```python
import numpy as np
import pandas as pd
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense, Dropout
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split
import joblib

# Load dataset
data = pd.read_csv('battery_data.csv')

# Select features and target variable
features = ['Battery_Temp', 'Engine_Temp', 'Voltage', 'Vibration', 'Fire']
target = ['Anomaly']  # Assuming labels are 0 (Normal) or 1 (Abnormal)

X = data[features].values  # Only use available columns
y = data[target].values

# Normalize input features
scaler = MinMaxScaler(feature_range=(0, 1))
X_scaled = scaler.fit_transform(X)

# Reshape for LSTM (samples, time steps, features)
X_reshaped = X_scaled.reshape((X_scaled.shape[0], 1, X_scaled.shape[1]))

# Split dataset
X_train, X_test, y_train, y_test = train_test_split(X_reshaped, y, test_size=0.2,
random_state=42)

# Build LSTM Model
model = Sequential([
    LSTM(50, return_sequences=True, input_shape=(1, X_reshaped.shape[2])),
    Dropout(0.2),
    LSTM(50),
    Dropout(0.2),
    Dense(1, activation='sigmoid')  # Binary Classification
])

# Compile the model
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

# Train the model
model.fit(X_train, y_train, epochs=50, batch_size=16, validation_data=(X_test,
y_test))

# Save trained model
model.save('battery_safety_lstm.h5')

# Save the scaler for preprocessing in the real-time detection script
joblib.dump(scaler, 'scaler.pkl')

print("✅ Model training complete! Model saved as 'battery_safety_lstm.h5'.")
```