Classification Studies on Vibrational Patterns of Distributed Fiber Sensors using Machine Learning

Yada Sai Pranay
Department of Electronics and Communication Engineering
Indian Institute of Information Technology, Design and Manufacturing, Kancheepuram
Chennai, India
edm19b037@iiitdm.ac.in

Jagadeeshwar Tabjula
Department of Petroleum Engineering
Louisiana State University
Batan Rouge, LA 70803, USA
t.l.jagadeeshwar@gmail.com

Srijith Kanakambaran
Department of Electronics and Communication Engineering
Indian Institute of Information Technology, Design and Manufacturing, Kancheepuram
Chennai, India
srijith@iiitdm.ac.in

***Abstract*— Distributed fiber optic sensors are smart replacements to point sensors in monitoring vibrations over long distances with excellent resolution. In this paper, we investigate the use of machine learning models to classify different vibrational events. Spectrograms of vibrational events available on a public database is used for training and testing the machine learning models like Support Vector Machine, Ensemble learning and K-Nearest Neighbour. The best accuracy of 86.1% is obtained for Support Vector classifier after hyperparameter tuning with 5-fold cross validation.**

Keywords— Distributed fiber sensing, Machine Learning, Vibration Pattern

I. INTRODUCTION

Distributed fiber optic sensors are gaining a lot of attention recently due to their unique ability of integrating multiple sensing points in an optical fiber along long distances [1]. Compared to point-sensing and quasi-point sensing approaches, such optical fiber-based sensors offer true distributed sensing of physical parameters like temperature, strain or vibration at any point along the sensing length. As such, these sensors are quite useful in applications like leak detection in oil and gas pipelines, border security, intrusion detection etc. [1, 2]. Such sensors typically deliver huge amount of data depending on the sensing range and resolution. To analyze and classify the large amount of data gathered from such sensors, it becomes imperative to use artificial intelligence algorithms and learning techniques including machine learning (ML) or deep learning [3]. Although deep learning technique is demonstrated for classification of events in [4], in this paper, we have explored the use of machine learning in order to reduce the complexity and computational requirement. In this work, the authors have explored the use of machine learning models like Support Vector Machines (SVM), Ensemble Learning, and K-Nearest Neighbor (KNN) to recognize and classify various events in a distributed fiber optic vibrational sensor.

II. METHODOLOGY

The data for analysis was obtained from a publicly available dataset available in [4]. The distributed vibration sensing system in [4] consists of a pulsed narrow linewidth laser, fiber amplifier, circulator sensing fiber and a photodetector. The backscattered Rayleigh component is detected at the photodetector followed by the data acquisition unit. Spectral subtraction is performed for the backscattered light from fiber for noise reduction and then short-time Fourier transform (STFT) was used to obtain the 200 × 128 sized spectrogram grayscale images reported in the public dataset [4]. The authors have investigated data corresponding to five different events namely dropping a weight, friction, knocking, Piezo Ceramic Sheet Vibration (PCSV) and noise, thereby giving a total dataset of 21000 images. Fig. 1 shows a few sample spectrograms of vibrational pattern of events obtained after spectral subtraction and STFT.

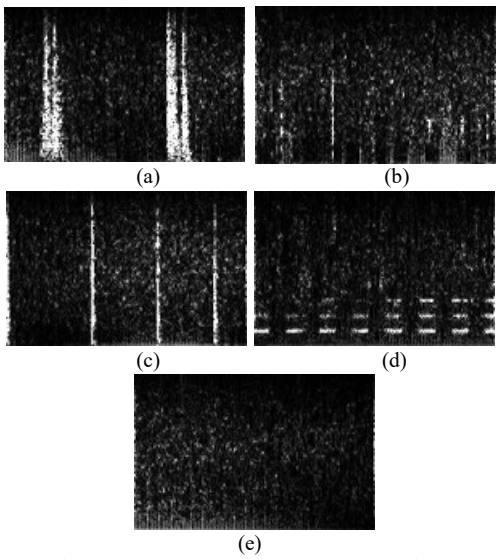

Fig. 1. Sample Images corresponding to events (a) dropping (b) friction (c) knocking (d) PCSV (e) noise

This dataset is used to train and test the ML models. We have investigated the suitability of three different techniques to classify the various events:

- Support Vector Machine (SVM): The goal of this algorithm is to find the right decision boundary (called a hyperplane) such that the

978-1-6654-9291-1/22/$31.00 ©2022 IEEE.

new data point is placed in the relevant category in the future. Support Vectors are the data points that are nearest to a hyperplane and have an effect on its position. Support Vector Classifier (SVC) technique using SVM is used in this model.

- Ensemble Learning: This is the process of strategically generating and combining multiple models, like classifiers, to address a specific machine learning problem. Ensemble learning is most commonly used to improve classification and prediction [6]. Random Forest Classifier technique in Ensemble learning is used in this model.
- K-Nearest Neighbour (KNN): This technique assumes the similarity among the new case as well as the available cases and places the new case within classification that is closest similar to available categories [7]. The KNN algorithm stores all available data and uses similarity to classify new data points. K-Neighbors Classifier (KNC) technique in KNN is used in this model.

The following parameters were extracted from the image dataset and were used as features for the ML models:

- *Mean*: The mean is the ratio of sum of samples in a dataset with total number of samples in the dataset. Here the data set consists of the pixel values of the image.

$$\mu_i = \frac{1}{N}\sum_{i=1}^{N} X(n)_i \quad (1)$$

μ_i – Mean,
$X(n)_i$ – Value of X(n) at i
N – Total number of samples

Here X(n) represents the one dimensional signal obtained from the image by taking average along the columns

- *Standard deviation:* The standard deviation quantifies how dispersed the data is in relation to the mean. A lower value indicates that data is centered around the mean, whereas a large value indicates that data is more distributed.

$$\sigma = \sum_{i=1}^{N} \sqrt[2]{\frac{(X(n)_i - \mu_x)^2}{N}} \quad (2)$$

- *Zero Crossing Rate:* The Zero Crossing Rate is the frequency with which the signal's sign changes during the frame. In other words, it says how many times the sign of the signal shifts from positive to negative and vice versa.
- *Event Strength:* It is the ratio of number of samples in an Event above the threshold frequency with total number of events. It gives the average number of samples in an event.

$$E = \frac{Number\ of\ Samples\ in\ an\ Event}{Total\ Number\ of\ Events} \quad (3)$$

- *Relative Spectral Energy:* The spectrogram was divided into 4 ranges 0-1 kHz, 1-2 kHz, 2-3 kHz, and 3-4 kHz. The relative share of spectral energy in each range is calculated as the ratio of energy in frequency range to the total energy.
- *Skewness*: Skewness is a measure of the asymmetry of a distribution. Skewness can be either positive or negative or zero.

$$\text{Skewness} = \sum_{i=1}^{N} \frac{(X(n)_i - \mu_x)^3}{(N-1)*\sigma^3} \quad (4)$$

- *Kurtosis*: Kurtosis is a statistical measure used to describe the degree to which scores cluster in the tails or the peak of a frequency distribution.

$$\text{Kurtosis} = \sum_{i=1}^{N} \frac{(X(n)_i - \mu_x)^4}{(N-1)*\sigma^4} \quad (5)$$

- *Correlation coefficient*: The correlation between two variables indicates that changes in one variable are related to changes in the other variable. Here the two variables considered are the original image and median filtered image.

$$C = \frac{\sum_{i=1}^{N}(X_i - \mu_x)(Y_i - \mu_y)}{\sqrt{\sum_{i=1}^{N}(X_i - \mu_x)^2 \sum_{i=1}^{N}(Y_i - \mu_y)^2}} \quad (6)$$

C – Correlation Coefficient
X – Original Image
Y – Median Filtered Image

- *Edge Detection*: It indicates the changes in pixel values both horizontal (X-axis) and vertical (Y-axis) direction. Wherever there is a sudden change in pixel values, an edge is detected.

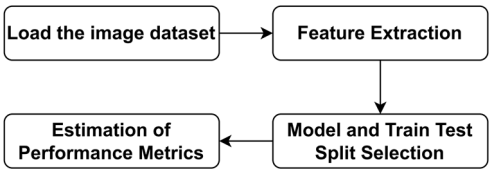

Fig. 2. Flowchart of Methodology

The overall flowchart describing the methodology is shown in Fig. 2. From the images in the dataset, the features for the model have to be extracted. Features used in this work are: Mean, Std, Correlation, Relative Spectral Energy, Zero Crossing Rate, Skewness, Kurtosis, Event Strength, and Edge Detection which have been discussed previously. Further, the appropriate model and train-test split has to be chosen. Then, the relevant code has to be compiled and executed. Finally, the metrics that estimate performance of the model like precision, accuracy, recall and f1 score can be extracted.

III. RESULTS

The features to be used in the classification models were described in the previous section. Box plots were used to analyze the suitability of a feature for the classification model. From the several box plots,

9 features were found to be suitable. Among these, the box plots for Zero Crossing Rate, Standard Deviation, Correlation and Event Strength are shown in Fig. 2.

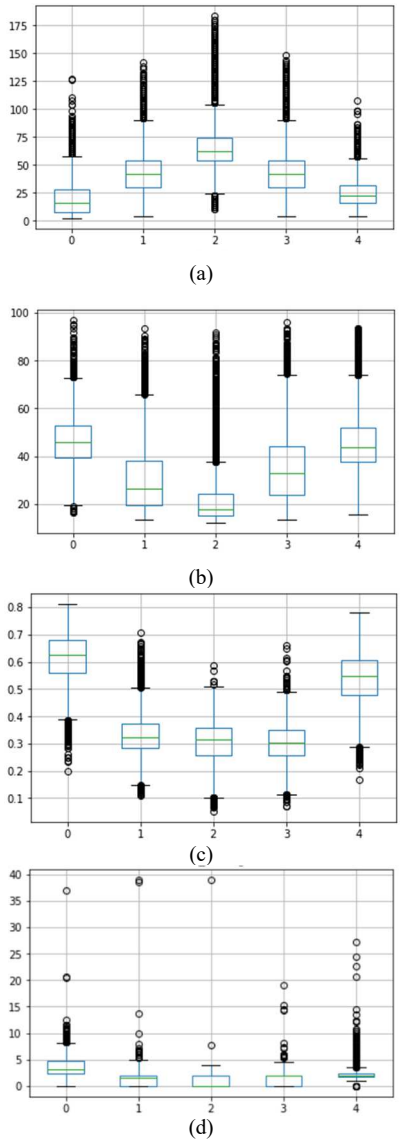

Fig. 3. Box Plots for (a) Zero Crossing Rate (b) Standard Deviation (c) Correlation (d) Event Strength

The above features were applied as inputs to the machine learning models. Three ML models – Support Vector Classifier (SVC), Random Forest Classifier (RFC) and K-Nearest Neighbour Classifier (KNC) were considered. The performance of various train-test splits such as 70-30, 80-20 and 90-10 was investigated for each of the models. The performance parameters compared were Precision, Recall, F1 score, and Accuracy as defined below:

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (7)$$

$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (8)$$

$$\text{F1 Score} = \frac{2 * Precision * Recall}{Precision + Recall} \quad (9)$$

$$\text{Accuracy} = \frac{True\ Positive + True\ Negative}{True\ Positive + False\ Positive + True\ Negative + False\ Negative} \quad (10)$$

The results obtained for single trial for all the machine learning models are summarized in Table 1. It may be noted that the best performance for SVC and RFC is obtained for a train-test split of 90-10. However, in the case of KNC, the best performance is obtained for a train-test split of 70-30. The maximum accuracy of 86.67% was obtained for SVC with a train-test split of 90-10. The confusion matrix for this case is illustrated in Fig. 4.

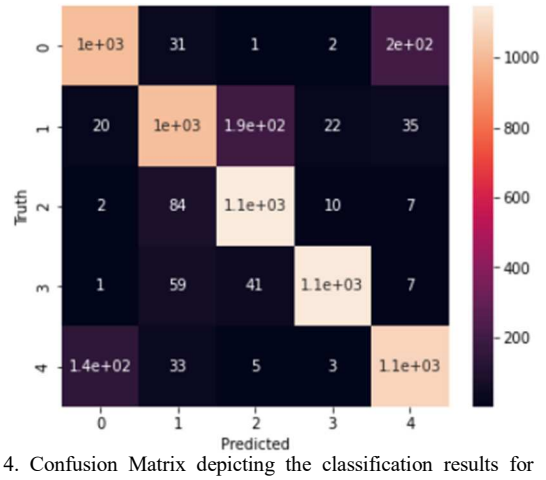

Fig. 4. Confusion Matrix depicting the classification results for various events

Further, 5-fold cross-validation and hyper parameter tuning[8] was attempted to improve the model performance. The hyperparameters considered for SVC, RFC and KNC were the number of hyperplanes, estimators and neighbours respectively.

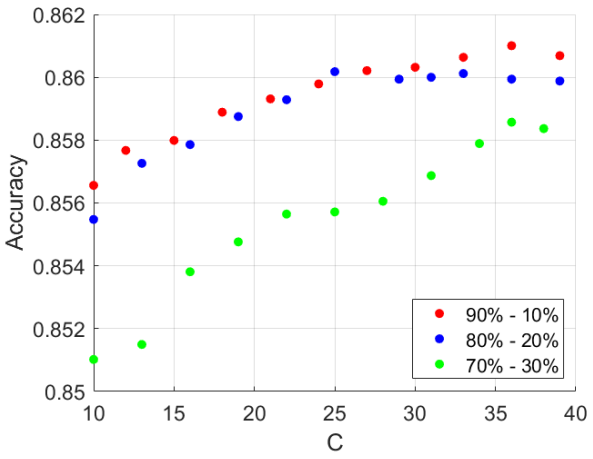

Fig. 5. Accuracy vs number of hyperplanes (C) plot for different test train splits (a) 70 % - 30% (b) 80 % - 20% (c) 90 % - 10% in SVC

Table 1: Evaluation Metrics of machine learning models

Model	SVC				RFC				KNC			
Split	Pr	Re	F1	Ac	Pr	Re	F1	Ac	Pr	Re	F1	Ac
70-30	86	85.8	85.8	85.75	82.2	82	82	83.03	80.9	80	80.2	80.3
80-20	86.5	86.3	86.3	86.33	82.1	81.9	82	82.88	80.6	79.7	79.9	80
90-10	86.7	86.5	86.6	86.67	81.8	81.6	81.7	83.14	80.2	79.1	79.3	79.09

Pr – Precision, Re – Recall, F1 – F1 Score, Ac – Accuracy

Fig. 5 represents the plots for accuracy for different values of number of hyperplanes (C) and different train-test splits. For each train-test split, the maximum accuracy is noted, and the corresponding C value is recorded in Table 2. It has been observed that the maximum accuracy of 86.1% has been obtained for 90%-10% train-test split in SVC.

Table 2: Accuracy of SVC

Split	*C*	*Accuracy(%)*
70%-30%	36	85.86
80%-20%	25	86.02
90%-10%	36	86.10

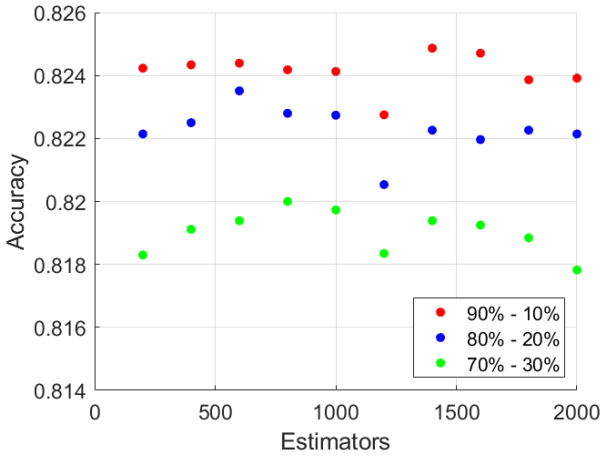

Fig. 6. Accuracy vs estimators plot for different test train splits (a) 70% - 30% (b) 80 % - 20% (c) 90 % - 20% in Random Forest

Fig. 6 represents the plots for accuracy for different values of estimators and different train-test splits in RFC. Similar to the previous case, the highest value of accuracy corresponding to each split and accuracy at that estimator value is noted down. Table 3 shows that a maximum accuracy of 82.49% is obtained in 90%-10% split in RFC.

Table 3: Accuracy of RFC

Split	*Estimators*	*Accuracy(%)*
70%-30%	800	82.00
80%-20%	600	82.35
90%-10%	1400	82.49

Fig. 7 represents the plots of accuracy for different values of neighbours for different train test splits in KNC. Table 4 summarizes the accuracy for each case and the corresponding number of neighbours. It has been again noticed that the best accuracy of 80.51% has been obtained for 14 neighbours in the train-test split of 90%-10% in KNC.

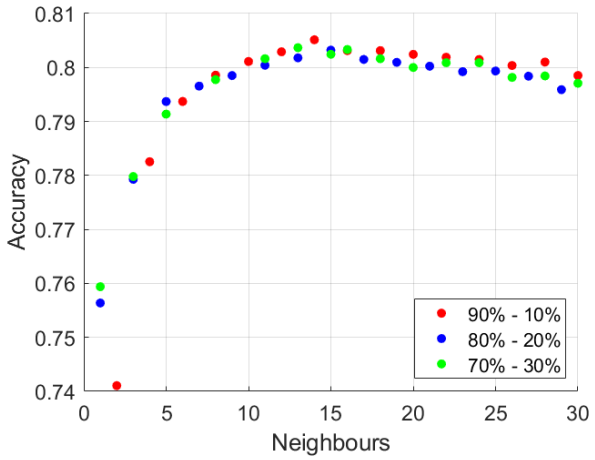

Fig. 7: Accuracy vs neighbours plot for different test train splits (a) 70% - 30% (b) 80 % - 20% (c) 90 % - 10% in KNC

Table 4: Accuracy of KNC

Split	*Neighbors*	*Accuracy(%)*
70%-30%	13	80.37
80%-20%	15	80.32
90%-10%	14	80.51

IV. CONCLUSION

In this paper, the use of machine learning models were investigated for classifying five vibrations events in a distributed fiber optic vibrational sensor. Different features were extracted from vibrational pattern dataset consisting of 21000 spectograms. The best accuracy of 86.1% has been obtained using 5-fold cross validation with hyperparameter value of 36 using support vector classifier in support vector machine model. Further extensions to this work can be attempted primarily in terms of improving the performance of the machine learning models. Multiple new features could be potentially included to improve the accuracy. Another

possible extension to this work is to investigate the suitability of deep learning models for classification.

REFERENCES

[1] X. Liu, B. Jin, Q. Bai, Y. Wang, D. Wang, and Y. Wang, "Distributed Fiber-Optic Sensors for Vibration Detection", *Sensors*, vol. 16, no. 8, p. 1164, Jul. 2016, doi: 10.3390/s16081164.

[2] K. Hicke, M. -T. Hussels, R. Eisermann, S. Chruscicki and K. Krebber, "Condition monitoring of industrial infrastructures using distributed fibre optic acoustic sensors," 2017 25th Optical Fiber Sensors Conference (OFS), 2017, pp. 1-4, doi: 10.1117/12.2272463.

[3] Z. -Y. Dai, Y. Liu, L. -X. Zhang, Z. -H. Ou, C. Zhou and Y. -Z. Liu, "Landslide monitoring based on high-resolution distributed fiber optic stress sensor," 2008 1st Asia-Pacific Optical Fiber Sensors Conference, 2008, pp. 1-4, doi: 10.1109/APOS.2008.5226289.

[4] Yining Pan, Tingkun Wen, Wei Ye - "Time attention analysis method for vibration pattern recognition of distributed optic fiber sensor" – Optik - International Journal for Light and Electron Optics 2022

[5] J. Bell, Machine Learning: Hands-On for Developers and Technical Professionals. Wiley, 2020.

[6] A. Cutler, D. R. Cutler, and J. R. Stevens, Random Forests. Boston, MA: Springer US, 2012, pp. 157–175

[7] G. Guo, H H. Wang, D. Bell, Y. Bi, and K. Greer, "Knn model-based approach in classification," in On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE, R. Meersman, Z. Tari, and D. C. Schmidt, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 986–996

[8] R. G. Mantovani, A. L. D. Rossi, J. Vanschoren, B. Bischl and A. C. P. L. F. de Carvalho, "Effectiveness of Random Search in SVM hyper-parameter tuning," *2015 International Joint Conference on Neural Networks (IJCNN)*, 2015, pp. 1-8, doi: 10.1109/IJCNN.2015.7280664.