Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 1, No. 3, January : 2024

AN AI-DRIVEN FAULT IDENTIFICATION APPROACH FOR INDUSTRIAL INTERNET OF THINGS

Sourav Kumar Bhoi[1], Kalyan Kumar Jena[1*], Ashish Kumar Mohanty[1], Monalisa Kar[1]
[1]Department of Computer Science and Engineering, Parala Maharaja Engineering College, Berhampur-761003, India Email: sourav.cse@pmec.ac.in, kalyan.cse@pmec.ac.in[*], ashishmohanty@outlook.com, monalisakar99@gmail.com

Abstract:
Industrial fault detection is an integral part of any industrial setting. Any industrial environment possesses potential threats that may be life-threatening such as electrical, mechanical, chemical, biological, ergonomically hazards, etc. When these faults are not detected that can cause great damage to life and property. To ensure safety, early detection is required to prevent any mishaps and do the needful before anything goes wrong. In this work, an AI-Enabled fault detection approach is proposed for the detection of faults in the Industrial Internet of Things (IIoT) systems. This approach is focused on a hybrid method (voting classifier) which stacks the support vector machine (SVM), decision tree (DT) and logistic regression (LR) methods for improving fault detection process at an AI-Enabled cloud server for detecting the faults. The proposed fault detection approach helps in fault detection process by collecting the sensor data from several IoT devices embedded in various machine parts and analysing these data using the supervised machine learning (ML) methods. The proposed method is compared with various supervised ML methods such as K-nearest neighbour (KNN), SVM, and DT. The performance of these methods is evaluated using the performance parameters such as accuracy, precision, recall, F1 score. The proposed hybrid method is able to perform better as compared to the other methods. This work is carried out using Python 2.7 and Jupyter notebook.
Keywords: Fault Detection, Supervised Machine Learning, Industrial Internet of Things, Voting Classifier

1. Introduction
IoT plays an important role in industries using new technologies and methods[1-27]. It is used to monitor, manage and actuate machinery and equipment from the click of a button or even automatically by the use of artificial intelligence. The use of IoT sensors to detect heat, pressure and other metrics lead us to know about the health state of machinery, making us able to judge when to do the needful maintenance for the longevity of machinery and ensuring safety. However achieving this is a challenge in the real world. Collecting and analysing such huge amounts of data in a less amount of time is the need of the hour for taking quick decisions. Also, ensuring synchronisation and quality of the data is a challenge. But evolving technologies are making us enable to do such things efficiently via a wide range of IoT devices and the cloud aiding it for the storage, analysis and synchronisation processes.
A fault is any error or condition which leads to the malfunctioning of any device which interprets the conditions wrongly producing misleading data. Fault detection is crucial and critical to any prevailing industrial operations in the present-day scenario industries and factories especially in the places where the conditions are harsh and equipment are critical to the functioning and production of the factory. Any fault, temporary or permanent, may lead to degradation of quality result in substandard products and may even cause loss to life and property, which may be severe. This all will lead to a stagnant condition, overcoming which will take years and fortunes to come.
Avoiding hazards from becoming disasters is the aim. Achieving it will require adequate safety measures to be taken, awareness programs should be run and governmental regular checks should be there to avoid any irregularities. But monitoring and detecting irregularities in such a huge premise is a challenge. Tragedies like the Bhopal Gas Leak in 1984, the Rourkela Blast Furnace accident in 1985, the Panipat Ammonia leak in 1992 and so on caused great miseries in the lives of many people. Proper monitoring and management at that time could have prevented these accidents from becoming

disasters. With the advent of new computational devices and techniques, monitoring, management, prevention, of faults is pretty viable and feasible, also in real time. IoT devices make it possible to collect huge amounts of data from a wide range of heterogeneous machines and equipment. Modern machine learning techniques make it possible to process such huge data with great precision and accuracy even using commodity hardware.
In this work, a dataset is taken from the source [23] to process and analyse conditions which led to a fault in the condition of an air compressor. First, data is analysed for any missing values in the pre-processing stage and checked for missing values and scaled so that calculations can be done on a normally distributed data. Then ML models such as Decision Trees, SVM and KNN can be used to classify the class labels. A hybrid model comprising of Logistic Regression, SVM and Decision Tree is used to improve the results using various partitions of the dataset. The performance is evaluated using metrics such as accuracy, precision, F1 score and recall.
The main contributions of this work are stated as follows.

- In this work, an AI-Enabled fault detection approach is proposed for the detection of faults in the IIoT systems.
- This approach is focused on a voting classifier which stacks the SVM, DT and LR methods for improving fault detection process at an AI-Enabled cloud server for detecting the faults.
- The proposed fault detection approach helps in the fault detection process by collecting the sensor data from several IIoT devices embedded in various machine parts.
- The proposed method is compared with the existing methods such as KNN, SVM, and DT.
- The performance of these methods is evaluated using the performance parameters such as accuracy, precision, recall, F1 score. This work is carried out using Python and Jupyter notebook.

The rest of the sections are organized as follows. Section 2 and Section 3 describes the related works and methodology of the work respectively. Section 4 describes the results and discussion. Finally, conclusion of the work is described in Section 5.

2. Related Works

Different works have been carried out by several researchers related to the fault detection mechanisms [1-27]. Some works are discussed as follows. Lo et al. [1] proposed that machines can be self-maintainable with the help of Artificial Intelligence (AI) with the advent of complex systems and low production costs and reconfigurable systems. Wen Sun et al. [2] proposed that industrial operations can be done intelligently with the aid of Artificial Intelligence (AI). Edge computing provides heterogeneity in it. Mokhtari et al. [3] proposed the use of supervised ML algorithms on measurement data in SCADA system is very promising and can even detect abnormal activities. A. Angelopoulos et al. [4] proposed the introduction of the IoT in industrial setting for getting manufacturing data collection in real-time and analysis brings a plethora of opportunities for fault detection, prediction, and prevention with machine learning techniques. Soldatos et al. [5] proposed that machine learning methods can be used for monitoring, prediction and management in the era of Industry 4.0 using heterogeneous IoT devices to predict any faults. Sarita et al. [6] proposed that principal component analysis has been used to check the health monitoring and remaining useful life estimation of electrical equipment connected to the grid using various parameters. Fernandes et al. [7] proposed that the development of fault detection mechanism of a predictive maintenance system is metallurgic industry. It tends to make an early prediction to prevent any mishaps.
Dutta et al. [8] proposed that in this work various machine learning algorithms has been used to find the odd behaviour in the pumps with the aid of predictive analysis. So that it can increase the lifespan of the pumping model. Kotsiantis et al. [9] proposed that objective of this work is to describe all the various supervised machine learning classification techniques. It describes various machine learning models. Choudhary et al. [10] focused on a general comparison among various state-of-the-art supervised machine learning algorithms. It tends to classify various classes according to the given data. Liu et al. [11] proposed that fault detection can be done in an industry by using different types of

sensors by monitoring and analysing data using machine learning and statistics. Hooman et al. [12] proposed that industrial development project based on the experiences that use tool for a commercial for formal modelling, compositional verification, modelling and code generation. In this work tool functionality is provided so that formal techniques have been introduced to detect the faults during the requirement phase and earlier design. Sousa et al. [13] proposed that IoT devices are being used in industrial systems to increase the productivity as well as the management. In this work, intelligent systems have been used to identify faults in electric generators of the wind turbines to improve the routines.
Seabra et al. [14] proposed that lost cost system, the intelligent system has been used for monitors the behaviour of electrical magnitude of appliances. The system has been used to analyse the store data, faults in the system, directly report to the user. Murugan et al. [15] proposed a system which helps to find the exact location of the fault and determines the fault using an Arduino microcontroller.it also detects the leakage current using a relay driver to avoid faults occurring repeatedly. Abdellafou et al. [16] proposed a whole new machine learning method for fault detection using a reduced kernel partial least square in static and online forms, for handling nonlinear dynamic systems. Jan et al. [17] proposed a sensor-fault detection and diagnosis system for the IoT devices so that they can build a fault tolerant industrial environment. Wu et al. [18] proposed machine learning methods for enhancement in boiler efficiency and reduction in pollutant emission as well as to manage Activities such as scheduled maintenance. F De Vita et al. [19] proposed A Semi-Supervised Bayesian Anomaly Detection Technique for Diagnosing Faults in Industrial IoT Systems. Derakhshan et al. [20] proposed methods for outlier detection in industrial. Its main objective is to detect earlier symptoms and isolate them for safety. From the above discussion, it is observed that detection of fault in the industrial IoT systems accurately is a challenging task in the current scenarios. So, there is a need for the development of enhanced methods for the detection of faults in a better way.

3. Methodology

In this section, we have mainly described the network model, proposed fault detection approach, best machine learning model selection, and dataset overview.

3.1 Network Model and Assumptions

Network model describes about the main building blocks of the network. The main building blocks are IIoT devices, Local Server (LS), Base Station (BS), Cloud Gateway (CG), and AI-Enabled Cloud Server. In this network, n number of IIoT devices are set in an Industry for any particular application of monitoring. The IIoT devices are connected with several necessary s number of sensors as per the application. IIoT devices has enough power supply, computing capability and memory for storage. It has wireless/wired capability to send and receive data using transmitter and receiver respectively. Local Server is a server which has enough power supply, computing and storage capability. It has Internet facility. It can send/receive data from the IIoT devices and also from the BS. BS is responsible for forwarding data to the AI-Enabled Cloud Server/LS and can also receive data from cloud server and LS. CG is the gateway for routing the packets from BS to cloud server or cloud server to BS. AI-enabled cloud server is the server which gives services to the user. In this work, fault detection service is provided by the cloud server. It is made AI-enabled by training the cloud server with a intelligent machine learning model for predicting the fault status of the IIoT devices from the behaviours of it. The cloud server has enough storage and computing capability. It has high speed Internet facility for providing quality of services to the end devices. Fig. 1 shows the network model of the proposed approach.

3.2 Proposed AI-Enabled Fault Detection Approach

In this work, the IIoT based system is focused. The proposed IIoT architecture is mentioned in Fig. 1. There are n number of IoT devices having several sensors which can communicate with local server. The IoT devices will send the readings/behaviour (data) to LS. The local server will send the data to BS nearest to it. From BS the data will be send to the GT. The GT will send it to the AI-enabled cloud server. In this cloud server the data will be analysed using a machine intelligence based model such as KNN, SVM, DT and voting classifier (hybrid model). The best model which is selected among these models is used as a fault status prediction model in the cloud server. In this work, the accuracy of these models are considered for selecting the best model. After processing of data by the selected model, the target label (TL) status will be updated as faulty or non-faulty in the cloud server for the IIoT device. From the cloud server, the TL status will be sent to the GT. From the GT, the TL status will be sent to the LS. In the LS, the TL status will be updated in the local server. Algorithm 1 described the whole process. The flow of work is also mentioned in Fig. 2.

For analysis of the model, we have proposed two lemmas to show the computational complexity of the proposed approach using time complexity and message complexity. The two lemmas are shown in Lemma 1 and Lemma 2 with the proofs of them.

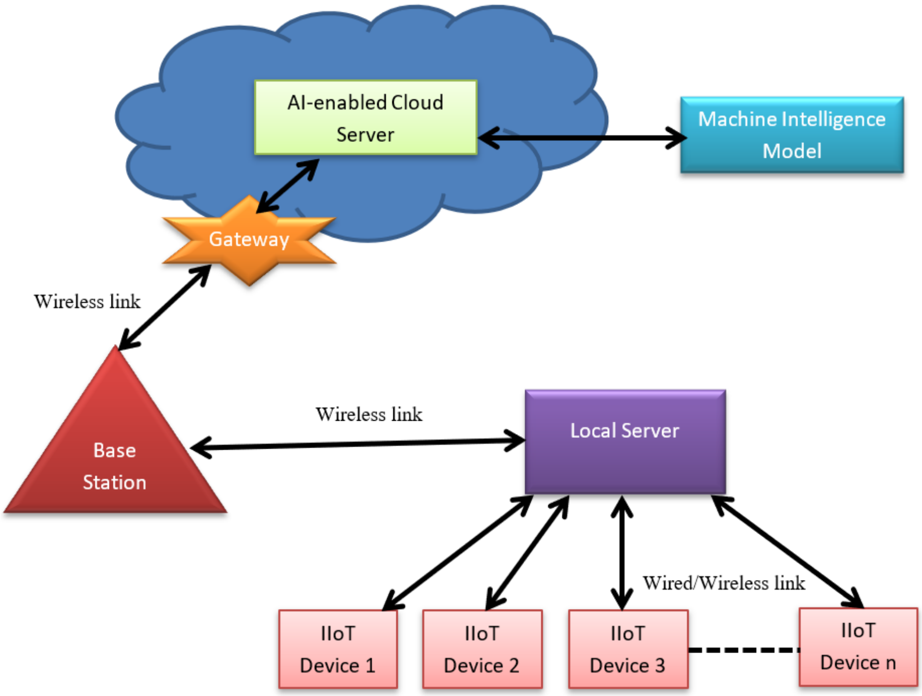


Fig. 1 Proposed IIoT Architecture

Algorithm 1: Algorithm for AI-enabled fault detection approach for IIoT systems

Input: Sensor Data

Output: Update at Local Server

```
For i = 1 to n (n represents the number of IIoT devices in the network)
    IoT device i sends readings/behavior to Local Server;
    Save (data) at local server;
    Local server sends data to Base station;
    Base station forwards data to Gateway;
    Gateway forward data to AI-Enabled cloud server;
    TL=Best_Machine_Intelligence_Model(data); //faulty or non-faulty as TL
    Update (TL) status in cloud server;
    AI-enabled cloud server sends TL status to BS; //through GT
    GT sends TL to LS;
    BS sends TL to LS;
    Update (TL) status in LS;
End For
```

Lemma 1: The time complexity of detection of IIoT devices in the network as faulty or non-faulty is O(NM).

Proof: Let, there are N number of IIoT devices sending sensor readings as {s1,s2,…,sK}. The readings are then sent to LS. The LS forwards it to AI-enabled cloud server. AI-enabled cloud server uses the best ML model selected to predict the target label TL as faulty or non-faulty by using the sensor readings. The number of steps for detecting the fault status is same for all the IIoT devices. If M steps is there then N*M steps are needed for predicting the fault status of all IIoT devices. So, the time complexity of the network for detection of fault is O(NM).

Lemma 2: The message complexity for detection of IIoT devices in the network as faulty or non-faulty is O(NP).

Proof: Let, there are N number of IIoT devices with sending sensor readings as P_i={$IIoT_i$, s1,s2,…,sK}, where P_i is the single unit message for IIoT device i and K is the number of sensor readings. The P is then sent to LS. The LS forwards it to AI-enabled cloud server. AI-enabled cloud server uses the best ML model selected to predict the target label TL as faulty or non-faulty by using the P_i message. The number of messages required for N devices are N*P. So, the time complexity for detection of faulty or non-faulty devices in the network is O(NP).

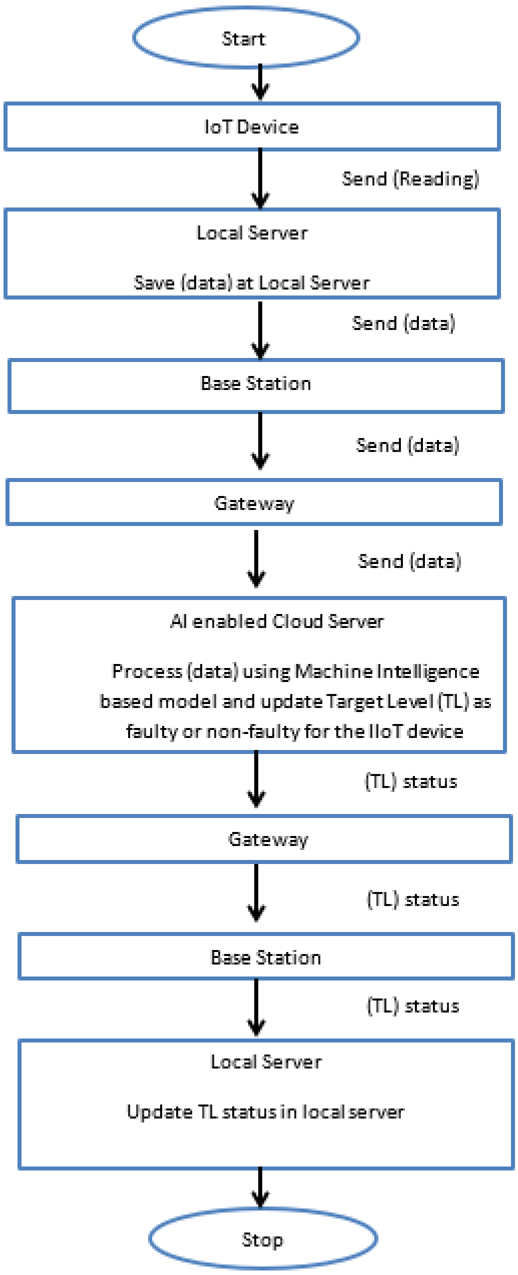


Fig. 2 Work flow diagram of proposed model

3.3 Selection of Best Machine Learning Model

The selection mechanism of machine intelligence-based model to make the cloud server AI-enabled is described in Fig. 3. A standard dataset is considered for training the models. Before that, preprocessing operation is performed to get the structured data. These data will be processed by the models such as KNN, SVM, DT and voting classifier (hybrid model). Afterwards, the accuracy of each model will be analysed and the higher accuracy model will be selected for processing the data (sensor reading/behaviour) in the cloud sever. The KNN, SVM, DT and voting classifier models are described as follows.

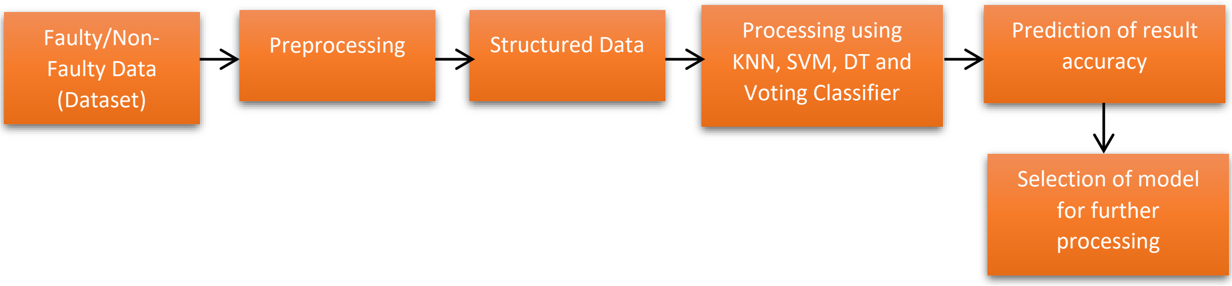


Fig. 3 Selection mechanism for ML based model

3.3.1 KNN

KNN belongs to a class of classification and pattern recognition algorithms. It was proposed by Fix and Hodges in the year 1951. These algorithms are used for are used to predict the discrete values such as true or false, male or women, emails as spam or not. Unlike most algorithms, it is a non-parametric classification algorithm that means it uses neighbour points information to predict the class of the target using Euclidean distance (d) in 2-D space as mentioned in equation (1).

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \qquad (1)$$

The selection of K-value is a challenging task. If chosen a small integer, it will highly influence the results.If a large value of K is chosen the computation will be quite highly intensive and results will be very diminishing. So, approximations can be done.Cross validation can be done on multiple values of K for choosing the optimal model. There have been many uses of the KNN algorithms in many areas of life such as classifying diseases and tumors from X-rays, identifying faults in industrial environments and localisation of facial features and detection. In this work,KNN is used extensively for categorising sensor data in a healthy and non- healthy states.This algorithm produces very promising results for such an unstructured dataset. Hence, KNN is a simple algorithm and executes quickly for small datasets.It works well even on the unstructured data.But takes a lot of space for large datasets and is computationally intensive for finding distances of respective feature points.

3.3.2 SVM

SVM was given by Vapnik and Alexey Ya Chervonenkis. Later to model real life scenarios, non-linearity was introduced using kernel functions. These classes of algorithms are used for classification that means they are used to predict the binary values such as true or false, male or women, emails as spam or not, faulty or not faulty in our case. More formally a support vector machine creates a hyperplane to separate linearly separable points and according to that, it gives the prediction result. The mathematical equation of the hyperplane is mentioned in equation (2) where W represents normal vector, X represents data points and b represents the linear coefficient.

$$W^t X - b = 0 \qquad (2)$$

In addition to perform linear classifier SVM can perform non-linear classification very efficiently by using kernel functions like polynomials, sigmoid function or the most commonly used for complex data is the Gaussian Radial basis Function (RBF). For real world problems, it can go up to several dimensions.

SVM finds out the best hyperplane that divides the classes. SVM takes all the data points and it draws a line that is called hyperplane .Hyperplane has been used to divide the classes and this line is termed

as decision boundary. SVM does it by using support vectors which are the points closest to the line from classes. Then we compute the distance between the line and support vector. This distance is called margin.The optimal hyperplane is found using the maximum margin.
Selecting the hyperplane is a challenging task. If chosen hyperplane with low marginal distance then, it will give a lot more complications in real world problems while classifying the values. If a large value of the marginal distance is chosen the computation will be quite highly intensive and results will be very accurate while giving the predictions. So approximations cannot be done and it is required to take the maximum marginal distanced hyperplane for best results. For cross validation, we can take different hyperplanes and validate the fact of the maximum marginal distance theory.
SVM can be used in several areas. Some of these are described as follows.

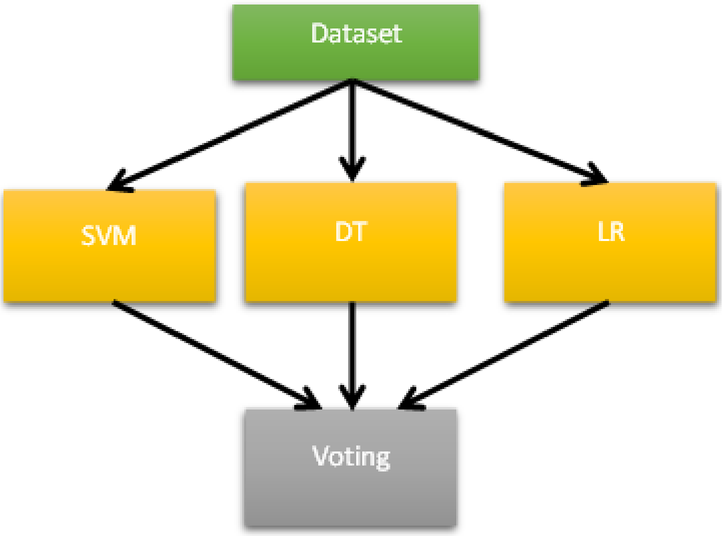
- Face detection: It uses the parts of the image and creates a feature plane and identifies a person's face by localising by a boundary box.
- Bioinformatics: It is used for protein classification based on the genes in the Deoxyribonucleic acid (DNA) to identify any inherent features
- Hand-writing Recognition: It is used to classify localised boundary boxes in image parts to valid alphabet, numbers or special characters.It is even used for the validation of the signatures.
- Text categorisation: It uses the contents of the documents to classify it into web pages,news, articles.

3.3.3 DT
A DT is ideal for visualizing the decisions and their consequences and costs related to them. A DT is an upside down tree with its root at its top. Traditional techniques use a single decision boundary, but where multiple boundaries are there and the data set is complex, DT can be used for the classification task. Basically, a DT consists of the following components.
Root Node: It is the node present at the beginning of the tree. From here the tree starts diving itself based on various features.
Decision nodes: These are the nodes that we get after splitting the root node.
Leaf Nodes: Nodes after which further splitting is not possible
A DT is formed by selecting the feature that is important and their relationships can be viewed easily, first in the root node and then it is done in a recursive manner. The recursion becomes complete when partitioning cannot further proceed or it doesn't add any value to predictions. This algorithm doesn't need any domain knowledge and hence is appropriate for exploratory knowledge discovery. It can operate on very high dimensional data. For the partition to be efficient to be good the entropy i.e., the measure of randomness should be less and information gain should be very high.

- Prediction: Using previous data and recognising patterns to predict the future.
- Manipulating values: Decision trees can be used to merge categorical features into manageable numbers.

DT is quite favourable because of its subtle advantages that means it is able to generate understandable rules, perform classifications without intensive computation and are able to handle both continuous and categorical value. This method is not suitable so because it is prone to errors with a small sample size.
3.3.4 Hybrid Method (Voting Classifier)
Voting Classifier is a model which hybridises many models into one model and predicts the output classification based on probabilistic analysis of the votes of the candidate models and produces the optimal results. It is represented in Fig. 4. Rather than getting results from different single models it hybridises, which increases its reliance since knowledge of the herd always gives the optimal results according to game theory. It can be classified into two types which are described as follows.
Hard Voting: Here the voting is based on majority that means the classified class that got the majority votes among the candidate models will result in the final classification for the features supplies.
Soft Voting: Here the voting is based on the max average of the predicted class among the predicted sets of classified class probabilities. In this work the voting classifier uses a hybrid method which is

the combination of SVM, DT and LR methods. Algorithm 2 describes the proposed voting classifier mechanism.



Algorithm 2: Voting Classifier

Input: For (X_i Y_i,) X_i denotes the sensor values of features and Y_i denotes class labels for X_i for each i = 1,2,3,…,n

x is the point which class label is to be known

Output: The class for the respective features will be classified into the classes.

Step 1: Models train on training data

Step 2: Performance is compared.

Step 3: Voting takes place.

Step 4: The performance of the Majority Voting with the SVM, DT and LR is then used to find optimal results.

Fig. 4 Voting Classifier Representation

4. Results and Discussion

This work have implemented using the Python and its facilitating tools and libraries like Jupyter notebooks, Numpy, Seaborn, Sciket-Learn. Due to the simplicity and modularity of the Python programming language, it produces with its numerous toolkits, libraries, built-in classes and function for data processing, visualisation and implementation. This work have carried out in a Hewlett-Packard (HP) G7 notebook PC with Intel i3-1005G1 processor 2 cores and 4 logical processors having 4GB of RAM. The notebook runs in 64-bit version of Windows 10. For performing this work Python distribution of Anaconda was used.

This work mainly focuses on the following key points.

1. The input data is taken as a .csv file.

2. The data is then preprocessed by slicing the data into independent and dependent features and then scaled using Sklearn's Standardscaler.
3. Supervised ML algorithms such as KNN, SVM and DT are used for the classification to identify healthy states from the non-healthy states.
4. Ensemble learning is introduced in the form of Voting Classifier for stacking SVM, DT and LR for achieving stable and improved results.
5. The performance of the models is then checked by using performance metrics like accuracy, recall, precision.
6. The results are then verified using cross validation and checking the results for various weights for training and testing.

The dataset [23] was gathered on an air compressor which was reciprocating placed in IIT-Kanpur. It consists of 8 classes such as Leakage Inlet Valve (LIV) fault, Bearing fault, Rider belt fault, Leakage Outlet Valve (LOV) fault, Non-Return Valve (NRV) fault and Flywheel fault. The dataset comprises of 1800 rows and 255 columns that means there are 254 features extracted by sensors to consider for class labels. There are 8 classes out of which are a healthy one. In this work, the dataset is taken by analysing it in time and frequency domains. The features are reduced in dimensions.

In this work, the python code has been cross validated 10 times and their average is taken into consideration. Classification methods of SVM, KNN, DT and Voting Classifier have used to carry out the processing. For a classification problem the following outcomes are possible.

- True positives: It refers to the prediction belongs to the class that it really belongs to.
- True negatives: It refers to the prediction does not belong to the class that it really belongs to.
- False positives: It refers to the prediction does belong to a class that it does, in reality, does not belong to.
- False negatives: It refers to the prediction that do not belong to a class when in fact it does.

The performance metrics for checking the performance of a model are described as follows.

- Accuracy: It is predictions that are correct for the test dataset. It is represented in equation (3).

$$\text{Accuracy} = \frac{\text{Correct predictions}}{\text{all predictions}} \tag{3}$$

- Precision: It is the true positives to all which were predicted to belong in a certain class. It is represented in equation (4).

$$\text{Precision} = \frac{\text{true positives}}{\text{true positive+false positives}} \tag{4}$$

- Recall: It is the ratio of the predictions that belong to a class to the ones that really belong in the class. It is represented in equation (5).

$$\text{recall} = \frac{\text{true positives}}{\text{true positives+false negatives}} \tag{5}$$

- F1 Score: It is an alternative of accuracy metrics which provides the same weightage to precision as well as recall while performance measurement with the help of accuracy.

In this work, the training and testing ratios are considered as 60:40, 65:35 and 70:30 as mentioned in case-I, case-II and case-III respectively to determine the performance of KNN, SVM, DT and voting classifier models.

Case-I (60:40):

In this scenario, a comparison is made between the different algorithms like KNN, DT, SVM and voting classifier. The performance analysis of these methods is mentioned in Table 1 and Fig. 5. From the results, it is observed that the accuracy (in %) of KNN, DT, SVM and proposed voting classifier (hybrid method) are 97.7, 93.3, 97.2 and 98.4 respectively. The precision value (in units) of KNN, DT, SVM and proposed voting classifier are 98, 93, 98 and 99 respectively. The F1 score value (in units) of KNN, DT, SVM and proposed voting classifier are 98, 93, 98 and 98 respectively. The recall value (in units) of KNN, DT, SVM and proposed voting classifier are 97, 94, 98 and 98 respectively. So, from the results, it is observed that the proposed voting classifier method is able to provide better classification accuracy as compared to other methods and its accuracy (in %) is 98.4. The precision value of proposed voting classifier method is higher as compared to other methods. However, the F1 score and recall values of proposed voting classifier method is similar to some other methods as mentioned in Table 1 and Fig. 5.

Table 1 Performance analysis results of voting classifier and other methods in 60:40 scenario

ML technique	Accuracy	Precision	F1 Score	Recall
KNN	97.7	98	98	97
DT	93.3	93	93	94
SVM	97.2	98	98	98
Voting Classifier	98.4	99	98	98

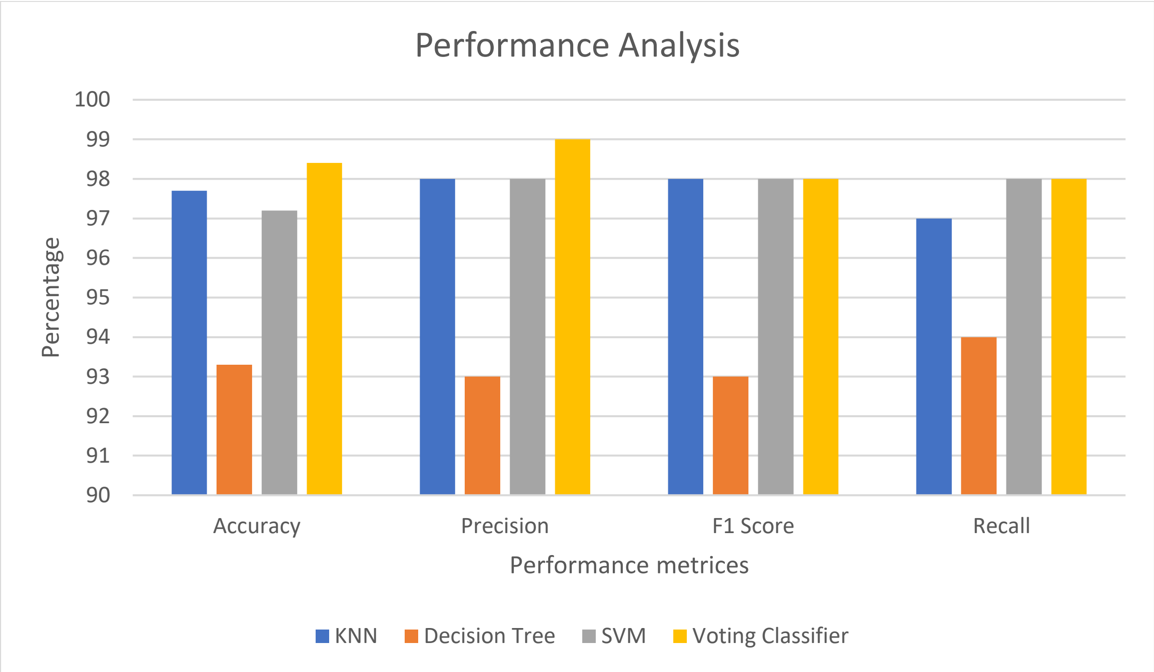


Fig. 5 Performance representation different algorithms in 60:40 scenario

Case-II (65:35):

In this scenario, a comparison is made between the different algorithms like KNN, DT, SVM and voting classifier. The performance analysis of these methods is mentioned in Table 2 and Fig. 6. From the results, it is observed that the accuracy (in %) of KNN, DT, SVM and proposed voting classifier are 96.9, 92.2, 98.7 and 99.04 respectively. The precision value (in units) of KNN, DT, SVM and proposed voting classifier are 97, 92, 99 and 99 respectively. The F1 score value (in units) of KNN, DT, SVM and proposed voting classifier are 97, 92, 99 and 99 respectively. The recall value (in units) of KNN, DT, SVM and proposed voting classifier are 97, 93, 99 and 99 respectively. So, from the results, it is observed that the proposed voting classifier method is able to provide better classification accuracy as compared to other methods and its accuracy (in %) is 99.04. However, the precision, F1 score and recall values of proposed voting classifier method is similar to some other methods as mentioned in Table 2 and Fig. 6.

Table 2 Performance analysis results of voting classifier and other methods in 65:35 scenario

ML technique	Accuracy	Precision	F1 Score	Recall
KNN	96.9	97	97	97
Decision tree	92.2	92	92	93

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 1, No. 3, January : 2024

SVM	98.7	99	99	99
Voting Classifier	99.04	99	99	99

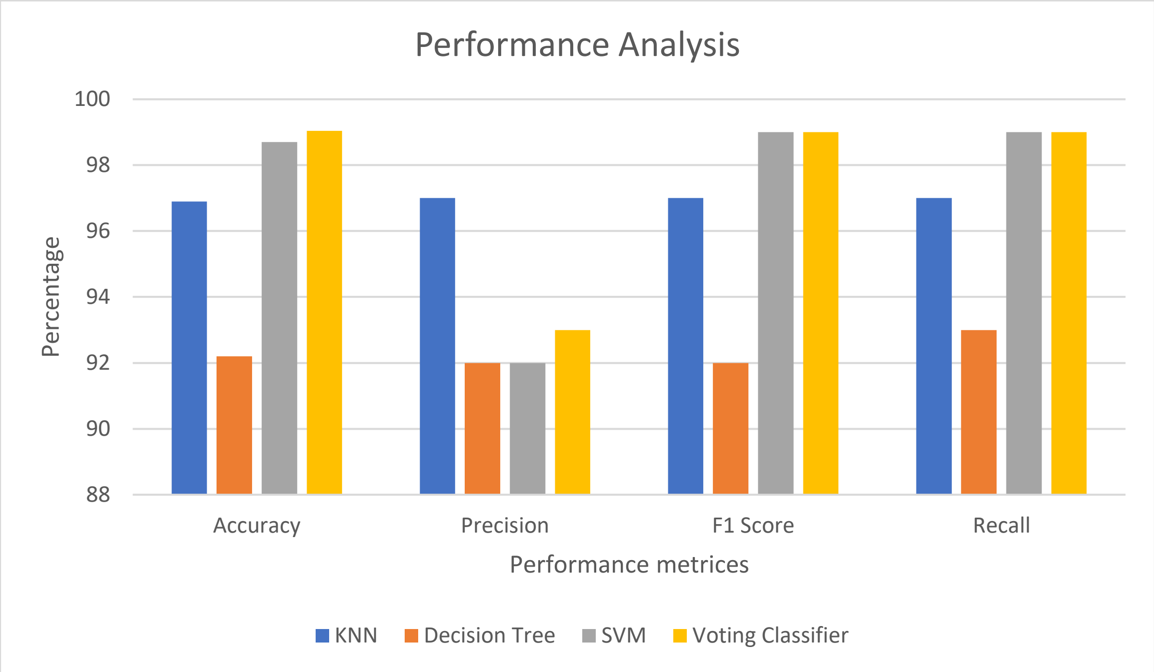


Fig. 6 Performance representation different algorithms in 65:35 scenario

Case-III (70:30):

In this scenario, a comparison is made between the different algorithms like KNN, DT, SVM and voting classifier. The performance analysis of these methods is mentioned in Table 3 and Fig. 7. From the results, it is observed that the accuracy (in %) of KNN, DT, SVM and proposed voting classifier are 97.1, 94.6, 99.1 and 99.17 respectively. The precision value (in units) of KNN, DT, SVM and proposed voting classifier are 97, 95, 99 and 99 respectively. The F1 score value (in units) of KNN, DT, SVM and proposed voting classifier are 97, 95, 99 and 99 respectively. The recall value (in units) of KNN, DT, SVM and proposed voting classifier are 97, 95, 99 and 99 respectively. So, from the results, it is observed that the proposed voting classifier method is able to provide better classification accuracy as compared to other methods and its accuracy (in %) is 99.17. However, the precision, F1 score and recall values of proposed voting classifier method is similar to some other methods as mentioned in Table 3 and Fig. 7.

Table 3 Performance analysis results of voting classifier and other methods in 70:30 scenario

ML technique	Accuracy	Precision	F1 Score	Recall
KNN	97.1	97	97	97
Decision tree	94.6	95	95	95
SVM	99.1	99	99	99
Voting Classifier	99.17	99	99	99

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 1, No. 3, January : 2024

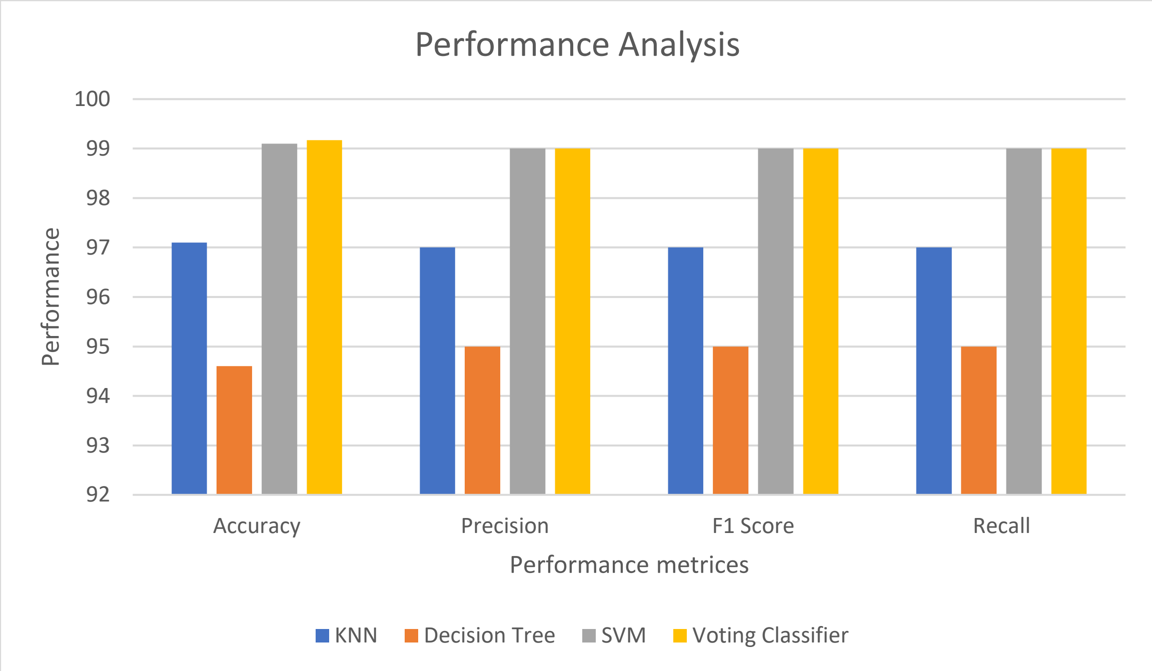


Fig.7 Performance representation different algorithms in 70:30 scenario

From the above analysis, it is observed that, the proposed voting classifier method is able to provide better accuracy as compared to KNN, DT and SVM methods in the mentioned cases.

In this work, an experimental demonstration have described by considering three sensors and one server device for fault detection as mentioned in Fig.8. In this experiment, the training and testing ratio is considered as 70:30. Here, the smart phones are considered as IoT devices and the dataset is created using IoT devices which are called raw data and . Smartphones can function not only as sensors but also as actuators. We took 3 mobile devices and installed Sensor Reader app in them. In this sensor reader app, Gyroscope is used for the orientation of the data with respect to the environment. It has been tested in 3 different phases with the help of gyroscope sensor to get 3 different vibration values which are shown in different csv files. One csv file of data has been named as a healthy class which means it indicates there are no faults in the dataset. The remaining two csv files of data have been named as a non-healthy which means it indicates there is a fault in the dataset. The data which is generated with the help of gyroscope sensor has been stored x, y, z coordinator. Then it has been combined with the help of an excel sheet. In the first test, Gyroscope in mobile has been performed steady position for that it has been created a csv file in which data has been present. In the second and third test, gyroscope in mobile has been performed more vibration for that it has been created csv file in which data has been present. In these 3 cases of the experiment, data points have been created in every 10 millisecond. Each csv file have been merged and resulted in a single csv file which includes both healthy and non-healthy class. After that, the dataset has been used in different machine learning algorithm in which KNN, Decision Tree, SVM and Voting Classifier has been used. In this experiment, the voting classifier provides better accuracy as compared to other methods and its accuracy (in %) is 98.99.

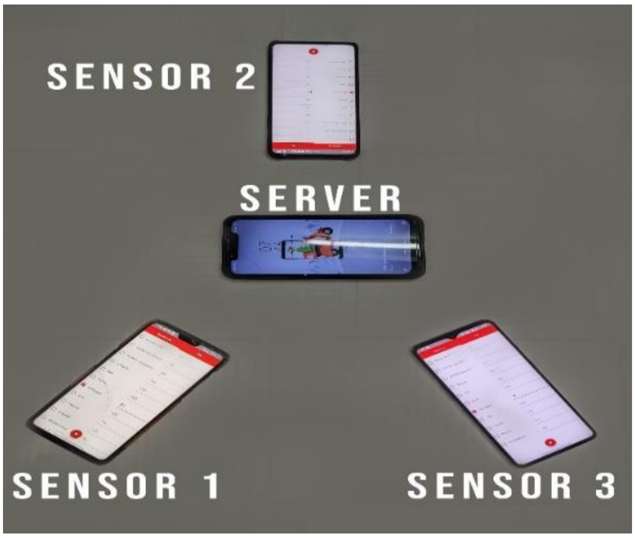


Fig. 8 Experimental Set-up

The experimental results of proposed voting classifier and other methods are mentioned in Table 4 and Fig. 9. From Table 4 and Fig. 9, it is observed that the accuracy (in %) of KNN, DT, SVM and proposed voting classifier are 98.84, 98.96, 97.05 and 98.99 respectively. The precision value (in units) of KNN, DT, SVM and proposed voting classifier are 97, 99, 97 and 99 respectively. The F1 score value (in units) of KNN, DT, SVM and proposed voting classifier are 97, 99, 97 and 99 respectively. The recall value (in units) of KNN, DT, SVM and proposed voting classifier are 97, 99, 97 and 99 respectively. So, from the results, it is observed that the proposed voting classifier method is able to provide better classification accuracy as compared to other methods and its accuracy (in %) is 98.99. However, the precision, F1 score and recall values of proposed voting classifier method is similar to some other methods as mentioned in Table 4 and Fig. 9.

Table 4 Experimental results of voting classifier and other methods in 70:30 scenario

ML technique	Accuracy	Precision	F1 Score	Recall
KNN	98.84	97	97	97
Decision tree	98.96	99	99	99
SVM	97.05	97	97	97
Voting Classifier	98.99	99	99	99

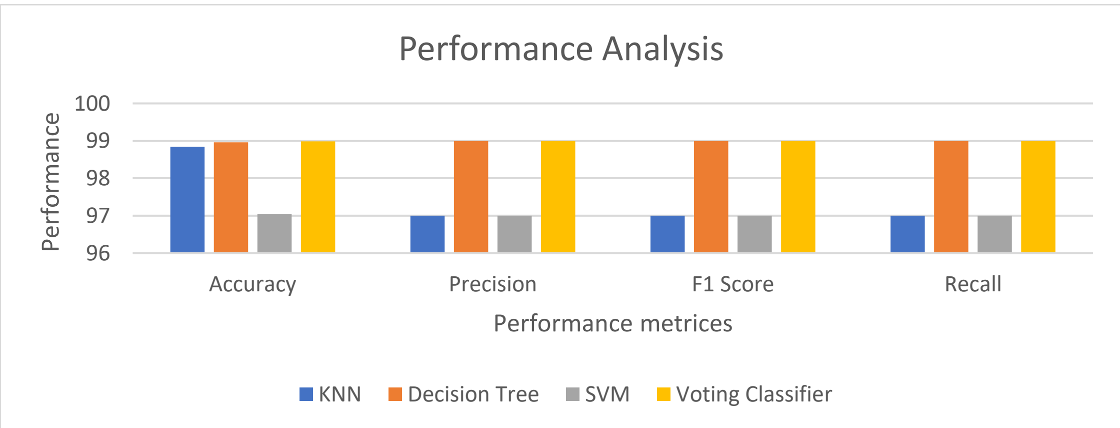


Fig.9 Experimental results representation of different algorithms in 70:30 scenario

From Fig. 5 –Fig. 9 and Table 1 – Table 4, it is observed that, the proposed voting classifier performs better as compared to the other ML based models such as KNN, DT and SVM in the mentioned cases.

So, the proposed voting classifier model can be used in the AI enabled cloud server for the detection of faulty or non-faulty data for further processing in this scenario.

6. Conclusion

In this work, a voting classifier method (hybrid method) is proposed for the detection of fault in the IIoT systems. The proposed voting classifier method is focused on the combination of the ML based methods such as SVM, DT and LR. The proposed method is compared with the existing ML based methods such as KNN, DT and SVM in terms of accuracy (in %), precision, F1 score and recall. From the results, it is concluded that, the proposed method is able to detect the fault in industrial IoT system environment in a better way as compared to KNN, DT and SVM. The overall accuracy of the proposed method in detecting the fault is around 99%. However, the overall accuracy of KNN, DT and SVM methods are around 97.6%, 94.8% and 98% respectively. So, the proposed approach is able to perform better as compared other methods in these scenarios for the detection fault. This work can be extended to develop enhanced methods to improve the performance for the detection of fault in industrial IoT environment. This work can also be extended to analyse the results by considering the other ML based methods such as random forest, neural network and Naïve Bayes.

References

[1] Lo, N. G., Flaus, J. M., & Adrot, O. (2019, July). Review of machine learning approaches in fault diagnosis applied to IoT systems. In 2019 International Conference on Control, Automation and Diagnosis (ICCAD) (pp. 1-6). IEEE.
[2] Sun, W., Liu, J., & Yue, Y. (2019). AI-enhanced offloading in edge computing: When machine learning meets industrial IoT. IEEE Network, 33(5), 68-74.
[3] Mokhtari, S., Abbaspour, A., Yen, K. K., & Sargolzaei, A. (2021). A machine learning approach for anomaly detection in industrial control systems based on measurement data. Electronics, 10(4), 407.
[4] Angelopoulos, A., Michailidis, E. T., Nomikos, N., Trakadas, P., Hatziefremidis, A., Voliotis, S., & Zahariadis, T. (2020). Tackling faults in the industry 4.0 era—a survey of machine-learning solutions and key aspects. Sensors, 20(1), 109.
[5] Christou, I. T., Kefalakis, N., Zalonis, A., & Soldatos, J. (2020, May). Predictive and explainable machine learning for industrial internet of things applications. In 2020 16th International Conference on Distributed Computing in Sensor Systems (DCOSS) (pp. 213-218). IEEE.
[6] Sarita, K., Kumar, S., & Saket, R. K. (2021). Fault Detection of Smart Grid Equipment Using Machine Learning and Data Analytics. In Advances in Smart Grid Automation and Industry 4.0 (pp. 37-49). Springer, Singapore.
[7]Fernandes, M., Canito, A., Corchado, J. M., & Marreiros, G. (2019, June). Fault detection mechanism of a predictive maintenance system based on autoregressive integrated moving average models. In International Symposium on Distributed Computing and Artificial Intelligence (pp. 171-180). Springer, Cham.
[8] Dutta, N., Kaliannan, P., & Subramaniam, U. (2021). Application of Machine Learning Algorithm for Anomaly Detection for Industrial Pumps. In Machine Learning Algorithms for Industrial Applications (pp. 237-263). Springer, Cham.
[9] Kotsiantis, S. B., Zaharakis, I., & Pintelas, P. (2007). Supervised machine learning: A review of classification techniques. Emerging artificial intelligence applications in computer engineering, 160(1), 3-24.
[10] Choudhary, R., & Gianey, H. K. (2017, December). Comprehensive review on supervised machine learning algorithms. In 2017 International Conference on Machine Learning and Data Science (MLDS) (pp. 37-43). IEEE.
[11] Liu, Y., Yang, Y., Lv, X., & Wang, L. (2013). A self-learning sensor fault detection framework for industry monitoring IoT. Mathematical problems in engineering, 2013.

[12] Hooman, J., Mooij, A. J., & van Wezep, H. (2012, June). Early fault detection in industry using models at various abstraction levels. In International Conference on Integrated Formal Methods (pp. 268-282). Springer, Berlin, Heidelberg.
[13]de Sousa, P. H. F., Navar de Medeiros, M., Almeida, J. S., Rebouças Filho, P. P., & de Albuquerque, V. H. C. (2019). Intelligent Incipient Fault Detection in Wind Turbines based on Industrial IoT Environment. Journal of Artificial Intelligence and Systems, 1(1), 1-19.
[14] Seabra, J. C., Costa, M. A., & Lucena, M. M. (2016, September). IoT based intelligent system for fault detection and diagnosis in domestic appliances. In 2016 IEEE 6th International Conference on Consumer Electronics-Berlin (ICCE-Berlin) (pp. 205-208). IEEE.
[15]Murugan, N., Senthil Kumar, J. S., Thandapani, T., Jaganathan, S., & Ameer, N. (2020). Underground Cable Fault Detection Using Internet of Things (IoT). Journal of Computational and Theoretical Nanoscience, 17(8), 3684-3688.
[16] Abdelgayed, T. S., Morsi, W. G., & Sidhu, T. S. (2017). Fault detection and classification based on co-training of semisupervised machine learning. IEEE Transactions on Industrial Electronics, 65(2), 1595-1605.
[17] Said, M., Abdellafou, K. B., & Taouali, O. (2020). Machine learning technique for data-driven fault detection of nonlinear processes. Journal of Intelligent Manufacturing, 31(4), 865-884.
Jan, S. U., Lee, Y. D., & Koo, I. S. (2021). A distributed sensor-fault detection and diagnosis framework using machine learning. Information Sciences, 547, 777-796.
[18]Meng, Y., Wu, X., Oladejo, J., Dong, X., Zhang, Z., Deng, J., ... & Pang, C. H. (2021). Application of Machine Learning in Industrial Boilers: Fault Detection, Diagnosis, and Prognosis. ChemBioEng Reviews, 8(5), 535-544.
[19] De Vita, F., Bruneo, D., & Das, S. K. (2021, August). A Semi-Supervised Bayesian Anomaly Detection Technique for Diagnosing Faults in Industrial IoT Systems. In 2021 IEEE International Conference on Smart Computing (SMARTCOMP) (pp. 31-38). IEEE.
[20] Sharghivand, N., & Derakhshan, F. (2021). Classification and Intelligent Mining of Anomalies in Industrial IoT. In AI-Enabled Threat Detection and Security Analysis for Industrial IoT (pp. 163-180). Springer, Cham.
[21] R. Thriukovalluru, S. Dixit, R. K. Sevakula, Nishchal K. Verma and A. Salour, Generating Feature Sets for Fault Diagnosis using Denoising Stacked Auto-encoder, IEEE International Conference on Prognostics and Health Management, Canada USA, pp. 1-7, June 2016
[22] Nishchal K. Verma, R. K. Sevakula, S. Dixit and A. Salour, Intelligent Condition Based Monitoring using Acoustic Signals for Air Compressors, IEEE Transactions on Reliability, vol. 65, no. 1, pp. 291-309, 2016.
[23] Air Compressor Fault Dataset: https://drive.google.com/open?id=10s0eakkpyOTq-8zOl4uDMJ4LlHmAcdEk, accessed on Sep 2021.
[24] Swain, R. R., Khilar, P. M., & Bhoi, S. K. (2018). Heterogeneous fault diagnosis for wireless sensor networks. *Ad Hoc Networks*, *69*, 15-37.
[25] Bhoi, S. K., & Khilar, P. M. (2016). Self soft fault detection based routing protocol for vehicular ad hoc network in city environment. *Wireless Networks*, *22*(1), 285-305.
[26] Tran, M. Q., Doan, H. P., Vu, V. Q., & Vu, L. T. (2023). Machine learning and IoT-based approach for tool condition monitoring: A review and future prospects. *Measurement*, *207*, 112351.
[27] Abualsauod, E. H. (2023). Machine learning based fault detection approach to enhance quality control in smart manufacturing. *Production Planning & Control*, 1-9.