Early detection of system failure using machine learning techniques

Conference Paper · September 2023

4 authors:

Hasibul Islam
Hajee Mohammad Danesh Science and Technology University

Md Shahzamal
Macquarie University

Md. Dulal Haque
Hajee Mohammad Danesh Science and Technology University

Md. Selim Hossain
Hajee Mohammad Danesh Science and Technology University

LOC Page

Contents

CHAPTER 1

Early detection of system failure using machine learning techniques

Hasibul Islam

Department of Electronics and Communication Engineering (ECE), Hajee Mohammad Danesh Science and Technology University, Dinajpur, Bangladesh.

Md Shahzamal

Honorary Postdoctoral Fellow, Department of Computing, Macquarie University, Australia.

Md. Dulal Haque

Department of Electronics and Communication Engineering (ECE), Hajee Mohammad Danesh Science and Technology University, Dinajpur, Bangladesh.
Postdoctoral Researcher, Shanghai Institute of Technical Physics, Chinese Academy of Science, China.

Md. Selim Hossain

Department of Electronics and Communication Engineering (ECE), Hajee Mohammad Danesh Science and Technology University, Dinajpur, Bangladesh.

CONTENTS

Early fault detection has been a key research area of Industry 4.0 which can significantly improve productivity and reduce financial damage for many industries. This paper investigated machine learning techniques that can proactively identify future system failures up to one week ahead and help the engineering team to schedule preventive maintenance works in advance. We leveraged historical operational and

health data collected from various sensors to build a future failure day prediction model. In this study, decision trees, random forests, and XGBoost classifiers were chosen as the candidate models to predict the numeric class that indicates the number of days remaining before the system can fail. However, class 7 indicates 7 days or more days until the next failure. Initially, baseline models were trained with a mean and standard deviation of numerical variables. All three models provided high accuracy up to 90% for the day 1 and day 2 predictions, but only 44% accuracy for day 7. Two weaknesses were identified in the baseline models. High numbers of day 7 classes were predicted as the near day classes (3 or 4 days remaining until the next failure) which will prompt unnecessary maintenance tasks. Secondly, many near-day classes (3 or 4 days) were predicted as day 7 classes which will prompt sudden maintenance tasks as these failure events will be identified 1-2 days in advance by the accurate day 1 and day 2 class predictions. Our research identified several features based on the available data and built XGB classifiers with 83% accuracy for the day 7 class and improved 5 - 30% prediction accuracy over the other classes compared to baseline models. The proposed model reduced both over and sudden maintenance tasks recommended by the baseline models.

1.1 INTRODUCTION

The fourth Industrial Revolution brought a huge change in modern industry by implementing advanced technologies such as Artificial Intelligence (AI), Machine Learning (ML), the Internet of Things (IoT), and Robotics. With these changes, the complexity of the industrial systems also increased. Many systems are connected to operate on a single task, or one system may depend on another system's output [1]. Due to the continuous operation of those systems, the abrasions and fatigue of its major components happen. In case of failure or faults in these systems, it costs a lot in productivity. Using modern technologies, we can build an efficient and low-cost predictive maintenance system [2]. The process of analyzing and studying historical data to detect faults before it takes place and performing appropriate operation according to the fault is referred to as predictive maintenance. Before the explosion of current technology, the term industrial maintenance refers to the service done by technicians or experts to manage or check the condition of the machinery. But due to advancements in sensors and Internet of Things (IoT) technology, the importance of predictive maintenance has increased over the past decades. Most of the systems exhibit some abnormalities, such as increased power consumption, overheating, abnormal vibrations, and other indicators before ultimate failure. By collecting such behavior using various sensors or monitoring devices we can predict its next failure before it happens. Detecting early signs of failure makes maintenance work more effective. Previous time series data about the past operational state is another important parameter to make early prediction more accurate [3, 4, 5, 6].

ML-based predictive maintenance systems have become so popular because the availability of computing power makes machine learning processes faster and more efficient [7]. Developing a predictive maintenance system faces challenges of real-time responsiveness, noisy data, limited high-quality data availability, variable data distribution, and selecting suitable predictive models. The main benefits include minimizing downtime, and associated costs, and extending equipment life.

There are several research areas in predictive maintenance such as early fault prediction, remaining useful life (RUL) prediction, fault detection and diagnosis, condition monitoring, sensor data analysis, and predictive maintenance itself differ depending on the application area. This paper investigated techniques for predicting failure events of a machinery system 7 days in advance. It's Similar to an RUL prediction method but in a small duration. How long a system will remain operational or when it might fail is considered an RUL prediction. In a traditional maintenance system, maintenance is done after a failure occurs or sometimes ends up over-maintenance. Early prediction of a system fault can solve this problem by predicting the failure event in advance.

This study uses a time series dataset, which includes five distinct data sources, incorporating both numeric and categorical values, collectively representing records from 100 different machines. As previously stated, this problem is approached in two distinct ways, in the baseline method, the XGBoost classifier predicts with 94% accuracy when the system is very close to failure, but its effectiveness diminishes when forecasting failures occurring four or more days in advance. However, in the proposed method, after adding generated attributes and features the performance is improved by about 20% in predicting the more days away from failure.

1.2 RELATED WORK

The importance of effective and sustainable early fault detection systems is increased with the advancement in industrial systems. There are a lot of studies or models invented for early fault detection in predictive maintenance over the past years. Most published works' strategies can be loosely divided into two types [8]: data-driven and model driven. The model-based approaches demand a priori physical and domain knowledge of the system. The well-established model-based methodologies might be successfully used after the creation of the process model based on fundamental principles [9]. Such methods are useful for detecting early faults and effectively describing the fault type, but it is always a tough task to gather domain knowledge from an observed system. On the other hand, data-driven strategies offer an effective alternate approach by analyzing and studying the necessary processed data to be directly pulled from an enormous amount of the accumulated data [10, 11, 12]. Appropriate data is the most important parameter for building predictive models. With the advancement in sensor technology, data transmission speed, and IOT devices monitoring data gradually accumulates and presents the characteristics of big data, which provides a development opportunity for data-driven early prediction methods. Most data-driven early prediction methods use deep learning (DL), statistical models, or machine learning (ML) algorithms. Deep learning-based early prediction methods used Artificial Neural Networks (ANN) [13], Convolutional Neural Networks (CNN) [14]. An unsupervised machine learning algorithm is proposed for early fault detection in predictive maintenance in [15]. For analyzing and describing large monitoring data, multivariate statistical process monitoring methods are used. Which employ the input and output information of the process, and are popular for the purpose of fault diagnosis, mostly principal component analysis (PCA) [16, 17, 18] and partial least squares (PLS) [19].

PCA is the most used algorithm for reducing the dimensionality of multivariate monitoring data. Many hybrid approaches used PCA for fault detection such as using Kernel PCA [20]. In many studies [21, 22, 23] different feature extraction methods are proposed. The extracted features are provided to classifiers as inputs. In [23], Li et al presented a neural-network-based motor-bearing fault diagnosis method using time and frequency-based features, which achieved average detection accuracy between 88.75% and 96.25% for different numbers of hidden neurons.

In most of the studies early fault prediction is done a day before the actual failure. It requires major maintenance or replacement of major components. Where our proposed method made early fault predictions a week in advance. Which makes the early fault detection system more effective.

1.3 METHODOLOGY

This paper we propose an early prediction method of system failure. The primary aim of this study is to enhance system failure prediction by extending the lead time from one day to one week prior to the occurrence of failure. The proposed research methodology encompasses three fundamental steps: data preprocessing, feature engineering, and evaluating prediction accuracy. Fig. 1.1 shows the complete flowchart of the proposed method.

1.3.1 Data Sources

In this study, we have used the Microsoft Azure AI Predictive Maintenance dataset which contains five different data sources about machine health conditions and usage. The data sources are-

- **Telemetry Data:** It is a time series data consisting of voltage, rotation, pressure, and vibration value measured from 100 different machines in real time averaged over every hour collected during 2015. More specifically the voltage value represents the amount of voltage a machine is operating at, rotation speed reflects the instant rotation speed of a machine, pressure value represents the level of pressure a machine goes through, vibration value is the generated vibration amount by a machine. Those values represent the machine's health condition. This data source has a major contribution in predicting machine failure. Table 1.1 shows the basic statistical insight of these attributes-

  It reveals that the maximum variance between minimum and maximum value of voltage is 157.79, rotation is 556.59, pressure is 134.72 and vibration is 61.92. So, the rotation value contains maximum variance which would reflect the difference between failure state and operational state more clearly. Visual representation of the trend in each attribute helps to clearly describe each attribute more

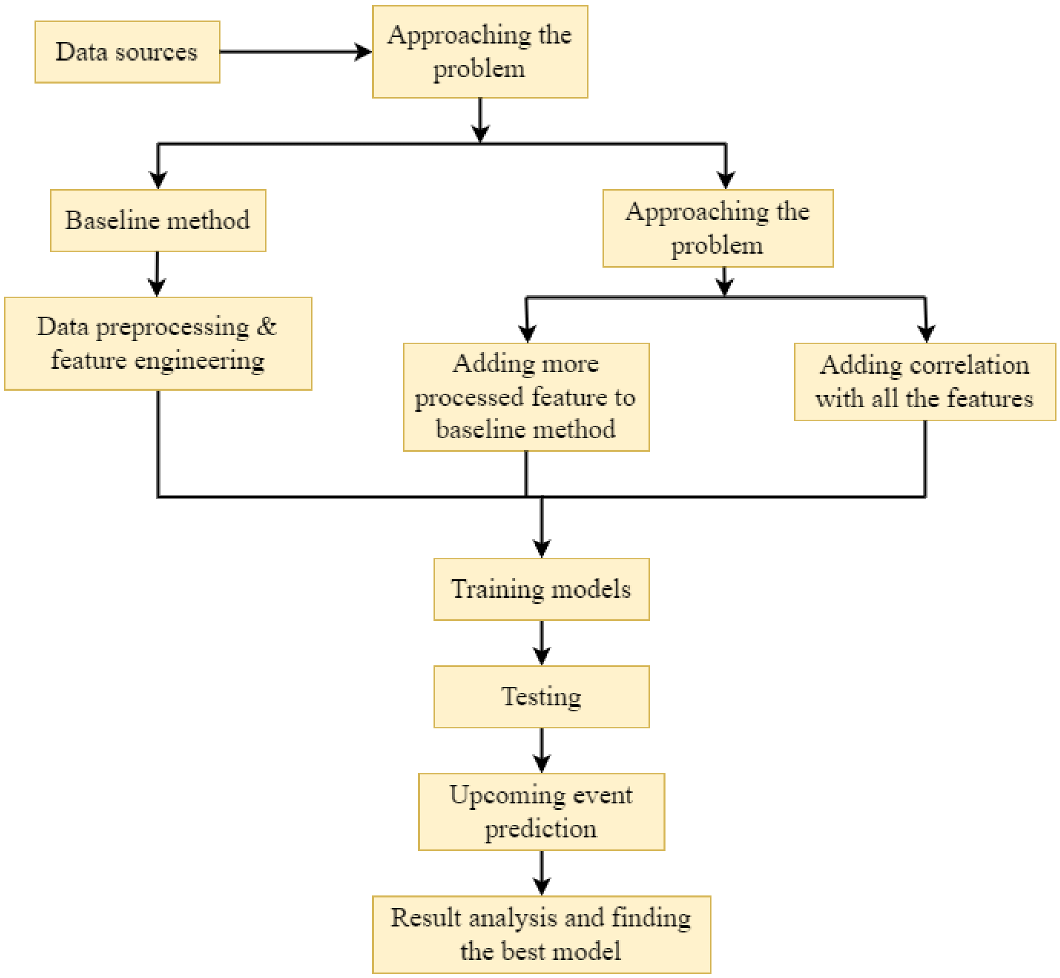

Figure 1.1 Methodology for predicting failure events a week in advance.

Table 1.1 Statistical measure of telemetry numeric attributes.

Statistical Info.	**Voltage**	**Rotation**	**Pressure**	**Vibration**
Mean	170.77	446.60	100.85	40.38
Std	15.50	52.67	11.04	5.37
Min	97.33	138.43	51.23	14.87
Max	255.12	695.02	185.95	76.79

clearly. Thus, in one month (01/01/2015 to 01/02/2015) time duration the trend in each variable for one machine is shown in Fig. 1.2
It shows that the voltage and vibration have remarkable variation over time where the rotation and pressure values contain sharp changes. Which might represent a failure event.

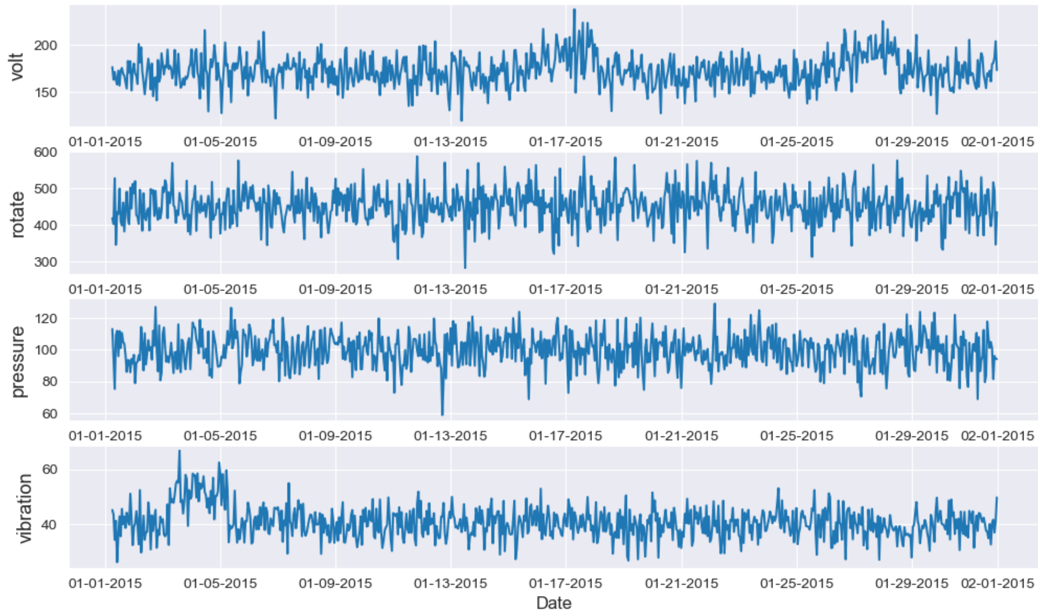

Figure 1.2 Trend in telemetry attributes (voltage, rotation, pressure, and vibration).

- **Failure History:** It has records of changed components due to a major failure or breakdown of a machine. There are four different types of components that need to change during the failure event of a machine. Fig. 1.3 represents the percentage of each type of component replacement. The total number of failure events during the observation period is 761.

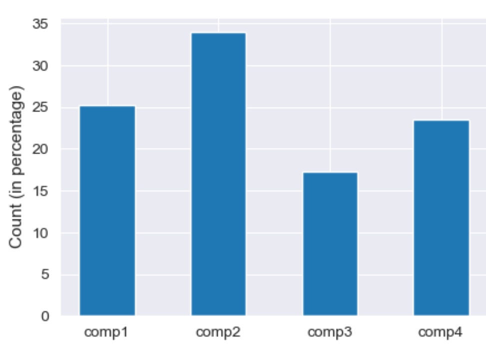

Figure 1.3 Failed components

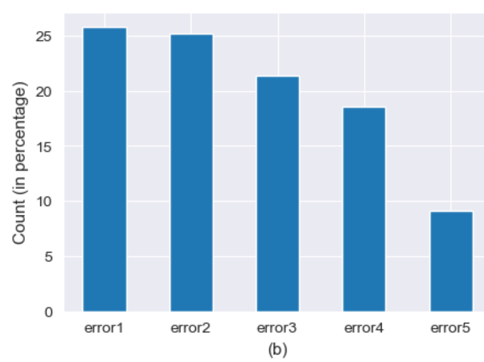

Figure 1.4 Types of error

- **Machine Features:** It consists of the model and age value of each different machine. There are four different models of machine used. Year in service is considered as the age of a machine. Fig. 1.5 shows each type of machine with their age-
- **Maintenance History:** The regular maintenance and required maintenance

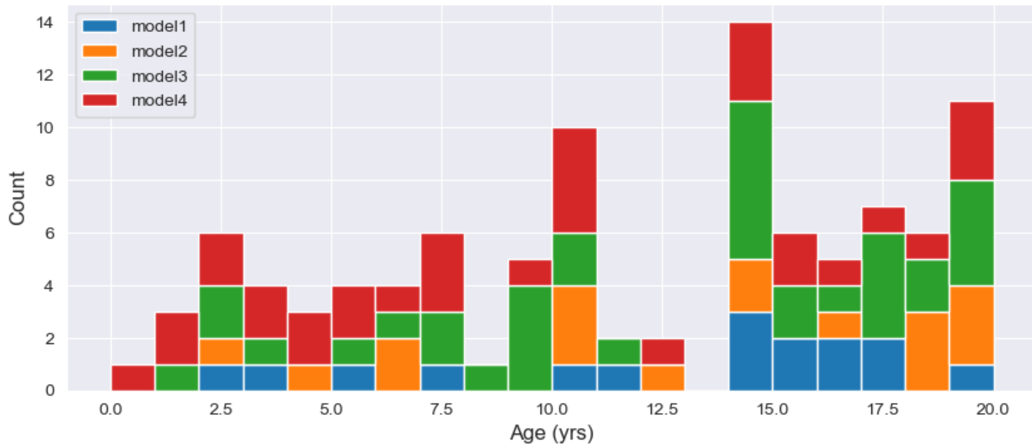

Figure 1.5 Different types of models and ages of the machines.

due to failure are recorded including the changed component. Thus, it includes redundant information corresponding to failure history data.

- **Error History:** It provides a non-breaking error history while the machine is still operational. Error type can be considered as a categorical value consisting of five different kinds of error. The contribution of each type of error in a total of 3919 error events is shown in Fig. 1.4.

1.3.2 Data Prepossessing & Feature Engineering

Processing raw data is a crucial step to make the data suitable for ML algorithms. In the dataset employed for this study, all data sources except the machine features are time series data recorded in real-time averaging over every hour. Timestamp labels in time series data aid attribute extraction over a fixed period.

The failure status is processed for early prediction and added with other processed attributes according to the belonging machine id and the occurring hour on the specific date. To forecast faults occurring a week in advance, we employ a classification framework that involves categorizing failure statuses into 7 distinct classes, each corresponding to the number of days preceding the actual failure according to Algorithm 1. The contribution of each target class in total is shown in Fig. 1.6.

Algorithm 1. Target Class Preparation Algorithm for the proposed method.

```
while (!df. end) then
    if (current_day != failed_day) then
        day_to_fail = failed_day - current_day
        if (day_to_fail == 7) then
            day_to_fail = 7
```

```
    end
    else
        day_to_fail = 7
    failed_day = next_failed_day
    end
end
```

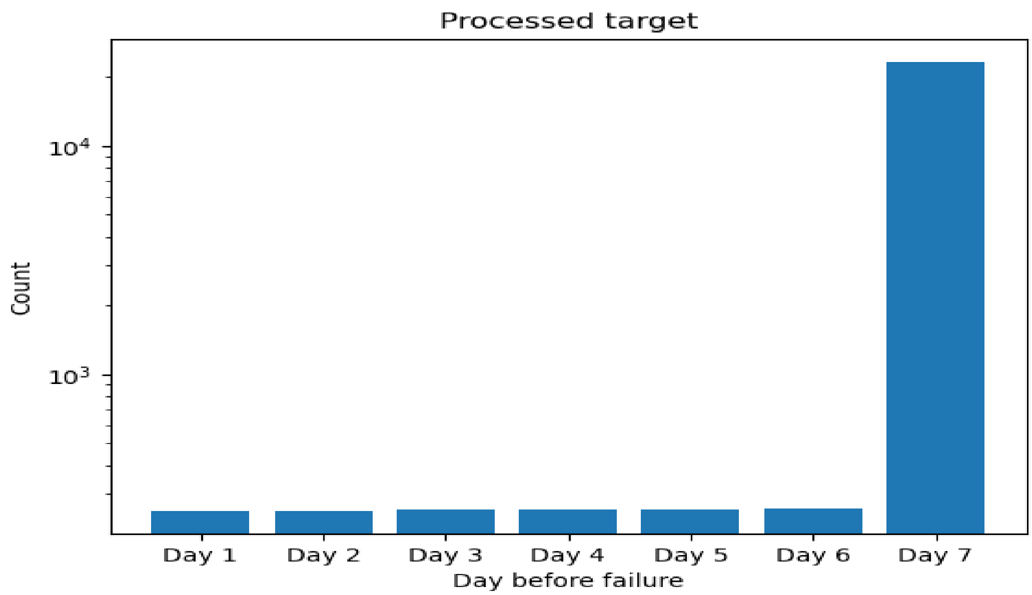

Figure 1.6 Processed target variable which represents the number of days before failure.

The Telemetry data source is a time series data that provides major attributes regarding machine health information which are important for predicting failure events. The timestamp label is floored to date so that it includes 24 samples from 24 hours and then mean and standard deviation is taken over a day for all telemetry attributes according to equation 1.1 and 1.2.

$$\text{Mean } \underline{x}(24 \text{ hours }) = \frac{\sum_{i=1}^{24} x_i}{24} \tag{1.1}$$

$$\text{Std}\, \sigma(24 \text{ hours }) = \sqrt{\frac{\sum_{i=1}^{24} \cdot (x_i - \underline{x})^2}{24}} \tag{1.2}$$

The variability of processed feature data across different manipulated target classes is illustrated in Fig. 1.7 through a box plot.

It indicates that the variation in most of the features with respect to the processed target variables is ambiguous except for the rotation features and some variation in vibration standard deviation.

The error event from the error history data source needs extra attention to make

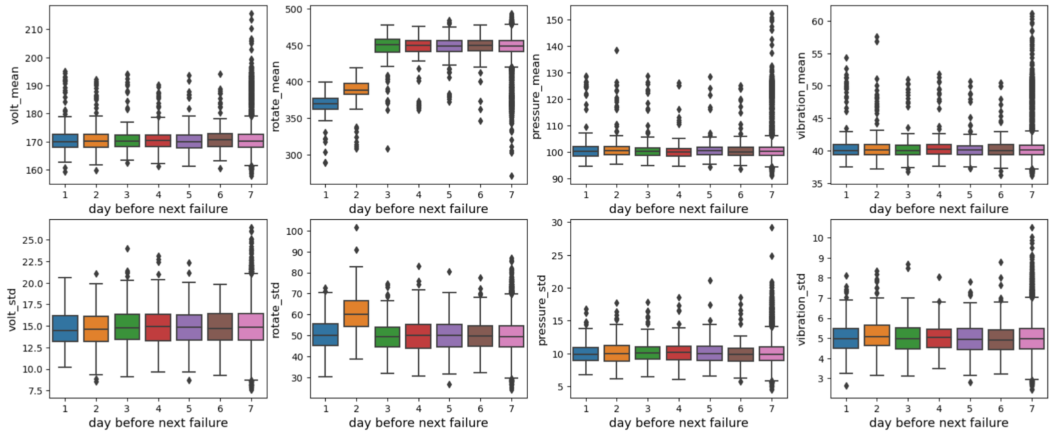

Figure 1.7 Variation in feature data with respect to the processed target class.

it more useful in predicting major failure. As error value associated with separate machines with respect to their recording time. Hence days since the previous error might be more insightful for the next failure event. The days since the previous error are also calculated by iterating over each error status comparable to Algorithm 1 but this time, we go forward from an error event and add a day to the next sample, and each error status is treated as identical. Years in service might be another important feature that makes a machine error prone. The model or origin of a machine might also be considered a parameter for repeated failure events. As the machine learning algorithm handles numerical value better than string objects, the model of a machine is encoded in numeric value. It demonstrates that models 2 and 3 have a major contribution in the processed target class.

Correlations between major attributes over a period reflect the relation between different attributes in data which leads to the failure of a machine but the period over which the correlation is taken should be chosen carefully. Unless it will just add dimension to data that help the model to predict uncertainty. Here we consider the correlation between one processed telemetry attribute with all other telemetry attributes on the next day. The variability of correlation feature across all target classes is illustrated in Fig. 1.8 through a box plot.

1.3.3 Candidate Model

As previously pointed out, the problem is considered as a classification problem, hence the most popular classification machine learning algorithms including Decision Tree (DT), Random Forest Classifier (RF), and XGBoost Classifier are used for their compatibility with the data source and other use cases.

Decision Tree (DT) is a popular supervised machine learning algorithm used for both classification and regression problems but mostly it is preferred for classification problems. Industrial data might encounter a small variation in the data, DT acts as unstable in this case. It also has the risk of overfitting. A random forest (RF) algorithm is used to back up those drawbacks of the Decision tree.

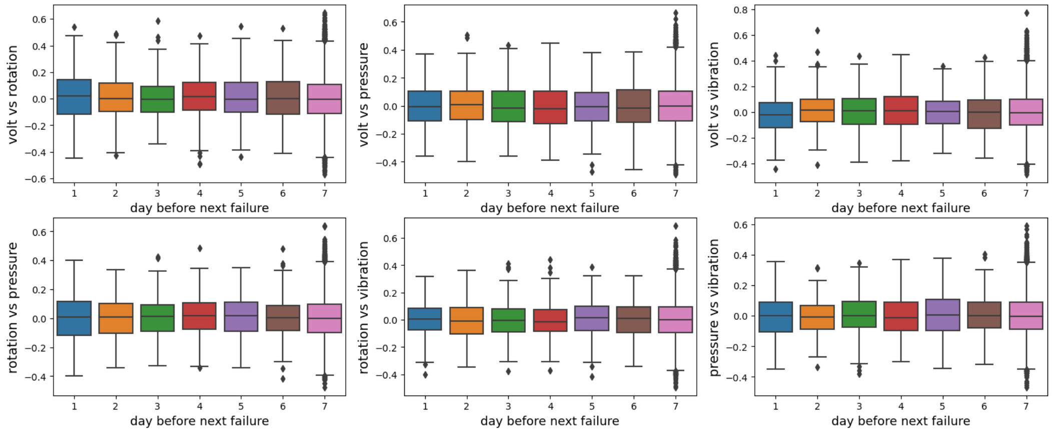

Figure 1.8 Variance in correlation between numerical attributes over a day.

Random forest (RF) offers advantages over single decision trees, including reduced overfitting, robustness to noise and outliers, and improved accuracy. However, RF has limitations regarding imbalanced datasets, and finding optimal hyperparameters requires extensive experimentation. To recover those issues of the Random Forest algorithm, the XGBoost classifier (XGB) is used.

XGBoost classifier can handle skewed data and class imbalance better than the Random Forest algorithm by using appropriate weighting and sampling techniques. It also gives more improved accuracy. In XGBoost, when the model fails to predict the anomaly for the first time, it gives more preferences and weightage to it in the upcoming iterations thereby increasing its ability to predict the class with low participation.

1.4 RESULTS AND EVALUATION

The performance accuracy of the baseline method and the proposed methods in predicting machine failure in advance is presented by comparing the score of used algorithms. The algorithms aren't used in their default state to avoid overfitting and underfitting, instead the parameters are chosen using k-fold cross validation. The optimal values for the maximum depth parameter, applicable to all models, as well as the number of estimators for the Random Forest Classifier and the learning rate for the XGB Classifier, were determined through a systematic experimentation process. Moreover, considering the imbalance in target classes, both oversampling and undersampling techniques are employed and during splitting the dataset for model training and testing a stratified process is used to balance the failure event samples in the train and test dataset. The main challenge of working with an imbalanced dataset is that it produces poor performance on the minority class, to address that minority class data can be synthesized from the existing samples instead of deleting or duplicating samples. Here, we apply the synthetic minority oversampling technique (SMOTE). In addition, a Random sampler for the majority class is utilized.

In the baseline method, only the mean and standard deviation values of all the telemetry variables are used to predict the failure event. The bar plot in Fig. **??** shows the prediction accuracy scores of different algorithms across the target classes representing the seven days. The findings indicate that the prediction accuracy is notably higher when the machine failures are soon. The accuracy of DT is about 89%, RF is about 88% and XGB is 95% when the model predicts failures one day ahead. The accuracy remains almost the same for predicting failures two days ahead. However, the prediction accuracy for the subsequent day's decreases.

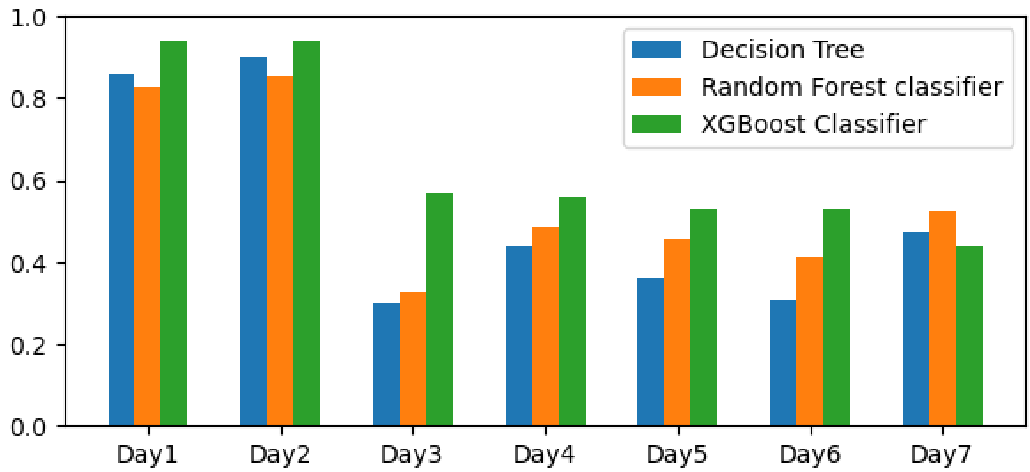

Figure 1.9 Accuracy scores of the baseline method across distinct target classes.

The prediction accuracy for 3 to 7 days fluctuates for both DT and RF classifiers while accuracy was almost similar for 3 to 6 days for XGB classifier. The lowest prediction accuracy for XGB was found for class day 7. This is because class day 7 has many training samples that assume high variations which are not captured by the XGB. As a result, the trained prediction model is less accurate. Interestingly, the prediction accuracy of DT and RF was high for class day 7 compared to XGB. The reason might be these algorithms capture the relationship with these large numbers of samples in class on day 7. In summary, DT and RF provide the lowest accuracy score respectively 30% and 32% at day 3, and XGBClassifier provides the lowest score 44% at day 7.

The number of true positives, true negatives, false positives, and false negatives for each class is shown in the confusion matrix of Fig. 1.10 for all models. This shows that all models have a nearly equivalent number of true positive predictions on day 1 and day 2. Rarely true class day 1 and day 2 were predicted as the higher class. This is good for failure detection if the model is accurate when the machine is about to fail. Only one day 1 class is predicted as a day 4 class in XGB classifier. However, some other classes except day 1 and day 2 were predicted as class day 1 and day 2. Predicting higher days class as lower days class is not dangerous as the preventive maintenance will be carried out in advance. However, XGB shows low numbers in the miss classification of far days classes into near days classes. For the classes from day 3 to day 4, near days classes are often predicted as far-day classes. This means the model underestimated the required predictive maintenance. The DT performs better in this regard with a low number of near-day classes as day 7 classes. The proposed model presented below shows the improvement in class predictions.

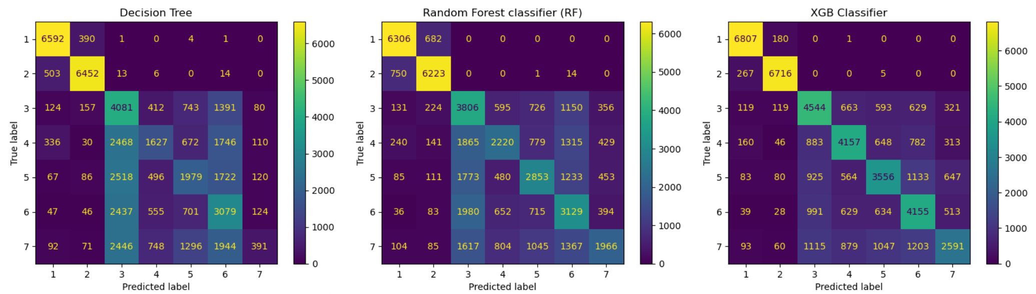

Figure 1.10 Confusion Matrix for all trained models (Predicted Class vs Actual Class).

In the proposed method, an initial enhancement involves augmenting the decision attributes by incorporating additional features such as age, model, and days since the previous failure with the existing baseline method features, which improved the overall prediction scores by 5-10%.

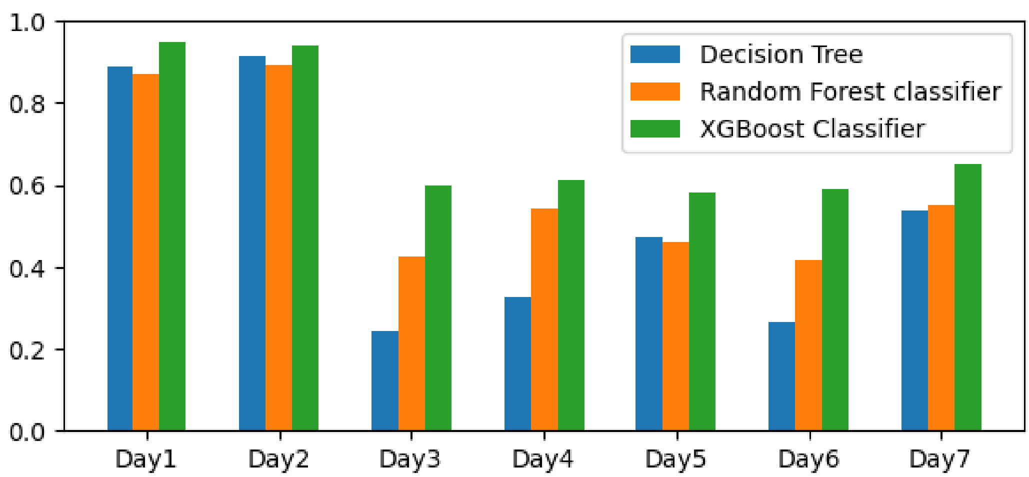

Figure 1.11 Accuracy scores of the baseline method across distinct target classes.

The results on Fig. 1.11 reveal that the DT algorithm demonstrates minimal improvement in predicting day 1 and day 2 by 3.53% and 2.25%, but experiences a decline in performance for day 3, day 4, and day 6. However, RF and XGB got some extra strength in predicting the far days because of newly added relevant features. RF shows 10% improvement on day 3 and 6% on day 4. While the XGB attained almost a 20% improvement on day 7 and 6% on day 6.

The confusion matrix is shown in Fig. 1.12 for all models. Adding these features, DT and RF miss classification got worse. It is seen that a lot of near-day classes are predicted as far-day classes. This means it will prompt a delayed predictive maintenance schedule and the machine may fail before maintenance. Similar results are found for XGB. However, it provided a much lower miss classification of near-day classes to far-day classes. Our correlation-based features will improve such scenarios.

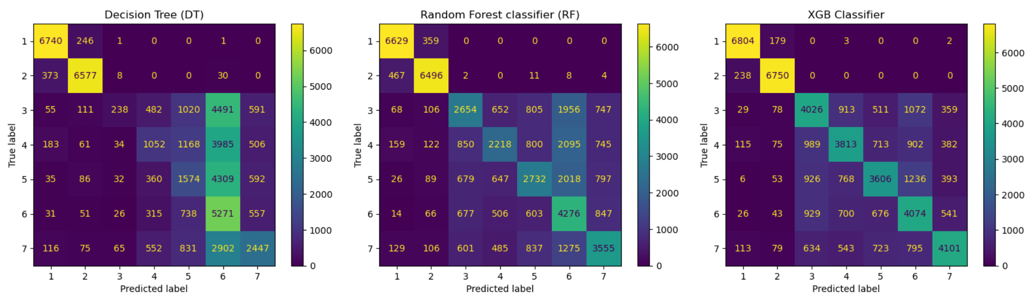

Figure 1.12 Feature extended method confusion matrix (predicted class vs actual class).

To improve the model performance in the second approach, the processed correlation values are added with all the other existing attributes and the parameter of our used algorithm remains the same, which leads to the gain of the overall prediction score of RF classifier and XGB classifier by another 5-12%.

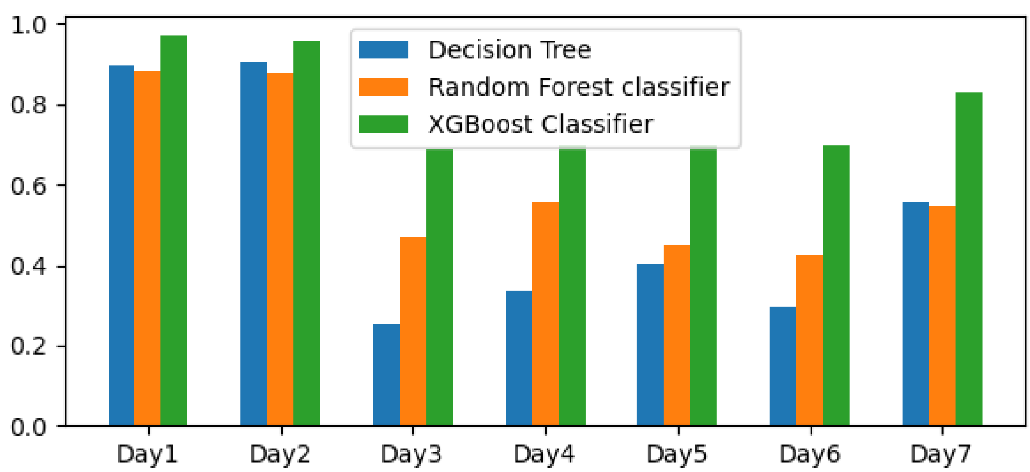

Figure 1.13 Correlation-based method accuracy score in different target classes.

The major accuracy gain is reflected in the XGBoost classifier in Fig. 1.13. It increases the ability to predict far days accurately. The lowest precision 69% is shown on day 3 and from day 4 to day 6 the score is 70% which represents a notable improvement of 15-17% compared to the baseline method. The confusion matrix for the approach is shown at Fig. 1.14.

The confusion matrix shows that the XGB classifier is now more capable of correctly classifying most instances. Improvement in accuracy score is gained because of enough appropriate number of features. In this method, the miss classification of near-day classes to far-day classes in both DT and RF remains worse. Therefore, DT and RF did not improve significantly by the new features. On the other hand, the

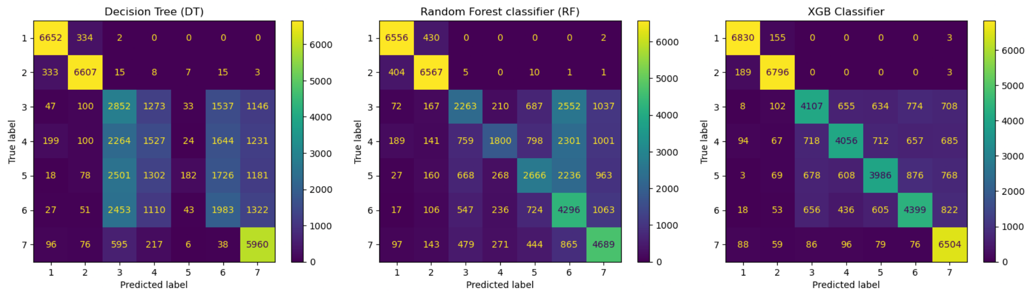

Figure 1.14 Correlation-based method confusion matrix (predicted class vs actual class).

miss classification of near-day classes to far-day classes remain the same or lower in the XGB, and prediction accuracy is improved by 5 - 30% depending on the classes. However, XGB achieved 83% accuracy for class day 7. According to the confusion matrix, very few 7-day classes are classified as near-day classes. This would prevent unnecessary predictive maintenance. It means the model can predict with high accuracy which machine can fail over 6 days. About 17% of day classes are predicted as class day 7. But they will be corrected by the day 1 and day 2 classifiers when we are approaching the failure days. It means there might be some sudden schedule of predictive maintenance which was classified by the day 7 class. However, about 90% of predictive tasks are in the day 7 classes. So, reduction in the unnecessary schedule of predictive maintenance is the key improvement of our research. The performance evaluation results for all methods are presented in Table 1.2.

Table 1.2 Performance score of all methods across different classes.

Method	ML Models	Day1	Day2	Day3	Day4	Day5	Day6	Day7
Baseline method	DT	0.85	0.89	0.30	0.43	0.36	0.31	0.47
	RF	0.82	0.85	0.32	0.48	0.45	0.41	0.52
	XGB	0.94	0.94	0.57	0.56	0.53	0.53	0.44
Extended features method	DT	0.88	0.91	0.24	0.32	0.47	0.26	0.53
	RF	0.87	0.89	0.42	0.54	0.46	0.41	0.54
	XGB	0.95	0.94	0.60	0.61	0.58	0.59	0.65
Correlation features method	DT	0.89	0.90	0.25	0.33	0.40	0.29	0.55
	RF	0.88	0.88	0.46	0.55	0.45	0.42	0.54
	XGB	0.97	0.96	0.69	0.70	0.70	0.70	0.83

1.5 CONCLUSION AND FUTURE WORK

In this paper, we built a machine-learning model to detect the system failure a week before it happened. This approach not only mitigates the risk of sudden system failure but also significantly reduces system downtime. It plays an important role in the overall productivity of a system. In most studies, early fault prediction has typically been performed one day prior to the occurrence of actual failure. In contrast, our proposed method enables early fault predictions to be made a week in advance, resulting in a significantly enhanced effectiveness of the early fault detection system. This distinguishing feature renders our method unique and autonomous compared to other existing approaches, enabling it to operate independently and demonstrate efficacy in achieving its objectives. In contrast, rather than predicting a large number as a remaining useful life, it could be an effective alternative approach to classify the target variable into seven distinct classes representing the days of the week.

Early prediction might also be done more than a week before. It will increase the number of target variables if the task is considered a classification problem. Which will become more challenging. As the number of failure event samples is always less than the operational sample in a real system that's why the collected dataset always be imbalanced. XGB Classifier performed better than the other two models in our proposed method. It gets the advantage of an imbalanced dataset over the other two models. As it is identified that class day 7 is predicted with high accuracy with the XGB classifier, it would be interesting for future research directions to add another classifier on top of this seven-day classifier to predict whether it can fail within seven days or outside seven days. Then, we can build a classifier to predict which day it can fail within seven days.

Bibliography

[1] H. Lasi, P. Fettke, H.-G. Kemper, T. Feld, and M. Hoffmann, "Industry 4.0. business & information systems engineering, 6 (4), 239-242," 2014.

[2] C. Biegler, A. Steinwender, A. Sala, W. Sihn, and V. Rocchi, "Adoption of factory of the future technologies," in *2018 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC)*, pp. 1–8, IEEE, 2018.

[3] S. D. Fassois and J. S. Sakellariou, "Time-series methods for fault detection and identification in vibrating structures," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 365, no. 1851, pp. 411–448, 2007.

[4] P. Malhotra, L. Vig, G. Shroff, P. Agarwal, *et al.*, "Long short term memory networks for anomaly detection in time series.," in *Esann*, vol. 2015, p. 89, 2015.

[5] B. Chen, Y. Liu, C. Zhang, and Z. Wang, "Time series data for equipment reliability analysis with deep learning," *IEEE Access*, vol. 8, pp. 105484–105493, 2020.

[6] K. S. Kiangala and Z. Wang, "An effective predictive maintenance framework for conveyor motors using dual time-series imaging and convolutional neural network in an industry 4.0 environment," *Ieee Access*, vol. 8, pp. 121033–121049, 2020.

[7] T. Hwang, "Computational power and the social impact of artificial intelligence," *arXiv preprint arXiv:1803.08971*, 2018.

[8] S. Yin, S. X. Ding, X. Xie, and H. Luo, "A review on basic data-driven approaches for industrial process monitoring," *IEEE Transactions on Industrial electronics*, vol. 61, no. 11, pp. 6418–6428, 2014.

[9] I. Hwang, S. Kim, Y. Kim, and C. E. Seah, "A survey of fault detection, isolation, and reconfiguration methods," *IEEE transactions on control systems technology*, vol. 18, no. 3, pp. 636–653, 2009.

[10] A. Kusiak and A. Verma, "A data-driven approach for monitoring blade pitch faults in wind turbines," *IEEE Transactions on Sustainable Energy*, vol. 2, no. 1, pp. 87–96, 2010.

[11] M. Baptista, S. Sankararaman, I. P. de Medeiros, C. Nascimento Jr, H. Prendinger, and E. M. Henriques, "Forecasting fault events for predictive maintenance using data-driven techniques and arma modeling," *Computers & Industrial Engineering*, vol. 115, pp. 41–53, 2018.

[12] L. Guo, R. Li, and B. Jiang, "A data-driven long time-series electrical line trip fault prediction method using an improved stacked-informer network," *Sensors*, vol. 21, no. 13, p. 4466, 2021.

[13] P. Bangalore and L. B. Tjernberg, "An artificial neural network approach for early fault detection of gearbox bearings," *IEEE Transactions on Smart Grid*, vol. 6, no. 2, pp. 980–987, 2015.

[14] T. Ince, S. Kiranyaz, L. Eren, M. Askar, and M. Gabbouj, "Real-time motor fault detection by 1-d convolutional neural networks," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 11, pp. 7067–7075, 2016.

[15] N. Amruthnath and T. Gupta, "A research study on unsupervised machine learning algorithms for early fault detection in predictive maintenance," in *2018 5th international conference on industrial engineering and applications (ICIEA)*, pp. 355–361, IEEE, 2018.

[16] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and intelligent laboratory systems*, vol. 2, no. 1-3, pp. 37–52, 1987.

[17] I. T. Jolliffe, *Principal component analysis for special types of data*. Springer, 2002.

[18] V. Choqueuse, M. E. H. Benbouzid, Y. Amirat, and S. Turri, "Diagnosis of three-phase electrical machines using multidimensional demodulation techniques," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 4, pp. 2014–2023, 2011.

[19] R. Muradore and P. Fiorini, “A pls-based statistical approach for fault detection and isolation of robotic manipulators,” *IEEE Transactions on Industrial Electronics*, vol. 59, no. 8, pp. 3167–3175, 2011.

[20] J. Yang, Y. Chen, and Z. Sun, “A real-time fault detection and isolation strategy for gas sensor arrays,” in *2017 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, pp. 1–6, IEEE, 2017.

[21] J. Liu, W. Wang, F. Golnaraghi, and K. Liu, “Wavelet spectrum analysis for bearing fault diagnostics,” *Measurement Science and Technology*, vol. 19, no. 1, p. 015105, 2007.

[22] R. Yan, R. X. Gao, and X. Chen, “Wavelets for fault diagnosis of rotary machines: A review with applications,” *Signal processing*, vol. 96, pp. 1–15, 2014.

[23] B. Li, M.-Y. Chow, Y. Tipsuwan, and J. C. Hung, “Neural-network-based motor rolling bearing fault diagnosis,” *IEEE transactions on industrial electronics*, vol. 47, no. 5, pp. 1060–1069, 2000.