```
import pandas as pd
import numpy as np
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import mean_squared_error
import tensorflow as tf

# 读取数据
data = pd.read_csv('C:/HAC.csv')
train_data = data[['Operating temp', 'Air speed', 'Relative Humidity', 'met', 'clo', 'temperature']]
train_label = data['PMV']

# 数据标准化
scaler = StandardScaler()
train_data = scaler.fit_transform(train_data)

# 构建 FNN 神经网络模型
model = tf.keras.Sequential([
    tf.keras.layers.Dense(64, activation='relu', input_shape=(train_data.shape[1],)),
    tf.keras.layers.Dense(32, activation='relu'),
    tf.keras.layers.Dense(1)
])

# 编译模型
model.compile(optimizer='adam', loss='mean_squared_error')

# 训练模型
model.fit(train_data, train_label, epochs=100, batch_size=32)

# 预测并计算均方误差
y_pred = model.predict(train_data)
mse = mean_squared_error(train_label, y_pred)
print("均方误差 (MSE):", mse)

# 将预测结果添加到数据集中
data['Predicted PMV'] = y_pred

# 构建新的训练数据集和标签
new_train_data = data[['Predicted PMV', 'Operating temp', 'Air speed', 'Relative Humidity',
'met', 'clo']]
new_train_label = data['temperature']

# 数据标准化
new_train_data = scaler.fit_transform(new_train_data)
```

```
# 构建新的 FNN 神经网络模型
new_model = tf.keras.Sequential([
    tf.keras.layers.Dense(64, activation='relu', input_shape=(new_train_data.shape[1],)),
    tf.keras.layers.Dense(32, activation='relu'),
    tf.keras.layers.Dense(1)
])

# 编译新模型
new_model.compile(optimizer='adam', loss='mean_squared_error')

# 训练新模型
new_model.fit(new_train_data, new_train_label, epochs=100, batch_size=32)

# 预测并计算均方误差
new_y_pred = new_model.predict(new_train_data)
mse = mean_squared_error(new_train_label, new_y_pred)
print("均方误差 (MSE):", mse)
```