Applying XGBoost for Fault Prediction in Industrial Production Line

Chao Chen*, Xu Li, Kai Wang
School of Artificial Intelligence, Guangzhou Huashang University, Guangzhou, Guangdong, China
**Corresponding Author.*

Abstract: In the era of Industry 4.0, the level of intelligence and automation of production lines is crucial for improving production efficiency. This study addresses the issue of fault prediction in industrial production lines by constructing an automatic alarm model using XGBoost and neural network technology to enhance the intelligence of production lines and optimize scheduling. By analyzing the characteristics of fault data and using correlation matrices and time series differencing methods to build feature engineering, the model achieves a precision rate of up to 97.99%, effectively predicting fault trends. Furthermore, the model is applied to actual data to automatically alarm faults and statistically analyze fault frequency and duration. At the same time, by using correlation analysis and multiple linear regression models, the study calculates production and qualification rates, revealing their relationships with production lines and operators, and presents them in graphical form. The models and methods in this study have practical application value for improving industrial production efficiency.

Keywords: XGBoost; Machine Learning; Fault Prediction; Industrial Automation

1. Introduction

1.1 Research Background

With the rapid development of information technology, industrial production lines are gradually transitioning towards intelligence and automation. The application of intelligent control technology enables automated production lines to automatically complete processes such as item conveyance, material filling, product packaging, and quality inspection, greatly improving production efficiency and product quality while reducing production costs. However, as the scale and complexity of industrial production continue to expand, the challenges faced by automated production lines are becoming increasingly prominent. Traditional automated systems often lack sufficient intelligence and flexibility to effectively deal with changes and exceptions in the production process. Moreover, there is a lack of effective interconnection between the equipment and systems on the production line, leading to inefficient information transfer and collaboration, which affects overall production efficiency and quality management. Therefore, how to further enhance the intelligence level of automated production lines and optimize the coordination and flexibility of the production process has become a key issue that needs to be addressed in the current industrial field [1].

1.2 Literature Review

The integration of machine learning (ML) techniques with industrial processes, particularly in the context of fault prediction and maintenance, has seen significant advancements in recent years. This section provides a comprehensive review of the literature pertaining to the application of ML in production lines and pipeline safety assessment, highlighting the key findings and methodologies employed.

The proliferation of Industry 4.0 has necessitated the adoption of ML for enhancing the efficiency and reliability of production lines. Kang et al. [2] conducted a systematic literature review, identifying the application of ML in various industrial domains, with a focus on quality control and fault diagnosis. Their study underscored the dominance of supervised learning and the frequent use of artificial neural networks (ANN) in addressing production line problems.

The importance of data analytics in production lines has been emphasized by several researchers. For instance, Crespino et al. [3] highlighted the challenges in handling the increasing data volumes in aerospace manufacturing, where real-time predictive analysis can improve output quality by identifying anomalies. Nakazawa and Kulkarni [4] utilized deep convolutional encoder-decoder neural networks for detecting wafer map defect anomalies in semiconductor manufacturing, demonstrating the potential of deep learning in defect detection.

Predictive maintenance (PdM) has emerged as a critical area in industrial asset management. Paolanti et al. [5] proposed a machine learning approach for predictive maintenance in industry 4.0, emphasizing the need for condition monitoring and remaining useful life prediction. Yu et al. [6] presented a global manufacturing big data ecosystem for fault detection in predictive maintenance, showcasing the synergy between big data and ML in enhancing maintenance strategies.

The application of ML in predictive maintenance has been further explored by Aydemir and Paynabar [7], who focused on image-based prognostics using deep learning approaches. Similarly, Weber and Reimann [8] introduced a platform to manage machine learning models in Industry 4.0 environments, highlighting the practical implementation of ML in industrial settings.

Pipeline safety, a critical aspect of the oil and gas industry, has also seen the application of ML techniques. Elshaboury et al. [9] developed data-driven models for forecasting failure modes in oil and gas pipelines, using multilayer perceptron (MLP) neural networks, radial basis function (RBF) neural networks, and multinomial logistic (MNL) regression. Their models achieved high accuracy rates, demonstrating the effectiveness of ML in predicting pipeline failures.

Liu et al. [10] proposed an XGBoost algorithm-based model for the safety assessment of pipelines, achieving an accuracy of 98.5% and highlighting the potential of XGBoost in pipeline risk assessment. This study, along with others, suggests that ML can significantly reduce the costs associated with non-destructive examinations (NDE) and engineering assessments (EA) in the pipeline industry.

Pang [11] presents a deep learning-based approach for adaptive fault prediction and maintenance in production lines, addressing limitations in traditional methods. The study introduces a model that incorporates wide convolutional feature extraction, customized gating, and multi-layered progressive extraction modules. It utilizes Wasserstein distance for fault stage division and employs L2 regularization and neuron dropout for optimization, enhancing prediction accuracy and maintenance efficiency. This research contributes to the field by offering a more precise and adaptable strategy for fault prediction, which is crucial for improving production line performance and reducing operational costs.

1.3 Problem Formulation

The research questions in this paper mainly focus on how to enhance the intelligence level of automated production lines and optimize the coordination and flexibility of the production process. Key issues include:

1)How to build an effective fault alarm model to achieve automatic and immediate fault alarms, reducing production interruptions and economic losses.

2)How to optimize personnel allocation through data analysis and model construction to reduce resource waste and improve production efficiency.

3)How to address the lack of effective interconnection between equipment and systems on the production line to improve information transfer and collaboration efficiency.

This paper will explore an automatic alarm model based on XGBoost and neural networks, and how to improve the intelligence level of production lines and optimize the production process through data mining and machine learning technology. The ultimate goal of the research is to provide a feasible method and technical route for the automatic recognition of production line faults and personnel allocation, in order to improve production efficiency and product quality, and reduce production costs.

2. Materials and Methods

2.1 Dataset

The dataset used in this study covers 37 fields, including date, time, production line number,

material push number, material waiting to be grasped number, qualification certificate, unqualified number, fault code, etc. The training set used is the operating data of 10 production lines for one year, approximately 75,000,000 rows, and the test set is the operating data of another 2 production lines for one year (excluding fault information fields), approximately 15,000,000 rows. The dataset is diverse and includes text and missing values. After preprocessing, including text conversion, missing value filling, and feature construction, these data provide a basis for training the XGBoost model to predict fault alarms, analyze equipment fault frequency, and duration.

2.2 XGBoost

XGBoost (Extreme Gradient Boosting) is an ensemble learning algorithm, an optimized version of the Gradient Boosting Machine (GBM). It has shown excellent performance and accuracy in handling large-scale datasets, especially in regression and classification problems [12]. The XGBoost model builds a powerful prediction model by integrating multiple weak learners (usually decision trees). The core idea is to sum the prediction results of multiple weak learners with weights to improve the overall model's predictive ability. [13]

2.2.1 Gradient Boosting Framework

The gradient boosting framework of XGBoost can be represented as:

$$\hat{y}_i = \sum_{k-1}^{K} \gamma_k (f_k(x_i; \theta_k) \quad (1)$$

Where $\hat{y}_i$ is the fault prediction result of the model for the i sample, K is the number of weak learners (decision trees); f_k is the prediction function of the k weak learner, γ_k is the weight of the k weak learner, and θ_k is the parameter of the k weak learner.

2.2.2 Objective Function and Regularization

The objective function of XGBoost not only includes prediction error but also adds regularization terms to prevent overfitting, which is particularly important for the generalization ability of fault prediction models:

$$Obj = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \quad (2)$$

where l is the loss function, and Ω is the regularization term, usually containing L_1 and L_2 regularization:

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \quad (3)$$

where γ is the L_1 regularization coefficient, λ is the L_2 regularization coefficient, T is the number of leaves in the tree, and w_j is the score (i.e., prediction value) of the j leaf node.

2.2.3 Feature Importance Evaluation

In fault prediction, identifying which features have a significant impact on fault occurrence is very valuable. XGBoost provides an intuitive method for feature importance evaluation, identifying important features by analyzing the contribution of features in the model's split points:

$$FeatureImportance = \frac{Gain}{\sum_{all\ features} Gain} \quad (4)$$

where $Gain$ is the sum of gains when the feature splits in all trees.

Applying XGBoost to the problem studied in this paper, the model construction can be divided into the following steps. First, construct extreme gradient boosting trees, where XGBoost selects features and split points that maximize gains at each split. Gain can be represented as:

$$Gain = \frac{1}{2}\left(\frac{\sum_{i \in I_L} g_i - \sum_{i \in I_R} g_i}{H_L + H_R}\right)^2 \quad (5)$$

where I_L and I_R are the sample sets of the left and right child nodes after splitting, respectively, g_i is the gradient of the i sample, and H_L is the second derivative of the H_R sample.

$$P(y = j|x) = \frac{e^{f(x)_j}}{\sum_{k-1}^{K} e^{f(x)_k}} \quad (6)$$

where $f(x)_j$ represents the model's predicted score for class j, and K is the total number of fault types.

Overall, the XGBoost algorithm holds a significant position in the field of production line fault prediction due to its excellent performance and generalization capabilities. By adjusting model parameters, such as the maximum depth of decision trees and the learning rate, the performance of the XGBoost model can be further optimized to suit specific production line environments and fault prediction requirements.

3. Analysis and Results

3.1 Data Processing

Before constructing a production line fault prediction model, it is essential to conduct an in-depth analysis of the production line data to identify and extract key data features. This process involves not only feature engineering for fault data but also plays a crucial role in the accuracy and generalization capability of subsequent models.

Firstly, data preprocessing is carried out: (1) Data cleaning: Remove invalid or incomplete records to ensure the consistency and reliability of the dataset. (2) Format standardization: Convert all data into a unified format for ease of subsequent processing and analysis. (3) Missing value treatment: Interpolate or delete missing data to prevent information loss from biasing the model. (4) Outlier detection: Identify and handle outliers to reduce their adverse effects on model training.

Next, feature engineering is conducted to extract useful information from the data: (1) Feature selection: Identify features most relevant to fault prediction, reducing model complexity and improving predictive efficiency. (2) Feature transformation: Standardize or normalize features to the same scale, enhancing the model's convergence speed and accuracy. (3) Feature construction: Create new features based on existing data to reveal hidden patterns and relationships.

Finally, statistical analysis methods are used to explore the correlations between various faults and other data features.

From the correlation matrix heatmap shown in Figure 1, it is evident that there is a significant correlation between most data features and fault data. This indicates that faults are often directly caused by the operational status of earlier processes, such as the number of times the capping device presses down on bottle caps onto product bottles and the number of times the screwing device screws on the product bottle caps, which have a direct impact on the occurrence of subsequent faults. Through this intuitive visualization method, we can identify key process data that are closely related to the occurrence of faults, providing important input features for subsequent fault prediction models.

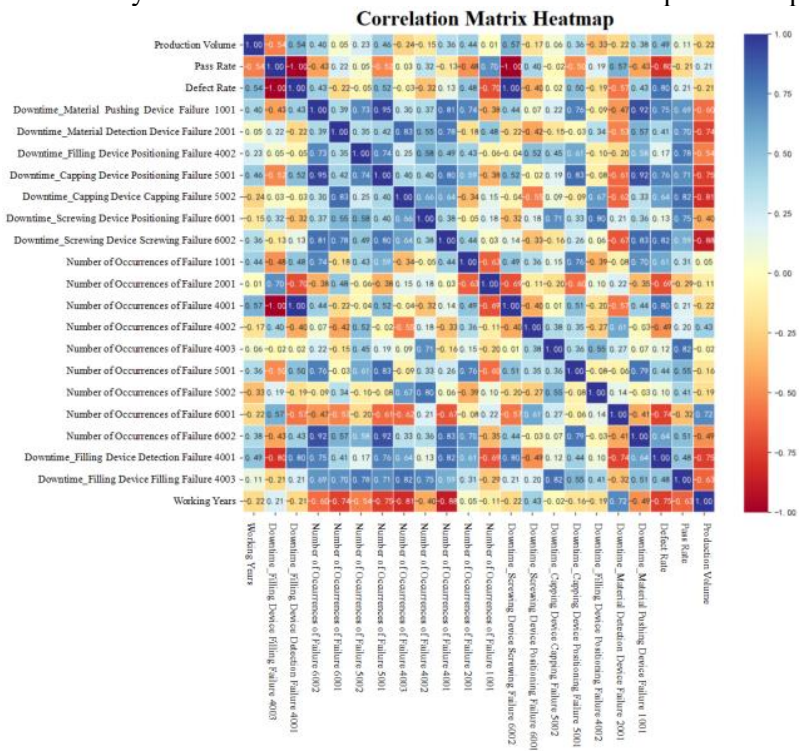

Figure 1. Correlation Matrix Heatmap

3.2 Model Development

This study employs the XGBoost algorithm to construct a fault prediction model for production lines. Initially, the dataset is preprocessed using the pandas library in Python. The specific steps include: handling missing values in the fault data by iterating through the data to mark the start time of faults and calculating the duration of faults, filling all

missing values with 0. Additionally, to address the issue of data imbalance, this study employs oversampling methods to balance the class distribution. During the feature extraction phase, the processed dataset is divided into a feature set and a label set, which are then converted into tensor format under the PyTorch framework for subsequent model training.

```
from imblearn.over_sampling import SMOTE
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import torch
import torch.nn as nn
import torch.optim as optim
from sklearn.model_selection import train_test
_split
from torch.utils.data import DataLoader, Tenso
rDataset
from imblearn.over_sampling import SMOTE
import xgboost as xgb
from sklearn.metrics import accuracy_score
from sklearn.utils import resample
from tqdm import tqdm
import os

def incremental_xgboost_training (file_path, e
xisting_model, new_model_name, params=No
ne, num_rounds=100):
    # Load data
    df = pd.read_csv(file_path)

    # Data preprocessing
    df.iloc[:, 2] = df.iloc[:, 2].str.extract('(\d+)').
astype(int)
    df.iloc[:, -9:] = df.iloc[:, -9:].applymap(lamb
da x: 1 if x != 0 else 0)
    df = df.astype(int)

    error_df = pd.DataFrame()
    fault_columns = [col for col in df.columns if
 'fault ' in col]

    # Calculate fault duration
    for col in fault_columns:
        error_df[f'{col}_start'] = (df[col]! = 0) &
(df[col]. shift (1) == 0)
        error_df[f'{col}_duration'] = df.groupby
((df[col] == 0).cumsum())[col].transform('coun
t') * error_df[f'{col}_start']
        duration_temp = pd.Series(index=error_d
f.index, dtype='float64')
        for i in error_df.index[error_df[f'{col}_sta
rt']]:
            duration_temp.iloc[i:i + error_df.at[i, f'
{col}_duration']] = error_df.at[i, f'{col}_durati
on']

        error_df[f'{col}_duration'] = duration_te
mp.fillna(0).astype(int)

    # Add fault duration to data
    for col in fault_columns:
        df[f'{col}_duration'] = error_df[f'{col}_d
uration']

    X = df.iloc[:, :-18].values
    y = df.iloc[:, -18:-9].values

    X_tensor = torch.tensor(X, dtype=torch.float
32)
    y_tensor = torch.tensor(y, dtype=torch.float
32)

    dtrain = xgb.DMatrix(X_tensor, label=y_ten
sor)

    # If parameters are not provided, set default
parameters
    if params is None:
        params = {
            'max_depth': 3,
            'eta': 0.3,
            'objective': 'binary:logistic',
            'eval_metric': 'auc',
            'num_feature': 28 # Set the number of f
eatures to 28
        }

    # Continue training the model
    model = xgb.train(params, dtrain, num_roun
ds, xgb_model=existing_model)

    # Rename the model
    model.set_attr (name='name', value=new_m
odel_name)

    return model
```

Subsequently, the preprocessed feature sets and label sets are converted into the DMatrix data structure required by the XGBoost algorithm. Building on this, this study employs an incremental learning strategy, combining existing models with new data for training. In terms of parameter settings, this study adjusts

key parameters, including the maximum depth of the trees and the learning rate. After making predictions on the test set, the predicted probability values are converted into integer form prediction labels, and the model's accuracy, precision, recall, and F1 score are calculated after multiple rounds of training to comprehensively evaluate the model's performance. Through the aforementioned steps, this study successfully constructed an XGBoost fault prediction model, providing technical support for the automatic alarming of production line faults.

Table 1. Model Training Results

Evaluation Metrics	Values
Accuracy	97.9745%
Precision	0.0545
Recall	0.16565
F1 Score	0.08204

Table 1 presents the performance evaluation results of the model. The accuracy rate reaches 97.9745%, indicating that the model has a high overall predictive accuracy. However, the precision is only 0.0545, which may be attributed to the high sensitivity of the model leading to more false positives. The recall rate is 0.16565, indicating that a significant number of positive samples are not correctly identified, suggesting a deficiency in the model's ability to recognize positive samples. The F1 score is 0.08204, further reflecting the imbalance in model performance. Accuracy reflects the overall predictive accuracy of the model, precision focuses on the accuracy of positive predictions, recall emphasizes the ability to identify positive samples, and the F1 score is a comprehensive reflection of precision and recall. By utilizing specific machine learning libraries to evaluate model predictions, a deeper understanding of the model's performance can be gained, providing important reference for subsequent model optimization and practical application.

3.3 Prediction

To further understand and address the issues of faults within production lines, and to enhance production efficiency and quality, this study initially employed the seaborn library to generate histograms that graphically represent the distribution of the XGBoost model's predictive outcomes (refer to Figure 2). Histograms serve as a pivotal analytical instrument for assessing the precision and robustness of predictive models, capable of elucidating the probabilistic distributional characteristics of the model's forecasts.

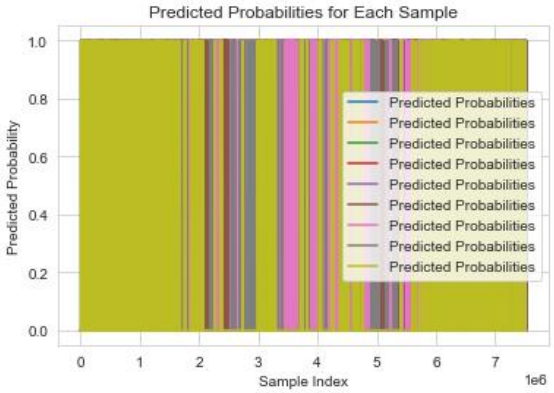

Figure 2. Predicted Probabilities for Each Sample

As depicted in Figure 2, the histogram demonstrates that the predicted probabilities are primarily distributed within the intervals of 0 - 0.2 and 0.6 - 0.8. Within the 0 - 0.2 range, the data points are relatively concentrated, indicating that the model predicts a higher number of negative class samples (i.e., no fault occurrence) with a consistent probability, reflecting a higher certainty in the model's predictions for this range. Conversely, the 0.6 - 0.8 range also contains a certain number of data points, representing the model's prediction of positive class samples (i.e., fault occurrence), but the probability distribution is more dispersed, suggesting a lower certainty in the model's predictions for this segment. Overall, the distribution of the model's predictive outcomes exhibits a certain skewness, which may be related to the frequency of fault occurrences in the actual data. By analyzing the histogram, we can gain a visual understanding of the model's performance tendencies across different prediction categories, providing a basis for subsequent model optimization. For instance, targeting intervals with ambiguous prediction probabilities, further adjustments to model parameters or improvements in feature engineering could be implemented to enhance the model's predictive accuracy and stability. Additionally, calculating the start and end positions of faults, as well as their duration, is crucial for understanding the occurrence and progression of faults, offering important references for fault diagnosis and repair. By creating a DataFrame to record the start times and durations of faults and merging it with the

original data, we can obtain richer information for in-depth analysis and processing. Renaming the column headers and resetting the index enhances the readability and usability of the data, facilitating subsequent data analysis and processing. Saving the resulting DataFrame as a CSV file facilitates integration and sharing with other tools and systems for further analysis and processing. Overall, the aforementioned processes and outcomes are of significant importance for thoroughly understanding and addressing fault issues in production lines, and for improving production efficiency and quality. The prediction results are presented in Table 2.

Table 2. Model Prediction Results (Partial Data Shown)

Fault Id	1001			1002			...
S/N	dt.	Start time	Dur. (sec.)	dt.	Start time	Dur. (sec.)	...
0	12	9072	6	4	8258	3	...
1	15	5250	157	4	8435	4	...
2	17	20882	9	4	10742	7	...
3	17	20902	140	4	10939	7	...
4	24	25432	9	5	10939	7	...
...	...	...	...	...	...	...	...

4. Conclusions

This study has demonstrated that the fault prediction model based on the XGBoost algorithm is highly effective and practical in real-world applications. The model's high accuracy rate has proven its predictive capabilities in complex industrial environments, particularly in reducing production interruptions and optimizing resource allocation. Despite the challenges faced in precision and recall rates, these metrics also reveal areas for improvement in the model's identification of specific fault types. Future work will focus on enhancing these performance indicators and extending the model to more production line scenarios to verify its generalization capabilities. Overall, this study not only provides an effective tool for fault prediction in production lines but also offers strong technical support for the intelligent and automated production in the context of Industry 4.0.

4.1 Model Advantages

The XGBoost model proposed in this study has shown significant advantages in fault prediction for production lines, primarily reflected in its exceptional accuracy rate of up to 97.99%. This not only proves the model's reliability in predicting faults in most cases but also provides strong assurance for the stable operation of production lines. Furthermore, the model's automated alarm system responds promptly to potential production issues, reducing the need for manual intervention, increasing response speed and efficiency, and aiding in the reduction of resource waste and enhancement of overall production efficiency, thus strengthening the scientific basis for decision-making.

4.2 Model Disadvantages

Although the model excels in accuracy, its performance in precision and recall rates leaves room for improvement, which may point to issues with the model's performance under specific conditions, especially in identifying positive sample types. This could be due to data imbalance or the model's insufficient sensitivity to certain features. Therefore, further model optimization is needed to enhance its sensitivity and identification capabilities for fault occurrences while maintaining high accuracy.

4.3 Model Prospect

Looking ahead, the XGBoost model from this study has the potential for further optimization and expansion. By adjusting model parameters and improving feature engineering, the model's predictive accuracy and stability can be enhanced. The application of model ensemble methods, such as combining XGBoost with neural networks, may further improve the model's generalization and predictive performance. Additionally, integrating the model into real-time monitoring systems for real-time fault prediction and alarms will further elevate the level of intelligent production. Ultimately, by extending the models and methods from this study to other

industrial sectors, there is potential to improve the overall efficiency and quality of industrial production.

Acknowledgments

This study was supported by the Guangzhou Huashang College Youth Academic Research Projects (Project No. 2023HSQX044).

References

[1] Stojanović D, Joković J, Tomašević I, et al. Algorithmic approach for the confluence of lean methodology and industry 4.0 technologies: Challenges, benefits, and practical applications. Journal of Industrial Intelligence, 2023, 1(2): 125-135.

[2] Kang, Z., Catal, C., & Tekinerdogan, B. (2020). Machine learning applications in production lines: A systematic literature review. Computers & Industrial Engineering, 149, 106773. https://doi.org/10.1016/j.cie.2020.106773

[3] Crespino, A. M., Corallo, A., Lazoi, M., Barbagallo, D., Appice, A., & Malerba, D. (2016). Anomaly detection in aerospace product manufacturing: Initial remarks. In 2016 IEEE 2nd International Forum on Research and Technologies for Society and Industry Leveraging a Better Tomorrow (RTSI), 1–4. https://doi.org/10.1109/RTSI.2016.7740644

[4] Nakazawa, T., & Kulkarni, D. V. (2019). Anomaly detection and segmentation for wafer defect patterns using deep convolutional encoder–decoder neural network architectures in semiconductor manufacturing. IEEE Transactions on Semiconductor Manufacturing, 32(2), 250–256. https://doi.org/10.1109/TSM.2019.2897690

[5] Paolanti, M., Romeo, L., Felicetti, A., Mancini, A., Frontoni, E., & Loncarski, J. (2018). Machine learning approach for predictive maintenance in industry 4.0. In 2018 14th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA), 1–6. https://doi.org/10.1109/MESA.2018.8449150

[6] Yu, W., Dillon, T., Mostafa, F., Rahayu, W., & Liu, Y. (2020). A global manufacturing big data ecosystem for fault detection in predictive maintenance. IEEE Transactions on Industrial Informatics, 16(1), 183–192. https://doi.org/10.1109/TII.2019.2915846

[7] Aydemir, G., & Paynabar, K. (2020). Image-based prognostics using deep learning approach. IEEE Transactions on Industrial Informatics, 16(9), 5956–5964. https://doi.org/10.1109/TII.2019.2956220

[8] Weber, C., & Reimann, P. (2020). MMP – A platform to manage machine learning models in Industry 4.0 environments. In 2020 IEEE 24th International Enterprise Distributed Object Computing Workshop (EDOCW), 91–94. https://doi.org/10.1109/EDOCW49879.2020.00025

[9] Elshaboury, N., Al-Sakkaf, A., Alfalah, G., & Abdelkader, E. M. (2022). Data-Driven Models for Forecasting Failure Modes in Oil and Gas Pipes. Processes, 10, 400. https://doi.org/10.3390/pr10020400

[10] Liu, W., Chen, Z., & Hu, Y. (2020). XGBoost algorithm-based prediction of safety assessment for pipelines. Journal of Natural Gas Science and Engineering, 73, 103377. https://doi.org/10.1016/j.jngse.2020.103377

[11] Pang, J. L. (2023). Adaptive Fault Prediction and Maintenance in Production Lines Using Deep Learning. International Journal of Simulation Modelling, 22(4), 734-745. https://doi.org/10.2507/IJSIMM22-4-CO20

[12] Zhang S, Zhu X, Anduv B, et al. Fault detection and diagnosis for the screw chillers using multi-region XGBoost model. Science and Technology for the Built Environment, 2021, 27(5): 608-623.

[13] Zhang C, Wang D, Song C, et al. Interpretable learning algorithm based on XGBoost for fault prediction in optical network//2020 Optical Fiber Communications Conference and Exhibition (OFC). IEEE, 2020: 1-3.

