



基于无线测振系统的刀具状态预测

汇报人：吴昊宇，殷士翀，赵昕宇， 2025年12月30日
马张诚，卞瑞杰





目 录

CONTENTS

饮水思源
爱國榮校

1

01

绪论

02

工作原理

03

数据采集与预处理

04

深度学习模型构建

05

结论



上海交通大学

SHANGHAI JIAO TONG UNIVERSITY

绪 论

Introduction

饮水思源
爱国荣校



01 研究背景与意义

- 切削颤振是**刀具磨损**的重要诱因
- 复杂工况下刀具状态难以准确评估
- 振动监测支撑**刀具状态感知与寿命管理**

02 关键技术与研究基础

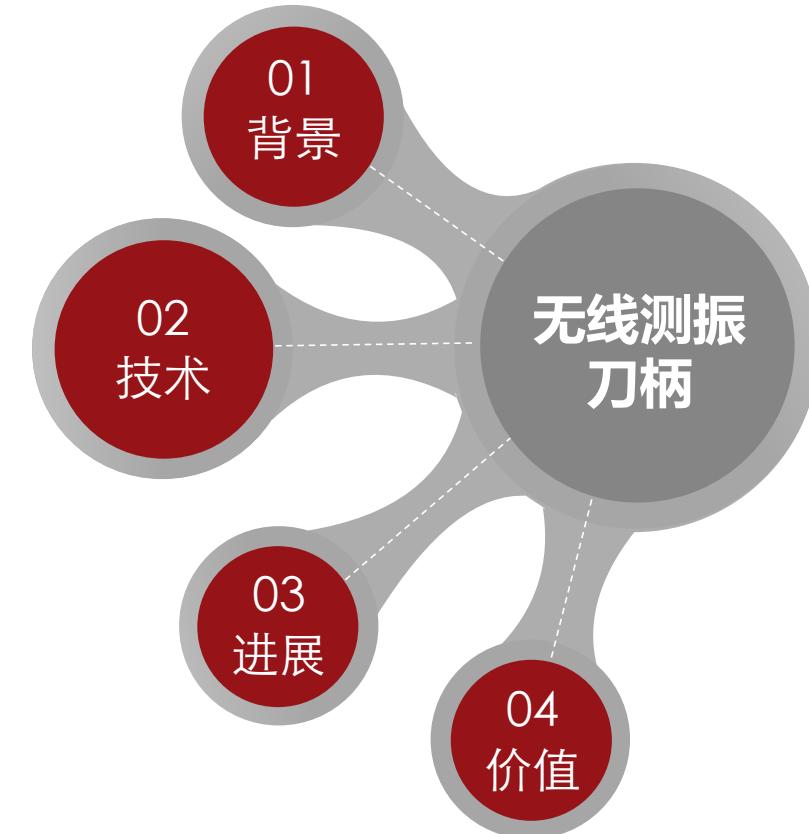
- **刀具振动监测**: 加速度计、应变片等传感技术
- **智能刀具集成**: 传感器嵌入、结构刚度与动平衡设计
- **无线数据传输**: 解决高速旋转条件下的布线问题

03 典型研究进展

- 主动控制方法抑制切削颤振
- 振动信号的在线采集与传输
- 颤振判别与加工稳定性分析

04 应用价值

- 实现刀具状态的在线监测，避免过早换刀
- 提高刀具**寿命利用率**，降低加工成本
- 在保证加工质量的前提下，提升**加工稳定性**





上海交通大学
SHANGHAI JIAO TONG UNIVERSITY

工作原理

2

饮水思源
爱国荣校

无线测振刀柄优点



无线测振刀柄



集成于刀柄内部，
靠近**振动源**



无线传输，减少电
缆干扰，信号**更直
接**



可方便地在不同刀柄间
移植，保持测量**一致性**

远离**振动源**



通过长电缆传输，易受
电磁干扰，信号**衰减大**



每次使用需重新安装和校准，
不同设备或工况下重复性差



传统外置传感器



高灵敏度压电加速度传感器

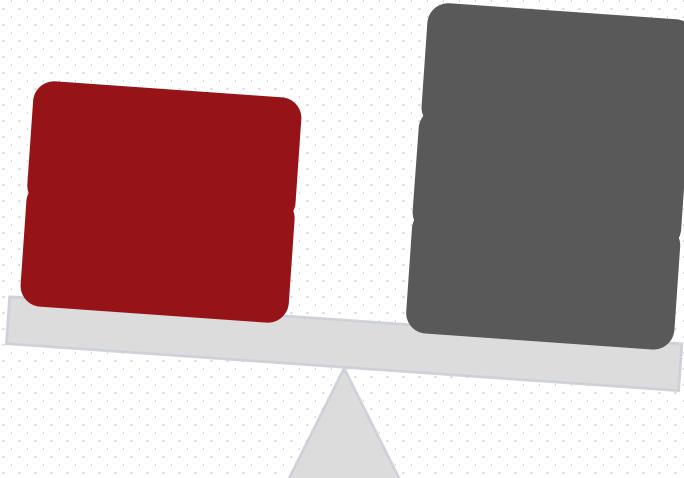


上海交通大学
SHANGHAI JIAO TONG UNIVERSITY

- 频域响应宽
- 动态范围大
- 抗干扰能力强

优点

原理



- 基于压电效应，当传感器受到振动产生**加速度**时，压电晶体受到**惯性力**作用而产生**电荷**，电荷量与加速度成正比。

电荷信号，经高输入阻抗缓冲→带通/抗混叠滤波→模数转换变为**数字数据**，再通过**蓝牙**传输至平板终端

工作原理流程图



上海交通大学
SHANGHAI JIAO TONG UNIVERSITY





上海交通大学

SHANGHAI JIAO TONG UNIVERSITY

1

数据采集与预处理

3

饮水思源
爱国荣校

数据集选择、目的

ADD YOUR TITLE HERE ADD YOUR TITLE HERE

采用Kistler声发射传感器测量切削过程中产生的高频应力波。这三个传感器采集的数据通过NI数据采集卡统一采集，形成**7通道信号**，分别是**x方向切削力、y方向切削力、z方向切削力、x方向振动、y方向振动、z方向振动和声发射信号**。每次走刀后，使用LEICMZ12显微镜测量每个刀片后刀面平均磨损带宽VB作为刀片的磨损

➤ 实验目的

进行**回归**任务，根据输入的切削信号，对于未知的刀具进行VB磨损的预测



上海交通大学
SHANGHAI JIAO TONG UNIVERSITY



数据集特征



120000*7*315*3

```
data/
  └── c1/
      ├── c_1_001.csv
      ├── c_1_002.csv
      ├── c_1_003.csv
      ...
      ... (311 个文件省略)
      └── c_1_315.csv
  (共 315 个文件)

  └── c4/
      ├── c_4_001.csv
      ├── c_4_002.csv
      ├── c_4_003.csv
      ...
      ... (311 个文件省略)
      └── c_4_315.csv
  (共 315 个文件)

  └── c6/
      ├── c_6_001.csv
      ├── c_6_002.csv
      ├── c_6_003.csv
      ...
      ... (311 个文件省略)
      └── c_6_315.csv
  (共 315 个文件)
```

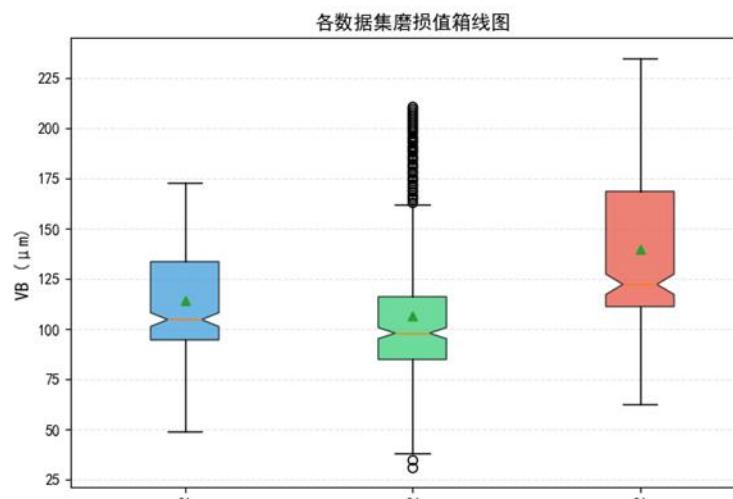
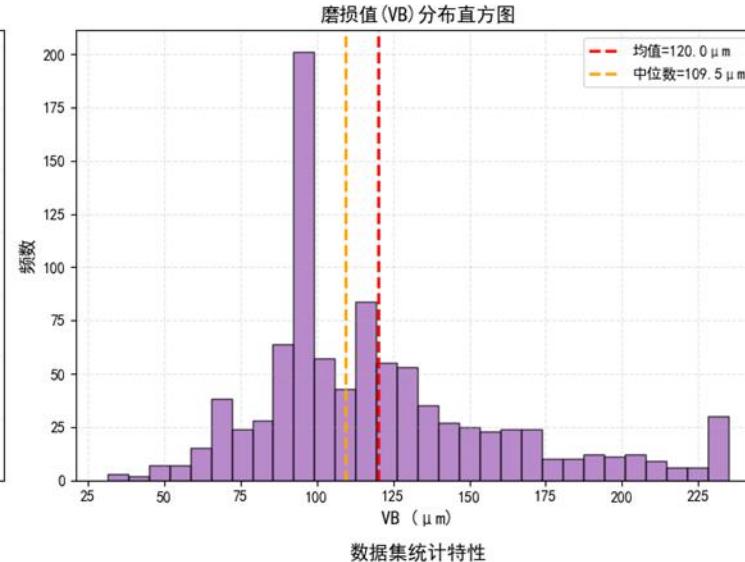
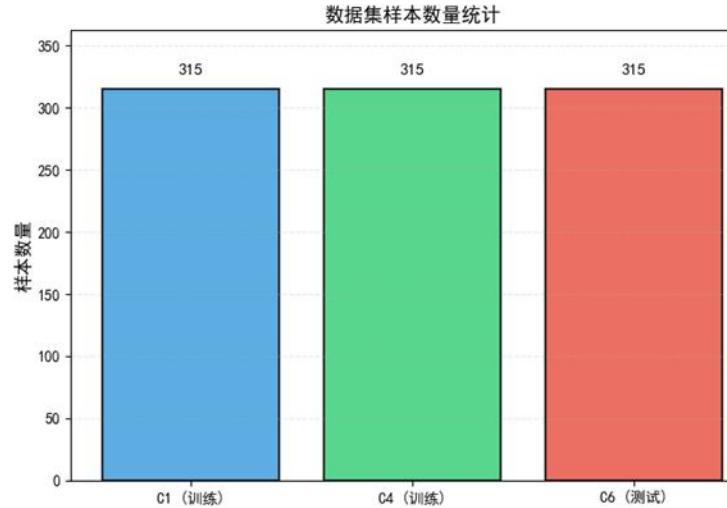
127385	0.889	1.449	0.867	-0.041	-0.041	-0.036	-0.004
127386	0.811	1.395	0.885	-0.024	-0.017	-0.02	-0.004
127387	0.755	1.396	0.908	0.175	0.183	0.182	-0.004
127388	0.636	1.283	0.878	-0.03	-0.021	-0.03	-0.004
127389	0.534	1.25	0.873	-0.023	-0.008	-0.017	-0.004
127390	0.466	1.164	0.832	0.214	0.222	0.223	-0.004
127391	0.428	1.054	0.802	0.035	0.03	0.039	-0.004
127392	0.383	0.918	0.75	-0.013	-0.017	-0.01	-0.004
127393	0.315	0.796	0.666	0.193	0.197	0.201	-0.004
127394	0.274	0.724	0.598	0.085	0.092	0.099	-0.005
127395	0.248	0.692	0.531	-0.085	-0.082	-0.071	-0.004
127396	0.245	0.689	0.483	0.184	0.186	0.194	-0.004
127397	0.22	0.63	0.415	0.067	0.066	0.073	-0.004
127398	0.218	0.537	0.346	-0.099	-0.098	-0.087	-0.004
127399	0.207	0.483	0.292	0.111	0.114	0.125	-0.004

列表内容实例

数据集特征



➤ 数据量大、数据维数多、数据处理复杂



数据集	样本数	均值 (μm)	标准差	最小值	最大值
C1	315	113.9	27.8	48.9	172.7
C4	315	106.4	38.5	31.4	210.9
O6	315	139.8	46.7	62.8	234.7
全部	945	120.0	41.0	31.4	234.7

数据预处理——原始信号特征提取



➤ 滑动窗口分割

目的: 将长时间序列分割成多个短窗口，增加训练样本数量

参数设置:

-**窗口大小**: 1024个采样点 $\approx 66.5\text{ms}$

-**步长**: 512个采样点 (50%重叠)

-**重叠策略**: 提高时间分辨率，捕获渐变的磨损过程

```
def extract_features_from_signal(signal,window_size=1024,step=512):
    for start in range(0, n - window_size +1, step):
        seg = signal[start:start+window_size]
```

➤ 时域特征提取

```
def time_domain_features(x):
    mean = np.mean(x)
    std = np.std(x)
    pp = np.max(x)- np.min(x)
    rms = np.sqrt(np.mean(x**2))
    skew = np.mean(((x - mean)/(std +1e-8))**3)
    kurt = np.mean(((x - mean)/(std +1e-8))**4)
    return[mean, std, pp, rms, skew, kurt]
```

➤ 频域特征提取

```
def freq_domain_features(x,fs=15385):
    N = len(x)
    X = np.fft.rfft(x)/ N          # FFT变换
    amp = np.abs(X)                # 频谱幅值
    freqs = np.fft.rfftfreq(N,d=1/fs)

    energy = np.sum(amp**2)
    main_freq = freqs[np.argmax(amp[1:])+1]
    mean_freq = np.sum(amp * freqs)/ np.sum(amp)
    var_freq = np.sum(amp * (freqs - mean_freq)**2)/
    np.sum(amp)
    return[energy, main_freq, mean_freq, var_freq]
```

数据预处理——特征筛选与降维



➤ 基于Pearson相关性的特征筛选

目的: 去除与目标变量 (刀具磨损VB) 相关性弱的特征, 减少噪声和冗余

方法: 计算每个特征与磨损值的Pearson相关系数

筛选准则: 保留**相关性高于0.3**的特征

```
def select_features_by_correlation(X,y,threshold=0.3):
    selected = []
    for i in range(X.shape[1]):
        r, _ = pearsonr(X[:, i], y)
        if abs(r) >= threshold:
            selected.append(i)
    return X[:, selected], selected
```

➤ Z-score标准化

作用:

1. 使所有特征服从**均值0、标准差1**的分布
2. 加速神经网络梯度下降收敛
3. 防止数值范围大的特征主导模型

```
scaler = StandardScaler()
X_train_scaled =
scaler.fit_transform(X_train_sel)
X_test_scaled = scaler.transform(X_test_sel)
```

➤ PCA主成分分析

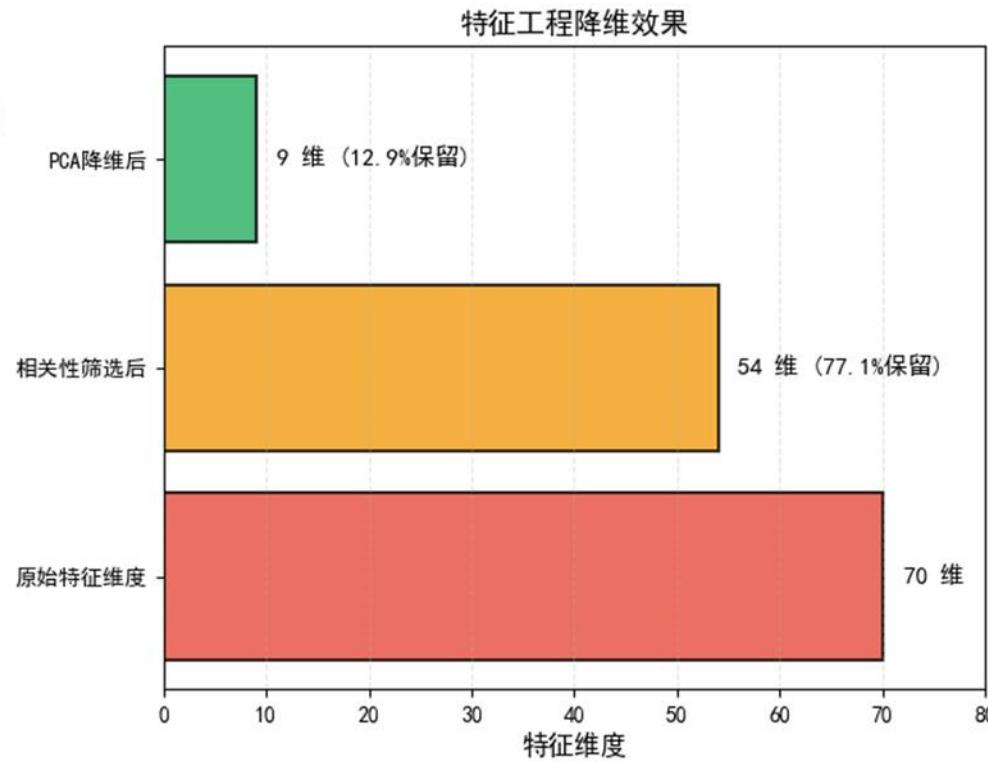
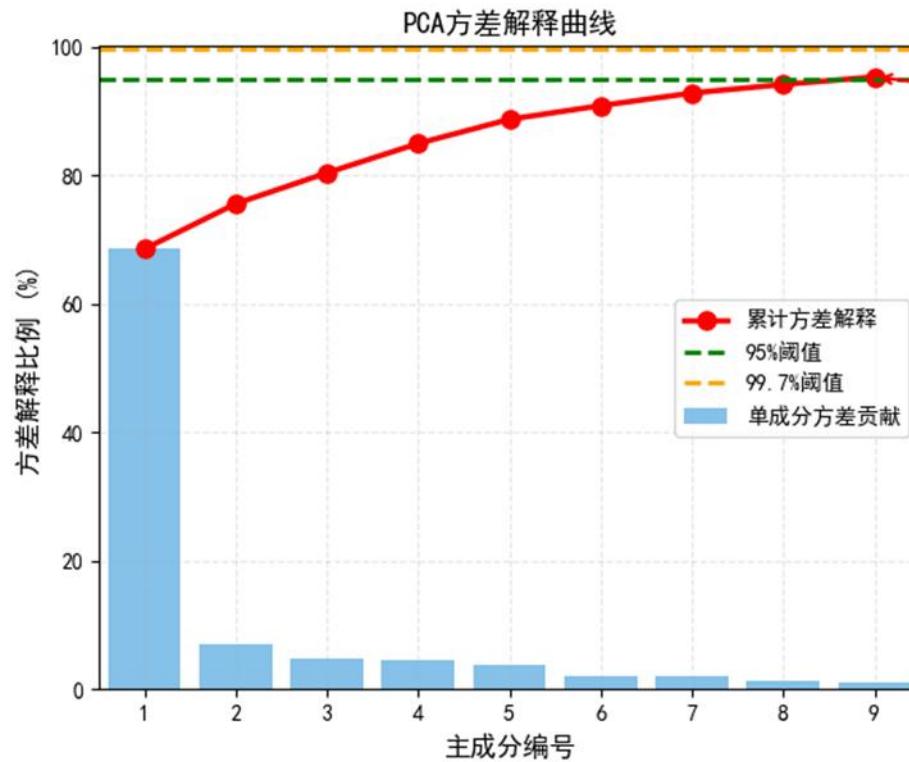
原理:

1. 计算特征的协方差矩阵
2. 求解特征值和特征向量
3. 选择前K个最大特征值对应的主成分
4. 将数据投影到新的低维空间

30-50维 → 12维

```
pca = PCA(n_components=12)
X_train_pca =
pca.fit_transform(X_train_scaled)
X_test_pca = pca.transform(X_test_scaled)
```

数据预处理——特征筛选与降维



➤ PCA主成分分析

原理：

1. 计算特征的协方差矩阵
2. 求解特征值和特征向量
3. 选择前K个最大特征值对应的主成分
4. 将数据投影到新的低维空间

30-50维 → 12维

```
pca = PCA(n_components=12)
X_train_pca =
pca.fit_transform(X_train_scaled)
X_test_pca = pca.transform(X_test_scaled)
```

数据预处理——时间序列构造



上海交通大学
SHANGHAI JIAO TONG UNIVERSITY

➤ 滑动窗口序列化

原始特征序列: [f1, f2, f3, f4, f5, f6, f7, f8, f9, f10, f11, f12, ...]

磨损标签序列: [y1, y2, y3, y4, y5, y6, y7, y8, y9, y10, y11, y12, ...]

构造的序列:

序列1: [f1, f2, f3, f4, f5, f6, f7, f8, f9, f10] → y10

序列2: [f2, f3, f4, f5, f6, f7, f8, f9, f10, f11] → y11

序列3: [f3, f4, f5, f6, f7, f8, f9, f10, f11, f12] → y12

```
def create_sequences(X,y,seq_len=10):
    xs, ys =[], []
    for i in range(len(X)- seq_len +1):
        xs.append(X[i:i+seq_len])      # 连续10个特征向量
        ys.append(y[i+seq_len-1])       # 最后时刻的磨损值
    return np.array(xs), np.array(ys)
```

➤ 数据加载器配置

```
train_loader = DataLoader(train_dataset,batch_size=32,shuffle=True)
test_loader = DataLoader(test_dataset,batch_size=32,shuffle=False)
```



上海交通大学
SHANGHAI JIAO TONG UNIVERSITY

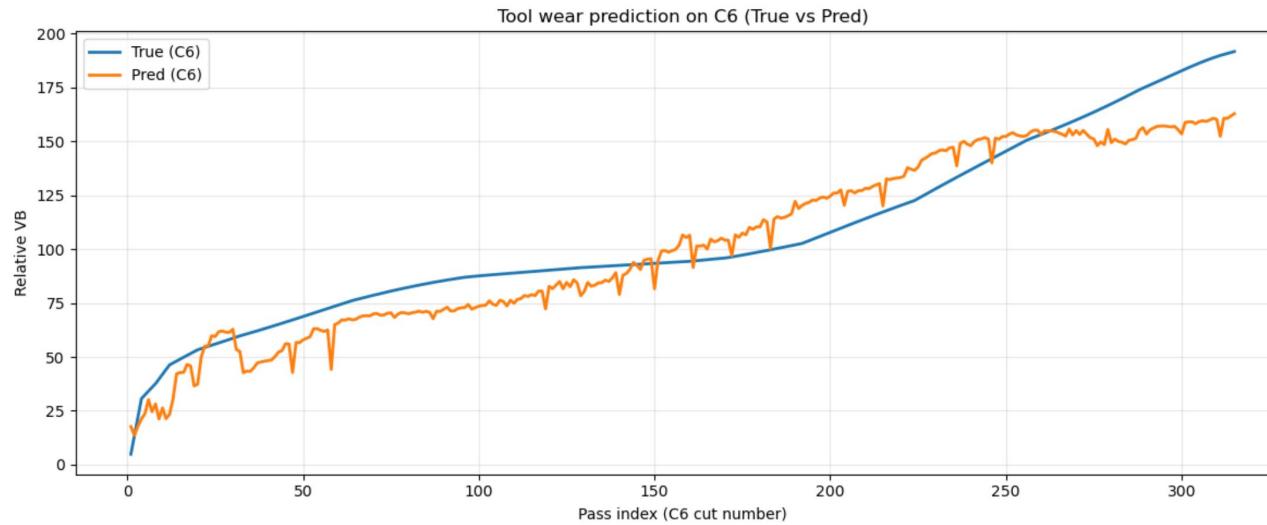
分类任务—— 深度学习模型构建



饮水思源
爱国荣校

分类任务——MLP模型

multi-layer perceptron



模型

```
def __init__(self, input_dim):
    super(MLPRegressor, self).__init__()
    self.net = nn.Sequential(
        nn.Linear(input_dim, 256),
        nn.LeakyReLU(),
        nn.BatchNorm1d(256),
        nn.Dropout(0.35),
        nn.Linear(256, 128),
        nn.LeakyReLU(),
        nn.BatchNorm1d(128),
        nn.Linear(128, 64),
        nn.LeakyReLU(),
        nn.BatchNorm1d(64),
        nn.Linear(64, 32),
        nn.LeakyReLU(),
        nn.BatchNorm1d(32),
        nn.Linear(32, 1)
    )
```

基线模型

通过全连接层提取
特征与磨损之间的
非线性映射关系

评价指标：

MAE = 12.06, RMSE = 13.79

$$\underline{R^2 = 0.88}$$



➤ MLP结构

MLP模型是一种基线模型，通过全连接层提取特征与磨损之间的非线性映射关系

特点：

- 结构简单、训练速度快
- 适合处理低维统计特征
- 对噪声敏感，表达能力有限

```
class MLP(nn.Module):
    def __init__(self, input_dim, hidden_sizes, output_dim):
        super(MLP, self).__init__()
        layers = []
        prev_dim = input_dim
        for h in hidden_sizes:
            layers.append(nn.Linear(prev_dim, h))
            layers.append(nn.ReLU())
            prev_dim = h
        layers.append(nn.Linear(prev_dim, output_dim))
        self.model = nn.Sequential(*layers)

    def forward(self, x):
        return self.model(x)
```

➤ CNN结构

CNN模型利用一维卷积自动学习切削信号的局部特征，能够有效捕获磨损演化过程中的时序变化模式

特点：

- 能自动学习时序局部特征
- 对磨损模式变化更敏感
- 需要大量窗口样本训练

```
class CNN1D(nn.Module):
    def __init__(self, num_classes, window_size=2048):
        super(CNN1D, self).__init__()

        self.conv_layers = nn.Sequential(
            nn.Conv1d(7, 16, kernel_size=5, padding=2),
            nn.ReLU(),
            nn.MaxPool1d(2),

            nn.Conv1d(16, 32, kernel_size=5, padding=2),
            nn.ReLU(),
            nn.MaxPool1d(2)
        )
```

```
        conv_output_len = window_size // 4 # 两次池化
        self.feature_dim = 32 * conv_output_len

        self.fc_layers = nn.Sequential(
            nn.Linear(self.feature_dim, 128),
            nn.ReLU(),
            nn.Linear(128, num_classes)
        )

    def forward(self, x):
        x = self.conv_layers(x)
        x = x.view(x.size(0), -1)
        x = self.fc_layers(x)
        return x
```



CNN 分支：从原始信号窗口中提取深度特征

```
def extract_features(self, x):
    """提取卷积层输出，不经过分类层"""
    x = self.conv_layers(x)
    x = x.view(x.size(0), -1)
    return x

def build_cnn_feature_per_sample(model, X_raw_windows, sample_ids, num_samples):
    X_raw_windows = np.transpose(X_raw_windows, (0, 2, 1)) # (N, 7, 2048)
    X_tensor = torch.tensor(X_raw_windows, dtype=torch.float32)

    model.eval()
    with torch.no_grad():
        feats = model.extract_features(X_tensor) # (num_windows, feature_dim)

    feats = feats.numpy()
    feat_dim = feats.shape[1]

    sample_feats = np.zeros((num_samples, feat_dim))
    counts = np.zeros(num_samples)

    for f, sid in zip(feats, sample_ids):
        sample_feats[sid] += f
        counts[sid] += 1

    counts[counts == 0] = 1
    sample_feats /= counts[:, None]

    return sample_feats
```

- MLP 分支：从 PCA 特征中提取统计特征
- 特征拼接：将两类特征在特征维度上进行 concat

```
def train_fusion_model(X_pca, cnn_feats, y_labels):
    X_fused = np.concatenate([X_pca, cnn_feats], axis=1)
    print("融合特征 shape:", X_fused.shape)

    clf = train_mlp_classifier(X_fused, y_labels)
    return clf
```

融合模型优势

- 特征互补：CNN 捕获局部时序模式，MLP 捕获全局统计特征
- 鲁棒性更强：对不同刀具、不同磨损阶段适应性更好
- 性能提升明显：在 c1、c4、c6 中均达到或接近 100% 准确率

模型评估与验证：对比实验+消融实验



上海交通大学
SHANGHAI JIAO TONG UNIVERSITY

➤ 对比实验

模型	c1 Accuracy	c1 Macro F1	c4 Accuracy	c4 Macro F1	c6 Accuracy	c6 Macro F1
MLP (PCA)	0.95	0.96	1.00	1.00	1.00	1.00
CNN (滑动窗口)	0.99	0.99	0.98	0.65	0.97	0.96
MLP+CNN (融合)	1.00	1.00	0.98	0.66	0.98	0.98

➤ 消融实验

模型	c1 Macro F1	c4 Macro F1	c6 Macro F1
CNN (step=2048)	0.99	0.65	0.96
CNN (step=1024)	0.98	0.79	0.94

融合模型性能最优：MLP+CNN 在三把刀具上表现稳定，准确率达 98–100%。

特征互补有效：去掉 CNN 或 PCA 后性能均下降，说明两类特征对模型都有贡献。

窗口重叠提升少数类识别：step=1024 明显改善 c4 的 Macro F1，增强了对严重磨损的识别能力



上海交通大学
SHANGHAI JIAO TONG UNIVERSITY

回归任务—— 深度学习模型构建

5

饮水思源
爱国荣校

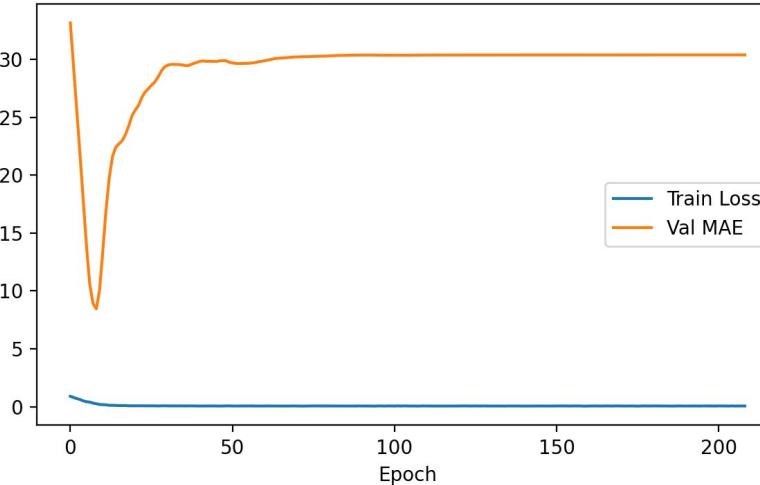
回归任务——MLP模型

multi-layer perceptron

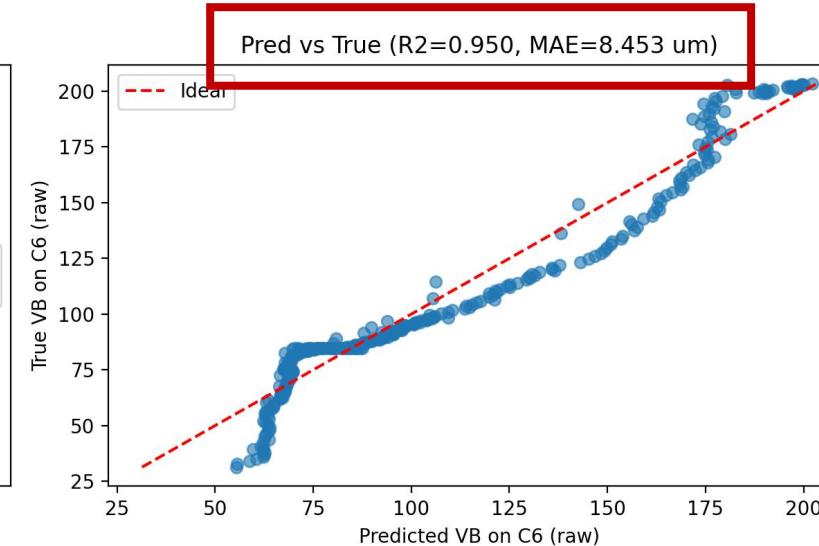


上海交通大学
SHANGHAI JIAO TONG UNIVERSITY

Training Curve (C1+C4 train)

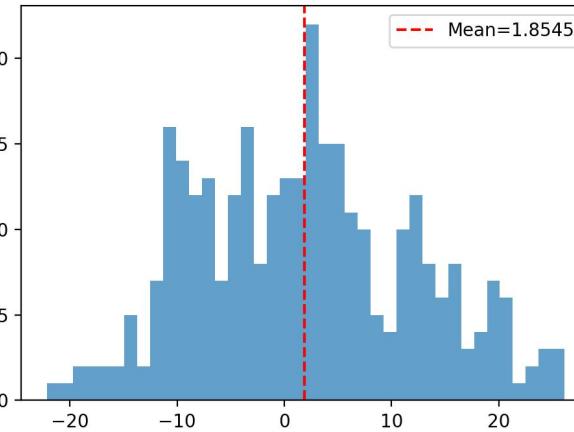


Pred vs True ($R^2=0.950$, $MAE=8.453 \mu m$)

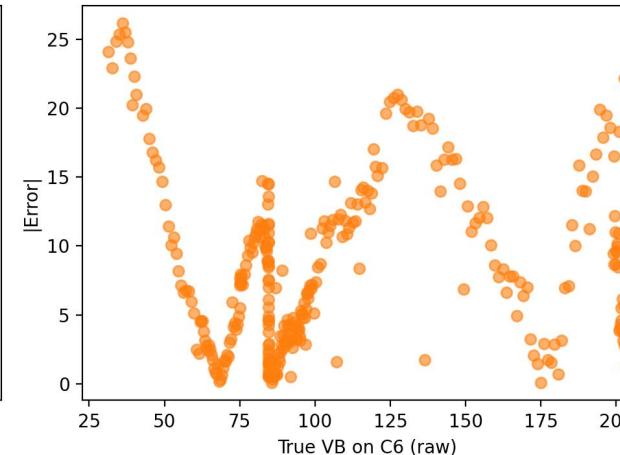


多层感知机，是基础的人工神经网络结构，通过多层全连接层实现特征的非线性映射

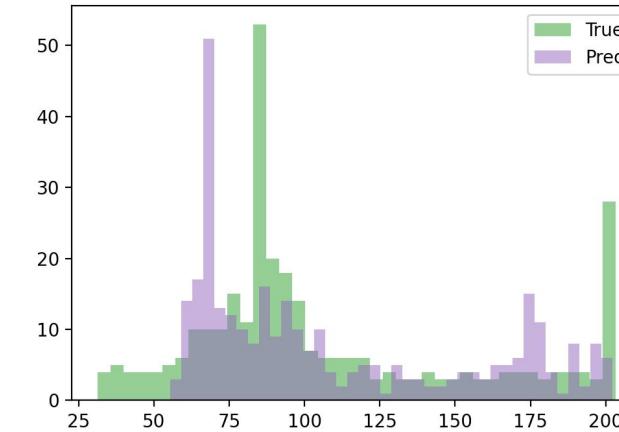
Residuals (Pred - True)



Absolute Error vs True



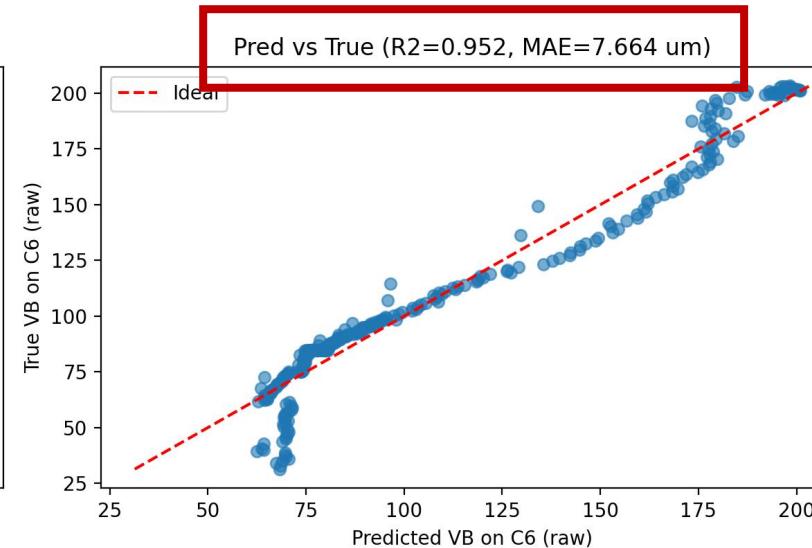
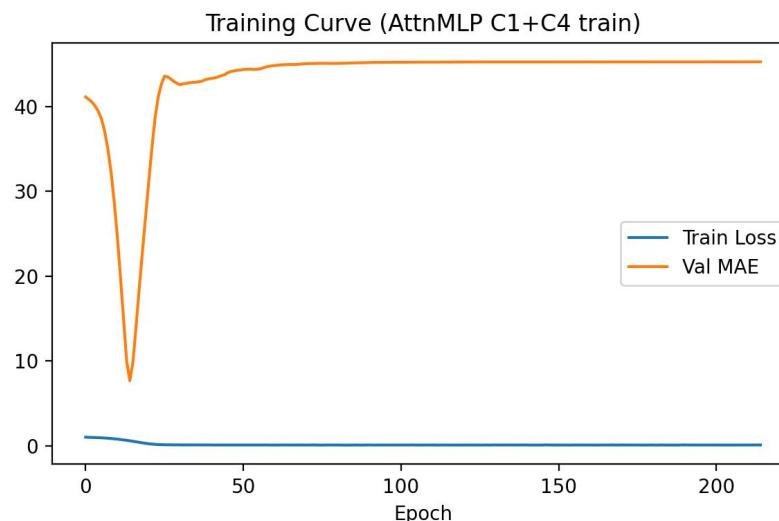
Distribution: True vs Pred (C6)



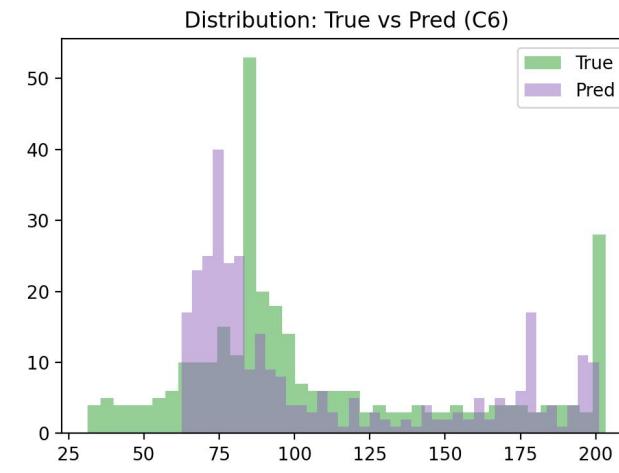
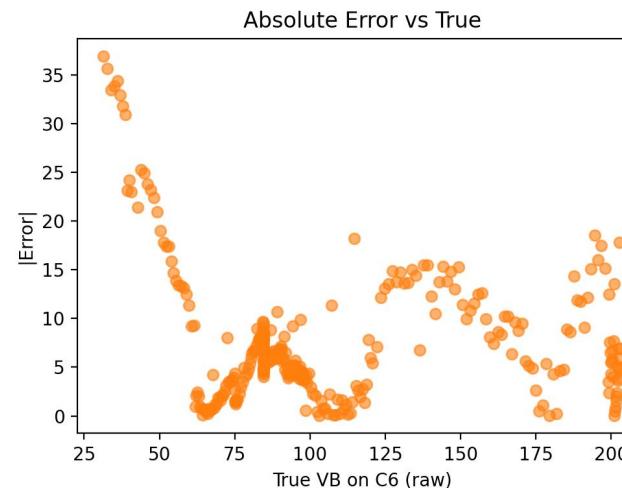
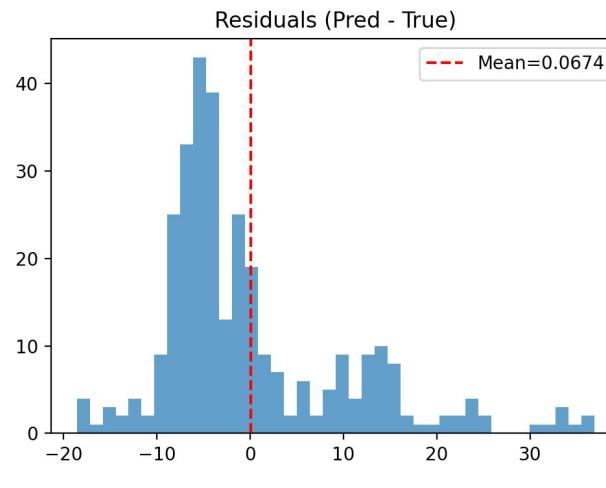
回归任务——MLP+Attention模型



multi-layer perceptron



在传统 MLP 基础上加入注意力机制，能自主聚焦输入中的关键信息，增强对重要特征的权重分配，提升模型的特征捕捉针对性

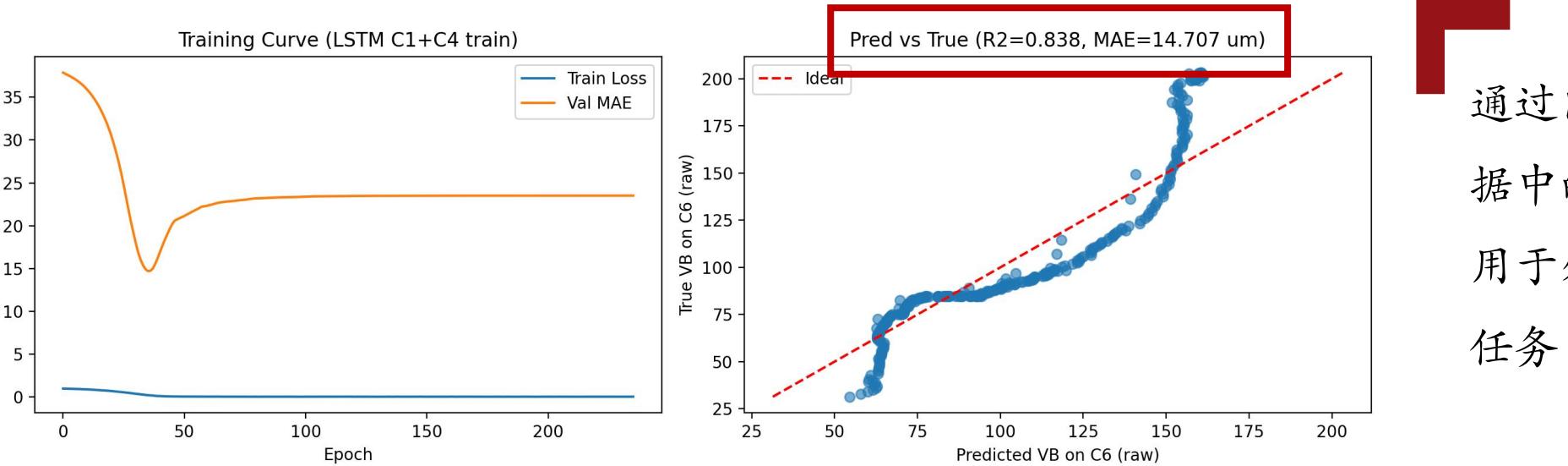


回归任务——LSTM模型

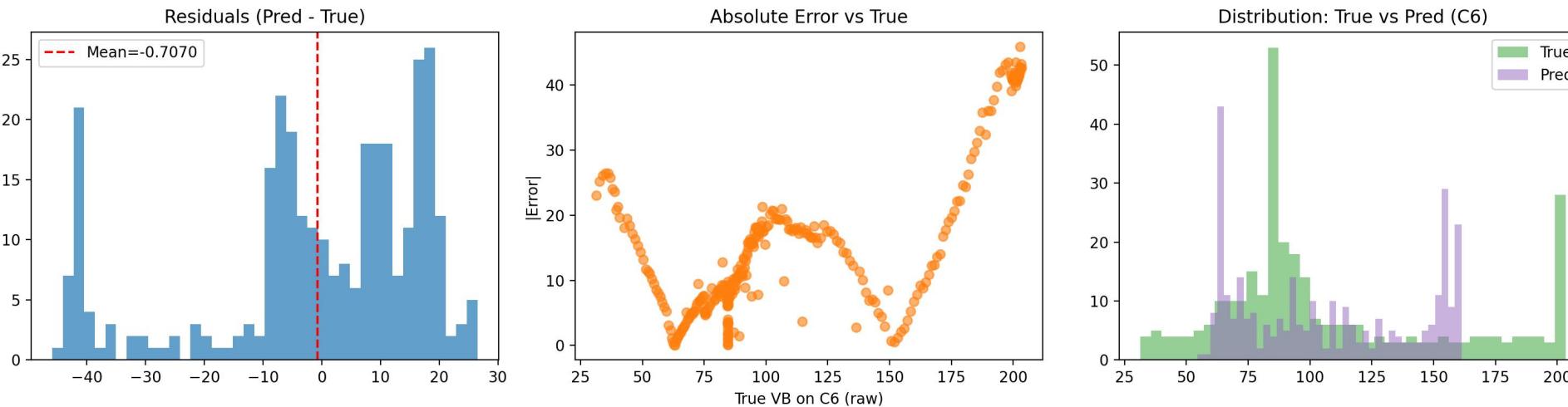
multi-layer perceptron



上海交通大学
SHANGHAI JIAO TONG UNIVERSITY



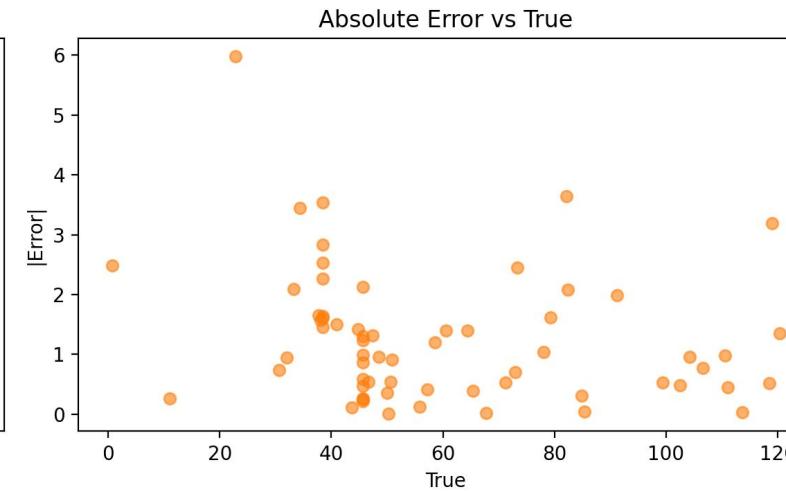
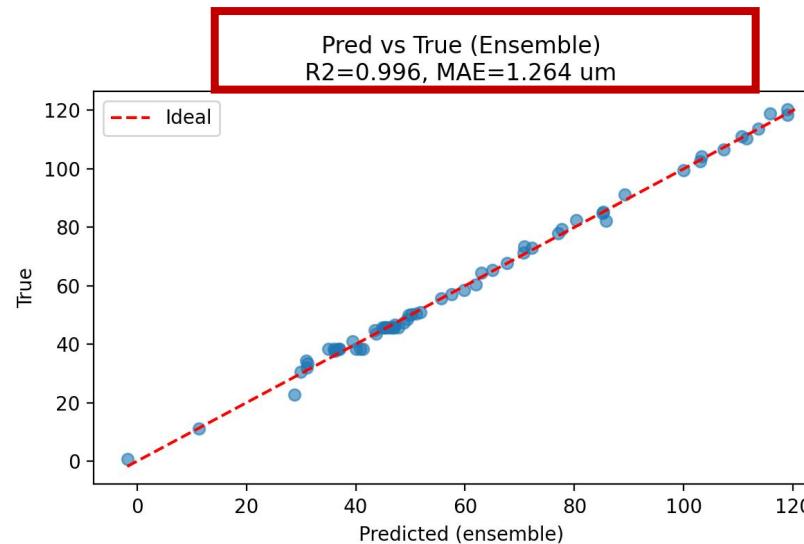
通过门控机制解决序列数据中的长期依赖问题，常用于处理时序相关的输入任务



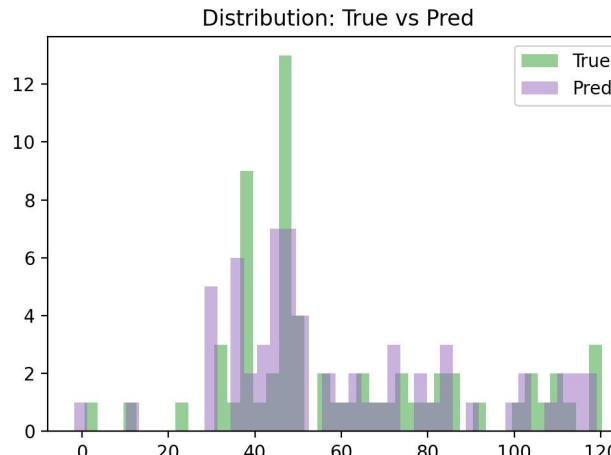
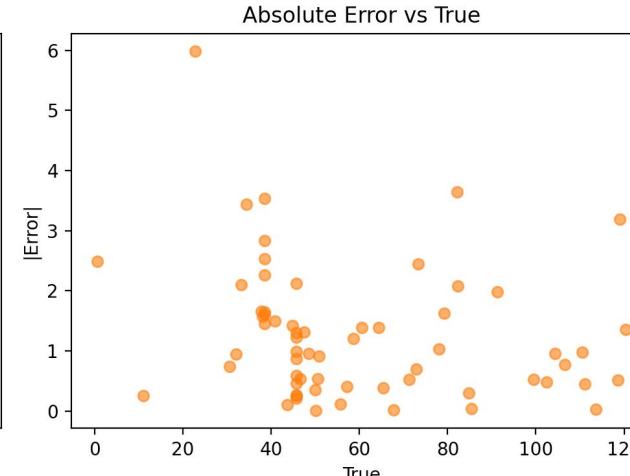
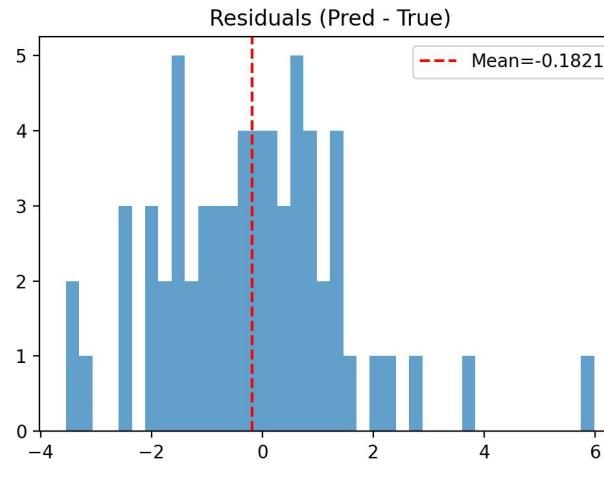
回归任务——集成学习模型



multi-layer perceptron



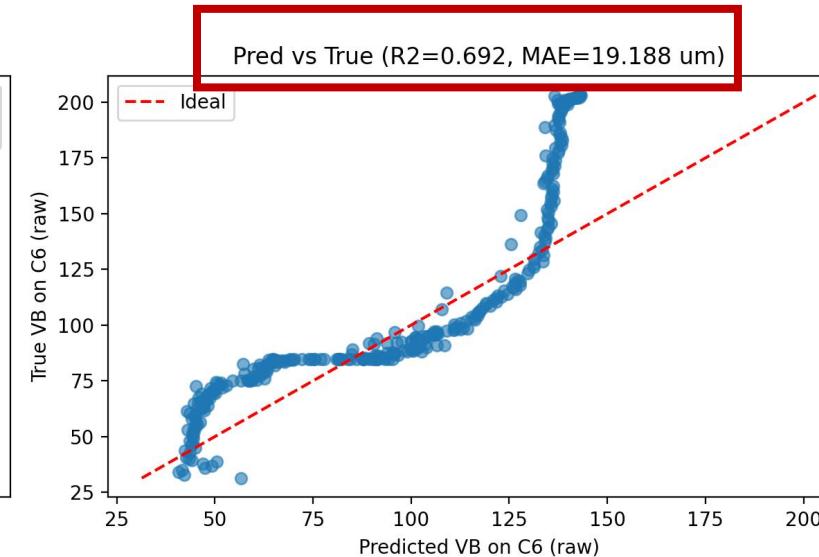
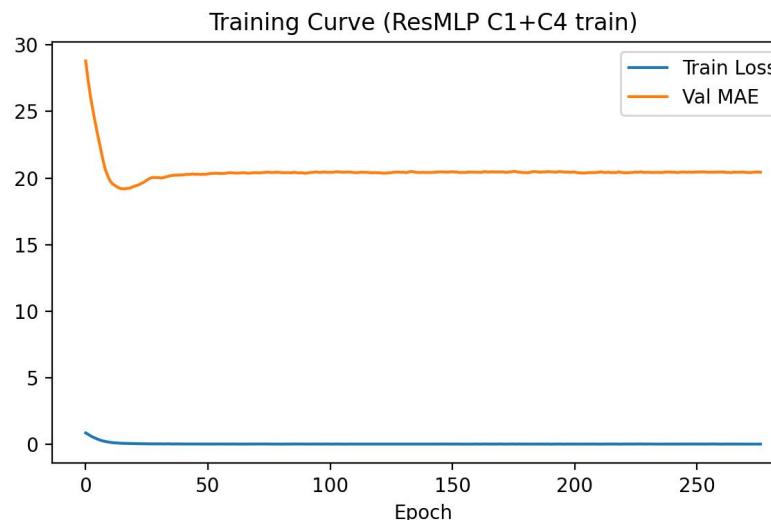
融合多个不同模型的预测结果（如投票、加权等方式），利用“群体智慧”降低单一模型的误差，提升整体稳定性与泛化能力



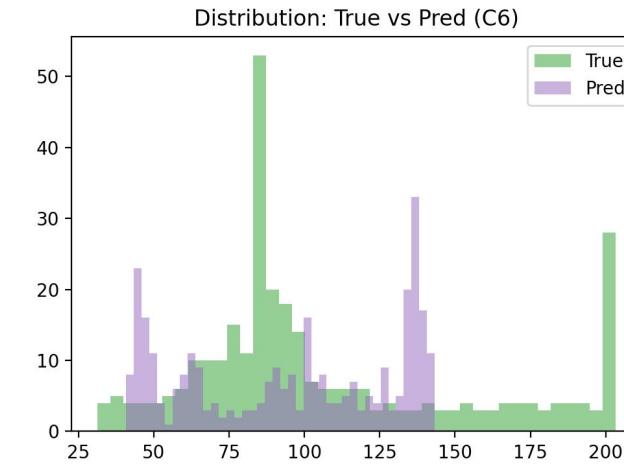
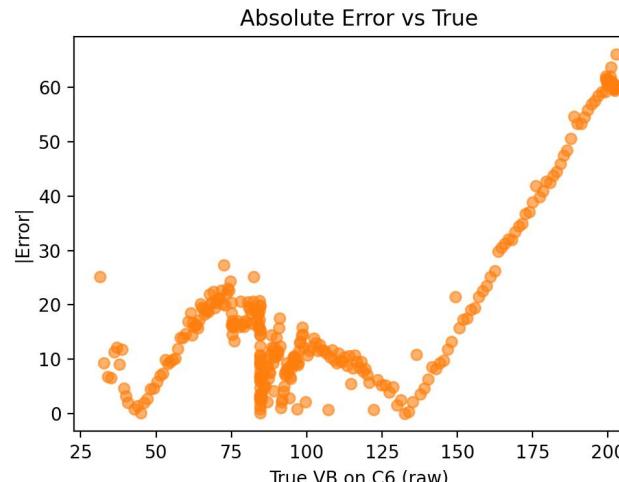
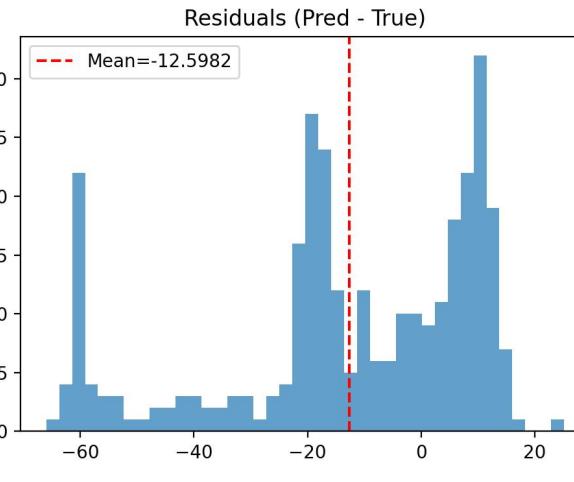
回归任务——ResMLP模型



multi-layer perceptron



结合残差连接的 MLP 变体，
通过残差结构缓解深层网
络的梯度消失问题，让模
型在增加层数的同时仍能
有效训练，保留原始特征
的直接传递





➤ 对比实验

模型	MAE (μm)	RMSE (μm)	R^2
MLP	8.453	10.412	0.95
ResMLP	19.188	25.912	0.692
MLP Attention	7.664	10.185	0.952
LSTM	14.707	18.773	0.838
EL	1.264	1.674	0.996



上海交通大学

SHANGHAI JIAO TONG UNIVERSITY

结

论

5

饮水思源
爱国荣校



方案

构建了以无线测振刀柄为核心的感知与智能分析方案

原理

将压电式加速度传感器内置于刀柄、采用Wi-Fi/蓝牙等无线链路与边缘预处理实现靠近振动源的高质量信号获取。



算法

在算法层面，形成了“滑窗分割—时域/频域特征提取—相关性筛选与降维—时序建模”的完整数据流程，并采用CNN, MLP等深度回归模型对刀具后刀面平均磨损带宽VB进行预测

应用

方案支持实时监测、寿命预测与工艺参数优化，为产线的降本增效与质量稳定提供可落地的技术路径



上海交通大学
SHANGHAI JIAO TONG UNIVERSITY

欢迎提问

Welcoming for communication.

饮水思源

爱国荣校

