JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Fault Diagnosis in Rotor Blades using ML and IOT

Mohnish Salian[1], Rajesh B. Buktar[2]

[1,2]*Department of Mechanical Engineering, Sardar Patel College of Engineering, Mumbai-400058, India.*

Abstract

Condition monitoring of machines is gaining importance in industry because of the need to increase reliability and to decrease possible loss of production due to machine breakdown. The use of vibration and acoustic emission (AE) signals is quite common in the field of condition monitoring of rotating machinery. By comparing the signals of a machine running in normal and faulty conditions, detection of faults like mass unbalance, rotor rub, shaft misalignment, gear failures and bearing defects is possible. These signals can also be used to detect the incipient failures of the machine components, through the on-line monitoring system, reducing the possibility of catastrophic damage and the machine down time. A study is presented to compare the performance of rotor fault detection using two different classifiers, namely, artificial neural networks (ANNs) and support vector machines (SMVs). The RPM, current, voltage and lift force of a rotating machine with normal and defective rotor are processed for feature extraction. The extracted features from preprocessed signals are used as inputs to the classifiers for two-class (normal or fault) recognition. Further, this pretrained model is deployed on a cloud platform to carry out predictions over the cloud. This gives mobility to the system which allows us to monitor condition from anywhere. With the help of HTML, an interface is set up so user can interact with the instance. The web-based interface lets user enter values of required feature and outputs the condition of the rotor blade.

Keywords: Fault Diagnosis; Rotor Blades; kernel Support Vector Machine; Artificial Neural Network.

1. Introduction

In modern production, fault diagnosis techniques of mechanical equipment have increasingly gained important. If a device failure is not discovered and eliminated timely, it may cause serious damage or, in some cases, a breakdown. Therefore, the importance of fault diagnosis in the production line or mechanical systems should not be neglected. Rotor blades are present in almost every vehicle or device that flies. Some examples are; helicopters, drones, plane engines, wind turbine. Being a rotating component, rotor blades are subjected to fatigue stress. It is important to give a fast and accurate detection of the existence of a fault in an installation during the operation process, since rotor blades are crucial part of the system and an unexpected failure may cause serious damage to the entire system, and in some cases to life and property. Therefore, detection capability of fault diagnosis systems must be improved. The early failure diagnostic methods include the use of hearing, shock pulse and resonance demodulation technologies. However, the accuracy and efficiency of these diagnostic methods cannot reach the standard of the industry level.

With the continuous development of diagnostic techniques, artificial intelligence, such as expert system, artificial neural network (ANN), fuzzy logic, immune genetic algorithm has been widely used in machine fault diagnosis. SVM and ANN have been widely used in fault detection in rotating elements such as bearings and gears [1][2]. Apart from these two algorithms, genetic algorithm and optimization technique such as ant colony optimization have also been used in the past [3][4]. CNN based method yields an overall good classification accuracy, without relying on extensive domain knowledge for detecting faults [5]. With the help of Big Data, Digital Manufacturing & Supply chain, IOT can improve productivity in the manufacturing industry [6]. Studies have been conducted to review the impact of vibration analysis with the help of SVM to detect faults in wind turbines [7]. Vibration analysis has been widely used in fault diagnosis and feature extraction, the corresponding time-frequency analysis methods have been applied on the fault diagnosis and CM. These methods can achieve prospective effects and play important roles, such as the EMD method in diagnosing wind turbine blades crack [8]. A vibration analysis in frequency domain coupled with fuzzy system has been tested to detect structural damage [9]. Another study explored the impact of vibratory hub load along with SVM in detecting faults in composite helicopter rotor blade [10]. SVM based techniques have also been explored by certain studies

[11][12]. Furthermore, a data-mining based approach with Genetic Programming algorithm has also been studied in the past [13]. Studies show that the neural network can detect and quantify both single and multiple faults on the blade from noise-contaminated simulated vibration and blade response test data [14]. Detection based on the measurements of the noise emitted by a UAV were used to build a classification model to detect unbalanced blades in a UAV propeller [15]. The detection and identification of sensor failure in RUAV has been investigated along with the design of a sensor fault diagnosis system with neural network based adaptive threshold [16].

Most of the studies conducted in the past explore the effect of vibration in time domain and frequency domain. The purpose of this paper is to study the effect of RPM, lift force, current and voltage in training a ML model to detect faults in rotor blades, and extract appropriate features based on dependency. This study also evaluates the efficiency of two machine learning models namely, kernel SVM and ANN, to identify which model gives higher accuracy in detecting faults. This work presents an IOT system for fault diagnosis in rotor motor system, combined with a cloud solution for classification of the operational state of rotor blades. For this study, a prototype setup of rotor motor system with 4 sensors; current, voltage, RPM and lift force is used. The setup is run with rotor blades induced with certain defects, and the sensor readings are recorded. Finally, we use these readings to train the above-mentioned machine learning models to test accuracy of each model. Furthermore, this model is deployed on cloud platform to test the effectiveness and responsiveness of the model on cloud.

2. Machine Learning

SVM algorithms use a set of mathematical functions that are defined as the kernel. The function of kernel is to take data as input and transform it into the required form. Different SVM algorithms use different types of kernel functions. These functions can be different types. For example; linear, nonlinear, polynomial, radial basis function (RBF), and sigmoid. Introduce Kernel functions for sequence data, graphs, text, images, as well as vectors. The most used type of kernel function is RBF. Because it has localized and finite response along the entire x-axis. The kernel functions return the inner product between two points in a suitable feature space. Thus, by defining a notion of similarity, with little computational cost even in very high-dimensional spaces.

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it learn for themselves. Supervised machine learning algorithms can apply what has been learned in the past to new data using labeled examples to predict future events. Starting from the analysis of a known training dataset, the learning algorithm produces an inferred function to make predictions about the output values.

2.1. Kernel Support Vector Machine

SVM algorithms use a set of mathematical functions that are defined as the kernel. The function of kernel is to take data as input and transform it into the required form. Different SVM algorithms use different types of kernel functions. These functions can be different types. For example; linear, nonlinear, polynomial, radial basis function (RBF), and sigmoid. Introduce Kernel functions for sequence data, graphs, text, images, as well as vectors. The most used type of kernel function is RBF. Because it has localized and finite response along the entire x-axis. The kernel functions return the inner product between two points in a suitable feature space. Thus, by defining a notion of similarity, with little computational cost even in very high-dimensional spaces. Gaussian kernel maps all points in the graph space to a certain height. The height is greater as we move towards the centre. This helps in separating data point belonging to one category that are encompassed by data set of a different category.

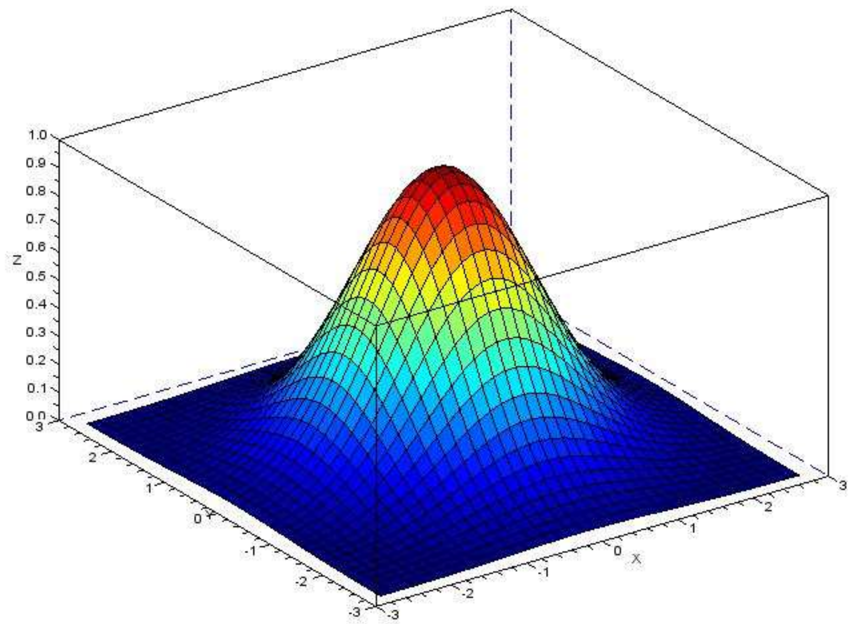

Fig. 1. James D. McCaffrey. Gaussian Kernel, 2014, jamesmccaffrey.wordpress.com

$$K(x, y) = e^{\frac{-(xi-l)^2}{2\sigma^2}} \tag{1}$$

here, K - mapping function
x - coordinate
σ - standard deviation
l - mean

2.2. ANN Model

Artificial Neural Networks or ANN is an information processing paradigm that is inspired by the way the biological nervous system such as brain process information. It is composed of large number of highly interconnected processing elements(neurons/nodes) working in unison to solve a specific problem. Biological Neurons (also called nerve cells) or simply neurons are the fundamental units of the brain and nervous system, the cells responsible for receiving sensory input from the external world via dendrites, process it and gives the output through Axons. A single layer neural network is called a Perceptron. It gives a single output. The Activation function is important for an ANN to learn and make sense of something really complicated. Their main purpose is to convert an input signal of a node in an ANN to an output signal. This output signal is used as input to the next layer in the stack. Activation function decides whether a neuron should be activated or not by calculating the weighted sum and further adding bias to it. The motive is to introduce non-linearity into the output of a neuron. If we do not apply activation function then the output signal would be simply linear function (one-degree polynomial). Now, a linear function is easy to solve but they are limited in their complexity, have less power. Without activation function, our model cannot learn and model complicated data such as images, videos, audio, speech, etc.

Neural networks require a trainer in order to describe what should have been produced as a response to the input. Based on the difference between the actual value and the predicted value, an error value also called Cost Function is computed and sent back through the system. Cost Function is one half of the squared difference between actual and output value. For each layer of the network, the cost function is analyzed and used to adjust the threshold and weights for the next input. Our aim is to minimize the cost function. The lower the cost function, the closer the actual value to the predicted value. In this way, the error keeps becoming marginally lesser in each run as the network learns how to analyze values. We feed the resulting data back through the entire neural network. The weighted synapses connecting input variables to the neuron are the only thing we have control over. As long as there exists a disparity between the actual value and the predicted value, we need to adjust those weights. Once we tweak them a little and run the neural network again, A new Cost function will be produced, hopefully, smaller than the last. We need to repeat this process until we scrub the cost function down to as small as possible.

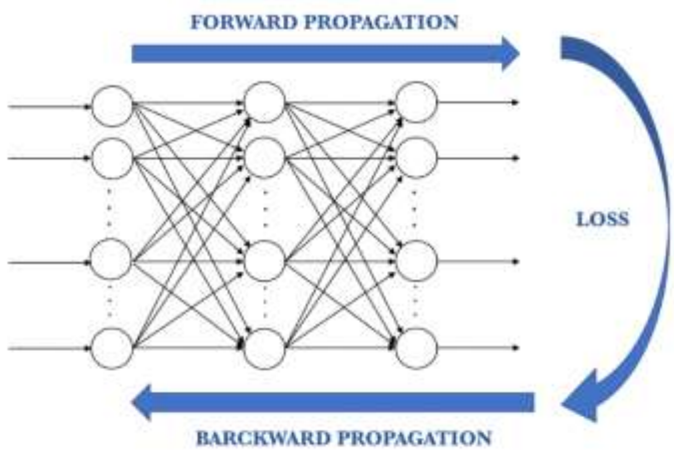

Fig. 2. Jordi Torres. Propagation in ANN, 2018, www.towardsdatascience.com

3. Internet of Things

Internet of Things allows us to remotely manage and monitor various aspects of a system. To maximize operation efficiency, the sensor network has to be carefully designed. Specifications of appropriate sensos which provide the required data must be chosen carefully. Apart from sensors,

3.1. Selection of IOT Sensors

Numerous types of sensors have been developed. Selection of sensors is a crucial part in designing an IOT system. Without proper sensors, it's difficult to capture the entire function of a system. Two aspects should be considered when selecting a sensor; specification notification which describes the sensor's detailed features and a formal visualized representation of the sensor information [20]. A three-sieve selection tool has been developed, which is a straightforward method for sensor selection that balances performance requirements, environmental constraints, and other factors with economic considerations [21]. The sieve method simplifies analysis by progressively focusing attention on then sensors most likely to perform as needed. It has been found that ILP solution is optimal for the service matchmaking but consume more time which may not be scalable in large-scale IoT systems [22]. It helps to raise the level of abstraction to program various IOT sensors. A study has shown that multi-objective evolutionary algorithm (MOEA) helps in selection of sensors [23].

3.2. Selection of IOT Platform

A cloud IoT platform must monitor IoT endpoints and event streams, analyze data at the edge and in the cloud, and enable application development and deployment. These are the essential functions required for virtually any IoT implementation.

4. Methodology

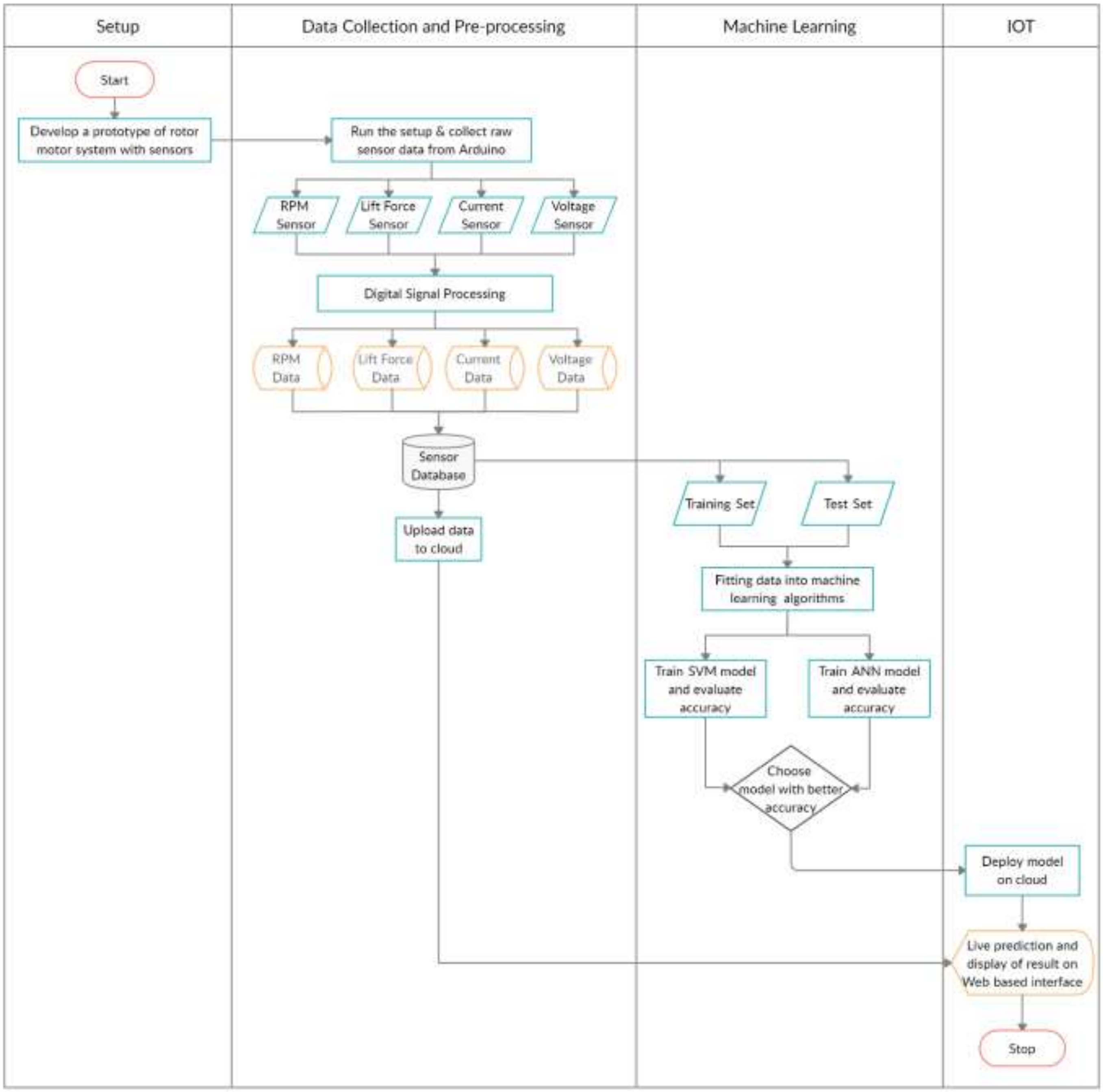

Fig. 3. Methodology followed in study

The work is divided into four sections;

1. Setup: A prototype is developed for the purpose of the experiment.
2. Data collection and pre-processing: This step involves collecting data from the sensors installed in the setup and applying pre-processing techniques such as digital signal processing and sensor data fusion.
3. Machine Learning: The next step is to train machine learning models and test the accuracy of the models.
4. IoT: The final step is to create an instance on cloud and to upload the pretrained models to the cloud platform.

5. Experiment Setup

The below picture shows the setup for the experiment. It consists of the propeller connected to a motor, which is attached to a wooden plank with four screws. The wooden plank is hinged to a vertical wooden piece which in turn is attached to the base. As shown in the picture, the setup is put down and set in position. Under the wooden plank is a metal cantilever which has a 3D printed piece attached to it. This piece is connected to a displacement measuring sensor. This sensor is used to measure force generated by the propeller. Since displacement is directly proportional to force in a cantilever, the movement of the beam can be directly translated to force. A hall sensor is attached to the motor. It measures RPM of the motor. The motor RPM can be varied with the help of RPM controller which sits on the wooden plank. Apart from the sensors visible in the picture, the setup also consists of current and voltage sensor. The setup is housed inside a thick acrylic enclosure. The sensors are connected to an Arduino. The Arduino collects sensor data, preprocesses them and sends them to the computer. The data is stored in csv format on the computer. This data is used to train the machine learning model.

Fig. 4. Setup top view

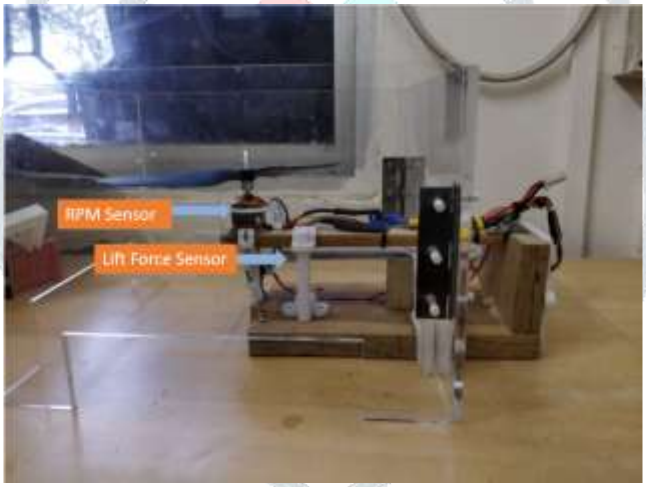

Fig. 5. Setup side view

5.1. Data Collection

Four rotor blades with unique defects have been used for this experiment. Blade chip, blade wear, bent blade and loosely fitted blade are the four defects introduced in the blades. Each rotor blade is mounted on the setup and the motor is started. Sensor data from each sensor is collected by the Arduino and is stored in a computer in csv format. As mentioned earlier current, voltage, RPM and lift force are captured by these sensors. This data undergoes preprocessing before training the model. The data generated is continuous and analogous. Digital signal processing is required to obtain discrete data. A sequence of samples from a measuring device produces a temporal or spatial domain representation, whereas a discrete Fourier transform produces the frequency domain representation. In DSP, digital signals are analyzed in one of the following domains: time domain (one-dimensional signals), spatial domain (multidimensional signals), frequency domain, and wavelet domains. In this study, an upper bound and lower bound threshold values are set. All data points outside the boundaries are filtered out. The remaining dataset undergoes pre-processing required for the ML training phase.

5.2. Data Pre-processing for kernel SVM

The first part of any machine learning model is importing the dataset. We do this with the help of pandas library. The input data is stored in the X variable and the output data in the Y. The next part is encoding. Since the output data is in the form of string, its needs to be converted into numbers. These numbers don't hold any value and hence they are called dummy values. The algorithm assigns numbers to each distinct string.

Before

Type
Normal
Normal
Normal
Normal
Normal
Loosely Fitted
Loosely Fitted
Loosely Fitted
Loosely Fitted
Loosely Fitted
Damaged
Damaged
Damaged
Damaged
Damaged

After

Type
2
2
2
2
2
1
1
1
1
1
0
0
0
0
0

Fig. 6. Label Encoder

The next part is to split the dataset into train and test. Here, the train set holds 75% of the data and the test set holds 25%. The input data has multiple features. These features have different range of values. To normalize the data, standard scaling is applied. This method standardizes features by removing the mean and scaling to unit variance. The standard score of a sample x is calculated as:

$z = (x - u) / s$

where u is the mean of the training samples or zero if with mean=False, and 's' is the standard deviation of the training samples or one if with std=False.

Centering and scaling happen independently on each feature by computing the relevant statistics on the samples in the training set. Mean and standard deviation are then stored to be used on later data using transform.

5.3. Training the kernel SVM Model

After importing the SVC, we can create new model using the predefined constructor. This constructor has many parameters. Some of the parameters are:

- Kernel: The kernel type to be used. The most common kernels are 'rbf '(this is the default value), poly or sigmoid, but you can also create your own kernel.
- C: This is the regularization parameter described in the Tuning Parameters section.
- Gamma: This was also described in the Tuning Parameters section.
- Degree: It is used only if the chosen kernel is poly and sets the degree of the polynomial
- Probability: This is a Boolean parameter and if it's true, then the model will return for each prediction, the vector of probabilities of belonging to each class of the response variable. So basically, it will give you the confidences for each prediction.
- Shrinking: This shows whether or not you want a shrinking heuristic used in your optimization of the SVM, which is used in Sequential Minimal Optimization.

5.4. Data Pre-processing and training of ANN

In the data preprocessing stage, we only perform standard scaling. Encoder is not required here. In this model, there are 6 neurons in the input layer and the activation function if 'relu'. After the input layer, there is one hidden layer with 6 nodes or neurons. In the output layer consists of 3 neurons with 'sigmoid' activation function.

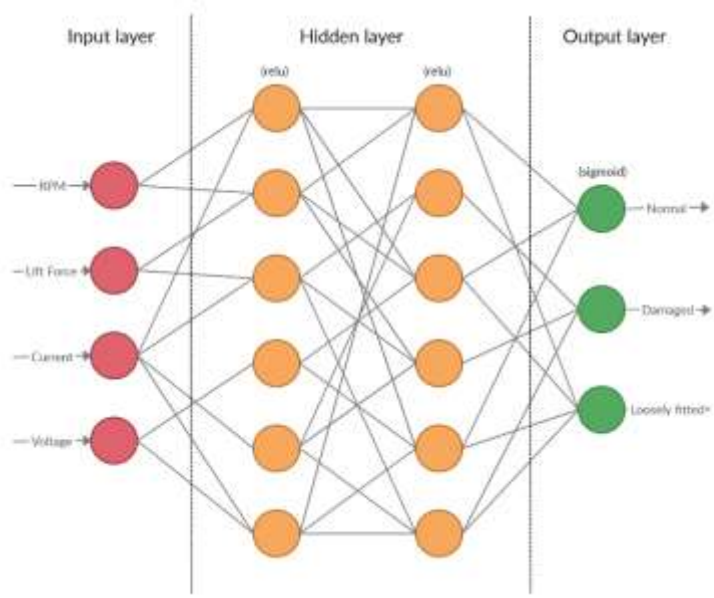

Fig. 7. ANN Network

6. Cloud Framework

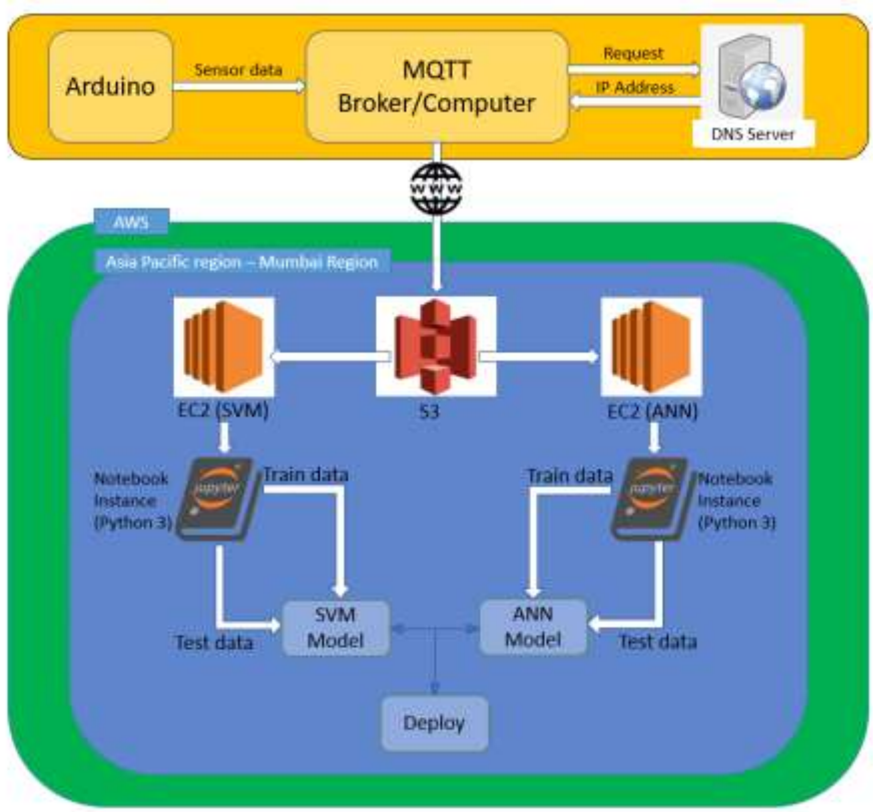

Fig. 8. Label Encoder

After running the setup and collecting data for kernel SVM and ANN, we have trained the two models based on the two algorithms. The next step is to integrate the setup with cloud platform. This involves migrating the model and deploying it on cloud. AWS (Amazon Web Services) is one of the most well-known cloud platforms with features such as easy scaling and quick deployment. In this experiment, we use AWS to deploy our machine learning model. AWS has many services out of which we require two. The S3 service lets us store data, so we use this service to store our pretrained model and data collected from the Arduino. Also, any future data collected will directly transfer into the S3 bucket. The EC2 service is computing services in the cloud. We deploy our model using this service. We created an EC2 instance for our cloud-based machine learning predictions. Instances consist of Amazon Machine Images (AMI) which are pre-configured templates that wraps everything you need for your server including the operating system. We followed all the required steps in the process of creating an instance, from setting up security to installing all the necessary packages on the server. We transferred the pretrained model to S3 bucket and accessed the model from the instance. For demo purposes, the server needs a front end which was written in HTML. Below is an image of the interface. The server was linked to the interface. Once the model is deployed, the two fields provided can be filled with respective values picked up by the sensors and the page will output the condition of the rotor blades.

Fig. 9. HTML GUI

7. Results and Discussion

The last step of the algorithm is to predict results using the test data and compare them to the real results. This is done using the .pred() method. Using the 'confusion matrix' function we can generate a matrix which compares the actual and predicted results.

True value → / Prediction ↓	Normal	Loosely fitted	Damaged
Normal	138	23	0
Loosely fitted	6	174	0
Damaged	0	0	176

Fig. 10. Label Encoder

The numbers in the green cells indicate correct predictions and the ones in the orange cells indicate incorrect predictions. The accuracy of the model is 94.39%.

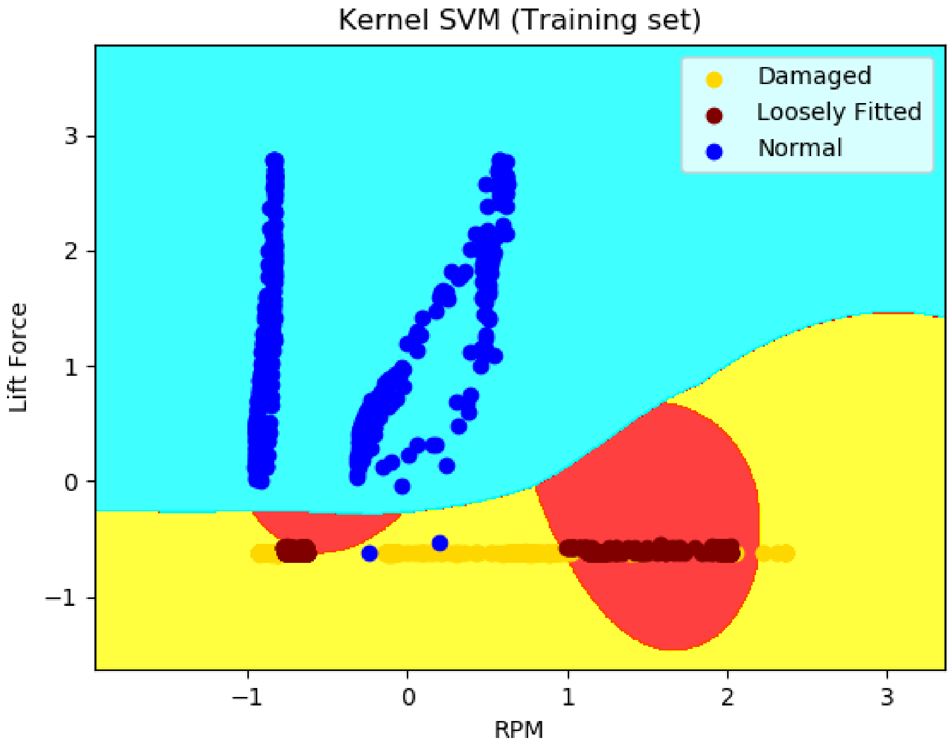

Fig. 11. Kernel SVM training set plot

With an accuracy of approximately 94%, the model manages to predict almost every result correctly. But as evident from the graph, this is due to over-fitting. Overfitting happens when a model learns the detail and noise in the training data to the extent that it negatively impacts the performance of the model on new data. This means that the noise or random fluctuations in the training data is picked up and learned as concepts by the model. The problem is that these concepts do not apply to new data and negatively impact the model's ability to generalize.

Overfitting is more likely with nonparametric and nonlinear models that have more flexibility when learning a target function. As such, many nonparametric machine learning algorithms also include parameters or techniques to limit and constrain how much detail the model learns.

Over time, as the algorithm learns, the error for the model on the training data goes down and so does the error on the test dataset. If we train for too long, the performance on the training dataset may continue to decrease because the model is overfitting and learning irrelevant detail and noise in the training dataset. At the same time the error for the test set starts to rise again as the model's ability to generalize decreases.

The sweet spot is the point just before the error on the test dataset starts to increase where the model has good skill on both the training dataset and the unseen test dataset.

True value → Prediction ↓	Normal	Loosely fitted	Damaged
Normal	111	20	0
Loosely fitted	0	117	0
Damaged	0	0	144

Fig. 12. ANN Confusion Matrix

With a slight difference of 0.5% compared to SVM accuracy, the accuracy of the ANN model is 94.89%. This again goes to show that overfitting has taken place. As discussed earlier, it is possible to avoid overfitting. But in our case, it is not possible train a good machine learning model since the data generated from the setup during experimentation is inaccurate.

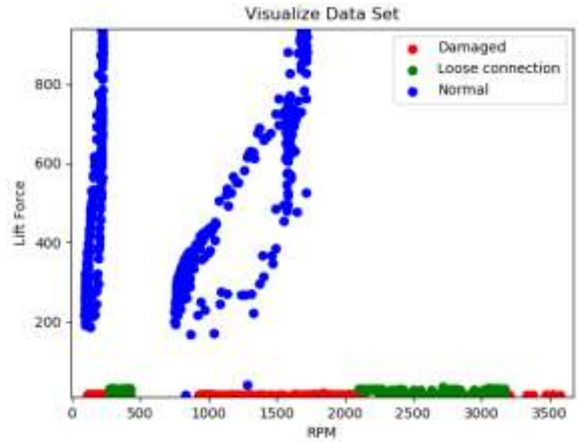

Fig. 13. Trainset plot

From the graph it is clear that the lift force generation is non uniform and does not follow any curve. As for the damaged and loosely fitted propellers, the lift force is zero. It can be seen that the data collected from the sensors is inaccurate.

8. Conclusion

This paper presents comparison of efficiency of two machine learning models in detecting faults in rotor blades, namely, kernel SVM and ANN. A prototype setup with Arduino was used in this experiment. Current, voltage, RPM and lift forces are the features used in models training. The model developed is then deployed on AWS server for cloud-based fault detection. For the purpose of this study, the server portal was connected to web portal with web interface developed using HTML. Inputting sensor values generates a result indicating whether the rotor has sustained any damage or not. In case the rotor has sustained any damage, the model will specify the type of damage. In terms of accuracy, ANN performs better than kernel SVM with an accuracy of 94.89%. Although based on observation, the SVM plot show signs of overfitting. This is due to low variance and can be avoided with accurate experiment readings. There is much room for improvement in the maintenance domain, and machine learning paired with internet of things can help us bridge this gap.

References

[1] L. B. Jack, A. K. Nandi (2001). Fault Detection using support vector machine and artificial neural network, augmented by genetic algorithm. Mechanical Systems and Signal Processing (2002) 16(2–3), 373–390.

[2] B. Samanta (2004). Gear fault detection using artificial neural networks and support vector machines with genetic algorithm, Mechanical Systems and Signal Processing 18 (2004) 625–644.

[3] Xu Li a, A'nan Zheng b (2013). Rolling element bearing fault detection using support vector machine with improved ant colony optimization, Measurement 2726–2734.

[4] P.K. Kankar, Satish C. Sharma (2011). Fault diagnosis of ball bearings using machine learning methods, Expert Systems with Applications 1876–1886.

[5] Olivier Janssen, Viktor Slavkovikj (2016). Convolutional Neural Network Based Fault Detection for Rotating Machinery. Journal of Sound and Vibration.

[6] Laura Contreras, Juan Perez. (2018). Importance of the Use of the Internet of Things and its Implications in the Manufacturing Industry. European Scientific Journal, 14, 1857–7881.

[7] Nassim Laouti, Nida Sheibat-Othman, Sami Othman (2011). Support Vector Machines for Fault Detection in Wind Turbines. The International Federation of Automatic Control Milano (Italy) August 28 - September 2, 2011

[8] W.Y. Liu, B.P. Tang (2014). The structure healthy condition monitoring and fault diagnosis methods in wind turbines: A review. Renewable and Sustainable Energy Reviews 44 (2015) 466–472.

[9] Prashant M. Pawar, Ranjan Ganguli (2003). Genetic fuzzy system for damage detection in beams and helicopter rotor blades. Comput. Methods Appl. Mech. Engrg. 192 (2003) 2031–2057.

[10] Prashant M. Pawar and Sung Nam Jung. (2008), Support Vector Machine based Online Composite Helicopter Rotor Blade Damage Detection System. Journal of Intelligent Material Systems and Structures, Vol. 19.

[11] Tribeni Prasad Banerjee, Swagatam Das (2012). Multi-sensor data fusion using support vector machine for motor fault detection. Information Sciences 217 (2012) 96–107.

[12] Pedro Santos, Luisa F. Villa (2015). An SVM-Based Solution for Fault Detection in Wind Turbines. Sensors 2015, 15, 5627-5648; doi:10.3390/s150305627.

[13] Andrew Kusiak, and Anoop Verma (2011). A Data-Driven Approach for Monitoring Blade Pitch Faults in Wind Turbines. Ieee Transactions on Sustainable Energy, Vol. 2, No. 1, January 2011.

[14] Ranjan Ganguli and Inderjit Chopra. Helicopter Rotor System Fault Detection using Physics-Based Model and Neural Networks. AIAA Journal Vol. 36, No. 6, June 1998.

[15] Gino Iannace, Giuseppe Ciaburro and Amelia Trematerra (2019). Fault Diagnosis for UAV Blades Using Artificial Neural Network. Robotics 2019, 8, 59; doi:10.3390/robotics8030059.

[16] Juntong Qi, Zhe Jiang, Xingang Zhao (2007). Fault Detection Design for RUAV with an Adaptive Threshold Neural-Network Scheme. IEEE International Conference on Control and Automation WeD5-5 Guangzhou, CHINA.

[17] Adam Bondyra, Przemysław Gasior (2017). Fault Diagnosis and Condition Monitoring of UAV Rotor using Signal Processing. Signal Processing algorithms, architectures, arrangements, and applications SPA 2017.

[18] A. Freddi, S. Longhi, A. Monteri (2009). A Model-Based Fault Diagnosis System for Unmanned Aerial Vehicles. 7th Ifac Safeprocess (Safeprocess'09) Barcelona, Spain, June 30 - July 3, 2009.

[19] Wai Keng Ngui, Mohd Salman Leong (2017). Blade Fault Diagnosis using Artificial Neural Network. International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 4 (2017) pp. 519-526.

[20] Masayuki Hirayama (2016). Sensor selection method for IOT systems – focusing on embedded system requirements. MATEC Web of Conferences 59, 01002, ICFST 2016.

[21] Paul M. Jones, Quentin Lonne (2018). A Straightforward Route to Sensor Selection for IoT Systems, Research-Technology Management, 61:5, 41-50, DOI: 10.1080/08956308.2018.1495965.

[22] Chun-Cheng Lin, Der-Jiunn Deng (2017). Many-Objective Sensor Selection in IoT System. IEEE Wireless Communications, June 2017. DOI: 10.1109/MWC.2017.1600409.

[23] Shih-Yuan Yu, Chi-Sheng Shih (2014). QoS Oriented Sensor Selection in IoT System. 2014 IEEE International Conference on Internet of Things (iThings 2014), Green Computing and Communications (GreenCom 2014), and Cyber-Physical-Social Computing (CPSCom 2014). DOI 10.1109/iThings.2014.37.

1. [1][2][3] SVM and ANN have been widely used in fault detection in rotating elements such as bearings and gears.

2. [5] P.K. Kankar, Satish C. Sharma (2011). Fault diagnosis of ball bearings using machine learning methods, Expert Systems with Applications 1876–1886.

3. [3][4] Apart from these two algorithms, genetic algorithm and optimization technique such as ant colony optimization have also been used in the past.

4. [4] Xu Li a, A'nan Zheng b (2013). Rolling element bearing fault detection using support vector machine with improved ant colony optimization, Measurement 2726–2734.

5. [2] With the help of Big Data, Digital Manufacturing & Supply chain, IOT can improve productivity in the manufacturing industry.

6. [6] Olivier Janssen, Viktor Slavkovikj (2016). Convolutional Neural Network Based Fault Detection for Rotating Machinery. Journal of Sound and Vibration.

7. [7] Nassim Laouti, Nida Sheibat-Othman, Sami Othman (2011). Support Vector Machines for Fault Detection in Wind Turbines. The International Federation of Automatic Control Milano (Italy) August 28 - September 2, 2011

8. [8] W.Y. Liu, B.P. Tang (2014). The structure healthy condition monitoring and fault diagnosis methods in wind turbines: A review. Renewable and Sustainable Energy Reviews 44 (2015) 466–472.

9. [9] Prashant M. Pawar and Sung Nam Jung. (2008), Support Vector Machine based Online Composite Helicopter Rotor Blade Damage Detection System. Journal of Intelligent Material Systems and Structures, Vol. 19.

10. [12] Prashant M. Pawar, Ranjan Ganguli (2003). Genetic fuzzy system for damage detection in beams and helicopter rotor blades. Comput. Methods Appl. Mech. Engrg. 192 (2003) 2031–2057.

11. [13] Tribeni Prasad Banerjee, Swagatam Das (2012). Multi-sensor data fusion using support vector machine for motor fault detection. Information Sciences 217 (2012) 96–107.

12. [14] Pedro Santos, Luisa F. Villa (2015). An SVM-Based Solution for Fault Detection in Wind Turbines. Sensors 2015, 15, 5627-5648; doi:10.3390/s150305627.

13. [15] Andrew Kusiak, and Anoop Verma (2011). A Data-Driven Approach for Monitoring Blade Pitch Faults in Wind Turbines. Ieee Transactions on Sustainable Energy, Vol. 2, No. 1, January 2011.

14. [16] Ranjan Ganguli and Inderjit Chopra. Helicopter Rotor System Fault Detection using Physics-Based Model and Neural Networks. AIAA Journal Vol. 36, No. 6, June 1998.

15. [17] Gino Iannace, Giuseppe Ciaburro and Amelia Trematerra (2019). Fault Diagnosis for UAV Blades Using Artificial Neural Network. Robotics 2019, 8, 59; doi:10.3390/robotics8030059.

16. [18] Juntong Qi, Zhe Jiang, Xingang Zhao (2007). Fault Detection Design for RUAV with an Adaptive Threshold Neural-Network Scheme. IEEE International Conference on Control and Automation WeD5-5 Guangzhou, CHINA.

17. [19] Adam Bondyra, Przemysław Gasior (2017). Fault Diagnosis and Condition Monitoring of UAV Rotor using Signal Processing. Signal Processing algorithms, architectures, arrangements, and applications SPA 2017.

18. [20] A. Freddi, S. Longhi, A. Monteri (2009). A Model-Based Fault Diagnosis System for Unmanned Aerial Vehicles. 7th Ifac Safeprocess (Safeprocess'09) Barcelona, Spain, June 30 - July 3, 2009.

19. [21] Wai Keng Ngui, Mohd Salman Leong (2017). Blade Fault Diagnosis using Artificial Neural Network. International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 4 (2017) pp. 519-526.