Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control
Journal homepage: http://kinetik.umm.ac.id
ISSN: 2503-2267
Vol. 9, No. 1, February 2024, Pp. 21-28

Entropy-based feature extraction and k-nearest neighbors for bearing fault detection

Sinta Uri El Hakim*[1], Irfan Bahiuddin[1], Rokhmat Arifianto[2], Syahirul Alim Ritonga[3]
Department of Mechanical Engineering, Vocational College, Universitas Gadjah Mada, Yogyakarta, Indonesia[1]
National Research and Innovation Agency, Indonesia[2]
Department of Mechanical and Industrial Engineering, Faculty of Engineering, Universitas Gadjah Mada, Yogyakarta, Indonesia[3]

Article Info
Keywords:
K-Nearest Neighbors, Entropy, Feature Extraction, Bearing, Support Vector Machine

Article history:
Received: August 05, 2023
Accepted: November 27, 2023
Published: February 28, 2024

Cite:
S. U. E. Hakim, I. Bahiuddin, R. Arifianto, and S. A. Ritonga, “Entropy and K-Nearest Neighbors-Based Feature Extraction for Bearing Fault Detection”, KINETIK, vol. 9, no. 1, Feb. 2024. Retrieved from https://kinetik.umm.ac.id/index.php/kinetik/article/view/1814

*Corresponding author.
Sinta Uri El Hakim
E-mail address:
uriel.hakim@ugm.ac.id

Abstract
Bearing failures in rotating machines can lead to significant operational challenges, causing up to 45-55% of engine failures and severely impacting performance and productivity. Timely detection of bearing anomalies is crucial to prevent machine failures and associated downtime. Therefore, an approach for early bearing failure detection using entropy-based machine learning is proposed and evaluated while combined with a classifier based on K-Nearest Neighbors (KNN) and Support Vector Machine (SVM). Entropy-based feature extraction should be able to effectively capture the intricate patterns and variations present in the vibration signals, providing a comprehensive representation of the underlying dynamics. The results of the classification carried out by KNN-Entropy have an accuracy value of 98%, while the SVM-Entropy model has an accuracy of 96%. Hence, the Entropy-based feature extraction giving the best accuracy when it is coupled with KNN.

1. Introduction

Bearing is an important and vulnerable component in a rotating machine [1], [2]. Based on a reliability survey on rotating machines, 45~55% of engine failures are caused by bearing failure [3], [4]. The statistic is in line with a research conducted by the Electric Power Research Institute EPRI, which states that 41~42% of induction motor failures are caused by damage to the bearings, and the other 36% are caused by defects in the stator [5], [6], [7]. Damage or defects in bearings, such as wear, cracks, scratches, and deformation, can cause vibration and noise in the engine, which results in engine failure when operating [8]. To avoid failure of the machine, it is necessary to carry out preventive maintenance.

Bearings consist of several parts, including inner rings, outer rings, rollers, and several other small parts. One of the bearing applications is the utilization of train propulsion. Hence, it is important to have well-planned bearing design and maintenance [5]. Performing preventive maintenance on rotating machines can help prevent severe damage to the bearings [9]. Several researchers have developed vibration analysis based on Machine Learning (ML) and Deep Learning (DL). Vibration, in general, can also be used as an early warning system on rotating machines or other machines with the potential for vibration.

In 2019, Pandarakone classified bearing damage using the SVM method with an accuracy rate of 95% [10]. Pandarakone uses RBF-type kernels with cost parameter and gamma parameter values of 1 and 0.33, respectively. Using vibration data, Liu et al. [11] 2021 conducted a study to detect early bearing damages. Liu compared the SVM and KNN classifiers with the MSAF-20-MAX feature extraction. This study shows that the classification results using KNN-MSAF-20-MAX can produce higher accuracy than SVM. Chérrez et al. [7] researched bearing fault detection using the combination of vibration and acoustic data based on supervised ML. Chérrez compared three different ML methods to detect bearing damage: MFE-SVM, RWE, and TSFDR-LDA. In this study, the TSFDR-LDA showed the best result, with an accuracy of 98,28%. In 2023, Song et al. [12] studied bearing surface damage using the YOLOv4 object detection algorithm. Song used the bounding box method to determine the defects on a photo of the bearing surface. The accuracy value of this study reached a value of 97.96%.

Lee et al. [13] conducted a study by dividing the method into three outlines. First, Lee converts the vibration data into two-dimensional data (image) using the Short-Time Fourier Transform (SFTF). Second, the data is preprocessed using Mel Frequency Cepstral Coefficients (MFCCs). Third, the data is divided into certain classes using Class

Activation Map (CAM). Lu et al. [8] carried out machine vision-based defect classification using Lc-MNN to perform image segmentation. This study resulted in an accuracy of up to 99.5%, but the drawbacks of this method are that it requires much computation and takes a long time. Tastimur et al. [14] studied bearing damage detection by applying CNN as an image classification. Tastimur acquires vibration data which is then converted into an image vibration signal. The processed 2D signal image is then segmented and trained using CNN.

In the previous studies, various techniques and methods have been carried out to develop bearing fault. They have explored various techniques and methods for developing bearing fault detection, emphasizing the need for quick and lightweight algorithms for real-time applications. Such algorithms should be suitable for implementation in microcontrollers. This research, however, primarily concentrates on the classification of damage in bearing components. To this end, it introduces a feature extraction approach based on Entropy. Entropy is chosen for its ability to enhance data differentiation, as it is particularly effective at pinpointing and highlighting the most critical features in a dataset. It assesses the data's degree of disarray or unpredictability, thereby facilitating a clearer separation between different classes or categories. This feature extraction method combines widely used classifiers, specifically KNN and SVM.

The paper is divided into several parts, namely sections 2, 3, and 4. In section 2, we will discuss the methods and data sets used. This section will also explain the data preprocessing, processing, and evaluation procedures. In section 3, we will discuss more about the results of training and testing the model using the classifier that has been selected. In section 4, we will explain the conclusions of the research.

2. Research Method

The general architecture of the proposed method is shown in Figure 1. Firstly, the data was obtained from the accelerometer sensor or other sources. After receiving the data, the next step was preprocessing the data by using the signal segmentation method. Afterwards, feature extraction was performed on the data using combination of four Entropy, namely: Approximate Entropy, Sample Entropy, Slope Entropy, and Dispersion Entropy. This data were split five times using Random Split Data, and each split data were trained by 30 iterations. The features were inputted to the training process for building a classified method.

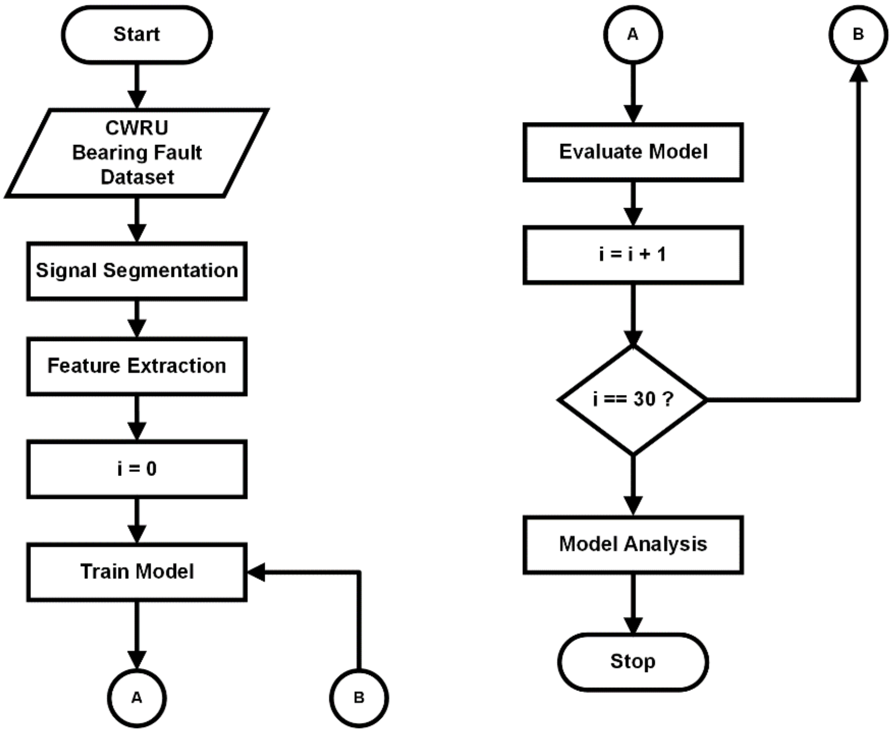


Figure 1. Flowchart of the Training and Testing Dataset

2.1 The Data Set

CRWU (Case Western Reserve University) provides easily accessible open-source data for research purposes. One of the data supplied by CRWU is the bearing dataset which conducted the experiments using a 2 HP electric motor. Data were taken using an accelerometer sensor which was placed in a 12 o'clock position at both the Drive-End and Fan-End motor housing [15] [16]. Digital data collected was taken at 12,000 samples per second and 48,000 samples per second for each position. CRWU performs vibration measurements on bearings with three defect locations categories: Inner ring, Outer ring, and Ball. 2.

Defects in bearings were prepared by using Electro-Discharge Machining (EDS) with different diameters, namely 0.007, 0.014, 0.021, and 0.028 inches. This study used data on the Drive End (DE) and Fan End (FE) with detailed specifications, as shown in Table 1. This data were divided into eight bearing defect locations plus one normal bearing condition. Hence, the total number of classes used in the study was nine for all.

Table 1. Fault Location and Bearing Label Class

Class	Fault Location	Diamater	Depth	Data Sample	Bearing Manufacture
0	Ball defect	0.021	0.011	201	SKF
1	Ball defect	0.007	0.011	201	SKF
2	Ball defect	0.014	0.011	158	SKF
3	Inner ring	0.007	0.011	202	SKF
4	Outer ring	0.007	0.011	201	SKF
5	Inner ring	0.021	0.011	201	SKF
6	Outer ring	0.014	0.011	202	SKF
7	Normal	-	-	200	SKF
8	Outer ring	0.021	0.011	200	SKF
9	Inner ring	0.014	0.011	201	SKF

The dataset that has been obtained were preprocessed by performing signal segmentation from 48,000 signals per second to 2,400 signals per segment. Hence, the total segmented signals were 1966 DE signals and 1966 FE signals. DE and FE, data segmentation results were then grouped into classes by dividing the number of each class, as shown in Table 2. After that, this signal was extracted using Feature Extraction. The signal extraction results were divided into two parts: Data Training and Data Testing, with a ratio of 75% and 25%, respectively. In this study, the split data was carried out five times, and each data split was trained in 30 training iterations. It was done to see whether the model has a good level of accuracy despite the changes to the arrangement of the training data and testing data.

2.2 Simulation Set-Up

The study utilized Google Colab, which offers a setup-free platform with complimentary access to computational resources. Google Colab facilitates a virtual environment, effectively executing all processes in the cloud. Data must be uploaded to Google Drive for accessibility in Google Colab. In this research, all training and testing data operations were conducted in the cloud environment.

This investigation compared two classifier types, KNN and SVM, both employing the same feature extraction methods. For KNN in this research, hyperparameters were configured with a n_neighbors value of 5 and Euclidean distance as the metric. For SVM, the selected hyperparameter was C=1.0, using an RBF kernel, with gamma set to the scale (1 / (number of features * variance of all features)).

This research aims to assess whether feature extraction through Entropy yields better results compared to Descriptive Statistics (DS). Consequently, this study performed training and testing using analogous hyperparameters for both KNN and SVM with DS feature extraction. The DS parameters included were mean, median, kurtosis, and skewness.

2.3 Entropy for Feature Extraction

Entropy is widely used in research to measure the disorder or uncertainty in a system [17]. If the uncertainty of the system tends to be small, then the system can move towards order. The outstanding value of a system is obtained when the entropy value is zero. Meanwhile, if the system uncertainty tends to be high, then the system is considered unstable. This study used Entropy to find and determine hidden features in each signal. Then, the feature were generated with a unique value according to the nature of the Entropy, which was random. Four types of Entropy were used, including Approximate Entropy, Sample Entropy, Slope Entropy, and Dispersion Entropy.

Approximate Entropy (ApEn) [18] is an algorithm used to measure the regularity of a system without knowing information from the system. ApEn is used to help classify a reasonably complex system. The ApEn value is determined in the following stages:

1) Let the data sequence consists of N data, the value of $X = [x(1), x(2), x(3), \dots, x(N)]$
2) Let $x(i)$ is a sub-sequence value of X, it can be written as $x(i) = [x(i), x(i+1), x(i+2), \dots, x(i+m-1)]$ for $1 \leq i \leq N - m,$ where m represent the sample number.
3) If r is level filter noise, then it can be written using Equation 1:

$$r = k \times SD \text{ for } k = 0, 0.1, 0.2, 0.3, \dots, 0.9 \tag{1}$$

where SD is the standard deviation of X sequence data.

4) If $\{x(j)\}$ is a sequence set obtained from $x(j)$ by varying the value of j from 1 to N. Then each order of $x(j)$ in the set $\{x(j)\}$ compared to $x(i)$. In this process, there are two parameters represented as $C_i^m(r)$ and $C_i^{m+1}(r)$. Hence, the Equation 2 and Equation 3 can be defined as follow:

$$C_i^m(r) = \frac{\sum_{j=1}^{N-m} k_j}{N-m} \tag{2}$$

where,

$$k = \begin{cases} 1, & if|x(i) - x(j)|\, for\ 1 \leq j \leq N-m \\ 0, & otherwise \end{cases}$$

and

$$C_i^{m+1}(r) = \frac{\sum_{j=1}^{N-m} k_j}{N-m} \tag{3}$$

where k value is determined by:

$$k = \begin{cases} 1, & if\ |x(i) - x(j)| \leq r\ and\ |x(i+1) - x(j+1)| \leq r\ for\ 1 \leq j \leq N-m \\ 0, & otherwise \end{cases}$$

5) Hence, the equation $\Phi^m(r)$ dan $\Phi^{m+1}(r)$ are obtained using Equation 4 and Equation 5:

$$\Phi^m(r) = \frac{\sum_{i=1}^{N-m} \ln(C_i^m(r))}{N-m} \tag{4}$$

$$\Phi^{m+1}(r) = \frac{\sum_{i=1}^{N-m} \ln(C_i^{m+1}(r))}{N-m} \tag{5}$$

6) $\mathrm{ApEn}\ (\mathrm{m, r, N})$ Equation formulated using $\Phi^m(r)$ dan $\Phi^{m+1}(r)$, then the formula can be obtained using Equation 6:

$$ApEn(m, r, N) = \Phi^m(r) \text{ -}\Phi^{m+1}(r) = \frac{1}{N-m}\left[\sum_{i=1}^{N-m} \ln\left(\frac{C_i^m(r)}{C_i^{m+1}(r)}\right)\right] \tag{6}$$

Richman [19] introduced a new calculation algorithm called Sample Entropy (SampEn). It tends to be lighter in computing and can reduce half of the computations performed by ApEn. The differences between SampEn and ApEn can be seen in the following stages:

1) The denominator in SampEn only consists of $SampEn\ (N, m)$, which does not involve itself in the calculations.
2) ApEn calculates the logarithm first and then multiplies it up. SampEn does the opposite.
3) The comparison value of $d[x(i), x(j)]$ less than r compared to the total value of the distance $\mathrm{N - m}$ which is represented as $B_i^m(r)$ and it can be formulated as shown in Equation 7:

$$B^m(r) = \frac{1}{N-m+1} \sum_{i=1}^{N-m+1} B_i^m(r) \tag{7}$$

4) Repeat the steps above to get the value of $B_i^{m+1}(r)$
5) Hence SampEn can be formulated as shown in Equation 8:

$$Samp_{En}(m, r) = \lim_{N\to\infty} [-\ln \frac{B_i^{m+1}(r)}{B^m(r)} \tag{8}$$

6) Where N is a finite number, then SampEn Equation 9 became:

$$Samp_{En}(m, r, N) = \ln\left(\frac{B_i^{m+1}(r)}{B^m(r)}\right) \tag{9}$$

Rostaghi and Azami [20] introduced Dispersion Entropy (DispEn) in 2016. It was developed to overcome the shortcomings of the previous Entropy method, which still needed to be faster to compute data. In addition, DispEn is also used to multiply the amplitude of various time series signals. Like other Entropy properties, DispEn is also very sensitive to signal amplitude and frequency changes. For univariate signal data $N: x = x_1, x_2, \ldots, x_N$, DispEn has 4 stages:

1) Mapped the value of $x_j (= 1,2,\ldots,N)$ into c class by the range 1 to c. The normal cumulative distribution function (NCDF) is used to map the x value to $y = y_1, y_2, \ldots, y_N$. Then y_j was assigned to integers 1 to c using linear algorithm. For each data, it was represented by $z_j^c = round(cy_j + 0.5)$, where z_j^c denotes the j-th number.
2) Each $z_i^{m,c}$ vector with embedding dimension m and time delay d was formulated based on $z_i^{m,c} = \{z_i^c, z_{i+d}^c, z_{i+(m-1)+d}^c\}$, where $i = 1,2,\ldots,N-(m-1)d$. Each $z_i^{m,c}$ time series was mapped into dispersion pattern $\pi_{v_0 v_1,\ldots,v_{m-1}}$ where $z_i^c = v_0, z_{i+d}^c = v_1, \ldots, z_{i+(m-1)d}^c = v_{m-1}$. The number of possible dispersion pattern to be assigned to the time series $z_i^{m,c}$ equal with c^m.
3) For every dispersion pattern c^m, the relative frequency can be obtained as shown in Equation 10:

$$p(\pi_{v_0 v_1 \ldots v_{m-1}}) = \frac{Number\ \{i \mid i \le N-(m-1)\,d,\ z_i^{m,c}\ has\ type\ \pi_{v_0 v_1 \ldots v_{m-1}}\}}{N-(m-1)d} \tag{10}$$

where $p(\pi_{v_0 v_1 \ldots v_{m-1}})$ is the number of dispersion pattern $\pi_{v_0 v_1 \ldots v_{m-1}}$ which can be assigned to $z_i^{m,c}$.
4) Then DispEn with embedding dimension m, time delay d, and number of classes c can be shown as written in Equation 11:

$$Disp_{En}(x,m,c,d) = \sum_{\pi=1}^{c^m} p(\pi_{v_0 v_1 \ldots v_{m-1}}) \cdot \ln(p(\pi_{v_0 v_1 \ldots v_{m-1}})) \tag{11}$$

Slope Entropy (SlopEn) was introduced in 2019 by Cuesta-Frau [21]. Slope Entropy is a development of the previous type of Entropy. The 2 new parameters were added in SlopEn, they were γ and δ, where each parameter represent increment horizontal dan increment vertical respectively. Hence, the stages of SlopEn can be described as follows:

1) Given the time series data of $Z = \{z_i, i = 1,2,\ldots,n$, hence the value of $Z_1 = \{z_1, z_2, \ldots, z_m\}$, $Z_2 = \{z_2, z_3, \ldots, z_{m+1}\}$, $Z_k = \{z_k, z_{k+1}, \ldots, z_n\}$, where m is embedding dimension and $k = n - m + 1$.
2) Decided threshold symbol parameter γ and δ. If $z_i - z_i > \delta$ then the symbol describe as +2, If $\gamma < z_{i+1} - z_i < \delta$ the symbol describe as +1, If $|z_i - z_i| \le \gamma$ the symbol describe as 0, Jika $-\delta < z_{i+1} - z_i < -\gamma$ the symbol describe as -1, and if $z_{i+1} - z_i < -\delta$ the symbol describe as -2, where $\delta > \gamma > 1$.
3) The sequence of symbol patterns obtained from the previous stage is $Y_1 = \{y_1, y_2, \ldots, y_{m-1}\}$, $Y_2 = \{y_2, y_3, \ldots, y_m\}$, and $Y_k = \{y_k, y_{k+1}, \ldots, y_{n-1}\}$, where y_j is the symbol associated with $z_{j+1} - z_j$ and $k = n - m + 1$.
4) Given the total recorded number of symbols is t_i, $i = 1, 2, \ldots, S$, and the recorded relative frequencies is p_i, hence it can be formulated as shown in Equation 12:

$$p_i = \frac{t_i}{J}, i = 1,2, \ldots ., S \tag{12}$$

5) Hence, the SlopEn Equation can be written using Equation 13 as follows:

$$Slop_{En}(x,m,N,\gamma,\delta) = -\sum_{i=1}^{S} p_i \ln p_i \tag{13}$$

2.4 K-Nearest Neighbors (KNN) for Classifier

K-nearest neighbors (KNN) is a classifier widely used in several implementations because of its ease of use. KNN uses a non-parametric classifier to classify new, unknown features, where the computation is done online [22]. For its use, KNN utilizes several techniques, including Euclidean distance, City Block, and the Cosine formula. However, KNN makes more use of the Euclidean distance to identify the nearest neighbor and calculate the class output value [23]. You can use the following Euclidean Equation, presented in Equation 14, to calculate the distance between two points between A and B in feature space:

$$Euclidean(A, B) = \sqrt{\frac{\sum_{i=1}^{m}(x_i - y_i)^2}{m}} \tag{14}$$

For data points that do not have a uniform standard, it is necessary to normalize the data with the equation as shown in Equation 15:

$$di = \sqrt{(a_x - a_r)^2 + (b_x - b_r)^2 + \left(c_x - c_y\right)^2 + \cdots} \tag{15}$$

Where $a, b,$ and c are testing data, x is the data training point and r is the data testing point.

2.5 Support Vector Machine (SVM) for Classifier

Vapnik first introduced the Support Vector Machine (SVM) in the 1960s. SVM has been widely used in pattern recognition for decades and performs excellently [24]. SVM works to find the value of margin optimization between two classes. However, for its development, SVM is widely used in multi-class [25].

In a data set with training samples, a number N is represented by $\{(x_i, y_i), i = 1, 2, \dots, N\}$ where $x_i \in R^n$ is feature vector and $y_i \in \{-1, 1\}$ is the label. Then, the linear classifier equation is shown in Equation 16.

$$f(x) = w^T x + b \tag{16}$$

where w and b classified as unknown parameters, it is necessary to determine the decision function, which is formulated by using Equation 17:

$$y_i f(x) \geq 1, \qquad i = \{1,2, \dots, N\} \tag{17}$$

In some cases, SVM does not include a separating hyper lane, so it is necessary to add a slack variable, as shown by the Equation 18.

$$y_i f(x) \geq 1 - \xi_i \tag{18}$$

where $\xi_i \geq 0$ and $i = \{1,2, \dots, N\}$. Hence the SVM Equation can be shown using Equation 19 as follows:

$$\begin{aligned} &\min(w, b, \xi) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\xi_i \\ \text{s.t.} \quad &y_i f(x) \geq 1 - \xi_i,\ \xi_i \geq 0, i = 1,2, \dots, N \end{aligned} \tag{19}$$

where C is a constant corresponding to the value of $\|w\|^2$. To perform linear separability, the original feature is placed in a higher dimensional space with a function $\Phi(x)$ called the kernel trick. Hence, the corresponding kernel can be shown in Equation 20.

$$K\left(x_i, x_j\right) = \Phi^T(x_i)\Phi(x_j) \tag{20}$$

Based on the corresponding kernel equation, the decision function can be formulated using Equation 21:

$$f(x_i) = \sum_{k=1}^{N_s} y_k \alpha_k K(x_i, s_k) + b \tag{21}$$

where α_k Lagrange multiplication and S_k is called a support vector where the Lagrange product is nonzero. Using the kernel method, SVM can overcome problems with linear separable hyperplanes. SVM can use several kernels, including linear, polynomial, sigmoid, and RBF.

3. Results and Discussion

After adding feature extraction training using KNN and SVM as the classifier, a comparison of Accuracy, Precision, Recall, and F-1 were achieved. Based on the data in Table 3, KNN-Entropy and SVM-Entropy have an average Accuracy, Precision, Recall, and F-1 of 98% and 96%, respectively. The accuracy figure was juxtaposed with that derived from integrating KNN-DS (K-Nearest Neighbors Descriptive Statistics) and SVM-DS (Support Vector Machine-Descriptive Statistics), in order to ascertain if the feature extraction performed by Entropy yielded the highest accuracy.

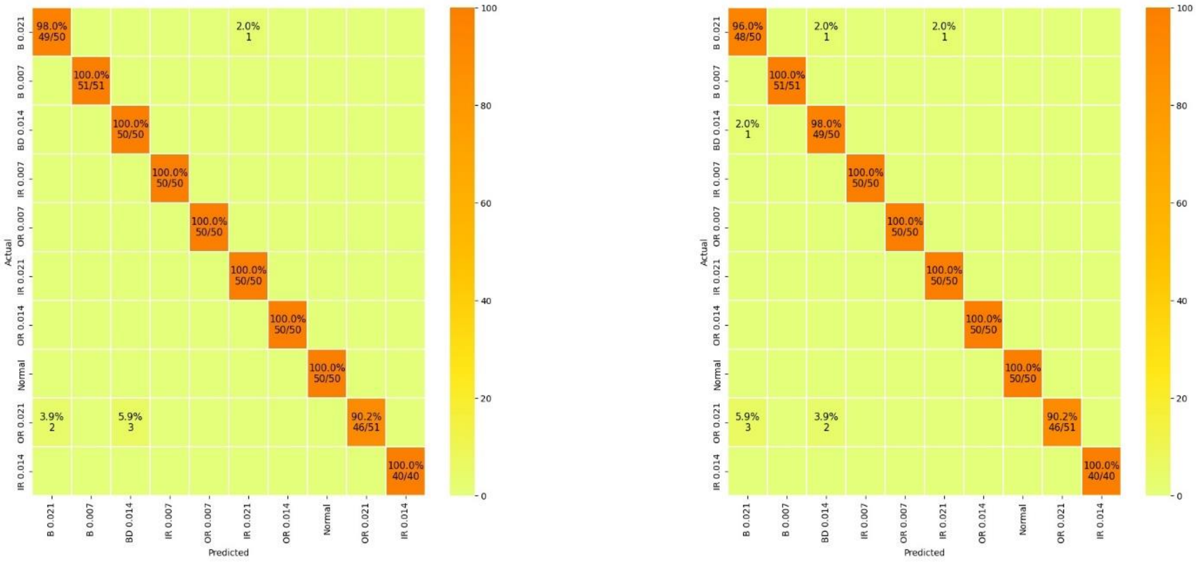
The result from KNN-DS and SVM-DS have lower accuracy: 79% and 71%, respectively. It is shown that the Entropy model is better for classifying data than the DS model. Entropy can generate features uniquely based on its algorithm steps. The Entropy algorithm allows feature extraction to be carried out based on the time domain and frequency domain. In DS, it can only be done based on time series calculations (mean, median, kurtosis, and skewness).

When choosing the classifiers, KNN outperformed SVM. This superiority of KNN arises from its methodology of classifying by measuring the query's distance to all data points, sorting them from nearest to farthest. The classification of a query depends on the distance to its nearest neighbors. This study obtained the best K value when K = 5. On the other hand, SVM primarily focuses on separating categories using linear boundaries, which has limitations. This approach increases the likelihood of misclassifying data that does not conform to specific classes, leading to a higher potential for classification errors. Figure 3 illustrates the comparative results of the KNN-Entropy and SVM-Entropy model classifications.

Table 2. The average result of the evaluation metric for each method

Model	Accuracy	Precision	Recall	F1 Score
KNN-DS	0,7919 ± 0,0034	0,7904 ± 0,0037	0,7927 ± 0,0036	0,79 ± 0,0032
KNN-Entropy	**0,9828 ± 0,0031**	**0.9833 ± 0.0032**	**0,9833 ± 0,003**	**0,9827 ± 0,0031**
SVM-DS	0,7194 ± 0,0725	0,7039 ± 0,093	0,719 ± 0,0723	0,6922 ± 0,0913
SVM-Entropy	0,964 ± 0,0373	0,9632 ± 0,0477	0,9642 ± 0,0402	0,9622 ± 0,0461

However, in order to be able to perform calculations using a microcontroller, a light and fast computation is required. This study did not only compared the value of the evaluation metric, but also the computation time performed. The KNN-Entropy combination model has a classification computing time of 0.0304 ± 0.0055 s. Meanwhile, SVM-Entropy has a computing time of 0.0075 ± 0.0012 s. The computation time required by KNN-Entropy is five times longer than SVM-Entropy. However, a more in-depth usability justification must be applied for further application to bearing fault detection. This justification is used to determine the use of the model on the microcontroller. To apply the KNN-Entropy model, we need to ensure that the microcontroller has a specification suitable for longer and heavier computation. Meanwhile, for low-level microcontrollers, SVM-Entropy can be applied for classification.



(a) (b)

Figure 2. (a) KNN-Entropy Classification Model, (b) SVM-Entropy Classification Model

4. Conclusion

This study aims to develop a bearing fault detection by comparing the feature extractions. Entropy-based feature extraction can provide a higher accuracy value than descriptive statistics. Hence, Entropy-based feature extraction is very good for increasing bearing fault detection classification accuracy. While the best classifier is obtained when using

KNN, the KNN-Entropy combination has higher accuracy than the SVM-Entropy combination. Thus, in this study, the combination of methods capable of producing the best accuracy is the KNN classifier with Entropy-based feature extraction. Future studies should endeavor to create models that not only maintain the high levels of accuracy, but also optimize the computational efficiency.

Acknowledgment

This research is supported by Young Lecturer Research Grant Universitas Gadjah Mada Number 5985/UN1.P.II/Dit Lit/PT.01.03/2023.

References

[1] G. Xin, N. Hamzaoui, and J. Antoni, “Semi-automated diagnosis of bearing faults based on a hidden Markov model of the vibration signals,” *Meas. J. Int. Meas. Confed.*, vol. 127, no. May, pp. 141–166, 2018. https://doi.org/10.1016/j.measurement.2018.05.040

[2] R. Liu, B. Yang, E. Zio, and X. Chen, “Artificial intelligence for fault diagnosis of rotating machinery: A review,” *Mech. Syst. Signal Process.*, vol. 108, pp. 33–47, 2018. https://doi.org/10.1016/j.ymssp.2018.02.016

[3] A. Guedidi, A. Guettaf, A. J. M. Cardoso, W. Laala, and A. Arif, “Bearing Faults Classification Based on Variational Mode Decomposition and Artificial Neural Network,” in *International Symposium on Diagnostics for Electrical Machines, Power Electronics and Drives (SDEMPED)*, Toulouse, France: IEEE, 2019, pp. 391–397. https://doi.org/10.1109/DEMPED.2019.8864830

[4] R. Azeddine, B. Djamel, and B. Hicham, “A signal processing approach to modeled bearing faults detection in electric system,” pp. 202–206, 2022.

[5] C. Abdelkrim, M. Salah, N. Boutasseta, and L. Boulanouar, “Detection and classification of bearing faults in industrial geared motors using temporal features and adaptive neuro-fuzzy inference system,” *Heliyon*, vol. 5, no. March, p. e02046, 2019. https://doi.org/10.1016/j.heliyon.2019.e02046

[6] P. Zhang, Y. Du, T. G. Habetler, and B. Lu, “A survey of condition monitoring and protection methods for medium-voltage induction motors,” *IEEE Trans. Ind. Appl.*, vol. 47, no. 1, pp. 34–46, 2011. https://doi.org/10.1109/TIA.2010.2090839

[7] J. Pacheco-Cherrez, J. Fortoul-Diaz, F. Cortes-Santacruz, L. M. Aloso-valerdi, and D. I. Ibarra-zarate, “Bearing fault detection with vibration and acoustic signals : Comparison among different machine leaning classification methods,” vol. 139, no. May, 2022. https://doi.org/10.1016/j.engfailanal.2022.106515

[8] M. Lu and C. Chen, “Applied Sciences Detection and Classification of Bearing Surface Defects Based on Machine Vision,” *Adv. Appl. Ind. Inform. Technol.*, vol. 11(4), 2021. https://doi.org/10.3390/app11041825

[9] P. Nivesrangsan, “Bearing Fault Monitoring by Comparison with Main Bearing Frequency Components Using Vibration Signal,” in *2018 5th International Conference on Business and Industrial Research (ICBIR)*, Bangkok, Thailand: IEEE, 2018, pp. 292–296. https://doi.org/10.1109/ICBIR.2018.8391209

[10] S. E. Pandarakone, Y. Mizuno, and H. Nakamura, “Evaluating the Progression and Orientation of Scratches on Outer-Raceway Bearing Using a Pattern Recognition Method,” *IEEE Trans. Ind. Electron.*, vol. 66, no. 2, pp. 1307–1314, 2019. https://doi.org/10.1109/TIE.2018.2833025

[11] Y. Liu, Y. Cheng, Z. Zhang, and S. Yang, “Early Fault Diagnosis of Bearing Faults Using Vibration Signals,” *Proc. 2021 IEEE 3rd Int. Conf. Civ. Aviat. Saf. Inf. Technol. ICCASIT 2021*, pp. 747–751, 2021. https://doi.org/10.1109/ICCASIT53235.2021.9633352

[12] K. K. Song *et al.*, “An Improved Bearing Defect Detection Algorithm Based on Yolo,” *Int. Symp. Control Eng. Robot.*, pp. 184–187, 2022. https://doi.org/10.1109/ISCER55570.2022.00038

[13] S. Lee *et al.*, “A Study on Deep Learning Application of Vibration Data and Visualization of Defects for Predictive Maintenance of Gravity Acceleration Equipment,” *Multidiscip. Digit. Publ. Inst. Appl. Sci.*, vol. 11(4), no. Special Issue: Artificial Intelligence for Sustainable Services, Applications and Education, 2021. https://doi.org/10.3390/app11041564

[14] C. Tastimur, M. Karakose, I. Ayd, and E. Akin, “Defect Diagnosis of Rolling Element Bearing using Deep Learning,” pp. 3–7, 2018.

[15] CRWU, “Case Western Reserve University (CRWU) Bearing Data Center Website

[16] D. Neupane and J. Seok, “Bearing fault detection and diagnosis using case western reserve university dataset with deep learning approaches: A review,” *IEEE Access*, vol. 8, pp. 93155–93178, 2020. https://doi.org/10.1109/ACCESS.2020.2990528

[17] M. Borowska, “Entropy-based algorithms in the analysis of biomedical signals,” *Stud. Logic, Gramm. Rhetor.*, vol. 43, no. 56, pp. 21–32, 2015. https://doi.org/10.1515/slgr-2015-0039

[18] S. M. Pincus, “Approximate entropy as a measure of system complexity,” in *Proceedings of the National Academy of Sciences of the United States of America*, 1991, pp. 2297–2301. https://doi.org/10.1073/pnas.88.6.2297

[19] J. S. Richman and J. R. Moorman, “Physiological time-series analysis using approximate entropy and sample entropy maturity in premature infants Physiological time-series analysis using approximate entropy and sample entropy,” *Am. J. Physiol. Hear. Circ. Physiol.*, vol. 278, pp. H2039–H2049, 2000.

[20] M. Rostaghi and H. Azami, “Dispersion Entropy: A Measure for Time-Series Analysis,” *IEEE Signal Process. Lett.*, vol. 23, no. 5, pp. 610–614, 2016. https://doi.org/10.1109/LSP.2016.2542881

[21] D. Cuesta-Frau, “Slope Entropy: A new time series complexity estimator based on both symbolic patterns and amplitude information,” *Entropy*, vol. 21, no. 12, 2019. https://doi.org/10.3390/e21121167

[22] A. Sharma and L. Mathew, “Bearing Fault Diagnosis Using Weighted K-Nearest Neighbor,” in *2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI)*, IEEE, 2018, pp. 1132–1137. https://doi.org/10.1109/ICOEI.2018.8553800

[23] M. A. Vishwendra *et al.*, “A Novel Method to Classify Rolling Element Bearing Faults Using K -Nearest Neighbor Machine Learning Algorithm,” vol. 8, no. September, pp. 1–11, 2022. https://doi.org/10.1115/1.4053760

[24] C. J. . Burges, “Tutorial on Support Vector Machine for Pattern Recognition,” *Data Min. Knowl. Discov.*, vol. 2, pp. 121–167, 1998. https://doi.org/10.1023/A:1009715923555

[25] C. Z. Hu, M. Y. Huang, Q. Yang, and W. J. Yan, “On the use of EEMD and SVM based approach for bearing fault diagnosis of wind turbine gearbox,” *Proc. 28th Chinese Control Decis. Conf. CCDC 2016*, no. 2, pp. 3472–3477, 2016. https://doi.org/10.1109/CCDC.2016.7531583