Available *online* at : http://jnte.ft.unand.ac.id/

**Jurnal Nasional Teknik Elektro**

| ISSN (Print) 2302-2949 | ISSN (Online) 2407-7267 |

# Intermittent Oscillation Diagnosis in a Control Loop Using Extreme Gradient Boosting

*Dana Fatadilla Rabba, Awang Noor Indra Wardana, Nazrul Effendy*

*Department of Nuclear Engineering and Engineering Physics, Faculty of Engineering, Universitas Gadjah Mada, Indonesia*

ARTICLE INFORMATION

Received: July 13, 2022
Revised: October 24, 2022
Available online: November 30, 2022

KEYWORDS

Control loop, oscillation diagnosis, feature extraction, extreme gradient boosting

CORRESPONDENCE

Phone: +62 (0274) 580882
E-mail: awang.wardana@ugm.ac.id

**A B S T R A C T**

The control loop in the industry is a component that must be maintained because it will determine the plant's performance. Most industrial controllers experience oscillations with various causes, such as noise, oscillation, backlash, dead band, hysteresis, random variation, and poor controller tuning. The oscillation diagnosis system, which can understand the oscillation type characteristics, is built based on machine learning because it is dynamic and not based on specific rules. This study developed an online oscillation diagnosis program using the extreme gradient boosting (XGBoost) method. The data was obtained through the simulation of the Tennessee Eastman process. The data is segmented on specific window sizes, and then time series feature extraction is performed. The extraction results are then used to build an XGBoost model capable of performing oscillation diagnosis tasks. There are seven types of oscillations tested in this study. The model that has been made is implemented online with the help of sliding windows. The results show that the XGBoost model performs best when the data window size is 100, with the accuracy performance and the F1 score of the model in classifying the type of oscillation being 0.918 and 0.905, respectively. The model can detect the type of oscillation with an average diagnosis time of 712 seconds on diagnostic tests.

## INTRODUCTION

The loop control system is a crucial component in the industrial world. It is important because the control group functions to maintain the process variables at their set point so that the quality of the products produced is under predetermined specifications [1]. In addition, the performance of the control group in a factory represents the condition of the factory. The occurrence of oscillations in the control loop indicates that the control loop has poor performance [1].

Problems related to this performance make control loop performance monitoring (CPM) an essential aspect to improve. CPM includes various automation techniques to ensure controllers, actuators, and sensors are functioning correctly [2]. Among other sub-fields of CPM, oscillation detection is one of the issues that has received substantial attention due to the frequent occurrence of control loop oscillations and their negative impact on plant profitability. The oscillations directly impact the plant's regular operation, energy consumption, and raw materials, thus affecting the plant's profitability.

Based on a survey conducted on 26,000 PID controllers from various process industries, there are 16% of control loops with excellent performance, 16% with acceptable performance, 22% with moderate performance, 10% with poor performance, and the remaining 36% open control loop [3]. It shows that the control loop oscillation is an unresolved problem.

Every control loop in the industry must be checked to determine the oscillation occurrence in the control loops [4], [5]. Typically, process industries have between 500 and 5000 control loops [6]. Visual inspection consumes resources and may result in undetected oscillations [7]. An automatic oscillation detection technique is needed in an industrial process to overcome this limitation.

Oscillation problems can be caused by noise, oscillation, backlash, dead band, dead zone, hysteresis, random variation, and wrong controller tuning [8]. The many causes of the oscillations are also a challenge in the process industry. Solving oscillation problems and developing solutions becomes more manageable when the cause of oscillations can be identified and distinguished with certainty. In addition to many reasons that need to be detected, the oscillations that appear intermittently are challenging. Intermittent oscillation is a type of oscillation that occurs at irregular intervals and occurs intermittently. The characteristics of this type of oscillation appear spontaneously erratic and can disappear automatically [9]. The duration and intensity of the intermittent oscillations are indeterminate. The difference between the occurrence of sporadic oscillations with normal stable oscillations

https://doi.org/10.25077/jnte.v11n3.1040.2022

makes intermittent oscillations more challenging to detect. In addition, the causes of intermittent oscillations that appear can vary over a period. Therefore, oscillations that occur intermittently from various types of sources need to be detected automatically to reduce the decline in system performance in the long term.

The current oscillation detection method is mainly conducted offline. Several techniques are still performed offline, such as integral of absolute control error, zero crossing of auto covariance, wavelet transform, modified empirical mode decomposition, and discrete cosine transform [10]. One of the difficulties in implementing this method in online mode is the determination of rule-based evaluation and window size [11]. Therefore, another technique that can adapt to new signals and does not require complicated rules must be applied.

As technology develops, the innovative factory concept encourages using sensors, devices, and intelligent machines so that factories can continuously collect production-related data [11]. The creative factory concept enables a change in the error detection program at the factory. The program was initially based on a method with specific rules and worked in offline mode to be based on a machine learning method that is intelligent and can work in online mode (automatic) [10]. With the contribution of machine learning algorithms, error detection can be carried out accurately and as early as possible, thereby reducing factory downtime [12]. In intermittent oscillation detection, machine learning algorithms are applied to make the implementation more straightforward and robust. Several machine learning methods have been applied to several applications [12]–[18].

Research on methods for detecting control loop oscillations in the process industry has been widely carried out and published during the last few years. However, the oscillation detection method applied is still rule-based. It works in offline mode (manual), while industrial needs demand a technique that is smart, not complex, and works in an online way (automatic) [11]. Several studies discuss the application of machine learning for oscillation detection, especially in the Tennessee Eastman (TE) process. As Downs and Vogel [11] proposed, the TE process has been used as a benchmark in various process control and monitoring studies. Several previous studies are described as follows.

Gao and Hou [19] devised a method for identifying oscillation defects based on a support vector machine (SVM) model coupled with principal component analysis (PCA) and grid search (GS) in the TE process. PCA decreases feature dimensions, whereas GS optimizes parameters. After the initial scaling and normalization of the data, PCA was used to minimize the dimensional features. The performance of SVM is then improved using GS performance in a subsequent step using a genetic algorithm and particle swarm optimization. The SVM model and PCA-GS combination offer a remarkable classification accuracy of close to 99%. The GS-PCA-based SVM was compared to other SVM-based error diagnosis methods in simulation testing with the TE Process. It had the highest average of 96.77% and the best computing efficiency with the lowest average duration of 1.35 seconds.

A deep belief network (DBN)-based oscillatory fault detection technique was proposed by Zhang and Zhao [20] in 2017. An improved DBN model with a one-class-one-model architecture is used to extract features from continuous process data in the spatial and temporal domains. The findings demonstrate that when diagnosing random-type faults, the DBN sub-network with Gaussian activation function outperformed the sub-network with Sigmoid activation function. Additionally, the enhanced DBN model-based fault diagnosis used in the TE process performed well, with an average error diagnosis rate of 82.1% for all 20 faults.

Wu and Zhao investigated the application of a deep convolutional neural network (DCNN) model for diagnosing faults in the Tennessee Eastman (TE) chemical process [21]. The DCNN model is built with a convolutional layer, a pooling layer, a dropout layer, and a fully connected layer to extract features in the spatial and temporal domains. A feature map of dimension m x n is created from the raw process data, where m denotes the length of the sample time and n is the total number of variables. The experimental findings demonstrate the effectiveness of the DCNN-based model for fault diagnosis in the TE process. All 20 different types of faults had an average fault detection rate (FDR) of 88.2%, except for three problems that were extremely challenging to identify. Additionally, dynamic diagnosis time and diagnostic performance were investigated, with an optimum diagnosis time of 2-3 hours and an FDR of 0.941.

Park et al. [22] investigated fault diagnosis and detection using a combination autoencoder and long short-term memory (LSTM) network in 2019. A straightforward autoencoder assesses multivariate time series data to find unusual errors for fault detection while training the model using standard data. The LSTM network is then given the predicted raw fault data to determine the fault type. To accelerate the convergence of the overall structure with fewer numerical issues, two trained autoencoder and LSTM models are coupled. The performance of fault identification and diagnosis is then assessed using the TE process benchmarks. The autoencoder and LSTM network work together quickly and correctly to detect deviations from typical behavior. Yao et al. [23] developed a hybrid intelligent fault diagnosis strategy for chemical processes using penalty iterative optimization. The models used in the research are DBN and adaptive lifting wavelet (ALW). The ALW method can adaptively set the threshold function to match different data sets.

Based on several studies that have been reviewed, most of the studies use offline methods, and only two studies discuss online applications. The term offline in this study refers to a program that works without being connected to a network. This method is usually applied regularly to check for faults in the control loops. However, the program cannot detect in real time when an oscillation disturbance occurs in a process variable. Meanwhile, the term online refers to a program directly connected to a network. The sensor measurement results are sent directly and continuously to a computer through a predetermined grid, in this case, using a simulator. The program in the computer will immediately classify the type of oscillation as soon as the data is sent. This method can predict sudden changes in variables and detect oscillation disturbances in real time when fluctuations first appear [24].

Therefore, this study developed a method for detecting intermittent oscillation using the extreme gradient boosting (XGBoost) method that can work online. Furthermore, the machine learning method

was chosen because the model built is more straightforward, and the computation time is lower than deep learning. In addition, XGBoost computing is rated for its quiet processing time, which is efficient and easily scalable [25].

## METHOD

The data used in this research is in the form of TE process output variable data, totaling 52 variables containing 11 manipulated variables and 41 measured variables. TE process data was obtained through simulations performed for each type of fault for 8 hours. The TE process includes the overall dynamics of a chemical plant with 20 marks, with five different types: step, random variation, stiction, tuning error, and unknown. To get more optimal results, the detection of oscillations in this study was only carried out on the reactor and condenser. Figure 1 shows the process diagram of the TE process.

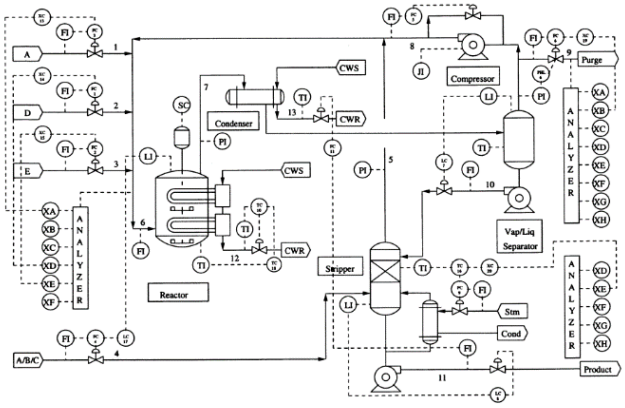


Figure 1. Diagram of Tennessee Eastman process [26]

There are six faults in the reactor and condenser, each type of step, random variation, and stiction. However, the stiction fault in the reactor coolant loop is not used because the oscillations are not visible and are not easy to differentiate. The tuning error type fault is added to this research by changing the controller constant when the simulation runs, with tuning errors of 0.5x and 1.5x of the proportional constant. The total length of the data from this study is 230,400 data with details of 7 types of oscillating data, and 1 type of non-oscillating data, each type of data is 28,800. The variables used in this study are shown in Table 1. IDV (15) was not studied because, in previous studies conducted by Zhan et al. [20], Wu et al. [21], and Jia et al. [27], IDV (15) is very difficult to classify. IDV (15) behavior is like IDV (0), so the program will need to diagnose no oscillation.

The dataset of simulation results for 8 hours was segmented based on the size of the window to be tested: 50, 100, 150, 200, and 250. The label of each segment will follow the label of the faults. Furthermore, time series feature extraction is used to extract a dataset of time series characters. Feature extraction applies three domains, statistical, temporal, and spatial. Features with a high degree of importance and correlation are selected to reduce the number of features.

Table 1. Details of the Faults in This Study

| TE's fault name | Process Variables | Oscillation's Type | Fault's name in this study |
|---|---|---|---|
| IDV (0) | Normal/No fault | Normal | IDV (0) |
| IDV (4) | The temperature of the reactor cooling water inlet | Step | IDV (1) |
| IDV (5) | The temperature of the condenser cooling water inlet | Step | IDV (2) |
| IDV (11) | The temperature of the reactor cooling water inlet | Random variation | IDV (3) |
| IDV (12) | The temperature of the condenser cooling water inlet | Random variation | IDV (4) |
| IDV (14) | Condenser cooling water valve | Sticking | IDV (5) |
| Additional IDV | Low proportional tuning constant | Tuning Error | IDV (6) |
| Additional IDV | High proportional tuning constant | Tuning Error | IDV (7) |

Then PCA is applied to condense the extensive set of variables into a smaller group that retains most of the data information, reducing the dimensions of the large dataset. Accuracy inherently suffers when a data set's number of variables is reduced. Even said, PCA can minimize dimensionality while preserving the essential details of the data by sacrificing a little accuracy for simplicity. In simple terms, PCA can be calculated by the following steps:

1. determine the dataset,
2. subtract each value from the average of its corresponding column,
3. compute the covariance matrix,
4. calculate the eigenvectors and eigenvalues of the covariance matrix,
5. choose components and create a feature vector, and
6. obtain a new set of data.

Furthermore, the XGBoost model is built for oscillation diagnosis using Python. XGBoost is a decision tree based on gradient boosting designed to be easily scalable [25]. Like gradient boosting in general, XGBoost builds on the additive expansion of the objective function by minimizing the loss function. The XGBoost model is created with varying data window sizes, namely 50, 100, 150, and 200. Then the model is tested with test data that has been prepared to represent the model's performance when used on data that has not been studied. The XGBoost model testing results are then evaluated using accuracy and F1 score metrics.

The next step is to vary the hyperparameters of the XGBoost model to get the model with the best performance. The hyperparameters that are varied are learning rate (0.05, 0.1, 0.15, 0.2), minimum child weight (1, 3, 4, 5), maximum depth (1, 3, 5, 7), minimum split loss (0, 0.1, 0.2, 0.3) and alpha (0, 0.001, 0.005, 0.01). Hyperparameter variation is done by combining all


https://doi.org/10.25077/jnte.v11n3.1040.2022

hyperparameters. Overall, a total of 1024 model combinations were tested and compared. The main metrics compared to this model are the F1 score and accuracy. The model with the best performance is used in the next step.

The best XGBoost models are implemented online. TE output variable data is sent in real-time with the Message Queuing Telemetry Transport (MQTT) protocol at certain time intervals. The program will first wait for the received data to fill the data window. When the data window is complete, the program will perform a time series feature extraction with the features specified in the data preparation step. The data extraction results are then fed to the XGBoost model. The XGBoost then produces the oscillation diagnosis results. The results of online model testing are the time it takes for the model to detect and classify the type of oscillation that appears correctly.

On the detection of oscillations of each type, the simulation is run for 30 minutes. This type of fault is activated after the first 100 seconds. Correct diagnosis is considered if the model can correctly detect the type of oscillation 30 times. The time interval for each diagnosis is 15 seconds. Meanwhile, the intermittent oscillation classification test was carried out by activating a fault type for 30 minutes and then turning it off for 15 minutes. After that, another kind of fault is triggered for another 30 minutes. The model's behavior in classifying the type of intermittent oscillation is tested. The detection time of each changing oscillation is calculated to compare it with the average time of the online classification results for every single non-intermittent type of fault.

## RESULTS AND DISCUSSION

### *Time Series Feature Extraction and Selection Results*

In the data preparation stage, the simulation data for 8 hours is segmented based on the size of the windows to be tested, which are 50, 100, 150, 200, and 250, respectively. The label of each segment will follow the label of the control group. Furthermore, time series feature extraction is performed on XMV and XMEAS data from each data subset using the TSFEL library. The results of feature extraction are then combined into one complete dataset. Finally, the extracted dataset is selected to reduce the number of features.

Feature selection is made by creating a simple XGBoost model on all existing data. A single decision tree calculates the importance of the feature for each attribute. A single decision tree performance measure is measured by the purity (Gini index) used to select the specified error function. In simple terms, the more attributes used to make critical decisions with a single decision tree, the higher their relative importance.

Based on the feature importance index, several variables are obtained: total energy, wavelet energy, FFT mean coefficient, mean absolute deviation, spectral centroid, spectral kurtosis, spectral decrease, turning points, wavelet energy, and ECDF percentile. The complete extraction of these variables is then used to filter the entire data. As a result, the filtered data was reduced by more than 90%, from 9620 variables to 936 variables. These 936 variables are used for processing in the next step.

### *Effect of PCA Implementation*

The dataset selected for variables is reprocessed using the PCA method. PCA is a method that is often used to reduce the number of dimensions of the data while retaining information from the dataset [20]. However, in this study, the PCA applied was also tested to determine whether the new variable PCA results could be used as an additional variable.

The dataset selected for variables is reprocessed using the PCA method. PCA is a method that is often used to reduce the number of dimensions of the data while retaining information from the dataset [20]. However, in this study, the PCA applied was also tested to determine whether the new variable PCA results could be used as an additional variable.

When performing value transformations, the value of the eigenvectors in the covariance matrix is correlated with the total number of eigenvectors. The larger the eigenvalues, the more significant the variance explained by the principal components. Each of the variances is cumulatively added until it reaches a predetermined threshold. In this study, a 95% threshold of explainable variance is used.

Based on Figure 2, the number of components has a reasonably significant variance value at first. It can be seen from the graph that the cumulative variance increase is relatively high up to the 200th component. After that amount, the cumulative variance graph becomes sloping. The number of features of 200 represents 95% of the variance of the entire column (initially 936 columns). Therefore, 200 main components of PCA are selected for further modeling.

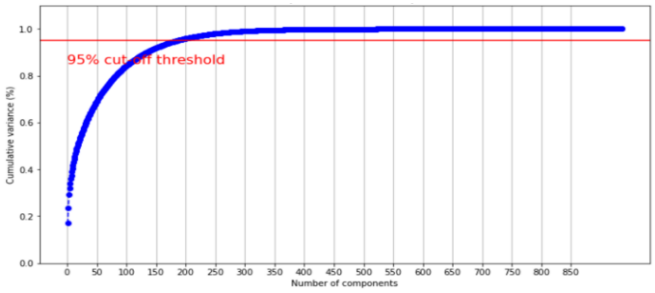


Figure 2. Graph of the Number of Components versus Cumulative Variance

PCA-processed datasets are compared with unprocessed datasets and PCA-combined datasets. The test results after PCA processing are shown in Figures 3 and 4. The three results show no significant difference between the accuracy and F1 scores. However, the model combined with the PCA feature has better performance, with an accuracy value of 0.91 and an F1 value of 0.9.

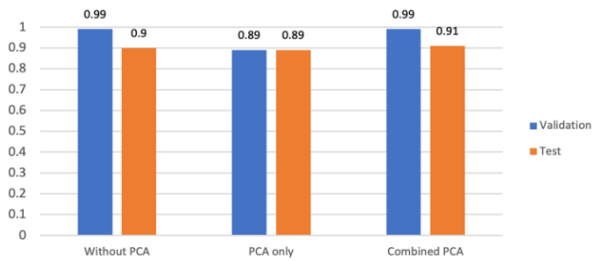


Figure 3. Graph of Accuracy Test Results with PCA

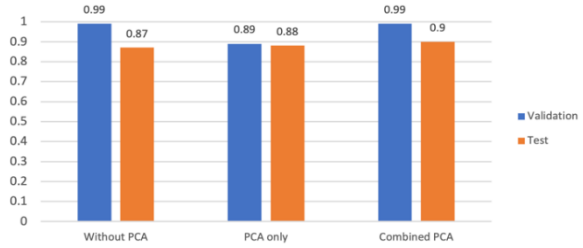


Figure 4. Graph of F1 score Test Results with PCA

### *Effect of Window Size*

The window size is an important thing that needs to be optimized when the model testing is in online mode. The selection of a good window size can capture the oscillations that occur correctly so that no oscillation information is lost. Tests were carried out on window sizes of 50, 100, 150, 200, and 250. Variations in window size also affect the amount of data obtained for training, validation, and testing. Figure 5 shows the difference in the amount of data received.

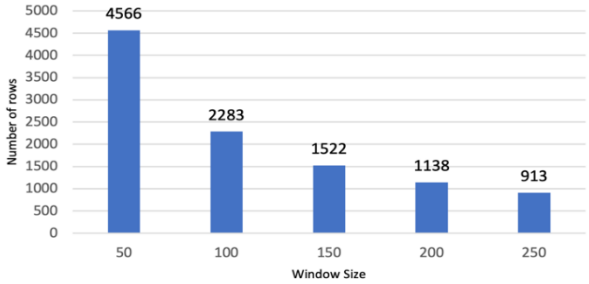


Figure 5. Graph of the Effect of Window Size Variations on the Number of Rows

Simply put, the larger the window size, the less data we will get. Because a large window size requires extensive data, the amount of data obtained is less. Then the various window sizes were tested for the performance of the XGBoost model. The results of the window size test on accuracy and F1 score are shown in Figure 6 and Figure 7, respectively.

Figure 5 shows the model has a good performance with an accuracy above 80%. The decrease in validation and test data accuracy is expected because the model has never encountered the data during training. If we look further, models with outstanding accuracy are obtained by models with window sizes of 50, 100, and 150 because there is no significant difference in the accuracy values, both during validation and testing. However, the window size 100 is the best of the accuracy tests carried out because it has the highest accuracy value.

The results of the F1 score test in Figure 6 show the model has a pretty good performance, with all F1 scores above 75%. The decrease in the F1 score occurred in the test data because it was new data outside the training dataset. If we look further, the model with an excellent F1 score is obtained by the model with a window size of 50 and 100 because the F1 score value has no significant difference, both during validation and testing. However, the window size of 100 is the best from the F1 score test carried out because it has the highest F1 score during validation and testing.

### *Effect of Hyperparameter's Variation*

Optimization is carried out on hyperparameters to get better XGBoost model performance. The hyperparameters consist of learning rate (0.05, 0.1, 0.15, 0.2), minimum child weight (1, 3, 4, 5), maximum depth (1, 3, 5, 7), minimum split loss (0, 0.1, 0.2, 0.3) and alpha (0, 0.001, 0.005, 0.01). The system was optimized by combining all hyperparameters with four variations of hyperparameters. The procedure produces 1024 possible combinations. The results of the performance test get the XGBoost model with the following parameters:

1. learning rate = 0.2,
2. maximum depth = 7,
3. minimum child weight = 7,
4. minimum split loss = 0.2 and
5. alpha = 0.005

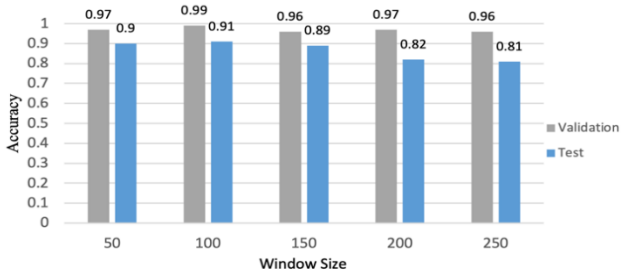


Figure 6. Graph of Window Size Variation Test on Accuracy

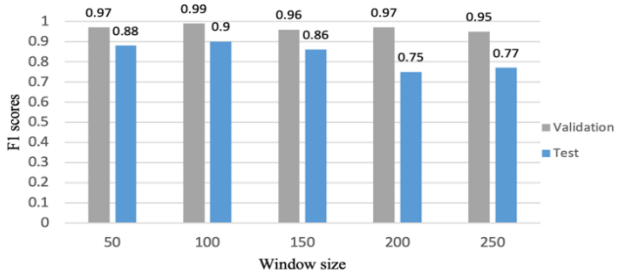


Figure 7. Graph of Window Size Variation Test on F1 score

The results of the hyperparameter combination optimization increase the accuracy and F1 score, as shown in Figure 8. However, the increase in model performance was not significant. Compared with the initial model without optimization, the initial model has an accuracy value and F1 score of 0.910 and 0.900, respectively. In contrast, the model after optimization has an accuracy value and f-score of 0.918 and 0.905, respectively. Although it does not significantly improve performance, hyperparameter optimization can improve performance by preventing the model from being overfitted.

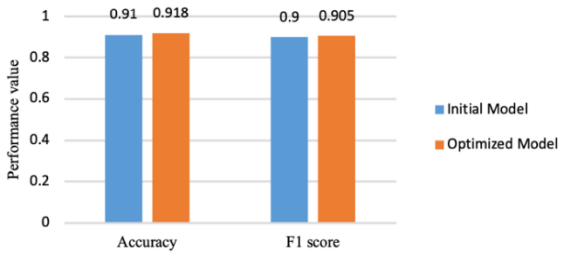


Figure 8. Comparison graph of model performance before and after optimization


https://doi.org/10.25077/jnte.v11n3.1040.2022

### *Diagnosis Oscillation Test*

Online oscillation diagnosis testing tests the time it takes for each type of oscillation to be detected correctly by the program. The program can detect fluctuations correctly if it can detect the correct type of fault three times in a row. The online oscillation diagnosis test results are shown in Table 2.

Table 2. Online Oscillation Diagnosis Time Test Results for Each Fault

| Fault Type | Diagnosis Time (s) |
|---|---|
| IDV (1) | 870 |
| IDV (2) | 1120 |
| IDV (3) | 1170 |
| IDV (4) | 840 |
| IDV (5) | 330 |
| IDV (6) | 310 |
| IDV (7) | 350 |
| Average | 712 |

### *Diagnosis of Oscillation for Each Type*

Figures 9-15 show the program's behavior when it detects each oscillation type. The program can catch very well at IDV 0 or in regular operation without significant errors. It is because the standard procedure is the default value which should not have any mistakes. Meanwhile, the program shows a difference in behavior when detecting IDV 1 (Figure 9) and 2 (Figure 10) oscillations. At the beginning of the initiation of the change, the program detects IDV 3 and 4 for IDV 1 and 2, respectively. The step oscillation initially only affects some variables, making the program detect that the fluctuation is a random variation. However, after about 5 minutes, the program could correctly see IDV 1 and 2. Then IDV 3 (Figure 11) and 4 (Figure 12) are difficult to detect. It can be seen in the figure that the program is only able to detect this type of fault once, and then the prediction returns to normal condition. The occurrence and length of time are random, making it difficult for the program to detect them correctly. The program takes more than 13 minutes to detect this type of interference properly. For IDV 5 (Figure 13), the program behavior is almost the same as when seeing IDV 1 and 2, where the program detects random variables in both the reactor and condenser. The behavior difference of IDV 1, 2, 5, 6, and 7 tends to be seen faster by the program due to the significant difference between the process variable values.

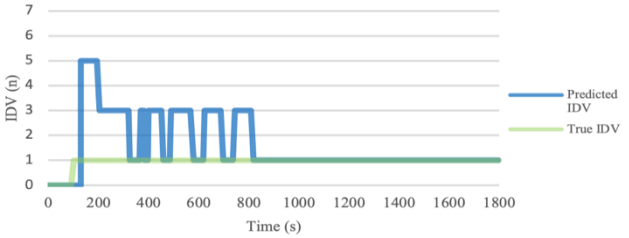


Figure 9. Prediction results of the model under IDV (1)

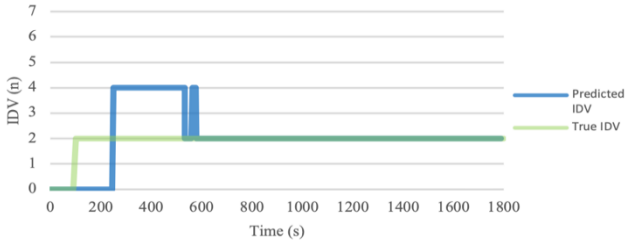


Figure 10. Prediction results of the model under IDV (2)

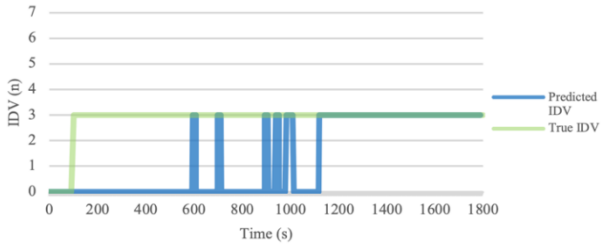


Figure 11. Prediction results of the model under IDV (3)

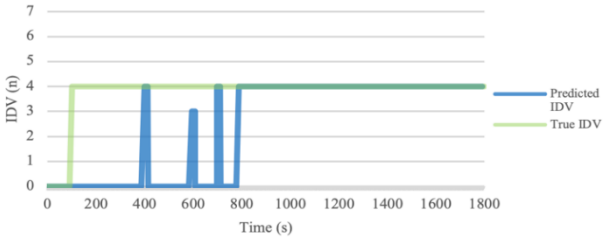


Figure 12. Prediction results of the model under IDV (4)

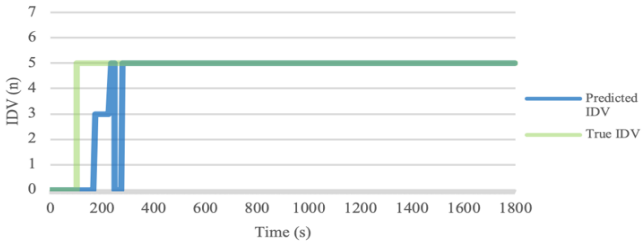


Figure 13. Prediction results of the model under IDV (5)

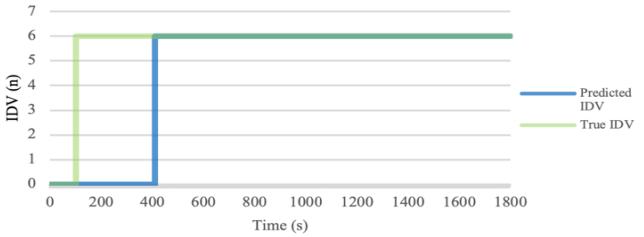


Figure 14. Prediction results of the model under IDV (6)

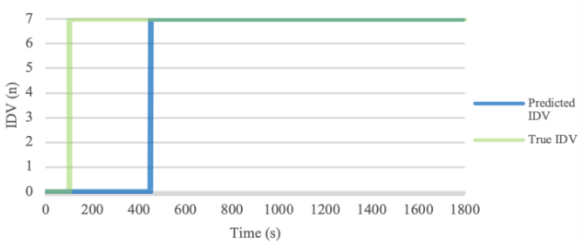


Figure 15. Prediction results of the model under IDV (7)

We compared the computational complexity of the proposed system with the DCNN method. Five selected faults are compared and shown in Table 3. Due to differences in IDV names, IDV in the previous study was named according to this study. In addition, the test conditions are different. Wu et al. used the sampling length of 20 and 10 with a three minutes interval [21], while we used the data length of 100 with a one-second interval. However, the sampling period was the same, i.e., 15 seconds. The calculation of the diagnosis time is based on the time it takes the model to diagnose the fault 30 times consecutively correctly. Overall, the average diagnosis time required for each type of fault since its initiation is 712 seconds. The model can diagnose faults faster than the DCNN model with a sample length of 20, although it is still slower than the DCNN model with a sample length of 10.

Table 3. Comparison of the fault diagnosis times (s) of each model

| Fault Type | DCNN (Sample length = 20) [21] | DCNN (Sample length = 10) [21] | This Study |
|---|---|---|---|
| IDV (1) | 480 | 480 | 870 |
| IDV (2) | 1560 | 1320 | 1120 |
| IDV (3) | 1140 | 1080 | 1170 |
| IDV (4) | 780 | 540 | 840 |
| IDV (5) | 720 | 720 | 330 |
| Average | 936 | 828 | 866 |

### *Online Intermittent Oscillation Diagnosis Test*

The intermittent oscillation test online is done by changing the type of oscillation after 30 minutes of an oscillation introduced. The overall test results are shown in Figures 16 and 17. The results indicate the model can detect well because the oscillations can be detected correctly, although not constant, both in test 1 and test 2. It is normal because every new data obtained can affect the overall information of the existing oscillations. The model can detect precisely and stably in less than 900 seconds. The value follows the previous test with an average diagnosis time of 866 seconds for IDV (1) to IDV (5).

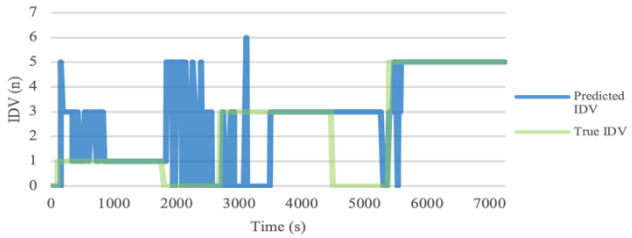


Figure 16. Testing Result of Intermittent Oscillation 1

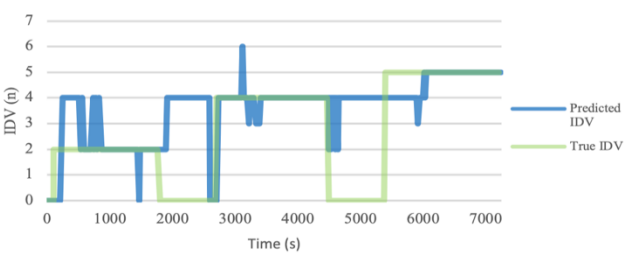


Figure 17. Testing Result of Intermittent Oscillation 2

## CONCLUSIONS

The research apply the XGBoost method to an oscillation diagnosis program that can work online and in real time. The testing results indicate the XGBoost model's best performance by a data window size of 100. The model has an excellent performance in classifying the type of oscillation with an accuracy value and F1 score of 0.918 and 0.905, respectively.

In addition, based on the diagnostic test that has been done, the program can detect oscillations quite well. Since its initiation, the average diagnosis time required for each type of fault is 712 seconds. The model can diagnose faults faster than the DCNN model with a sample length of 20, although it is still slower than the DCNN model with a sample length of 10.

## ACKNOWLEDGEMENTS

We thank the Faculty of Engineering, Universitas Gadjah Mada, for providing facilities for this research.

## REFERENCES

[1] D. Zheng, X. Sun, S. K. Damarla, A. Shah, J. Amalraj, and B. Huang, "Valve Stiction Detection and Quantification Using a K-Means Clustering Based Moving Window Approach," *Ind. Eng. Chem. Res.*, vol. 60, no. 6, pp. 2563–2577, Feb. 2021, doi: 10.1021/acs.iecr.0c05609.

[2] Y. Shardt *et al.*, "Determining the state of a process control system: Current trends and future challenges," *Can. J. Chem. Eng.*, vol. 90, no. 2, pp. 217–245, Apr. 2012, doi: 10.1002/cjce.20653.

[3] R. Srinivasan, R. Rengaswamy, and R. Miller, "Control Loop Performance Assessment. 1. A Qualitative Approach for Stiction Diagnosis," *Ind. Eng. Chem. Res.*, vol. 44, no. 17, pp. 6708–6718, Aug. 2005, doi: 10.1021/ie0490280.

[4] M. Yağcı and Y. Arkun, "An Integrated Application of Control Performance Assessment and Root Cause Analysis in Refinery Control Loops," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 11650–11655, Jan. 2020, doi: 10.1016/j.ifacol.2020.12.651.

[5] S. Shayestehmanesh and J. C. Peyton Jones, "Stochastic modeling and prediction of IMEP for closed loop knock control performance assessment," *Control Engineering Practice*, vol. 92, p. 104130, Nov. 2019, doi: 10.1016/j.conengprac.2019.104130.

[6] L. Desborough and R. Miller, "Increasing Customer Value of Industrial Control Performance Monitoring — Honeywell's Experience," *AIChE Symp. Ser.*, p. 153, 2002.

[7] M. A. Paulonis and J. W. Cox, "A Practical Approach for Large-Scale Controller Performance Assessment, Diagnosis, and Improvement," *J. Process Control*, vol. 13, p. 155, 2003.

[8] J. W. V. Dambros, J. O. Trierweiler, and M. Farenzena, "Oscillation detection in process industries – Part I: Review of the detection methods," *Journal of Process Control*, vol. 78, pp. 108–123, Jun. 2019, doi: 10.1016/j.jprocont.2019.04.002.

[9] Y. Zhao, J. Zhang, and Q. Zhao, "Online Monitoring of Low-Frequency Oscillation Based on the Improved Analytical Modal Decomposition Method," *IEEE Access*, vol. 8, pp. 215256–215266, 2020, doi: 10.1109/ACCESS.2020.3040848.


https://doi.org/10.25077/jnte.v11n3.1040.2022

[10] L. Xie, X. Lang, A. Horch, and Y. Yang, "Online Oscillation Detection in the Presence of Signal Intermittency," *Control Eng. Pract.*, vol. 55, p. 91, 2016.
[11] J. W. V. Dambros, J. O. Trierweiler, M. Farenzena, and M. Kloft, "Oscillation Detection in Process Industries by a Machine Learning-Based Approach," *Ind. Eng. Chem. Res.*, vol. 58, no. 31, pp. 14180–14192, Aug. 2019, doi: 10.1021/acs.iecr.9b01456.
[12] N. Effendy, E. D. Kurniawan, K. Dwiantoro, A. Arif, and N. Muddin, "The Prediction of Oxygen Content of the Flue Gas in a Gas-Fired Boiler System Using Neural Networks and Random Forest," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 11, no. 3, pp. 923–929, 2022.
[13] S. N. Sembodo, N. Effendy, K. Dwiantoro, and N. Muddin, "Radial basis network estimator of oxygen content in the flue gas of debutanizer reboiler," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 12, no. 3, pp. 3044–3050, 2022, doi: 10.11591/ijece.v12i3.pp3044-3050.
[14] N. Effendy, D. Ruhyadi, R. Pratama, D. F. Rabba, A. F. Aulia, and A. Y. Atmadja, "Forest quality assessment based on bird sound recognition using convolutional neural networks," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 12, no. 4, Art. no. 4, Aug. 2022, doi: 10.11591/ijece.v12i4.pp4235-4242.
[15] M. A. Dwijaya, U. A. Ahmad, R. P. Wijayanto, and R. A. Nugrahaeni, "Model Design of The Image Recognition of Lung CT Scan for COVID-19 Detection Using Artificial Neural Network," *JNTE*, pp. 21–28, Mar. 2022, doi: 10.25077/jnte.v11n1.984.2022.
[16] M. Ismail and S. A. D. Prasetyowati, "Classification Of Alcohol Type Using Gas Sensor And K-Nearest Neighbor," *JNTE*, pp. 59–64, Mar. 2022, doi: 10.25077/jnte.v11n1.989.2022.
[17] K. Amiroh, B. A. S. Aji, and F. Z. Rahmanti, "Real-Time Accident Detection Using KNN Algorithm to Support IoT-based Smart City," *JNTE*, pp. 65–70, Mar. 2022, doi: 10.25077/jnte.v11n1.999.2022.
[18] E. D. Kurniawan, N. Effendy, A. Arif, K. Dwiantoro, and N. Muddin, "Soft sensor for the prediction of oxygen content in boiler flue gas using neural networks and extreme gradient boosting," *Neural Comput & Applic*, 2022, doi: 10.1007/s00521-022-07771-8.
[19] X. Gao and J. Hou, "An improved SVM integrated GS-PCA fault diagnosis approach of Tennessee Eastman process," *Neurocomputing*, vol. 174, pp. 906–911, Jan. 2016, doi: 10.1016/j.neucom.2015.10.018.
[20] Z. Zhang and J. Zhao, "A deep belief network based fault diagnosis model for complex chemical processes," *Computers & Chemical Engineering*, vol. 107, pp. 395–407, Dec. 2017, doi: 10.1016/j.compchemeng.2017.02.041.
[21] H. Wu and J. Zhao, "Deep convolutional neural network model based chemical process fault diagnosis," *Computers & Chemical Engineering*, vol. 115, pp. 185–197, Jul. 2018, doi: 10.1016/j.compchemeng.2018.04.009.
[22] P. Park, P. D. Marco, H. Shin, and J. Bang, "Fault Detection and Diagnosis Using Combined Autoencoder and Long Short-Term Memory Network," *Sensors*, vol. 19, no. 21, p. 4612, Oct. 2019, doi: 10.3390/s19214612.
[23] Y. Yao, J. Zhang, W. Luo, and Y. Dai, "A Hybrid Intelligent Fault Diagnosis Strategy for Chemical Processes Based on Penalty Iterative Optimization," *Processes*, vol. 9, no. 8, p. 1266, Jul. 2021, doi: 10.3390/pr9081266.
[24] D. F. Rabba, "Klasifikasi Jenis Osilasi Intermiten pada Kalang Kontrol dengan Ekstraksi Fitur Time Series dan Extreme Gradient Boosting," Undergraduate Final Project, Universitas Gadjah Mada, Yogyakarta, 2022.
[25] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco California USA, Aug. 2016, pp. 785–794. doi: 10.1145/2939672.2939785.
[26] J. J. Downs and E. F. Vogel, "A plant-wide industrial process control problem," *Computers & Chemical Engineering*, vol. 17, no. 3, pp. 245–255, Mar. 1993, doi: 10.1016/0098-1354(93)80018-I.
[27] X. Jia, W. Tian, C. Li, X. Yang, Z. Luo, and H. Wang, "A Dynamic Active Safe Semi-Supervised Learning Framework for Fault Identification in Labeled Expensive Chemical Processes," *Processes*, vol. 8, no. 1, p. 105, Jan. 2020, doi: 10.3390/pr8010105.

## AUTHORS BIOGRAPHY

**Dana Fatadilla Rabba**
He received his B.Eng. Degree in Engineering Physics from Universitas Gadjah Mada, Indonesia, in 2022. His research interests include control systems and artificial intelligence.

**Awang Noor Indra Wardana**
He received the Dr.-Ing. Degree in Embedded Systems from Kassel University, Germany. He is currently an assistant professor in the Department of Nuclear Engineering and Engineering Physics, Faculty of Engineering, Universitas Gadjah Mada. His research interests include software development in automation and control systems.

**Nazrul Effendy**
He received his Ph.D. Degree from Chulalongkorn University, Thailand, in 2009. He is currently an associate professor in the Department of Nuclear Engineering and Engineering Physics, Faculty of Engineering, Universitas Gadjah Mada. His research interests include artificial intelligence, machine learning, embedded systems, and their applications.