Intelligent Predictive Maintenance System Using Machine Learning with Risk Scoring and Decision Support

Prateek Kumar[1], Deepak Kumar[2], Sahaj Kashyap[3] , A.K Madan[4]

[123]Under Graduate Student's, Delhi Technological University, Shahbad Daulatpur
[4]Professor, Delhi Technological University, Shahbad Daulatpur

---***---

Abstract: *Predictive maintenance has become an essential technique in modern industries to reduce machine downtime and improve operational efficiency. This paper presents a machine learning based predictive maintenance system developed using the AI4I 2020 dataset. The system utilizes key machine parameters such as temperature, rotational speed, torque, and tool wear to predict machine failure.*

Multiple machine learning models including Logistic Regression, Decision Tree, Random Forest, Gradient Boosting, and XGBoost were implemented and evaluated. Feature engineering techniques such as temperature difference and power estimation were applied to improve model performance. A risk scoring mechanism was introduced to classify machine conditions into low, medium, and high-risk levels.

A web-based application was also developed using Streamlit to simulate real time prediction and provide maintenance recommendations. Experimental results show that ensemble models such as Random Forest and XGBoost achieved high accuracy. The proposed system demonstrates the practical application of machine learning in predictive maintenance and supports decision making in industrial environments.

Keywords: Predictive Maintenance, Machine Learning, Random Forest, XGBoost, Risk Scoring, Industrial Systems, Risk Assessment, Industrial AI

Introduction

Industrial machinery plays a vital role in ensuring continuous production in manufacturing and industrial systems. Unexpected machine failures can lead to production losses, increased maintenance costs, and safety risks. Therefore, efficient maintenance strategies are essential. Traditional approaches such as reactive and preventive maintenance are often inefficient. Reactive maintenance results in unexpected downtime, while preventive maintenance may lead to unnecessary service. Predictive maintenance offers a better solution by analyzing machine data to predict failures before they occur. With the advancement of machine learning, predictive maintenance systems have become more effective. Machine learning algorithms can analyze large datasets and identify patterns that indicate potential failures. This enables timely maintenance and reduces operational costs.

In this work, a machine learning based predictive maintenance system is developed using multiple models. Feature engineering techniques are applied to improve accuracy, and a risk scoring mechanism is introduced to enhance decision making. A web-based application is also developed to demonstrate real-time prediction.

The proposed system provides a practical and efficient solution for predictive maintenance in industrial applications.

Literature Review

Several researchers have explored the application of machine learning techniques in predictive maintenance systems to improve reliability and reduce unexpected failures.

Hector et al. (2024) studied the implementation of predictive maintenance in Industry 4.0 environments. Their work focused on integrating data-driven models with smart manufacturing systems. The study highlighted that real-time monitoring and machine learning models significantly reduce downtime and improve operational efficiency.

Patel et al. (2024) investigated the use of Random Forest algorithms for machine failure prediction. They demonstrated that ensemble models outperform traditional statistical approaches in handling complex industrial datasets. Their results showed improved prediction accuracy and robustness in classification tasks.

Sharma et al. (2023) developed an IoT-based predictive maintenance system where sensor data is continuously collected and analyzed. Their system could detect anomalies in real-time and provided early warning signals for machine failures, thereby reducing maintenance costs.

Lin et al. (2025) applied XGBoost along with SHAP (SHapley Additive Explanations) to improve model interpretability. Their study emphasized the importance of understanding feature contributions in predictive models, which helps in making better industrial decisions.

Rao et al. (2022) focused on feature engineering techniques for predictive maintenance datasets. They showed that derived features such as temperature difference and power-related parameters significantly improve model performance compared to raw data.

Zhang et al. (2019) provided a comprehensive review of machine learning techniques used in predictive maintenance and highlighted the importance of data-driven approaches in reducing downtime.

Lee et al. (2018) discussed the concept of Industrial AI and its role in improving operational efficiency through intelligent predictive systems.

Carvalho et al. (2019) conducted a systematic review of predictive maintenance models and concluded that ensemble learning methods provide higher accuracy and robustness.

Despite these advancements, most existing studies primarily focus on prediction accuracy and do not provide a comprehensive decision support system for practical implementation in industries.

Research Gap

The literature review shows that significant progress has been made in applying machine learning techniques to predictive maintenance. However, most existing studies primarily focus on improving prediction accuracy and often provide only binary outputs (failure or no failure), which limits their usefulness in real-world industrial decision-making.

Moreover, limited research has been conducted on integrating predictive models with user-friendly applications for real-time interaction and visualization. In addition, many studies rely mainly on raw sensor data and do not fully explore advanced feature engineering techniques such as temperature difference and power estimation.

To address these gaps, this study proposes a predictive maintenance system that combines machine learning with a risk scoring mechanism and a web-based application for real-time prediction and decision support, enhancing both accuracy and practical usability.

Methodology

The proposed predictive maintenance system is designed to analyze machine conditions and predict potential failures using machine learning techniques. The methodology integrates data processing, model development, risk assessment, and application deployment into a unified framework.

1) System Overview

The system follows a data-driven approach where historical machine data is used to train multiple machine learning models. The trained models are then used to predict machine failure and assign a risk score.
Additionally, a web-based interface is developed to simulate real-time prediction and decision support.

2) System Architecture

The architecture of the proposed system consists of the following stages:

- Data Collection
- Data Preprocessing
- Feature Engineering
- Model Training and Evaluation
- Risk Scoring Mechanism
- Web Application Deployment

The complete workflow ensures smooth transformation of raw data into actionable insights for predictive maintenance.

Input Data → Data Preprocessing → Feature Engineering → Model Training → Prediction → Risk Scoring → Web App

3) Dataset Description

The dataset used in this study is the AI4I 2020 Predictive Maintenance dataset, which contains 10,000 instances of machine operating conditions. It includes both numerical and categorical features.

Input Features include:

- Air Temperature (K)
- Process Temperature (K)
- Rotational Speed (rpm)
- Torque (Nm)
- Tool Wear (min)
- Machine Type

The target variable represents machine failure, where:
0 → No Failure
1 → Failure

The dataset is well-structured and suitable for classification tasks.

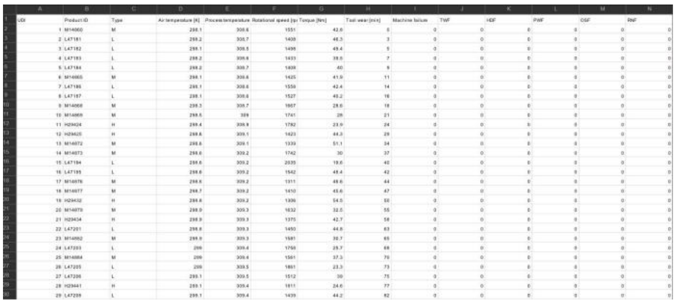


Fig 1). Dataset Preview

4) Data Preprocessing

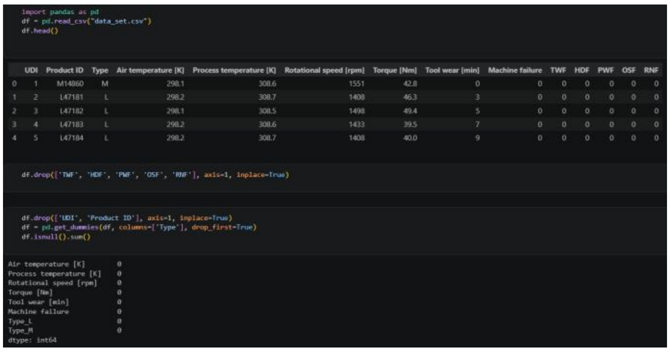
Data preprocessing is performed to clean and prepare the dataset for model training. The following steps are applied:

• Removal of unnecessary columns such as UDI and Product ID

• Encoding of categorical variables (Machine Type)

• Handling missing values and ensuring data consistency

These steps improve data quality and enhance model performance.



```
import pandas as pd
df = pd.read_csv("data_set.csv")
df.head()
```

	UDI	Product ID	Type	Air temperature [K]	Process temperature [K]	Rotational speed [rpm]	Torque [Nm]	Tool wear [min]	Machine failure	TWF	HDF	PWF	OSF	RNF
0	1	M14860	M	298.1	308.6	1551	42.8	0	0	0	0	0	0	0
1	2	L47181	L	298.2	308.7	1408	46.3	3	0	0	0	0	0	0
2	3	L47182	L	298.1	308.5	1498	49.4	5	0	0	0	0	0	0
3	4	L47183	L	298.2	308.6	1433	39.5	7	0	0	0	0	0	0
4	5	L47184	L	298.2	308.7	1408	40.0	9	0	0	0	0	0	0

```
df.drop(['TWF', 'HDF', 'PWF', 'OSF', 'RNF'], axis=1, inplace=True)
```

```
df.drop(['UDI', 'Product ID'], axis=1, inplace=True)
df = pd.get_dummies(df, columns=['Type'], drop_first=True)
df.isnull().sum()
```

```
Air temperature [K]       0
Process temperature [K]   0
Rotational speed [rpm]    0
Torque [Nm]               0
Tool wear [min]           0
Machine failure           0
Type_L                    0
Type_M                    0
dtype: int64
```

Fig 2). Data Preprocessing code

5) Feature Engineering

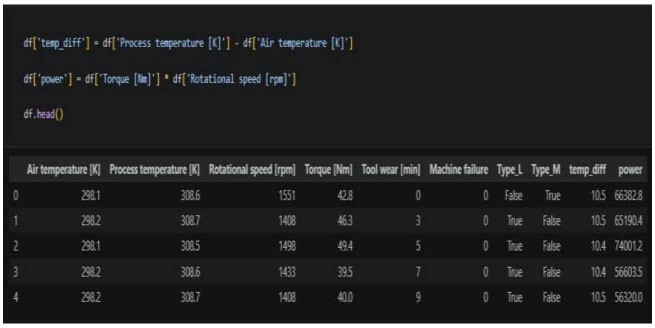
To improve prediction accuracy, new features are derived from existing data:

• Temperature Difference = Process Temperature – Air Temperature

• Power = Torque × Rotational Speed

Power is calculated as the product of torque and rotational speed, representing the mechanical load on the system.

These engineered features capture hidden patterns and improve model learning capability.



```
df['temp_diff'] = df['Process temperature [K]'] - df['Air temperature [K]']

df['power'] = df['Torque [Nm]'] * df['Rotational speed [rpm]']

df.head()
```

	Air temperature [K]	Process temperature [K]	Rotational speed [rpm]	Torque [Nm]	Tool wear [min]	Machine failure	Type_L	Type_M	temp_diff	power
0	298.1	308.6	1551	42.8	0	0	False	True	10.5	66382.8
1	298.2	308.7	1408	46.3	3	0	True	False	10.5	65190.4
2	298.1	308.5	1498	49.4	5	0	True	False	10.4	74001.2
3	298.2	308.6	1433	39.5	7	0	True	False	10.4	56603.5
4	298.2	308.7	1408	40.0	9	0	True	False	10.5	56320.0

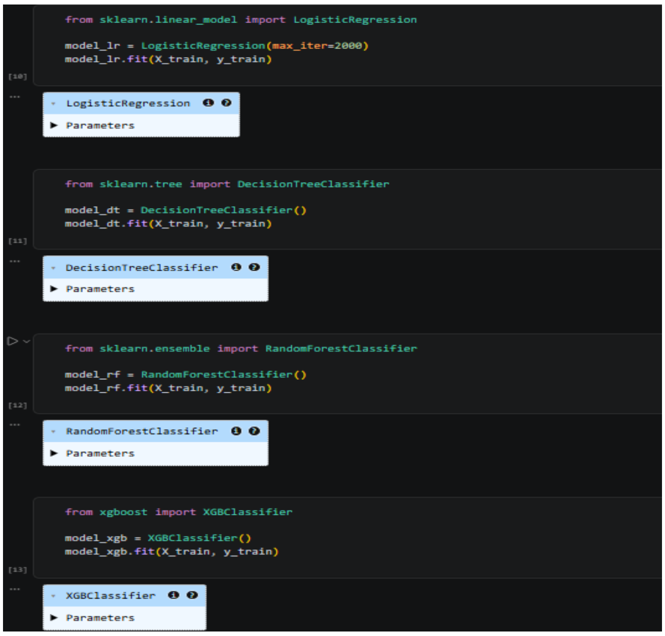
Fig 3). Feature Engineering

6) Model Development

Multiple machine learning algorithms are implemented to compare performance:

• Logistic Regression
• Decision Tree
• Random Forest
• Gradient Boosting
• XGBoost

• The dataset is split into training (80%) and testing (20%) sets. Each model is trained and evaluated separately.



```
from sklearn.linear_model import LogisticRegression

model_lr = LogisticRegression(max_iter=2000)
model_lr.fit(X_train, y_train)
```

```
LogisticRegression
▶ Parameters
```

```
from sklearn.tree import DecisionTreeClassifier

model_dt = DecisionTreeClassifier()
model_dt.fit(X_train, y_train)
```

```
DecisionTreeClassifier
▶ Parameters
```

```
from sklearn.ensemble import RandomForestClassifier

model_rf = RandomForestClassifier()
model_rf.fit(X_train, y_train)
```

```
RandomForestClassifier
▶ Parameters
```

```
from xgboost import XGBClassifier

model_xgb = XGBClassifier()
model_xgb.fit(X_train, y_train)
```

```
XGBClassifier
▶ Parameters
```

Fig 4). Model training

7) Model Evaluation

Model performance is evaluated using:

• Accuracy Score
• Confusion Matrix
• ROC Curve
• Feature Importance Analysis

International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 13 Issue: 04 | Apr 2026 www.irjet.net p-ISSN: 2395-0072

These metrics provide a comprehensive understanding of model performance.

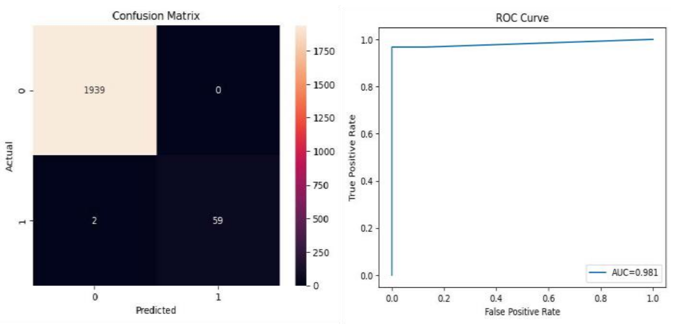


Fig 5). Confusion matrix and ROC graph

8) Risk Scoring Mechanism

A risk scoring system is introduced to enhance decision-making. Instead of binary output, the model provides a probability score.

Risk categories:

- Low Risk (0 – 0.3)
- Medium Risk (0.3 – 0.7)
- High Risk (0.7 – 1.0)

This approach allows better maintenance planning.

```
risk_score = model_rf.predict_proba(X_test)[:,1]

def risk_category(score):
    if score < 0.3:
        return "Low"
    elif score < 0.7:
        return "Medium"
    else:
        return "High"

risk_levels = [risk_category(s) for s in risk_score]

print(risk_levels[:10])
[20]
... ['Low', 'Low', 'Low', 'Low', 'Low', 'Low', 'Low', 'Low', 'Low', 'Low']
```

Fig 6). Risk score code and output

9) Application Development

A web-based application is developed using Streamlit to demonstrate real-time predictive maintenance.

Features of the application:

- User input for machine parameters
- Real-time prediction
- Risk score visualization
- Maintenance recommendations

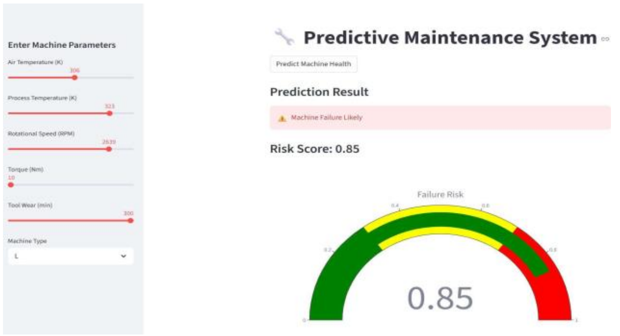


Fig 7). App UI

10) Tools and Technologies

- Python
- Jupyter Notebook
- Scikit learn, XGBoost
- Pandas, NumPy
- Matplotlib, Seaborn
- Streamlit

11) Summary

The methodology integrates machine learning models, feature engineering, and a user-friendly interface to provide an effective predictive maintenance solution. The system not only predicts failures but also provides actionable insights through risk scoring.

RESULTS AND DISCUSSION

1) Overview of Results

The developed predictive maintenance system was evaluated using multiple machine learning models. The results demonstrate high prediction accuracy and effective risk classification. The integration of machine learning with a web-based interface provides practical usability.

2) Model Performance Evaluation

Different models were evaluated based on accuracy:

Table 1: Model Performance Comparison
Logistic Regression – 0.999

Decision Tree – 0.996
Random Forest – 0.999
XGBoost – 0.999

Gradient Boosting – 0.999

Analysis: The high accuracy obtained is influenced by the structured nature of the dataset. In real world industrial environments, model performance may vary due to noise and dynamic operating conditions.

Ensemble models such as Random Forest and XGBoost showed superior performance due to their ability to capture complex patterns. Decision Tree showed slightly lower accuracy due to overfitting.

International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 13 Issue: 04 | Apr 2026 www.irjet.net p-ISSN: 2395-0072

```
from sklearn.metrics import accuracy_score

models = {
    "Logistic": model_lr,
    "Decision Tree": model_dt,
    "Random Forest": model_rf,
    "XGBoost": model_xgb,
    "Gradient Boosting": model_gb
}

for name, model in models.items():
    y_pred = model.predict(X_test)
    print(name, "Accuracy:", accuracy_score(y_test, y_pred))

Logistic Accuracy: 0.9735
Decision Tree Accuracy: 0.982
Random Forest Accuracy: 0.991
XGBoost Accuracy: 0.99
Gradient Boosting Accuracy: 0.99
```

Fig 8). Accuracy comparison

3) Confusion Matrix Analysis

The confusion matrix shows the classification performance of the model.

Analysis:

- High true positives and true negatives
- Very low misclassification
- Strong prediction capability

4) ROC Curve Analysis

The ROC curve demonstrates the classification capability of the model.

Analysis:

- AUC value close to 1
- Excellent class separation
- High model reliability

5) Feature Importance Analysis

Feature importance analysis identifies key parameters influencing predictions.

Key Observations:

- Power (Torque × Speed) is the most significant feature
- Temperature difference also plays an important role

6) Risk Scoring Results

The risk scoring system provides a probability-based assessment of machine failure.

Categories:

- Low Risk
- Medium Risk
- High Risk

Analysis:

Risk scoring enhances interpretability and helps prioritize maintenance actions.

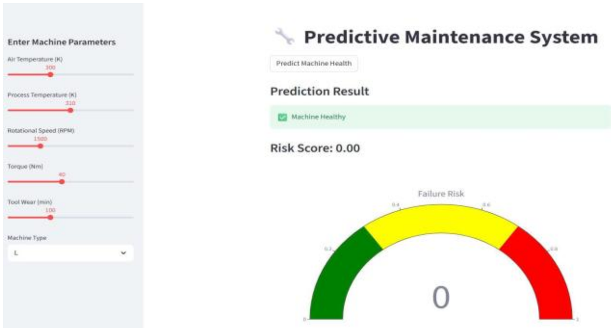


Fig 9). Risk gauge

7) Application Results

The Streamlit application successfully simulates real-time predictive maintenance.

Observed Features:

- Real-time input handling
- Instant prediction output
- Risk visualization
- Maintenance suggestions

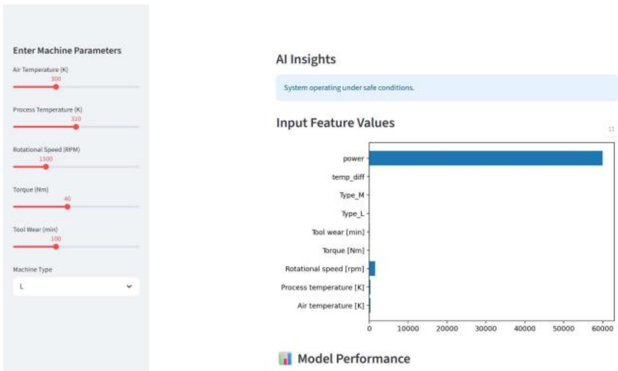


Fig 10). App UI

8) Discussion

The results indicate that machine learning models can effectively predict machine failures with high accuracy. However, the high accuracy is partly due to the structured dataset used in this study.

In real world industrial environments, challenges such as noisy sensor data, missing values, and varying operating conditions may reduce performance.

The integration of risk scoring and application interface improves usability and bridges the gap between theoretical models and practical implementation.

9) Summary

The proposed system demonstrates strong predictive performance and practical applicability. The combination of machine learning, feature engineering, and real-time interface makes it a robust predictive maintenance solution.

Conclusion

This paper presents a machine learning based predictive maintenance system designed to predict machine failures and assist in maintenance decision-making. The system utilizes the AI4I 2020 dataset and implements multiple classification models, including Logistic Regression, Decision Tree, Random Forest, Gradient Boosting, and XGBoost. Feature engineering techniques such as temperature difference and power estimation were applied to enhance model performance.

Experimental results show that ensemble models, particularly Random Forest and XGBoost, achieved superior accuracy and stability compared to other models. In addition to prediction, a risk scoring mechanism was introduced to classify machine conditions into low, medium, and high-risk levels, providing more informative insights than a simple binary output.

Furthermore, a user-friendly web application was developed using Streamlit to demonstrate real-time prediction and visualization of machine health. The integration of predictive models with an interactive interface improves the practical applicability of the system.

Although the model achieved high accuracy on the given dataset, it is important to note that real-world industrial environments may present additional challenges such as noisy data and varying operating conditions. Overall, the proposed system demonstrates the effective use of machine learning in predictive maintenance and provides a foundation for future real-time industrial applications.

Acknowledgment

The authors would like to express their sincere gratitude to the faculty and project supervisor for their continuous guidance and support throughout the development of this project. We also thank our institution for providing the necessary resources to complete this work successfully.

References

[1] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," Proc. KDD, 2016.

[2] L. Breiman, "Random Forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.

[3] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," JMLR, 2011.

[4] AI4I 2020 Predictive Maintenance Dataset, UCI Repository.

[5] J.Lee, H. Davari, J. Singh, and V. Pandhare, "Industrial AI: Applications with Sustainable Performance," IEEE Access, vol. 6, pp. 56415–56425, 2018.

[6] S. Zhang, S. Ding, and L. Xie, "A Review of Machine Learning-Based Predictive Maintenance," IEEE Access, vol. 7, pp. 162837–162852, 2019.

[7] D. B. Rawat et al., "Cyber-Physical Systems for Predictive Maintenance," IEEE Communications, 2018.

[8] F. Chollet, "Deep Learning with Python," Manning Publications, 2018.

[9] A. Carvalho, R. Soares, R. Vita, R. Francisco, J. Basto, and S. Alcalá, "A Systematic Literature Review of Machine Learning Methods Applied to Predictive Maintenance," Computers & Industrial Engineering, vol. 137, 2019.