**Naïve Bayes Classifier Optimization for Bearing Fault Detection**

Matthew J Kenney

Post University

MAT490: Mathematics Independent Study

Dr. Franklin Orellana

April 26, 2023

**Introduction**

As of February 2023, wind energy comprises 10.2% of all energy supply in the U.S. (U.S. Energy Information Administration, 2023). By 2030, the percentage is predicated to nearly double to 20% (U.S. Energy Information Administration, 2023). The Levelized Cost of Electricity (LCOE) a measure of the cost of energy for a power plant where: (Patapati, 2021). LAZARD (2021) reports Wind energy with the lowest LCOE among other sources (including Solar, Geothermal, Gas Peaking, Nuclear, Coal and Gas Combined Cycle) at $26/MWh. Further, the LCOE of Wind power realized the second largest decline in LCOE between 2009 – 2021 at a 72% drop (the first being Solar PV-Crystalline with a 90% drop). With the increasing demand for cheaper alternative energy, it’s easy to see quickly rising demand predicted in 2030.

Along with rising demand for supply, comes a rising demand for Wind Power Plant machine integrity. In 2015, the National Renewable Energy Laboratory (NREL – a national laboratory of the U.S. Department of Energy) established a database, now called the Gearbox Reliability Database (GRD), to track turbine gearbox faults from utility-scale Wind Power Plants. The purpose of the GRD is to “

* Categorize top wind turbine gearbox failure modes
* Identify possible root causes, and
* Direct future wind turbine gearbox reliability research and development (R&D) activities

(Alliance for Sustainable Energy, LLC, 2023).”

From 750 records in the GRD between 2009 – 2015, 76% of the gearbox failures were caused by faulty bearings. Thus, a strong interest has emerged in predictive bearing health analysis in the wind turbine energy industry. Vibration analysis is an attractive option since vibration data gathering is non-destructive and can be achieved without equipment disassembly. Although, such an analysis takes much experience and knowledge to accurately interpret.

Many machine learning solutions have been proposed to accurately interpret vibrational data. One common challenge in classifying bearing health using vibration analysis is feature selection. From a vibration signal, many features can be extracted. However, most of those features contain similar factors causing feature correlation. Therefore, Naïve Bayes (NB) classifiers are problematic due to lacking the feature independence assumption. Zhang et al. (2018) overcame the issue utilizing a Decision Tree (DT) to select low correlated features. Despite lacking feature independence, Yi et al. (2017) showed NB can achieve higher accuracy than other models with no feature engineering. Furthermore, Zhang et al. (2018) found success with NB and highly correlated features for bearing remaining useful life (RUL) prediction; choosing only features >90% correlation.

This paper proposes a NB machine learning classifier for bearing health using amplitude and velocity readings. The proposed method takes a dataset with amplitude vibration readings during variable shaft speed conditions, sorts the dataset along shaft speed or acceleration, groups the dataset in periods, calculates a statistical feature for each group, then places each feature in a histogram bin. This paper explains the data source, introduces the concept of the Naïve Bayes algorithm, describes the proposed algorithm, compares classifier accuracy with 10 different feature – domain sets, compares classifier accuracy with 100 different period quantity – bins per period quantity combinations with 2 different feature – domain sets, then concludes with a discussion on the findings in this paper and suggestions for future development.

**Data Source**

The data contains vibration amplitude and speed recordings captured by Huang & Baddour, 2019. Five bearing classes were evaluated: healthy, inner fault, outer fault, ball fault, and combination fault. Each class was tested in 4 rotational speed conditions with 3 trials: increasing speed, decreasing speed, increasing then decreasing, decreasing then increasing. The total lot consists of 60 datasets. Vibration readings were sampled at 200,000 Hz for 10 seconds, totaling 2 million datapoints per dataset. Speed was measured by an encoder: EPC model 775, 1024 CPR (Cycles Per Revolution). The encoder scale is not reported but assumed to be x2 based on the reported frequency range per test and the resulting velocity calculations. Vibration amplitudes were measured with an accelerometer: ICP accelerometer, Model 623C01. The experimental bearings were ER16K ball bearings. The test rig is composed of an AC Drive, motor and two bearings for support: one healthy, the other experimental (Appendix, Figure 1).

**Algorithm Design**

*Naïve Bayes – Brief Introduction*

Naïve Bayes equation calculates the *posterior probability*, , of the class according to the predictor where :

The term is the *prior probability* of the class (Zhang Z. , 2019). It describes the probability the class will occur based on the class size and total population. The term , is the *likelihood probability* all the predictors in will occur given the classification is true (Zhang Z. , 2019). The *evidence probability* term, , is the probability all the predictors in X will occur (Zhang Z. , 2019).

The posterior probability is calculated for each possible class. The object is then classified by the class with highest probability:

where, class index and the total number of classes (Zhang Z. , 2019).

*Algorithm Naïve Bayes Calculations*

Prior probabilities are calculated:

where is the number of instances in class ; and is the total population including all classes.

Each class has a single feature value for each period. The range of each period is broken up into bins. If the class value falls within the range of a bin, it is counted as 1. The bins on the extremities of the range for any period include the infinity value, and the bins in the middle are finite. For instance, if a period included 4 bins to cover a range of 20 points, the 3rd highest bin will hold all values between 10 – 15; and the 4th highest bin will hold all values from >= 16. Therefore, for each period, all classes will fall into exactly one bin.

Subsequently, where is the frequency count, the equation for the likelihood probability of bin in period for class becomes:

Likewise, the equation for the evidence probability of bin in period becomes:

*Characteristic Bin Prioritization*

On many occasions, a bin will carry multiple classes for a period. These bins are not useful for classification. Whenever a bin carries only instances of one class, the model considers it a characteristic bin for that class. While training, a dictionary is built for all the classes and their characteristic bins. For testing, the feature value for a dataset is placed in bins for the same number of bins and the same number of periods as the training set. Then, bins used for comparison to a class are determined by Pseudocode 1.

Pseudocode 1: Test Bin Selection

1. Create a list of all the bins occupied by the test set, .
2. Create two more lists:
   1. Bins that are in the test set bin list and not a characteristic bin for the class, .
   2. Bins that are in the test set bin list and not a bin for the class, .
3. If :
   1. Remove all bins in from .
   2. Else: remove all bins in from .
4. Return

With Pseudocode 1, the classifier retains as many bins as possible while prioritizing the characteristic bins. Also, it removes the possibility of a likelihood or evidence of 0% probability in one period to erroneously affect the overall probability.

**Sampling Method**

Pseudocode 2 shows the sampling method. Random sampling is achieved with the pandas.DataFrame.sample method. Each class is sampled independently to ensure an equal number of training samples per class.

Pseudocode 2: Sampling Method

1. Create an array where is the classification, is the dataset index, is the total number of datasets, is the feature value, is the period index, and is the total number of periods.
2. Create the master pandas.DataFrame object with .
3. Initialize pandas.DataFrame list
4. For each , where is the set of all categories:
   1. Randomly select training set using the pandas.DataFrame.sample method.
   2. Append to .
5. Combine all dataframes in to a single DataFrame, .
6. Create the test set, by dropping the row indexes of from .
7. Return ,

**Feature Selection**

*Feature Equations*

Five features and two domains are compared totaling 10 tests. The domains include Velocity and Acceleration. The array for Velocity is calculated:

Where is and array containing encoder pulses, is the timespan of for the test, is cycles per revolution, and is the scale of the encoder (Kelley, 2023) (Precision Microdrives, 2021). Acceleration is calculated by finding the change of pulse rate and dividing it by time. Five features, presented by Hui et al. (2017), are compared: Skewness, Kurtosis, Crest, Shape, Impulse, and Margin (Table 1). Grouping by the sorted Velocity, and then the sorted Acceleration, graphs were constructed for the mean of each feature (Appendix, Figure 2). The Feature – Domain set identifiers are listed in Table 2.

Table 1: Features

|  |  |  |
| --- | --- | --- |
| Feature | Description | Equation[[1]](#footnote-1) |
| Skewness | A measure of symmetry across the sample mean.[[2]](#footnote-2) |  |
| Kurtosis | A measure of tail heaviness of the normal distribution. A heavier tail has more outliers.[[3]](#footnote-3) |  |
| Crest | The peak to RMS ratio of a waveform.[[4]](#footnote-4) |  |
| Shape | Parameter that affects the general shape of a distribution.[[5]](#footnote-5) |  |
| Impulse | Height of peak to the mean signal level.[[6]](#footnote-6) |  |
| Margin | Peak amplitude to squared mean of squared roots of absolute amplitudes – also called “clearance factor.”[[7]](#footnote-7) |  |

Table 2: Feature – Domain IDs

|  |  |
| --- | --- |
| Feature – Domain | ID |
| Skewness – Velocity | SV |
| Skewness – Acceleration | SA |
| Kurtosis – Velocity | KV |
| Kurtosis – Acceleration | KA |
| Crest – Velocity | CV |
| Crest – Acceleration | CA |
| Shape – Velocity | SHV |
| Shape – Acceleration | SHA |
| Impulse – Velocity | IV |
| Impulse – Acceleration | IA |
| Margin – Velocity | MV |
| Margin – Acceleration | MA |

*Feature Test Methods*

Each feature was tested for model Accuracy along Velocity, and then Acceleration. Model parameters were set at 20 periods, 10 bins, and 50% sampling. For each new test, a new training and test set are sampled with Pseudocode 2. For each test, Table 3 shows statistical data for model Accuracy after running the classifier 50 times. Appendix Figure 3 shows the results of Table 3 in a box and whisker plot.

Table 3:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | **Accuracy (%)** | | | | | | |
| **Feature** | | **Mean** | **Std. Dev.** | **Min** | **Max** | **Median** | **25%** | **75%** |
| **Skewness** | | | | | | | | |
|  | Velocity | 56.933 | 5.032 | 50.000 | 70.000 | 56.667 | 53.333 | 60.000 |
|  | Acceleration | 57.200 | 5.527 | 46.667 | 73.333 | 56.667 | 53.333 | 60.000 |
| **Kurtosis** | | | | | | | | |
|  | Velocity | 58.933 | 2.375 | 53.333 | 60.000 | 60.000 | 60.000 | 60.000 |
|  | Acceleration | 59.067 | 2.337 | 53.333 | 60.000 | 60.000 | 60.000 | 60.000 |
| **Crest** | | | | | | | | |
|  | Velocity | 77.867 | 5.938 | 63.333 | 86.667 | 78.333 | 73.333 | 83.333 |
|  | Acceleration | 79.600 | 4.886 | 70.000 | 90.000 | 80.000 | 76.667 | 83.333 |
| **Shape** | | | | | | | | |
|  | Velocity | 63.987 | 11.963 | 50.000 | 90.000 | 60.000 | 53.333 | 73.333 |
|  | Acceleration | 63.333 | 12.293 | 43.333 | 90.000 | 56.667 | 53.333 | 73.333 |
| **Impulse** | | | | | | | | |
|  | Velocity | 74.267 | 5.596 | 60.000 | 86.667 | 73.333 | 70.000 | 76.667 |
|  | Acceleration | 73.933 | 7.429 | 56.667 | 86.667 | 76.667 | 67.500 | 80.000 |
| **Margin** | | | | | | | | |
|  | Velocity | 72.200 | 8.237 | 50.000 | 86.667 | 73.333 | 66.667 | 79.167 |
|  | Acceleration | 73.333 | 7.284 | 56.667 | 83.333 | 75.000 | 66.667 | 80.000 |

Between Velocity and Acceleration, for all features, there appears no significant difference in mean or median. Kurtosis shows the least variation with 60% accuracy nearly across the board in all tests spanning the 25 percentile and above in both domains. However, the best overall test set is Crest in the Acceleration Domain (CA) in terms of data distribution and accuracy. With CA, normally distributed data is apparent:

1. The mean and median values are close at 0.4% points apart, second only to Skewness in the Velocity Domain at 0.266% points apart.
2. The x1.5 whisker lengths are equal on both ends at 6.7% points, which is near equal to the 25 – 75% quartile length, 6.6% points.
3. The dataset exhibits no outliers.

In terms of Accuracy, CA achieved the highest at 79.6%, 95% CI [78.246, 80.954].

**Period and Bin Quantity Optimization**

The algorithm is further optimized by varying bin and period sizes. Feature – Domain sets CA and KA were selected for comparison. The period-quantity parameter and the bin-quantity parameter respectively: . A step-size of 10 units for each parameter was chosen to maximum the test range while minimizing the time cost for testing. Therefore, ten period quantities are each tested with ten bin quantities, totaling 100 tests per Feature – Domain set. The test procedure is in Pseudocode 3.

Pseudocode 3: Bin and Period Size Optimization Procedure

1. Inputs:
   1. , highest number of periods to test
   2. , step size for the number of periods
   3. , highest number of bins to test
   4. , step size for the number of bins
   5. , number of samples to average per test.
2. Initialize results array,
3. Initialize
4. Initialize
5. For each Feature – Domain set, :
   1. Initialize number of periods, .
   2. Initialize number of bins,
   3. For in
      1. Perform Pseudocode 2 steps 1 – 2, return
      2. For in
         1. Initialize test list,
            1. For in :

Perform Pseudocode 2 steps 3 – 7, return and .

Perform model training module, , return model parameters: prior, likelihood, and evidence probabilities.

Perform model testing module, , return accuracy rate, .

Append accuracy rate to .

* + - * 1. Append with tuple,

1. Return

Figure 5 shows that raising gradually raises classifier accuracy, plateauing at , for the KA feature set. Neither feature set improves accuracy with varying , and the CA feature set shows no improvement with varying either parameter. The maximum accuracy achieved is 97% with the KA feature set where and ; and where and (Table 5). The CA feature set achieved at most 83% accuracy where and ; and where and (Table 4). Running the classifier with KA while and achieves an Accuracy of 93.8%, 95% CI [93.053, 94.547] (Table 6).

Table 4: Classifier Accuracy with CA and Varying nPeriods and nBins

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Units= % | **nBins** | | | | | | | | | |
| **nPeriods** | **10** | **20** | **30** | **40** | **50** | **60** | **70** | **80** | **90** | **100** |
| **10** | 69 | 81 | 77 | 70 | 66 | 71 | 71 | 64 | 72 | 70 |
| **20** | 78 | 82 | 78 | 81 | 76 | 77 | 76 | 66 | 70 | 69 |
| **30** | 79 | 77 | 73 | 72 | 77 | 72 | 68 | 70 | 72 | 69 |
| **40** | 79 | 73 | 80 | 79 | 80 | 79 | 76 | 80 | 67 | 71 |
| **50** | 78 | 80 | 77 | 71 | 73 | 80 | 73 | 82 | 76 | 80 |
| **60** | 80 | 73 | 74 | 76 | 80 | 71 | 77 | 77 | 71 | 74 |
| **70** | 82 | 76 | 77 | 80 | 69 | 78 | 69 | 72 | 70 | 68 |
| **80** | 79 | 78 | 83 | 77 | 79 | 77 | 76 | 78 | 72 | 77 |
| **90** | 77 | 80 | 78 | 83 | 82 | 78 | 78 | 72 | 76 | 73 |
| **100** | 73 | 73 | 80 | 80 | 79 | 76 | 77 | 71 | 76 | 71 |

Highlighted entries are the maximum accuracy rates achieved.

Table 5: Classifier Accuracy with KA and Varying nPeriods and nBins

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Units= % | **nBins** | | | | | | | | | |
| **nPeriods** | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 |
| 10 | 60 | 63 | 73 | 73 | 78 | 89 | 89 | 91 | 87 | 82 |
| 20 | 58 | 67 | 74 | 81 | 80 | 90 | 92 | 88 | 89 | 90 |
| 30 | 58 | 68 | 71 | 74 | 78 | 89 | 92 | 91 | 91 | 89 |
| 40 | 60 | 67 | 79 | 78 | 81 | 92 | 88 | 90 | 93 | 91 |
| 50 | 60 | 68 | 76 | 80 | 88 | 89 | 89 | 93 | 90 | 91 |
| 60 | 59 | 71 | 77 | 81 | 80 | 87 | 92 | 91 | 93 | 91 |
| 70 | 60 | 71 | 86 | 84 | 89 | 87 | 92 | 92 | 96 | 94 |
| 80 | 60 | 78 | 78 | 77 | 86 | 86 | 89 | 94 | 97 | 96 |
| 90 | 60 | 69 | 77 | 74 | 81 | 90 | 93 | 92 | 96 | 92 |
| 100 | 60 | 73 | 77 | 86 | 86 | 84 | 91 | 92 | 93 | 97 |

Highlighted entries are the maximum accuracy rates achieved.

Table 6: Classifier Accuracy with KA – Summary Statistics, n=50

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Accuracy (%)** | | | | | | |
| **Feature Set** | **Mean** | **Std. Dev.** | **Min** | **Max** | **Median** | **25%** | **75%** |
| KA | 93.800 | 2.695 | 86.667 | 100.000 | 93.333 | 93.333 | 96.667 |

**Discussion**

Surprisingly, the KA feature set outperformed CA by raising the number of bins per period. It’s not hard to see why KA improves with bin quantity per period:

1. Kurtosis is a measure of outliers in a data set.
2. Different bearing flaw types can be more severe than others.
3. With a variance of severity, there’s a variance of outliers in amplitude readings.
4. More bins per period creates a greater chance for a bearing health class to exclusively occupy a bin; and therefore, creating more *characteristic bins* for the class.
5. With more characteristic bins, there is a greater opportunity for a test set to have more matching characteristic bins for any class.
6. If the test set has more matching characteristic bins than bins overall, the likelihood and evidence probabilities are calculated using only the characteristic bins (Pseudocode 1) – which leads to a more accurate result.

In contrast, Crest is the proportion of peak amplitude and RMS. Since it measures a proportion, scale has no effect – thereby causing little difference between varying bearing flaw types where amplitude scale . An exception is the distinction between combination faults, inner race faults, and all other faults – which is clear in Figure 2 for “Mean Crest.”

Feature selection can be approved by testing other features while varying the number of periods and bins per period. For instance, Figure 2 for “Mean Shape” shows good separation between classes, maybe even more so than Mean Kurtosis. Further analysis with Shape might have equal or better results than Kurtosis. However, since the Shape feature also shows more dispersion (with Acceleration, 5.26 times more standard deviation than Kurtosis), it may still be less accurate due to fewer characteristic bin allocation.

Bin quantity per period can be optimized by varying bin quantity per period based on the values in each period, not to be applied equally across all periods. Bin sizes can be based on the smallest margin between any two classes, thereby nearly guaranteeing a characteristic bin for each class in all periods.

**Appendix**

Figure 1: Test rig

Diagram

Description automatically generated

(Huang & Baddour, 2019)

Figure 2: Mean Features Grouped by the Sorted Velocity and Acceleration

Chart, line chart

Description automatically generated

Chart, line chart

Description automatically generated

Chart

Description automatically generated with medium confidence

Chart

Description automatically generated

Chart, line chart

Description automatically generated

Chart, line chart

Description automatically generated

Figure 3: Classifier Accuracy by Feature – Domain Box and Whisker Plot

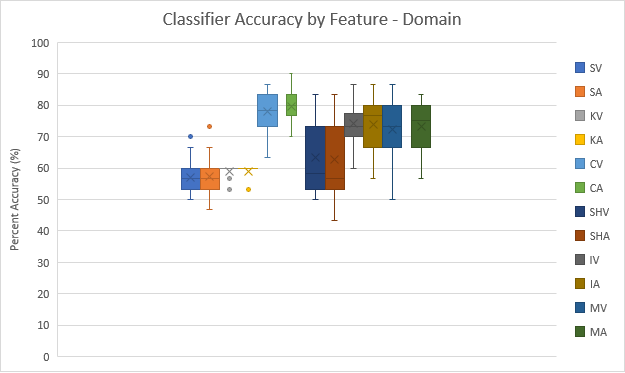


Figure 4: Classifier Accuracy with CA by Varying Period and Bin Quantity

Figure 5: Classifier Accuracy with KA by Varying Period and Bin Quantity

# **References**

Alliance for Sustainable Energy, LLC. (2023). Gearbox Reliability Database. *U.S. Department of Energy | National Renewable Energy Laboratory*. Retrieved from https://grd.nrel.gov/

Altaf, M., Akram, T., Khan, M. A., Iqbal, M., Ch, M. M., & Hsu, C.-H. (2022). A New Statistical Features Based Approach for Bearing Fault Diagnosis Using Vibration Signals. *Sensors, 22*(5). doi:10.3390/s22052012

Glen, S. (2023). *Shape Parameter: Definition and Examples*. Retrieved from statisticshowto.com: https://www.statisticshowto.com/shape-parameter/

Huang, H., & Baddour, N. (2019). Bearing Vibration Data under Time-varying Rotational Speed Conditions. *Mendeley Data, V2*. doi:10.17632/v43hmbwxpm.2

Hui, K. H., Ooi, C. S., Lim, M. H., Leong, M. S., & Al-Obaidi, S. M. (2017). An improved wrapper-based feature selection method for machinery fault diagnosis. *PLoS ONE, 12*(12), 1-10. doi:10.1371/journal.pone.0189143

Kelley, J. (2023). *What’s an Encoder’s PPR, CPR, and LPR? | CUI Devices*. Retrieved from cuidevices.com: https://www.cuidevices.com/blog/what-is-encoder-ppr-cpr-and-lpr

Mobius Institue. (2022). *Vibration Analysis Dictionary: Terms - Mobius Institute*. Retrieved from mobiusinstitute.com: https://www.mobiusinstitute.com/vibration-analysis-dictionary/

National Institute of Standards and Technology. (n.d.). *Measures of Skewness and Kurtosis*. Retrieved from U.S. Department of Commerce: https://www.itl.nist.gov/div898/handbook/eda/section3/eda35b.htm

Patapati, S. (2021). *Powe Plant Cost Comparison*. Retrieved from large.stanford.edu: http://large.stanford.edu/courses/2021/ph240/patapati2/

Precision Microdrives. (2021). *Encoder Resolution: ppr and cpr - Precision Microdrives*. Retrieved from precisionmicrodrives.com: https://www.precisionmicrodrives.com/encoder-resolution-ppr-and-cpr

The MathWorks, Inc. (2023). *Signal Features - MATLAB & Simulink*. Retrieved from mathworks.com: https://www.mathworks.com/help/predmaint/ug/signal-features.html

U.S. Energy Information Administration. (2023). *What is U.S. electricity generation by energy source?* Retrieved from eia.gov: https://www.eia.gov/tools/faqs/faq.php?id=427&t=3

Yi, X. J., Chen, Y. F., & Hou, P. (2017). Fault diagnosis of rolling element bearing using Naïve Bayes classifier. *Vibroengineering PROCEDIA, 14*, 64-69. doi:10.21595/vp.2017.19153

Zhang, N., Wu, L., Wang, Z., & Guan, Y. (2018). Bearing Remaining Useful Life Prediction Based on Naive Bayes and Weibull Distributions. *Entropy, 20*(12), 944. doi:10.3390/e20120944

Zhang, N., Wu, L., Yang, J., & Guan, Y. (2018). Naive Bayes Bearing Fault Diagnosis Based on Enhanced Independence of Data. *Sensors, 18*, 463-480. doi:10.3390/s18020463

Zhang, Z. (2019, April 14). *Naive Bayes Explained*. Retrieved from towardsdatascience.com: https://towardsdatascience.com/naive-bayes-explained-9d2b96f4a9c0

1. (Hui, Ooi, Lim, Leong, & Al-Obaidi, 2017) [↑](#footnote-ref-1)
2. (National Institute of Standards and Technology, n.d.) [↑](#footnote-ref-2)
3. (National Institute of Standards and Technology, n.d.) [↑](#footnote-ref-3)
4. (Mobius Institue, 2022) [↑](#footnote-ref-4)
5. (Glen, 2023) [↑](#footnote-ref-5)
6. (The MathWorks, Inc., 2023) [↑](#footnote-ref-6)
7. (The MathWorks, Inc., 2023) [↑](#footnote-ref-7)