ELKHA : Jurnal Teknik Elektro, Vol. 15 No.2, October 2023, pp. 120 - 125
ISSN: 1858-1463 (print), 2580-6807 (online)

Multi-oscillations Detection for Process Variables Based on K-Nearest Neighbor

Muhammad R. Amrullah[1)], Awang N. I. Wardana[2*)], and Agus Arif[3)]

[1,2,3)] Department of Nuclear Engineering and Engineering Physics, Universitas Gadjah Mada, Indonesia
Corresponding Email: *) awang.wardana@ugm.ac.id

Abstract **- In the process industry, a control system is important to ensure the process runs smoothly and keeps the product under predetermined specifications. Oscillations in process variables can affect the decreasing profitability of the plant. It is important to detect the oscillation before it becomes a problem for profitability. Various methods have been developed; however, the methods still need to improve when implemented online for multi-oscillation. Therefore, this research uses a machine learning-based method with the K-Nearest Neighbour (KNN) algorithm to detect multi-oscillation in the control loop, and the detection methods are made to carry out online detection from real plants. The developed method simulated the Tennessee Eastman Process (TEP), and it used Python programming to create a KNN model and extract time series data into the frequency domain. The Message Queuing Telemetry Transport (MQTT) communication protocol has been used to implement as an online system. The result of the implementation showed that two KNN models were made with different window size variations to get the best performance model. The best model for multi-oscillation detection was obtained with an F1 score of 76% for detection.**

Keywords: K-nearest neighbor; machine learning; multi oscillation; process automation

I. INTRODUCTION

In the industrial sector, especially the process industry, a control system is important to ensure the process runs smoothly and keeps the product under predetermined specifications. However, the operation of control systems in the process industry will not always run well because of the disturbances indicated by the oscillations. Oscillation is a common problem in the industrial control loop where between 30% and 41% of the control loop in the industry has oscillated [1]. In the control process, oscillations were indicated by the time series data trend that did not meet a predetermined condition in the period.

The oscillation caused by external disturbance, aggressive controller, and sticking valve [2] can adversely affect processes such as increased energy consumption and material waste, low-quality products, and damage to the control instrument. In general, oscillation can affect the plant's profitability [3]. Therefore, it is necessary to handle oscillation when it occurs. The first step to handling oscillation is early detection when the oscillation occurs in a control loop [4].

The conventional method of oscillation detection is monitoring and analyzing the process variable (PV) data over a period with direct inspection at the control loops. Previous research has been conducted to develop a more efficient method for oscillation detection. The methods are generally based on calculating certain parameters and setting a threshold, then evaluating it with if/else commands using the computational algorithm to determine the oscillation within the process variable's data [1]. However, the methods still require a huge amount of time and are not so efficient since, in the industrial plant, there are hundreds or thousands of control loops [5]. For example, there is an application of the decomposition method using Empirical Mode Decomposition (EMD) and Fast Adaptive Chirp Mode Decomposition (FACMD) algorithm for oscillation detection in the control loop [6], [7]. Then, a combination of EMD and Delay Vector Variance (DVV) was developed for oscillation detection in chemical industries [8]. There is also a method based on the frequency domain property of time series data analysis using the Estimation of Signal Parameters via Rotational Invariance Technique (ESPRIT) that measures the frequency in data signal at Point of Common Coupling (PCC) to detect oscillation in control loops [9], [10].

The threshold-based method runs well in a regular oscillation with higher efficiency than the direct inspection method. Unfortunately, in the industrial process, some disturbances, such as intermittent and multi-oscillation, can occur in oscillation with non-regular amplitude or frequency. This non-regular oscillation has factors and parameters that make the calculation more complex. Thus, this method becomes too complex and hard to implement [1].

The machine learning-based method is another method for detecting oscillation in control loops is the machine learning-based method [11]. The machine learning-based method has various algorithms that have been implemented, such as the application of the Convolutional Neural Network (CNN) and Principal

Manuscript received 2023-08-02; revised 2023-09-26; accepted 2023-09-27

Component Analysis (PCA) algorithm for online detection [12]. Then, there is a combination of a Support Vector Machine (SVM) with a generalized statistical variable to detect oscillation caused by a stiction valve in the control loop [13]. Also, there is an improvisation Multilayer Feedforward Neural network for oscillation detection due to stiction in control valves [14]. However, these methods are implemented in a single data of control loops. The machine learning-based method that had been developed shows that the machine learning method is simpler than the threshold-based method.

This paper provides a new contribution by giving a simpler method based on machine-learning. This research aims to develop online machine learning-based methods that can be used to detect multi-oscillation that is caused by external disturbance, aggressive controller, and sticking valves in real-time control loop data. The method was developed using a K-Nearest Neighbor (KNN) algorithm integrated with MQTT protocol and sliding windows technique to provide continuous detection.

II. METHODOLOGY

The method proposed in this research will be based on machine learning using the KNN algorithm as part of a thesis regarding the application of KNN for detecting and diagnosing multi-oscillation [15]. KNN is used in this research because, with the optimum value of k neighbor, KNN becomes an effective algorithm for classification tasks with outlier and noised datasets. KNN is based on the distance metric like Euclidean distance to measure the similarity between a training sample and a testing sample, then identify the nearest neighbors of classified data according to the computed similarities, and finally decide the class label of the testing sample by the majority vote among the neighbors. This research used KNN to classify the process variable dataset with noised data. The KNN algorithm will be implemented in computing devices to process data and determine the presence of multi-oscillation.

To simulate that the methods can be implemented in the online system. The KNN-based detection methods device is connected to the Data Acquisition System to receive and process variable datasets online. The communication using Message Queuing Telemetry Transport (MQTT) communication protocol. The implementation concept is shown in Figure. 1. While MQTT is a protocol that uses the server-client concept [16], the KNN-based detection method devices become subscribers that receive data from a Data Acquisition System.

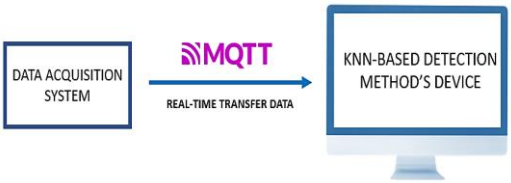

Figure 1. Implementation concept

Implementing the online detection method requires a fast and efficient signal analysis to process continuous incoming data [17]. The incoming continuous data does not always have an oscillation within it. The incoming data must be segmented to replace the old data with the new data in the optimum fix-sized data length to reduce the computing load and get the high-efficiency execution program. Thus, the sliding windows technique determines the data that will be analyzed. The sliding window technique will set a "window" with a certain data length, and when the new incoming data arrives, the initial data will be deleted to insert new data.

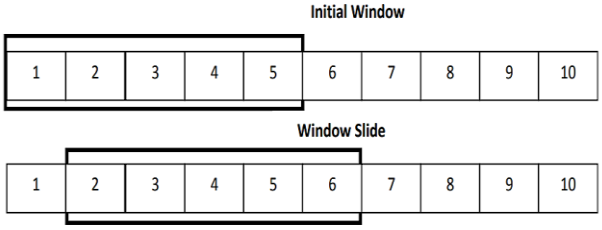

Figure 2. Sliding window process [19]

This process will run continuously, like a window that slides among data visualized in Figure. 2. The sliding window is one widely used technique to analyze and segment continuous data [18]. Therefore, this research will determine the window size for a more efficient method. Three kinds of window sizes are used, 100, 150, and 200 lengths of data, which found that the three optimal window sizes were used to obtain the best-performing model by varying the window size from 0 to 200 [12]. One of the three size windows will be chosen based on the result of the F1 score for each size in the evaluation phase. This research was carried out in four main stages, shown in Figure 3.

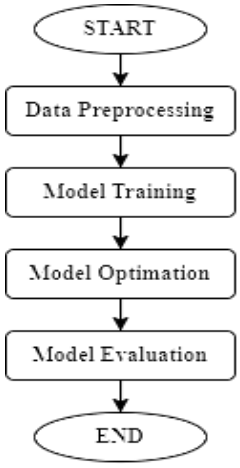

Figure 3. Methods [18]

Two categories of data are used to build the KNN model for multi-oscillation detection. The first category is non-multi oscillation data, PV data under normal operation, and data with a single oscillation. The second category is multi-oscillation data, which can be divided into three types of multi-oscillation, as shown in Table 1.

Table 1. Type of Training Data

Label	Oscillation Source
Non-Multi oscillation	Normal
	Sticking Valve
	Poor-tunning controller
	External Disturbance
Multi-oscillation Type I	Sticking Valve and Poor-tunning controller
Multi-oscillation Type II	Sticking Valve and External Disturbance
Multi-oscillation Type III	External Disturbance and Poor-tunning controller

The data used to create the KNN model is obtained from the Tennessee Eastman Process (TEP) simulation. TEP is a simulation program developed by the Eastman Chemical Company that provides a natural process of an industry for evaluating control monitoring techniques and methods. The simulation process in TEP is condition-based, the actual chemical process by which the components, kinetics, and operating conditions have been modified. The TEP dataset is a benchmark for comparing different anomaly detection approaches [20]. Therefore, in this research, TEP will provide generated artificial data representing industrial process data in a few setting conditions. The data is then mapped into two segments: training and testing data.

Training data is temperature data inside the reactor unit in TEP that is used to build the KNN model. The temperature inside the reactor is controlled by adjusting the reactor coolant flow with the control loop shown in Figure 4. Training data is retrieved when the process is under normal conditions and when a single oscillation occurs with three different error sources. Two categories of data are used to build the KNN model for multi-oscillation detection. The first category is non-multi oscillation data, PV data under normal operation, and data with a single oscillation. The second category is multi-oscillation data, which can be divided into three (3) types of multi-oscillation, as shown in Table 1.

The data used to create the KNN model is obtained from the Tennessee Eastman Process (TEP) simulation. TEP is a simulation program developed by the Eastman Chemical Company that provides a natural process of an industry for evaluating control monitoring techniques and methods. The simulation process in TEP is condition-based, the actual chemical process by which the components, kinetics, and operating conditions have been modified. The TEP dataset is a benchmark for comparing different anomaly detection approaches [20]. Therefore, in this research, TEP will provide generated artificial data representing industrial process data in a few setting conditions.

Training data is temperature data inside the reactor unit in TEP that is used to build the KNN model. The temperature inside the reactor is controlled by adjusting the reactor coolant flow with the control loop shown in Figure 4. Training data is retrieved when the process is under normal conditions and when a single oscillation occurs with three different error sources.

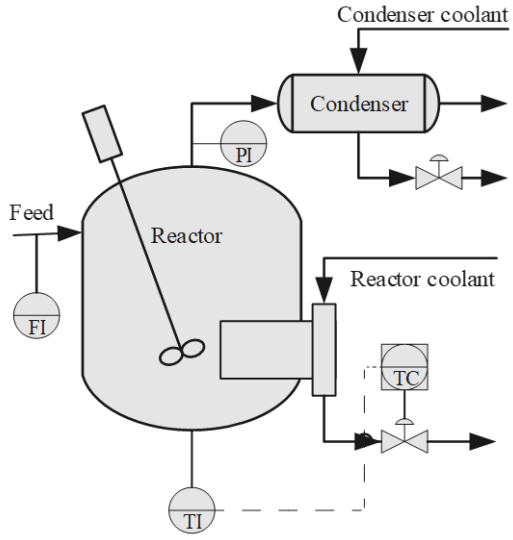

Figure 4. TEP reactor unit control system [20]

Training data retrieval is carried out during the simulation until the data reaches a predetermined length. The results of retrieving training data with a range of training data for each label are shown in Table 1. The multi-oscillation data consist of two single oscillation that occurs simultaneously. However, multi-oscillation data with more than two oscillation sources will be ignored because two sources are enough to represent a multi-oscillation.

The training data obtained is then processed at the initial stage: data segmentation, normalization, extraction, and labeling. Data segmentation is the division of process variable time series data (PV data) based on the size of a specified window. Each control block on the training data is divided into subsets with a length corresponding to the window size. The segmentation process is shown in Figure 5. The training data is divided into 200 subsets according to each window size.

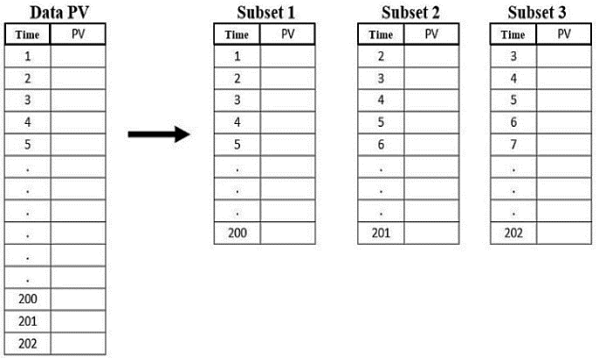

Figure 5. Data segmentation [15]

Frequent domain features are extracted from data using the frequency feature extraction library in a time series already available in Python [21]. The extraction of frequency features from time series data is then divided into three domains: temporal, statistical, and spectral. In

this study, extraction is configured in a spectral domain that will produce frequency domain features from the data. Extraction is performed on each training data subset, and each subset's extraction results will be unified into a new dataset with a frequency domain feature. The feature extraction process is shown in Figure. 6. Feature selection is performed to get features with a strong influence on model detection results while reducing the computational load of detection. Features strongly impacting the model are features with a high-value correlation coefficient.

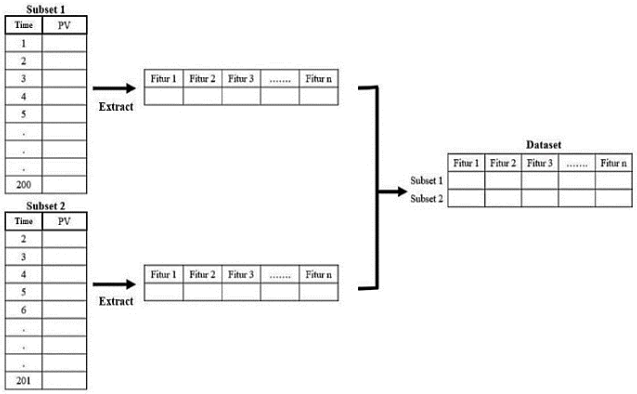

Figure 6. Data extraction process [15]

Segmentation is performed on the train data based on the specified window size. Three window sizes are to be tested, with sizes 100, 150, and 200. The segmentation process is carried out as in Figure 3 with a subset size according to the size of the test window. Segmentation was performed on four types of data from the training data in Table I, and each label was segmented into 200 subsets, resulting in a total of 800 subsets for each test window. The data in the subset is then extracted using the frequency feature extraction library in the time series to derive the frequency domain feature from the data. The extraction result from each subset will result in one row of data with columns containing the values of the frequency domain data feature. The extraction results from all subsets are then reunited in one new dataset. The extraction results are then normalized before being used to train the model. Based on this, training data is obtained, with details shown in Table 2.

Table 2. Training Data After Extraction Process

Label	Total Data
Non-Multi Oscillation	4 x 200 data
Multi-oscillation Type I	1 x 200 data
Multi-oscillation Type II	1 x 200 data
Multi-oscillation Type III	1 x 200 data

Depending on the desired conditions, the test data is taken from several control cards. Non-multi-oscillation test data is the PV data in the TE process unit under normal conditions. Meanwhile, multi-oscillation test data is data from several units in the TE process according to the cause of the multi-oscillation type. 11 non-multi oscillation condition data and 5 data for each kind of multi-oscillation were taken for the test data. Therefore, in the test data, a total of 26 data were obtained. The details of the test data are shown in Table 3, each process variable containing 200 sampled data. The testing data was obtained with different process variables from training data but with similar oscillation conditions. It will deliver the model's performance in detecting multi-oscillation in other units with additional variable and shows its performance in the new environment. The testing data represent the multi-oscillation and non-multi-oscillation data with various oscillation sources. Therefore, the version of the machine learning model can describe how the model performs in a different environment.

Table 3. Testing Data

Label	Total Data
Non-Multi Oscillation	11 x 200 data
Multi-oscillation Type I	5 x 200 data
Multi-oscillation Type II	5 x 200 data
Multi-oscillation Type III	5 x 200 data

The detection model is trained using two classes of data: non-multi oscillation data labeled 0 and multi-oscillation data labeled 1. The model training stages are performed using the Python programming language. The machine learning model uses the KNN algorithm. The model is then trained using the dataset prepared in Table 2. Three window sizes are used in each model type: 100, 150, and 200 data lengths.

The next stage is an optimization process that is carried out by determining the optimal hyperparameter value for the model. In the KNN model used, the hyperparameter that can be optimized is the Value of k in the neighbor parameter. Optimization is then performed by varying the k value from 1 to 10 to get the best-performing model [22].

The trained model is then tested using test data to obtain a sliding window size with the best machine-learning performance parameters at this evaluation stage. The performance parameter is accuracy in the form of F1 values through confusion matrix analysis. The F1 score measures a model's accuracy on a dataset. It evaluates binary classification systems, classifying examples as 'positive' or 'negative'. The confusion matrix is visualized in a table where each row refers to the actual class recorded in the test data.

Each column refers to the class of prediction results by the classification model. In the confusion matrix, a True Positive (TP) will be obtained where the positive label value is predicted correctly, False Positive (PS) where the negative label value is predicted as a positive label, True Negative (TN) where the negative label is predicted correctly, and False Negative (FN) where the positive label value is predicted as a negative label, as shown in Figure 7.

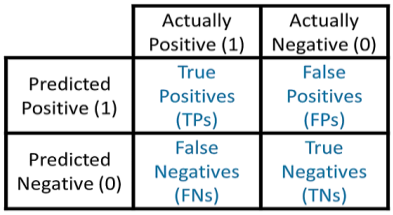

Figure 7. Confusion matrix

From Figure 7, precision and recall values are obtained, where accuracy is obtained from the ratio of correct positive data to all samples predicted as positive. Meanwhile, recall is obtained from the ratio of expected positive data to all data samples labeled positive. Precision indicates the reliability of the machine learning model in classifying the model as positive, and recall measures the model's ability to detect positive samples. The following equation uses the precision and recall values to obtain the F1 value. The optimal KNN model is then implemented in the Python program using the MQTT protocol, which provides the detection process to be carried out online. This method is set to update the data within the sliding window and give the detection result every second.

III. Results and Discussion

Machine learning model training is created using the Python programming language. The model is then trained using the data processed in the data preparation stage. At this stage of the training model, the initial k-values parameter of 5 was used. Further optimizations were made to the models that had been made to get the best k-value parameter for each model. The optimization results in the form of the best k value for the detection sub-program are shown in Table 4.

Table 4. Optimization Result or K-Value Parameter On Various Sliding Windows

Sliding Window Length	Best Value of k
100	3
150	15
200	5

The best k parameter obtained from the optimization stage will recreate the initial program by entering the k value obtained. The optimization result model is then evaluated to determine the model with the best performance. The evaluation is done by testing the optimization result model using test data, as shown in Table 5.

Table 5. Confusion Matrix for Three Sizes of Window

Sliding windows length	k	True Positive	True Negative	False Positive	False Negative
100	3	13	6	5	2
150	15	14	3	8	1
200	5	13	7	2	4

The results obtained in Table 5 were then calculated to get the accuracy for detecting multiple oscillations. This level of accuracy is obtained from the F1 score of each model shown in Table 6.

Table 6. Confusion Matrix for Three Sizes of Window

Sliding windows length	k	F1 Score
100	3	71%
150	15	58 %
200	5	76%

Based on the evaluation results, the best-performing model is a machine learning model with a KNN algorithm with a parameter k of 5 and a sliding window size of 200 data. The model is then implemented online by creating an online detection program based on Python programming. Process variable data is sent from the TE process to the MQTT broker. The detection program will then retrieve the data at the broker for later detection. The detection program will display the time variable, the process variable Value (sensor measurement results), and the detection results. The detection program created is then reevaluated using test data to determine if there is a change in performance. The evaluation results show that the online program has a detection performance with an unchanged F1 value of 76%. It means that this method can detect 76% of multi-oscillation data correctly in the total of multi-oscillation data that has been inputted.

The developed method in this research has an F1 score of 76%, which is higher than other works, such as using the Deep Feedforward Network (DFN) algorithm with an accuracy of 70 % for sensor noised data [1]. The method has a higher chance of detecting multi-oscillation correctly. This method also updated incoming data within sliding windows and showed the detection result in setting time conditions.

IV. Conclusion

This research developed the multi-oscillation detection based on the machine learning KNN algorithm. It implemented online detection using the MQTT protocol. The result showed that the developed method performs well based on the F1 score parameter of the implementation that can be reached until 76%. It also shows that it can be implemented for online multi-oscillation detection in the industrial process. For further development, this developed method will be combined with intermittent and single oscillation data to complete the oscillation method for process variables in control loops.

References

[1] J. W. V. Dambros, J. O. Trierweiler, M. Farenzena, and M. Kloft, "Oscillation Detection in Process Industries by a Machine Learning Approach," Industrial & Engineering Chemistry Research, vol. 58, no. 31, pp. 14180-14192, 2019.

[2] M. Yagci and J. M. Boling, ”Reinforcing Hurst Exponent with Oscillation Detection for Control Performance Analysis: An Industrial Application,” IFAC PapersOnLine, vol. 55, no. 6, pp. 772-777, 2022.

[3] 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, 2019.

[4] J. W. V. Dambros, M. Farenzena and J. O. Trierweiler, ”Oscillation Detection and Diagnosis in Process Industries by Pattern Recognition Technique,” IFAC-PapersOnLine, vol. 52, no 1, pp. 299-304, 2019.

[5] S. Sharma, V. Kumar and K.P.S.Rana, ”Automatic Oscillations Detection and Quantification In Process Control Loops Using Linear Predictive Coding,” Engineering Science and Technology, an International Journal, vol. 23, pp. 123-143, 2020.

[6] Q. Chen, J. Chen, X. Lang, L. Xie, S. Lu and H. Su, ”Detection and Diagnosis of Oscillations in Process Control by Fast Adaptive Chirp Mode Decomposition,” Control Engineering Practice, vol. 97, pp. 1-26, 2020.

[7] M. Gurtner, P. Zips, M. Atak, J. Ophey and A. Kugi, ”Improved EMD-based Oscillation Detection for Mechatronic Closed-Loop Systems,” IFAC-PapersOnLine, vol. 52, no. 15, pp. 370-375, 2019.

[8] M. F. Aftab, M. Hovd and S. Sivalingam, ”Diagnosis of Plant-Wide Oscillations by Combining Multivariate Empirical Mode Decomposition and Delay Vector Variance,” Journal of Process Control, vol. 83, no. 1, pp. 177-186, 2019.

[9] A. Kumar, R. K. Panda, A. Mohapatra, S. N. Singh and S. C. Srivastava, ”Mode of Oscillation Based Islanding Detection of Inverter Interfaced DG Using ESPRIT,” Electric Power Systems Research, vol. 200, pp. 1-9, 186, 2021.

[10] M. Luan, S. Li, D. Gan and D. Wu, ”Frequency domain approaches to locate forced oscillation source to control device,” International Journal of Electrical Power and Energy System, vol. 117, pp. 1-21, 2020.

[11] J. Wang and C. Zhao, ”Variants of Slow Feature Analysis Framework For Automatic Detection and Isolation of Multiple Oscillations in Coupled Control Loops,” Computers & Chemical Engineering, vol. 141, pp. 1-19, 2020.

[12] Y. Henry, C. Aldrich and H. Zabiri, ”Detection and Severity Identification of Control Valve Stiction in Industrial Loops Using Integrated Partially Retrained CNN-PCA Frameworks,” Chemometrics and Intelligent Laboratory Systems, vol. 206, pp. 1-14, 2020.

[13] Y. A. Yazdi, H. T. Shandiz and H. G. Narm, ”Stiction Detection in Control Valves Using a Support Vector Machine With a Generalized Statistical Variable,” ISA Transactions, vol. 126, pp. 407-414, 2022.

[14] A. A. A. M. Amiruddin, H. Zabiri, S. S. Jeremiah, W. K. The and B. Kamaruddin, ”Valve Stiction Detection Through Improved Pattern Recognition Using Neural Networks,” Control Engineering Practice, vol. 90, no. 1, pp. 63-84, 2019.

[15] M. R. Amrullah, “Application of K-Nearest Neighbor Algorithm as Online Multioscillation Detection And Diagnosis Method in Tennessee Eastman Process Control Loop,” Bachelor Thesis, Universitas Gadjah Mada, Yogyakarta, 2022.

[16] H. Kurdi and V. Thayananthan, ”Authentication mechanisms for IoT system based on distributed MQTT brokers: review and challenges,” Procedia Computer Science, vol. 194, no.1, pp. 132-139, 2021.

[17] J. P. Salameh, S. Cauet, E. Etien, A. Sakout and L. Rambault, ”A new modified sliding window empirical mode decomposition technique for signal carrier and harmonic separation in non-stationary signals: Application to wind turbines,” ISA Transaction, vol. 89, no.1, pp. 20-30, 2019.

[18] J. Xie, K. Hu, G. Li and Y. Guo, ”CNN-based driving maneuver classification using multi-sliding window fusion,” Expert Systems with Applications, vol. 169, pp. 1-9, 2021.

[19] H. S. Hota, R. Handa and A. K. Shrivas, ”Time Series Data Prediction Using Sliding Window Based RBF Neural Network,” International Journal of Computational Intelligence Research, vol. 13, no. 5, pp. 1145-1156, 2017.

[20] l. Lomov, M. Lyubimov, I. Makarov and L. E. Zhukov, ”Fault Detection in Tennessee Eastman Process With Temporal Deep Learning Models,” Journal of Industrial Information Integration, vol. 23, pp. 1-15, 2021.

[21] M. Barandas, D. Folgado, L. Fernandes, S. Santos, M. Abreu, P. Bota, H. Liu, T. Schultz and H. Gamboa, ”TSFEL: Time Series Feature Extraction Library,” SoftwareX, vol. 11, pp. 1-7, 2020.

[22] G. S. K. Ranjan, A. K. Verma and S. Radhika, ”K-Nearest Neighbors and Grid Search CV Based Real Time Fault Monitoring System for Industries,” in IEEE 5th International Conference for Convergence in Technology (I2CT), pp. 1-5, Bombay, India., 2019.