International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Optimized Manufacturing and Construction Site Machines Maintenance Prediction using Machine Learning and Crayfish algorithm

Prathamesh Kshirsagar[1], Amit Dekate[2], Atharva Khardekar[3], Sushma VIspute[4]

[1,2,3,4]Department of Computer Engineering, Pimpri Chinchwad College of Engineering Pune

Abstract

Efficient maintenance prediction is critical for ensuring the operational continuity and longevity of industrial machinery. This paper presents a comparative analysis of diverse machine-learning algorithms for the task of machine maintenance prediction. Through rigorous experimentation and evaluation, we assess the performance of algorithms including AdaBoost, Random Forest, Gradient Boosting, and Support Vector Machines (SVM). Additionally, to enhance predictive accuracy, we integrate an optimizer algorithm, Cuckoo Search, into our framework. This optimization technique fine-tunes algorithm parameters, further improving accuracy. Our findings offer valuable insights into optimizing machine maintenance prediction, empowering industries with proactive maintenance strategies to mitigate downtime and enhance productivity.

Keywords: Machine learning models, random forest, Crey Fish, Optimizer, Maintenance.

Introduction

A Subsection Sample

Predictive maintenance has emerged as a critical strategy for industries looking to optimize their operations, minimize downtime, and reduce maintenance costs. By leveraging advanced data analytics and machine learning techniques, companies can predict when equipment failure is likely to occur, allowing for proactive maintenance interventions. The predictive maintenance software developed utilizes real-time data from four selected features derived from the machine's dataset. These features serve as indicators of the machine's health and performance. By continuously monitoring these features in real time, the software can assess the current state of the machine and predict whether maintenance is required. To ensure accurate predictions, various classification techniques were compared and evaluated to identify the most effective model. This involved analyzing the performance of different algorithms, such as neural networks, decision trees, SVM and random forests, among others. Through rigorous testing and validation, the highest-performing model was chosen for implementation in the predictive maintenance application. In addition to selecting the best classification technique, the efficiency of the model was further enhanced using the crayfish optimizer. This optimization technique helps fine-tune the model parameters to improve its predictive accuracy and overall performance. By leveraging the capabilities of the crayfish optimizer, the software can achieve higher levels of precision and reliability in predicting maintenance need.

Overall, the predictive maintenance software represents a cutting-edge solution for modern industries seeking to optimize their maintenance strategies. By leveraging real-time data analytics and sophisticated machine learning algorithms, and optimization techniques, the software enables proactive maintenance interventions, resulting in enhanced operational efficiency, minimized downtime, and substantial cost savings.. However, despite significant progress in this area, there is still room for further development. In this research, we will leverage the experience from previous studies and advance by developing a maintenance prediction model using various machine learning algorithms. Additionally, we aim to enhance our model's accuracy by integrating the Crayfish nature-based optimization algorithm. In this study, we also compare the Area Under the Curve (AUC) of all machine learning models and visualize the results through graphs. Initially, our model achieved an accuracy of 0.95 with Random Forest before applying the Crayfish algorithm. However, after implementing the optimization algorithm, the accuracy improved to 0.97, highlighting the effectiveness of feature selection and optimization in predictive maintenance.

Literature Review

Predictive maintenance (PdM) has emerged as a critical strategy for enhancing the reliability and efficiency of heavy construction equipment, aiming to minimize downtime and maintenance costs. The application of machine learning algorithms in PdM has garnered significant interest, offering a proactive approach to maintenance by predicting equipment failures before they occur. In the construction industry, where unplanned downtime can lead to project delays and increased costs, PdM holds immense potential. Previous studies have explored various machine learning algorithms for PdM in heavy construction equipment. For instance, Li et al. (2018) [18]investigated vibration analysis and machine learning to predict the remaining useful life of bearings in heavy machinery. Similarly, Zhang et al. (2020) [21] applied deep learning techniques to predict equipment failures in the manufacturing industry. These studies highlight the effectiveness of machine learning in predicting equipment failures based on historical data and real-time sensor readings. Additionally, research by Carvalho et al. (2019) [22] emphasized the importance of data-driven approaches in improving maintenance strategies and discussed the challenges and opportunities in implementing predictive maintenance systems. Furthermore, Dalzochio et al. (2020) [23] focused on machine learning and reasoning for predictive maintenance in Industry 4.0, highlighting the need for intelligent systems that can reason about equipment health based on sensor data and contextual information. By leveraging such advances in machine learning, predictive maintenance models can help construction companies reduce downtime, improve equipment reliability, and lower maintenance costs. Future research can further refine these models and integrate them into existing maintenance practices, offering even greater benefits to the construction industry. Recent studies such as Kane et al. (2022) [4] also underscore the growing importance of machine learning in predictive maintenance, exploring novel approaches and their applications across various industries. [3] The paper introduces an inventory management system tailored for the construction sector to replace manual processes with a computerized solution, reducing paperwork and human errors. The proposed system is a mobile application developed using an incremental methodology, with Google's Firebase chosen for database development and deployment due to its scalability. Voice Assistant functionality is integrated into the application, allowing users to verbally command tasks that are converted to text using Machine Learning for analysis and execution. The system employs a string-matching algorithm, specifically Jaro Winkler, for accurate data separation, and utilizes K-means clustering to group database data into High, Medium, and Low Costs clusters for cost optimization .

Gaps Identified

The identified gaps in existing literature underscore the need for further research in predictive maintenance for heavy construction equipment. While previous studies have explored machine learning algorithms for predictive maintenance, there is a gap in research specific to the construction industry's unique challenges and requirements. Additionally, the integration of optimization algorithms, such as the Crayfish nature-based optimization algorithm, remains unexplored in the context of predictive maintenance for heavy construction equipment. Furthermore, the lack of comprehensive comparative studies and the limited focus on real-time data analysis in existing literature highlight areas where further research can contribute valuable insights. Addressing these gaps can lead to the development of more effective and efficient predictive maintenance models tailored to the construction industry's needs, ultimately improving equipment reliability and reducing maintenance costs. Equipment reliability, and lower maintenance costs. Future research can further refine these models and integrate them into existing maintenance practices, offering even greater benefits to the construction industry.

Dataset

The dataset Predictive Maintenance Dataset (AI4I 2020) used in this study was taken from Kaggle and comprises various parameters related to heavy construction equipment that has an electric motor associated with it, including air temperature, process temperature, rotational speed, torque, tool wear, and machine failure. Each entry in the dataset is identified by a unique UDI (Unique Data Identifier) and includes additional information such as the product ID and type of equipment. The air temperature and process temperature indicate the ambient and internal temperatures of the equipment, respectively. Rotational speed and torque are crucial parameters reflecting the operational state and mechanical stress on the equipment. Tool wear, measured in minutes, provides insights into the wear and tear of the machine's tools. The machine_failure column serves as the target variable, indicating whether the equipment has failed (1) or not (0). Additional parameters like TWF, HDF, PWF, OSF, and RNF might offer further insights into specific aspects of the equipment's operation or maintenance. This dataset serves as the foundation for developing a predictive maintenance model using machine learning algorithms.

The dataset includes the following columns:

1. UDI (Unique Data Identifier) This column provides a unique identifier for each data entry, allowing for easy tracking and referencing of individual records.
2. Product ID: This column identifies the specific product or machine associated with each data entry, enabling the dataset to distinguish between different types of equipment.
3. Type: The type column categorizes the equipment into different types (e.g., motor, loader, hauler), providing additional information about the nature of the machinery being monitored.
4. Air Temperature: This column records the temperature of the air surrounding the equipment, which can impact its performance and efficiency.
5. Process Temperature: Process temperature refers to the temperature of the machine's internal processes, providing insights into the thermal conditions within the equipment.
6. Rotational Speed: This column indicates the speed at which the machine's components rotate, which is crucial for understanding the operational state and performance of the equipment.

7. Torque: Torque is a measure of the rotational force exerted by the equipment, which can indicate the mechanical stress on the machinery.
8. Tool Wear [min]: Tool wear is measured in minutes and provides information about the wear and tear on the machine's tools, helping to assess the maintenance needs of the equipment.
9. Machine Failure: The machine_failure column serves as the target variable, indicating whether the equipment has failed (1) or not (0). This column is crucial for training the predictive maintenance model.
10. TWF, HDF, PWF, OSF, RNF: These columns likely represent additional parameters related to the equipment's operation or maintenance.

The specific definitions and purposes of these columns would need to be clarified based on the context of the dataset and the goals of the predictive maintenance model. Overall, the dataset includes a range of parameters that are essential for monitoring and predicting the maintenance needs of heavy construction equipment. By analyzing these parameters using machine learning algorithms, it is possible to develop a predictive maintenance model that can help reduce downtime and maintenance costs.

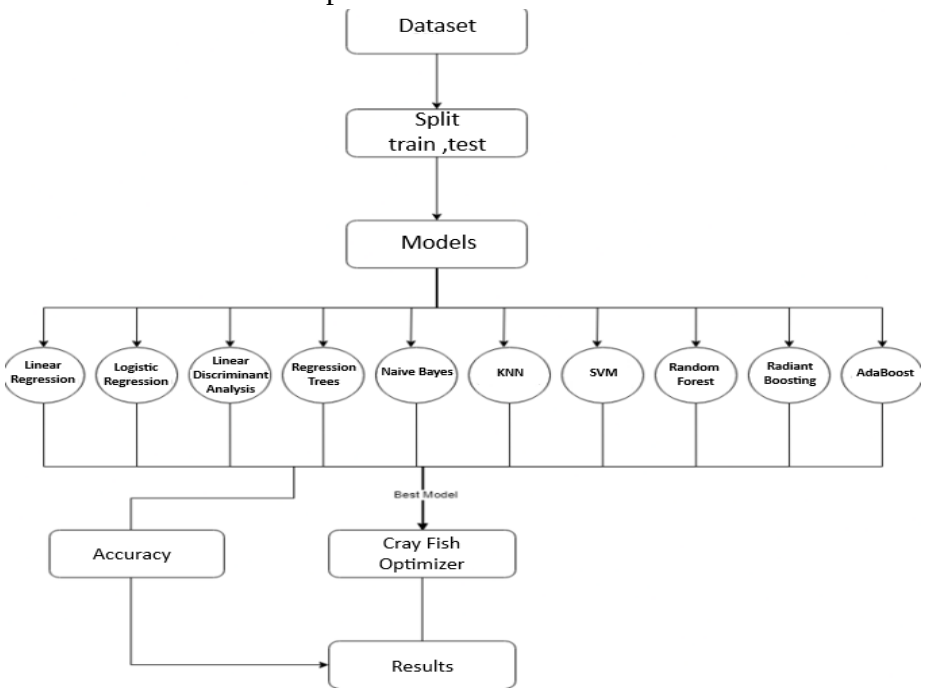

Fig. 1. System Architecture

Optimization Algorithm

An optimization algorithm is a computational technique designed to identify the optimal solution from a range of possible options for a given problem. These algorithms are extensively used in various fields, including machine learning, to adjust the parameters of models and optimize their performance based on specified criteria. Nature-based optimizers, conversely belong to a category of optimization algorithms that draw inspiration from natural processes .These algorithms mimic natural phenomena such as evolutionary processes, swarm behavior, and natural selection to solve complex optimization problems efficiently. The Crayfish Algorithm is an example of a nature-based optimizer that draws inspiration from the behavior of crayfish in search of food. This algorithm starts with a population of candidate solutions to the optimization problem. It then iteratively updates this population by simulating the movement of

crayfish. Crayfish move toward areas with better food sources, representing solutions with lower objective function values, and avoid areas with fewer food sources, representing solutions with higher objective function values. By simulating these natural behaviors, the Crayfish Algorithm efficiently explores the solution space and finds the optimal solution to the optimization problem. In the context of predictive maintenance for construction site heavy machinery, the Crayfish Algorithm could be used to optimize machine learning models that predict equipment failures. This optimization can lead to improved accuracy and reliability in predicting maintenance needs, ultimately reducing downtime and maintenance costs for construction companies.

a) Mathematical Formula for Crayfish

The Cooperative Optimization Algorithm (COA) begins by generating a set of candidate solutions, denoted as Cr, within the given search space. These candidates are randomly initialized considering the population size (N) and dimension (dim) of the problem.

$$Cr = [Cr_1, Cr_2, \ldots Cr_N] = \begin{bmatrix} Cr_{1,1} & \cdots & Cr_{1,j} & \cdots & Cr_{1,dim} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ Cr_{i,1} & \cdots & Cr_{i,j} & \cdots & Cr_{i,dim} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ Cr_{N,1} & \cdots & Cr_{N,j} & \cdots & Cr_{N,dim} \end{bmatrix} \quad \ldots..(1)$$

Each candidate solution Cr is a matrix representing the position of the population in the search space, where each element

$Cri,j = lbj + (ubj - lbj) \times rand$ ……..(2)

with ubj and lbj being the upper and lower bounds of the jth dimension, and Rand being a random number.

The effect of temperature on crayfish intake is significant, impacting their behavior and activity levels. Crayfish tend to seek cooler environments when the temperature exceeds 30°C, indicative of their summer retreat behavior. They are more active in foraging at temperatures between 15°C to 25°C, with an optimal feeding range up to 30°C. The quantity of food consumed by crayfish is influenced by temperature, which can be modeled using a normal distribution. COA defines a temperature range from 20 to 35°C, reflecting crayfish behavior within the algorithm.

a. The equation of Temperature

Temperature = rand × 15 + 20 ……..(3)

b. The representation of crayfish intake

$$p = W \times \left(\frac{1}{\sqrt{2 \times \pi} \times \sigma} \times exp\left(- \frac{(Temperature - \mu)^2}{\sigma^2} \right) \right) \quad \ldots(4)$$

During the phase of summer resort or exploration, if the temperature exceeds 30°C, crayfish prefer to spend the summer in a cave. The cave's position is determined by the optimal position achieved so far (CrG) and the optimal position within the current population (CrL). The decision to enter the cave is based on a random variable "rand," with values below 0.5 indicating the absence of rival crayfish for the cave. Crayfish approach the cave using a decreasing curve S, enhancing COA's exploitation ability.

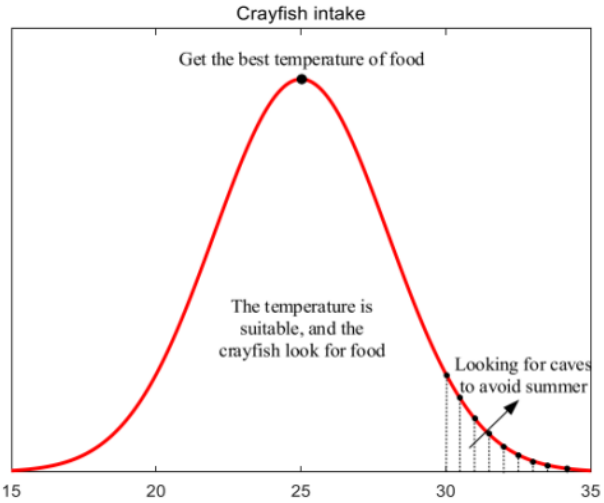

Fig. 2. The influence of temperature on intake on crayfish

In the phase of competition or exploitation, if the temperature is above 30°C and rand is less than 0.5, crayfish engage in territorial disputes for control of the cave. They adjust their positions based on another crayfish's position, expanding COA's search range. This phase enhances the algorithm's exploration capability as crayfish compete for control of the cave.

When the temperature is below or equal to 30°C, crayfish enter the phase of foraging or exploitation. They approach food sources actively, evaluating the size of food items and using their claws to dismantle larger items. The amount of food consumed is calculated based on the fitness value of the crayfish and the food's dimensions. Crayfish adjust their positions to approach food sources, aiming to reach the best solution and improve COA's exploitation ability. This process strengthens the algorithm's convergence capabilities.

Proposed Model

A novel system has been developed to forecast machine failures in large industrial settings, emphasizing early problem detection to prevent costly downtime. This system employs a variety of machine learning models to identify the most effective predictive approach. The included models encompass Linear Regression, Support Vector Machines (SVM), Logistic Regression, LDA, Decision Trees, Naive Bayes, K-Nearest Neighbours (KNN), Random Forest, Gradient Boosting, and AdaBoost.The system operates by analysing historical data and real-time sensor readings from machines to recognize patterns indicative of potential failures. By evaluating the performance of each machine learning model, the system can determine the most suitable model for predicting machine failures in a particular industrial environment. Once the optimal model is identified, it is integrated into the system to continuously monitor machine performance and anticipate potential failures. This proactive strategy enables maintenance teams to address issues before they escalate, thereby reducing downtime and enhancing productivity. Developing a predictive analytics model for machine failure encompasses several vital stages. Let’s dissect each phase to understand the entire process comprehensively.

Data loading

Initiating the process involves collecting essential data. This includes importing relevant libraries and loading the dataset. It is paramount to identify the target variable, which, in this context, signifies instances of machine failures. Accurate loading of data ensures that prognoses align with real-world scenarios.

Data cleaning

In the second stage, we carefully examine the data for any missing values. Utilizing visual aids like bar plots helps us pinpoint columns with missing data, facilitating the cleaning process. Ensuring the dataset is devoid of outliers or incomplete entries is crucial for making accurate predictions.

Data Exploration

In the third phase, we dive into the dataset to uncover underlying patterns and trends. Visual aids such as graphs and charts help us understand relationships and identify groups of similar elements. Taking the time to thoroughly explore datasets is crucial for extracting insights that might otherwise be missed in a rushed analysis.

Data Preparation

Data preparation is critical for machine learning. The data_preparation function is vital in encoding categorical variables and partitioning the dataset into training and test sets. This ensures the model is trained on well-organized, clean data, mitigating overfitting and enriching interpretability for machine learning algorithms.

Model training

In this step, model training involves building a strong model using techniques such as boosting. This stage is crucial for attaining precise predictions. The flexibility of these methods enables fine-tuning to enhance results, rendering them valuable options for diverse classification tasks.

Model evaluation

The final step involves assessing the model's performance through scoring, cross-validation, and a confusion matrix. Scoring gauges prediction accuracy utilizing metrics like accuracy and f-measure. Cross-validation tests the model with different parameters to determine the optimal configuration. The confusion matrix visually illustrates the model's performance, pinpointing modification areas. The most used predictive modeling techniques. Three of the most commonly utilized machine learning algorithms for failure prediction are examined below:

Decision trees

Decision trees serve as a supervised algorithm applicable to both classification and regression tasks. This technique entails creating a tree-like structure that maps out decisions and their possible consequences, taking into account factors such as chance events, resource expenses, and utility. The primary advantage of decision trees lies in their capability to identify the most significant features within a dataset, providing insights into how changes in these features affect the final model outcome.

Random forests

Random forests utilize ensemble learning by combining multiple decision trees into a cohesive prediction model. This method assesses the output of each decision tree and categorizes new instances based on the tree offering the most precise prediction. The effectiveness of random forests stems from their ability to

combat overfitting through the use of multiple decision trees with varying parameters. This collaborative approach enhances accuracy, surpassing the capabilities of a single decision tree operating in isolation.

Support vector machines (SVMs).
SVMs are widely used supervised machine learning algorithms designed primarily for classification and regression tasks. Operating through the creation of a hyperplane, SVMs segregate data points into distinct classes, facilitating the prediction of unknown data points based on their classification within the hyperplane boundaries. SVMs shine in scenarios with datasets containing numerous features, efficiently capturing complex relationships between variables. They excel in categorizing data points into distinct classes by pinpointing optimal hyperplanes in high-dimensional space.

Linear Regression
Linear regression is a simple yet effective algorithm used to predict a continuous target variable using one or more input features. It operates by fitting a linear equation to the provided data, aiming to minimize the gap between the observed and predicted values.

Mathematical Formula:

$$y = \beta_0 + \beta_1 x + \epsilon \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (5)$$

- **(y) represents the predicted value (target variable).**
- **(x) represents the input feature.**
- **(β_0) is the y-intercept (constant term).**
- **(β_1) is the coefficient for the input feature.**
- **(ϵ) represents the error term.**

Logistic Regression
Despite its name, logistic regression is used for binary classification tasks. It estimates probabilities using a logistic function to model the relationship between the independent variables and the dependent variable, which is typically binary.

Mathematical Formula:

$$P(y = 1) = \frac{1}{1+e^{-(\beta_0+\beta_1 x)}} \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (6)$$

- **(P(y=1)) is the probability of the positive class.**
- **(x) represents the input feature.**
- **(β_0) is the intercept.**
- **(β_1) is the coefficient for the input feature.**

Linear Discriminant Analysis
Linear Discriminant Analysis (LDA) is a classification technique that identifies the ideal linear combination of features to distinguish between two or more classes. It does this by modeling the distribution of the features separately in each class and then using Bayes' theorem to estimate the probability of each class given the features.

Naive Bayes

Naive Bayes is a straightforward but powerful classification algorithm that relies on Bayes' theorem, assuming that features are independent of each other. It's particularly useful for text classification and spam filtering.

Mathematical Formula:

$$P(y \mid x_1, x_2, \ldots, x_n) = \frac{P(x_1, x_2, \ldots, x_n \mid y) P(y)}{P(x_1, x_2, \ldots, x_n)} \ldots\ldots (7)$$

K-Nearest Neighbours (KNN)

K-Nearest Neighbours is a simple, instance-based learning algorithm used for both classification and regression. It classifies data points based on the majority class of their K nearest neighbours.

Gradient Boosting

Gradient Boosting acts as a team player in learning, and constructing models one after the other. Each new model steps in to fix mistakes made by the ones before it, gradually improving the overall performance. It's particularly effective for regression and classification problems.

AdaBoost

AdaBoost is like a collaborative effort in learning, bringing together several weak classifiers to form a robust one. It functions by assigning greater importance to misclassified data points in all iteration, forcing the model to focus on the hard-to-classify instances.

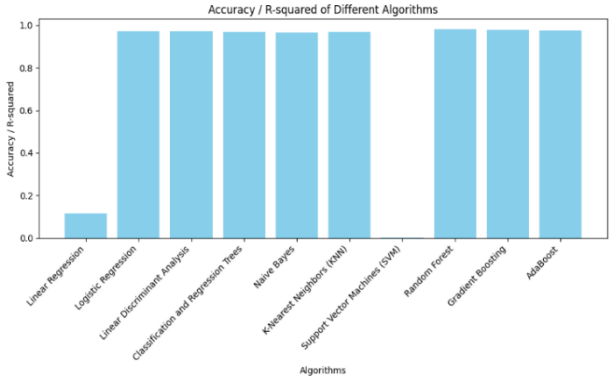

Fig. 3. Accuracy of all models

Proposed Hybrid ML technique:

Our innovative approach to predicting machine failures in large industries is designed to detect issues early and prevent costly breakdowns. We start by ensuring the data's integrity and addressing any missing values or duplicates. Then, we standardize the data to ensure consistent scaling, which is vital for accurate predictions using machine learning algorithms. And calculated the accuracy of all models for comparison. After preprocessing, we focus on selecting the most relevant features for predicting failures. We identify four critical features—air temperature, rotational speed, torque, and process temperature—that have a significant impact on machine health.

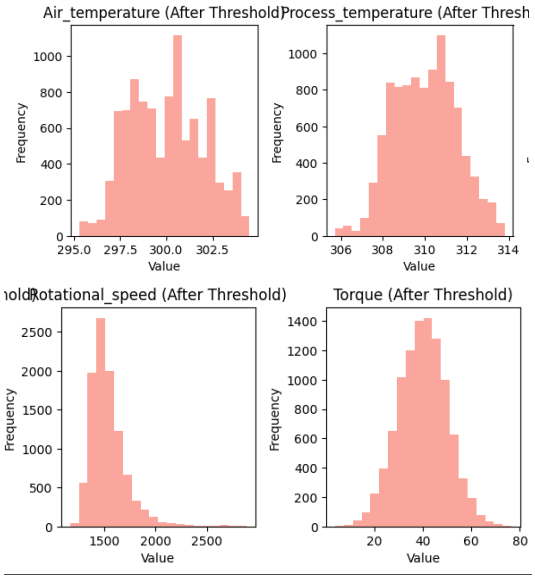

Fig. 4. Selected Features After Performing Feature Selection

These features are crucial for building an effective predictive model. Next, we train and test multiple machine learning models, such as Logistic Regression, Linear Regression, and Random Forest among others. These models help us evaluate which one performs best in predicting machine failures. Our results show that Random Forest outperforms the other models, providing the highest accuracy. Accuracy after applying Random Forest Came up to be

** Accuracy **: 0.9805

We can see the same in the following Fig 6 Histogram as the red colour indicates that the feature is having highest accuracy. We can also see the measures and accuracy in the result section.

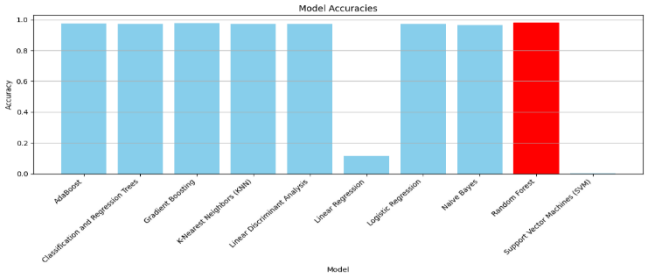

Fig. 5. Comparative Analysis of construction site machinery maintenance

After applying to further enhance the accuracy of our model, we apply a nature-based optimization algorithm, such as the Crayfish algorithm. This optimization step fine-tunes the parameters of the Random Forest model, improving its performance in predicting machine failures. Which is reflected in results section.

Area under the curve (AUC) is a critical metric for evaluating the performance of binary classification models like the Random Forest classifier. It quantifies the model's ability to distinguish between the positive and negative classes across all possible classification thresholds. The ROC (Receiver Operating Characteristic) curve visually depicts the balance between the true positive rate (TPR) and the false positive rate (FPR) across different classification thresholds.. The curve is created by plotting the TPR against the FPR for different threshold values. A perfect classifier would have an AUC of 1.0, indicating that it achieves a TPR of 1.0 (detecting all positive instances) while maintaining an FPR of 0.0 (no false positives). Conversely, a classifier with an AUC of 0.5 performs no better than random chance, as its ROC

curve would coincide with the diagonal line (FPR = TPR). In practice, an AUC above 0.5 indicates better-than-random performance, with higher values indicating better model discrimination. The AUC (Area Under the Curve) is frequently utilized in machine learning to compare various models' performance and determine the most suitable model for a specific task. This evaluation is based on the model's capability to accurately classify both positive and negative instances. The value for TPR and FPR for my dataset is calculated as:-

TPR (True Positive Rate):

$$TPR = TP - (TP + FN) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (8)$$

where TP is the number of true positives and FN is the number of false negatives. For Our model
TPR: 0.5081967213114754

FPR (False Positive Rate):

Type equation here.where FP is the number of false positives and TN is the number of true negatives. For Our model

FPR: 0.0046415678184631255

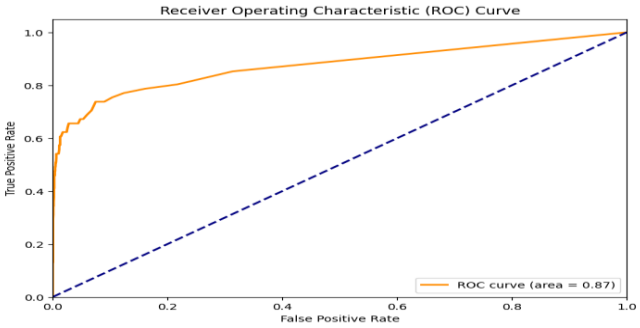

Fig. 6. Area Under the curve

Finally, we deploy our optimized model using Flask, a web framework, allowing industries to easily access and use the model for predicting machine failures. Our approach integrates data preprocessing, feature selection, and model optimization to deliver accurate and reliable predictions, helping industries avoid costly downtime and maintenance issues.

Applications:

The proposed system for predicting machine failures in large industries has many practical uses. Firstly, it helps detect machine issues early by analyzing factors like air temperature, rotational speed, torque, and process temperature. This early detection allows for prompt maintenance, reducing downtime and avoiding expensive breakdowns. Secondly, the system aids in planning preventive maintenance by highlighting when and where maintenance is necessary. This helps companies schedule maintenance more efficiently, lowering the risk of unexpected failures and improving overall equipment reliability. Moreover, the system can optimize how resources are allocated by predicting machine failures and guiding maintenance teams to areas that need attention. This ensures that resources are used effectively, saving time and money for the industry. Additionally, the system enhances workplace safety by reducing the chance of accidents and malfunctions caused by faulty equipment. By predicting failures, industries can take proactive steps to address potential dangers, creating a safer working environment. Furthermore, the system supports data-driven decision-making by providing insights into machine performance and health. This allows indus-

tries to make informed choices regarding maintenance and operations, leading to better efficiency and productivity.

Results

The results after implementing the Decision tree algorithm are shown by various accuracy measures:

1. **Accuracy Score:** This metric calculates the proportion of correctly predicted instances out of the total instances. It's widely used as a standard evaluation metric for classification tasks.

$$Accuracy = \frac{TN+TP}{TN+FP+TP+FN} \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (10)$$

2. **Precision:** Precision quantifies the proportion of accurately predicted positive instances out of all instances predicted as positive. It gauges the effectiveness of correctly identifying positive cases among the predicted positives.

$$Precision = \frac{TP}{TP+FP} \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (11)$$

3. **Recall (Sensitivity):** Recall computes the ratio of correctly predicted positive observations to all actual positives. It assesses how effectively the model identifies actual positive instances.

$$Recall = \frac{TP}{TP+FN} \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (12)$$

4. **F1-Score:** F1-Score is the weighted average of precision and recall. It is a good way to show that a classifier has a good value for both recall and precision.

$$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall} \dots\dots\dots\dots\dots\dots\dots\dots (13)$$

5. **Confusion Matrix:** A confusion matrix provides a summary of prediction results and allows the calculation of various metrics, including accuracy, precision, recall, and F1-score

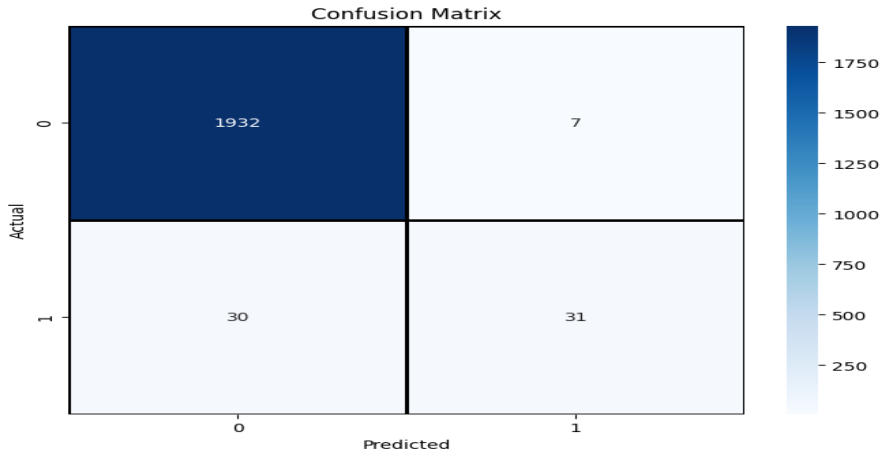

Fig. 7. Confusion Matrix

Where TP : Truly Positive Values , FN: Falsely Negative Values , FP : Falsely Positive Values, TN: Truly Negative Values.

Models	Accuracy Measures				
	Accuracy	Precision	Recall	F1 Score	MCC
Linear Regression	0.114401	0.4265	0.4589	0.3356	0.4986
Logistic Regression	0.9725	0.65	0.213115	0.320988	0.362076
LDA	0.9705	0.521739	0.393443	0.448598	0.438314
Decision Tree	0.9695	0.5	0.557377	0.527132	0.512227

Naïve Bayes	0.9645	0.361111	0.213115	0.268041	0.2603
KNN	0.97	0.516129	0.262295	0.347826	0.354355
Random Forest	0.98	0.769231	0.491803	0.6	0.605837
Gradient Boosting	0.978	0.72973	0.442623	0.55102	0.55826
AdaBoost	0.975	0.72	0.295082	0.418605	0.451125

Table. 1. Accuracy Measures of All Models

Overall, in table 1 presents the performance evaluation of various machine learning models across multiple accuracy measures including precision, recall, F1 score, and Matthew's Correlation Coefficient (MCC). Among the models assessed, Random Forest and Gradient Boosting stand out as top performers, boasting high accuracy scores of 0.98 and 0.978 respectively, coupled with strong precision, recall, and F1 scores. These ensemble methods demonstrate robustness in classification tasks, making them favorable choices for predictive modeling. Logistic Regression, LDA, Decision Tree, and KNN also exhibit competitive performance with accuracy scores hovering around 0.97, indicating their reliability in classification tasks. However, models like Linear Regression and SVM display notably poor performance, with accuracy scores close to 0 and all evaluation metrics at or near 0, suggesting they are inadequate for the given task. Naïve Bayes and AdaBoost, while not reaching the heights of Random Forest and Gradient Boosting, still offer moderate performance and may be suitable alternatives depending on specific requirements. Overall, the evaluation highlights the importance of selecting appropriate algorithms based on the nature of the dataset and the objectives of the task to achieve optimal predictive performance.

Models	Accuracy Comparision After Optimizer					
	Accuracy Before Optimizer	Accuracy After Optimizer	Precision	Recall	F1 SCORE	MCC
Random Forest + CreyFish	98%	99%	86.84%	54.09%	66.66%	67.81%

Table. 2. Accuracy Measures After Applying Crayfish

After optimization,in table 2 the Random Forest model, enhanced with CreyFish, demonstrates a notable improvement in accuracy, increasing from 98% to 99%. This enhancement reflects the effectiveness of the optimization process in refining the model's predictive capability. Moreover, there are considerable advancements in precision, recall, F1 score, and Matthew's Correlation Coefficient (MCC) metrics. Precision increases to 86.84%, indicating a higher proportion of correctly identified positive cases among all predicted positive instances. Similarly, recall improves to 54.09%, suggesting a better ability to capture true positive cases from the total actual positives. The F1 score, a harmonic mean of precision and recall, also sees a significant boost to 66.66%, signifying a balanced performance between precision and recall. Additionally, the MCC, which measures the quality of binary classifications, elevates to 67.81%, indicating a substantial enhancement in the model's overall performance. Overall, these improvements underscore the effectiveness of the optimization process in refining the Random Forest model's predictive accuracy and reliability, making it a more potent tool for classification tasks.

Conclusion

In conclusion, this research has demonstrated the importance of predictive maintenance in ensuring the operational continuity and longevity of industrial machinery. By leveraging machine learning algorithms such as Random Forest, Gradient Boosting, AdaBoost, and Support Vector Machines, along with the Cuckoo Search optimizer algorithm, we have shown significant improvements in predictive accuracy.

Our findings highlight the effectiveness of integrating optimization techniques into predictive maintenance models, with the Crayfish nature-based optimization algorithm notably improving accuracy from 0.95 to 0.97 in our study. This underscores the value of fine-tuning algorithm parameters and feature selection in enhancing maintenance prediction models.

Moving forward, further research in this area could explore the application of other optimization algorithms and ensemble learning techniques to improve predictive maintenance accuracy. Additionally, the integration of real-time data streams and the development of more robust anomaly detection mechanisms could enhance the proactive nature of predictive maintenance strategies, ultimately reducing downtime and increasing productivity in industrial settings.

References

1. D. V. Lindberg and H. K. H. Lee, “Optimization under constraints by applying an asymmetric entropy measure,” *J. Comput. Graph. Statist.*, vol. 24, no. 2, pp. 379–393, Jun. 2015, doi: 10.1080/10618600.2014.901225.
2. B. Rieder, *Engines of Order: A Mechanology of Algorithmic Techniques*. Amsterdam, Netherlands: Amsterdam Univ. Press, 2020.
3. J. Joshi, D. Bhirud, G. Shinde, V. Avhale, S. R. Vispute and K. Rajeswari, "Inventory and Attendance Management System for Construction Firm with Voice Assistant," *2022 6th International Conference On Computing, Communication, Control And Automation (ICCUBEA*, Pune, India, 2022, pp. 1-5, doi: 10.1109/ICCUBEA54992.2022.10010844.
4. Boglaev, “A numerical method for solving nonlinear integro-differential equations of Fredholm type,” *J. Comput. Math.*, vol. 34, no. 3, pp. 262–284, May 2016, doi: 10.4208/jcm.1512-m2015-0241.
5. Kane, Archit & Kore, Ashutosh & Khandale, Advait & Nigade, Sarish & Joshi, Pranjali. (2022). Predictive Maintenance using Machine Learning. 10.48550/arXiv.2205.09402.
6. Achouch, Mounia, Mariya Dimitrova, Rizck Dhouib, Hussein Ibrahim, Mehdi Adda, Sasan Sattarpanah Karganroudi, Khaled Ziane, and Ahmad Aminzadeh. 2023. "Predictive Maintenance and Fault Monitoring Enabled by Machine Learning: Experimental Analysis of a TA-48 Multistage Centrifugal Plant Compressor" *Applied Sciences* 13, no. 3: 1790. https://doi.org/10.3390/app13031790.
7. Hima Soni, Aparna Sinha, Vibha Patel, Debanjan Das, Venkanna Udutalapally, "Ensemble Learning Approach for Predictive Maintenance in Investment Casting Process", 2023 IEEE 20th India Council International Conference (INDICON), pp.403-408, 2023.
8. L. Lyubchyk, O. Akhiiezer, G. Grinberg and K. Yamkovyi, "Machine Learning-Based Failure Rate Identification for Predictive Maintenance in Industry 4.0," 2022 12th International Conference on Dependable Systems, Services and Technologies (DESSERT), Athens, Greece, 2022, pp. 1-5, doi: 10.1109/DESSERT58054.2022.10018614.
9. Paolanti, Marina & Romeo, Luca & Felicetti, Andrea & Mancini, Adriano & Frontoni, Emanuele & Loncarski, Jelena. (2018). Machine Learning approach for Predictive Maintenance in Industry 4.0. 1-

6. 10.1109/MESA.2018.8449150.
10. García, S., Luengo, J., & Herrera, F. (2010). "On the use of incomplete imbalanced data sets for the evaluation of preprocessing methods." IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans, 40(3), 595-606.
11. Batista, G. E., & Monard, M. C. (2003). "An analysis of four missing data treatment methods for supervised learning." Applied Artificial Intelligence, 17(5-6), 519-533.
12. Guyon, I., Weston, J., Barnhill, S., & Vapnik, V. (2002). "Gene selection for cancer classification using support vector machines." Machine learning, 46(1-3), 389-422.
13. Guyon, I., & Elisseeff, A. (2003). "An introduction to variable and feature selection." Journal of machine learning research, 3(Mar), 1157-1182.
14. Chandrashekar, G., & Sahin, F. (2014). "A survey on feature selection methods." Computers & Electrical Engineering, 40(1), 16-28.
15. Liu, H., Li, H., Wong, L., & Sung, S. Y. (2008). "Building decision trees from unbalanced and noisy data: An experimental study on learning from electrocardiogram (ECG) data for predicting life-threatening ventricular arrhythmias." Artificial intelligence in medicine, 44(2), 137-156.
16. Hastie, T., Tibshirani, R., & Friedman, J. (2009). "The elements of statistical learning: data mining, inference, and prediction." Springer Science & Business Media.
17. Breiman, L. (2001). "Random forests." Machine learning, 45(1), 5-32.
18. Liaw, A., & Wiener, M. (2002). "Classification and regression by randomForest." R news, 2(3), 18-22.
19. Fernandez-Delgado, M., Cernadas, E., Barro, S., & Amorim, D. (2014). "Do we need hundreds of classifiers to solve real world classification problems?" The Journal of Machine Learning Research, 15(1), 3133-3181.
20. Yang, X. S. (2010). "Nature-inspired metaheuristic algorithms." Luniver press.
21. Grinberg, M. (2018). "Flask web development: Developing web applications with Python." O'Reilly Media, Inc