Journal of Applied Informatics and Computing (JAIC)
Vol.8, No.2, December 2024, pp. 248~256
e-ISSN: 2548-6861

Bagging Nearest Neighbor and its Enhancement for Machinery Predictive Maintenance

Muhammad Irfan Arisani [1], Muljono [2]*
Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Dian Nuswantoro
111202012634@mhs.dinus.ac.id [1], muljono@dsn.dinus.ac.id [2]*

Article Info

Article history:

Received 2024-08-02
Revised 2024-08-12
Accepted 2024-08-13

Keyword:

Bagging Method,
Binary Classification,
Machine Learning,
Nearest Neighbor.

ABSTRACT

K-nearest Neighbor is a simple algorithm in Machine learning for such a prediction classification task which plays in valuable aspects of understanding big data. However, this algorithm sometimes does a lacking job of classification tasks for many different dataset characteristics. Therefore, this study will adopt enhancement methods to create a better performance of the nearest-neighbor model. Thus, this study focused on nearest neighbor enhancement to do a binary classification task from the extremely unbalanced dataset of a machine failure problem. Firstly, this study will create new features from the machinery dataset through the feature engineering processes and transform the chosen numerical features with standardization steps as the proper scaling. Then, the modified under-sampling method will be given which will reduce the amount of the majority class to 4.75 times that of the minority class. Next is the applied grid-search tuning which will find the right parameter combinations for the nearest-neighbor model being applied. Furthermore, the previous pre-processing steps will be combined with an additional bagging method. Finally, the resulting bagged KNN will present a 0.971 rate of accuracy, 0.555 rate of precision, 0.781 rate of recall, 0.649 rate of f1-score, 0.95 auc of ROC curve, and 0.702 auc of precision-recall curve.

I. INTRODUCTION

Machine learning is a field of study that receives insights from data using various computational calculations [1] and turns it into models that can be used for future predictions [2]. In recent decades, its applications have proliferated in many different fields including medical diagnosis [3], [4], industrial areas [5], [6], etc. Moreover, the algorithms are also developing from traditional statistics into advanced deep learning [7]. Those fields will help several parties solve particular problems and make intelligent decisions to achieve a better future prediction, specifically a regression or classification task.

K-nearest Neighbor, or KNN, is known for its simplicity and ease of interpretability, making it one of the most powerful machine learning algorithms [8]. Recent research shows that the KNN algorithm performs well on classification tasks [9], [10]. Despite its advantages for classifying, performing the ideal parameter combinations could be another challenging process [11]. KNN will predict the testing data with the number of *k* representing the number of nearest neighbors which could be bothersome to find its finest number. Also, selecting various distance formulas will affect the model performance. On the general problems of machine learning, unbalanced data could also potentially affect the model performance [12]. It refers to the small number of minority classes that are difficult to identify by the KNN or other models.

While the researchers struggle to optimize the KNN algorithms, there is also an advanced ensemble method called bagging. This method is commonly used in the Random Forest algorithm, combining multiple tree models from the random sampling data train. With its first appearance in 2001 [13] and its success in solving many practical problems [14], [15], Random Forest has become widely used in modern machine learning that adopts the bagging method [16]. Researchers also know that the Random Forest can effectively handle unbalanced data. However, the concept of the bagging method is rarely used in other classical model algorithms.

This study aims to offer an alternative approach to utilizing KNN algorithms to improve model performance, especially for highly unbalanced datasets, using the bagging method. This study starts with applying standardization, adjusting the finest parameters with grid-search tuning, and adopting the under-sampling method. Lastly, the earlier applied method also combined with the bagging is presented. This study compares, evaluates, and analyzes the model performance results of the KNN enhancement by calculating the accuracy, precision, recall, f1-score, ROC, and precision-recall curve.

II. RELATED WORKS

The study of machinery predictive maintenance was initiated first by Matzka S. in his proceedings in 2020 [6] using the machinery dataset. With the bagged trees ensemble classifier known as the Random Forest model, the proceedings implicitly state that the Model has 0.983 accuracy and 0.781 f1-score by calculating its confusion matrix result. Lastly, Sharma N. et al also published their article with the KNN model and the same dataset implicitly resulting in 0.978 accuracy and 0.563 f1-score [17].

On the other hand, the bagging method had been inspired to be implemented into a specific scenario. One of the well-known studies conducted by Chen is the implementation of the bagging method in the Lasso algorithm for six different datasets [18]. In the other study conducted by Luthfi, a bagging method was applied in the KNN algorithm with various pre-processing steps. It is supposed to handle the multilabel classification on the corn dataset, then resulting in 0.793 accuracy and 0.819 f1-score [19]. Lastly, in the most recent study done by Arisani et al., the bagged KNN was also applied to the machinery dataset to predict the type of machine failure with quite good and competitive results [20].

For a specific topic like this study, the previous research described how well the performance of bagging on the Random Forest model compared to the common KNN model. Therefore, this study will combine the ensemble learning method with the simple machine learning algorithm by merging the bagging method on the KNN model with its pre-processing steps. The bagging method will play an important role in this study in gaining a better performance prediction on the common KNN method.

III. RESEARCH METHOD

The study begins by importing the machinery dataset illustrated in Figure 1. Furthermore, the dataset will enter the feature engineering steps described in the previous study [6]. As a note, this study will compare three different KNN models divided by several used methods. For instance, the third bagged KNN model will also contain the second previous pre-processing methods. Then, all KNN models will have the same ratios of data splitting. Finally, all of the KNN models will be compared and evaluated by calculating the accuracy, precision, recall, f1-score, ROC, and precision-recall curve.

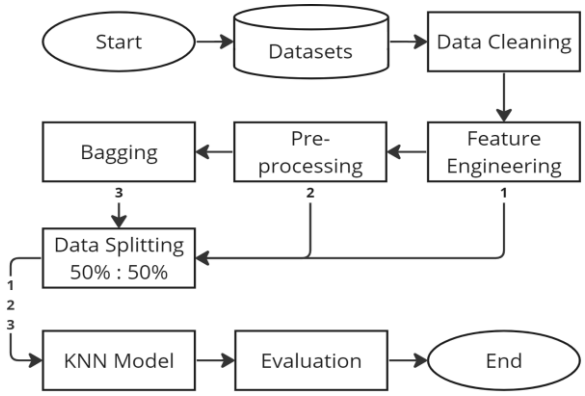

Figure 1 In the research workflow, the number represents the current used with/without the previous method. Starting with number 1 as the base model without any pre-processing steps and ending with number 3 as the combined methods.

A. Datasets

Matzka S. in his proceeding made an experiment with a bunch of original machinery data called the AI4I 2020 Predictive Maintenance Dataset [6]. It was first introduced at an international conference called 2020 Third AI for Industries [21]. The machinery dataset is open-source and can be accessed via Kaggle or UCI Machine Learning Repository.

It's interesting to note that the machinery dataset contains three different classification tasks based on the amount of containing labels. This could be an interesting topic for researchers to find out which machine learning algorithms are optimal and suitable for each problem category. It is important to remember that this study will be focusing on binary classification which will predict whether a machine is failing or not using the finest bagged KNN model.

Originally, there were a total of 14 features consisting of 10000 rows of data, two different labels referring to fail or not-fail machines, and five different labels as the failure types. More details about the machinery dataset show that there are 9661 rows of not-fail machines and 369 rows of failed ones which leads to extremely unbalanced data. Last but not least, all of them were free from data duplicates, missing values, and inconsistent data which made it easier for this experiment.

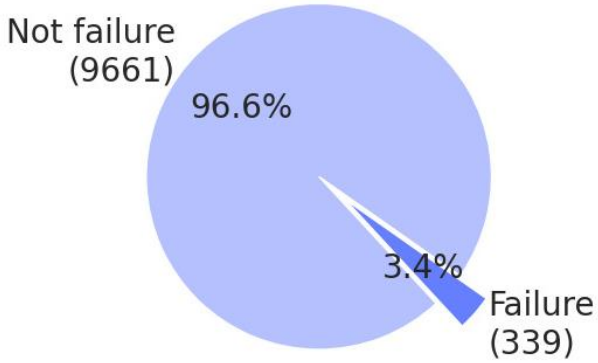

Figure 2 The extremely unbalanced machinery dataset label. As a note 'not failure' machine will be represented as zero, while 'failure' as one.

This study also generated three new engineered features based on the previous study which require general mathematic formulas. The detailed information on original, engineered, and used features is explained in Table I below. Since the used features result in better performance and this study focuses on the binary classification task, some features may not be used in the further process.

TABLE I
AI4I 2020 PREDICTIVE MAINTENANCE DATASET

Features	Description
UDI	Unique ID.
Product ID	Machine ID.
Type	Machine type, L = low, M = medium, H = high.
Air temperature [K]	Temperature outside the machine, in Kelvin.
Process temperature [K]	Temperature inside the machine, in Kelvin.
Rotational speed [rpm] *	Rotational speed, in rotation per minute.
Torque [Nm]	Torque, in Nanometer.
Tool wear [min]	Tool wear, in minutes.
Machine failure	Machine label, 0 = not fail, 1 = fail.
TWF	Failing in tool wear part.
HDF	Failing in the cooling process causes heat dissipation.
PWF	Failing in abnormal machine power.
OSF	Failing due to overstrain.
RNF	Failing in random (unknown) reason.
Temperature difference [K] *, **	Temperature differences between air and process, in Kelvin.
Power [W] *, **	The product of torque and rotational speed, in Watt.
Strain [minNm] *, **	The product of torque and tool wear, in minute Nanometer.
*, all of the used features for the KNN models. **, engineered features.	

B. Pre-processing

Before doing the ensemble learning method, it is important to ensure that the machinery dataset is suitable for the further process. Therefore, this study will adopt three different pre-processing steps sequentially before continuing to the modeling.

B.1. Feature Engineering

This study has also carried out a series of feature engineering to obtain unique features from the results of physics equation calculations. There are three independent features such as temperature difference, power, and strain that have been engineered referring to previous research which is presented in Equations 1-3 respectively. With a bunch of trial-and-error experiments, this study has chosen three independent engineers and one original feature as the best selection for the KNN model. Future research is needed to ensure how effective the feature combination and feature engineering are in predicting machine failure problems specific to binary classification tasks.

$$Td = |At - Pt| \quad (1)$$

$$Pw = Rs * T * 0.104 \quad (2)$$

$$S = T * Tw \quad (3)$$

B.2. Standardization

On the machinery dataset, it is recognized that every feature has a different data distribution or ratio. The different scaling ratios of each feature could lead to bad predictions for the KNN model because of the improper distance data point. Therefore, it is important to rescale all of the features by using the scaler method. In this study, one way to apply that method is by using standardization or data rescaling using z-normalization. The standardization formula is given as described in Equation 4, where x_i is the current data points, $\bar{x}$ is the mean of the current dependent feature, σ is the standard deviation, and x'_i is the result of scaled data [22].

$$x'_i = \frac{(x_i - \bar{x})}{\sigma} \quad (4)$$

B.3. Modified Under-sampling

In the previous explanation, we realized that we're facing extremely unbalanced data in the first place which will affect the KNN model performance if we don't handle it correctly. One way to handle unbalanced data which also be used in this study is by using the under-sampling method. This method will randomly choose the data points from the majority class just as much as the minority class has [23]. The chosen sampling method is considered in this study since it doesn't require a complex mathematical formula to do so, such as over-sampling using SMOTE.

The pure random under-sampling method will ensure the same amount of each class to be used for the modeling. However, since the remaining majority of data points will be unused, this study will modify the under-sampling step so the amount of the majority class will be about 4.75 times more than the minority class. This is considered since the pure under-sampling method may eliminate some insightful information from the majority class and the modified technique will exclude that disadvantage.

B.4. Grid-search Tuning

KNN will predict the data testing by calculating its distance with the nearest k neighbor of each data train. Sometimes, the distance formula or the number of k was not in the optimal measure resulting in a bad model prediction. Therefore, the hyperparameter tuning steps with the grid-search method are carried out in this study. This method will find all possible parameter combinations and select its best based on the better model performance. Since the KNN is a simple algorithm, then the parameters to be tuned are only the distance formula and the number of k, at least for this study. Some similar studies that adopted this method resulted in well-performed models on classification tasks [24], [25].

B.5. Data Splitting

Before splitting the data is done, it's important to note that the machinery dataset originally contained extremely unbalanced data as described in Figure 2. The inappropriate data splits without concerning the data labels could lead to biased model performance. Therefore, the data-splitting process will adopt a stratified random sampling to ensure the same amount of each split [26]. This study uses the 50:50 split ratio for the data training and testing and combines it with stratified random sampling.

C. Modelling

This study will explain the algorithms for the modeling steps which were initiated by the pure KNN model and then combined with the various pre-processing steps and bagging method.

C.1. K-nearest Neighbor

KNN algorithm works by calculating the distance of data points and data train, so it's going to need a distance formula. After performing a grid-search tuning step, this study shows that Euclidean distance is one of the best parameters for the machinery dataset. Furthermore, how the KNN algorithms work is presented in Figure 3.

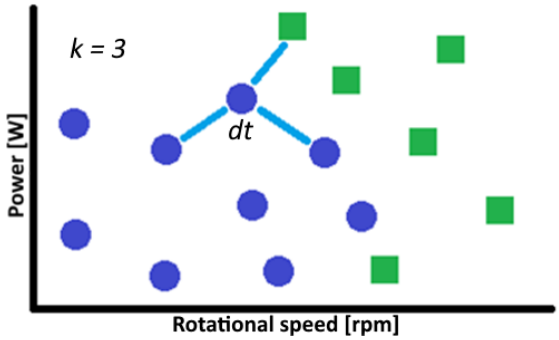

Figure 3 How the KNN algorithm in the two-dimensional vectors, with k as the number of neighbors and dt as the testing data point.

The interpretation of Euclidean distance in 2-dimensional vectors is easy to explain which can be found in Pythagoras formula [27], [28]. However, this study used four features and the Euclidean distance could be more complicated. So instead, the Euclidean formula will be presented in the following Equation 5.

$$d = \sqrt{\sum_{i=1}^{4} (fn_{te} - fn_{tr})^2} \quad (5)$$

C.2. Bagging Method

The idea behind the bagging can be found in the Random Forest algorithm which was already mentioned in the previous studies [6], [17]. It divided the training sample into multiple sub-samplings and created a Decision Tree for each of them. Then, every new prediction on the data testing will be based on the voting result of each tree. Another previous study was also inspired by this mechanism and noted that it uses the KNN as the main algorithm [19].

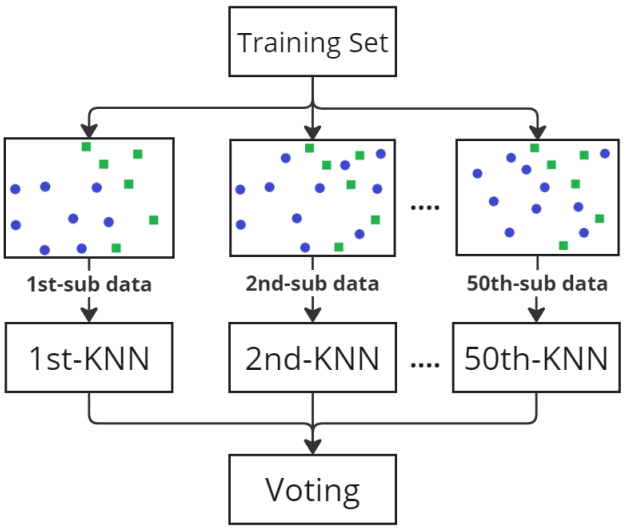

Figure 4 The bagging method on the KNN algorithm illustration.

A similar method will be applied to the KNN algorithm and this study will determine its performance result and whether it has good or bad prediction. To give a better understanding of how the bagged KNN works, this study also included the illustration in Figure 4. Note that the number of subsampling in this study is equal to 50 which means there are 50 different types of KNN with their own rules. Each subsampling that results in each KNN is created by the random sampling mechanism in this experiment.

D. Model Evaluation

To determine whether the model enhancement produces a good result, this study will include various evaluation techniques and perform a comparison between each KNN model. Starting with showing each model's predictions by presenting each confusion matrix. The different types of confusion matrices will be used to perform further calculations such as accuracy, precision, recall, f1-score, ROC, and precision-recall curve. The first four model evaluations will be described in Equations 6 and 9, while the last two will be presented in the results section.

$$accuracy = \frac{TN + TP}{TN + FN + TP + FP} \quad (6)$$

$$precision = \frac{TP}{TP + FP} \quad (7)$$

$$recall = \frac{TP}{TP + FN} \quad (8)$$

$$f1\ score = \frac{2 * precision * recall}{precision + recall} \quad (9)$$

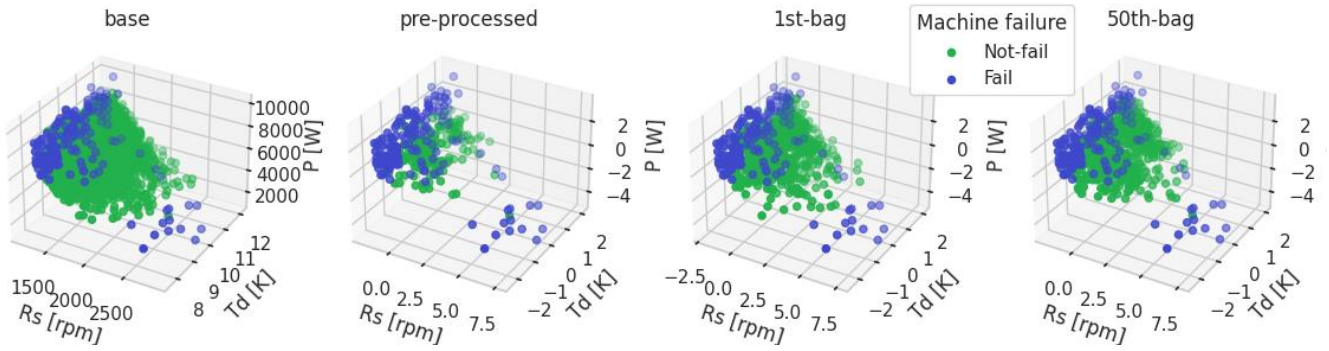

Figure 5 The visualization comparison in the three-dimensional vector described how the training data will rule and impact the KNN algorithm's performance.

IV. RESULT AND DISCUSSION

A. Implementation

As explained in the research method section, this study will experiment with the bagged KNN technique and compare it with the common KNN algorithms on the machinery dataset. All KNN algorithms will differ in the parameter configurations and the pre-processing steps resulting in different used datasets. For instance, the first pure KNN model will use the machinery dataset with a feature engineering step, while the second pre-processed KNN and the third bagged KNN will use the machinery dataset with an extra pre-processing step. We have also mentioned the used features for the modeling step since they've shown good results for the prediction performance.

At the beginning of the experiment, we started from the base KNN model with additional engineered features and it will purely understand the given training data. The final selected features will be based on the trial-and-error experiment, Then, the pre-processed KNN will adopt various classic pre-processing steps such as standardization and under-sampling resulting in different machinery dataset characteristics. Furthermore, the previous pre-processed parameters and dataset will be used again in the bagged KNN with the additional bagging method applied and modified under-sampling. Finally, Figure 5 visualizes the final data training differences for each model in three of the four features used.

B. Results

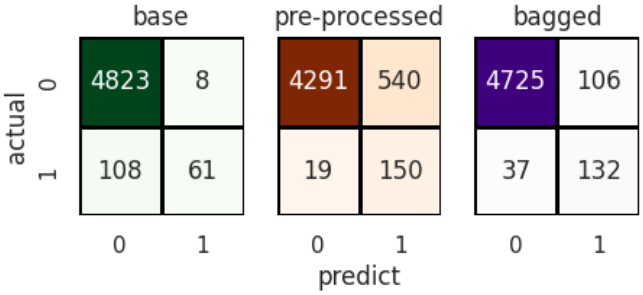

Figure 6 The resulting confusion matrices.

After several KNN models had been trained with the particular steps, it then predicted the testing data and produced various predictions. In general, any kind of machine learning model sometimes did a miss-prediction and that's normal. So, our job in solving the binary classification task will be to try to reduce the number of miss-prediction on the KNN models. Therefore, the representation of the KNN model prediction is given as a confusion matrix result in Figure 6. The confusion matrix results will be the initial calculation for further model evaluations. The resulting false and true predictions will be evaluated by several selected metrics which are accuracy, precision, recall, f1-score, ROC, and precision-recall curve sequentially.

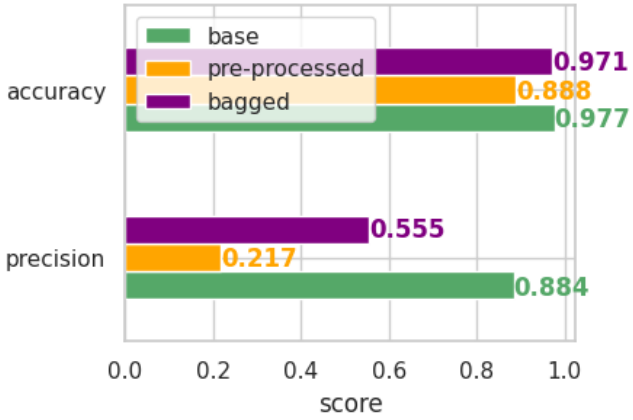

Figure 7 The accuracy and precision comparisons for each KNN model.

The resulting confusion matrices will have a basic role in performing further calculations, and this study will include four model evaluations. The first one is accuracy, which can be identified as how well the KNN models predict the testing data. Starting with the base KNN with 0.977, then the pre-processed KNN with 0.888, and the bagged KNN with 0.971. These accuracy scores indicate that the base KNN and bagged KNN perform similarly well, while the pre-processed KNN shows a drop in accuracy, suggesting that the pre-processing steps may need further refinement. The second one is precision, which can tell us the quality of KNN models to predict the machinery testing data, whether it's negative or positive label. The base KNN received 0.884, the pre-processed KNN received 0.217, and the bagged KNN received 0.555. The low precision of the pre-processed KNN suggests it struggles with identifying overall instances,

whereas the bagged and pre-processed KNN models show the least and superior improvements respectively. It then concludes that the bagged KNN is neither better nor worse in this metric. Finally, each accuracy and precision are visualized in Figure 7, providing a clear comparison of the models' performance.

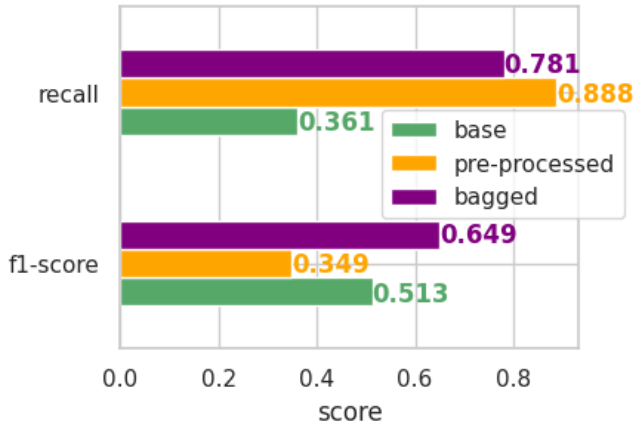

Figure 8 The recall and f1-score comparison for each KNN model.

The next two model evaluations in this study will be analyzing recall and f1-score matrices. The use of recall metric will tell us how well the KNN models predict the machinery testing data specifically for the positive label, while the f1-score matric will measure the balance of the importance of precision and recall. Starting with the base KNN model that achieves a recall of 0.361 and an f1-score of 0.513, indicating a moderate ability to balance precision and recall while struggling to identify a high proportion of positive instances. The pre-processed KNN, however, shows a reversal in performance trends with a high recall of 0.888 but a lower f1-score of 0.349, suggesting it identifies the most positive cases but lacks precision. Conversely, the bagged KNN model demonstrates the best overall predictions with a 0.781 score of recall and the highest f1-score of 0.649, indicating a superior balance between detecting positive instances and maintaining precision. This analysis suggests that while preprocessing greatly increases the ability to detect positive instances, bagged KNN provides the most balanced approach, effectively improving both the identification of positive cases and precision in predictions.

The fifth model evaluation in this study is calculating the receiver operating characteristic (ROC) curve. When there are true positives and false positives in the confusion matrix, the ROC curve will calculate the relationships between them. Unlike the accuracy metric in binary classification, the ROC curve specifically measures the model's ability to discriminate between two classes, in this case, failure and non-failure of machinery [29]. The ROC curve provides a visual representation that helps in selecting an optimal decision threshold. Figure 8 (left) shows that the bagged KNN had a larger area under the curve (AUC) and was approximately closer to the top-left corner compared to the other KNN techniques, indicating superior performance. This suggests that the bagged KNN is more effective in distinguishing between yes or no in predicting machine failure, making it a better choice for this classification task. The higher AUC value reflects its ability to maintain a higher true positive rate while minimizing the false positive rate across different threshold settings.

The final model evaluation in this study involves the precision-recall curve. Unlike the ROC curve, the precision-recall curve will calculate the trade-off within the precision and recall (true positive rate) and its relations. This is particularly important for the machinery dataset, which is unbalanced [30]. In unbalanced datasets, identifying the minority class will be much easier to calculate with the precision-recall curve as it directly accesses the model's performance. A good classification model will have a precision-recall graph closer to the top-right corner, indicating high precision (fewer false positives) and high recall (fewer false negatives). Figure 8 (right) shows that the bagged KNN has a superior precision-recall curve compared

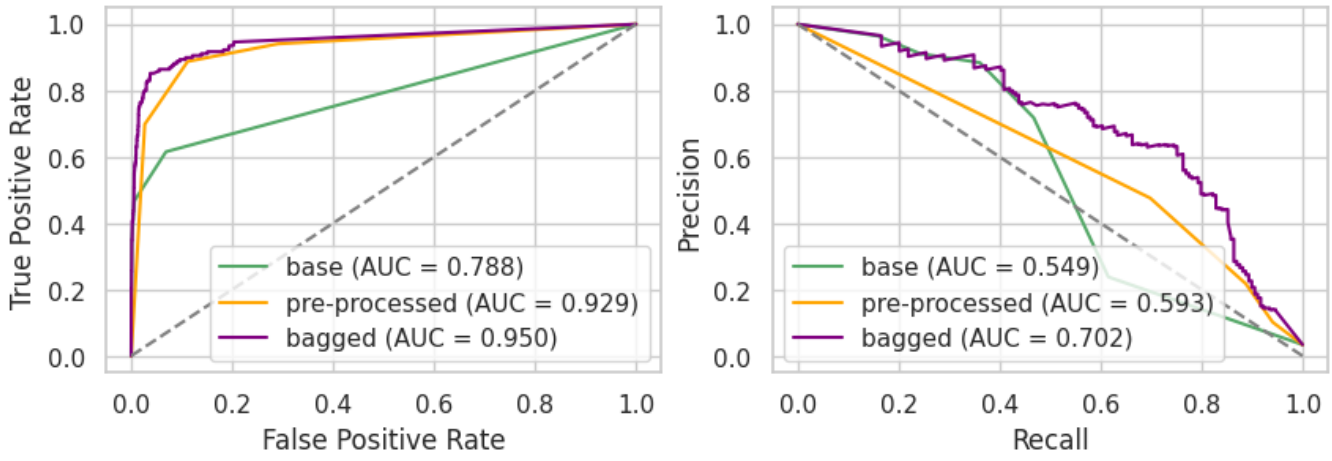

Figure 9 The ROC (left) and precision-recall (right) comparisons for each KNN model.

TABLE II
MATRICES COMPARISON FOR EACH KNN MODEL WITH HIGH PRECISION VALUES

		Accuracy*	Precision*	Recall*	F1-score*	ROC Curve**	Precision-Recall Curve**	Average
Base	**Results**	0.977	0.884	0.361	0.513	0.788	0.549	0.679
Pre-processed	**Results**	0.888	0.217	0.888	0.349	0.929	0.593	0.644
	Vs. Base	- 0.089	- 0.667	+ 0.527	- 0.164	+ 0.141	+ 0.044	- 0.035
Bagged	**Results**	0.971	0.555	0.871	0.649	0.950	0.702	0.783
	Vs. Base	**- 0.006**	**- 0.329**	+ 0.51	+ 0.136	+ 0.162	+ 0.153	+ 0.104
	Vs. Pre-processed	+ 0.083	+ 0.338	**- 0.017**	+ 0.3	+ 0.021	+ 0.109	+ 0.139

*, metrics rate in score
**, metrics rate in auc

to the other KNN models. This suggests that the bagged KNN not only maintains a high true positive rate but also effectively reduces false positives, making it a robust choice for handling imbalanced data. The improved performance of the bagged KNN highlights its ability to balance precision and recall, ensuring more reliable predictions in identifying machinery failures.

The bagged KNN shows interesting results on the machinery dataset compared to the common KNN techniques. In Table II above, the performance comparison between three different approaches of the KNN model is given by presenting various evaluation metrics. The base KNN model shows high accuracy (0.977) and precision (0.884), but it struggles with recall (0.361) and f1-score (0.513), indicating it tends to miss many true positives. Then, the pre-processed KNN model improves recall significantly (+0.527) to 0.888, but this comes at the expense of precision (-0.667), resulting in a lower f1-score (0.349). On the other hand, the bagged KNN model provides a more balanced improvement, enhancing recall (+0.51) and slightly reducing precision compared to the base KNN. Still, it substantially boosts the f1-score to 0.649 and overall performance metrics like the ROC and Precision-Recall Curves. This indicates that while pre-processing can drastically improve recall, bagging delivers a better balance across all metrics, making it a more robust choice for scenarios where both precision and recall are critical.

TABLE III
THE ACCURACY AND F1-SCORE COMPARISONS FOR VARIOUS STUDIES

Model	Accuracy	F1-score
Matzka S.'s Random Forest	0.983	0.781
Sharma N. et al's KNN	0.978	0.563
Pre-processed KNN	0.888	0.349
Bagged KNN	0.971	0.649

The bagged KNN also shows the competitive performance with previous studies, as shown in Table III. The combined techniques with additional bagging indicate that this modified KNN excels in predicting machinery failures, although it still falls short of Matzka S.'s Random Forest model which loses 0.012 accuracy and 0.132 f1-score. Moreover, the bagged KNN gains in accuracy and f1-score compared to the pre-processed KNN, but loses 0.007 accuracy and gains 0.086 f1-score compared to Sharma N. et al.'s KNN. This suggests that while the bagged KNN improves the f1-score significantly over other traditional KNN approaches, there is still room for improvement. The bagging approach enhances the model's ability to identify failures more accurately by reducing variance and improving robustness, but optimizing further could bridge the gap with more advanced models like Random Forest.

C. Discussions

After a series of experiments on the KNN algorithm, several advantages of the bagged KNN model have emerged. First, as illustrated in Figures 7-8, the evaluation of key metrics reveals that the bagged KNN model balances the pure and the pre-processed KNN model. Specifically, three out of the six evaluation metrics demonstrate improved performance with the bagged KNN. Second, implementing the bagging method within the KNN algorithm leads to more stable predictions compared to a single KNN model. This increased stability is crucial, as it enhances the model's reliability and robustness. In the context of big data, stable and reliable predictions are essential for gaining deeper insights and making informed decisions. Therefore, the bagged KNN model not only provides better performance metrics but also ensures greater consistency in its predictions, making it a valuable tool for future analysis.

This research is also not free from shortcomings. First, the recall evaluation metric cannot compete with pre-processed KNN or Random Forest models in previous studies. This can be seen from the experimental results presented in Tables II and III where there are performance degradations in bagged KNN and it still didn't surpass the Random Forest performance. Second, calculating the distance in the bagged KNN algorithm requires quite a lot of computational resources which increases the time and memory to execute the model. Therefore, applying the bagging method to the KNN algorithm will make the computing process resource-hungry. Although these things are not the main focus of this study, they can be of some concern as notes for improvement for future studies. For instance, it is necessary to build an ensemble learning method that performs as best as it can, but

also with a workload as low as it does. Thus, the implementation of the bagging method needs to be developed further.

In conclusion, the study of the KNN model across different methods—base, pre-processed, and bagged—reveals distinct trade-offs in performance. Base KNN excels in precision and accuracy but suffers from low recall, leading to a moderate f1-score. The pre-processed KNN approach significantly boosts recall, which is crucial for identifying true positives, but at the cost of a substantial drop in precision, resulting in a lower overall f1-score. The bagged KNN model strikes the best balance which improves both precision and recall, and delivers the highest f1-score among the rest approaches. This makes bagging the most effective method for achieving a well-rounded performance, especially in situations where both precision and recall are critical.

V. CONCLUSIONS

In order to predict whether a machine failure will occur or not in the machinery dataset, this study shows that the bagged KNN model has quite good prediction performance. This research has also proven that the combination of bagging methods in the KNN algorithm can provide increased performance, but while maintaining its simplicity. However, this experiment also left notes for further improvement, such as regarding three of the evaluation matrices that were not good enough or related to increased computational workload.

Nevertheless, this study can be a guide for similar studies in the future. Furthermore, it is hoped that several future studies will experiment with the implementation of bagging methods in classical machine learning algorithms. This cannot be separated from scopes such as: finding the optimal number of sub-sections in bagging, testing similar methods with various types of datasets, implementing bagging methods in various types of algorithms, comparing appropriate algorithms in implementing bagging methods other than tree-based algorithms, and so on.

REFERENCES

[1] M. O. K. Mendonça, S. L. Netto, P. S. R. Diniz, and S. Theodoridis, "Machine learning: Review and Trends," *Signal Processing and Machine Learning Theory*, pp. 869–959, Jan. 2024, doi: 10.1016/B978-0-32-391772-8.00019-3.

[2] T. W. Edgar and D. O. Manz, "Machine Learning," *Research Methods for Cyber Security*, pp. 153–173, Jan. 2017, doi: 10.1016/B978-0-12-805349-2.00006-6.

[3] R. Detrano *et al.*, "International application of a new probability algorithm for the diagnosis of coronary artery disease," *Am J Cardiol*, vol. 64, no. 5, pp. 304–310, Aug. 1989, doi: 10.1016/0002-9149(89)90524-9.

[4] B. Abbasi and D. M. Goldenholz, "Machine learning applications in epilepsy," *Epilepsia*, vol. 60, no. 10, pp. 2037–2047, Oct. 2019, doi: 10.1111/EPI.16333.

[5] M. Bertolini, D. Mezzogori, M. Neroni, and F. Zammori, "Machine Learning for industrial applications: A comprehensive literature review," *Expert Syst Appl*, vol. 175, Aug. 2021, doi: 10.1016/J.ESWA.2021.114820.

[6] S. Matzka, "Explainable Artificial Intelligence for Predictive Maintenance Applications," *Proceedings - 2020 3rd International Conference on Artificial Intelligence for Industries, AI4I 2020*, pp. 69–74, Sep. 2020, doi: 10.1109/AI4I49448.2020.00023.

[7] N. Sharma, R. Sharma, and N. Jindal, "Machine Learning and Deep Learning Applications-A Vision," *Global Transitions Proceedings*, vol. 2, no. 1, pp. 24–28, Jun. 2021, doi: 10.1016/J.GLTP.2021.01.004.

[8] D. Lopez-Bernal, D. Balderas, P. Ponce, and A. Molina, "Education 4.0: Teaching the Basics of KNN, LDA and Simple Perceptron Algorithms for Binary Classification Problems," *Future Internet 2021, Vol. 13, Page 193*, vol. 13, no. 8, p. 193, Jul. 2021, doi: 10.3390/FI13080193.

[9] A. Enhanced Student *et al.*, "Enhanced Student Admission Procedures at Universities Using Data Mining and Machine Learning Techniques," *Applied Sciences 2024, Vol. 14, Page 1109*, vol. 14, no. 3, p. 1109, Jan. 2024, doi: 10.3390/APP14031109.

[10] G. Fischer *et al.*, "Classification of the Pathological Range of Motion in Low Back Pain Using Wearable Sensors and Machine Learning," *Sensors 2024, Vol. 24, Page 831*, vol. 24, no. 3, p. 831, Jan. 2024, doi: 10.3390/S24030831.

[11] N. Bhatia, "Survey of Nearest Neighbor Techniques," *IJCSIS) International Journal of Computer Science and Information Security*, vol. 8, no. 2, 2010, Accessed: Jan. 31, 2024. [Online]. Available: http://sites.google.com/site/ijcsis/

[12] V. S. Spelmen and R. Porkodi, "A Review on Handling Imbalanced Data," *Proceedings of the 2018 International Conference on Current Trends towards Converging Technologies, ICCTCT 2018*, Nov. 2018, doi: 10.1109/ICCTCT.2018.8551020.

[13] L. Breiman, "Random forests," *Mach Learn*, vol. 45, no. 1, pp. 5–32, Oct. 2001, doi: 10.1023/A:1010933404324/METRICS.

[14] K. Chan-Bagot *et al.*, "Integrating SAR, Optical, and Machine Learning for Enhanced Coastal Mangrove Monitoring in Guyana," *Remote Sensing 2024, Vol. 16, Page 542*, vol. 16, no. 3, p. 542, Jan. 2024, doi: 10.3390/RS16030542.

[15] D. Z. Syeda and M. N. Asghar, "Dynamic Malware Classification and API Categorisation of Windows Portable Executable Files Using Machine Learning," *Applied Sciences 2024, Vol. 14, Page 1015*, vol. 14, no. 3, p. 1015, Jan. 2024, doi: 10.3390/APP14031015.

[16] G. Biau and E. Scornet, "A Random Forest Guided Tour," *Test*, vol. 25, no. 2, pp. 197–227, Nov. 2015, doi: 10.1007/s11749-016-0481-7.

[17] N. Sharma, T. Sidana, S. Singhal, and S. Jindal, "Predictive Maintenance: Comparative Study of Machine Learning Algorithms for Fault Diagnosis," *SSRN Electronic Journal*, Jun. 2022, doi: 10.2139/SSRN.4143868.

[18] K. Chen and Y. Jin, "An ensemble learning algorithm based on Lasso selection," *Proceedings - 2010 IEEE International Conference on Intelligent Computing and Intelligent Systems, ICIS 2010*, vol. 1, pp. 617–620, 2010, doi: 10.1109/ICICISYS.2010.5658515.

[19] Moch. Lutfi, "Implementasi Metode K-Nearest Neighbor dan Bagging Untuk Klasifikasi Mutu Produksi Jagung," *agromix*, vol. 10, no. 2, 2019, doi: 10.35891/agx.v10i2.1636.

[20] M. I. Arisani and M. Muljono, "Peningkatan Kinerja K-Nearest Neighbor menggunakan Bagging pada Permasalahan Ragam Kelas terhadap Pemeliharaan Prediktif Permesinan," *JUSTIN (Jurnal Sistem dan Teknologi Informasi)*, vol. 12, no. 2, pp. 373–379, Apr. 2024, doi: 10.26418/JUSTIN.V12I2.78503.

[21] IEEE Computer Society and Institute of Electrical and Electronics Engineers., "2020 Third International Conference on Artificial Intelligence for Industries : AI4I 2020 : proceedings : virtual conference, 21-23 September 2020.," p. 83.

[22] L. B. V. de Amorim, G. D. C. Cavalcanti, and R. M. O. Cruz, "The Choice of ScalingTtechnique Matters for Classification Performance," *Appl Soft Comput*, vol. 133, Dec. 2022, doi: 10.1016/j.asoc.2022.109924.

[23] R. Mohammed, J. Rawashdeh, and M. Abdullah, "Machine Learning with Oversampling and Undersampling Techniques: Overview Study and Experimental Results," in *2020 11th International Conference on Information and Communication Systems, ICICS 2020*, Institute of Electrical and Electronics

Engineers Inc., Apr. 2020, pp. 243–248. doi: 10.1109/ICICS49469.2020.239556.

[24] D. M. Belete and M. D. Huchaiah, "Grid search in hyperparameter optimization of machine learning models for prediction of HIV/AIDS test results," *International Journal of Computers and Applications*, vol. 44, no. 9, pp. 875–886, 2022, doi: 10.1080/1206212X.2021.1974663.

[25] I. Muhamad and M. Matin, "A Hyperparameter Tuning Using GridsearchCV on Random Forest for Malware Detection," *MULTINETICS* , vol. 9, no. 1, pp. 43–50, May 2023, doi: 10.32722/MULTINETICS.V9I1.5578.

[26] S. Faiqotul Ulya, Y. Sukestiyarno, P. Hendikawati, and D. Juli, "Analisis Prediksi Quick Count dengan Metode Stratified Random Sampling dan Estimasi Confidence Interval Menggunakan Metode Maksimum Likelihood," *Unnes Journal of Mathematics*, vol. 7, no. 1, pp. 108–119, Nov. 2018, doi: 10.15294/UJM.V7I1.27385.

[27] Y. Miftahuddin, S. Umaroh, and F. R. Karim, "Perbandingan Metode Perhitungan Jarak Euclidean, Haversine, dan Manhattan dalam Penentuan Posisi Karyawan," *Jurnal Tekno Insentif*, vol. 14, no. 2, pp. 69–77, Aug. 2020, doi: 10.36787/jti.v14i2.270.

[28] D. N. S. Gonçalves, C. D. M. Gonçalves, T. F. De Assis, and M. A. Da Silva, "Analysis of the Difference between the Euclidean Distance and the Actual Road Distance in Brazil," *Transportation Research Procedia*, vol. 3, pp. 876–885, Jan. 2014, doi: 10.1016/J.TRPRO.2014.10.066.

[29] K. Hajian-Tilaki, "Receiver Operating Characteristic (ROC) Curve Analysis for Medical Diagnostic Test Evaluation," *Caspian J Intern Med*, vol. 4, no. 2, p. 627, 2013, Accessed: Feb. 29, 2024. [Online]. Available: /pmc/articles/PMC3755824/

[30] J. Davis and M. Goadrich, "The relationship between precision-recall and ROC curves," *ACM International Conference Proceeding Series*, vol. 148, pp. 233–240, 2006, doi: 10.1145/1143844.1143874.