Machine Learning in AWS for IoT-based Oil Pipeline Monitoring System

Sabreen J. Siwan*
Control and Systems Engineering Department, University of Technology, Baghdad, Iraq.
E-mail: cse.19.04@grad.uotechnology.edu.iq

Waleed F. Shareef
Control and Systems Engineering Department, University of Technology, Baghdad, Iraq.

Ahmed R. Nasser
Control and Systems Engineering Department, University of Technology, Baghdad, Iraq.

Received September 16, 2021; Accepted December 15, 2021
ISSN: 1735-188X
DOI: 10.14704/WEB/V19I1/WEB19209

Abstract

The world's economy is dominate by the oil export business, which is heavily reliant on oil pipelines. Due to the length of the pipes and the harsh environment through which they pass, continuous structural health monitoring of pipelines using normal methods is difficult and expensive. In this paper, an IoT system integrated with cloud services is propose for oil pipeline structure monitoring. The system is based on collecting data from sensor nodes attached to the pipeline structure, which collectively form a network of IoT devices connected to the AWS cloud. Measurements from sensor nodes are collect, stored, and filtered in AWS cloud. Measurements are also make accessible to users through the internet in real-time using Python web framework, Flask, and sending alarms via email in real-time. The performance of the system is evaluate by applying damaging events (hard knocking) on the oil pipeline at several distances. Analysis of IoT data by machine learning classification algorithms, apply and comparison between SVM, Random Forest Classifier, and Decision Tree to determine the best one, and then built in EC2 Linux in AWS to analyses the measurements and classify new events according to their distances from the sensor nodes. The proposed system is test on field measurements that were collect in Al-Mussaib Gas Turbine Power Station in Baghdad. Among the three classifiers, Random Forest achieved 90% classification rate.

Keywords

IoT, Machine Learning, AWS, ESP32cam, Monitoring Oil Pipeline.

Introduction

The process of transporting petroleum products is carry out through a network of pipelines that transports petroleum products (oil and gas). The pipelines are a safe and cost-effective mode of transportation. However, risk concerns like terrorism and sabotage, corruption, "Hot-Zones" pose problems (Layth et al., 2019; Prasanta, 2004). Pipeline spills have averaged 76,000 barrels per year, or more than 3 million gallons, since 1986. This equates to 200 barrels of oil each day, resulting in almost 500 deaths (biological diversity, 2021).

Monitoring and maintaining oil pipelines have a direct impact on the economy, as well as any danger of leakage or pipe breakage, leading to pipeline collapse and, as a result, irreparable damages such as financial and human losses, as well as severe environmental pollution, especially when the leak is not detected in a timely manner. Hence, it is necessary to have a monitoring system that effectively and in real-time can determine the fault location to facilitate the maintenance and replacement process without affecting the export operation and thus the economy of the country (Laith et al., 2019). Traditional means, such as regular foot patrols and aircraft observation with small planes or helicopters are difficult to implement and cost-ineffective due to the length of the pipes and the severe climate surrounding the pipeline. Although providing a high level of security, periodic monitoring in traditional means requires extensive human intervention, which makes it more expensive. So it is required to use systems that can properly and in real-time monitor oil pipelines (Nasheed & Waleed, 2018). As an example of recent work on traditional monitoring methods, an unmanned aerial vehicle's aerodynamic study (UAV) was present in (Ernesto et al., 2020). The UAV can be equipped with a tiny infrared camera to monitor oil leaks in a pipeline network utilizing oil-related infrared radiation, it is high cost, Monitoring is for a certain period and not in real time has Legal constraints. Smart pigs use all another traditional system using smart PIG, to detect and analyze prospective issues, Transmitters, sensors, GPS, eddy current, magnetic fields, ultrasonic, and acoustics. However, it is a rather expensive treatment. Pipeline monitoring and inspection by pigs has been estimate to cost up to $56,000 per kilometer of pipeline (Oil and Gas IQ, 2019, 2015). after the planning is complete, the trained crew may need hours to properly load the pig into the pipe, and the running distance will only be a few kilometers. In (Rahul et al., 2020) designing a system for monitoring and controlling gas pipelines using the PLC and SCADA to improve in locating the leakage. If there is leakage, it will automatically shut off the process for a particular area or whole system and control by manual. The buzzer will alarm if there is any pressure drop. However, this system is a High initial cost and a skilled person is required to handle the operation.

In addition, used fiber optics for monitoring oil pipeline in (Liang et al.,2018) a new application based on the OFDR approach was present to monitor pipeline corrosion and leaks by detecting dispersed hoop strain over the pipeline's outer surface. The findings demonstrate this technology has a lot of potential in pipeline safety monitoring because it allows for high efficiency and accuracy systems. However, because this research is still in its early stages, more research is need, such as a more accurate leakage localization approach. In addition, Fiber optics is expensive and difficult to install.

On the other hand, the emergence of the ubiquities and wireless sensor networks in the last decade benefitted this industry and enabled high-quality monitoring systems at a reasonable cost, in real-time, without human intervention. In (Nasheed & Waleed ,2018), a WSN monitoring system was propose, aided with algorithms for detecting and categorizing damaging events. Several low-cost sensor nodes each consisting of a microcontroller, 3-axis accelerometer, and RF transceiver were connected wirelessly to transfer the sensor measurement to a central station for processing. A similar system is present in (Ahmed & Waleed., 2020). To overcome the problem of saturating the RF transceiver with a large amount of data transmission, the microcontroller unit was replace by a microcomputer unit to analyze the data locally. Each node contains high computational In addition to the 3-axis accelerometer, there is a 1.2GHz quad-core ARM Cortex-A53 (64Bit) processor for in-situ data processing and RF unit. Every two neighboring nodes filter and analyze the data collected at each node locally in real time. Only the estimation findings were later send to a base station supervisor for display. However, WSN based systems are still bounded by power limitation which limits their computational ability and its intra-communication. An IoT framework designed to track and manage the industrial environment was present in (Alaa et al., 2019). The system uses gas and temperature sensors connected to and Node MCU. If the temperature rises or falls above or below the pre-set thresholds, the air conditioning would be turn on automatically. Similarly, if the number of toxic gases rises, the fan device will be enable. The system tracks and manages the environment wirelessly using a mobile application built on the Blynk platform.

Oil companies have been looking for ways to minimize cost, and integration and automation are two of the easiest ways to do so. It is aiming to reduce human error and gain real-time insights from the wide range of sensors found on a petroleum pipeline, an IoT solution that can connect all of these data threads has been developed as a realistic choice (Network World, 2019). The advent of the Internet of Things (IoT) technology in pipeline monitoring makes it effective with the introduction of embedded sensing systems, whose primary requirement is to perform accurate and real-time measurements (Shuruq et al., 2020).

The cloud computing technology is appropriate for the construction of a service platform for the Internet of Things, The major advantage of the cloud is that it elastically scales to meet fluctuating demand and provides an environment that scales up and down immediately in response to need and makes data available and easy access online all time. The usage of cloud vendors' IAAS and PAAS services minimizes the difficulty of system development and makes the system more stable and cost effective.

The capacity to determining real-time damage location utilizing a combine of machine learning classification algorithms, a wireless data transmission system based on IoT, and cloud computing technologies, gives this research study a competitive advantage and contributes to knowledge. It is the design and implementation of an oil pipelines monitoring system, where it will be possible to know the status of the pipes remotely through a web application, Backed by an alarm message that reaches the mobile via e-mail. As a result, pipelines may be monitored in real time from anywhere on the world, Leakage detection and localization, reducing cost, easy and intelligent system, reduce energy consumption because the analysis and storage data did in the cloud and in real-time, without human intervention. To the author's knowledge, past pipeline monitoring research has rarely is real-time transmission and monitoring emphasized of damaged data via Wi-Fi to an IoT or supported it by cloud computing. This paper presents Petroleum pipeline monitoring using the internet of things, cloud computing additionally, utilize an ESP32-cam with a built-in Wi-Fi-module to create a wireless device that can gather vibration signals from accelerometer sensors on pipes during a real-world measurement test and send the data to Amazon Web Services (AWS) for storage, organizing, and analytics.

The rest of the paper has been organize as follows: Sections 2 research Methodology, section3 illustrate the experiment setup and data collection. Section 4 presented the machine-learning algorithm used. On the other hand, sections 5 show the results and discussion .and the conclusion in section 6.

Material and Method

The proposed system consists of three layers: perception, network, and application layers. The perception layer is the hardware foundation of the system. It consists of the structure being monitored (Oil pipeline), the sensors (accelerometers), and the microcontroller board (ESP32). The network layer represents the communication protocol (Wi-Fi and MQTT) through which the nodes are connected and to the cloud. Finally, the Application layer consists of AWS cloud services ((IoT core, DynamoDB, Lambda, API gateway, EC2, IAM management rule). The machine learning part of the work is build inside the Lambda and

EC2 services, as explained later. These layers are in Fig. 1 and are discussed thoroughly in the next sections.

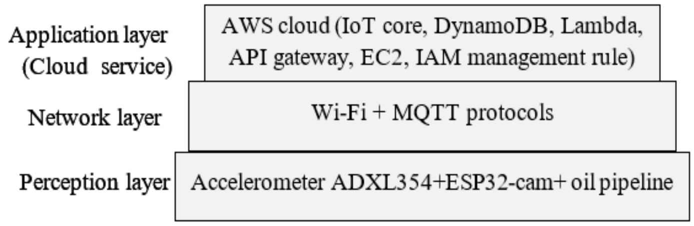


Fig. 2 Architecture of IoT-based oil pipeline Monitoring System

A. Perception Layer Design (Node Design)

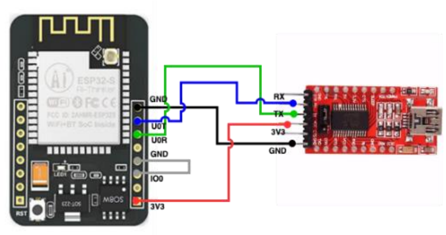
The perception layer consisted of the following commercially available components: ADXL 345 accelerometer as vibration sensor, ESP32-cam Wi-Fi Module as the mainboard microcontroller, FTDI adapter for microcontroller programming, and finally, battery and lithium charge shield. The ADXL345 is a small, thin integrated IC, (Shown in Figure 2). It has a very low power consumption of measurement mode, it uses only 23 µA and in standby mode, it uses only 0.1 µA. The ADXL345 is a three-axis accelerometer with a 10-bit resolution at 2, 4, and 8 G (G = gravity acceleration, which is approximately 9.807m/s2), as well as 13-bit data at ranges up to 16 G. In 2's complement format, the sensor digital output is compress into two bytes. The sensor's digital interface is either SPI or I2C. The output data rate can range from 0.1 to 3200 Hz [4], which is ideal for the sampling rate needed for the proposed application. The ADXL345 is an accelerometer sensor that Due to its reliable operation, precision, and sensitivity, is frequently used for defect detection in various machinery, and it provides low noise levels (Ferdinando et al., 2013*)*. It is cheap, available in the market and, easy to connect the sensor to the controller it is only used four cables.

Fig. 2 ADXL345 accelerometer

ESP32-CAM is one type of Node MCU a lightweight, low-power camera module. It features an OV2640 camera and a TF card slot onboard. It is a full-featured microcontroller. It is inexpensive and easy to use and is perfect for IoT devices. The ESP32-CAM board has no USB port, so it is need to use an FTDI adapter for programming the board (espressif, 2021). ESP32 is shown in Fig. 3-a, while the connection to the FTDI is shown in Fig. 3-b.

The final node in Fig. 3-c and it consists of a node consisting of ESP32cam, battery, and lithium charge shield. The node size is less than 2×2×6 cm3. The sensor is connect later since it is need to be attach directly to the pipe.

(a)



(b)

(c)

Fig. 3 IoT node for monitoring oil pipelines
(a) ESP32-CAM, (b) connection of ESP32cam to FTDI, (c) Final node with sensor and battery

B. Network layer (Communication Protocols)

- This layer defines the communication technology and communication protocol used in the proposed system. Wi-Fi is the popular name for wireless local area networks based on the IEEE 802.11b standard (Paul & Hui, 2003). It is use as communication technology (it is a built-in MCU node). Message Queue Telemetry Transport MQTT Protocol is a standardized publish/subscribe Push protocol (Hamid & Sara, 2018). It is the most commonly used IoT application layer protocol, which is design to ensure excellent communication between low power and resource-constrained devices (Hawraa & Hamid, 2020), a lightweight, designed for low bandwidth. MQTT's features make it an excellent option for sending high volumes of sensor messages to analytics platforms and cloud solutions (Sabrine et al., 2021). It is use to transfer the sensor's measurement to the cloud.

C. Application layer (Cloud Computing)

At the application level, which interacts with the end-user, the application layer is responsible for delivering services and determining protocols for message passing (Saba et al., 2017). The services used in this work are:

IoT Device: Without the need to provision or manage servers, AWS IoT Core allows you to connect IoT devices to the AWS cloud. It can handle billions of devices and trillions of communications, and it can reliably and securely process and route those messages to AWS destinations and other devices.

- DynamoDB: Amazon DynamoDB is a NoSQL database. It is allows developers to create modern, serverless apps that scale internationally to support petabytes of data and tens of millions of read and write requests per second.
- EC2: Amazon Elastic Compute Cloud is a cloud computing service that provides secure, resizable compute capacity. Its goal is to make web-scale cloud computing more accessible to developers (Amazon, 2021).

Figure 4 shows the flow of sensor measurement from the sensor to the remote user. The sensor measurement is collect and stored in an IoT node, periodically, then transferred through MQTT protocol to the AWS cloud. First, the data is stored in the IoT core, and then stored in the DynamoDB. At each new batch of data in DynamoDB, the software in EC2 is trigger. The software in EC2 is divide into two programs, one to display the raw data measurement through the web application. The other one performs machine learning on the collected data.

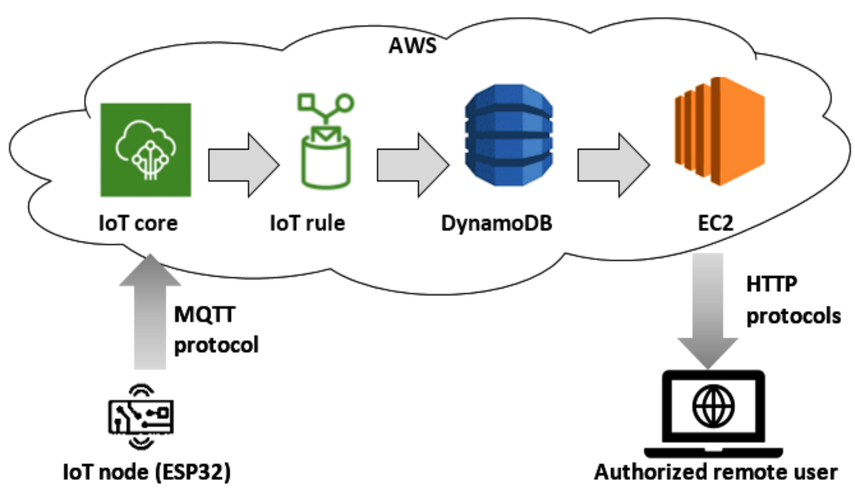


Fig. 4 Data flow through the proposed system

Experiment Setup and Data Collection

The proposed system was test on data field measurements were collect in Al-Mussaib Gas Turbine Power Station in Baghdad. IoT nodes were fix on the live but empty oil pipeline. The two nodes (Node1 and Node2) are 50 meters apart. The pipe was then subject to a damaging event (manual knocking by a hammer at a rate of 1 knock per second or slower to simulate a tampering action). The event was repeat at several distances: 1, 12, 24, 36, and 48 meters from Node1. During the test, accelerometer measurements were continually record at both IoT nodes. The ADXL345 accelerometer at each node was place horizontally on the top of the oil pipeline, so the X-axis is perpendicular to the pipe, and the Y-Z plane is parallel to the earth's surface as shown in Fig. 5. Then, the measurements were normalize so that the vibrational signal amplitude ±16G corresponds to ±1. Normalization of the features is a general requirement for many machine-learning algorithms (Ian & Frank, 2016). The X-axis measurements must be adjust by 1G to account for the earth's gravity acceleration.

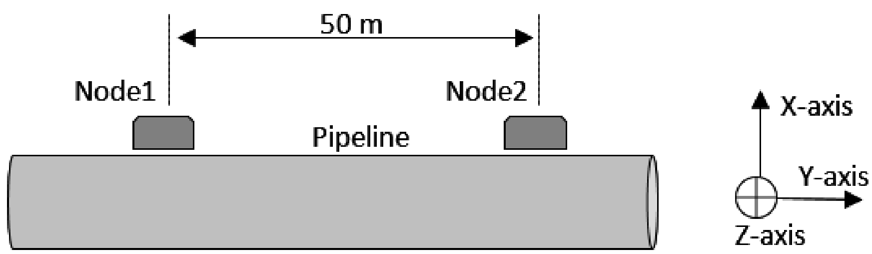


Fig. 5 Node placement on the pipeline

Fig. 6 shows a segment of the recorded vibration signal for approximately 6 seconds. The Blue, Red, and Green graphs correspond to X-axis, Y-axis, and Z-axis measurements respectively. The readings in the first five seconds represent the background noise of the system, i.e., the normal state measurements. After the sixth second, the pipeline was subject to five consecutive knocks with periods of one to two seconds between one knock and another. In the first 100 msec after each knock, the sensor reading is saturated which means the acceleration is outside the range ±16G, then the signal is quickly tapered off to the background noise level within a second. It is worth noticing that since the knocking action is parallel to the X-axis, the X-axis signal (Blue graph) tapers off more slowly than the other two axes' signals.

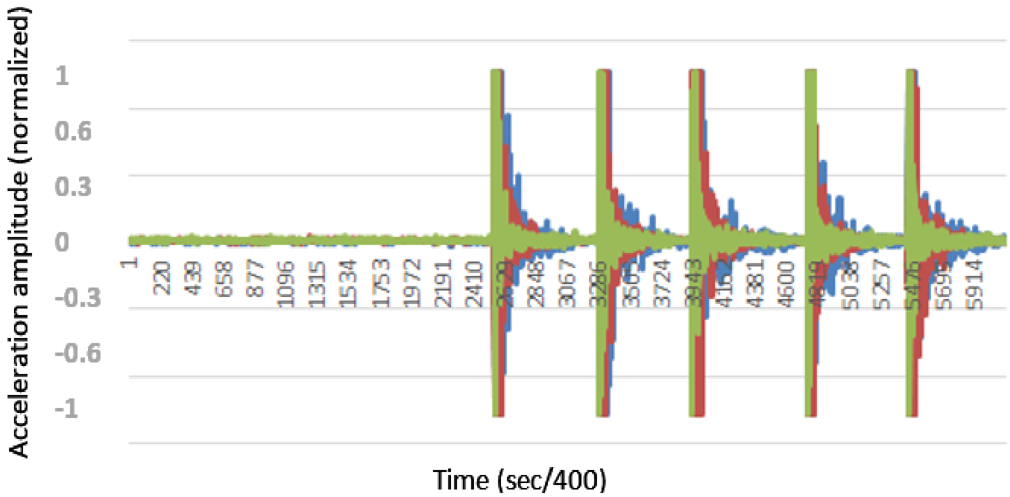


Fig. 6 The segment of normalized vibration signal

Machine learning

Machine learning is a growing branch of computational algorithms that aim to mimic human intelligence by learning from the environment. Machine learning techniques have been effectively apply in a variety of fields, the most important of which is data mining. People are prone to making mistakes when doing studies or attempting to create correlations between different features. This makes it more difficult for them to come up with solutions to difficulties. Machine learning may often be successfully used to these issues, enhancing system efficiency and machine design (Issam et al., 2015).

In this paper, machine learning (supervised) is implement to determine the distance of damaging events on the pipeline by classifying vibrational signals based on the damaging event distances from the node. Three widely used machine-learning algorithms are implement in python and deployed in EC2 AWS to classify the measurements in the proposed system. These methods are Support Vector Machine (SVM), Decision Tree, Random Forest Classifier, and. A brief description of these methods and their benefits are stated next subsection.

A. Machine Learning Algorithms in the Proposed System

SVM stands for supervised machine learning. It is primarily use to overcome problems with categorization each data item was represented as a point in n-dimensional space (where n is the number of features), with the value of each feature being the SVM algorithm's value for a certain coordinate. Then we classify the data by selecting the hyper-plane that clearly separates the two classes (Carolina et al., 2012).

Decision tree is a supervised learning technique usually used to solve classification problems. Internal nodes represent dataset attributes, branches represent decision rules, and each leaf node provides the conclusion in this tree-structured classifier (ISSAM et al., 2015).

Random forest classifier is a supervised learning algorithm mainly used for classification problems. It creates decision trees on data samples, then gets the prediction from each of them, and finally selects the best solution utilizing voting. An ensemble method is better than a decision tree because it reduces over-fitting by averaging the result (Mariana & Lucian, 2016).

B. Data Sets and Classification Classes

Measurements were collect while applying damaging events at several distances from the IoT nodes for classifying them with machine learning algorithms data was divide into sets. Each set represents the measurements (acceleration along X, Y, and Z-axes) from the two nodes for approximately 10 to 15 seconds (4000 to 6000 measurements, each measurement consists of three axes readings). Four sets of data were record at each distance for repeatability. In addition to four sets of data were recorded without a damaging event (without knocking) to represent the normal (healthy) state of the pipeline. This makes the total amount of data sets is 24 as explained in Eq. 1

$$\text{Total Data set} = (\text{number of event distances} + \text{no event case}) * 4 = (5+1) * 4 = 24 \text{ sets} \qquad (1)$$

To make sure that the algorithms are apply correctly two cases were consider. In Case1 only sets from three distances (1, 12, and 24 meters) were used in addition to the healthy state sets, which means that data should be classified into four classes. In Case2, all data sets were consider (1, 12, 24, 36, 48 meters) in addition to the healthy state data sets, which means that data should be, classify into six classes. If the algorithms were apply correctly, we would expect to see that Case1 gave better classification results than Case2. This is because adding two more classes that represent data further away from the nodes would negatively affect the classification. Finally, the data set was divide into 90% for learning and 10% for testing.

Result and Discussion

Machine learning using the three algorithms is compare to each other based on two parameters, accuracy and F1-score, which is the result of Confusion matrix parameters. In the confusion matrix, there are four parameters: True positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). The accuracy and F1-score parameters are calculate as in Equations (2)-(5). The parameter value for each algorithm is in Table 1.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (2)$$

$$Precision = \frac{TP}{TP+FP} \quad (3)$$

$$Recall = \frac{TP}{TP+FN} \quad (4)$$

$$F1\ score = \frac{2\times Precision\times Recall}{Precision+Recall} \quad (5)$$

Table 1 Accuracy and F1 score for classification algorithms

Algorithm	**Case 1 (4 classes)**		**Case 2 (6 classes)**	
	Accuracy	**F1 score**	**Accuracy**	**F1 score**
SVM	0.65	0.52	0.62	0.39
Decision Tree Classifier	0.83	0.76	0.79	0.64
Random Forest Classifier	0.90	0.86	0.87	0.76

Comparing the results of Case1 and Case2 shows better classification accuracy in Case1 for the same method for each method. This is expect since generally fewer classes lead to better classification as well as the two additional classes in Case2 contain results from farther events, which leads to ambiguous classification. It is worth noticing that Random Forest classification accuracy with six classes is better than all the other methods with four and six classes.

Figure 7 shows the confusion matrix of the Random Forest Classifier results in Case1. The columns represent the predicted class and the rows represent the true class. Random Forest Classifier achieved high testing accuracy in two cases with a mean accuracy of 90%, 87% respectively.

The matrix shows that the classification of closer distances is far better than the classification of farther distances. This suggests that the proposed sensor, MCU, and the analysis algorithm are barely sufficient for classifying the signal with 50 meters between every two adjacent nodes. This can be directly address by deploying the system with distances less than 50 meters between nodes or using more advanced sensors and/or analysis algorithms.

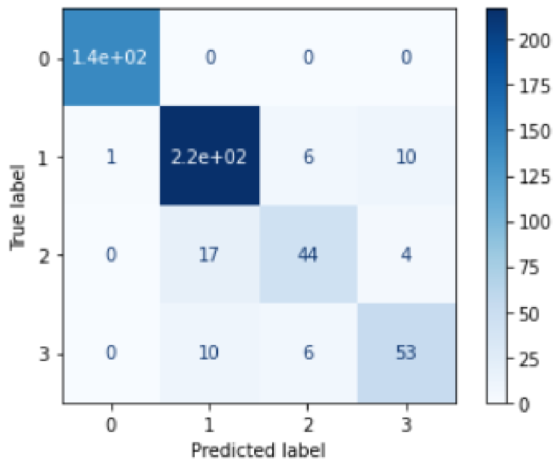


Fig. 7 Confusion matrix of Decision Tree Classifier

Finally, the prediction result is make available to authorized users through a web application as shown in Figure 8. The web page shows other information such as the maximum, minimum, mean value of the readings also plots the readings in a chart. The web page automatically refreshes every five seconds. The web page also displays the most recent batch of raw measurements. The Green, Red and Yellow graphs correspond to X-axis, Y-axis, and Z-axis measurements respectively.

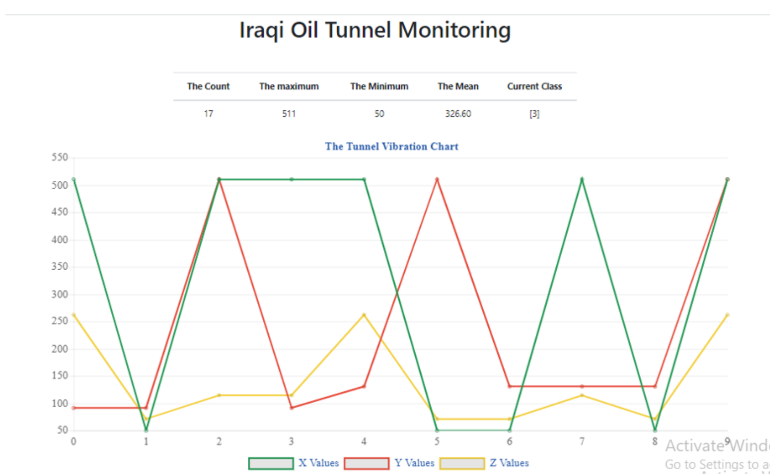


Fig. 8 Web application for a monitor oil pipeline

In web applications can see multiple columns are, the count is the count of vibration readings. The maximum and minimum are a maximum reading and minimum of readings, which are in range (+511, -512). The mean is the mean of readings and the current class is the state of the pipe according to the class number. At the same time, in case of damaged pipe only as in figure 9, alarm was send via email on mobile.

Fig. 9 The state of oil pipelines by email service

Conclusion

This paper presented a systematic walkthrough for designing an IoT-based oil pipeline monitoring system with cloud-based data storage and classification. Although wireless networks that are make from small-size low-cost commercial off-the-shelf components have been around for almost a decade, these networks are easily overwhelm by big data generated from monitoring systems designed for a large structure like an oil pipeline. The availability of commercial cloud (like AWS) that provide not only the Platform as a Service but also the Software as a Service, made it a great companion with IoT-based monitoring system. Incorporating cloud-based services' storage and processing alleviate the heavy burden on the nodes limited resources. The system is low-cost and unlimited in terms of storage and processing of large data based on the computing service. The web interface and its event log review are two of the most useful elements of our proposed system. It may be access securely from any linked terminal and can send email alerts, implying that cloud technology makes data available online.

The propose system, with the use of technology and artificial intelligence, can assist the oil industry in conducting, such inspections with minimal losses in terms of staff safety, time, cost, and other factors. A low-cost pipeline monitoring system that detects damage in real time, location, and the ability to view the findings of a pipe case in real time on a web application from anywhere in the world, as well as alarm issuing via email.

Finally, machine-learning algorithms are deploy in AWS-EC2 to determine the event distance from the IoT node. Several algorithms are write in Python 2.7.18 inside the EC2 service. Among the selected algorithms, Random Forest Classifier showed best results in term of classification accuracy and F1-score, which is expect since it is the most advanced algorithm, which based on ensemble learning.

References

Kraidi, L., Shah, R., Matipa, W., & Borthwick, F. (2019). Analyzing the critical risk factors associated with oil and gas pipeline projects in Iraq. *International Journal of Critical Infrastructure Protection, 24,* 14-22.

Dey, P.K. (2004). Decision support system for inspection and maintenance: a case study of oil pipelines. *IEEE transactions on engineering management, 51*(1), 47-56.

https://www.biologicaldiversity.org/campaigns/americas_dangerous_pipelines/

Fawzi, L.M., Alqarawi, S.M., Ameen, S.Y., & Dawood, S.A. (2019). Two Levels alert verification technique for smart oil pipeline surveillance system (SOPSS). *International Journal of Computing and Digital Systems, 8*(02), 115-124.

Mossa, N.F., Shareef, W.F., & Shareef, F.F. (2018). Design of Oil Pipeline Monitoring System based on Wireless Sensor Network. *Iraqi Journal of Computers, Communication and Control & Systems Engineering, 18*(2), 53-62.

Elvira-Hernández, E.A., López-Huerta, F., Vázquez-Leal, H., Hernández-Escobedo, Q., & Herrera-May, A.L. (2020). Aerodynamic analysis of an unmanned aerial vehicle with infrared camera for monitoring oil leakage in pipeline networks. *Acta universitaria,* 30.

Oil and Gas IQ (2019). *Strategy, management and information, IIoT in oil and gas.* https://www.oilandgasiq.com/content-hub/industrial-internet-of-things

Oil and Gas IQ (2015). *Unmanned inspection for oil and gas: ROVs, pigs, ROI and obstacles.* https://www.oilandgasiq.com/strategy-management-and-information/whitepapers/analysis-unmanned-inspection-for-oil-gas

Joshi, R.S., Patil, P.P., Dalavi, R.N., Patil, A.C., & Joshi, S.K. (2020). A Review on Monitoring and Control Parameters of Gas Pipeline by Using PLC and Scada System. *International Journal of Engineering Applied Sciences and Technology, 4*(12), 628-631.

Ren, L., Jiang, T., Jia, Z. G., Li, D. S., Yuan, C. L., & Li, H. N. (2018). Pipeline corrosion and leakage monitoring based on the distributed optical fiber sensing technology. *Measurement, 122,* 57-65.

Hadi, A.H., & Shareef, W.F. (2020). In-Situ Event Localization for Pipeline Monitoring System Based Wireless Sensor Network Using K-Nearest Neighbors and Support Vector Machine. *Journal of Al-Qadisiyah for computer science and mathematics, 12*(3), 1-15.

Jaber, A.A., Al-Mousawi, F.K.I., & Jasem, H.S. (2019). Internet of things based industrial environment monitoring and control: a design approach. *International Journal of Electrical & Computer Engineering (2088-8708), 9*(6), 4657-4667.

Gold, J. (2019). *"How the oil and gas industry exploits IoT".* Network World. https://www.networkworld.com/article/3445204/how-the-oil-and-gas-industry-exploits-iot.html

Hashim, S.A., Jawad, M.M., & Wheedd, B. (2020). Study of Energy Management in Wireless Visual Sensor Networks. *Iraqi Journal of Computers, Communications, Control and Systems Engineering, 20*(1), 68-75.

Ferdinando, H., Khoswanto, H., & Purwanto, D. (2013). Performance evaluation of MMA7260QT and ADXL345 on self-balancing robot. *Telkomnika, 11*(1), 1-10.

www.espressif.com https://loboris.eu/ESP32/ESP32-CAM%20Product%20Specification.pdf

Henry, P.S., & Luo, H. (2002). WiFi: what's next? *IEEE Communications Magazine, 40*(12), 66-72.

Hasan, H. M., & Jawad, S. A. (2018). IoT Protocols for Health Care Systems: A Comparative Study. *International Journal of Computer Science and Mobile Computing, 7*(11), 38-45.

Abdul Ameer, H.R., & Hasan, H.M. (2020). Enhanced MQTT Protocol by Smart Gateway. *Iraqi Journal of Computers, Communications, Control and Systems Engineering, 20*(1), 53-67.

Khriji, S., Benbelgacem, Y., Chéour, R., Houssaini, D.E., & Kanoun, O. (2021). Design and implementation of a cloud-based event-driven architecture for real-time data processing in wireless sensor networks. The Journal of Supercomputing, 1-28.

Al-Rubaye, S., Kadhum, E., Ni, Q., & Anpalagan, A. (2017). Industrial internet of things driven by SDN platform for smart grid resiliency. *IEEE Internet of Things Journal, 6*(1), 267-277.

aws.amazon.com. https://aws.amazon.com/ar/products/

Frank, E. (2016). *Data Mining: Practical Machine Learning Tools and Techniques* Ed. 4. Elsevier Science.

El Naqa, I., Li, R., & Murphy, M.J. (Eds.). (2015). Machine learning in radiation oncology: theory and applications. Springer.

Crisci, C., Ghattas, B., & Perera, G. (2012). A review of supervised machine learning algorithms and their applications to ecological data. Ecological Modelling, 240, 113-122.

Belgiu, M., & Drăguţ, L. (2016). Random forest in remote sensing: A review of applications and future directions. *ISPRS journal of photogrammetry and remote sensing, 114*, 24-31.