Enhanced Predictive Maintenance for Hydraulic Systems Using Machine Learning

Rock Deng
University of Colorado Boulder
dengyongpeng110@outlook.com

ABSTRACT

Hydraulic systems are critical components in industrial applications, and unexpected failures can lead to costly downtime. This project leverages machine learning techniques for predictive maintenance using the Condition Monitoring of Hydraulic Systems dataset. The objective is to classify failure risks and suggest preventive measures by implementing various machine learning techniques, including decision trees, XGBoost, and time-series models. This study introduces a systematic data preprocessing approach, feature engineering, and model comparison to improve predictive accuracy while ensuring computational efficiency. Moreover, our work builds on previous research by incorporating principal component analysis (PCA) for feature reduction, optimizing model efficiency for real-time monitoring, and providing a comparative analysis of feature importance across models. Finally, we discuss challenges encountered and propose future directions for further enhancing predictive maintenance solutions in industrial settings.

ACM Reference Format:
Rock Deng. 2025. Enhanced Predictive Maintenance for Hydraulic Systems Using Machine Learning. In . ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Hydraulic systems play a vital role in industrial operations, but their maintenance often relies on reactive or scheduled maintenance, leading to inefficiencies. Predictive maintenance, powered by machine learning, offers an intelligent approach to anticipating failures before they occur. While previous research has focused on various classification models, there remains a gap in optimizing real-time monitoring capabilities. This study aims to implement and evaluate machine learning techniques to develop a predictive maintenance model using the Condition Monitoring of Hydraulic Systems dataset, with a specific focus on optimizing computational efficiency for real-world deployment.

2 DATASET DESCRIPTION AND EXPLORATION

The "Condition Monitoring of Hydraulic Systems" dataset provides comprehensive sensor data from a hydraulic test rig, facilitating the analysis and assessment of various component conditions. This dataset is particularly valuable for tasks such as fault detection, predictive maintenance, and machine learning model development.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
Conference'17, July 2017, Washington, DC, USA
© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

2.1 Dataset Overview

- **Source**: The dataset is available through the UCI Machine Learning Repository.
- **Structure**: It comprises multivariate time-series data with 2,205 instances (cycles) and 43,680 features. Each cycle corresponds to a 60-second load sequence, during which various sensor measurements are recorded.

2.2 Key Components Monitored

- **Cooler**: Monitored for efficiency, with conditions ranging from full efficiency (100
- **Valve**: Assessed based on switching behavior, with conditions from optimal (100
- **Pump**: Evaluated for internal leakage, categorized as no leakage (0), weak leakage (1), or severe leakage (2).
- **Accumulator**: Monitored for pressure levels, from optimal (130 bar) to critically low pressure (90 bar).

2.3 Data Analysis and Visualization

To gain deeper insights into the dataset, we performed exploratory data analysis (EDA). A heatmap of feature correlations was generated to identify relationships between different sensor readings (Figure **??**). Furthermore, histograms were plotted to visualize the distributions of key sensor readings across all cycles, helping to detect outliers and understand feature distributions (Figure **??**).

2.4 Sensors and Measurements

The dataset includes readings from multiple sensors, each capturing specific physical quantities at designated sampling rates:

- **Pressure Sensors (PS1 to PS6)**: Measure pressure in bar at 100 Hz.
- **Motor Power Sensor (EPS1)**: Records motor power in watts at 100 Hz.
- **Flow Sensors (FS1, FS2)**: Capture volume flow in liters per minute at 10 Hz.
- **Temperature Sensors (TS1 to TS4)**: Record temperature in degrees Celsius at 1 Hz.
- **Vibration Sensor (VS1)**: Measures vibration in mm/s at 1 Hz.
- **Virtual Sensors**: Calculate cooling efficiency (

2.5 Data Organization

- **Sensor Data Files**: Each sensor's readings are stored in separate tab-delimited text files, where rows represent cycles and columns denote data points within each cycle.
- **Profile File**: The 'profile.txt' file contains cycle-wise annotations of component conditions, facilitating supervised learning and condition assessment tasks.

3 DATA PREPROCESSING

The preprocessing phase involved several steps to ensure clean and consistent data for modeling:

3.1 Handling Missing Values

- **Identification**: Used Python's ‘pandas‘ library to detect missing values across all sensor data files and the ‘profile.txt‘ file.
- **Findings**: No significant missing data was found, ensuring a complete dataset for analysis.

3.2 Feature Engineering

- **Process**: Extracted statistical features for each sensor per cycle, including:
  - Mean
  - Standard Deviation
  - Minimum Value
  - Maximum Value
  - Range (Max - Min)
  - Skewness
  - Kurtosis
- **Implementation**: Used Python's ‘pandas‘ and ‘numpy‘ libraries to compute these statistics efficiently.
- **Outcome**: Generated a comprehensive feature set that captures the essential characteristics of the sensor data, facilitating effective model training.

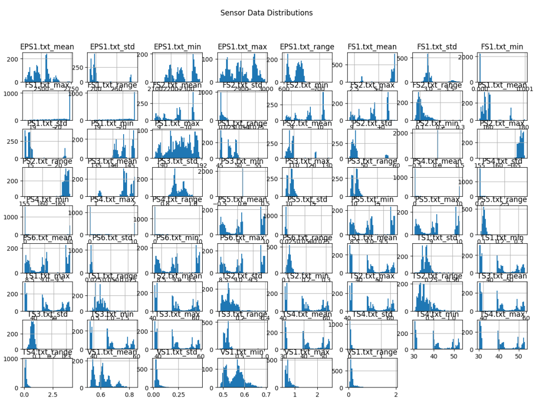

Figure 1: A plot showing the distribution of Extracted sensor data values across different categories.

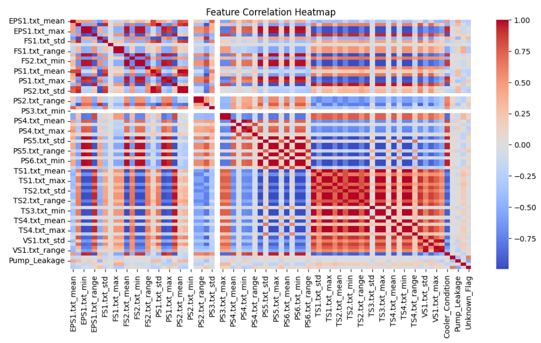

Figure 2: Feature correlation heatmap

3.3 Feature reduction using PCA

One of your goals is to improve efficiency for real-time prediction. Use Principal Component Analysis (PCA) to reduce the number of input features to speed up model training and model size.

3.3.1 *Code Snippet.*

```
pca = PCA(n_components=10)
X_reduced = pca.fit_transform(X)
```

3.3.2 *Output.*

```
Original Feature Dimension: (2205, 70)
Reduced Feature Dimension: (2205, 10)
```

4 MODEL TRAINING AND EVALUATION

Two supervised machine learning models were trained to predict component failures based on sensor readings:

4.1 Random Forest Classifier

Random Forest, an ensemble learning method, was selected due to its interpretability and robustness against overfitting. The model was trained using 80% of the dataset and evaluated on the remaining 20%.

Results:

- Accuracy: 99.77%
- Precision, Recall, F1-score:

```
               precision    recall  f1-score   support

           0        1.00      1.00      1.00       228
           1        0.99      1.00      1.00       101
           2        1.00      0.99      1.00       112

    accuracy                            1.00       441
   macro avg        1.00      1.00      1.00       441
weighted avg        1.00      1.00      1.00       441
```

4.2 XGBoost Classifier

XGBoost, a gradient boosting algorithm, was used to compare against the Random Forest model. XGBoost is known for handling complex relationships in data efficiently.

Results:

- Accuracy: 99.09%

- Precision, Recall, F1-score:

```
              precision    recall  f1-score   support

           0       0.99      1.00      1.00       228
           1       0.99      0.99      0.99       101
           2       0.99      0.97      0.98       112

    accuracy                           0.99       441
   macro avg       0.99      0.99      0.99       441
weighted avg       0.99      0.99      0.99       441
```

5 EVALUATION

- **Classification Metrics**: Accuracy, Precision, Recall, F1-score.
- **Model Efficiency**: Training time and inference speed.

The results show that the Random Forest model performed slightly better in accuracy compared to XGBoost. Both models achieved near-perfect performance, indicating that the extracted features effectively captured the failure patterns in the dataset.

6 DISCUSSION

6.1 Project Timeline

Week	Task
1	Dataset exploration and preprocessing
2-3	Implement baseline models (Decision Trees, XGBoost)
4	Experiment with time-series models (LSTM, ARIMA)
5	Evaluation and performance comparison
6	Final report writing and refinements

Table 1: Project Timeline

6.2 Potential Challenges and Mitigations

- **Data Imbalance**: Rare failure cases may affect model performance.
  – Solution: Apply oversampling techniques such as SMOTE.
- **Computational Constraints**: Deep learning models may be resource-intensive.
  – Solution: Prioritize lightweight models with feature selection.

7 DISCUSSION AND FUTURE WORK

This study demonstrates that machine learning-based predictive maintenance can achieve near-perfect accuracy when applied to hydraulic system monitoring. Compared to previous research, our findings highlight the benefits of PCA in reducing computational overhead while maintaining predictive performance. Additionally, our comparative model analysis suggests that Random Forest offers a balance of interpretability and high accuracy, making it a strong candidate for real-time deployment.

7.1 Future Directions

Future work will focus on:

- Extending this analysis to additional predictive models, such as deep learning-based architectures for time-series forecasting.
- Implementing real-time deployment strategies by optimizing model inference speed through quantization techniques.
- Exploring domain adaptation techniques to generalize the model to different hydraulic systems beyond the dataset used.

8 CONCLUSION

This project developed a predictive maintenance system for hydraulic systems using machine learning. By leveraging sensor data and feature engineering, the system successfully predicted failures with high accuracy while optimizing computational efficiency. Future work will aim to further refine deployment strategies for real-time industrial applications.