其次是 NAV 门限。

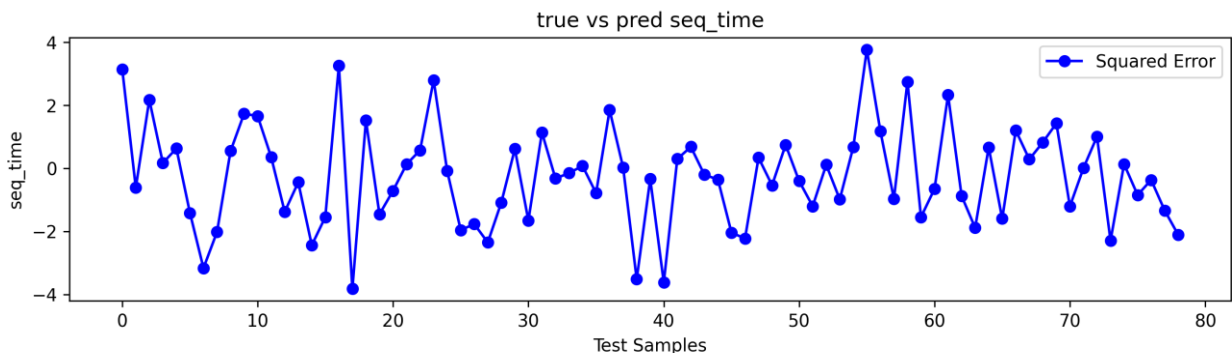


图 13 2AP 模型最优的预测值与真实值的误差情况



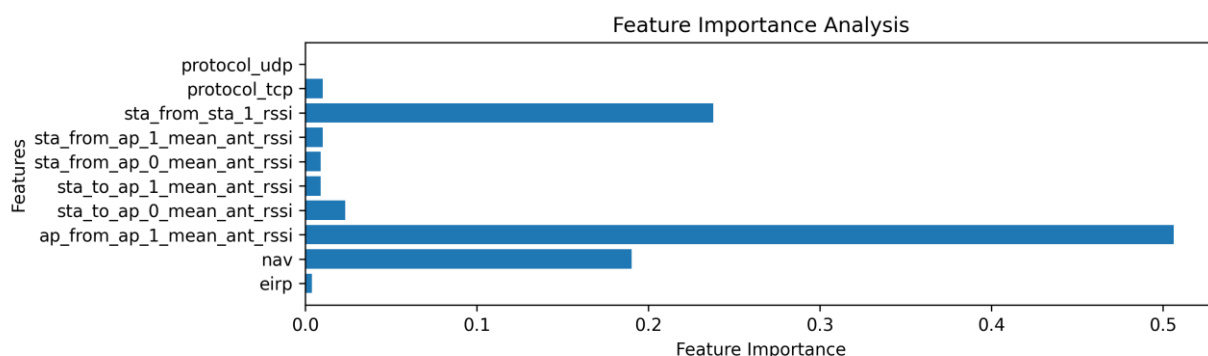


图 14 2AP 模型各特征的贡献情况

同时，经过我们对参数的调整，自编码器（AE）和变分自编码器（VAE）可以达到较低的均方误差（MSE），但经过我们对测试数据的预测，如图 15 和图 16，我们发现自编码器（AE）会出现预测结果为负值的情况，并且变分自编码器（VAE）的预测结果也与我们的设想偏离较远。我们分析这应该与训练数据集不够充分，导致神经网络产生过拟合的原因有关。因此我们最终选择了 XGBoost 的输出作为最终结果。

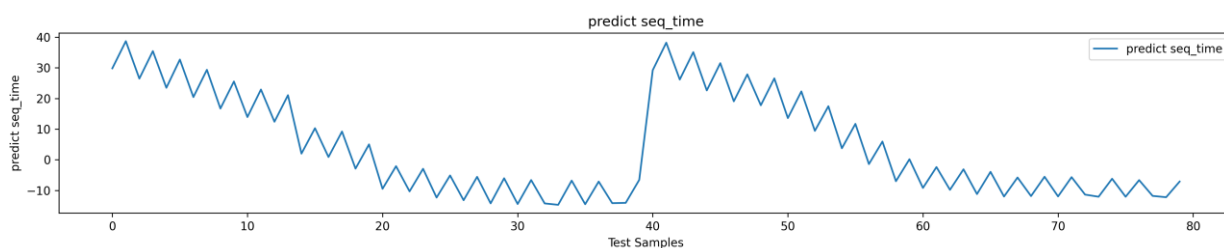


图 15 2AP 自编码器神经网络（AE）在测试集上的输出结果

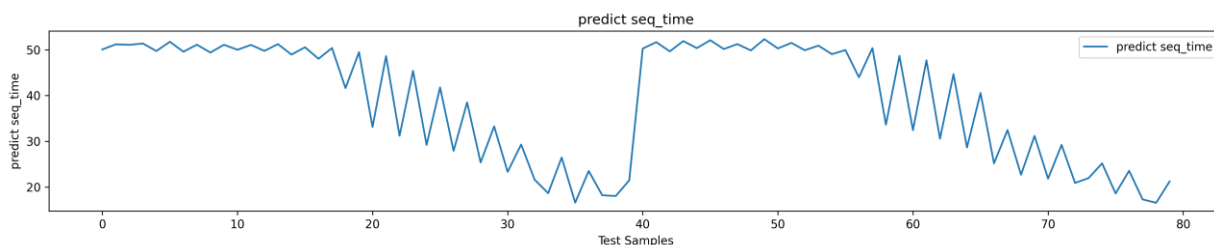


图 16 2AP 变分自编码器神经网络（VAE）在测试集上的输出结果

在 3AP 场景下，我们训练的 XGBoost 模型在训练数据集上的预测结果与真实值的均方误差（MSE）为 26.552。预测结果如下图 17 所示，可以看到预测值与真实值的误差较小。同时，我们输出了各个特征对分类的贡献情况，结果如下图 18 所示，可以看出 RSSI 对预测结果的贡献最高，其次是 NAV 门限。

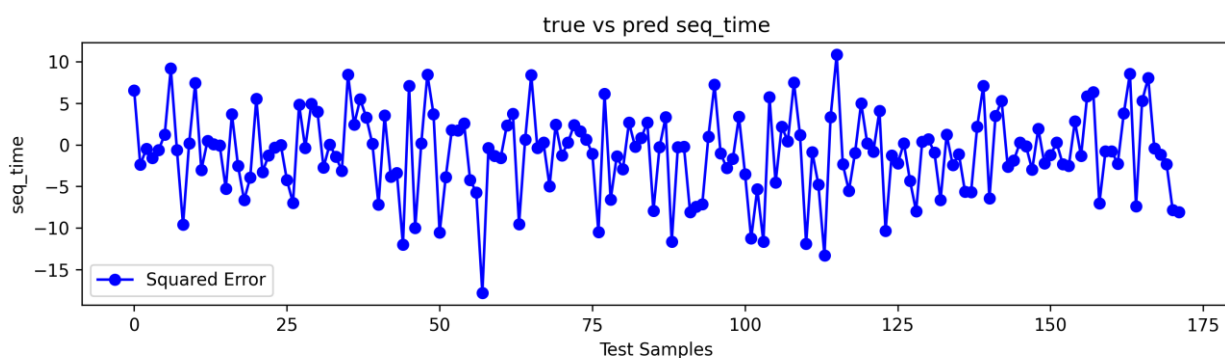


图 17 3AP 模型最优的预测值与真实值的对比情况

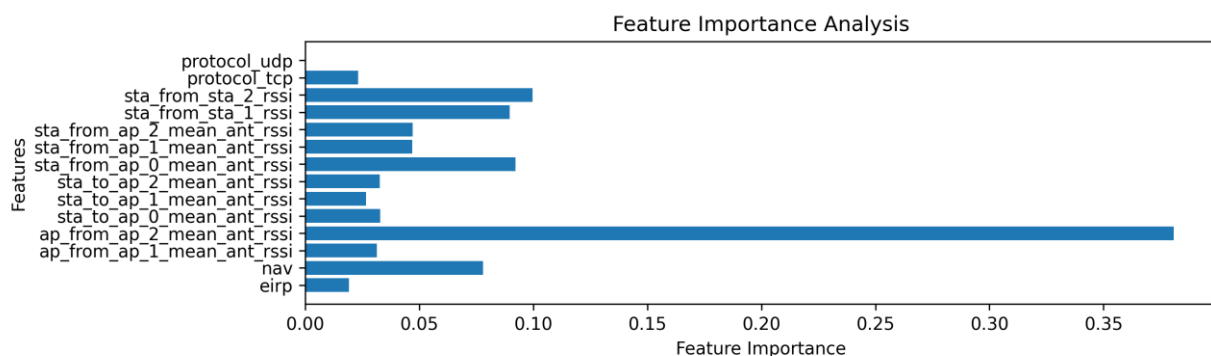


图 18 3AP 模型各个特征的贡献情况

同时，与 2AP 的测试验证类似，如图 19 和图 20，我们发现自编码器（AE）会出现预测结果为负值的情况，并且变分自编码器（VAE）的预测结果也与我们的设想偏离较远。我们分析这应该与训练数据集不够充分，导致神经网络产生过拟合的原因有关。因此我们最终选择了 XGBoost 的输出作为最终结果。

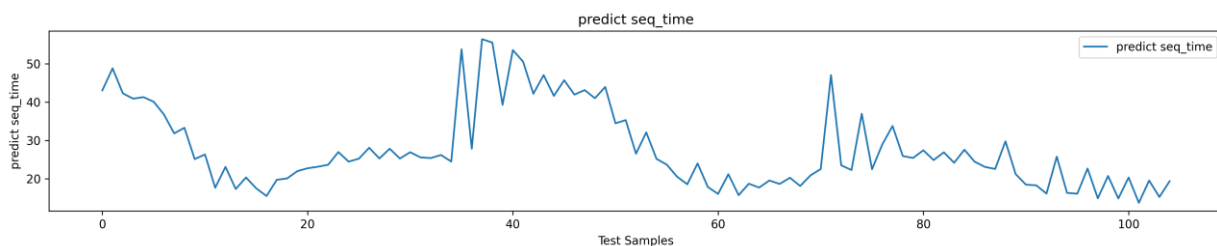


图 19 3AP 自编码器神经网络（AE）在测试集上的输出结果

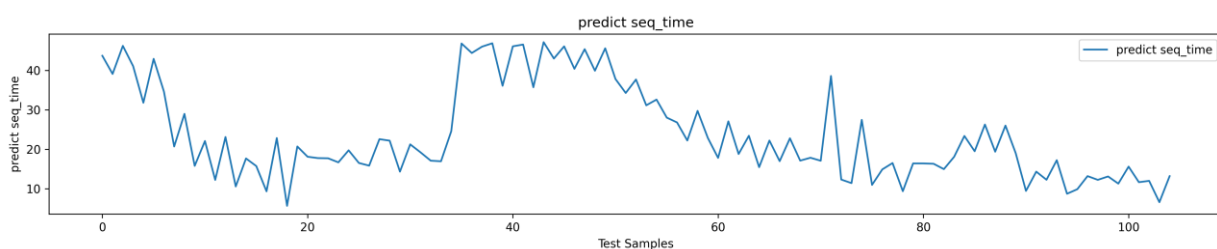


图 20 3AP 变分自编码器神经网络（VAE）在测试集上的输出结果

## 4.6 问题一的预测结果

我们将 2AP 和 3AP 分为不同的预测模型进行预测。

图 21 展示了使用 2AP 模型，对 test\_set\_1\_2p.csv 的 AP 发送序列时长预测结果。

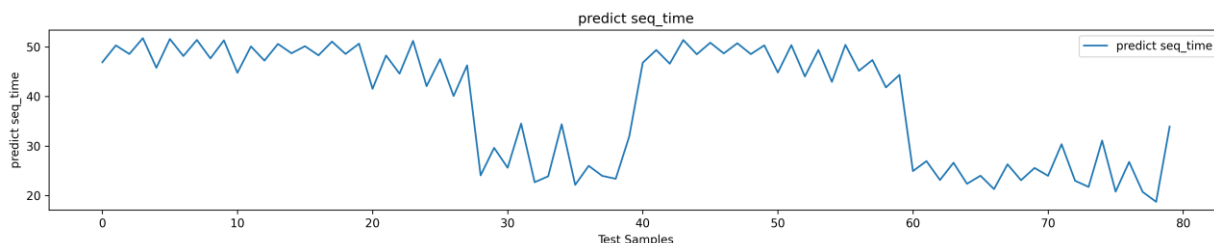


图 21 2AP 模型 test\_set\_1\_2p.csv 的预测结果

图 22 展示了使用 3AP 模型，对 test\_set\_1\_3p.csv 的 AP 发送序列时长的预测结果。

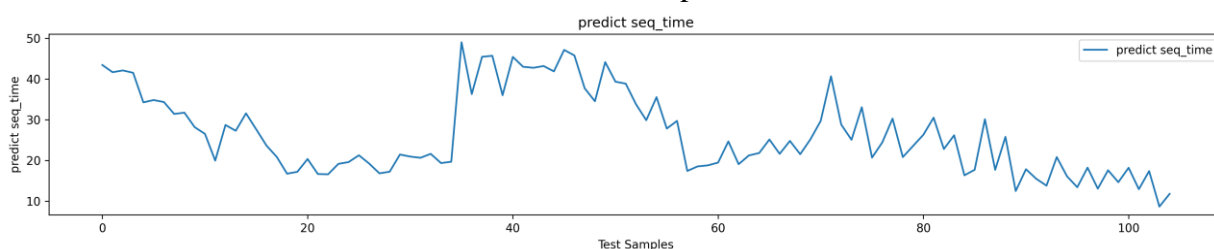


图 22 3AP 模型 test\_set\_1\_3p.csv 的预测结果

在增强数据集上训练的模型，预测结果如下图所示。

图 25 2AP 模型的预测结果展示了使用 2AP 模型，对 test\_set\_1\_2p.csv 的 AP 发送序列时长预测结果。

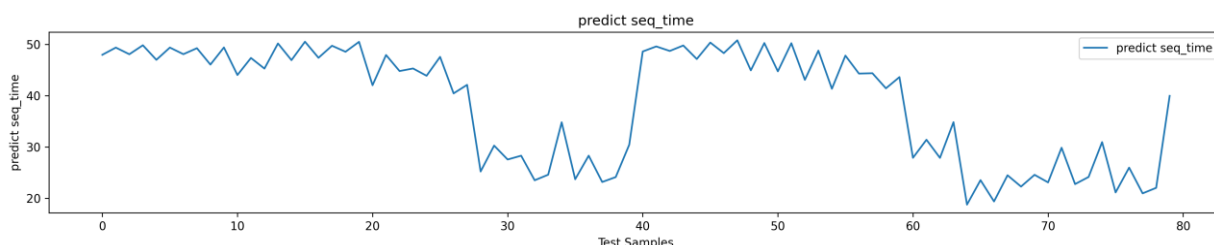


图 23 2AP 模型 test\_set\_1\_2p.csv 的预测结果

图 26 3AP 模型的预测结果展示了使用 3AP 模型，对 test\_set\_1\_3p.csv 的 AP 发送序列时长的预测结果。

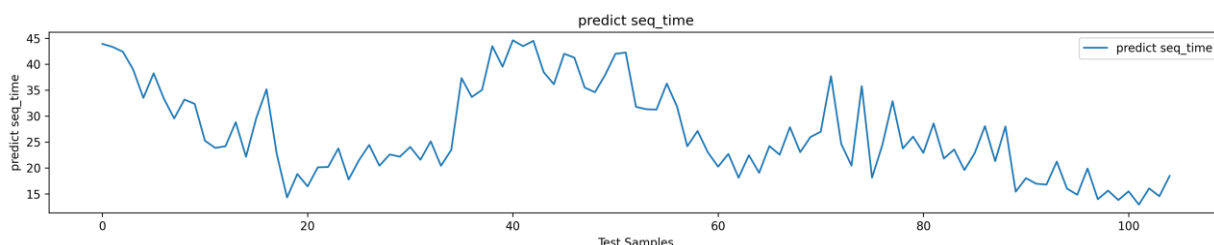


图 24 3AP 模型 test\_set\_1\_3p.csv 的预测结果

经过分析，在增强数据集上训练的模型预测结果更符合预期结果，我们认为增强数据集是提升模型预测发送序列时长（seq\_time）性能的重要步骤。

## 5 问题二的模型建立与求解

### 5.1 问题分析

问题二要求对 AP 发送数据选用最多次数的(MCS, NSS)进行建模，并在测试集进行预测。这是一个预测模型，模型的输入与问题一类似，包含测试基本信息：网络拓扑、业务流量、门限、节点间 RSSI。不同在于：由于问题一已经预测了 AP 的发送机会，因此对于模型输入，可以选择 AP 发送机会，也就是 seq\_time 指标作为输入。也可以不选择 seq\_time 指标作为输入，因为 seq\_time 也是基于测试基本信息预测得出，测试基本信息包含了所有能预测(MCS, NSS)的信息。

问题二需要重点关注 RSSI 信息和门限信息，因为 RSSI 和门限信息是影响信道传输方式和 SINR 的主要因素。而由于 SINR 会影响 PER，在 AMC 算法下，(MCS,NSS)会更新。因此，需要重点关注 RSSI 和门限信息。通过对数据集分析，可以发现(MCS,NSS)的数量是有限的，MCS 取值范围为[0, 11]，NSS 取值范围为{1, 2}。在 20MHz 带宽时，选用不同 (MCS, NSS) 组合的 PHY Rate (Mbps) 如下表 11 所示，由于(MCS,NSS)数量有限，因此我们可以考虑使用分类的方法而非回归方法来进行建模预测。

表 11 MCS, NSS 组合 PHY Rate (Mbps) 对照表

MCS	0	1	2	3	4	5	6	7	8	9	10	11
NSS1	8.6	17.2	25.8	34.4	51.6	68.8	77.4	86.0	103.2	114.7	129.0	143.4
NSS2	17.2	34.4	51.6	68.8	103.2	137.6	154.9	172.1	206.5	229.4	258.1	286.8

### 5.2 特征选择

基于问题一特征分析，RSSI、发送功率 (eirp)、NAV 门限和流量类型 (protocol) 这些因子会影响发送机会，进而影响 AP 的调制编码方案 (MCS) 和空间流数 (NSS)，所以对于问题二选用相同的特征。同问题一，2AP、3AP 均选用 XGBoost 为基准，对 mean\_RSSI + sum\_RSSI + max\_RSSI 三类 RSSI 做消融实验分析，结果如下表 12 和表 13。最终只选用 mean\_RSSI 单类特征，在特征维度和拟合效果上均较优。

表 12 2AP RSSI 消融实验预测准确率对比

特征	特征数量	准确率 (Accuracy)
mean_RSSI	6	93.51%
mean_RSSI + sum_RSSI	11	88.46%
mean_RSSI + max_RSSI	11	83.33%
mean_RSSI + sum_RSSI + max_RSSI	16	91.02%

表 13 3AP RSSI 消融实验预测准确率对比

特征	特征数量	准确率 (Accuracy)
----	------	----------------

mean_RSSI	10	75.58%
mean_RSSI +sum_RSSI	18	71.8%
mean_RSSI +max_RSSI	18	65.39%
mean_RSSI +sum_RSSI + max_RSSI	26	73.10%

### 5.3 模型参数设置

根据上述题目的分析，我们发现问题二的目标是预测(MCS,NSS)的数值，该问题可以被看做一个分类问题，即输入上述提取的特征后，对(MCS,NSS)的组合进行分类预测。于是，我们选择了线性回归、XGBoost、随机森林、支持向量分类、朴素贝叶斯、神经网络分类模型进行训练，并对比这些模型预测的精度，模型的参数设置如下表 14 所示。

表 14 分类模型的默认参数设置情况

模型名称	参数名称	参数含义	参数值
XGBClassifier	n_estimators	决策树的个数	100
	learning_rate	学习率	0.3
	max_depth	决策树最大深度	6
	booster	基学习器	gbtree
	objective	损失函数	binary:logistic
RandomForestClassifier	n_estimators	决策树的个数	100
	criterion	评判函数	gini
	max_depth	最大深度	None
	min_samples_split	叶节点上最小样本数	3
	max_features	分裂的最大特征数	sqrt
SVC	kernel	核函数	rbf
	C	正则化参数	1.0
	epsilon	容许的误差范围	0.1
	gamma	核函数中的系数	scale
MLPClassifier	hidden_layer_sizes	隐藏层信息	(64, 32)
	max_iter	最大迭代参数	2000
	alpha	L2 正则化参数	0.001
	activation	激活函数	relu
	solver	梯度下降法	adam
LogisticRegression	penalty	正则化的类型	l2
	solver	优化算法	lbfgs
	max_iter	最大迭代参数	100
	multi_class	多类别	multinomial
EncoderDecoderClassifier	hidden_size	隐藏层大小	64
	epoch	训练轮数	100
	lr	学习率	0.001
VAEClassifier	batch_size	批量大小	1
	hidden_size	隐藏层大小	64

latent_size	隐变量大小	32
epoch	训练轮数	1000
lr	学习率	0.001

## 5.4 模型训练

我们使用模型预测的(MCS,NSS)与训练数据集的(MCS,NSS)结果进行准确率（accuracy）的评价。实验结果如下表 15 和表 16 所示，结果表明，在 2AP 和 3AP 的场景下，XGBoost 均表现出更高的准确率，因此我们最终选择了 XGBoost 模型作为预测模型，并通过参数搜索寻找最优的模型。同问题一，由于训练数据集的规模较小，本文同时增强数据集进行预测，发现模型性能更好，且泛化能力强。最终原始数据集和增强数据集的训练精度如下表所示。

表 15 2AP 模型的(MCS,NSS) 预测准确率对比

数据集	模型名称	准确率（Accuracy）
原始数据集	线性回归（LogisticRegression）	79.48%
	XGBoost 回归（XGBClassifier）	93.51%
	随机森林回归（RandomForestClassifier）	92.31%
	支持向量回归（SVC）	76.92%
	朴素贝叶斯（GaussianNB）	67.95%
	多层感知机（MLP）	87.18%
	自编码器（AE）	64.12%
	变分自编码器（VAE）	78.11%
增强数据集	线性回归（LogisticRegression）	93.53%
	XGBoost 回归（XGBClassifier）	100%
	随机森林回归（RandomForestClassifier）	100%
	支持向量回归（SVC）	97.43%
	朴素贝叶斯（GaussianNB）	82.05%
	多层感知机（MLP）	97.43%
	自编码器（AE）	97.43%
	变分自编码器（VAE）	69.23%

表 16 3AP 模型的(MCS,NSS)预测准确率对比

数据集	模型名称	准确率（Accuracy）
原始数据集	线性回归（LogisticRegression）	51.16%
	XGBoost 回归（XGBClassifier）	75.58%
	随机森林回归（RandomForestClassifier）	71.51%
	支持向量回归（SVC）	65.69%
	朴素贝叶斯（GaussianNB）	46.51%
	多层感知机（MLP）	68.02%
	自编码器（AE）	45.22%
	变分自编码器（VAE）	69.15%

增强数据集	线性回归（LogisticRegression）	59.88%
	XGBoost 回归（XGBClassifier）	98.89%
	随机森林回归（RandomForestClassifier）	97.63%
	支持向量回归（SVC）	86.04%
	朴素贝叶斯（GaussianNB）	47.09%
	多层感知机（MLP）	94.18%
	自编码器（AE）	92.44%
	变分自编码器（VAE）	15.11%

在 2AP 场景下，我们训练的 XGBoost 模型在训练集上预测准确率可以达到 100%，XGBoost 在训练集上的预测结果报告（report）如下表 17 所示。

表 17 2AP 模型最优的预测结果

(MCS,NSS)	精确率（precision）	召回率（recall）	F1-Score	样本个数
2_10	1.0	1.0	1.0	6
2_11	1.0	1.0	1.0	54
2_3	1.0	1.0	1.0	1
2_4	1.0	1.0	1.0	8
2_5	1.0	1.0	1.0	3
2_8	1.0	1.0	1.0	3
2_9	1.0	1.0	1.0	5
总计	1.0	1.0	1.0	78

在 3AP 场景下，我们训练的 XGBoost 模型在训练集上的预测准确率可以达到 88.37%，XGBoost 在训练集上的预测结果报告（report）如下表 18 所示。

表 18 3AP 模型最优的预测结果

(MCS,NSS)	精确率（precision）	召回率（recall）	F1-Score	样本个数
1_5	0.0	0.0	0.0	1
2_10	0.90	0.82	0.86	11
2_11	0.95	0.99	0.97	78
2_3	0.0	0.0	0.0	1
2_4	0.92	0.96	0.94	25
2_5	0.78	0.64	0.70	11
2_6	0.62	0.62	0.62	8
2_7	0.80	0.80	0.80	10
2_8	0.81	0.81	0.81	16
2_9	0.75	0.82	0.78	11
总计	0.87	0.88	0.88	172

## 5.5 问题二的预测结果

我们将 2AP 和 3AP 分为不同的预测模型进行预测。由于预测的(MCS,NSS)可以对应到具体的数值，因此我们将预测的(MCS,NSS)对应到原始的数值进行展示。

2AP 模型的(MCS,NSS)对应值预测结果如图 25 所示。

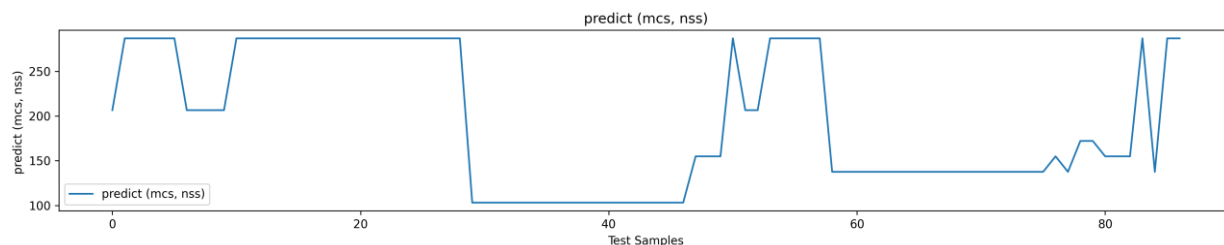


图 25 2AP 模型的预测结果

3AP 模型的(MCS,NSS)对应值预测结果如图 26 所示。

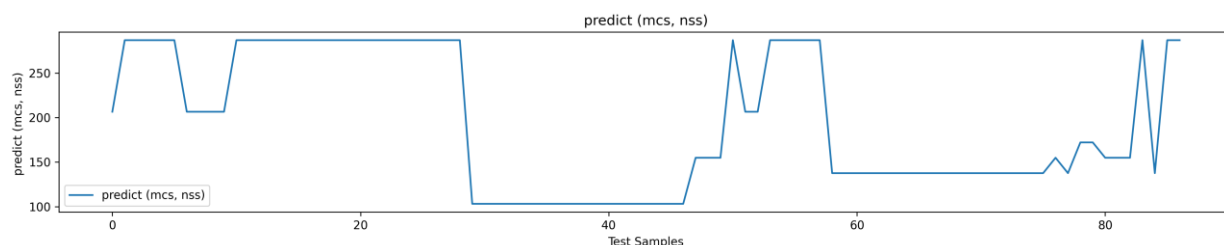


图 26 3AP 模型的预测结果

在增强数据集上训练的模型，预测结果如下图所示。

2AP 模型的(MCS,NSS)对应值预测结果如图 31 所示。

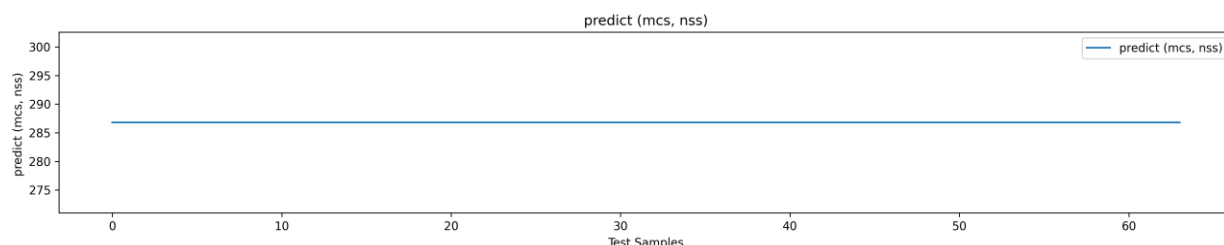


图 27 2AP 模型的预测结果

3AP 模型的(MCS,NSS)对应值预测结果如图 32 所示。

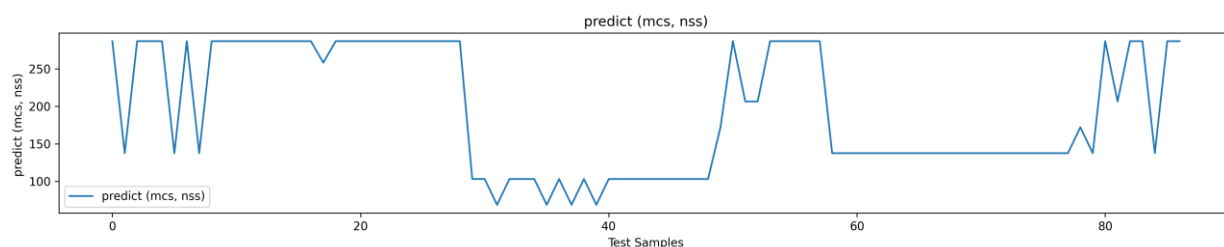


图 28 3AP 模型的预测结果

经过分析，在增强数据集上训练的模型预测结果更符合预期结果，我们认为增强数据集是提升模型分类预测调制编码方案（MCS）和空间流数（NSS）性能的重要步骤。



## 6 问题三的模型建立与求解

### 6.1 问题分析

问题三要求对系统吞吐量建模。系统吞吐量受到许多因素影响，包括 AP 发送机会、(MCS,NSS)、丢包率等。

#### 6.1.1 非测试基本信息的影响因素

在问题一中，我们已经分析了测试基本信息对 AP 发送机会的影响，由于会影响 AP 发送机会，这些因素也都会影响这些系统吞吐量。因此下面的分析主要在于非测试基本信息的影响因素。

**数据帧的聚合个数 (num\_ampdu)：**实际的传输场景中，数据帧会进行聚合，聚合时，由于聚合的子 MSDU 之间，及子 MPDU 之间需要插入冗余字段，实际一个 PPDU 传输时长里，仅 MSDU 的总字节数是有效传输数据，进行吞吐量计算时才被计入，因此，AMSDU 和 AMPDU 聚合的准确评估对吞吐量预测也有影响。同等情况下，num\_ampdu 越高，系统吞吐量越低。

**其他空闲时间 (ther\_air\_time)：**该参数表示 AP 除发送数据帧外，发送管理帧等其他帧和接收所占用的时间。注意到在有些场景下，某个 ap 的 other\_air\_time 始终为 0。具体来说，测试数据集中，除了 training\_set\_2ap\_loc0\_nav86 和 training\_set\_2ap\_loc1\_nav86 以外，其他 11 个文件，均包含一个 AP 的 other\_air\_time 始终为 0。同时，我们注意到 training\_set\_3ap\_loc31\_nav82 中，存在一个 other\_air\_time 值达到了 1181768.274。除此之外，其他所有的值都小于 60。此外，我们还发现 other\_air\_time 和 seq\_time 成负相关，也和吞吐量成负相关。这在数据集通过相关性分析中可以非常明显地观察出来。

**丢包率 (per)：**丢包率的增加会导致传输效率降低，从而直接影响到网络吞吐量。我们通过数据集集中的 per 和 throughput 作相关性分析，也发现 per 和 throughput 成负相关的关系。

#### 6.1.2 影响因素与系统吞吐量的相关性

对系统吞吐量进行精准预测是比较困难的，难以用具体公式呈现出来。然而，我们可以分析系统吞吐量的影响因素，建立吞吐量与各个影响因素的相关性，从而对系统吞吐量的大小变化有宏观把控。如下表 19 所示：

表 19 系统吞吐量的影响因素

影响因素	原理	与系统吞吐量的相关性
AP 的发送机会	增加传输的时间	正
AP 与相邻 BSS AP 的 RSSI	减少 AP 的发送机会	负
NAV 门限	增加 AP 的发送机会	正
BSS 内的 AP 功率	增加 AP 的发送机会	正
相邻 BSS 的 AP 功率	减少 AP 的发送机会	负
SINR	增加传输的有用数据比例	正

数据帧的聚合个数	减少传输的有用数据比例	负
丢包率	减少传输的时间	负
其他空闲时间	减少传输的时间	负
同步/异步传输	-	复杂
业务流量类型	-	复杂

### 6.1.3 同步传输与异步传输

同步传输和异步传输是影响系统吞吐量的重要因素，对系统吞吐量的影响比较复杂。

同步传输，指的是当两个节点间的  $RSSI > ED$  时，能互相侦听情况下发生的传输。同步传输包括交替进行和同时发生的传输。同步传输下两个 AP 交替抢到信道，偶然同时发送。这种情况下两个 AP 交替利用信道发送信号，平均意义下每个 AP 只能使用一半的时间用来发送信号。因此，同步传输与系统吞吐量呈现负相关。

异步传输指的是当两个节点间的  $RSSI$  较小时，AP 总是不互听时情况下发生的传输。这种情况下，STA 接收数据时可能受到较小的邻区干扰和环境底噪。这种情况下 AP 都能占满信道的所有时间。因此，异步传输与系统吞吐量呈现正相关。

另一方面，同步传输和异步传输的  $SINR$  也不一样。同步传输时，因为能互相“侦听”，STA1 收到 AP1 的信号时候，AP2 此时处于静默期，不会干扰信号，所以噪声基本只来自环境底噪，因而  $SINR$  高。异步传输时，两个节点不能互听。但是，来自其他节点的  $RSSI$  始终是存在的，STA1 收到 AP1 信号的同时，也会收到 AP2 的信号，这就会对信号造成干扰，因此除了环境底噪外，异步传输还受到相邻 AP 的干扰，因此  $RSSI$  较低。

总结来说，同步传输和异步传输会对系统吞吐量造成影响，是通过同时影响 AP 发送机会和  $SINR$  来影响的。同步传输的  $SINR$  高，但 AP 发送机会少；异步传输的 AP 发送机会高，但  $SINR$  大。实际过程还存在同步传输与异步传输的混合或者频繁切换，因此对系统吞吐量的影响更加复杂。

### 6.1.4 业务流量类型

业务流量类型指的是传输过程使用的协议，有 TCP 和 UDP 两种。

TCP (TransmissionControl Protocol) 是面向连接的传输层协议，提供可靠的数据传输，通过三次握手建立连接，并采用确认应答和重传机制保证数据完整性。TCP 的拥塞控制机制虽然保证了数据传输的可靠性，但在高延迟或高丢包率的网络中，TCP 的吞吐量会显著下降。这是因为每次丢包都会触发重传和窗口收缩，从而影响发送速率。

UDP (User Datagram Protocol) 是无连接的传输层协议，提供不保证可靠性的快速数据传输，适用于实时应用，如视频流和在线游戏，不进行握手或重传。UDP 没有拥塞控制机制，因此可以最大化地利用可用带宽，传输速率仅受物理链路带宽的限制。这使得 UDP 在理想网络环境中可以获得比 TCP 更高的吞吐量。而由于缺乏可靠性保障，UDP 在丢包率高的网络环境中表现较差。

总结来说，TCP 对网络吞吐量的正相关性体现在数据传输的可靠性和完整性，负相关性体现在网络开销较大和窗口控制上。UDP 对网络吞吐量的正相关性体现在，负相关性体现在不可靠性带来的高丢包率，正相关性体现在连接快速网络开销小。因此，业务流量类型对网络吞吐量的影响是复杂的。

6.2 特征选择

对于问题三，为了预测系统吞吐量，在问题一、二的基础上，特征添加了实测中统计的数据帧真实调制编码方案（MCS）和空间流数（NSS），最终选用特征 RSSI、发送功率（eirp）、NAV 门限和流量类型（protocol）、（MCS，NSS）。同问题一，2AP、3AP 均选用 XGBoost 为基准，对 mean\_RSSI +sum\_RSSI + max\_RSSI 三类 RSSI 做消融实验分析，结果如下表 20 和表 21。最终选用 mean\_RSSI 单类特征，在特征维度和拟合效果上均较优。

表 20 2AP RSSI 消融实验均方误差对比

特征	特征数量	均方误差（MSE）
mean_RSSI	6	68.41
mean_RSSI +sum_RSSI	11	72.69
mean_RSSI +max_RSSI	11	80.52
mean_RSSI +sum_RSSI + max_RSSI	16	73.48

表 21 3AP RSSI 消融实验均方误差对比

特征	特征数量	均方误差（MSE）
mean_RSSI	10	334.45
mean_RSSI +sum_RSSI	18	369.80
mean_RSSI +max_RSSI	18	385.34
mean_RSSI +sum_RSSI + max_RSSI	26	354.63

6.3 模型训练

由于问题三的目标与问题一相同，均为预测模型，因此我们选用了与问题一相同的模型进行对比测试，并且默认参数保持一致。我们将训练集按照 8:2 的比例进行划分，其中训练集为 80%，测试集为 20%，并使用均方误差（MSE）评估模型在测试数据集上的效果。模型训练精度如下表 22 和表 23 所示。

表 22 WLAN 网络吞吐量预测模型均方误差对比

数据集	模型名称	均方误差（MSE）
原始数据集	线性回归（LinearRegression）	399.08
	XGBoost 回归（XGBoostRegression）	68.41
	随机森林回归（RandomForestRegression）	77.91
	支持向量回归（SVR）	2013.63
	多层感知机（MLP）	112.09
	自编码器（AE）	128.12
	变分自编码器（VAE）	142.66
增强数据集	线性回归（LinearRegression）	415.01
	XGBoost 回归（XGBoostRegression）	2.665
	随机森林回归（RandomForestRegression）	11.16
	支持向量回归（SVR）	914.03

多层感知机（MLP）	59.68
自编码器（AE）	117.77
变分自编码器（VAE）	65.41

表 23 WLAN 网络吞吐量预测模型均方误差对比

数据集	模型名称	均方误差（MSE）
原始数据集	线性回归（LinearRegression）	941.47
	XGBoost 回归（XGBoostRegression）	334.45
	随机森林回归（RandomForestRegression）	359.86
	支持向量回归（SVR）	1435.86
	多层感知机（MLP）	385.72
	自编码器（AE）	562.32
	变分自编码器（VAE）	744.23
增强数据集	线性回归（LinearRegression）	921.28
	XGBoost 回归（XGBoostRegression）	4.005
	随机森林回归（RandomForestRegression）	27.22
	支持向量回归（SVR）	914.54
	多层感知机（MLP）	194.28
	自编码器（AE）	315.74
	变分自编码器（VAE）	246.97

根据上表中均方误差的对比，我们选用了 XGBoost、随机森林、自编码器、变分自编码器作为主要优化的模型。经过模型调优我们发现，由于训练数据集较小，多层感知机和神经网络（AE、VAE 模型）容易出现过拟合，因此在该数据尺度下，我们选择参数调试后效果更优的 XGBoost 模型作为最终的预测模型。

在 2AP 场景下，我们训练的 XGBoost 模型在训练数据集上的预测结果与真实值的均方误差（MSE）为 58.11。预测结果如下图 29 所示，可以看到预测值与真实值的误差较小。同时，我们输出了各个特征对分类的贡献情况，结果如下图 30 所示，可以看出 RSSI 对预测结果的贡献最高，其次是(MCS,NSS)。

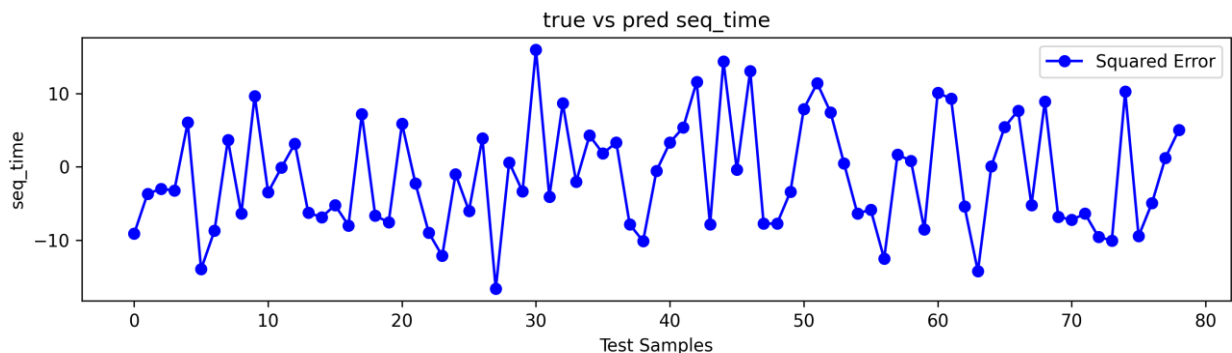


图 29 2AP 模型最优的预测值与真实值的误差情况

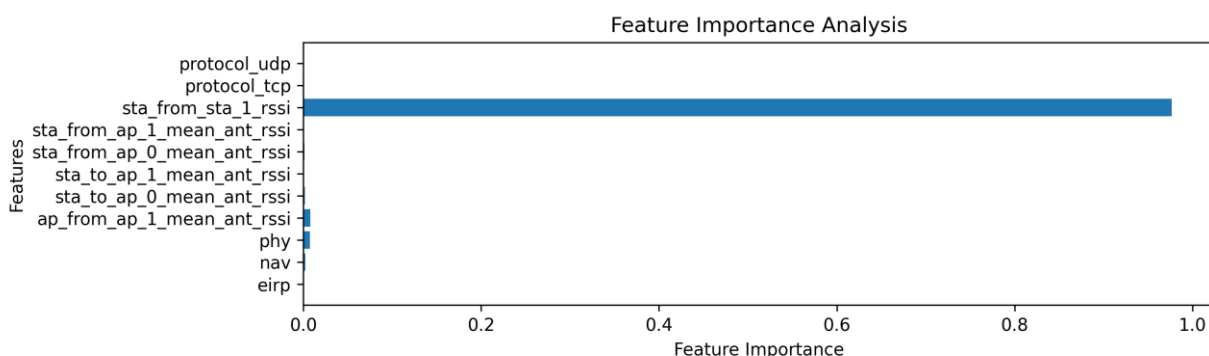


图 30 2AP 模型各特征的贡献情况

在 3AP 场景下，我们训练的 XGBoost 模型在训练数据集上的预测结果与真实值的均方误差（MSE）为 295.49。预测结果如下图 31 所示，可以看到预测值与真实值的误差较小。同时，我们输出了各个特征对分类的贡献情况，结果如下图 32 所示，可以看出 RSSI 对预测结果的贡献最高，其次是 NAV 门限和(MCS,NSS)。

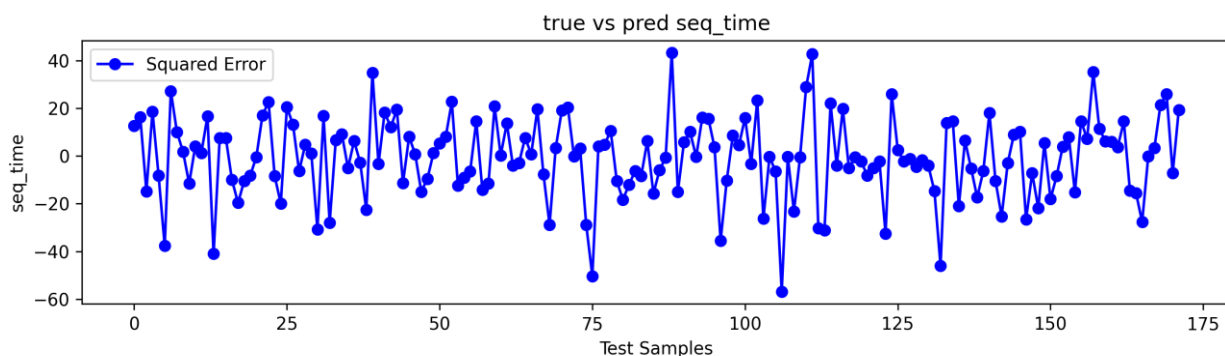


图 31 3AP 模型最优的预测值与真实值的对比情况

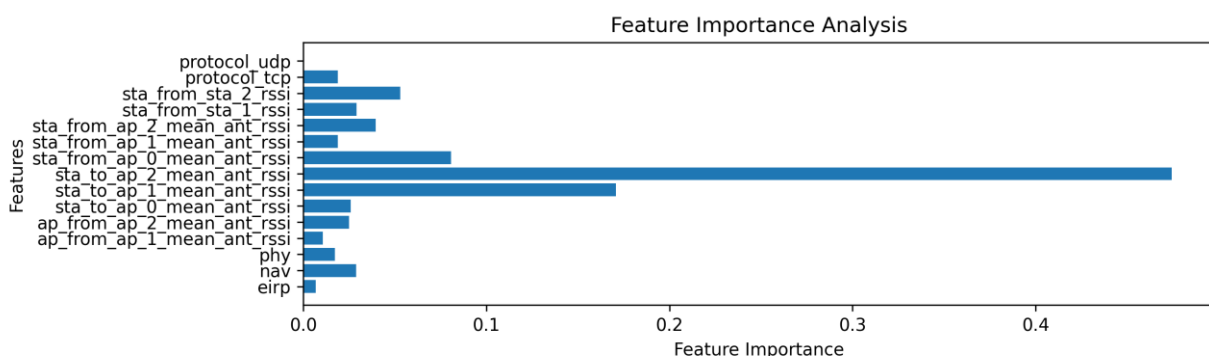


图 32 3AP 模型各个特征的贡献情况

## 6.4 问题三的预测结果

我们将 2AP 和 3AP 分为不同的预测模型进行预测。

图 33 展示了使用 2AP 模型，对 test\_set\_1\_2p.csv 的网络吞吐量的预测结果。

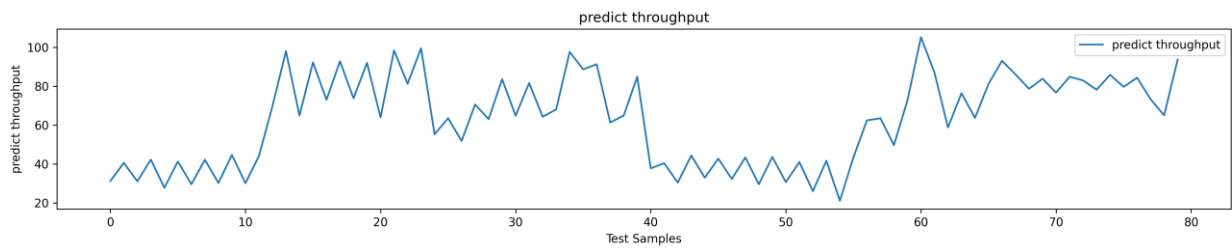


图 33 2AP 模型 test\_set\_1\_2p.csv 的网络吞吐量预测结果

图 34 展示了使用 3AP 模型，对 test\_set\_1\_3p.csv 的网络吞吐量的预测结果。

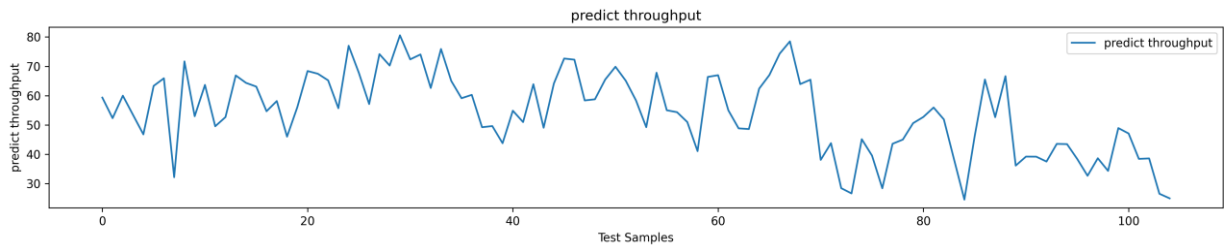


图 34 3AP 模型 test\_set\_1\_3p.csv 的网络吞吐量预测结果

在增强数据集上训练的模型，预测结果如下图所示。

图 35 展示了使用 2AP 模型，对 test\_set\_1\_2p.csv 的网络吞吐量的预测结果。

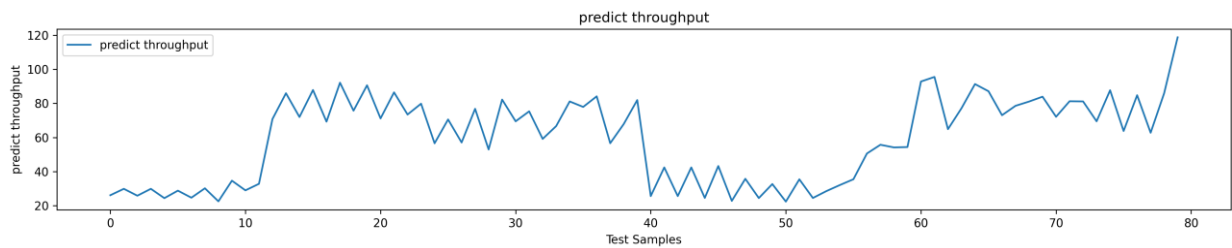


图 35 2AP 模型 test\_set\_1\_2p.csv 的网络吞吐量预测结果

图 36 展示了使用 3AP 模型，对 test\_set\_1\_3p.csv 的网络吞吐量的预测结果。

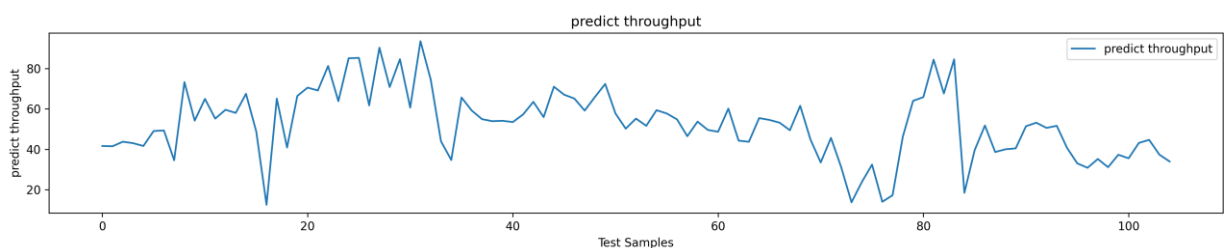


图 36 3AP 模型 test\_set\_1\_3p.csv 的网络吞吐量预测结果

经过分析，在增强数据集上训练的模型预测结果更符合预期结果，我们认为增强数据集是提升模型预测吞吐量（throughput）性能的重要步骤。

## 7 模型评价与展望

### 7.1 创新点

本文比较多种预测和分类模型，认为基于决策树的模型 XGBoost 作为预测模型和决策模型都取得了良好的性能，抗过拟合能力强，且具备可解释性，能给出每个特征的重要性分数。本研究通过在特征选择、模型应用以及数据处理等方面进行多维度优化，创新点如下：

特征选择。本文基于 PCA 分析和 XGBoost 贡献度排序，筛选出对 AP 发送机会影响最大的特征，包括 RSSI、发送功率（eirp）、NAV 门限、流量类型（protocol）。通过消融实验进一步验证特征对模型效果的影响，确保输入特征具备高度相关性和预测能力。

多模型对比。本文引入多个模型（XGBoost、随机森林、线性回归、自编码器、变分自编码器）并进行对比，最终选择 XGBoost 作为最佳模型，不仅降低了过拟合风险，也提升了模型的预测精度。

时序数据处理。针对 RSSI 时序数据长度不固定的问题，采用了求均值的方式将时序数据单值化，解决了时间序列数据处理复杂的问题，提高了模型稳定性。

### 7.2 优点

本文综合考虑了不同因素对 AP 发送机会和系统吞吐量的影响，包括 RSSI、门限、同步异步、SINR、网络拓扑等众多因素，同时测试不同模型的预测和分类性能，提供了一个较为全面评估，实现高精度预测。主要优点如下：

多维特征融合。本文将 RSSI、发送功率、NAV 门限等多维特征进行融合，能够精准捕捉 AP 发送机会和吞吐量的变化，提供了丰富的模型输入。

高精度预测。本文通过优化特征选择和模型调优，发现 XGBoost 模型在多次实验中表现出高精度的预测能力，尤其在 2AP 场景下，均方误差为 2.618，展现出极高的性能。

广泛适用性。XGBoost 模型不仅在 2AP 场景下表现优异，在 3AP 场景下也具备良好的适用性，说明该方法具有较好的泛化能力。

### 7.3 缺点

尽管模型具有较高的精度和适用性，由于影响性能因素众多（如：噪声、异常值、缺失值等）预测吞吐量仍然无法取得比较好的精度，主要缺点如下所示：

数据集有限。由于训练数据集较小，神经网络模型如自编码器和变分自编码器容易出现过拟合问题，无法充分展现其潜力。在更大规模数据集上，神经网络可能具备更强的表现。

时序信息丢失。本文将 RSSI 时序数据单值化处理虽然简化了模型输入，但在一定程度上丢失了时间维度的信息，可能影响对动态网络环境的捕捉。

### 7.4 展望

对网络吞吐量的预测是一个非常具有挑战性的工作。本文所做的工作不足以完美地建模网络吞吐量，在本实验中，XGBoost 的效果最佳，但可能存在更先进的深度学习模型，比如图神经网络。未来对工作的展望规划如下：

首先，丰富数据集特征，引入更多维度的指标极为重要（如：针对动态 RSSI 处理）。并且，可以通过特征融合和先进的预测分类方法弥补指标特征较少的不足，进一步提升对复杂网络吞吐量的预测精度。

其次，扩大数据集规模增强数据集与多场景训练能够提升模型的泛化能力。当前机器学习模型和神经网络模型性能均受限于数据规模，导致拟合不充分或过拟合现象。未来可以通过采集更多真实场景数据进行训练（如：基于仿真与现实的联合数据集）。

最后，模型的快速预测极为重要。在真实网络环境中，模型需要具备快速响应能力，能够随时根据网络状态的变化进行实时更新。可以探索在线学习算法与边缘计算结合，以便模型可以随着数据的更新而进行动态调整，从而提升在实际场景中的部署效果。

## 参考文献

- [1] 刘曼, 方旭明. 基于改进遗传算法的多 AP 联合传输方案研究[J]. 物联网学报, 7(3): 62-71.
- [2] 张文硕. 基于监督学习的自适应调制编码和功率分配算法研究[D]. 哈尔滨工业大学, 2019.
- [3] 杨明文. 自适应调制编码系统门限调整算法研究[J]. 数字技术与应用, 2015, 33(1): 129-129.
- [4] Zhou H, Kannan R, Swami A, et al. HTNet: Dynamic WLAN Performance Prediction using Heterogenous Temporal GNN[C]//IEEE INFOCOM 2023-IEEE Conference on Computer Communications. IEEE, 2023: 1-10.
- [5] Khan M A, Hamila R, Al-Emadi N A, et al. Real-time throughput prediction for cognitive Wi-Fi networks[J]. Journal of Network and Computer Applications, 2020, 150: 102499.
- [6] Chen T, Guestrin C. Xgboost: A scalable tree boosting system[C]//Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. 2016: 785-794.



## 附录

### 问题一结果

#### 1. test\_set\_1\_2ap.csv

test_id	protocol	ap_id	predict seq_time
1	tcp	ap_1	47.986717
1	tcp	ap_0	49.39396
2	udp	ap_1	48.09156
2	udp	ap_0	49.844635
3	tcp	ap_1	47.005688
3	tcp	ap_0	49.396145
4	udp	ap_1	48.104805
4	udp	ap_0	49.272575
5	tcp	ap_1	46.075718
5	tcp	ap_0	49.416714
6	udp	ap_1	44.042194
6	udp	ap_0	47.366165
7	tcp	ap_1	45.295605
7	tcp	ap_0	50.19452
8	udp	ap_1	46.94503
8	udp	ap_0	50.53933
9	tcp	ap_1	47.402225
9	tcp	ap_0	49.75201
10	udp	ap_1	48.577793
10	udp	ap_0	50.51046
11	tcp	ap_1	42.053913
11	tcp	ap_0	47.942562
12	udp	ap_1	44.816433
12	udp	ap_0	45.31743
13	tcp	ap_1	43.896427
13	tcp	ap_0	47.580956
14	udp	ap_1	40.460587
14	udp	ap_0	42.127872
15	tcp	ap_1	25.232754
15	tcp	ap_0	30.278662
16	udp	ap_1	27.569614
16	udp	ap_0	28.313213
17	tcp	ap_1	23.51872

17	tcp	ap_0	24.608147
18	udp	ap_1	34.822216
18	udp	ap_0	23.724892
19	tcp	ap_1	28.31923
19	tcp	ap_0	23.182726
20	udp	ap_1	24.144741
20	udp	ap_0	30.447378
21	tcp	ap_1	48.62781
21	tcp	ap_0	49.616734
22	udp	ap_1	48.731907
22	udp	ap_0	49.799755
23	tcp	ap_1	47.164913
23	tcp	ap_0	50.372997
24	udp	ap_1	48.311478
24	udp	ap_0	50.809315
25	tcp	ap_1	44.95097
25	tcp	ap_0	50.27852
26	udp	ap_1	44.759235
26	udp	ap_0	50.24503
27	tcp	ap_1	43.102352
27	tcp	ap_0	48.795048
28	udp	ap_1	41.36157
28	udp	ap_0	47.82464
29	tcp	ap_1	44.325176
29	tcp	ap_0	44.39049
30	udp	ap_1	41.447464
30	udp	ap_0	43.63939
31	tcp	ap_1	27.899597
31	tcp	ap_0	31.423546
32	udp	ap_1	27.886192
32	udp	ap_0	34.851536
33	tcp	ap_1	18.777332
33	tcp	ap_0	23.546978
34	tcp	ap_1	19.392454
34	tcp	ap_0	24.495754
35	udp	ap_1	22.292341
35	udp	ap_0	24.587076
36	tcp	ap_1	23.107164
36	tcp	ap_0	29.866457

37	udp	ap_1	22.774601
37	udp	ap_0	24.165804
38	tcp	ap_1	30.960241
38	tcp	ap_0	21.15319
39	tcp	ap_1	25.987211
39	tcp	ap_0	20.967913
40	udp	ap_1	22.043198
40	udp	ap_0	39.97783

## 2. test\_set\_1\_3ap.csv

test_id	protocol	ap_id	predict seq_time
1	tcp	ap_2	43.928886
1	tcp	ap_1	43.345074
1	tcp	ap_0	42.414394
2	udp	ap_2	38.99017
2	udp	ap_1	33.506836
2	udp	ap_0	38.252922
3	tcp	ap_2	33.358006
3	tcp	ap_1	29.540926
3	tcp	ap_0	33.18635
4	tcp	ap_2	32.355034
4	tcp	ap_1	25.238447
4	tcp	ap_0	23.863749
5	udp	ap_2	24.176401
5	udp	ap_1	28.815645
5	udp	ap_0	22.152376
6	tcp	ap_2	29.615435
6	tcp	ap_1	35.165714
6	tcp	ap_0	22.644392
7	tcp	ap_2	14.289313
7	tcp	ap_1	18.824682
7	tcp	ap_0	16.439041
8	udp	ap_2	20.108332
8	udp	ap_1	20.17859
8	udp	ap_0	23.76729
9	tcp	ap_2	17.760609
9	tcp	ap_1	21.45423
9	tcp	ap_0	24.424818

10	udp	ap_2	20.418394
10	udp	ap_1	22.59196
10	udp	ap_0	22.16516
11	tcp	ap_2	24.049559
11	tcp	ap_1	21.577269
11	tcp	ap_0	25.118767
12	udp	ap_2	20.407372
12	udp	ap_1	23.475739
12	udp	ap_0	37.319065
13	tcp	ap_2	33.68112
13	tcp	ap_1	35.048435
13	tcp	ap_0	43.482117
14	udp	ap_2	39.53173
14	udp	ap_1	44.617096
14	udp	ap_0	43.48966
15	tcp	ap_2	44.51778
15	tcp	ap_1	38.491375
15	tcp	ap_0	36.13697
16	tcp	ap_2	42.02349
16	tcp	ap_1	41.267
16	tcp	ap_0	35.50912
17	udp	ap_2	34.599808
17	udp	ap_1	37.96526
17	udp	ap_0	42.015583
18	tcp	ap_2	42.254665
18	tcp	ap_1	31.781908
18	tcp	ap_0	31.319628
19	udp	ap_2	31.258385
19	udp	ap_1	36.271133
19	udp	ap_0	31.830679
20	tcp	ap_2	24.16658
20	tcp	ap_1	27.121552
20	tcp	ap_0	23.017965
21	tcp	ap_2	20.237026
21	tcp	ap_1	22.69523
21	tcp	ap_0	18.107365
22	tcp	ap_2	22.44419
22	tcp	ap_1	19.056587
22	tcp	ap_0	24.21246

23	tcp	ap_2	22.545544
23	tcp	ap_1	27.857044
23	tcp	ap_0	23.038134
24	udp	ap_2	25.958107
24	udp	ap_1	26.973734
24	udp	ap_0	37.706905
25	tcp	ap_2	24.619013
25	tcp	ap_1	20.431782
25	tcp	ap_0	35.772896
26	tcp	ap_2	18.112286
26	tcp	ap_1	24.391665
26	tcp	ap_0	32.878113
27	udp	ap_2	23.751026
27	udp	ap_1	26.03441
27	udp	ap_0	22.900028
28	tcp	ap_2	28.577242
28	tcp	ap_1	21.81705
28	tcp	ap_0	23.547365
29	udp	ap_2	19.589794
29	udp	ap_1	22.834948
29	udp	ap_0	28.06216
30	tcp	ap_2	21.311792
30	tcp	ap_1	27.99468
30	tcp	ap_0	15.407279
31	udp	ap_2	18.024279
31	udp	ap_1	16.94784
31	udp	ap_0	16.787115
32	tcp	ap_2	21.219893
32	tcp	ap_1	15.99535
32	tcp	ap_0	14.822382
33	udp	ap_2	19.8821
33	udp	ap_1	13.948583
33	udp	ap_0	15.625843
34	tcp	ap_2	13.800107
34	tcp	ap_1	15.479302
34	tcp	ap_0	12.9194765
35	udp	ap_2	16.040634
35	udp	ap_1	14.534349
35	udp	ap_0	18.460938

## 问题二结果

### 1. test\_set\_2\_2ap.csv

test_id	protocol	ap_id	predict nss	predict mcs
1	tcp	ap_1	2	11
1	tcp	ap_0	2	11
2	udp	ap_1	2	11
2	udp	ap_0	2	11
3	tcp	ap_1	2	11
3	tcp	ap_0	2	11
4	udp	ap_1	2	11
4	udp	ap_0	2	11
5	tcp	ap_1	2	11
5	tcp	ap_0	2	11
6	udp	ap_1	2	11
6	udp	ap_0	2	11
7	tcp	ap_1	2	11
7	tcp	ap_0	2	11
8	udp	ap_1	2	11
8	udp	ap_0	2	11
9	tcp	ap_1	2	11
9	tcp	ap_0	2	11
10	tcp	ap_1	2	11
10	tcp	ap_0	2	11
11	udp	ap_1	2	11
11	udp	ap_0	2	11
12	tcp	ap_1	2	11
12	tcp	ap_0	2	11
13	tcp	ap_1	2	11
13	tcp	ap_0	2	11
14	tcp	ap_1	2	11
14	tcp	ap_0	2	11
15	udp	ap_1	2	11
15	udp	ap_0	2	11
16	tcp	ap_1	2	11
16	tcp	ap_0	2	11
17	udp	ap_1	2	11
17	udp	ap_0	2	11

18	tcp	ap_1	2	11
18	tcp	ap_0	2	11
19	udp	ap_1	2	11
19	udp	ap_0	2	11
20	tcp	ap_1	2	11
20	tcp	ap_0	2	11
21	udp	ap_1	2	11
21	udp	ap_0	2	11
22	tcp	ap_1	2	11
22	tcp	ap_0	2	11
23	tcp	ap_1	2	11
23	tcp	ap_0	2	11
24	udp	ap_1	2	11
24	udp	ap_0	2	11
25	tcp	ap_1	2	11
25	tcp	ap_0	2	11
26	tcp	ap_1	2	11
26	tcp	ap_0	2	11
27	tcp	ap_1	2	11
27	tcp	ap_0	2	11
28	udp	ap_1	2	11
28	udp	ap_0	2	11
29	tcp	ap_1	2	11
29	tcp	ap_0	2	11
30	udp	ap_1	2	11
30	udp	ap_0	2	11
31	tcp	ap_1	2	11
31	tcp	ap_0	2	11
32	udp	ap_1	2	11
32	udp	ap_0	2	11

## 2. test\_set\_2\_3ap.csv

test_id	protocol	ap_id	predict nss	predict mcs
1	tcp	ap_2	2	11
1	tcp	ap_0	2	5
1	tcp	ap_1	2	11
2	udp	ap_2	2	11
2	udp	ap_0	2	11

2	udp	ap_1	2	5
3	tcp	ap_2	2	11
3	tcp	ap_0	2	5
3	tcp	ap_1	2	11
4	udp	ap_2	2	11
4	udp	ap_0	2	11
4	udp	ap_1	2	11
5	tcp	ap_2	2	11
5	tcp	ap_0	2	11
5	tcp	ap_1	2	11
6	udp	ap_2	2	11
6	udp	ap_0	2	11
6	udp	ap_1	2	10
7	tcp	ap_2	2	11
7	tcp	ap_0	2	11
7	tcp	ap_1	2	11
8	udp	ap_2	2	11
8	udp	ap_0	2	11
8	udp	ap_1	2	11
9	tcp	ap_2	2	11
9	tcp	ap_0	2	11
9	tcp	ap_1	2	11
10	udp	ap_2	2	11
10	udp	ap_0	2	11
10	udp	ap_1	2	4
11	tcp	ap_2	2	4
11	tcp	ap_0	1	5
11	tcp	ap_1	2	4
12	udp	ap_2	2	4
12	udp	ap_0	2	4
12	udp	ap_1	1	5
13	tcp	ap_2	2	4
13	tcp	ap_0	1	5
13	tcp	ap_1	2	4
14	udp	ap_2	1	5
14	udp	ap_0	2	4
14	udp	ap_1	2	4
15	tcp	ap_2	2	4
15	tcp	ap_0	2	4



15	tcp	ap_1	2	4
16	udp	ap_2	2	4
16	udp	ap_0	2	4
16	udp	ap_1	2	4
17	tcp	ap_2	2	4
17	tcp	ap_0	2	7
17	tcp	ap_1	2	11
18	udp	ap_2	2	8
18	udp	ap_0	2	8
18	udp	ap_1	2	11
19	tcp	ap_2	2	11
19	tcp	ap_0	2	11
19	tcp	ap_1	2	11
20	tcp	ap_2	2	11
20	tcp	ap_0	2	5
20	tcp	ap_1	2	5
21	tcp	ap_2	2	5
21	tcp	ap_0	2	5
21	tcp	ap_1	2	5
22	tcp	ap_2	2	5
22	tcp	ap_0	2	5
22	tcp	ap_1	2	5
23	tcp	ap_2	2	5
23	tcp	ap_0	2	5
23	tcp	ap_1	2	5
24	udp	ap_2	2	5
24	udp	ap_0	2	5
24	udp	ap_1	2	5
25	tcp	ap_2	2	5
25	tcp	ap_0	2	5
25	tcp	ap_1	2	5
26	tcp	ap_2	2	5
26	tcp	ap_0	2	5
26	tcp	ap_1	2	5
27	tcp	ap_2	2	7
27	tcp	ap_0	2	5
27	tcp	ap_1	2	11
28	tcp	ap_2	2	8
28	tcp	ap_0	2	11

28	tcp	ap_1	2	11
29	tcp	ap_2	2	5
29	tcp	ap_0	2	11
29	tcp	ap_1	2	11

### 问题三结果

#### 1. test\_set\_1\_2ap.csv

test_id	protocol	ap_id	predict throughput
1	tcp	ap_1	26.139208
1	tcp	ap_0	29.892569
2	udp	ap_1	25.852642
2	udp	ap_0	29.89238
3	tcp	ap_1	24.421133
3	tcp	ap_0	28.819986
4	udp	ap_1	24.684921
4	udp	ap_0	30.203642
5	tcp	ap_1	22.547945
5	tcp	ap_0	34.66473
6	udp	ap_1	28.995037
6	udp	ap_0	32.817
7	tcp	ap_1	70.80857
7	tcp	ap_0	85.966705
8	udp	ap_1	72.00884
8	udp	ap_0	87.857956
9	tcp	ap_1	69.29766
9	tcp	ap_0	92.14989
10	udp	ap_1	75.73531
10	udp	ap_0	90.61688
11	tcp	ap_1	71.18753
11	tcp	ap_0	86.4755
12	udp	ap_1	73.47039
12	udp	ap_0	79.833176
13	tcp	ap_1	56.603397
13	tcp	ap_0	70.582794
14	udp	ap_1	57.075287
14	udp	ap_0	76.82491
15	tcp	ap_1	52.96899
15	tcp	ap_0	82.237045
16	udp	ap_1	69.48621

16	udp	ap_0	75.33587
17	tcp	ap_1	59.177593
17	tcp	ap_0	66.683975
18	udp	ap_1	81.146225
18	udp	ap_0	77.93293
19	tcp	ap_1	84.14507
19	tcp	ap_0	56.64082
20	udp	ap_1	67.88585
20	udp	ap_0	81.9548
21	tcp	ap_1	25.61847
21	tcp	ap_0	42.441185
22	udp	ap_1	25.625875
22	udp	ap_0	42.443333
23	tcp	ap_1	24.557661
23	tcp	ap_0	43.254776
24	udp	ap_1	22.803282
24	udp	ap_0	35.774426
25	tcp	ap_1	24.467499
25	tcp	ap_0	32.692165
26	udp	ap_1	22.394756
26	udp	ap_0	35.45614
27	tcp	ap_1	24.509893
27	tcp	ap_0	28.552303
28	udp	ap_1	32.10955
28	udp	ap_0	35.496
29	tcp	ap_1	50.5976
29	tcp	ap_0	55.74329
30	udp	ap_1	54.175167
30	udp	ap_0	54.39677
31	tcp	ap_1	92.83022
31	tcp	ap_0	95.52393
32	udp	ap_1	64.88321
32	udp	ap_0	77.15217
33	tcp	ap_1	91.33457
33	tcp	ap_0	87.12426
34	tcp	ap_1	73.07669
34	tcp	ap_0	78.58687
35	udp	ap_1	81.08025
35	udp	ap_0	83.93943
36	tcp	ap_1	72.15975

36	tcp	ap_0	81.25797
37	udp	ap_1	81.15649
37	udp	ap_0	69.49154
38	tcp	ap_1	87.70696
38	tcp	ap_0	63.805496
39	tcp	ap_1	84.79503
39	tcp	ap_0	62.791656
40	udp	ap_1	85.95832
40	udp	ap_0	118.76463

## 2. test\_set\_1\_3ap.csv

test_id	protocol	ap_id	predict throughput
1	tcp	ap_2	41.626408
1	tcp	ap_1	41.498085
1	tcp	ap_0	43.745792
2	udp	ap_2	43.01464
2	udp	ap_1	41.63161
2	udp	ap_0	49.044247
3	tcp	ap_2	49.32948
3	tcp	ap_1	34.541103
3	tcp	ap_0	73.30132
4	tcp	ap_2	54.2044
4	tcp	ap_1	64.986824
4	tcp	ap_0	55.191635
5	udp	ap_2	59.581245
5	udp	ap_1	58.02337
5	udp	ap_0	67.50674
6	tcp	ap_2	48.77247
6	tcp	ap_1	12.458032
6	tcp	ap_0	65.1262
7	tcp	ap_2	40.86254
7	tcp	ap_1	66.43241
7	tcp	ap_0	70.57008
8	udp	ap_2	69.14688
8	udp	ap_1	81.29226
8	udp	ap_0	63.815853
9	tcp	ap_2	85.12038
9	tcp	ap_1	85.31438
9	tcp	ap_0	61.763134

10	udp	ap_2	90.37063
10	udp	ap_1	70.8481
10	udp	ap_0	84.66796
11	tcp	ap_2	60.66858
11	tcp	ap_1	93.58863
11	tcp	ap_0	74.518295
12	udp	ap_2	43.946453
12	udp	ap_1	34.67124
12	udp	ap_0	65.651
13	tcp	ap_2	59.170013
13	tcp	ap_1	54.92107
13	tcp	ap_0	53.924564
14	udp	ap_2	54.110157
14	udp	ap_1	53.491085
14	udp	ap_0	57.2657
15	tcp	ap_2	63.51284
15	tcp	ap_1	55.933586
15	tcp	ap_0	71.02429
16	tcp	ap_2	67.04827
16	tcp	ap_1	65.13293
16	tcp	ap_0	59.1706
17	udp	ap_2	65.93501
17	udp	ap_1	72.38927
17	udp	ap_0	57.765568
18	tcp	ap_2	50.21115
18	tcp	ap_1	55.21495
18	tcp	ap_0	51.614464
19	udp	ap_2	59.376873
19	udp	ap_1	57.741703
19	udp	ap_0	54.81806
20	tcp	ap_2	46.478024
20	tcp	ap_1	53.70584
20	tcp	ap_0	49.580112
21	tcp	ap_2	48.684875
21	tcp	ap_1	60.15668
21	tcp	ap_0	44.269756
22	tcp	ap_2	43.70606
22	tcp	ap_1	55.45953
22	tcp	ap_0	54.5481
23	tcp	ap_2	53.164597

23	tcp	ap_1	49.41712
23	tcp	ap_0	61.532837
24	udp	ap_2	44.85878
24	udp	ap_1	33.458515
24	udp	ap_0	45.641876
25	tcp	ap_2	31.039455
25	tcp	ap_1	13.699479
25	tcp	ap_0	23.668692
26	tcp	ap_2	32.39947
26	tcp	ap_1	13.965358
26	tcp	ap_0	17.19192
27	udp	ap_2	46.249405
27	udp	ap_1	64.011024
27	udp	ap_0	65.83863
28	tcp	ap_2	84.40875
28	tcp	ap_1	67.66866
28	tcp	ap_0	84.589264
29	udp	ap_2	18.424278
29	udp	ap_1	39.560047
29	udp	ap_0	51.784355
30	tcp	ap_2	38.63637
30	tcp	ap_1	39.98437
30	tcp	ap_0	40.450165
31	udp	ap_2	51.40499
31	udp	ap_1	53.165478
31	udp	ap_0	50.59391
32	tcp	ap_2	51.62956
32	tcp	ap_1	40.691902
32	tcp	ap_0	33.019516
33	udp	ap_2	30.812807
33	udp	ap_1	35.180637
33	udp	ap_0	31.106968
34	tcp	ap_2	37.28293
34	tcp	ap_1	35.51568
34	tcp	ap_0	43.131653
35	udp	ap_2	44.677273
35	udp	ap_1	37.33188
35	udp	ap_0	33.928745