#### 一、数据集：

**1.1 训练集**

特征：热循环次数、电压、电流、功率、厚度、氩气、氢气、载气、送粉量、喷涂距离

目标变量：剥落面积百分比

取第六次实验的最后4条数据作为测试集，其他都作为训练集和验证集，经过网格搜索出在验证集上最优的模型，用于在测试集上的预报。共105条数据，其中喷涂距离(mm), 载气, 氢气(SPLM), 氩气(SPLM)方差为0。



**1.2 测试集**

取第六次实验的最后4条数据作为测试集，因为最后两条热循环次数为540，603的剥落比为11.7，11.8过于接近，所以取平均合并为一条数据：

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 热循环次数 | 电压(V) | 电流(A) | 功率(KW) | 厚度(μm) | 氩气(SPLM) | 氢气(SPLM) | 载气 | 送粉量(g/min) | 喷涂距离(mm) | **剥落面积百分比** |
| 439 | 68 | 650 | 44.2 | 300 | 35 | 5 | 300 | 25 | 120 | **9.258756** |
| 471 | 68 | 650 | 44.2 | 300 | 35 | 5 | 300 | 25 | 120 | **9.648355286** |
| 571 | 68 | 650 | 44.2 | 300 | 35 | 5 | 300 | 25 | 120 | **11.76395819** |

#### 二、机器学习模型结果

**2.1 Ridge**

模型参数： Ridge(alpha=12.75)

验证集R2 0.3716

|  |  |  |
| --- | --- | --- |
| **热循环次数** | **真实值** | **预测值** |
| 439 | 9.258756 | 11.54827606 |
| 471 | 9.648355286 | 12.20747852 |
| 571 | 11.76395819 | 14.26748623 |

测试集R2  -3.968633593929505

**2.2 Lasso**

模型参数： Lasso(alpha=1)

验证集R2 0.2958

|  |  |  |
| --- | --- | --- |
| **热循环次数** | **真实值** | **预测值** |
| 439 | 9.258756 | 12.35991267 |
| 471 | 9.648355286 | 13.09194711 |
| 571 | 11.76395819 | 15.37955474 |

测试集R2 -8.505493158756797

**2.3 ElasticNet**

模型参数： ElasticNet(alpha=1, l1\_ratio=0.4, max\_iter=100000)

验证集R2 0.2648

|  |  |  |
| --- | --- | --- |
| **热循环次数** | **真实值** | **预测值** |
| 439 | 9.258756 | 9.39320097 |
| 471 | 9.648355286 | 9.73929762 |
| 571 | 11.76395819 | 10.82084964 |

测试集R2 0.748028288901369

**2.4 KNeighborsRegressor**

模型参数： ElasticNet(alpha=1, l1\_ratio=0.4, max\_iter=100000)

验证集R2 0.8170

|  |  |  |
| --- | --- | --- |
| **热循环次数** | **真实值** | **预测值** |
| 439 | 9.258756 | 9.232152 |
| 471 | 9.648355286 | 9.232152 |
| 571 | 11.76395819 | 9.232152 |

测试集R2 -0.8115048521574397

**2.5 SVR**

模型参数： SVR(C=23)

验证集R2 0.8632

|  |  |  |
| --- | --- | --- |
| **热循环次数** | **真实值** | **预测值** |
| 439 | 9.258756 | 9.70684097 |
| 471 | 9.648355286 | 9.6633838 |
| 571 | 11.76395819 | 8.70102792 |

测试集R2 -1.6365274032720727

**2.6 GradientBoostingRegressor**

模型参数： GradientBoostingRegressor(min\_samples\_split=7, n\_estimators=300)

验证集R2 0.8632

|  |  |  |
| --- | --- | --- |
| **热循环次数** | **真实值** | **预测值** |
| **439** | **9.258756** | 10.38467302 |
| **471** | **9.648355286** | 8.67477266 |
| **571** | **11.76395819** | 11.1512662 |

测试集R2 0.2871317468957746

**2.7 RF**

模型参数： RandomForestRegressor(max\_features=8, n\_estimators=76)

验证集R2 0.7218

|  |  |  |
| --- | --- | --- |
| **热循环次数** | **真实值** | **预测值** |
| **439** | **9.258756** | 9.31971612 |
| **471** | **9.648355286** | 9.28635246 |
| **571** | **11.76395819** | 9.38002996 |

测试集R2  -0.6007214068606781

**2.8 XGBoost——Matadata 超参数优化搜索结果**

模型参数： XGBRegressor(base\_score=0.5, booster='gbtree', colsample\_bylevel=1,

colsample\_bynode=1, colsample\_bytree=1, enable\_categorical=False,

eta=0.3, gamma=0, gpu\_id=-1, importance\_type=None,

interaction\_constraints='', learning\_rate=0.300000012,

max\_delta\_step=0, max\_depth=3, min\_child\_weight=1, missing=nan,

monotone\_constraints='()', n\_estimators=210, n\_jobs=16,

num\_parallel\_tree=1, predictor='auto', random\_state=0, reg\_alpha=0,

reg\_lambda=1, scale\_pos\_weight=1, subsample=1, tree\_method='exact',

validate\_parameters=1, verbosity=None)

验证集R2 0.8042

|  |  |  |
| --- | --- | --- |
| **热循环次数** | **真实值** | **预测值** |
| **439** | **9.258756** | 8.525156 |
| **471** | **9.648355286** | 8.561746 |
| **571** | **11.76395819** | 8.680573 |

测试集R2  -2.088745785251392

**2.9 XGBoost——微软NNI搜索结果**

模型参数： XGBRegressor(max\_depth=3, learning\_rate=0.01, n\_estimators=700)

验证集R2 0.7423424412078885

|  |  |  |
| --- | --- | --- |
| **热循环次数** | **真实值** | **预测值** |
| **439** | **9.258756** | 9.930362 |
| **471** | **9.648355286** | 10.012045 |
| **571** | **11.76395819** | 10.477219 |

测试集R2  0.3839586389045452

相对误差 0.92613

最后一条相对误差 89.0306

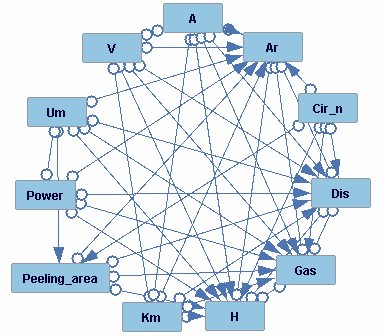
#### 三、因果分析

该实验的数据集变量说明如表1所示

**表1 数据集变量说明**

|  |  |
| --- | --- |
| 变量 | 变量含义 |
| Cir\_n | 热循环次数 |
| V | 电压(V) |
| A | 电流(A) |
| Km | 功率(KW) |
| Um | 厚度(μm) |
| Ar | 氩气(SPLM) |
| H | 氢气(SPLM) |
| Gas | 载气 |
| Power | 送粉量(g/min) |
| Dis | 喷涂距离(mm) |
| Peeling\_area | 剥落面积百分比 |

通过使用FCI和LiNGAM算法对上述数据集进行因果网络建模，可以分别得到一个初始的基本的因果网络图，如图1所示。

手机屏幕截图

低可信度描述已自动生成

(a)FCI (b)LiNGAM

图1 因果网络图

FCI 算法是一种基于约束的算法，它以输入样本数据和可选的背景知识为输入，并输出马尔可夫等价类（包括具有隐藏混杂因素的马尔可夫等价类），其中包含被判断为成立的条件独立关系集。FCI有两个阶段：邻接阶段和定向阶段。算法的邻接阶段从一个完整的无向图开始，然后执行一系列条件独立性测试，从而移除任意两个相邻变量之间的边，这些变量被判断为独立的，条件是观察变量的某个子集；任何导致去除邻接的条件集都会被存储。在邻接阶段之后，生成的无向图具有正确的邻接集，但所有边都是无向的。然后 FCI 进入定向阶段，该阶段使用存储的条件集去除邻接以定向尽可能多的边[Spirtes, 1993]。图1(a)应用FCI算法就出现了马尔可夫等价类的问题，即存在无法确定变量之间的因果方向。

为了避免出现马尔可夫等价类的问题，使用LiNGAM算法(Shimizu et al., 2006)，可以确定变量之间的因果方向。LiNGAM算法是一种基于函数分解的方法用于发现因果关系的方向性。其 思路是：在一个带有非高斯噪音的线性模型 X → Y 中，P(y)是两个非高斯分布的卷积，这个分布比 P(x) 更接近高斯分布。“比……更接近高斯分布”的关系可以进行更精确的数值估计，并用于估计特定箭头的方向性。图1(b)展示了使用LiNGAM算法后得到的因果网络图，将不存在边的变量进行剔除，如图2所示。

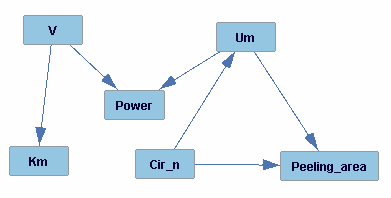


图2

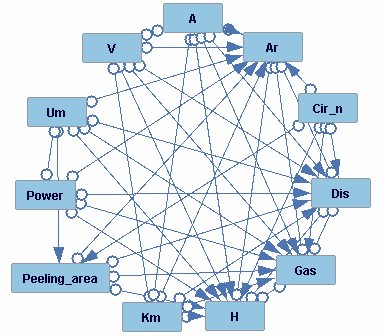
图2中包含了6条有向边，每条边表示变量之间的因果依赖方向，可以发现，Peeling\_area(剥落面积百分比)存在着两个直接原因，分别为Cir\_n(热循环次数)和Um(厚度)。这在实际中表现为Cir\_n(热循环次数)和Um(厚度)直接影响着Peeling\_area(剥落面积百分比)。当然该算法局限于不存在潜在的共同原因，即需要观测到所有变量。

该实验的数据集变量说明如表1所示

**表1 数据集变量说明**

|  |  |
| --- | --- |
| 变量 | 变量含义 |
| Cir\_n | 热循环次数 |
| V | 电压(V) |
| A | 电流(A) |
| Km | 功率(KW) |
| Um | 厚度(μm) |
| Ar | 氩气(SPLM) |
| H | 氢气(SPLM) |
| Gas | 载气 |
| Power | 送粉量(g/min) |
| Dis | 喷涂距离(mm) |
| Peeling\_area | 剥落面积百分比 |

通过使用FCI和LiNGAM算法对上述数据集进行因果网络建模，可以分别得到一个初始的基本的因果网络图，如图1所示。

手机屏幕截图

低可信度描述已自动生成

(a)FCI (b)LiNGAM

图1 因果网络图

FCI 算法是一种基于约束的算法，它以输入样本数据和可选的背景知识为输入，并输出马尔可夫等价类（包括具有隐藏混杂因素的马尔可夫等价类），其中包含被判断为成立的条件独立关系集。FCI有两个阶段：邻接阶段和定向阶段。算法的邻接阶段从一个完整的无向图开始，然后执行一系列条件独立性测试，从而移除任意两个相邻变量之间的边，这些变量被判断为独立的，条件是观察变量的某个子集；任何导致去除邻接的条件集都会被存储。在邻接阶段之后，生成的无向图具有正确的邻接集，但所有边都是无向的。然后 FCI 进入定向阶段，该阶段使用存储的条件集去除邻接以定向尽可能多的边[Spirtes, 1993]。图1(a)应用FCI算法就出现了马尔可夫等价类的问题，即存在无法确定变量之间的因果方向。

为了避免出现马尔可夫等价类的问题，使用LiNGAM算法(Shimizu et al., 2006)，可以确定变量之间的因果方向。LiNGAM算法是一种基于函数分解的方法用于发现因果关系的方向性。其 思路是：在一个带有非高斯噪音的线性模型 X → Y 中，P(y)是两个非高斯分布的卷积，这个分布比 P(x) 更接近高斯分布。“比……更接近高斯分布”的关系可以进行更精确的数值估计，并用于估计特定箭头的方向性。图1(b)展示了使用LiNGAM算法后得到的因果网络图，将不存在边的变量进行剔除，如图2所示。

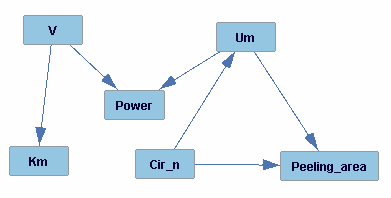


图2

图2中包含了6条有向边，每条边表示变量之间的因果依赖方向，可以发现，Peeling\_area(剥落面积百分比)存在着两个直接原因，分别为Cir\_n(热循环次数)和Um(厚度)。这在实际中表现为Cir\_n(热循环次数)和Um(厚度)直接影响着Peeling\_area(剥落面积百分比)。当然该算法局限于不存在潜在的共同原因，即需要观测到所有变量。