SN Computer Science (2022) 3:273
https://doi.org/10.1007/s42979-022-01181-2

ORIGINAL RESEARCH

dLeak: An IoT-Based Gas Leak Detection Framework for Smart Factory

Anamika Rajbanshi[1] · Debanjan Das[2] · Venkanna Udutalapally[3] · Rajarshi Mahapatra[2]

Received: 27 October 2021 / Accepted: 26 April 2022

Abstract
Gas industries very often suffer from leakage. This paper proposes an IoT-enabled acoustic-based leak detection system to detect leaks more accurately and rapidly that can improve the safety of such gas industries. Acoustic Emission (AE) signals are collected from a test setup of an industry, which is mimicked by a working kitchen where a pressure cooker whistle was used in the place of a leak. Each AE signal is segmented using a hamming window and features are extracted from every segment. According to our data set, a best fit model has been implemented. Naive Bayes, SVM and KNN classifiers are giving acceptable testing accuracy to detect the leaks. Signals are collected from the spot where the model has to be implemented. Then, these signals are transported to the supervisor's device using the cloud. The signals are finally tested in the implemented model which successfully predicts the presence of a leak in that spot. The Naive Bayes, SVM and KNN models gave high testing accuracy of 92.2%, 92.4% and 94.44% respectively, of which KNN gave the best performance.

Keywords Leak detection · IoT · Naive bayes · Support vector machine (SVM) · Machine learning · Acoustic sensing · Smart factory

Introduction

Industries are the main hub of any country. A main part of any country's development depends on its industrial development and production. Therefore, in today's generation, it is quite difficult to imagine livelihood without these industries. But, like the famous proverb "Two sides of the same coin", there are certain dangerous disadvantages that have been imposed on our societies by these industries. These industries work with substances that are highly toxic in nature and can be extremely dangerous for human life. Generally, these toxics are handled via pipelines. Therefore, rupturing of a pipeline can cause havoc due to the release of these industrial waste.

From a long time, researchers are finding methods to cope with this problem of leak detection in the pipelines [9]. Previously, various model-free and model-based techniques have been implemented regarding leak detection and research is still going on to completely eradicate this problem. A model-free system was developed which differentiates between abnormal and normal signals with the help of signal-to-noise ratio and cross-correlation of the signals [19]. Wang et al. show a model-based classification using SVM [20]. Further, Principle Component Analysis (PCA) is used for noise removal. However, the above mentioned paper has the disadvantage of attenuating the acoustic signal and sample leaks were difficult to manufacture. Many dangerous gas leak incidents in industries that have affected many workers as well as locals show the immediate need of a trusty worthy solution for these gas leak problems. In the modern times, IoT is used in various applications, such as in medical industry [10], smart cities [26] and also in educational institutes [27].The IoT platform helps to alert the officials working in the industries about any kind of possible leak that has been detected by the Leak detection systems

✉ Debanjan Das
debanjan@iiitnr.edu.in

Anamika Rajbanshi
ana.rajbanshi@gmail.com

Venkanna Udutalapally
venkannau@iiitnr.edu.in

Rajarshi Mahapatra
rajarshi@iiitnr.edu.in

[1] Department of Electrical Engineering, Jadavpur University, Kolkata, India

[2] Department of Electronics and Communication Engineering, IIIT, Naya Raipur, India

[3] Department of Computer Science and Engineering Engineering, IIIT, Naya Raipur, India

Published online: 05 May 2022

implemented in the factories as soon as any abnormality is predicted by the systems. Baiji et al. presented an IoT-based model which can detect leak and will be able to transfer the information about leak to any device with the help of IoT [2]. The above-mentioned paper uses logistic regression machine learning model for detection. However, the model output can be affected by outside pressure and environmental conditions. A better algorithm with more features can be implemented in that paper.

The present paper focuses on building a leak detection model or dLeak model which is to be implemented in any industry and then analysis of the leak is done with the help of IoT-enabled system, so that immediate action can be taken regarding the leak and, hence, the safety of the humankind will be maintained. Acoustic Emission (AE)-based signals generated from the pipelines are sensed by a microphone sensor and then uploaded to the cloud from where it can be transferred to your required device for processing and analysis of the signal. The AE signals are segmented using a hamming window and necessary features are extracted from each signal for leak prediction. Mel-Frequency Cepstral Coefficients (MFCC) and pitch are the two features that have been used for training the model and prediction of the signals. Certain features extracted from the signals are not necessary for training the machine learning model and, more importantly, it can lead to decrease of the accuracy of the dLeak model. Therefore, Minimum Redundancy Maximum Relevance (MRMR) technique is used for selecting important features of the signal and these features are used for training the model. After proper feature extraction and selection, a best fit model is selected that gives good accuracy after training the features extracted from the audio data set. This dLeak model is then used for predicting abnormalities in the new signals coming from the sensors attached in the pipelines.

The rest of the paper is organised as follows: Sect. 2 briefly describes and compares various existing methodologies and describes the contribution of the current paper in previous methodology's drawbacks. In Sect. 3, the detailed procedure of proposed methodology is described. Section 4 explains the experimental setup for this paper and Sect. 5V gives an elaborate description of result analysis. Finally, Sect. 5 concludes the paper with future works.

Related Prior Works and Contribution of the Current Paper

Related Prior Research

There are few research groups who have presented solutions for leak detection [19, 20, 2]. Among those studies, AE-based technique has been highly successful in detecting the leak that are very sensitive to the outer noise [24, 6, 15, 12–14, 22]. Machine learning techniques are gaining high significance for detecting leak using pattern recognition [25, 5, 23, 17]. Artificial Neural Network (ANN) has also been used to detect leak in pipelines where several sensors are placed [3]. Mahmud et al. show the use of IoT to carry information about any faults in structures [11]. In [16], the authors have implemented a technique of transforming the AE signals, collected from sensors that are installed in the pipelines, with a function such that the transformed signals can be used for training the leak detection model with minimum number of data set. They have used the theory of wave attenuation for finding out the position of leak in the pipeline. Root mean square (RMS), short time energy (STE) and average amplitude (AVA) are the features extracted from the transformed AE signals for model development. Furthermore, the paper implements a k-nearest neighbor (KNN) model to classify between normal and abnormal signals. They have also compared the difference in test result when direct AE signals are used for training the model and when transformed signals are used. Gaussian noise is further added into the signals for testing the robustness of the trained model. The same online monitoring method of an Ammonia plant by collecting AE signals is used in [18]. Leak detection in a propulsion system pipelines of sounding rocket is presented in [21]. Both time-domain and frequency-domain features are extracted from the AE signals and Multi-class SVM model is used for classification. 2-D image-based learning is also very popular in MFCC [8] [1]. However, due to decrease in text accuracy, deep learning method is not used in this paper.

Problem Identification

In [16], the authors have used particularly three features that are RMS, STE and AVA for training the model. However, certain other features can also be used for increasing the accuracy of the model. Moreover, they have used only 72 data sets for training a machine learning model which is quite low compared to the data needed for training a model. Also, the paper focuses on leaks generated by water pipelines only and does not specify anything about gas or other chemical pipelines. Since, the experiment was conducted in a laboratory simulation, therefore, no practical noise was induced in the system. Hence, we could not figure out the effectiveness of the system in practical environments. The transformation function, used in the paper, is based on a variable, β such that $\beta = e^{(-\alpha d)}$ where α is the attenuation coefficient in wave propagation and d is the distance between the sensors. If the parameter β is not chosen properly, then high sensitivity and reliability will not be achieved in the model. The authors have used particularly KNN classifier for identifying leak, whereas other classifiers can also be

used that can give higher training and testing accuracy. In cross-testing accuracy, the model shows 77.9% and 82 % in certain pressure conditions which shows that the model can misbehave in particular environmental conditions.

Proposed Solutions of the Current Paper

This paper has proposed an IoT framework as shown in Fig 1 for real-time monitoring and detection of gas leak. Sequential feature selection is implemented so that correct features are extracted from the audio data set used by us and the model developed using these features will be more robust. MFCC and Gammatone cepstral coefficients (GTCCs) are the features that give maximum accuracy when model is trained using them; hence, MFCC features are used in our model. It is a well-known fact in machine learning that an increase in training data will make the model more accurate for prediction as it is getting exposed to all kinds of inputs possible. Therefore, a total of 300 data sets are used for training the model. Moreover, the features extracted from the audio signals are trained in many classifiers to identify the correct classifier for our AE-based signals that will give the best training and testing accuracy. SVM and Naive-Bayes are the classifiers that are giving the highest accuracy in our model. In our work, the leak is produced by such a device that will mimic the leak sound of any gas or water pipelines. Hence, this work can be implemented in any gas or water pipelines. Our trained model is successful in classifying between normal and leak signal and, therefore, the use of transformed signals is not required which will in fact nullify the dependency of the sensitivity and reliability of the model on β. In [16], the authors are stating transformation-based feature extraction method better than the AE-based feature extraction method because there is an overlap between the states regarding the latter method. However, such an overlap is not observed in the current paper method which indicates features have been clearly extracted for training and there is no need for transformation.

Novelty and Contribution of the Current Paper

Previous gas tragedies in the societies due to leakage in industrial pipelines show how important it is for researchers to implement a solution for this problem as soon as possible. Development of a working model that can classify between leak and normal signal and retrieving that information immediately using IoT are the main aims of this paper. The major contributions in the current paper are shown below:

- This paper proposes an acoustic emission (AE)-based gas leak detection model which is capable of identifying gas leaks using edge-as-a-service to prevent accidents in an industry.
- The proposed model gives a good accuracy which is comparatively higher than the previous models and hence can be implemented for leak detection in practical scenarios.
- The number of training data points has been significantly increased, which helped in the development of a more precise and reliable model.
- The gas leak detection model is tested using an experimental setup in a working kitchen, where the noise from other kitchen appliances represents the environmental noise in an industry. The proposed model works successfully in spite of the noise, thereby establishing the robustness of the model.
- This approach requires no transformation of the AE signal; therefore, this method can be implemented in any kind of industry very easily.

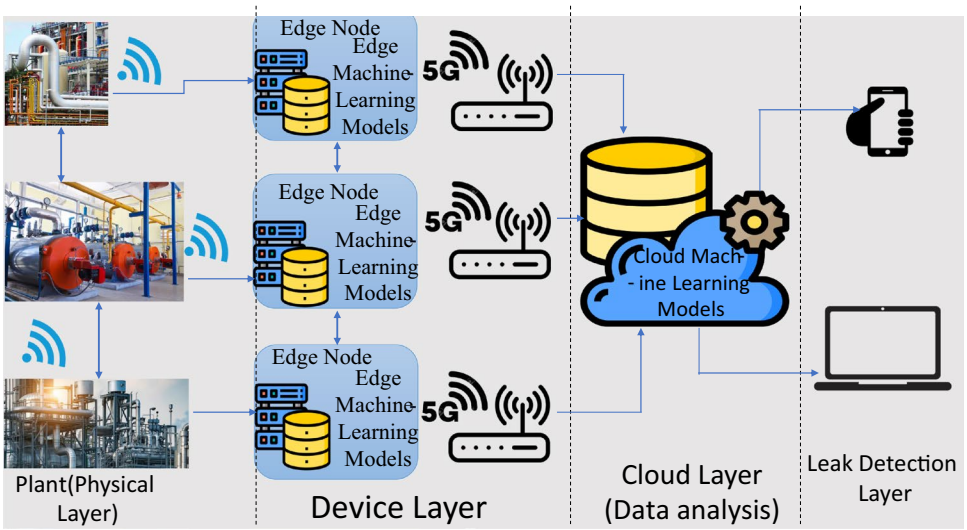

Fig. 1 Proposed IoT framework for smart gas leak detection in a plant

dLeak: Proposed Leak Detection Model

The methodology used in this paper is explained in Fig. 2 in a sequential manner. The project is divided in two parts: training and testing. Training part includes data collection and pre-processing, feature extraction, feature selection and machine learning training. The testing part includes AE signal collection, cloud uploading, transferring to laptop, data pre-processing and feature extraction and finally testing the condition of the signal in the developed model. Brief description of the entire method is given below:

First, the model has to be trained by necessary audio data. So, 300 samples are collected in total which are further classified in abnormal and normal signals. The entire data set is normalized and converted to mono-channel for better result. Then, the signals are organized in an audio data store and further divided into training and testing data store in the ratio of 7:3. MFCC and pitch are the features that are extracted from the data store. After feature extraction, all the features are prioritized using Minimum Redundancy Maximum Relevance (MRMR) and visualized using Principle Component Analysis (PCA). After ranking the features, the important features are used for machine learning training. After training with many models and implementing fivefold cross-validation, the best classifier is chosen which gives the maximum training accuracy. After changing certain hyper-parameters, implementing optimization and imposing higher cost matrix to certain misclassification classes the model are exported for testing with new data. The exported model gives a high testing accuracy with the testing data store. The trained model can be used for leak detection. The sensors attached in the pipelines are constantly measuring AE signals generated by the pipelines. These signals are uploaded to the cloud and can be downloaded to the laptop or desktop from the cloud with the use of IoT platform. These upcoming signals can now be tested with the developed model which will inform about the condition of the coming signals. In case of any abnormality in the signals, quick results can be obtained via IoT and necessary steps can be carried out by the industrial workers. The entire testing process in the plant is shown in Fig. 1.

This section is divided into further sub-sections: The first sub-section defines the data collection and the data preprocessing stage. The next subsection defines the formation of datastore which is done to reduce memory usage because of the large size of data and the visualization of data. The next sub-section elaborately describes the entire feature extraction process. It describes the features that are chosen for this paper and it also explains the signification of the features. The definite steps and the entire map of the feature extraction process are also defined. Sub-section 4 elaborates the necessity of feature section and the algorithm used for correctly selecting features. The final sub-section explains the training process and the method of choosing the best-fit model. It also explains the optimization procedure and misclassification cost specification to increase model accuracy.

Data Collection and Preprocessing

The first stage of machine learning is data collection. Initially, a total of 100 data are collected, with a duration of 2 min for each data, which are constituting equally of normal and abnormal signals. Then, the audio data are categorized in abnormal and normal folders. Data preprocessing is a crucial stage because it helps in reduction of complexity of the data as data from real world is unclean and ultimately helps to increase the success rate of the project. All the signals collected are first normalized in the range of + 1 to - 1. It has to be checked whether every signal has the same sample rate or not and if it is not the same then the sampling rate of every signal must be changed to the same value. Audio files can either be monophonic (mono) or stereophonic (stereo). Mono-signals are registered and replayed using a single audio channel, while stereo sounds are captured and replayed

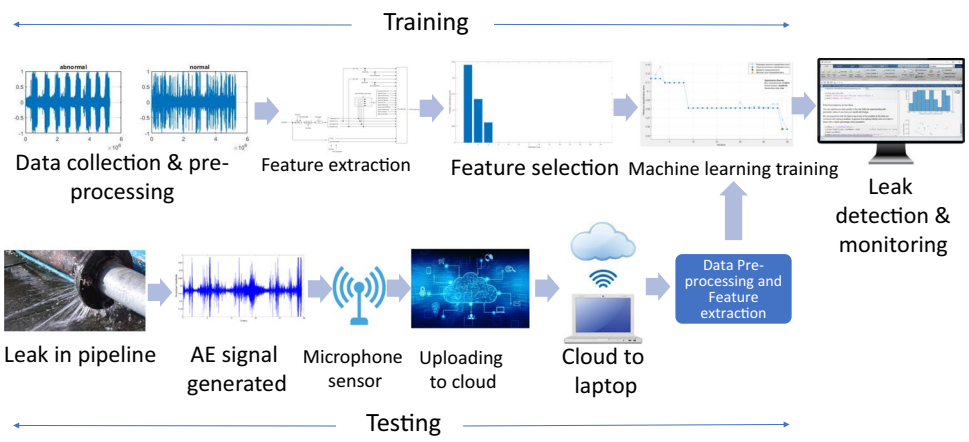

Fig. 2 Functional processes to detect leaks from acoustic signals

using two audio channels. For better training, the percentage of mono- and stereo sounds are calculated and if any stereo sound is present in the recorded sounds then that is converted to mono-sound.

Datastore Formation and Visualization

Datastore Formation

A datastore is a reservoir where data are collected that are too large to fit in memory. A datastore is used to scan and maneuver the data stored in multiple files on a disk, a remote location, or a database as a single unit. For creating a datastore, the file location of the folder containing the necessary audio files is to be provided. The necessary datastore will include all files and sub-folders within each folder and the data will be labelled according to its folder names. Further, the datastore is parted into training and testing datastore with the ratio of 7:3. Now, the datastore is augmented with the properties of time stretch, speed up, pitch shift, semitone shift, volume control, noise addition and time shift. Each audio signal is augmented to 2 other signals and thus, the number of total files increased from 100 to 300.

Data Visualization

The datastore created must be visualized to observe the quality of the data and to get an overall look of the data. The data must be visualized before and after augmentation to ensure data do not distort much from the original files.

Feature Extraction

The next step is extracting necessary features from the datastore for accurate training of the model. Each signal is analyzed before feature extraction by hamming window with a length of the sampling frequency of the signal. Hamming window, in digital signal processing (DSP) is defined for a non-causal signal and causal signal in equation (1) and (2) respectively.

$$\begin{aligned} w_n &= 0.54 + 0.46cos(2\pi n/(M-1)), for|n| \le (M-1)/2 \\ &= 0, otherwise; \end{aligned} \tag{1}$$

$$\begin{aligned} w_n &= 0.54 - 0.46cos(2\pi n/(M-1)), for|n| \le (M-1) \\ &= 0, otherwise; \end{aligned} \tag{2}$$

where M is the sampling frequency and n is the frequency of the signal. An overlap of 50% is applied in the hamming window. If the windows are not overlapping, it is possible to miss important information in the boundary of the signal.

Now, MFCC and pitch are extracted from each window. The MFCC features are averaged for the same audio file. The feature files are converted into a table format where each column of feature file becomes a variable in the table and the features are labelled according to the normal and abnormal label of the audio files.

Mel-Frequency Cepstral Coefficients (MFCC) and Pitch

There is a critical frequency bandwidth of the human ear and based on its known variation, MFCC is obtained which is quite a popular feature extraction method. Psychophysical studies have revealed that the sound frequency content of the audio signals as perceived by human does not follow a linear scale. Therefore, for each audio signal having a frequency, f(in Hertz), a scale termed as the 'Mel' scale is used to calibrate the pitch of the signal. The expression of 'Mel' scale for each frequency is shown in Eq. (3).

$$f_{mel} = 2595 \log_{10}(1 + \frac{f}{700}), \tag{3}$$

where f_{mel} is the subjective pitch in Mels for a particular frequency. Thus, MFCC can be defined as a feature set for audio signal which is implemented in speech and speaker identification methodologies.

MFCC coefficients include a set of Discrete Cosine Transform (DCT) decorrelated frameworks, which are calculated through a conversion of the logarithmically compressed filter-output energies, resultant of a conceptually spaced triangular filter bank that compiles the Discrete Fourier Transformed (DFT) speech signal. An M-point DFT of the discrete input signal y(n) is shown in Eq. (4).

$$Y(k) = \sum_{n=1}^{M} y(n) e^{\frac{-j2\pi nk}{M}} = 1 \tag{4}$$

where $1 \le k \le M$. Next, the filter bank which has linearly distributed filters in the Mel scale, are extracted on the spectrum. The filter output $\Psi_i(k)$) of the ith filter in the bank is defined by Eq. (5).

$$\begin{aligned} \Psi_i(k) &= 0, fork \le_{b_{i-1}} \\ &= \frac{k - k_{b_{i-1}}}{k_{b_i} - k_{b_{i-1}}}, fork_{b_{i-1}} \le\le k_{b_b} \\ &= \frac{k_{b_{i+1}} - k}{k_{b_{i+1}} - k_{b_i}}, fork_{b_{i-1}} \le\le_{b_{i+1}} \\ &= 0, fork \le_{b_{i+1}} \end{aligned} \tag{5}$$

If Q denotes the number of filters in the filter bank, then $\{K_{b_i}\}_{i=0}^{Q+1}$ are the borderline points of the filters and k shows us the coefficient parameters in the M-point DFT. The borderline items for each filter i (i=1,2,...,Q) are computed as uniformly distributed points in the Mel scale utilizing Eq. (6).

$$K_{b_i} = \left(\frac{M}{f_s}\right).f_{mel}{}^{-1}\left[f_{mel}(f_{low}) + \frac{i\{f_{mel}(f_{high}) - f_{mel}(f_{low})\}}{Q+1}\right] \tag{6}$$

where f_s is the sampling frequency in Hz and f_{low} and f_{high} are the low- and high-frequency border of the filter bank, respectively. f_{mel}^{-1} is the inverse of the conversion displayed in Eq. (3) and is shown in Eq. (7).

$$f_{mel}^{-1}(f_{mel}) = 700 \cdot \left[10^{\frac{f_{mel}}{2595}} - 1\right] \tag{7}$$

In the next step, the output energies e(i)(i=1,2,.....,Q) of the Mel-scaled band-pass filters are computed as an aggregate of the signal energies $|y(k)|^2$ falling into a given Mel frequency band subjected by the particular frequency output of $\Psi_i(k)$ as shown in Eq. (8).

$$e(i) = \sum_{K}^{M} |y(k)|^2 \Psi_i(k) \tag{8}$$

Finally, an implementation of DCT is computed over the log filter bank energies $\{\log[e(i)]\}_{i=1}^{Q}$ to de-correlate the energies and the final MFCC coefficients C_m are shown in equation (9).

$$C_m = \sqrt{\frac{2}{N}} \sum_{l=0}^{M-1} \log[e(l+1)].\cos\left[m\left(\frac{2l+1}{2}\right).\frac{\Pi}{Q}\right] \tag{9}$$

where m=0,1,2,...., R-1 and R are the required numbers of MFCCs. The importance of MFCC can be understood using [7]. Apart from MFCC, pitch of the audio signals is also used for successful feature extraction.

Feature Extraction Map

The method of entire feature extraction is shown in Fig. 3. In the first step, the audio captured is passed through a hamming window where the signal is segmented. The signal is then passed through a Fast Fourier Transform (FFT). The subjective pitch in Mels is obtained for the signal and the MFCC coefficients are calculated. Simultaneously, the pitch of the signal is also calculated. After that, necessary features are concatenated to make the feature set.

Feature Selection

Fourteen features in total are calculated from the entire datastore using the above described feature extraction methods which are finally stored in V_S and W_S. There is a need for feature selection because sometimes all the features are not required for the machine learning training and the excess features can deteriorate the accuracy of the model. Therefore, careful feature selection and feature visualization are required so that the model is trained from the correct features. MRMR algorithm is used for feature selection and Principle Component Analysis (PCA) is used for feature visualization before machine learning training.

Minimum Redundancy Maximum Relevance (MRMR) Algorithm

MRMR algorithm finds the best set of features F which will make A_F maximum, the importance of F in accordance with the response variable j, and will make B_F minimum, the uselessness of F, where A_F and B_F are defined with mutual information M in Eq. 10 and 11.

$$A_F = \frac{1}{|F|}\sum_{i\epsilon F} M(i,j), \tag{10}$$

$$A_F = \frac{1}{|F|^2}\sum_{i,k\epsilon F} M(i,k). \tag{11}$$

$|F|$ is the number of features in F.

All the combinations of $2^{|E|}$ are to be calculated for finding the best-fit set of F, where E is the total set of features. The MRMR algorithm categorizes the features with the help of the forward addition method, which provides $C(|E| \cdot |F|)$ computations, by using the mutual information quotient (Q) value.

$$Q_i = \frac{A_F}{B_F}, \tag{12}$$

where A_F and B_F are the importance and uselessness of a feature, respectively and are shown in Eqs. 13 and 14.

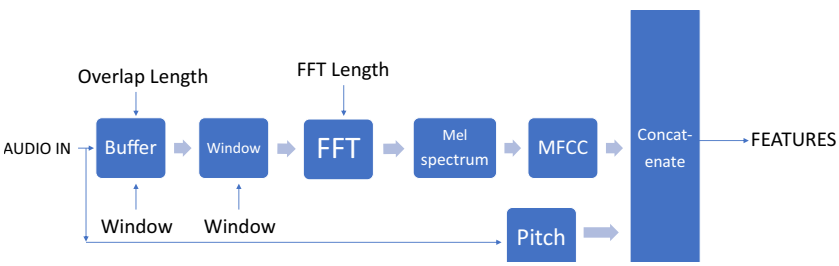

Fig. 3 Flowchart showing the necessary steps for feature extraction

$$A_F = M(i,j), \tag{13}$$

$$B_F = \frac{1}{|F|}\sum_{k\epsilon F} M(i,k). \tag{14}$$

The MRMR algorithm categorizes all the features in E and returns idx (all the features indicated by their importance) which makes the cost of computation C($|E|^2$). The MRMR function computes the significance of a feature using an empirical methodology and calculates the score. A large score value signifies that the particular predictor is important. The decline in the score characterises the assurance of feature selection. For example, if the algorithm is certain about a particular feature x, then the score value of the next most significant feature will be much smaller than the score value of x. The outputs are used to find the best-fit set S for a given number of features. The important features are selected successfully using this method. C. Ding, H. Peng elaborately describe this method [4].

The MRMR function identifies the best set of features that is conjointly and utterly different and can symbolize the response variable efficiently.The algorithm minimizes the feature set and maximizes the signification of a feature set to the response variable. The process calculates importance of features with the help of the communal knowledge of variables-pairwise mutual information and also the communal knowledge of a feature and the response.

Machine Learning Training

After careful selection of features, they are trained by classifiers of Naive Bayes, Nearest Neighbor, Decision Trees, Logistic Regression, Support Vector Machines (SVM), Discriminant analysis, and Ensemble classifiers with five-fold cross-validation. Further, the best fit model is chosen which gives the maximum accuracy. Then, optimization is implemented in the best fit model and it is observed whether implementation has increased or not. Accuracy and testing results can still be improved by imposing higher cost matrix to certain misclassification classes. After experimenting with the parameters/hyper-parameters, the model which has reasonable training time and prediction speed is exported for testing with new data. The necessary model is saved for future purpose.

Best Fit Model

After training the features with the above-mentioned classifiers, three classifiers are giving the best training accuracy which are Naive Bayes, SVM and k-nearest neighbour (KNN) classifier.

Naive Bayes classifiers are quite simple to elaborate and efficient for numerous class classification. The naive Bayes principle holds the Bayes theorem and makes the presumption that predictors are independent for certain conditions, given the class.Though the prediction is generally defiled in application, naive Bayes classifiers gravitate to generate posterior distributions that are resilient to biased class density assessments, especially where the posterior is 0.5 (the decision confinement). Naive Bayes classifiers appoint examinations to the most possible class (which basically means the maximum a posteriori decision rule). The algorithm is defined below:

1. Assess the densities of the predictors for each class.
2. Models posterior probabilities presenting to the Bayes rule which means for all m = 1,...,M,

$$\hat{P}(B = m|A_1, ..., A_P) = \frac{\pi(B = m)\Pi_{j=1}^{P} P(A_j|Y = k)}{\sum_{m=1}^{M} \pi(B = m)\Pi_{j=1}^{P} P(A_j|B = m)}, \tag{15}$$

   where:

   - B is the random variable parallel to the class index of an examination.
   - $A_1, ..., A_P$are the random predictors of an examination.
   - π(B=m) is the prior probability that a class index is m.

3. Categorizes an examination calculating the posterior probability for each class, and then attaches the examination to the class producing the maximum posterior probability.

If the predictors constitute a multinomial distribution, then the posterior probability

$$\hat{P}(B = m|A_1, ..., A_P) \propto \pi(B = m)P_{mnd}(A_1, \dots, A_P|B = m), \tag{16}$$

where $P_{mnd}(A_1, \dots, A_P|B = m)$ is the probability mass function of a multinomial distribution.

An SVM categorizes data by obtaining the finest hyperplane that segregates data points of one class from those of the other class. The fittest hyperplane for an SVM means the one with the maximum margin between the two classes. Margin means the maximal width of the slab parallel to the hyperplane that has no internal data points. The support vectors are the data points that are nearest to the segregating hyperplane; these points are on the confinement of the slab. SVMs can also use a soft margin, meaning a hyperplane that segregates many, but not all data points.

Classifying query points on the basis of their distance to points (or neighbors) in a training dataset can be an effortless yet effectual way of categorizing new points. Diverse metrics can be used to calculate the distance. Provided a set A of m points and a distance algorithm, k-nearest neighbor (kNN) search encounters the m nearest points in A to a query point or set of points. kNN-based functions are generally used as a standard machine learning guideline.

Model Optimization

After a model is chosen, the model can be tuned by selecting different advanced options. Some of these options are internal parameters of the model, or hyperparameters, that can strongly affect its performance. Instead of manually selecting these options, hyperparameter optimization can be used to automate the selection of hyperparameter values. For a given model type, it tries different combinations of hyperparameter values using an optimization scheme that seeks to minimize the model classification error, and returns a model with the optimized hyperparameters. The resulting model can be used for testing.

Misclassification Cost Specification

By default, the trained models will assign the same penalty to all misclassifications during training. For a given observation, the model will be assigned a penalty of 0 if the observation is classified correctly and a penalty of 1 if the observation is classified incorrectly. In some cases, this assignment is inappropriate. For example, suppose you want to classify patients as either healthy or sick. The cost of misclassifying a sick person as healthy might be five times the cost of misclassifying a healthy person as sick. For cases where you know the cost of misclassifying observations of one class into another, and the costs vary across the classes, specify the misclassification costs before training your models. For applying misclassification cost, a cost matrix must be specified where the row and column labels determine the classes of the response variable. The rows of the table consist of the true modules, and the columns direct to the predicted modules. So, the entry in row i and column j is the cost of misclassifying ith class observations into the j^{th} class. The diagonal entries of the cost matrix must be 0, and the off-diagonal entries must be non-negative real numbers. Hence, the cost matrix can be customized and then the model will be trained according to the machine learning problem.

Experimental Setup

The entire setup of an industry is mimicked by a working kitchen where the sound coming from a pressure cooker is similar to the sound of a leak. Also, the machine noises, process waters and the continuous industrial noises are represented by the mixture noise, tap water and the continuous noise of utensils, burning flames, etc. respectively. Here, the leaking is assumed to be at one point. In the mimicked setup, a microphone is used to capture the audio. In an industrial setup, audio sensors should be installed at regular in the entire plant to capture the audio and use it for gas leak detection. For our experiment, 100 data are collected containing leak and normal sound which have been further augmented to 300 data. When the trained model is tested using the running data which are captured by a sensor placed in the kitchen and then the signals are transmitted to a laptop via IoT medium. In the laptop, necessary features are extracted and these features are tested by the trained model present in the laptop which predicts the condition of the captured signal. The experimental setup is shown in Fig. 4. The model has been trained in an Intel core i3 7th Gen laptop. This model can be transferred to any device without any cost. For installation of the device, we need a sound sensor (Rs. 200), rechargeable batteries (Rs. 158), DC–DC converter (Rs. 449), solar panel (Rs 500), battery management system (Rs. 240). So, the overall cost would be Rs. 1547. However, the cost will be adjusted when used in large scale.

Results and Discussion

Rank of Features

After applying MRMR in the extracted features, a graph is plot to show the rank of the features and their score and the plot is shown in Fig. 5. The y-axis of the plot defines the MRMR score of the features and x-axis represents the features which is sorted on the basis of their importance in training the model.

Feature Visualization

The features extracted from the abnormal and normal data store must be visualized before training as it shows the extent of overlap between the different states. It also shows the scattering of the features as well. When PCA is applied to the feature matrix, a matrix where the rows are the data points and the columns are the features, it gives another matrix where each column of that matrix includes coefficients for

Fig. 4 Mimicked experimental setup for pressurised system's leak detection

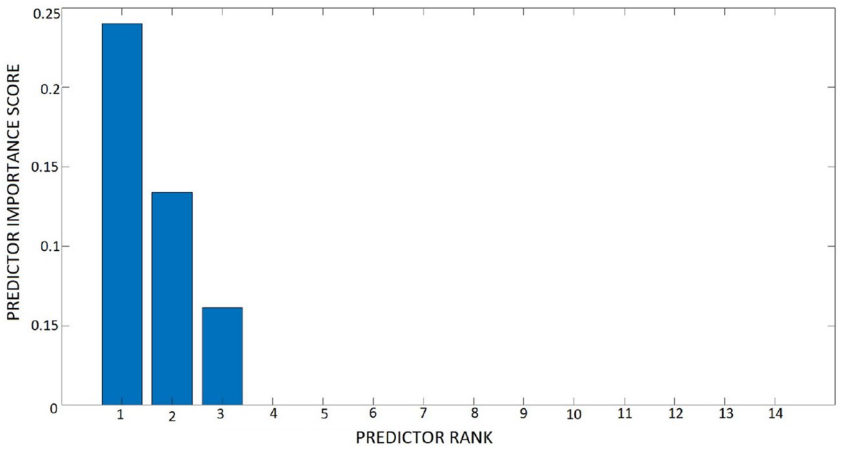

Fig. 5 The importance score of the extracted features as obtained by applying MRMR

each principal component, and the columns are in decreasing order of component variance.

PCA, in general, uses the singular value decomposition (SVD) algorithm and centers the total data. Further, the total variance explained by each principle component is demonstrated using a bar graph in descending order of their values. The bar graph obtained is shown in Fig. 6. The features are also visualized using a scatter plot which is a simple plot of one PCA variable against another and it is shown in Fig. 7.

Testing Output

After experimenting with various classifiers and their hyper-parameters, this paper proposes three different classifiers which give the maximum training and testing accuracy.

Four model parameters are determined for each model which are correct and incorrect prediction of the trained

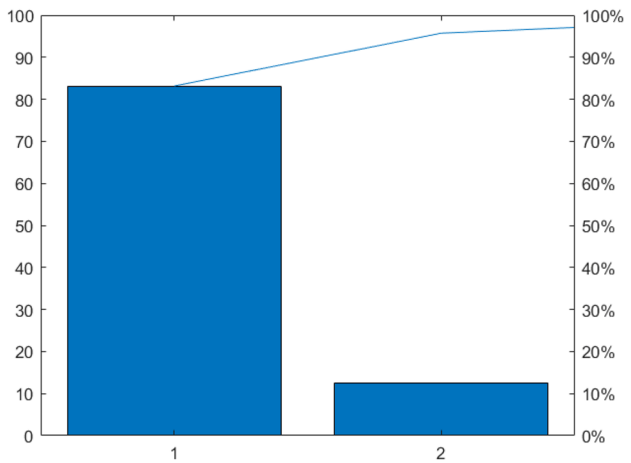

Fig. 6 Principle Component Analysis of the feature matrix

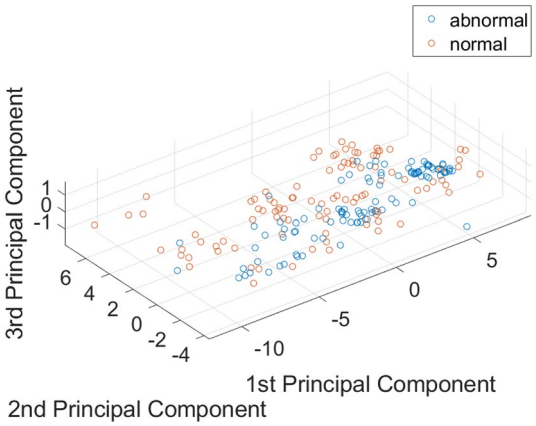

Fig. 7 Scatter plot of PCA variables of the features extracted from abnormal and normal data

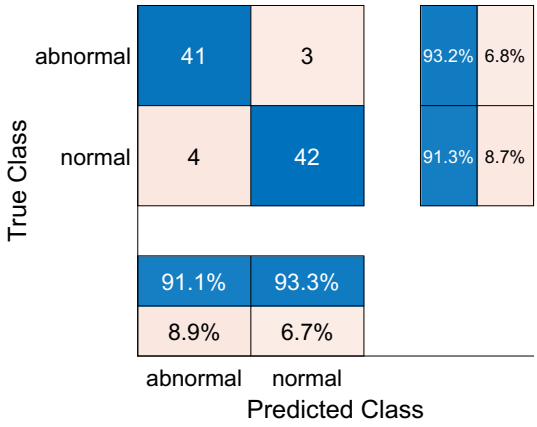

Fig. 8 Testing result of Naive Bayes model shown using Confusion matrix; the model is trained with 15 features and is tested on 90 data

model, training accuracy, confusion matrix and testing accuracy.

The testing results are shown using Confusion matrix. A confusion matrix is plotted for the true title targets and predicted title outputs. The titles are abnormal and normal in this paper. On the confusion matrix chart, the rows indicate the predicted class (Output Class) and the columns indicate the true class (Target Class). The diagonal cells indicate observations that are properly identified. The off-diagonal cells correspond to improper identifies observations. Both the number of observations and the percentage of the total number of observations are displayed in each cell. The column on the extreme right of the plot demonstrates the percentages of all the examples predicted to belong to each class that are properly and improperly identified. These metrics are often known as the precision (or positive predictive value) and false discovery rate, respectively. The row at the bottom of the plot displays the percentages of all the examples of each class that are properly and incorrectly identified. These metrics are often called the recall (or true positive rate) and false negative rate, respectively. The cell in the bottom right of the plot displays the total accuracy.

The first model is Naive Bayes which gives an initial training accuracy of 87.6% . After applying optimization, the accuracy does not improve. Further, the cost matrix is changed where the cost of predicting normal to abnormal is changed to 3 from 1. Then, the model shows a training accuracy of 90.5%. The test result of the final Naive Bayes is shown in Fig. 8.

The second model that gives proper training and testing accuracy is SVM. Important features are selected using MRMR algorithm and these four important features out of total fifteen features are used for machine learning training. The model has a final training accuracy of 92.4% with those four features. Testing accuracy of SVM is shown in Fig.9.

The final model is KNN classifier which gives an initial accuracy of 90% with four important features of total fifteen features. After optimization, it increases to 91.4%. Testing accuracy of model is shown in Fig. 10.

Figure 11 shows the minimum classification error plot that updates the model as the optimization runs.

Table 1 depicts the correct and false prediction of the 210 training data by the model trained by machine learning.

We can observe that SVM and KNN model are better than Naive Bayes in training and testing accuracy. However, the SVM and KNN model is developed from only four features, whereas Naive Bayes has used the entire fifteen features of the audio data. Therefore, Naive Bayes can predict correctly regarding any range of audio data recorded by the sensor. In certain conditions like leak 2 pressure 2 and leak 3 pressure 2, the model shows a testing accuracy of 83.28% and 90% respectively in [16]. However, the accuracy of the model of

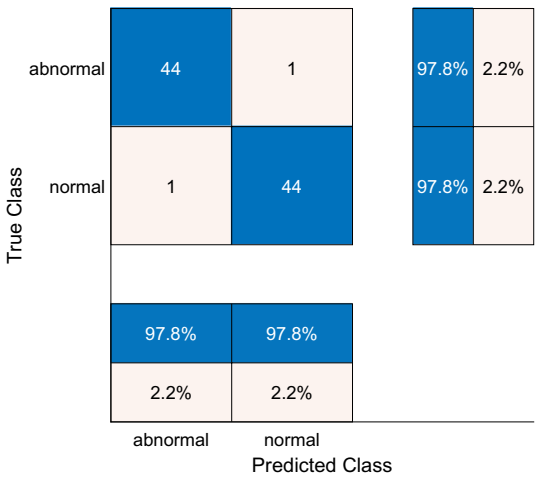

Fig. 9 Testing result of SVM model shown using Confusion matrix; the model is trained with 4 features and is tested on 90 data

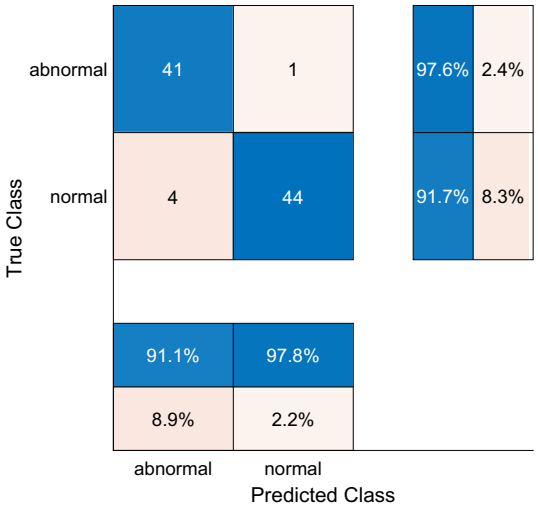

Fig. 10 Testing result of KNN model shown using Confusion matrix; the model is trained with 4 features and is tested on 90 data

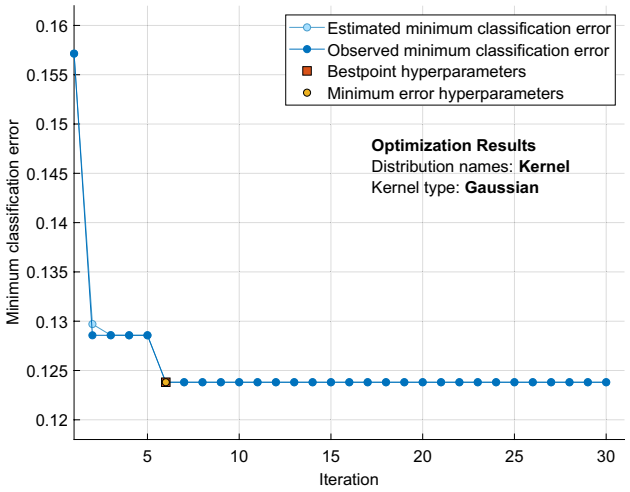

Fig. 11 Minimum classification error plot

Table 1 Correct and false prediction of the data by different trained models

Model name	Correct	False
Naive Bayes initial model	184	26
Naive Bayes final model	190	20
SVM final model	194	16
KNN initial model	192	18
KNN final model	193	17

our paper, instead of being developed in a very practical and noisy kitchen environment, is noteworthy which also makes it a very versatile model (Fig. 12).

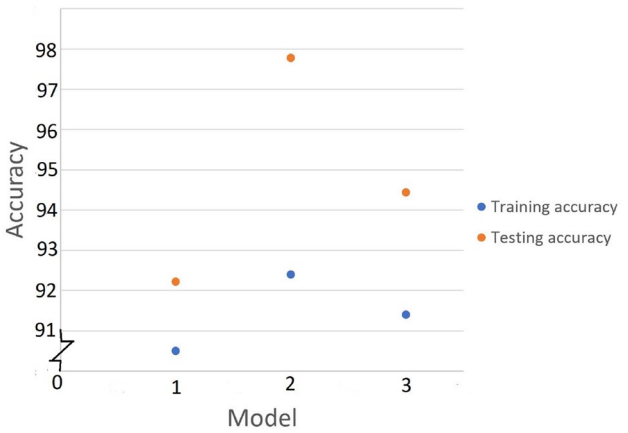

Fig. 12 Training and testing accuracy of different models;1=Naive Bayes,2=SVM,3=KNN

Table 2 Comparison of training accuracy of different trained models

Model name	Training accuracy	Testing accuracy
Naive Bayes model	90.5	92.2
SVM model	97.78	92.4
KNN model	91.4	94.44

Table 2 shows the training and testing accuracy of all the three models.

Conclusion and Future Works

This paper has focused on leak detection on industrial pipelines with the help of IoT-based model development. The accuracy of the models denotes that leak can be detected directly from the AE signals without any particular transformations of the signals. Implementation of IoT will significantly increase the safety of the pipelines by alerting the industrial officials about the condition of the signals generated by the pipelines. The entire experiment is conducted in a working kitchen where varieties of noise are present all the time which presents a similar kind of situation as that seen in any working industry. Therefore, the model developed by the captured signals can be used for predicting leaks in any practical conditions which makes the model more robust and effective. Hence, the trained model can be used for implementing a robust and effective leak detection system for the industries. Further, future works can be done to make the IoT-based leak detection model more effective. Instead of using hamming window, other windows can be used for feature extraction. Increasing data set can help in the formation of a more robust model. Implementation of other features can also be proved helpful. Deep-learning and

wavelet scattering are some methods that can be used for developing the model. Unsupervised learning can be used as a method to detect abnormal leaks.

Declarations

Conflict of interest The authors declare that they have no conflict of interest and there was no human or animal testing or participation involved in this research. Authors have collected the data experimentally.

References

1. Aboussaleh I, Riffi J, Mahraz AM, Tairi H, Brain tumor segmentation based on deep learning’s feature representation. J Imaging 2021; 7(12) . https://doi.org/10.3390/jimaging7120269. https://www.mdpi.com/2313-433X/7/12/269
2. Baiji Y, Sundaravadivel P, iLoLeak-Detect: An IoT-Based LoRAWAN-Enabled Oil Leak Detection System for Smart Cities. In: 2019 IEEE International Symposium on Smart Electronic Systems (iSES)(Formerly iNiS), 2019; pp. 262–267. IEEE
3. Caputo AC, Pelagagge PM. Using neural networks to monitor piping systems. Process Saf Progress. 2003;22(2):119–27.
4. Ding C, Peng H. Minimum redundancy feature selection from microarray gene expression data. J Bioinform Comput Biol. 2005;3(02):185–205.
5. Dreiseitl S, Ohno-Machado L. Logistic regression and artificial neural network classification models: a methodology review. J Biomed Inform. 2002;35(5–6):352–9.
6. Gao Y, Brennan M, Joseph P, Muggleton J, Hunaidi O. On the selection of acoustic/vibration sensors for leak detection in plastic water pipes. J Sound Vib. 2005;283(3–5):927–41.
7. Hossan, M.A., Memon, S., Gregory, M.A.: A novel approach for MFCC feature extraction. In: 2010 4th International Conference on Signal Processing and Communication Systems, 2010; 1–5 https://doi.org/10.1109/ICSPCS.2010.5709752
8. Kim JH, Kim BG, Roy PP, Jeong DM. Efficient facial expression recognition algorithm based on hierarchical deep neural network structure. IEEE Access. 2019;7:41273–85. https://doi.org/10.1109/ACCESS.2019.2907327.
9. Liu Z, Kleiner Y. State of the art review of inspection technologies for condition assessment of water pipes. Measurement. 2013;46(1):1–15.
10. Lu D, Liu T. The application of IOT in medical system. In: 2011 IEEE International Symposium on IT in Medicine and Education, 2011; 1, 272–275 https://doi.org/10.1109/ITiME.2011.6130831
11. Mahmud MA, Bates K, Wood T, Abdelgawad A, Yelamarthi K. A complete internet of things (IoT) platform for structural health monitoring (SHM). In: 2018 IEEE 4th World Forum on Internet of Things (WF-IoT), 2018; 275–279. IEEE
12. Mandal SK, Chan FT, Tiwari M. Leak detection of pipeline: An integrated approach of rough set theory and artificial bee colony trained SVM. Expert Syst Appl. 2012;39(3):3071–80.
13. Martini A, Troncossi M, Rivola A. Vibroacoustic measurements for detecting water leaks in buried small-diameter plastic pipes. J Pipeline Syst Eng Pract. 2017;8(4):04017022.
14. Nicola M, Nicola C, Vintila A, Hurezeanu I, Dută M. Pipeline leakage detection by means of acoustic emission technique using cross-correlation function. J Mech Eng Autom. 2018;8:59–67.
15. Puust R, Kapelan Z, Savic D, Koppel T. A review of methods for leakage management in pipe networks. Urban Water J. 2010;7(1):25–45.
16. Quy TB, Muhammad S, Kim JM. A reliable acoustic EMISSION based technique for the detection of a small leak in a pipeline system. Energies. 2019;12(8):1472.
17. Santos R, De Sousa E, Da Silva F, Da Cruz S, Fileti A. Detection and on-line prediction of leak magnitude in a gas pipeline using an acoustic method and neural network data processing. Brazilian J Chem Eng. 2014;31(1):145–53.
18. Tonheim J, Tveit R. Acoustic emission on-line monitoring of the ammonia plant NII secondary reformer exit (gas channel). Process Saf Progress. 1997;16(2):101–4.
19. Wang F, Lin W, Liu Z, Qiu X. Pipeline leak detection and location based on model-free isolation of abnormal acoustic signals. Energies. 2019;12(16):3172.
20. Wang F, Lin W, Liu Z, Wu S, Qiu X. Pipeline leak detection by using time-domain statistical features. IEEE Sens J. 2017;17(19):6431–42.
21. Wang S, Dong L, Wang J, Wang H, Ji C, Hong J. Experiment study on small leak detection and diagnosis for propulsion system pipelines of sounding rocket. IEEE Access. 2020;8:8743–53.
22. Xiao Q, Li J, Bai Z, Sun J, Zhou N, Zeng Z. A small leak detection method based on VMD adaptive de-noising and ambiguity correlation classification intended for natural gas pipelines. Sensors. 2016;16(12):2116.
23. Xu Q, Zhang L, Liang W. Acoustic detection technology for gas pipeline leakage. Process Saf Environ Protect. 2013;91(4):253–61.
24. Yazdekhasti S, Piratla KR, Atamturktur S, Khan AA. Novel vibration-based technique for detecting water pipeline leakage. Struct Infrastruct Eng. 2017;13(6):731–42.
25. Zadkarami M, Shahbazian M, Salahshoor K. Pipeline leakage detection and isolation: an integrated approach of statistical and wavelet feature extraction with multi-layer perceptron neural network (MLPNN). J Loss Prevent Process Ind. 2016;43:479–87.
26. Zanella A, Bui N, Castellani A, Vangelista L, Zorzi M. Internet of things for smart cities. IEEE Internet Things J. 2014;1(1):22–32. https://doi.org/10.1109/JIOT.2014.2306328.
27. Zhamanov A, Sakhiyeva Z, Suliyev R, Kaldykulova Z: IoT smart campus review and implementation of IoT applications into education process of University. In: 2017 13th International Conference on Electronics, Computer and Computation (ICECCO), 2017; 1–4. https://doi.org/10.1109/ICECCO.2017.8333334