State Classification for Worm Reduction Gears Based on Vibration Signal

Lappeenranta–Lahti University of Technology LUT

Bachelor’s Programme in Software and Systems Engineering, Bachelor's thesis

2023

Shangqi Zhao

Examiner(s): Dr. Doc., Damian Kedziora

Xiaobo Bi, ph.D.

ABSTRACT

Lappeenranta–Lahti University of Technology LUT

LUT School of Engineering Science

Software and Systems Engineering

Shangqi Zhao

State Classification for Worm Reduction Gears Based on Vibration Signal

Bachelor's thesis

2023

43 pages, 11 figures, 14 tables and 1 appendix

Examiners: Dr. Doc., Damian Kedziora and Xiaobo Bi, ph.D.

Keywords: Worm Reduction Gears, Vibration Signal, State Classification, Machine Learning, Cross Validation

With the development of artificial intelligence technologies and the increment of structural complexity of modern mechanical systems, introducing machine learning technology into state classification of rotation machinery has become a hot and vital realm. This research elaborates on the acquisition of source data, steps of processing the data including sample division and transformation of data structure, feature extraction, machine learning models training, examines availability of source data and data features, and undertakes research on applying three machine learning algorithms and different model training methodologies to identify the working conditions and the fault patterns of Worm Reduction Gears and conduct rigorous validating on trained models, comparison and analysis of verification results of different scenarios and come to certain conclusions, eventually revealing performance disparities in different algorithms performance and training methods.

ACKNOWLEDGEMENTS

I would like to express my profound gratitude to Xiaobo Bi, my supervisor, who provided me with source data for the research and his continuous guidance throughout this research. His vast knowledge, insights, and patience have been instrumental in completing this thesis.

SYMBOLS AND ABBREVIATION

Roman characters

R_{p}	rows of data a rotation create	[row]
f	frequency	[Hz]
n	Rotating speed	[r/min]
$R_{\max}$	max complete rotations	
R_{total}	total rows of data	[row]
V	size of validation set	[row]
k	fold	

Constants

g	gravitational acceleration	9,81 m/s

Abbreviations

SVM	Support Vector Machines
GDF	Gear Fault Diagnosis

Table of contents

Appendix

Appendix 1. Research environment and libraries

Figures

Tables

1 Introduction

Rotating machinery is very common in many industries and people's daily lives, machine learning is a fast-growing and future-promising realm applying to many fields. This research uses machine-learning methodologies to classify the working state of the Worm Reduction Gears. In this section, the research's background, current status of the field, outline and objectives of this research are elaborated.

1.1 Research background

With the development of industry and related technologies, manufacturing, transportation, and people's daily life, etc., increasingly depend on various rotating machinery types. The Worm Reduction Gears involved in this paper is a typical representative, it is often used in conveyor belts, medical testing equipment, and elevators. (EVER-POWER, 2022) The critical role of these machines emphasises the need for effective monitoring and maintenance, as the failure of rotating machinery leads to significant economic losses and potential safety accidents. (Abdeljaber et al., 2017)

Although traditional machinery state classification techniques have been widely used (Jardine et al., 2006), regular inspections of rotating machinery are often impossible to implement since the operation of rotating machinery is often invisible. The future of inspections is becoming increasingly clear thanks to the fast development of machine learning technology.

With the rapid development, many machine learning algorithms are matured, for instance, Random Forests, Gradient Boosting, and Support Vector Machines involved in this article, the quantity and quality of related research results have ushered recently, which provides a new feasible direction for mechanical state classification. However, these new methods also bring many challenges related to model training and practical application (Wen et al., 2019).

In general, on the one hand, the rapid development of machine learning and related academic results have provided valuable help to my work; on the other hand, the wide application of rotating machinery and the need for safer and more efficient detection methods makes the

work of practical significance. This paper further explores and contributes to this field of state classification based on machine learning and vibration signals using existing technologies, methodologies, tools, and academic research results.

1.2 Related research

State classification of rotating machinery is a growing field. Fusing traditional methods with innovative, matured machine learning has enriched this field on a large scale. According to the existing academic achievements, studies have successfully used machine learning tools to develop mechanical state classification models, utilising Random Forests, Gradient Boosting, etc., which have shown promise in state classification towards machinery. (Shen et al., 2013)

In “State of the art on vibration signal processing towards data-driven gear fault diagnosis”, by Zhang, S. et al., delineate the intricacies of vibration signal characteristics pertinent to common gear faults and extensively evaluate feature extraction techniques, notably adaptive mode decomposition, deconvolution, mathematical morphological filtering, and entropy. Simultaneously, it underscores the growing significance of machine learning for fault recognition. Through the assessment, the authors highlight the importance of precise feature extraction and the potential of data-driven gear fault diagnosis (GFD) in modern diagnostic procedures. (Zhang et al., 2022)

Under current status, though many existing studies have achieved specific results, there are still gaps in the research on Worm Reduction Gears, which limits the application of machine learning technology in Worm Reduction Gears (EVER-POWER, 2022). Simultaneously, there is currently a lack of a complete experimental system using machine learning technology and the vibration signal of rotating machinery to perform working state recognition. There are gaps in the understanding of the state classification process. This paper aims to use existing technologies to conduct research and achieve a specific result level for reference with the help of existing academic results.

1.3 Content and structure

This research contains the following content: The second chapter, methodology, elaborates on the implementation principle of the machine learning algorithm used in the research and its application in this research, as well as a detailed analysis of the validation methods involved in the research; the third chapter, conduct describes the apparatus, tools, programming languages and libraries used in the research, and the acquisition and pre-processing of source data are explained in detail, the research framework and the research details to be realised are also clarified; The fourth chapter, research outcome, endeavours of listing the results of each research item and analysing them to draw preliminary conclusions. In the final chapter, conclusion, the research outcomes are discussed and analysed in detail, and a general conclusion can be drawn. With all the efforts, this research will provide reference and contribution to product application and scientific research in related fields.

2 Methodologies

Within the realm of machine learning, the judicious selection of algorithms and validation methodologies critically underpins the academic rigour and validity of the research. This chapter meticulously elaborates on the algorithms operationalised in our investigation, and the methods harnessed for model validation. Their pivotal role, inherent functionalities, and integrative application concerning this research are systematically expounded.

2.1 Algorithms

As embarking the section of research methodologies, it's essential to shed light on the primary algorithms utilized. These algorithms form the cornerstone of our analytical approach. In the subsequent sections, the intricacies of Random Forest, Gradient Boosting, and Support Vector Machines will be elaborated, including listing their unique attributes and significance in the research.

2.1.1 Random Forest

Random Forest is a standard machine learning algorithm known for its robustness and versatility. At its core, it leverages multiple decision trees to generate predictions or classifications (Nik, 2023). This ensemble approach allows collective “voting” of each decision tree, improving accuracy and reducing the risk of overfitting (Yang et al., 2019). By selecting a random subset of the dataset and a different set of features for each tree, Random Forest ensures diversity in its model-building process. The versatility of this approach makes Random Forests well-suited for handling large datasets with complex inter-variable relationships, ensuring that individual features or outliers do not unduly influence the overall model (Walker et al., 2013).

Scikit-learn, used in this research, provides a comprehensive and efficient Random Forest implementation. The library simplifies data preprocessing, from handling missing values to encoding categorical data (Nik, 2023). Scikit-learn provides intuitive methods for model fitting, prediction, and evaluation. Users can easily split data, train models, and evaluate their

performance using accuracy metrics. The machine learning workflow is greatly simplified (Chen et al., 2020). Meanwhile, Scikit-learn's ability to visualise decision trees in Random Forests is a notable feature that demystifies the black-box nature of machine learning models (Zhao et al., 2021).

Employing Random Forest in Scikit-learn is a wise choice for complex or massive datasets. The application of this algorithm yields a high accuracy rate, and the trained model in similar scenarios can achieve an accuracy rate of 97% (Nik, 2023). Given the ease of use of the library, combined with the inherent advantages of Random Forests, it is an excellent choice for various machine-learning challenges (Walker et al., 2013).

2.1.2 Gradient Boosting

Gradient Boosting is another ensemble machine learning technique that differs fundamentally from Random Forest. While Random Forest builds multiple trees in parallel, Gradient Boosting constructs trees sequentially (Yang et al., 2019). This algorithm thrives on the philosophy of improving upon the mistakes of previous trees. Each successive tree attempts to correct the errors made by its predecessor, employing the concept of gradients in the boosting process (Chen et al., 2020). By adapting to the residuals or the differences between predicted and actual values, Gradient Boosting can, over many iterations, hone its predictions to minimise errors. Its strength lies in optimising model performance by focusing on where it goes wrong, allowing it to perform exceptionally well even with complex, non-linear data (Walker et al., 2013).

Scikit-learn, as a comprehensive machine learning library, naturally also provides an implementation for Gradient Boosting (EVER-POWER, 2022). The library ensures that even complex techniques, for instance, Gradient Boosting are easily accessible and manageable. Similar to Random Forests, in Scikit-learn, users can easily specify parameters, for instance, the learning rate (which controls how well each tree corrects the errors of its predecessors) or the number of boosting stages to run. (Zhao et al., 2021) allows data splitting, model training, and performance evaluation methods.

Given Gradient Boosting's ability to enhance its performance iteratively, it often becomes one of the top-performing algorithms for various machine-learning tasks. When

implementing functionalities with Gradient Boosting in Scikit-learn, users benefit from the functional implementation of Gradient Boosting and the efficient tools provided by the library (Chen et al., 2020). This synergy produces models that are both accurate and computationally efficient. The iterative nature of Gradient Boosting makes it particularly suitable for misclassification scenarios, which provides another reason for its choice in critical applications. (Louppe, 2015)

2.1.3 Support Vector Machine

Support Vector Machines (SVM) is a supervised learning model with prominent classification and regression analysis applications. The core principle of Support Vector Machines is to find a hyperplane that best divides a dataset into classes (Yang et al., 2019). This optimal hyperplane is the one that preserves the maximum and minimum distance to the training samples, and this distance is usually called the margin. Support Vector Machines are beneficial when dealing with high-dimensional spaces and when classes can be separated linearly or by custom kernel functions (Wei et al., 2019).

One of the outstanding properties of SVMs is their kernel tricks. This capability allows them to manage non-linearly separable data by mapping it into a high-dimensional space where it becomes linearly separable (Walker et al., 2013). This kernel function can be polynomial, radial basis function (RBF), sigmoid, etc., providing inherent flexibility for Support Vector Machines (Shi & Zhang, 2021).

This research uses the Scikit-learn library to implement Support Vector Machines. Using Scikit-learn, SVMs can be efficiently trained, evaluated, and tuned through its simple interface. Parameters of regularisation, kernel type, and other SVM-specific configurations can be adjusted effortlessly, enabling researchers to tailor SVMs to the specific needs of their datasets (EVER-POWER, 2022). Given its robustness against overfitting, especially in high-dimensional spaces, and its adaptability to kernel tricks, SVM is an indispensable tool for fine-grained classification tasks (Nik, 2023).

2.2 Validation methodologies

In machine learning, validation plays a pivotal role in this assessment, allowing people to understand and measure the generalisation capabilities of models. In the subsequent discussions, Holdout Validation and K-Fold Cross-Validation will be elaborated. While both serve the primary purpose of model validation, they have unique advantages, disadvantages, and ideal use cases. The insights from these methodologies will help select the optimal approach tailored to this research needs.

2.2.1 Holdout Validation

Holdout Validation is one of the most straightforward model validation techniques used in machine learning. It involves splitting the available dataset into training and validation sets (Yang et al., 2019). Models are trained on the training set and then validated or tested on the validation set. This approach provides a quick and straightforward way to determine how well a machine-learning model will perform on unseen data. The main advantages of this approach are its simplicity and computational efficiency, especially for large datasets.

For our specific scenario, Holdout Validation may not be the most efficient method considering the characteristics and amount of source data. A significant limitation is that the model's performance may vary greatly depending on the data points in the training set and the data points in the validation set, especially given the periodicity of rotating machine vibration data. This difference can lead to misleading performance metrics, especially if the dataset is not large enough or does not have a balanced representation of all classes (Huang et al., 2021). Therefore, Holdout Validation is only used here to compare accuracy with K-Fold Cross-Validation in 2.2.2.

2.2.2 K-Fold Cross-Validation

K-Fold, also known as N-Fold, cross-validation is a robust resampling technique designed to evaluate the generalisation ability of machine learning models to unknown data (Yang et al., 2019). Typically, data is divided into training, validation, and test sets. K-Fold Cross-Validation solves this problem by dividing the dataset into “k” different subsets (or folds).

During training, a single is used for validation, while the other folds are used for training. This will be repeated “k” times, once for each fold as the validation set, ensuring that each data point appears once in the validation set and k-1 times in the training set (Singh et al., 2021). The method thus provides a less biased model evaluation, providing an aggregated performance measure across all folds.

The main challenge in machine learning is not just training a model but making sure it generalises well to unseen data. Simple data splits can introduce high bias, especially when limited data leads to overly optimistic or pessimistic performance (Walker et al., 2013). As a single data subset contains the vibration data generated by the Worm Reduction Gears for one revolution of the input shaft or output shaft, K-Fold Cross-Validation provides a more robust assessment of model performance than Holdout Validation, as K-fold is not a single train-test split. Still, multiple ones are performed to ensure that the model is evaluated on different subsets of data. In our case, a single subset of data contains the vibration data Worm Reduction Gears when the input shaft or the output shaft rotates one revolution. This multiple evaluation allows for a better understanding of the average performance of a model and its variability across different data subsets (Manna, 2020). As we want to maximise data utility and gain a comprehensive understanding of the underlying performance of the model, so K-Fold Cross-Validation is a more appropriate choice.

The quantity and distribution of source data are significant, and the value of k plays an important role here. If it is too low, the bias is high; if it is too high (similar to Leave One Out CV or LOOCV), the variance will increase (Manna, 2020). In our specific scenario, the total size of the data set is limited. Considering the constraints and the need to maximise data utility without introducing bias, 50% cross-validate a more balanced choice. As 50% cross validate is not very computationally expensive while producing reliable performance (Zhao et al., 2021).

3 Conduct

This section aims to elaborate on the research conduct backgrounds, for instance, hardware and software infrastructures. This section also elaborates the acquisition, structure, and processing of source data. Finally, the research architecture, code design and research conduct items are clarified.

3.1 Research tools and environment

Delving deeper into the apparatus and infrastructure, this section provides an exhaustive elucidation of the tools employed. The two most essential components of the research foundation are hardware apparatus and software infrastructures. This section will elaborate on hardware apparatus and software infrastructures.

3.1.1 Research apparatus

Figure 1. Worm Reduction Gears

A Worm Reduction Gears (Figure 1) is a specialised mechanical device designed to decrease rotational speed using worm gears and producing power from the output shaft. The initial step of the whole research is to obtain the vibration data of various operating status (Table 1, Figure 2, and Figure 3) from the Worm Reduction Gears under different working conditions.

Table 1. Worm Reduction Gears working status mapping table

ID	Status
1	Normal
3	Worm wheel position offset
9	Input shaft worm jitters
13	Bearing roller scratch
15	Worm wheel scratch

Figure 2. Bearing roller scratch

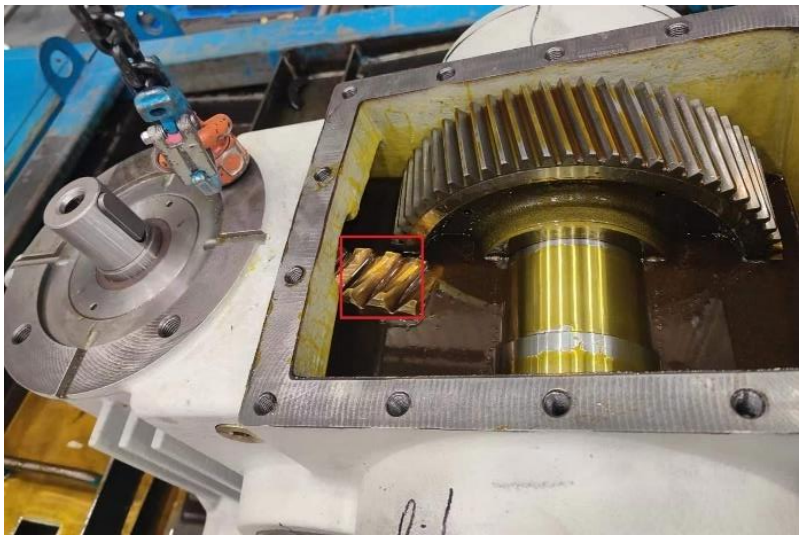

Figure 3. Worm wheel scratch

To ensure the reliability of this research and the persuasiveness of the results, two Worm Reduction Gears were collected vibration data under different working conditions. The data acquisition work of each Worm Reduction Gears is completed by three high-frequency acquisition acceleration sensors, that is, the whole research involves two Worm Reduction Gears and six acceleration sensors. The sensors and their output data are introduced in latter chapters.

3.1.2 Software infrastructure

This section is crucial for the research as most of the content of this research is related to software and systems. To achieve faster operating efficiency, more straightforward environment configuration and system control, this research chooses Ubuntu (a Linux distribution), which is typical and outstanding in the production environment. Considering the development cost and the richness and ease of use of machine learning libraries, this research uses Python3 as the primary programming language to implement data processing, algorithm packaging, training and verifications, export of verification results, etc. And use Bash scripts to implement the execution of multiple validations and the export of the test log. And use HTML and JavaScript to realise the auxiliary tool for calculating the classification accuracy of the verification result with a graphical user interface.

In implementing the main functionalities with Python3, the following Python libraries are used:

- openpyxl: for opening and closing Excel source files for loading data
- pandas: Used to preprocess data, extract data features, and transform them into structures suitable for training machine learning models.
- numpy: used to receive and return classification results.
- scikit-learn: A machine learning library that uses Random Forest, Gradient Boosting, Support Vector Machines, and scaler training models.
- matplotlib: used for data visualisation, showing the distribution of source data and the values of the four characteristics of mean, standard deviation, skewness, and kurtosis.

In addition, a JavaScript library is used as an auxiliary tool for calculating classification accuracy implemented by HTML and JavaScript:

- PapaParse: for parsing .csv files to load validation results and unparse calculation results.

For more detailed information about the software-related tools and dependencies used in this experiment are placed in the appendix.

3.2 Source data

For machine learning model training, the meticulous gathering and structuring of empirical data remain paramount. This section thoroughly explores the acquisition, organisation, and pre-processing of vibration signals from the Worm Reduction Gears mechanism. Through visual representations and comprehensive elucidations, this chapter delineates the modus operandi behind data assimilation and subsequent computational processing, paving the way for the research undertaking.

3.2.1 Data collection

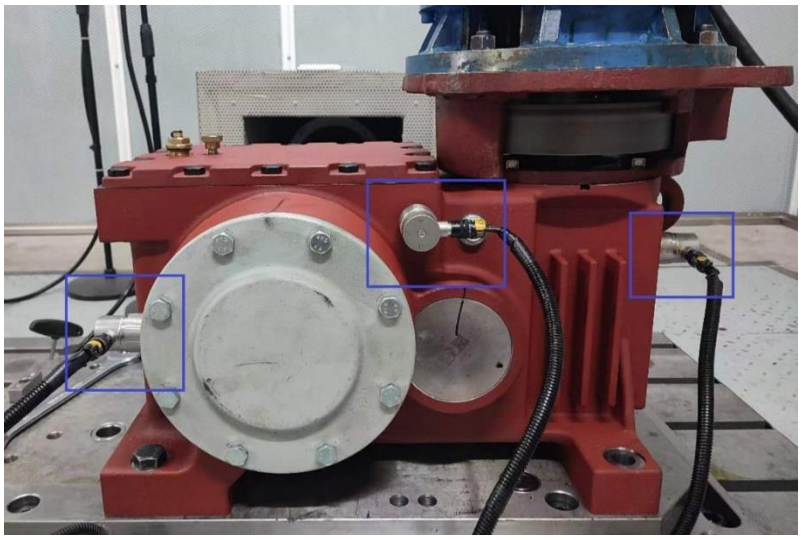

Figure 4. Sensor's position at the Worm Reduction Gears

Accurate and effective data acquisition is crucial in machine learning-based mechanical state classification. The raw data used in this research were detected by three 3-D acceleration sensors fixed on the housing of the Worm Reduction Gears (Figure 4), which is suitable for capturing vibration signals from various rotating machinery (EVER-POWER, 2022). These highly efficient sensors collect acceleration data in g to nine decimal places at 25,600 Hz.

3.2.2 Data structure

The raw data is stored in several Excel sheets in .xlsx format (Figure 5). Each dimension of the sensor will be held in a worksheet. Each sensor detects acceleration in three dimensions and detects five kinds of Worm Reduction Gears Work status data. Therefore, each Excel file represents all the data originating from a sensor, and it contains fifteen worksheets with one column of data, recording the acceleration data detected by a sensor in different dimensions and states. The three Excel files contain sensor data at three different locations representing the vibration data of one machine. This research involves the vibration data of two Worm Reduction Gears, so there are six Excel files.

A4 | 0.053297082

	A	B	C
1		NaN	
2	Standard\I25600 Hz		
3	Real	NaN	
4	0.053297		
5	-0.044002		
6	-0.146101		
7	-0.11639		
8	0.078697		
9	0.128867		
10	0.020618		
11	-0.040545		
12	-0.170591		
13	-0.199997		
14	-0.117221		
15	-0.017961		
16	0.075979		
17	0.143799		
18	0.245953		
19	0.199502		
20	0.135402		

A7 | 0.062695983

	A	B	C
1		NaN	
2	Standard\I25600 Hz		
3	Real	NaN	
4	-0.006932		
5	-0.004263		
6	0.074098		
7	0.062696		
8	0.037241		
9	0.039544		
10	0.087271		
11	0.140556		
12	0.104738		
13	0.04744		
14	0.108585		
15	0.115737		
16	0.033052		
17	0.060943		
18	0.009344		
19	0.007927		
20	0.101043		

A16 | -0.018724516

	A	B	C
1		NaN	
2	Standard\I25600 Hz		
3	Real	NaN	
4	0.099259		
5	-0.03169		
6	-0.024887		
7	0.028655		
8	0.122393		
9	0.137405		
10	0.040643		
11	-0.033437		
12	-0.080101		
13	-0.009686		
14	0.126839		
15	0.080364		
16	-0.018725		
17	0.030133		
18	0.0593		
19	0.088505		
20	0.134528		

Figure 5. Source data in Excel files

3.2.3 Data preprocessing

Figure 6. The steps of data pre-process

When executing in the program, first read the Excel file into the cache with the help of the “openpyxl” library, and then convert the training Dataset into a “DataFrame” instance defined in the “pandas” library through the methods defined in advance (Figure 6), and then adjust the structure of the data making it easier for machine learning. The data is firstly

divided into five datasets according to the state, and each data set contains 3-D data of three sensors of a specific working state, that is, a total of Nine columns of data (Figure 7).

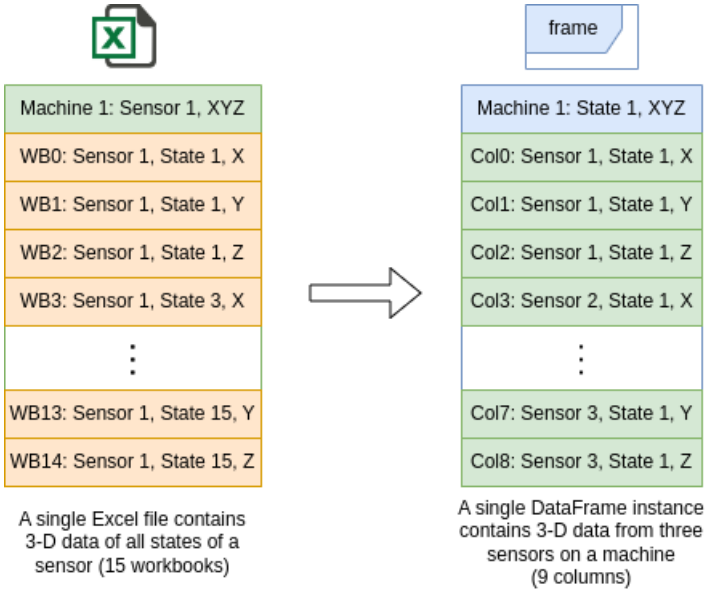

Figure 7. Structure of data before and after processing

In the last step of data preprocessing, the program will divide the data set into Training Dataset (training set) and Validation Dataset (validation set) according to the K-Fold Cross-Validation. The training set is used for training the machine learning model, and the validation set is used to verify the classification accuracy of the trained machine learning model.

To conduct cross-validation, the source data must be cut into multiple fragments according to rotating speed in the first place. Rotating speed n of input shaft is 1475 RPM (rows per minute), and output shaft is of 305 RPM. By using the equation, we can first get the number of rows of data a single rotation creates R_{p}.

$$R_{\mathrm{p}} = \left\lceil \frac{f}{n} \right\rceil \tag{1}$$

Then with R_{p} and total amount of rows R_{total}, get the maximum amount of complete rotation data fragments it can have R_{max}.

$$R_{\mathrm{max}} = \left\lfloor \frac{R_{\mathrm{total}}}{R_{\mathrm{p}}} \right\rfloor \tag{2}$$

And divided into training sets and validating data according to fold value k.

$$V = \frac{R_{\max}}{k} \tag{3}$$

The length of the pre-processed data frame has two forms, input shaft-cut data and output shaft-cut data. In this research the smallest K-Fold Cross-Validation training machine learning model unit is the amount of data equivalent to one rotation of the Worm Reduction Gears's input shaft or output shaft.

3.3 Research Architecture and implementation

This chapter meticulously detail the underpinnings and limitations of our code design that subsequently applied to the research. This chapter also underscoring the multifaceted research undertakes, introducing forthcoming research and their significant and expectancy.

3.3.1 Code design pattern

```
class GradientBoosting:
    def __init__(self, *sensor_data_list: SensorData) -> None:
        features = []
        labels = []
        for sensor_data in sensor_data_list:
            df = sensor_data.dataframing()
            features.append(self.extract_features(df))
            labels.append(sensor_data.state)
        features_df = pd.DataFrame(features)
        labels_s = pd.Series(labels)
        scaler = StandardScaler()
        features_scaled = scaler.fit_transform(features_df)
        clf = GradientBoostingClassifier(n_estimators=100, random_state=42)
        clf.fit(features_scaled, labels_s)
        self.classifier = clf
        self.scaler = scaler
```

Figure 8. Code snippet

The code implementation in this research follows the object-oriented programming (OOP) specification, which hides the implementation details but provides sufficient and easy-to-use interfaces for production (i.e. training model and verification). Follow the format suggested

in the official Python documentation, using snake case nomenclature for variable names, etc. striving to make them friendly and easy to understand. (Figure 8)

The code implementation of this research only aims to achieve the most basic functionalities required by the research. It might not be of the best performance. For instance, it can be significantly improved to achieve better I/O performance by transferring Excel files into .csv files, or storing data in a database, and lowering the memory overhead by using and releasing resources more rationally, etc.

3.3.2 Validation design

This research uses three algorithms: Random Forest, Gradient Boosting, and random vector machine. To ensure the consistency and reproducibility of the verification results, the "random_state" variable is uniformly limited to 42 when the algorithm is packaged. And "n_estimators" of Random Forest and Gradient Boosting are uniformly set to 100 (code: "RandomForestClassifier(n_estimators=100, random_state=42)"). Simultaneously, after verification, it is concluded that "linear" is an effective SVM kernel for this research (code: "SVC(kernel='linear', random_state=42)")

Validations of this research take the four most common features of the field:

- Mean: The average value of all the numbers in the column.
- Standard Deviation: Measures the amount of variation or dispersion of a set of values.
- Skewness: Measures the asymmetry of the probability distribution of a random variable about its mean.
- Kurtosis: Measures the probability distribution deviates from a normal distribution in terms of the heaviness of its tails.

The forthcoming research will be conducted as following:

- Examination of Source Data Availability

  This segment predominantly examines the classification precision achieved when the model is iteratively trained with consistent data quantities, utilising the rotation of the output shaft as the base unit. Employing the foundational machine learning algorithm,

this phase primarily serves as an initial validation of the research's premises and aims to discern the intrinsic training value of the data.

- Assessment of Distinctiveness in Source Data

  Central to this facet is verifying the model's classification accuracy by extracting data from the entire training set. The goal is to discern the conspicuousness of variations between vibration signal data across varied states and evaluate the feasibility of conducting training with a minimalistic feature set limited to four distinctive features.

- Comparative Analysis of Holdout Validation and 5-Fold Cross-Validation Techniques

  This research component is committed to delineating the accuracy differentials between the Holdout Validation and 5-fold cross-validation methodologies. Factors of source data, training algorithm, and parameters are maintained constant, and the machine learning model Random Forest is adopted for training endeavours.

- Differential Analysis of Input and Output Shaft Data Training Efficacy

  This component explores the classification efficacy for models trained by the input shaft-cut data as the minimum model training unit and output shaft-cut data as the minimum model training unit under different algorithm scenarios.

- Comparative Evaluation of Machine Learning Algorithms: Random Forest, Gradient Boosting, and Support Vector Machines

  This segment is dedicated to a comprehensive juxtaposition of the classification accuracy variances exhibited by the three algorithms described above, especially when subjected to different source data and distinct training methods.

- Examine Efficacy of Cross-Machine State Classification

  This component of research examines the accuracy and features of cross-machine state classification. As in real-life production, models are trained in advance and monitor vibration signals timely. Hence, cross-machine state classification is the research that closest real-life production scenario. This is the most valuable research but also the most challenging one.

4 Research Outcome

In this chapter, six research items introduced in the previous chapter will be described in detail and conducted. And their outcomes will be shown and discussed in the following paragraphs.

4.1 Source data availability

In this research, the original acquired data is directly divided into a training set and a validation set. The features of the training set are extracted only once, and the Random Forest model is trained. The availability of the source data is tested in extreme cases by continuously compressing the size of the validation set. The validation set sizes are 8192 rows, 4096 rows, 2048 rows, 1024 rows, 512 rows, and 256 rows.

Table 2. The classification result of the data of Worm Reduction Gears 1

validation set size	state 1	state 3	state 9	state 13	state 15
8192	1	3	9	13	15
4096	1	3	9	13	15
2048	1	3	13	13	15
1024	1	3	9	13	15
512	1	3	9	9	15
256	1	3	9	13	1

Table 3. The classification result of the data of Worm Reduction Gears 2

validation set size	state 1	state 3	state 9	state 13	state 15
8192	1	3	9	13	15
4096	1	3	9	13	15
2048	1	15	3	13	15
1024	15	15	9	9	15
512	15	15	3	9	15
256	15	15	3	9	15

The validation results (Table 2, Table 3) show that a high accuracy rate of state classification can be guaranteed when the verification data has more than 4096 lines, that is, the verification set accounts for 2% of the source data. This result shows that the source data is valid and highly reliable, which can guarantee the effectiveness of subsequent research.

Simultaneously, we can also see that the data collected from Worm Reduction Gears 1 is more reliable than Worm Reduction Gears 2, which has also been verified in later research.

4.2 Feature distinctiveness

This step mainly examines the differences in mean, standard deviation, skewness, and kurtosis of the source data under different working states of the Worm Reduction Gears to understand the usability of feature selection initially, that is, the ability to separate data in various states through the selected four data characteristics.

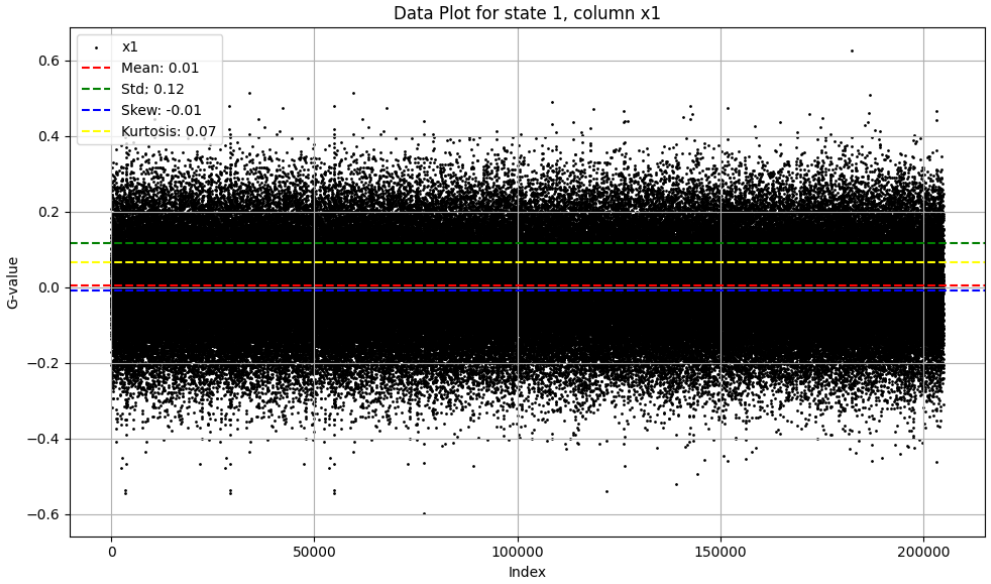

Figure 9. Source data (Worm Reduction Gears 1, state 1, x1) distribution and features

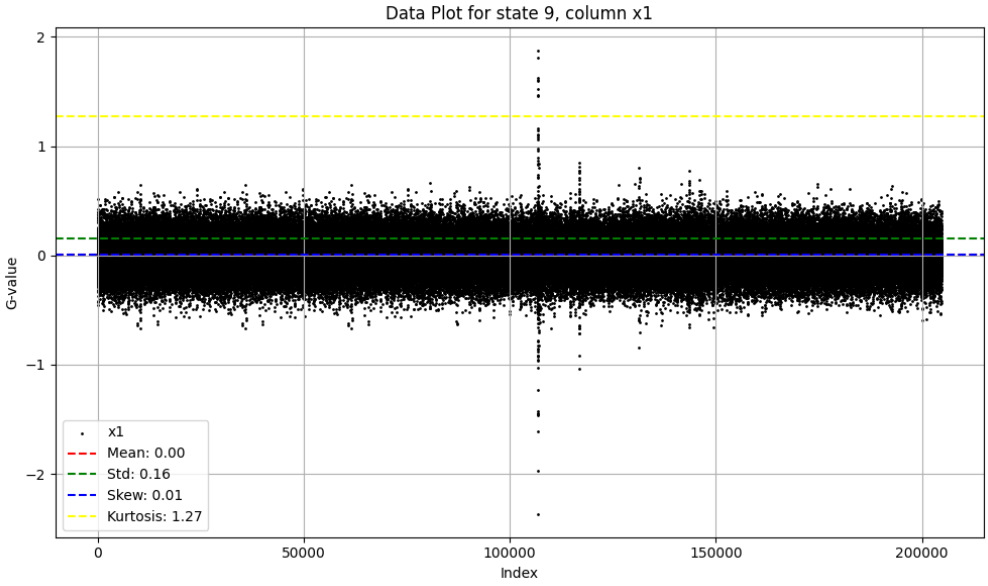

Figure 10. Source data (Worm Reduction Gears 1, state 9, x1) distribution and features

The result indicates that data from the same Worm Reduction Gears, sensor, and detect axis but of different working states have apparent differences (Figure 9, Figure 10), and the selected four features can show the difference. Hence, the selection of features is valid and suitable for model training, it often does not cause underfitting or overfitting.

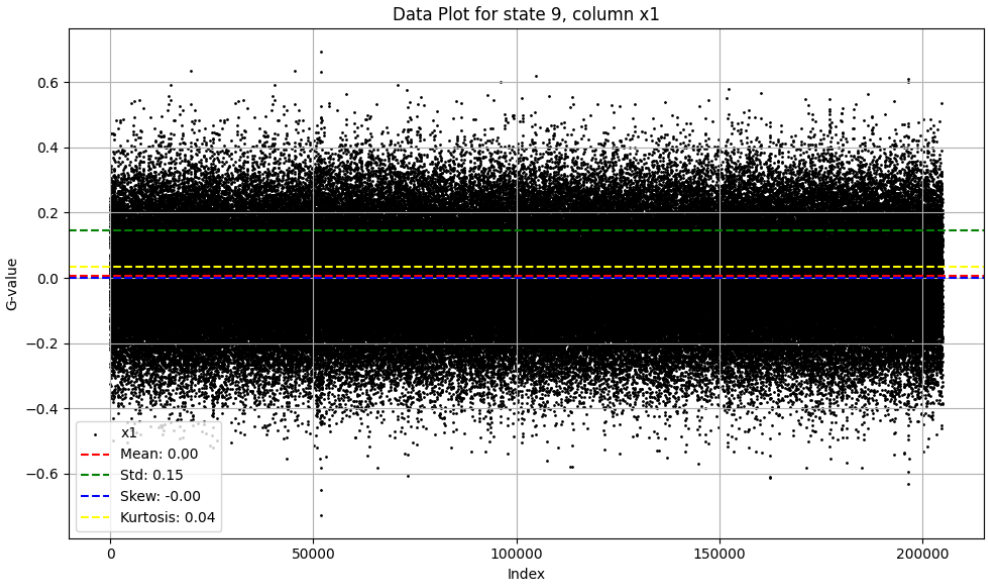

Figure 11. Source data (Worm Reduction Gears 2, state 9, x1) distribution and features

However, from the data of different Worm Reduction Gears but of the same working state and detect axis (Figure 11), it shown that data from two Worm Reduction Gears are of enormous difference, which might lead to low cross-machine working state classification accuracy. This part will be discussed in detail afterwards, and the conjecture will be confirmed.

4.3 Holdout Validation and 5-fold cross-validation

This part of the research explores the efficacy comparison between Holdout Validation and 5-fold cross-validation. In this section, the two groups chose the Random Forest implementation and the same parameter settings (as previously mentioned) to achieve the sole variable environment. Set the same training and verification ratio as 5-fold cross-validation for the Holdout Validation group; 80% or 163840 rows of data are used for training, 20% or 40960 rows of data to verify the model, and the cross-validation group uses output shaft-cut data to train the model, Holdout Validation scales a one-time training model. Using two sets of data from two Worm Reduction Gears to verify results respectively improves the credibility.

Table 4. Validations accuracy based on data of Worm Reduction Gears 1

validate method\state	State 1	State 3	State 9	State 13	State 15	Total
Holdout Validation	100%	100%	100%	97.436%	100%	99.487%
5-fold cross-validation	100%	100%	100%	100%	100%	100%

Table 5. Validations accuracy based on data of Worm Reduction Gears 2

validate method\state	State 1	State 3	State 9	State 13	State 15	Total
Holdout Validation	97.436%	100%	97.436%	97.436%	100%	98.462%
5-fold cross-validation	100%	100%	100%	100%	100%	100%

It can be seen from the verification results (Table 4, Table 5) that the accuracy rate of 5-fold cross-validation is higher than that of Holdout Validation, which is in line with the general law. Simultaneously, all the classification results of 5-fold cross-validation are correct, hence

its classification potential is higher than the upper limit of current research. Therefore, the rest of the studies adopted 5-fold cross-validation as the only model validation method. Simultaneously, the model trained by the data from Worm Reduction Gears 1 has a higher accuracy rate than Worm Reduction Gears 2, which is consistent with the first research's results.

4.4 Input and output shaft data training efficacy

This part is one of the cores of this research, so a comprehensive experimental verification has been carried out. This part of the research is dedicated to comparing the accuracy of the input shaft-cut data and output shaft-cut data training models under different algorithm scenarios.

Table 6. Sample division classification accuracy of Worm Reduction Gears 1

Algorithm	Input/output shaft-cut data	State 1	State 3	State 9	State 13	State 15	Total
Random forest	input shaft-cut data	100%	100%	100%	100%	100%	100%
	output shaft-cut data	100%	100%	100%	100%	100%	100%
Gradient boosting	input shaft-cut data	99.50%	98.00%	99.00%	98.50%	100%	99.00%
	output shaft-cut data	100%	100%	100%	95.00%	100%	99.00%
Support Vector Machines	input shaft-cut data	100%	100%	98.50%	100%	100%	99.70%
	output shaft-cut data	100%	100%	97.50%	100%	100%	99.50%

Table 7. Sample division classification accuracy of Worm Reduction Gears 2

Algorithm	Input/output shaft-cut data	State 1	State 3	State 9	State 13	State 15	Total

Random Forest	input shaft-cut data	100%	100%	100%	100%	100%	100%
	output shaft-cut data	100%	100%	100%	100%	100%	100%
Gradient Boosting	input shaft-cut data	100%	97.50%	98.00%	99.50%	100%	99.00%
	output shaft-cut data	100%	97.50%	98.00%	99.50%	100%	99.00%
Support Vector Machines	input shaft-cut data	100%	100%	100%	100%	100%	100%
	output shaft-cut data	100%	100%	100%	100%	100%	100%

Table 8. Average classification accuracy of all sample division groups

	State 1	State 3	State 9	State 13	State 15	Total
input shaft-cut data	99.917%	99.250%	99.250%	99.667%	100%	99.617%
output shaft-cut data	100%	99.583%	99.250%	99.083%	100%	99.583%

The verification results (Table 6, Table 7, Table 8) show that the classification capabilities of input shaft-cut data and output shaft-cut data are not much different under various circumstances, and most of them can be correctly classified. Which means the periodic data created by input shaft-cut and output shaft-cut are both valuable for training and have consistent properties based on their respective periods.

However, it is also worthwhile to notice that the stability of the input shaft-cut data set was generally inferior to the output shaft-cut data, as it manifested sporadic classification errors. This inconsistency can be attributed to the rapid rotation speed of the input shaft, which led to smaller data volumes in each sub-training set, underscoring the importance of appropriately sizing the training set.

4.5 Comparative evaluation of machine-learning algorithms

This part is one of the cores of this research, so a comprehensive experimental verification has been carried out. This part of the research is dedicated to comparing the accuracy of the

three algorithm models of Random Forest, Gradient Boosting, and Support Vector Machines under the same source data and parameter configuration scenarios.

Table 9. Algorithms based classification accuracy of Worm Reduction Gears 1

Input/output shaft-cut data	Algorithm	State 1	State 3	State 9	State 13	State 15	Total
input shaft-cut data	Random Forest	100%	100%	100%	100%	100%	100%
	Gradient Boosting	99.50%	98.00%	99.00%	98.50%	100%	99.00%
	Support Vector Machines	100%	100%	98.50%	100%	100%	99.70%
output shaft-cut data	Random Forest	100%	100%	100%	100%	100%	100%
	Gradient Boosting	100%	100%	100%	95.00%	100%	99.00%
	Support Vector Machines	100%	100%	97.50%	100%	100%	99.50%

Table 10. Algorithms based classification accuracy of Worm Reduction Gears 2

Input/output shaft-cut data	Algorithm	State 1	State 3	State 9	State 13	State 15	Total
input shaft-cut data	Random Forest	100%	100%	100%	100%	100%	100%
	Gradient Boosting	100%	97.50%	98.00%	99.50%	100%	99.00%
	Support Vector Machines	100%	100%	100%	100%	100%	100%
output shaft-cut data	Random Forest	100%	100%	100%	100%	100%	100%
	Gradient Boosting	100%	97.50%	98.00%	99.50%	100%	99.00%
	Support Vector Machines	100%	100%	100%	100%	100%	100%

Table 11. Average classification accuracy of all algorithm groups

	State 1	State 3	State 9	State 13	State 15	Total
Random Forest	100%	100%	100%	100%	100%	100%
Gradient Boosting	99.833%	98.083%	98.667%	98.417%	100%	99.000%
Support Vector Machines	100%	100%	99.083%	100%	100%	99.817%

When examining the classification ability of different algorithms with tables above (Table 9, Table 10, Table 11), it was identified that the Random Forest algorithm emerged as the most superior, consistently achieving a classification accuracy rate of 100% in every scenario. Support Vector Machines (SVM) followed with an average accuracy of 99.817%, while Gradient Boosting demonstrated the relatively weakest with an average accuracy of 99%. Nevertheless, when viewed from another perspective, even the least performing algorithm only exhibited an error rate of 1%. This suggests the efficacy of this research's machine learning training and validation methods. Analysing from the vantage point of the operational states of the Worm Reduction Gears, all algorithms classified state 1 and state 15 with complete or near-complete accuracy. The classification success rates for other states ranged from 98.083% to 98.667%, suggesting that the features of these states are not as distinctive as those of state 1 and state 15, rendering them more susceptible to misclassification.

4.6 Cross-machine state classification

Considering the crucial practical significance of this component, a comprehensive verification has been carried out around it. The mutual verification of the two machines has been carried out for the three algorithms and two data segmentation methods involved in this paper. The previous results show that the classification accuracy of the models trained by the two data-cutting methods is extremely close, this part mainly focuses on the availability and accuracy characteristics of the models trained by different algorithms and machines.

Table 12. Classification accuracy on Worm Reduction Gears 2 with model trained by Worm Reduction Gears 1

Input/output shaft-cut data	Algorithm	State 1	State 3	State 9	State 13	State 15	Total
Input shaft-cut data	Random Forest	97.949 %	0%	100%	5.128%	1.026%	40.821 %
	Gradient Boosting	8.718%	0%	96.923 %	5.128%	8.205%	23.795 %
	Support Vector Machines	12.821 %	5.641%	99.487 %	27.179 %	50.000 %	39.077 %
Output shaft-cut data	Random Forest	20.000 %	0%	98.750 %	21.250 %	61.250 %	40.250 %
	Gradient Boosting	0%	0%	100%	70.000 %	20.000 %	38.000 %
	Support Vector Machines	0%	0%	98.750 %	32.500 %	63.750 %	39.000 %

Table 13. Classification accuracy on Worm Reduction Gears 1 with model trained by Worm Reduction Gears 2

Input/output shaft-cut data	Algorithm	State 1	State 3	State 9	State 13	State 15	Total
Input shaft-cut data	Random Forest	0%	87.179 %	71.795 %	0%	53.846 %	42.564 %
	Gradient Boosting	0%	67.179 %	57.436 %	0%	97.949 %	44.513 %
	Support Vector Machines	0%	2.564%	90.256 %	0%	50.769 %	28.718 %
Output shaft-cut data	Random Forest	0%	12.500 %	98.750 %	0%	97.500 %	41.750 %
	Gradient Boosting	20.000 %	0%	86.250 %	1.250%	2.500%	22.000 %
	Support Vector Machines	0%	0%	100%	0%	92.500 %	38.500 %

The above tables (Table 12, Table 13) shows that the results align with the previous expectations. Due to the limited training data and the different working environments of

Worm Reduction Gears, the model can only achieve a limited success rate in classifying the operating status. Nevertheless, the results are still very informative. It can be seen from the table that State 9 is the easiest to be correctly classified. Under the influence of many unfavourable factors, it can still guarantee an accuracy rate of more than 90% in most scenarios and is entirely correct under specific scenarios.

Table 14. Average classification accuracy of all cross-machine validation groups

Algorithm	State 1	State 3	State 9	State 13	State 15	Total
Random Forest	35.983%	31.143%	90.182%	5.251%	44.749%	41.462%
Gradient Boosting	6.239%	22.393%	82.495%	13.584%	39.135%	32.769%
Support Vector Machines	4.274%	2.735%	96.373%	14.476%	59.631%	35.515%
Sum	15.499%	18.757%	89.683%	11.104%	47.838%	36.582%

Despite the modest success rate, the validation results offer substantial analytical value. State 9 emerged as the state most amenable to accurate classification, with an average accuracy rate of 89.683% across the three algorithms, peaking at 96.373% when using the Support Vector Machines (SVM). The Random Forest algorithm, when trained using the Worm Reduction Gears 2 data along with the input shaft-cut data and subsequently classifying Worm Reduction Gears 1, achieved an accuracy rate of 87.179% for State 3. When trained with Worm Reduction Gears 1 data and the output shaft-cut data for classifying Worm Reduction Gears 2, the Gradient Boosting algorithm managed an accuracy rate of 70% for State 13. Using the Gradient Boosting algorithm with Worm Reduction Gears 2 data and the input shaft-cut data to classify Worm Reduction Gears 1, State 15 exhibited an accuracy rate of 97.949%. Furthermore, when utilising the Random Forest and SVM algorithms, trained with Worm Reduction Gears 2 data and the output shaft-cut data for classifying Worm Reduction Gears 1 at State 15, respectively achieved accuracy rates of 97.500% and 92.500%. It's evident that for cross-machine state classification, different algorithms demonstrate markedly distinct performances under varying scenarios.

5 Conclusion

In applying machine learning to rotating machinery, it is essential to acquire accurate, comprehensive, and sufficiently voluminous data, as these factors serve as the most critical prerequisites for training a machine learning model. Employing three acceleration sensors simultaneously at distinct locations on the machine housing and capturing data at high frequencies provides a significant assurance of the usability of the original data. This strategy ensures the comprehensiveness of the data, the extreme test conducted in the previous chapter proves that the data is of well availability.

Similarly, selecting features from the original data is important, the extracted features are used as training set. In this research, mean, standard deviation, skewness, and kurtosis were chosen as the features to be extracted from the data. As it shows in the graph of previous chapter, the features selection is valid and effective.

For verifying the approach validation for the research, a comparative analysis of the outcomes from methods was conducted to determine the optimal approach for further research items. The results demonstrated that when utilising 5-fold cross-validation an astonishing accuracy rate of 100% was achieved. While for the Holdout Validation, the classifications accuracy is a bit lower. As a result, 5-fold cross-validation was selected as the validation method for subsequent tasks.

Sample partitioning methods and training algorithms are of utmost significance among the various factors influencing classification performance. Consequently, extensive validations and comparisons were undertaken concerning these two aspects. Experiments centred on sample partitioning revealed a negligible impact on results between the input shaft-cut data and output shaft-cut data partitioning methods. Given the negligible discrepancy between the results of the two methods, subsequent research items employ both partitioning methods for validation.

The noticeable feature from the algorithms comparation is that Random Forest is the best performing algorithm in this specific scenario, with data of any Worm Reduction Gears, and regardless of input shaft-cut data or output shaft-cut data, it can guarantee that it is wholly classified correctly. From the average results, the Support Vector Machines algorithm is

inferior to the Random Forest algorithm, and the accuracy of the Gradient Boosting algorithm is generally the worst.

The works of the research and conclusion can be summarised as follows:

- Source data preprocessing
  The availability of quality source data is vital; efficacious training data can substantially bolster classification capabilities.
- Features extraction
  The choice of feature extraction is also critically salient. An apt selection of features should neither underfit nor overfit the model.
- Sample partitioning
  Regarding rotating machinery, the subdivided samples should not be overly diminutive. Small training samples can inflate the probability of misclassification. Partitioning samples based on the output shaft is an optimal strategy for Worm Reduction Gears.
- Machine learning algorithms comparison
  Among the triad of tested algorithms, Random Forest generally outperforms Gradient Boosting and Support Vector Machines regarding classification proficiency. However, in certain specific contexts, other algorithms might exhibit superior performance.
- Cross-machine state classification
  Primary methodologies are not enough to achieve acceptable classification accuracy. More granular research could yield additional helpful insights. Combining multiple models for various classification scenarios can help improve accuracy. Applying deep learning, neural networks, or other advanced techniques could improve classification prowess.

References

Abdeljaber, O. et al. (2017) ‘Real-time vibration-based structural damage detection using one-dimensional convolutional neural networks’, Journal of Sound and Vibration, 388, pp. 154–170. doi:10.1016/j.jsv.2016.10.043.

Advantages and disadvantages of worm gear reducers (2022) Singapore power transmission manufacturer pte ltd. Available at: https://gear-reducers.com/advantages-and-disadvantages-of-worm-gear-reducers/ (Accessed: 25 August 2023).

Chen, J. et al. (2020) ‘Fault diagnosis of rotating machinery: A review and Bibliometric analysis’, IEEE Access, 8, pp. 224985–225003. doi:10.1109/access.2020.3043743.

Huang, D. et al. (2021) ‘Fault diagnosis of high-speed train bogie based on the improved-CEEMDAN and 1-D CNN algorithms’, IEEE Transactions on Instrumentation and Measurement, 70, pp. 1–11. doi:10.1109/tim.2020.3047922.

Jardine, A.K.S., Lin, D. and Banjevic, D. (2006) ‘A review on machinery diagnostics and prognostics implementing condition-based maintenance’, Mechanical Systems and Signal Processing, 20(7), pp. 1483–1510. doi:10.1016/j.ymssp.2005.09.012

Louppe, G. (2015) Understanding Random Forests: From theory to practice, arXiv.org. Available at: https://arxiv.org/abs/1407.7502 (Accessed: 21 August 2023).

Manna, S. (2020) K-fold Cross Validation for deep learning using keras, Medium. Available at: https://medium.com/the-owl/k-fold-cross-validation-in-keras-3ec4a3a00538 (Accessed: 21 August 2023).

Nik (2023) Introduction to Random Forests in Scikit-Learn (sklearn) • datagy, datagy. Available at: https://datagy.io/sklearn-random-forests/ (Accessed: 20 August 2023).

Shen, C. et al. (2013) ‘Fault diagnosis of rotating machinery based on the statistical parameters of wavelet packet paving and a generic support vector regressive classifier’, Measurement, 46(4), pp. 1551–1564. doi:10.1016/j.measurement.2012.12.011.

Shi, Q. and Zhang, H. (2021) ‘Fault diagnosis of an autonomous vehicle with an improved SVM algorithm subject to unbalanced datasets’, IEEE Transactions on Industrial Electronics, 68(7), pp. 6248–6256. doi:10.1109/tie.2020.2994868.

Singh, V. et al. (2021) ‘Artificial Intelligence application in fault diagnostics of rotating industrial machines: A state-of-the-art review’, Journal of Intelligent Manufacturing, 34(3), pp. 931–960. doi:10.1007/s10845-021-01861-5.

Walker, R., Perinpanayagam, S. and Jennions, I.K. (2013) ‘Rotordynamic faults: Recent advances in diagnosis and prognosis’, International Journal of Rotating Machinery, 2013, pp. 1–12. doi:10.1155/2013/856865.

Wei et al. (2019) ‘A review of early fault diagnosis approaches and their applications in rotating machinery’, Entropy, 21(4), p. 409. doi:10.3390/e21040409.

Wen, L., Gao, L. and Li, X. (2019) ‘A new deep transfer learning based on sparse auto-encoder for fault diagnosis’, IEEE Transactions on Systems, Man, and Cybernetics: Systems, 49(1), pp. 136–144. doi:10.1109/tsmc.2017.2754287.

Yang, J. et al. (2019) ‘Tri-axial vibration information fusion model and its application to gear fault diagnosis in variable working conditions’, Measurement Science and Technology, 30(9), p. 095009. doi:10.1088/1361-6501/ab2296.

Zhao, H. et al. (2021) ‘Multi-branch convolutional neural networks with integrated cross-entropy for fault diagnosis in diesel engines’, Measurement Science and Technology, 32(4), p. 045103. doi:10.1088/1361-6501/abcefb.

Zhao, M. et al. (2021) ‘Deep residual networks with adaptively parametric rectifier linear units for fault diagnosis’, IEEE Transactions on Industrial Electronics, 68(3), pp. 2587–2597. doi:10.1109/tie.2020.2972458.

Zhang, S. et al. (2022) ‘State of the art on vibration signal processing towards data-driven gear fault diagnosis’, IET Collaborative Intelligent Manufacturing, 4(4), pp. 249–266. doi:10.1049/cim2.12064.

Appendix 1. Research environment and libraries

Python libraries (bash command: pip list):

Package	Version
matplotlib	3.7.1
numpy	1.25.0
openpyxl	3.1.2
pandas	2.0.2
scikit-learn	1.2.2

AccuracyCalculator (HTML label: <script src="https://cdnjs.cloudflare.com/ajax/libs/PapaParse/5.3.0/papaparse.min.js"></script>):

PapaParse@5.3.0

Release information (bash command: lsb_release -a):

No LSB modules are available.

Distributor ID: Ubuntu

Description: Ubuntu 22.04.3 LTS

Release: 22.04

Codename: jammy

System information (bash command: uname -a):

Linux ****** 6.2.0-26-generic #26~22.04.1-Ubuntu SMP PREEMPT_DYNAMIC ****** x86_64 x86_64 x86_64 GNU/Linux