Research Article GU J Sci, Part A, 12(1): 154-174 (2025) 10.54287/gujsa.1544942

Gazi University
Journal of Science
PART A: ENGINEERING AND INNOVATION
http://dergipark.org.tr/gujsa

Improving Machine Failure Prediction with Grey Wolf, Whale Optimization, and Optuna Techniques

Vahid SİNAP[1*]

[1] *Ufuk University, Department of Management Information Systems, Ankara, Türkiye*

Keywords

Machine Failure Prediction

Grey Wolf Optimization

Whale Optimization Algorithm

Optuna

Abstract

Machine failure prediction is crucial for minimizing downtime and optimizing maintenance strategies in industrial settings. This study aims to enhance the accuracy of machine failure prediction models by integrating advanced hyperparameter optimization techniques with feature selection methods. Various optimization techniques, including Optuna, Hyperopt, and Spearmint, were evaluated, along with feature selection methods utilizing Grey Wolf Optimization (GWO) and Whale Optimization Algorithm (WOA). The findings reveal that the CatBoost model optimized with GWO and Optuna achieved the highest performance, with an accuracy of 88.3%, an F1 score of 88.3%, and a Matthews Correlation Coefficient (MCC) of 76.7%. In comparison, WOA demonstrated competitive yet slightly lower results, with the best accuracy of 85.9% achieved using CatBoost and Optuna. The study also highlights that Linear Discriminant Analysis (LDA), optimized with Optuna, showed notable performance, with an accuracy of 86.0%, an F1 score of 85.8%, and an MCC of 74.6% without feature selection, which improved to 87.8%, 87.8%, and 76%, respectively, with GWO-based feature selection. The overall results indicate that GWO outperforms WOA in improving model performance, particularly when paired with advanced hyperparameter tuning techniques.

Cite

Sinap, V. (2025). Improving Machine Failure Prediction with Grey Wolf, Whale Optimization, and Optuna Techniques. *GU J Sci, Part A*, *12*(1), 154-174. doi:10.54287/gujsa.1544942

Author ID (ORCID Number)

0000-0002-8734-9509 Vahid SİNAP

Article Process

Submission Date	06.09.2024
Revision Date	04.12.2024
Accepted Date	30.12.2024
Published Date	26.03.2025

1. INTRODUCTION

The industrial maintenance strategies are devised in a way that they utilize the predictive failure methodology to avoid system failures and thus reduce the unplanned downtime of equipment, machines, and processes. To this end, accurate prediction of future failures is essential to ensure timely planning of maintenance activities. Failure prediction approaches utilize both past and market data which signify system states, events and operations to predict the outcomes. Accurately analyzing this data enables failures to be predicted and preventive maintenance activities to be carried out in a timely manner. Fault prediction plays a critical role in the safety, efficiency and cost effectiveness of industrial operations. Unplanned downtime can lead to production losses, increased repair costs and even customer dissatisfaction. For example, Amazon's 49-minute outage in 2013 resulted in $4 million in lost sales (Tweney, 2013). According to a survey conducted by the Ponemon Institute, an organization reportedly loses an average of $138,000 per hour due to an outage (Ponemon Institute, 2011). Preventing such losses requires the effective use of failure prediction methods.

*Corresponding Author, e-mail: vahidsinap@gmail.com

Machine failures can be caused by various factors. These factors include mechanical wear and tear, improper assembly, improper use, environmental conditions such as excessive temperature and humidity. Mechanical wear can adversely affect the performance of components over time, leading to failures. Furthermore, uneven distribution of loads on equipment can also trigger the occurrence of failures. Incorrect installation situations can lead to improper machine operation and potential breakdowns, while incorrect usage scenarios can lead to unexpected downtime caused by operator errors. Environmental conditions, especially factors such as high temperature and humidity, stress the operating conditions of machines, increasing the likelihood of failure. Therefore, it is crucial for failure prediction systems to consider this variety of factors and analyze the data to make reliable predictions. Failure prediction increases the continuity and efficiency of industrial processes by identifying such situations in advance, enabling maintenance strategies to be implemented effectively (Zonta et al., 2020).

Over the last few years, machine learning (ML) has become more and more relevant and vital to prediction model training, evaluation of model performance, and deployment in the production environment. ML-based failure prediction models are increasingly being adopted due to the rapid progress of machine learning algorithms along with their free accessibility in software packages, and the abundance of industrial data from big data analytics and stream processing platforms (Leukel et al., 2021). ML-based failure prediction models are known to be effective for various systems such as agricultural machinery, wind turbines, aircraft components, information and communication technology devices, and manufacturing plants. In studies on agricultural machinery, predicting failures using sensor data and operating conditions increases machine efficiency and reduces unexpected downtime (Abdallah et al., 2021). Models developed for agricultural machinery are critical, especially during harvest periods, and ensure the continuity of these machines. Wind turbines are another important application area. In wind turbine failure prediction, a wide range of parameters such as vibration analysis and power generation data are used to determine the probability of failure in advance (Jin et al., 2021). Such an approach increases energy efficiency and reduces maintenance costs. ML-based studies on aircraft components perform component life predictions and failure risk analysis to improve the safety of aircraft (Li et al., 2023). Information and communication technology devices are another area where ML applications are effective. Failure predictions on network devices develop proactive maintenance strategies by analyzing system monitoring data to improve user experience and minimize service interruptions (Rzayeva et al., 2023). In manufacturing plants, ML-based fault prediction systems are used to monitor the status of machines on production lines and detect abnormal behavior (Ayvaz & Alpay, 2021).

The effectiveness of machine learning algorithms in the field of fault prediction is closely related not only to the selection of the right model, but also to the feature selection processes and hyperparameter optimization. Feature selection is a critical step that affects the success of the model, because in high-dimensional datasets, redundant or irrelevant features can increase the complexity of the model and lead to overlearning. Improper selection of features can negatively affect the overall performance of the model and lead to low accuracy in failure predictions. Hyperparameter optimization is another important step to improve the performance of machine learning models. Proper tuning of hyperparameters can greatly affect the generalization ability of the model. However, the process of hyperparameter optimization is often time-consuming and can be costly in terms of computational resources depending on the size of the dataset. Moreover, the performance results obtained with a given combination of hyperparameters may vary depending on the complexity of the model and the features used. This makes it difficult to find optimal hyperparameters. These limitations may affect the accuracy and reliability of the ML models used in failure prediction, reducing the applicability of the results. Therefore, effective feature selection and hyperparameter optimization process is vital for the success of failure prediction systems.

The field of failure prediction and industrial maintenance has witnessed substantial advancements through various research efforts, driven by the need to enhance operational efficiency and minimize unplanned downtimes. A significant body of work has focused on developing and refining methods for predicting failures in industrial systems, utilizing both traditional statistical approaches and modern machine learning techniques. These studies have contributed to a deeper understanding of failure mechanisms, improved predictive accuracy, and the development of more effective maintenance strategies.

Wahid et al. (2022) present a novel forecasting method for predictive maintenance (PdM) which applies a hybrid model with convolutional neural networks (CNN) and long short-term memory (LSTM) that includes a skip connection (CNN-LSTM). This approach leverages the strengths of both CNNs for high-level feature extraction and LSTMs for analyzing long-term dependencies in time-series data. Experiments conducted with Microsoft's case study dataset, containing failure history, maintenance records, error conditions, and telemetry data, show that the hybrid CNN-LSTM model achieves the highest prediction accuracy for machine failure compared to individual CNN or LSTM models.

Celikmih et al. (2020) focused on predicting aircraft system failures using machine learning models and a hybrid data preparation approach. They collected two years' worth of maintenance and failure data, identifying nine input variables and one output variable. Their method involves a two-stage process: using ReliefF for feature selection and a modified K-means algorithm to remove noisy data. The model's performance was evaluated with Multilayer Perceptron (MLP), Support Vector Regression (SVR), and Linear Regression (LR), using metrics such as Correlation Coefficient (CC), Mean Absolute Error (MAE), and Root Mean Square Error (RMSE). The study concludes that their hybrid approach effectively predicts failure counts.

Mishra and Manjhi (2018) addressed the challenges faced by financial organizations in managing a mix of new and old technologies within their branches, including ATMs and self-service devices. They proposed a service-oriented, vendor-focused approach to integrate maintenance and technical support strategies effectively. Their study emphasizes the importance of predictive maintenance in optimizing branch operations by leveraging predictive analytics and machine learning technologies. The proposed method and machine learning model predict the likelihood of device and component failures within a specified future timeframe, aiming to enhance customer satisfaction and improve overall business performance.

Campos et al. (2019) investigated the use of heterogeneous ensembles of ML techniques for Online Failure Prediction (OFP). They highlighted that while individual ML models are effective, combining various algorithms into ensembles can enhance performance by leveraging different biases. The study assessed the effectiveness of different base learners and combination methods to improve prediction accuracy. Results indicated that certain combinations of learners and techniques, even if not the best individually, can significantly enhance failure prediction. Additionally, analyzing the interactions between learners in successful ensembles provided valuable insights into their improved performance.

Wang et al. (2017) suggested a technique for performance monitoring and failure prediction in optical networks by means of machine learning. They assisted SVM (Support Vector Machine) and Double Exponential Smoothing (DES) to construct risk-aware models that are particular for equipment failure prediction. The study identified that this issue has not been thoroughly addressed before. Experimental results demonstrated that the DES-SVM method achieved an average prediction accuracy of 95% for forecasting optical equipment failures. This high accuracy indicates that the proposed method significantly improves traditional risk-aware models, enhancing the stability and reliability of optical networks by effectively predicting and mitigating equipment failure risks.

Mohammed et al. (2019) have presented the issue of failure prediction in high-performance computing and cloud systems, which is critical. As these systems have become more complicated, traditional strategies of fault-tolerance like regular checkpointing and replication are no longer sufficient. This study aims to examine the role of machine learning in the enhancement of the accuracy of failure prediction. The scientists invent the model of predicting failure by applying time series analysis and various machine learning algorithms, such as SVM, RF, KNN, Classification and Regression Trees (CART), and LDA. Their experimental results showed that the SVM-based model achieved an average prediction accuracy of 90%, demonstrating its effectiveness in predicting system and application failures more accurately compared to other algorithms. This suggests that machine learning techniques can significantly improve failure management and proactive measures in high-performance computing systems.

Pellegrini et al. (2015) introduced the Framework for Building Failure Prediction Models (F2PM), a machine learning-based framework designed to predict the Remaining Time to Failure (RTTF) of applications experiencing software anomalies. F2PM operates independently of specific applications by focusing on system-level features, thus enabling its use across various contexts without requiring manual adjustments. The framework performs feature selection to identify key system features that significantly impact RTTF prediction, allowing the creation of optimized models tailored to different feature sets. Users can evaluate these models based on prediction accuracy and model-building time, with experimental results demonstrating the framework's efficacy using the TPC-W e-commerce benchmark.

This study distinguishes itself from existing research by not only examining the performance of various machine learning algorithms for failure prediction but also comprehensively evaluating the combined impact of nature-inspired feature selection methods and modern hyperparameter optimization techniques. While previous studies often focus on specific approaches in isolation, this research integrates GWO and WOA for feature selection with hyperparameter optimization techniques including Optuna, Hyperopt, and Spearmint. GWO, inspired by the natural hunting behavior of grey wolves, provides an efficient mechanism for selecting optimal features, while WOA, based on whale hunting strategies, excels in handling complex data structures. For hyperparameter optimization, Optuna dynamically automates optimization processes, Hyperopt leverages Bayesian optimization for efficient searches in high-dimensional parameter spaces, and Spearmint enhances performance when the objective function is uncertain. The study evaluates these techniques on machine learning models, including Random Forest (RF), Gradient Boosting (GB), CatBoost, LDA, and K-Nearest Neighbors (KNN), to identify the most effective combinations for fault prediction. The findings aim to reveal how the integration of feature selection and hyperparameter optimization can enhance model reliability and efficiency, ultimately improving predictive maintenance strategies in industrial applications.

2. MATERIAL AND METHOD

This section outlines the dataset characteristics, details of the machine learning algorithms employed, the performance metrics used for algorithm comparison, and the data preparation procedures. The objective is to establish a robust methodological and analytical framework for the research, thereby enhancing the scientific contribution of the study and ensuring its reproducibility.

2.1. Dataset

The dataset used in this research is publicly available on the Kaggle platform. It contains sensor data collected from various types of machinery, such as industrial pumps, air compressors, and CNC machines, with the aim of predicting machine failures in advance. The dataset comprises 944 entries and is composed of multiple features recorded from sensors installed on the machines, along with a binary indicator of machine failures. A detailed description of each feature included in the dataset is provided in Table 1.

The heatmap in Figure 1 presents the correlation coefficients between various operational parameters of a machine and the binary failure indicator. Notably, the “fail” variable demonstrates a moderate positive correlation with the AQ (r = 0.58), indicating that elevated air pollution levels are associated with a higher likelihood of machine failure. Conversely, there is a moderate negative correlation between “fail” and the CS readings (r = -0.47), suggesting that reduced electrical current usage may be linked to an increased risk of failure. Additionally, the heatmap reveals significant interdependencies among features, such as a strong positive correlation between VOC levels and fail rate (r = 0.80), highlighting the complex interactions within the machine's operational environment.

Table 1. *Dataset Features and Descriptions*

Feature	Description
footfall	The count of people or objects passing by the machine.
tempMode	The temperature setting or mode of the machine.
AQ	The air quality index near the machine.
USS	Ultrasonic sensor readings indicating the proximity of objects to the machine.
CS	Electrical current usage of the machine as measured by current sensors.
VOC	The level of volatile organic compounds detected near the machine.
RP	The rotational position or revolutions per minute (RPM) of the machine parts.
IP	The pressure of the input supplied to the machine.
Temperature	The current operating temperature of the machine.
fail	A binary indicator of machine failure (1 indicates failure, 0 indicates no failure).

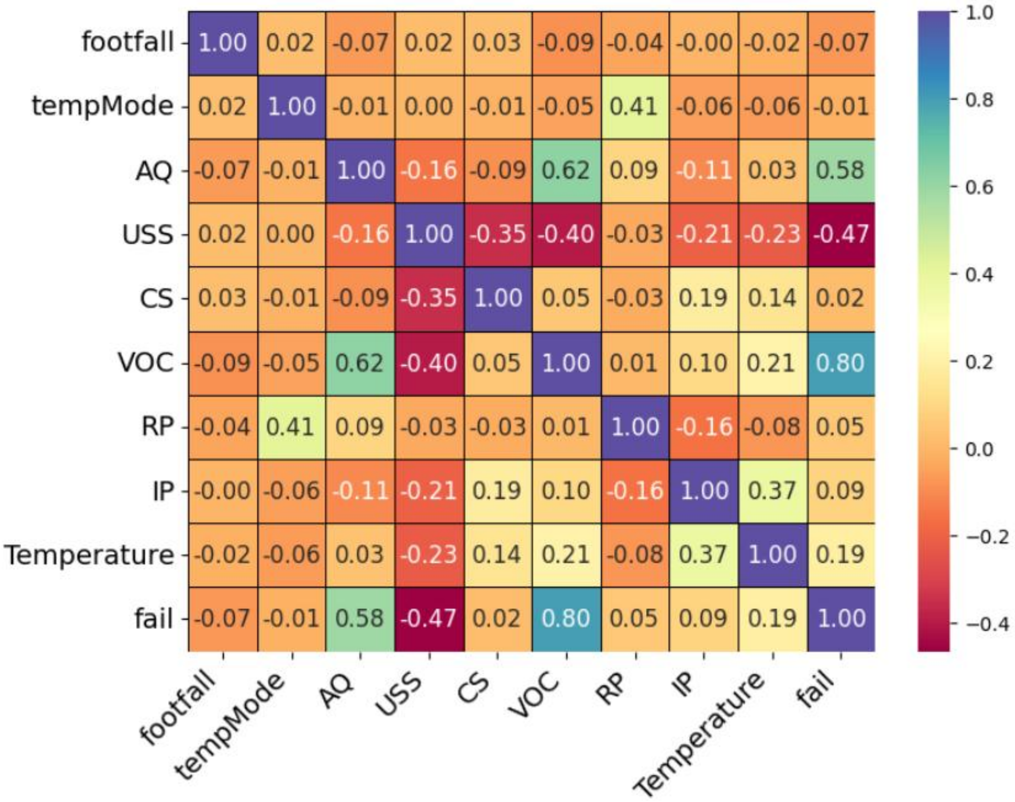

Figure 1. *Heatmap of the Dataset*

2.2. Data Preparation

The descriptive statistics of the dataset reveal varying scales and distributions among the features. For instance, the footfall variable exhibits a wide range (0 to 7300) with a mean of 306.38 and a high standard deviation of

1082.61, indicating significant variability in the number of people or objects passing by the machine. In contrast, other parameters, such as the AQ, USS, and CS, have more limited ranges and are approximately normally distributed, with moderate standard deviations (1.44, 1.38, and 1.27, respectively). Data normalization was conducted to minimize the quite large disparities between different features; thus, all the variables can affect equally the model's performance. The dataset is complete with no missing values, thereby the implemented methods of imputation were unnecessary.

2.3. Ensemble Learning

Ensemble learning is a powerful machine learning approach that combines multiple base models to improve overall prediction performance and robustness. In this study, two ensemble methods were employed: Voting Ensemble Classifier (VEC) and Stacking Ensemble Classifier (SEC). Both methods leverage the strengths of multiple classifiers to achieve better generalization compared to individual models. The VEC combines the predictions of several base learners through a majority voting scheme (hard voting) or by averaging their predicted probabilities (soft voting) (Lasotte et al., 2022). The prediction of the VEC is formally defined, as given in (1).

$$\hat{y} = \operatorname{argmax} \sum_{i=1}^{N} \omega_i \cdot f_i(x) \tag{1}$$

where $\hat{y}$ represents the final predicted class, ω_i denotes the weight assigned to the i-th base classifier $f_i(x)$, and N is the total number of classifiers.

The SEC, on the other hand, is a more sophisticated approach that trains a meta-classifier to the ensemble of predictions obtained from various base classifiers. In this technique, the outputs of the base classifiers are used as input features for the meta-classifier, which learns to optimally combine them to minimize error (Rajadurai & Gandhi, 2022). The prediction formula for SEC is expressed, as given in (2).

$$\hat{y} = g\big(f_1(x), f_2(x), \dots, f_N(x)\big) \tag{2}$$

where g is the meta-classifier function, and $f_1(x), f_2(x), \dots, f_N(x)$ are the base classifiers.

The choice of ensemble methods in this study was driven by their ability to address the limitations of single classifiers, such as overfitting and high variance, particularly in complex datasets with multiple operational parameters. For the VEC, base learners were selected to provide diverse perspectives: CatBoost, known for its effectiveness with categorical data and boosting capabilities; RF, which offers robustness through an ensemble of decision trees; and KNN, valued for its simplicity and ability to capture local patterns. In contrast, the SEC employs a meta-learning approach where a meta-learner is trained to optimally combine the outputs of several base learners. For this study, CatBoost was selected as the meta-learner due to its proficiency in managing complex interactions and its ability to effectively synthesize predictions from the base models, which include CatBoost, RF, and KNN.

2.4. Feature Selection

In this study, the GWO and the WOA were utilized for feature selection. The choice of features is a very important stage in the technique of machine learning, targeting to improve the performance of the model by recognizing the most relevant features and minimizing the dimensionality. This process helps in mitigating overfitting, improving model interpretability, and reducing computational costs. The GWO began by initializing a population of potential solutions, each representing a different subset of features from the dataset. These solutions were organized into a social hierarchy consisting of alpha, beta, delta, and omega wolves. The alpha wolves, representing the best-performing feature subsets, guided the optimization process, while the beta and delta wolves provided additional guidance to prevent premature convergence and maintain diversity in the search process (El-Kenawy & Eid, 2020). Throughout the optimization process, the GWO algorithm simulated the natural hunting strategies of grey wolves, consisting of three main phases: encircling, hunting, and attacking the prey. In the encircling phase, wolves updated their positions based on the locations of the alpha, beta, and delta wolves, enabling them to concentrate on the most promising areas of the feature space. During the hunting phase, the GWO calculated a weighted average of the positions of the top wolves, balancing exploration of new areas with exploitation of known good solutions. As the optimization progressed into the attacking phase, the algorithm increased its emphasis on exploitation to converge on the optimal subset of features.

The WOA, based on the hunting behavior of humpback whales, was also utilized for feature selection. WOA mimics the bubble-net feeding strategy of whales, which involves encircling prey and employing a spiral search technique. This approach enables WOA to navigate the feature space effectively and identify optimal feature subsets. WOA’s strength lies in its capacity to escape local optima and converge towards global solutions, thereby selecting features that contribute significantly to the model’s performance (Khaire & Dhanalakshmi, 2022).

2.5. Validation Method

Cross-validation is a widely used technique for assessing the performance and generalization ability of machine learning models. In this context, 5-fold cross-validation was implemented to ensure a comprehensive evaluation of the models. This method involves partitioning the dataset into five distinct subsets or "folds." In this process, every fold is used once as a test set while the remaining four folds are joined together to create the training set. This process is carried out a total of five times, with each fold serving as the test set once and only once. The 5-fold cross-validation procedure can be described using the following equation, as shown in (3).

$$\mathrm{CV} = \frac{1}{k}\sum_{i=1}^{k} \mathrm{Performance}(D_i) \tag{3}$$

Formally, let D denote the entire dataset, and D_i represent the i-th fold, where i ranges from 1 to 5.

2.6. Performance Metrics

To evaluate the performance of the models, several key metrics were utilized: Accuracy, F1 Score, and MCC. Each of these metrics provides different insights into the model's predictive capabilities and overall performance.

Accuracy: The accuracy is calculated as the number of correctly classified instances divided by the total number of instances. It is defined as (4):

$$\mathrm{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

where TP is the number of true positives, TN is the number of true negatives, FP is the number of false positives, and FN is the number of false negatives.

F1 Score: The F1 Score is the harmonic mean of Precision and Recall which gives a balance between the two metrics. It is particularly useful for imbalanced datasets. The F1 Score is provided in (5).

$$\mathrm{F1\ Score} = 2 \times \frac{\mathrm{Precision} \times \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}} \tag{5}$$

where Precision is defined as (6):

$$\mathrm{Precision} = \frac{TP}{TP + FP} \tag{6}$$

and Recall is defined as (7):

$$\mathrm{Recall} = \frac{TP}{TP + FN} \tag{7}$$

MCC: MCC provides a balanced measure of classification performance that considers all four categories of the confusion matrix. The MCC is calculated as shown in (8).

$$\mathrm{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP) \times (TP + FN) \times (TN + FP) \times (TN + FN)}} \tag{8}$$

2.7. Model Setups

The dataset is divided into two parts, with 80% allocated for training and 20% for testing, using a random state of 42 to ensure reproducibility. For determining the hyperparameters, three optimization techniques -Optuna, Hyperopt, and Bayesian optimization (Spearmint)- were employed, each bringing distinct advantages to the optimization process.

Optuna is a self-tuning hyperparameter optimization framework which finds optimal hyperparameters by applying a dynamic and efficient search algorithm. It employs a tree-structured Parzen estimator (TPE) for Bayesian optimization but is also capable of other search strategies like grid search or random search. Optuna allows for the definition of an objective function and performs optimizations in a highly flexible and scalable manner (Parra-Ullauri et al., 2023). It can prune unpromising trials early in the optimization process, speeding up the search and enhancing computational efficiency. Hyperopt is another popular hyperparameter optimization tool that focuses on efficient searching through large, high-dimensional spaces. It is based on Bayesian optimization, using the TPE as a surrogate model to guide the search process towards promising areas. Unlike traditional grid or random searches, Hyperopt incrementally builds a model of the objective function, enabling it to make informed decisions on where to sample next (Luo, 2016). This allows Hyperopt to effectively explore and exploit the hyperparameter space, particularly for complex and high-dimensional optimization tasks. Spearmint is a specific implementation of Bayesian optimization that is designed to handle black-box optimization problems where the objective function is expensive to evaluate (Archetti & Candelieri, 2019). It models the objective function using Gaussian processes and utilizes the expected improvement criterion to choose the next set of hyperparameters to evaluate (Young et al., 2018). Spearmint is particularly effective in scenarios where the evaluation of each set of hyperparameters is time-consuming or costly, as it aims to minimize the number of evaluations required to find the optimum solution. In this study, only the hyperparameter settings obtained through Optuna are presented, as it yielded the best results. Table 2 provides the optimized hyperparameter settings for the models.

For Optuna, the configuration included a total of 100 trials with the TPESampler for efficient search and the MedianPruner to eliminate fewer promising trials early. Hyperopt was utilized with a maximum of 100 evaluations, employing the TPE algorithm for Bayesian optimization, and incorporated various hyperparameter distributions such as uniform and choice. Bayesian Optimization, specifically Spearmint, was configured with 100 iterations and used the Expected Improvement acquisition function, with parameters set to kappa=1.96 and xi=0.01 to balance exploration and exploitation. Table 3 presents the hyperparameter settings used for each optimization technique.

The data analysis and model testing were conducted using the Python programming language. For data processing and manipulation, the pandas and NumPy libraries were employed, providing robust tools for handling large datasets. The scikit-learn library was utilized for model development and evaluation, offering a comprehensive range of machine learning algorithms and performance metrics. All analyses were performed within the Jupyter Notebook environment, which integrates code, text, and visualizations seamlessly. The computational experiments were executed on a PC equipped with an AMD Ryzen 7800X3D processor, operating at 4.2 GHz, and an NVIDIA GeForce RTX 4070 Ti GPU, supported by 32 GB of 6000 MHz DDR5 RAM. The system ran on Windows 11, ensuring a stable and efficient development environment.

Table 2. *Hyperparameter Settings for Models Optimized Using Optuna*

Model	Hyperparameter	Settings
RF	n_estimators, max_depth, min_samples_split, min_samples_leaf	120, 12, 3, 2
GB	n_estimators, learning_rate, max_depth, min_samples_split, min_samples_leaf	221, 0.07, 6, 3, 2
CB	iterations, learning_rate, depth, l2_leaf_reg, bagging_temperature	545, 0.03, 8, 3, 0.9
LDA	solver, shrinkage, n_components, tol	“lsqr”, “auto”, 1, 0.0001
KNN	n_neighbors, weights, algorithm, leaf_size, p	7, “distance”, “auto”, 30, 2
VEC	estimators, weights	[(“CatBoost”, CB()), (“RF”, RF()), ("KNN", KNN())], [1, 2, 1]
SEC	estimators, final_estimator	[(“CatBoost”, CB()), (“RF”, RF()), (“KNN”, KNN())], CatBoost

Table 3. *Hyperparameter Settings for Optimization Techniques*

Optimization Technique	Settings
Optuna	n_trials=100, sampler=TPESampler, pruner=MedianPruner
Hyperopt	max_evals=100, algo=tpe.suggest, domain=hp.choice, hp.uniform
Bayesian Optimization	n_iter=100, acq_func=Expected Improvement, kappa=1.96, xi=0.01

3. EXPERIMENTAL STUDY AND FINDINGS

The experimental phase of the study is examined in four distinct stages. In the first stage, the model performances on sub-datasets without feature selection and hyperparameter optimization are investigated and compared. The second stage focuses on the application of hyperparameter optimization techniques, Optuna, Hyperopt, and Spearmint, without feature selection. The third stage involves an evaluation of feature selection combined with hyperparameter optimization methods using GWO. Finally, the fourth stage assesses the impact of WOA on hyperparameter optimization methods. Table 4 provides the results for model performance without feature selection and hyperparameter optimization.

When examining Table 4, it is evident that among the evaluated models, CatBoost stands out with the highest performance across all metrics. It achieved an Accuracy of 0.797, an F1 Score of 0.795, and an MCC of 0.716. The average performance across models is an Accuracy of 0.787, an F1 Score of 0.785, and an MCC of 0.700, with a standard deviation of 0.0080 for Accuracy, 0.0078 for F1 Score, and 0.0107 for MCC.

Table 4. *Model Performance without Feature Selection and Hyperparameter Optimization*

Models	**Accuracy**	**F1**	**MCC**
CatBoost	0.797	0.795	0.716
LDA	0.790	0.788	0.701
GB	0.791	0.789	0.705
KNN	0.775	0.773	0.684
RF	0.780	0.779	0.694
Avg.	0.787	0.785	0.700
Std. Dev.	0.0080	0.0078	0.0107
VEC	0.785	0.782	0.703
SEC	0.790	0.787	0.708

Table 5 presents the performance metrics of various models optimized with different hyperparameter tuning techniques, specifically Optuna, Hyperopt, and Spearmint. Among the evaluated models, LDA optimized with Optuna achieved the highest performance, with an Accuracy of 0.860, an F1 Score of 0.858, and an MCC of 0.746. This indicates that LDA, when combined with Optuna for hyperparameter optimization, outperforms the other algorithms in terms of Accuracy, F1 Score, and MCC. When comparing these results to those presented in Table 4, where models were assessed without feature selection and hyperparameter optimization, we observe a notable improvement. For instance, CatBoost's performance increased from an Accuracy of 0.797, an F1 Score of 0.795, and an MCC of 0.716 to 0.823 with Optuna, 0.858 with Hyperopt, and 0.812 with Spearmint. Similarly, LDA's performance improved from an Accuracy of 0.790, an F1 Score of 0.788, and an MCC of 0.701 to the highest values with Optuna. The average performance metrics across models with hyperparameter optimization are also higher compared to those without. Specifically, the average Accuracy, F1 Score, and MCC are 0.829, 0.828, and 0.720 with Optuna; 0.823, 0.820, and 0.715 with Hyperopt; and 0.799, 0.797, and 0.708 with Spearmint. This improvement underscores the significant impact of hyperparameter tuning on enhancing model performance. The standard deviations of the performance metrics reveal variability among the different optimization techniques. Optuna shows standard deviations of 0.0194 for Accuracy, 0.0194 for F1 Score, and 0.0155 for MCC. Hyperopt exhibits higher variability with standard deviations of 0.0267 for Accuracy, 0.0263 for F1 Score, and 0.0239 for MCC, while Spearmint has the lowest variability, with standard deviations of 0.0098 for Accuracy, 0.0102 for F1 Score, and 0.0150 for MCC. Figure 2 shows the accuracy performance of different models optimized using Optuna, Hyperopt, and Spearmint without applying feature selection.

Table 6 illustrates the performance of various models optimized using GWO with three different hyperparameter tuning methods. Among the models evaluated, CatBoost optimized with Optuna achieves the highest performance, with an Accuracy of 0.883, an F1 Score of 0.883, and an MCC of 0.767. When comparing these results with those from Table 5, which showcases models without feature selection, it is evident that

GWO significantly enhances performance. For instance, CatBoost's performance with GWO and Optuna (Accuracy of 0.883, F1 Score of 0.883, MCC of 0.767) surpasses its performance without feature selection (Accuracy of 0.823, F1 Score of 0.821, MCC of 0.720). Similarly, LDA's performance improved from an Accuracy of 0.860, an F1 Score of 0.858, and an MCC of 0.746 with Optuna to 0.878, 0.878, and 0.760 respectively with GWO and Optuna. The average performance metrics with GWO are generally higher than those without. Specifically, the average Accuracy, F1 Score, and MCC are 0.874, 0.874, and 0.748 with Optuna; 0.858, 0.855, and 0.736 with Hyperopt; and 0.854, 0.851, and 0.731 with Spearmint. This highlights the effectiveness of GWO in improving model performance. Figure 3 presents the accuracy performance of various models optimized using Optuna, Hyperopt, and Spearmint with GWO-based feature selection.

Table 5. *Model Performance without Feature Selection*

Models	**Optuna**			**Hyperopt**			**Spearmint**		
	Accuracy	**F1**	**MCC**	**Accuracy**	**F1**	**MCC**	**Accuracy**	**F1**	**MCC**
CatBoost	0.823	0.821	0.720	0.858	0.855	0.740	0.812	0.810	0.729
LDA	0.860	0.858	0.746	0.832	0.830	0.725	0.802	0.800	0.712
GB	0.843	0.842	0.723	0.835	0.832	0.730	0.804	0.802	0.717
KNN	0.809	0.807	0.699	0.779	0.777	0.672	0.783	0.780	0.686
RF	0.812	0.811	0.711	0.810	0.808	0.707	0.794	0.792	0.698
Avg.	0.829	0.828	0.720	0.823	0.820	0.715	0.799	0.797	0.708
Std. Dev.	0.0194	0.0194	0.0155	0.0267	0.0263	0.0239	0.0098	0.0102	0.0150
VEC	0.817	0.815	0.736	0.795	0.792	0.710	0.790	0.788	0.703
SEC	0.828	0.826	0.746	0.800	0.796	0.715	0.795	0.792	0.710

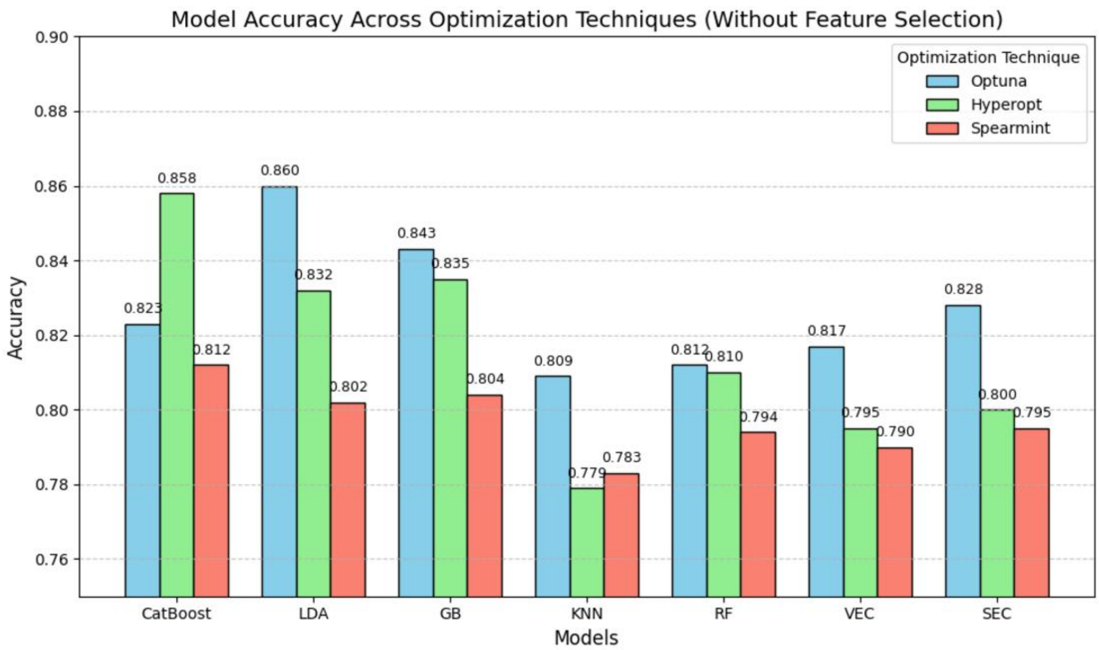

Figure 2. *Model Accuracy Across Optimization Techniques Without Feature Selection*

Table 6. *Model Performance with GWO-based Feature Selection and Hyperparameter Optimization*

Models	Optuna			Hyperopt			Spearmint		
	Accuracy	**F1**	**MCC**	**Accuracy**	**F1**	**MCC**	**Accuracy**	**F1**	**MCC**
CatBoost	0.883	0.883	0.767	0.873	0.869	0.754	0.873	0.868	0.754
LDA	0.878	0.878	0.760	0.855	0.851	0.742	0.855	0.852	0.742
GB	0.878	0.878	0.756	0.861	0.859	0.737	0.855	0.852	0.731
KNN	0.862	0.862	0.725	0.849	0.844	0.718	0.842	0.839	0.709
RF	0.867	0.867	0.734	0.854	0.850	0.728	0.846	0.843	0.719
Avg.	0.874	0.874	0.748	0.858	0.855	0.736	0.854	0.851	0.731
Std. Dev.	0.0087	0.0087	0.0180	0.0092	0.0097	0.0137	0.0119	0.0112	0.0179
VEC	0.873	0.873	0.745	0.858	0.855	0.733	0.850	0.847	0.724
SEC	0.878	0.878	0.757	0.865	0.860	0.748	0.868	0.865	0.750

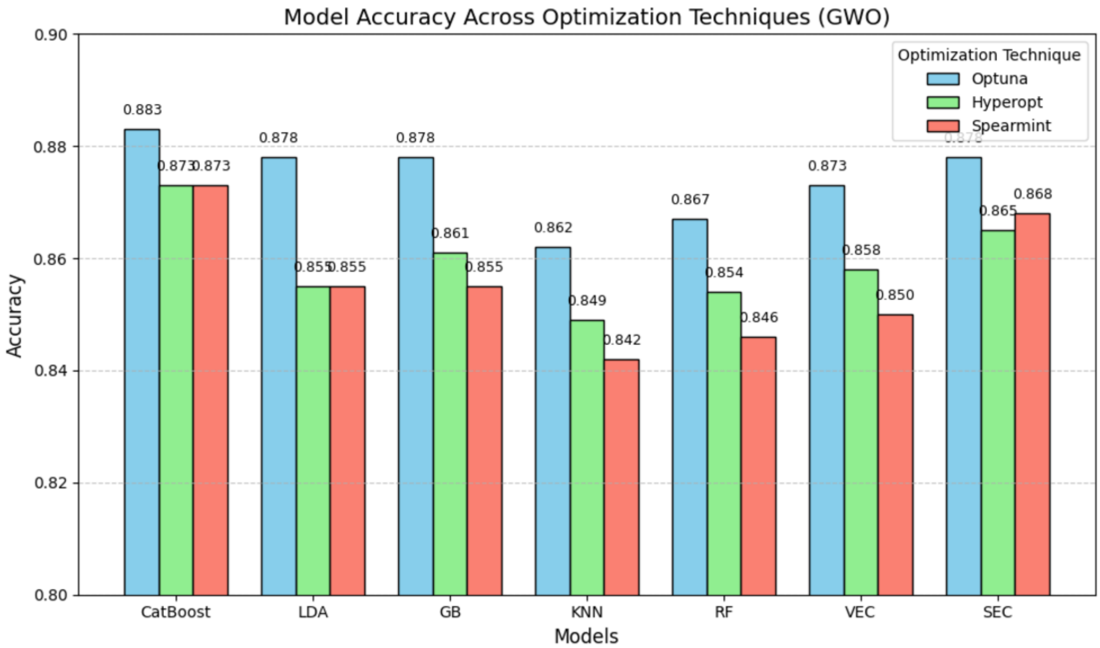

Figure 3. *Model Accuracy Across Optimization Techniques Using GWO-Based Feature Selection*

Table 7 presents the performance metrics of various models optimized using WOA with three different hyperparameter tuning methods. Among the models assessed, CatBoost with Optuna stands out with an Accuracy of 0.859, an F1 Score of 0.860, and an MCC of 0.759. This indicates that CatBoost, when optimized with WOA and Optuna, delivers superior performance compared to other models and tuning methods. Comparing these results with those from Table 6, which showcases models optimized with GWO, the performance metrics with WOA are slightly lower. For instance, CatBoost's performance with WOA and Optuna (Accuracy of 0.859, F1 Score of 0.860, MCC of 0.759) is marginally lower than its performance with GWO and Optuna (Accuracy of 0.883, F1 Score of 0.883, MCC of 0.767). Similarly, other models such as LDA and GradientBoost also show a reduction in performance with WOA compared to GWO. The average performance metrics with WOA are slightly lower than those achieved with GWO. This highlights that while WOA still provides strong results, GWO tends to deliver slightly better performance across the models evaluated. Figure 4 illustrates the accuracy results of models optimized with Optuna, Hyperopt, and Spearmint under WOA-based feature selection.

Table 7. *Model Performance with WOA-based Feature Selection and Hyperparameter Optimization*

Models	**Optuna**			**Hyperopt**			**Spearmint**		
	Accuracy	**F1**	**MCC**	**Accuracy**	**F1**	**MCC**	**Accuracy**	**F1**	**MCC**
CatBoost	0.859	0.860	0.759	0.846	0.847	0.742	0.791	0.794	0.732
LDA	0.872	0.874	0.754	0.857	0.859	0.742	0.797	0.799	0.716
GB	0.869	0.871	0.751	0.855	0.857	0.738	0.803	0.805	0.725
KNN	0.851	0.854	0.710	0.837	0.840	0.688	0.788	0.790	0.675
RF	0.854	0.856	0.724	0.842	0.844	0.702	0.785	0.788	0.695
Avg.	0.861	0.863	0.739	0.847	0.849	0.722	0.792	0.795	0.708
Std. Dev.	0.0091	0.0090	0.0214	0.0085	0.0082	0.0255	0.0072	0.0069	0.0233
VEC	0.866	0.868	0.735	0.855	0.858	0.725	0.814	0.816	0.720
SEC	0.871	0.872	0.752	0.855	0.856	0.740	0.804	0.805	0.724

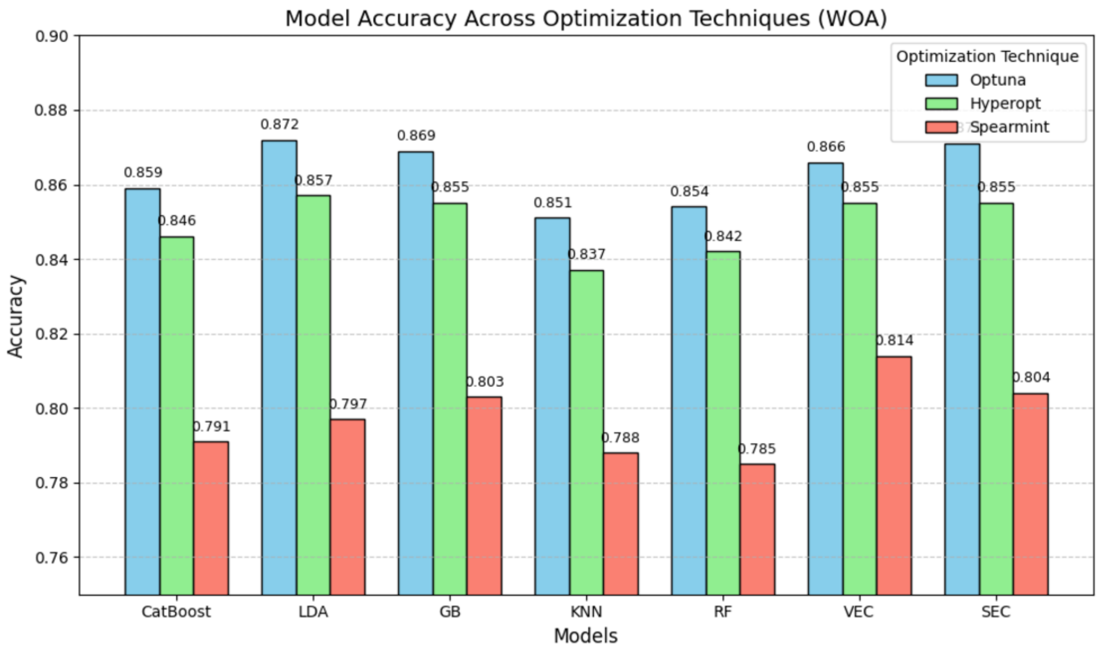

Figure 4. *Model Accuracy Across Optimization Techniques Using WOA-Based Feature Selection*

Figure 5 illustrates the comparison of Optuna accuracy scores across different feature selection techniques applied to various machine learning models. The feature selection methods include no feature selection (Without FS), Grey Wolf Optimization-based feature selection (GWO-based FS), and Whale Optimization Algorithm-based feature selection (WOA-based FS). Among the models, the highest accuracy was observed with CatBoost under the GWO-based FS approach, achieving a value of 0.883. Similarly, high accuracy was recorded for SEC and GB models under GWO-based FS, both reaching values of 0.878. In comparison, the WOA-based FS technique also produced competitive results, such as 0.872 for the GB model and 0.871 for the SEC model. Models without feature selection consistently demonstrated lower accuracy across all cases, with the KNN model yielding the lowest accuracy of 0.809. This indicates that the application of optimization-based feature selection methods significantly enhances model performance compared to scenarios without feature selection.

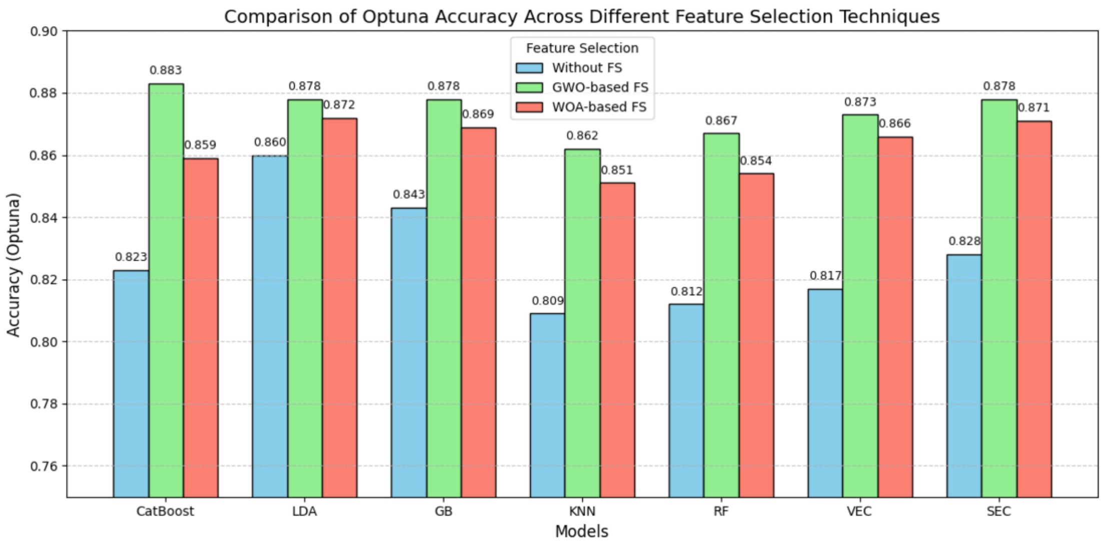

Figure 5. *Comparison of Optuna Accuracy Across Different Feature Selection Techniques*

4. DISCUSSION

The findings from this study provide a comprehensive analysis of the impact of hyperparameter optimization and feature selection techniques on the performance of various machine learning models. The results demonstrate that both hyperparameter optimization and feature selection significantly contribute to enhancing model performance, with some variations depending on the specific algorithms and optimization methods employed.

The first stage of the study, which analyzed model performance without feature selection and hyperparameter optimization, highlighted CatBoost as the best-performing model across all metrics. However, the results indicated considerable room for improvement with the application of optimization techniques. In the second stage, applying hyperparameter optimization methods without feature selection resulted in notable performance improvements, with LDA combined with Optuna yielding the highest accuracy. This suggests that LDA is particularly well-suited to benefit from fine-tuning using the Optuna framework. Significant gains observed in CatBoost's performance across all three optimization methods further underscore the model's adaptability. Optuna demonstrated consistent effectiveness, as evidenced by its lower standard deviation in performance metrics compared to Hyperopt and Spearmint. Recent findings in the literature highlight Optuna's efficient hyperparameter optimization capabilities, particularly in balancing exploration and exploitation during the search process (Yu & Zhu, 2020). Unlike traditional approaches like random or grid search, Optuna leverages an adaptive search mechanism using Tree-structured Parzen Estimators (TPE) and early stopping, which dynamically adjusts the search space based on previous iterations, optimizing computational resources (Akiba et al., 2019). Optuna’s pruning mechanism, which discards suboptimal trials early, further enhances its efficiency and robustness, particularly for high-dimensional datasets (Hassanali et al., 2024). Comparatively, Hyperopt and Spearmint, while effective, exhibit higher variability in performance metrics. Hyperopt’s reliance on TPE without adaptive pruning and Spearmint’s use of Gaussian Processes for optimization can lead to challenges in scalability and consistency in high-dimensional search spaces, resulting in higher variance in results (Chen et al., 2022; Tørring & Elster, 2022).

The third stage, which evaluated the combination of feature selection and hyperparameter optimization using the GWO method, showed that GWO is highly effective for improving model performance. CatBoost optimized with GWO and Optuna achieved the highest overall metrics, surpassing results from earlier stages, highlighting GWO's ability to enhance both feature selection and hyperparameter tuning to maximize machine learning model performance. The effectiveness of GWO stems from its nature-inspired algorithm, which mimics the hunting behavior of grey wolves, balancing exploration and exploitation in complex search spaces (Hatta et al., 2019). This dynamic search capability helps prevent premature convergence to local optima, a common issue in optimization tasks. Recent studies have highlighted GWO's advantages over traditional algorithms like Genetic Algorithms (GA) and Particle Swarm Optimization (PSO), particularly in feature selection and parameter tuning tasks (Seyyedabbasi & Kiani, 2021). By integrating feature selection, GWO not only identifies the most relevant features but also fine-tunes model hyperparameters, leading to significant improvements in predictive performance. This dual approach aligns with findings that metaheuristic algorithms like GWO often outperform traditional methods, especially in high-dimensional datasets where overfitting risks are greater (Shen & Zhang, 2022). Therefore, GWO provides a robust and efficient framework for optimizing machine learning models, confirming its potential for advanced data science applications.

The fourth stage, which assessed the impact of the WOA on hyperparameter optimization methods, further underscored the critical role of feature selection when combined with optimization strategies. Although WOA contributed to improved model performance, the results were generally lower than those achieved with the GWO method. The differences in performance between WOA and GWO can be attributed to the unique optimization strategies and the exploration-exploitation mechanisms inherent in each algorithm. WOA mimics the bubble-net hunting strategy of humpback whales, emphasizing a more stochastic exploration of the search space, which may lead to broader but less precise search outcomes (Mirjalili & Lewis, 2016). In contrast, GWO, which is inspired by the social hierarchy and hunting behavior of grey wolves, has been shown to maintain a more balanced trade-off between exploration and exploitation, allowing for more refined convergence to optimal solutions (Mirjalili et al., 2014). The slightly inferior performance of WOA compared to GWO may also reflect WOA's potential limitations in navigating the optimization landscape for certain types of datasets and models, particularly when complex interactions among features are present (Qaraad et al., 2022). Studies have demonstrated that while WOA can be highly effective for certain optimization tasks, its performance may be less consistent across diverse problem domains than GWO, which tends to offer a more stable optimization process due to its dynamic adaptation to the search environment (Wang et al., 2021). These findings suggest that while WOA offers valuable improvements, GWO may provide a more reliable approach for achieving optimal performance across a range of models and datasets.

5. CONCLUSION

This study investigated the impact of various hyperparameter optimization techniques and feature selection methods on the performance of machine learning models for machine failure prediction. The results demonstrate that integrating feature selection with hyperparameter optimization significantly enhances predictive performance. Among the evaluated methods, GWO paired with Optuna emerged as the most effective combination. Specifically, the CatBoost model optimized with GWO and Optuna achieved the highest performance, with an accuracy of 88.3%, an F1 score of 88.3%, and an MCC of 76.7%. LDA also showed considerable improvement, with its accuracy increasing from 86% without feature selection to 87.8% with GWO-based feature selection. Similarly, GB and SEC models achieved accuracy levels of 87.8% and 87.8%, respectively, under the GWO approach. In comparison, the WOA produced competitive results, with CatBoost achieving an accuracy of 85.9% and an MCC of 75.9%, though it consistently underperformed relative to GWO. These differences are attributed to the contrasting exploration-exploitation strategies of GWO and WOA. Overall, the findings confirm that GWO, particularly when combined with Optuna, provides a solid and effective framework for optimizing machine learning models in predictive maintenance tasks, offering a pathway to enhance industrial reliability and operational efficiency.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

REFERENCES

Abdallah, M., Lee, W. J., Raghunathan, N., Mousoulis, C., Sutherland, J. W., & Bagchi, S. (2021). Anomaly detection through transfer learning in agriculture and manufacturing IoT systems. https://doi.org/10.48550/arXiv.2102.05814

Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019, July). *Optuna: A next-generation hyperparameter optimization framework*. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining (pp. 2623-2631). https://doi.org/10.1145/3292500.3330701

Archetti, F., & Candelieri, A. (2019). Software Resources. In: *Bayesian Optimization and Data Science* (pp. 97-109). https://doi.org/10.1007/978-3-030-24494-1_6

Ayvaz, S., & Alpay, K. (2021). Predictive maintenance system for production lines in manufacturing: A machine learning approach using IoT data in real-time. *Expert Systems with Applications*, *173*, 114598. https://doi.org/10.1016/j.eswa.2021.114598

Campos, J. R., Costa, E., & Vieira, M. (2019). Improving failure prediction by ensembling the decisions of machine learning models: a case study. *IEEE Access*, *7*, 177661-177674. https://doi.org/10.1109/ACCESS.2019.2958480

Celikmih, K., Inan, O., & Uguz, H. (2020). Failure prediction of aircraft equipment using machine learning with a hybrid data preparation method. *Scientific Programming*, *2020*(1), 8616039. https://doi.org/10.1155/2020/8616039

Chen, T., Chen, X., Chen, W., Wang, Z., Heaton, H. W., Liu, J., & Yin, W. (2022). Learning to optimize: A primer and a benchmark. *The Journal of Machine Learning Research*, *23*(1), 8562-8620.

El-Kenawy, E.-S., & Eid, M. (2020). Hybrid gray wolf and particle swarm optimization for feature selection. *International Journal of Innovative Computing, Information and Control*, *16*(3), 831-844. http://doi.org/10.24507/ijicic.16.03.831

Hassanali, M., Soltanaghaei, M., Javdani Gandomani, T., & Zamani Boroujeni, F. (2024). Software development effort estimation using boosting algorithms and automatic tuning of hyperparameters with Optuna. *Journal of Software: Evolution and Process*, *36*(9), e2665. https://doi.org/10.1002/smr.2665

Hatta, N. M., Zain, A. M., Sallehuddin, R., Shayfull, Z., & Yusoff, Y. (2019). Recent studies on optimisation method of Grey Wolf Optimiser (GWO): a review (2014–2017). *Artificial Intelligence Review*, *52*, 2651-2683. https://doi.org/10.1007/s10462-018-9634-2

Jin, X., Chen, Y., Wang, L., Han, H., & Chen, P. (2021). Failure prediction, monitoring and diagnosis methods for slewing bearings of large-scale wind turbine: A review. *Measurement*, *172*, 108855. https://doi.org/10.1016/j.measurement.2020.108855

Khaire, U. M., & Dhanalakshmi, R. (2022). Stability investigation of improved whale optimization algorithm in the process of feature selection. *IETE Technical Review*, *39*(2), 286-300. https://doi.org/10.1080/02564602.2020.1843554

Lasotte, Y. B., Garba, E. J., Malgwi, Y. M., & Buhari, M. A. (2022). An ensemble machine learning approach for fake news detection and classification using a soft voting classifier. *European Journal of Electrical Engineering and Computer Science*, *6*(2), 1-7. https://doi.org/10.24018/ejece.2022.6.2.409

Leukel, J., González, J., & Riekert, M. (2021). Adoption of machine learning technology for failure prediction in industrial maintenance: A systematic review. *Journal of Manufacturing Systems*, *61*, 87-96. https://doi.org/10.1016/j.jmsy.2021.08.012

Li, J., King, S., & Jennions, I. (2023). Intelligent fault diagnosis of an aircraft fuel system using machine learning—A literature review. *Machines*, *11*(4), 481. https://doi.org/10.3390/machines11040481

Luo, G. (2016). A review of automatic selection methods for machine learning algorithms and hyper-parameter values. *Network Modeling Analysis in Health Informatics and Bioinformatics*, *5*, 18. https://doi.org/10.1007/s13721-016-0125-6

Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. *Advances in engineering software*, *95*, 51-67. https://doi.org/10.1016/j.advengsoft.2016.01.008

Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in engineering software*, *69*, 46-61. https://doi.org/10.1016/j.advengsoft.2013.12.007

Mishra, K., & Manjhi, S. K. (2018, November). *Failure prediction model for predictive maintenance*. In: 2018 IEEE International Conference on Cloud Computing in Emerging Markets (CCEM) (pp. 72-75). https://doi.org/10.1109/CCEM.2018.00019

Mohammed, B., Awan, I., Ugail, H., & Younas, M. (2019). Failure prediction using machine learning in a virtualised HPC system and application. *Cluster Computing*, *22*, 471-485. https://doi.org/10.1007/s10586-019-02917-1

Parra-Ullauri, J., Zhang, X., Bravalheri, A., Nejabati, R., & Simeonidou, D. (2023, March). *Federated Hyperparameter Optimisation with Flower and Optuna*. In: Proceedings of the 38th ACM/SIGAPP Symposium on Applied Computing (pp. 1209-1216). https://doi.org/10.1145/3555776.3577847

Pellegrini, A., Di Sanzo, P., & Avresky, D. R. (2015, May). *A machine learning-based framework for building application failure prediction models*. In: 2015 IEEE International Parallel and Distributed Processing Symposium Workshop (pp. 1072-1081). https://doi.org/10.1109/IPDPSW.2015.110

Ponemon Institute. (2011, February). *Cost of data center outages*. PDF

Qaraad, M., Amjad, S., Hussein, N. K., & Elhosseini, M. A. (2022). Large scale salp-based grey wolf optimization for feature selection and global optimization. *Neural Computing and Applications*, *34*(11), 8989-9014. https://doi.org/10.1007/s00521-022-06921-2

Rajadurai, H., & Gandhi, U. D. (2022). A stacked ensemble learning model for intrusion detection in wireless network. *Neural computing and applications*, *34*(18), 15387-15395. https://doi.org/10.1007/s00521-020-04986-5

Rzayeva, L., Myrzatay, A., Abitova, G., Sarinova, A., Kulniyazova, K., Saoud, B., & Shayea, I. (2023). Enhancing LAN Failure Predictions with Decision Trees and SVMs: Methodology and Implementation. *Electronics*, *12*(18), 3950. https://doi.org/10.3390/electronics12183950

Seyyedabbasi, A., & Kiani, F. (2021). I-GWO and Ex-GWO: improved algorithms of the Grey Wolf Optimizer to solve global optimization problems. *Engineering with Computers*, *37*(1), 509-532. https://doi.org/10.1007/s00366-019-00837-7

Shen, C., & Zhang, K. (2022). Two-stage improved Grey Wolf optimization algorithm for feature selection on high-dimensional classification. *Complex & Intelligent Systems*, *8*(4), 2769-2789. https://doi.org/10.1007/s40747-021-00452-4

Tørring, J. O., & Elster, A. C. (2022, May). *Analyzing search techniques for autotuning image-based gpu kernels: The impact of sample sizes*. In: 2022 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW) (pp. 972-981). https://doi.org/10.1109/IPDPSW55747.2022.00155

Tweney, D. (2013, August). *Amazon website goes down for 40 minutes, costing the company $5 million*. VentureBeat. URL

Wahid, A., Breslin, J. G., & Intizar, M. A. (2022). Prediction of machine failure in industry 4.0: a hybrid CNN-LSTM framework. *Applied Sciences*, *12*(9), 4221. https://doi.org/10.3390/app12094221

Wang, Z., Qin, C., Wan, B., & Song, W. W. (2021). A comparative study of common nature-inspired algorithms for continuous function optimization. *Entropy*, *23*(7), 874. https://doi.org/10.3390/e23070874

Wang, Z., Zhang, M., Wang, D., Song, C., Liu, M., Li, J., Lou, L., & Liu, Z. (2017). Failure prediction using machine learning and time series in optical network. *Optics Express*, *25*(16), 18553-18565. https://doi.org/10.1364/OE.25.018553

Young, M. T., Hinkle, J., Ramanathan, A., & Kannan, R. (2018, September). *Hyperspace: Distributed bayesian hyperparameter optimization*. In: 2018 30th International Symposium on Computer Architecture and High Performance Computing *(SBAC-PAD)* (pp. 339-347). https://doi.org/10.1109/CAHPC.2018.8645954

Yu, T., & Zhu, H. (2020). Hyper-parameter optimization: A review of algorithms and applications. https://doi.org/10.48550/arXiv.2003.05689

Zonta, T., Da Costa, C. A., da Rosa Righi, R., de Lima, M. J., da Trindade, E. S., & Li, G. P. (2020). Predictive maintenance in the Industry 4.0: A systematic literature review. *Computers & Industrial Engineering*, *150*, 106889. https://doi.org/10.1016/j.cie.2020.106889