# Feature Reduction in Random Forests with SHAP-Based Importance Scores*

Wiktoria Plechta[1]

[1]Faculty of Applied Mathematics, Silesian University of Technology, Kaszubska 23, 44100 Gliwice, Poland

**Abstract**

This paper investigates whether SHAP (SHapley Additive exPlanations) values can be useful for feature selection in Random Forest models. Using SHAP-based elimination, we aim to make models less resource-intensive. We conduct experiments across three binary classification datasets, progressively removing features with the lowest SHAP values and monitoring changes in accuracy, training time, and memory usage during training. Additionally, we analyze the stability of SHAP rankings throughout the feature reduction process and compare them with single-feature model performance. The results show that it is possible to significantly reduce resource consumption and in some cases even slightly improve performance by removing features with low SHAP values. In conclusion, SHAP-based feature selection offers a promising approach to building more efficient and interpretable tree-based models. We also identify directions for future research, including applications in multi-class settings and deeper analysis of redundant or interacting features.

**Keywords**

RandomForest, SHAP, FeatureSelection

## 1. Introduction

In recent years, learning methods have become widely used due to their high predictive performance. Examples are neural networks [1, 2], and fuzzy models [3, 4] or clustering algorithms and heuristics [5, 6]. Random Forests are one of them. They achieve good results in medicine [7, 8], energy technology [9, 10], businesses [11] classification or regression problems. However, despite their effectiveness, Random Forests are often considered "black box" models, making it difficult to understand how individual features influence predictions. This lack of interpretability presents challenges in areas where model transparency is crucial, such as healthcare, finance, or policymaking.

To address these challenges, model explainability techniques like SHAP (SHapley Additive exPlanations) [12] have been developed. SHAP assigns importance scores to features by estimating their contribution to the model's predictions, using principles from Shapley values in cooperative game theory. By providing a consistent and theoretically sound measure of feature impact, SHAP offers a powerful tool for interpreting complex models.

In this study, we explore whether SHAP values can be used not only for interpretation but also for feature selection. Specifically, we examine how removing low-importance features affects the performance of a Random Forest model. Our goal is to determine whether SHAP-based feature reduction can maintain or even improve performance while simplifying the model and reducing computational costs. Additionally, we investigate whether there are "safe" SHAP thresholds below which features can be removed with little to no negative effect on performance. We also compare SHAP-based feature importance to the accuracy achieved by training the model on individual features alone.

Interestingly, our results show that in some datasets, certain single features can achieve surprisingly high accuracy, comparable to the full feature set. However, this observation is nuanced: while one feature may perform well in isolation, retaining a small group of features can help stabilize predictions and increase model robustness, especially under distributional shifts. In the end, this paper aims to demonstrate how SHAP can serve not only as a tool for interpretability but also as a practical guide for model simplification and optimization.

*IVUS 2025: Information Society and University Studies 2025, May 15, Kaunas, Lithuania

wp311004@student.polsl.pl (W. Plechta)

## 2. Methodology

### 2.1. Random Forest

The Random Forest algorithm [13] is a well-known supervised learning method used for classification and regression tasks. It creates multiple decision trees during training and returns the most frequent class for classification or the average prediction for regression. It was chosen as the base model due to strong predictive performance, robustness to overfitting, and ability to handle various types of data with minimal preprocessing. As a tree-based ensemble, it is compatible with SHAP, providing an efficient and accurate computation of SHAP values. Additionally, Random Forest provides native feature importance measures, which gives room for further research. These properties make it an ideal model for studying the impact of feature reduction on both interpretability and performance.

### 2.2. SHapley Additive exPlanations

To interpret feature importance in Random Forest models, we apply SHAP (SHapley Additive exPlanations) values. SHAP provides mathematical way to explain how much each feature contributes to the final prediction made by a model. It works by fairly distributing the "credit" for a prediction among all input features, based on how each one influences the output when combined with others.

In this study, we use Performance SHAP instead of the standard Tree SHAP algorithm. Performance SHAP is a variant designed to evaluate feature importance based on changes in the model's performance metric (here accuracy) when specific features are marginalized, permuted, or otherwise manipulated. This allows for a broader, model-agnostic assessment of feature relevance, potentially capturing interactions and dependencies that Tree SHAP might not fully account for. Performance SHAP value for feature $i$ on instance $j$ can be expressed as:

$$\phi_i^{(j)} = \mathcal{M}(S \cup \{i\}, j) - \mathcal{M}(S, j) \tag{1}$$

where:

- $S \subseteq F \setminus \{i\}$ is a subset of features excluding feature $i$,
- $F$ is the set of all features,
- $\mathcal{M}(S, j)$ represents the model's performance for instance $j$ when only features in $S$ are available.

The SHAP values are computed across the entire training dataset. To obtain a global measure of feature importance, we aggregate the absolute SHAP values for each feature across all instances. Specifically, for a feature $i$, its overall importance score $I_i$ is given by:

$$I_i = \frac{1}{N} \sum_{j=1}^{N} |\phi_i^{(j)}| \tag{2}$$

where $\phi_i^{(j)}$ denotes the SHAP value of feature $i$ for instance $j$ and $N$ is the total number of instances in the dataset.

This aggregated importance $I_i$ is subsequently used to identify features with low predictive relevance, which are candidates for removal in the feature selection step.

### 2.3. Datasets

For the purpose of this article, we used 3 datasets with binary classification:

- Machine Failure (dataset A) - contains sensor data collected from various machines. Consists of 10 columns total, with 'fail' column being the target variable. The data originate from Kaggle[1] and are licensed under the Apache License 2.0.

[1] https://www.kaggle.com/datasets/umerrtx/machine-failure-prediction-using-sensor-data

- Manufacturing Defects (dataset B) - Factors Influencing Defect Rates in a Manufacturing Environment. Consists of 17 columns total, with 'DefectStatus' being the target variable. The data originate from Kaggle[2] and are licensed under Attribution 4.0 International (CC BY 4.0). [3]
- Company Bankrupt Prediction (dataset C) - contains data collected from the Taiwan Economic Journal for the years 1999 to 2009. 96 columns total, with 'Bancrupt?' being the target variable. The data originate from Kaggle[4].

### 2.4. Feature Selection

For feature selection, we calculate the mean absolute SHAP value for each feature and then remove the feature with the lowest value, one at a time for dataset A. For datasets B and C, we first remove features with significantly lower SHAP values, followed by removing one or two features at a time. This approach allows us to observe if there are any "sweet spots" where the model achieves the best accuracy.

### 2.5. Performance Comparison

We compare the baseline and reduced models by evaluating their accuracy, training time and memory usage during training. To gain deeper insights into the impact of feature removal, we also track how SHAP values evolve as features are progressively eliminated. This helps us understand not only the overall model performance but also how the importance of remaining features may shift during the feature selection process.

## 3. Experiments

### 3.1. Experimental Setup

All experiments were conducted using a standard 80/20 train-test split, ensuring that each dataset was divided into 80% for training and 20% for evaluation. To maintain consistency across all experiments, the same random seed was used during data shuffling and splitting. We used raw, non normalized data.

For the classification model, we employed a Random Forest classifier.The model was configured with the following parameters:

- **Number of trees:** 20
- **Maximum tree depth:** 10
- **Minimum samples per leaf:** 2
- **Criterion:** Gini impurity (default)
- **Bootstrap sampling:** Enabled
- **Random state:** Fixed

### 3.2. Results

In this section, we present the outcomes of our SHAP-guided feature elimination and individual feature performance analysis across three binary classification datasets. The results are structured into three parts: (1) feature elimination impact on model metrics, (2) changes in SHAP rankings during feature removal, and (3) comparison of SHAP importance with single-feature predictive power.

#### Feature Elimination and Model Performance

The following tables summarize the changes in classification accuracy, training time, and memory usage during training as features were progressively removed based on SHAP values. Each row corresponds to a different stage of the feature elimination process. Training time and memory usage values are averaged over five independent runs to account for variability and ensure reliable measurements.

[2] https://www.kaggle.com/datasets/rabieelkharoua/predicting-manufacturing-defects-dataset
[3] https://creativecommons.org/licenses/by/4.0/
[4] https://www.kaggle.com/datasets/fedesoriano/company-bankruptcy-prediction

**Table 1**
Performance metrics during SHAP-based feature elimination – Dataset A

| Remaining Features | Accuracy | Train Time (s) | Memory (MB) |
|---|---|---|---|
| All (9) | 91.0% | 1.76 | 0.67 |
| 8 | 89.9% | 1.64 | 0.65 |
| 7 | 89.4% | 1.56 | 0.64 |
| 6 | 90.5% | 1.34 | 0.66 |
| 5 | 91.0% | 0.86 | 0.69 |
| 4 | 91.0% | 0.36 | 0.76 |
| 3 | 90.4% | 0.31 | 0.92 |
| 2 | 91.5% | 0.19 | 0.99 |
| 1 | 91.1% | 0.06 | 0.58 |

**Table 2**
Performance metrics during SHAP-based feature elimination – Dataset B

| Remaining Features | Accuracy | Train Time (s) | Memory (MB) |
|---|---|---|---|
| All (16) | 95.8% | 77.13 | 2.00 |
| 13 | 95.7% | 71.29 | 1.76 |
| 7 | 95.7% | 36.06 | 1.41 |
| 4 | 95.7% | 18.18 | 1.14 |
| 3 | 92.9% | 14.31 | 1.24 |
| 2 | 83.0% | 7.70 | 1.26 |
| 1 | 82.4% | 8.11 | 1.16 |

**Table 3**
Performance metrics during SHAP-based feature elimination – Dataset C

| Remaining Features | Accuracy | Train Time (s) | Memory (MB) |
|---|---|---|---|
| All (95) | 96.7% | 992.94 | 20.30 |
| 76 | 96.9% | 773.28 | 15.66 |
| 68 | 97.0% | 700.47 | 14.32 |
| 59 | 96.7% | 608.07 | 12.44 |
| 41 | 96.8% | 422.06 | 9.25 |
| 25 | 96.5% | 266.27 | 5.82 |
| 15 | 96.7% | 151.02 | 3.73 |
| 10 | 96.9% | 93.48 | 2.55 |
| 6 | 97.0% | 51.95 | 1.94 |
| 4 | 96.4% | 36.10 | 1.86 |
| 2 | 96.3% | 17.03 | 1.56 |
| 1 | 96.5% | 12.53 | 1.54 |

In all three datasets, reducing the number of features resulted in only minor changes in accuracy but led to noticeable improvements in training efficiency. Dataset A maintained stable accuracy (90–91%) even as features were removed, peaking at 91.5% with just two features. Training time decreased predictably, dropping from 1.76 s (9 features) to 0.06 s (1 feature). However, memory usage exhibited a non-linear pattern: it initially decreased but increased slightly when only 3–2 features remained (0.92 MB and 0.99 MB) before dropping again with a single feature (0.58 MB). The memory increase might occur because the model worked harder to find optimal splits when only a few important features remained, requiring more temporary storage.

Dataset B retained high accuracy (95.7%) until only 3 features remained, after which performance declined sharply (92.9% with 3 features, 83.0% with 2). Training time and memory usage improved steadily.

Dataset C proved to be the most resilient, with accuracy remaining within 96.3 to 97. 0% even with aggressive feature reduction. Notably, the highest accuracy (97.0%) occurred with just 6 features, suggesting redundancy among the original 95 features. Computational efficiency improved dramatically: training time fell from 992.94 to 12.53 seconds, and memory usage dropped from 20.30 to 1.54 MB. Interestingly, we also observed that all models performed surprisingly well even when using only a single feature as input. This raised the question of whether Random Forest classifiers are particularly effective at extracting predictive power from individual features. To explore this, we conducted additional experiments to evaluate the accuracy of models trained on each feature separately.

An important consideration arising from our analysis is whether retraining the Random Forest model after removing features deemed unimportant by SHAP values is worthwhile, especially when the current model already demonstrates satisfactory performance. Retraining can offer several advantages: it typically reduces computational costs during inference and data preprocessing, improves model interpretability by simplifying the feature set, and may enhance generalization by eliminating irrelevant or redundant features that contribute to overfitting. Additionally, models with fewer features tend to be easier and faster to update in the future. However, retraining process requires additional time and resources, which may not be justified if the existing model already meets practical requirements. Moreover, removing features can alter prediction distributions, potentially affecting downstream systems that rely on consistent outputs. Finally, SHAP values may not fully capture complex feature interactions, so removing features solely based on individual importance risks losing valuable information. Overall, if the model is intended for long-term deployment, and SHAP analysis shows that certain features can be removed without accuracy loss, retraining to optimize the model is generally advisable. In contrast, for exploratory analyses or when improvements are minimal, retaining the original model may be a reasonable choice.

### Dynamics of SHAP Rankings During Feature Elimination

The figures below illustrate how the SHAP importance values of features change during the elimination process. Each plot shows the relative ranking of features across reduction steps.

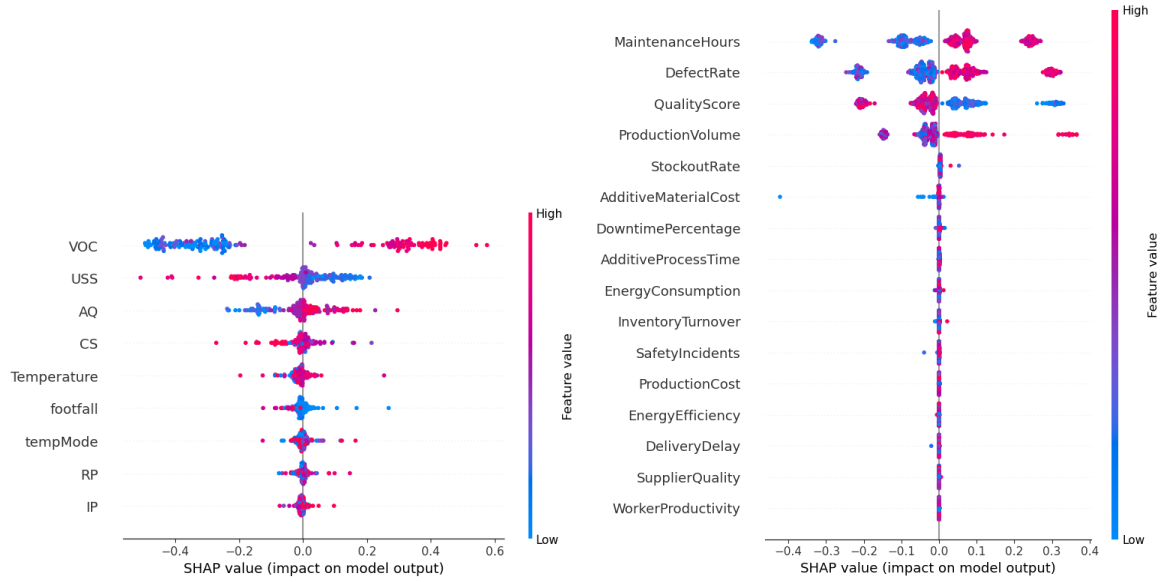


**Figure 1:** SHAP importance rankings for all features (datasets A and B).

SHAP rankings remained relatively stable, especially among the top-ranked features. However, we observed that features with lower initial importance scores were more likely to shift positions as others were removed. This suggests that while the most influential features tend to maintain their status, feature importance at the lower end of the ranking is more context sensitive. Relying solely on a

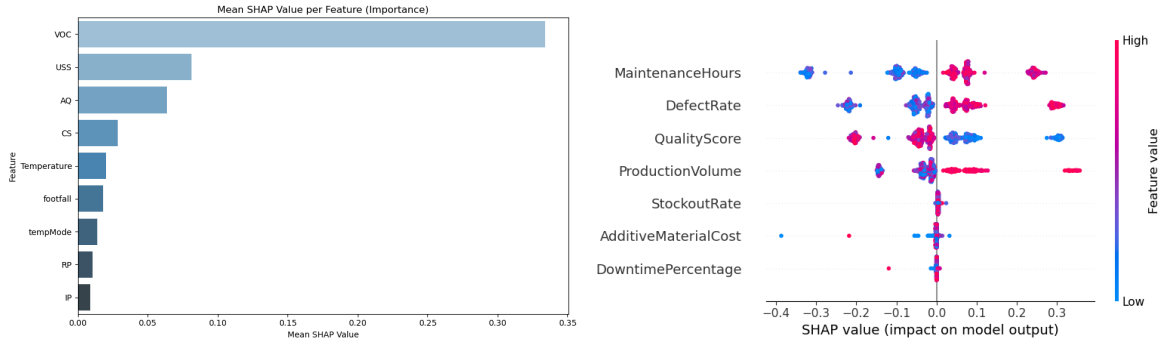


**Figure 2:** SHAP importance rankings for around 60% features (datasets A and B).

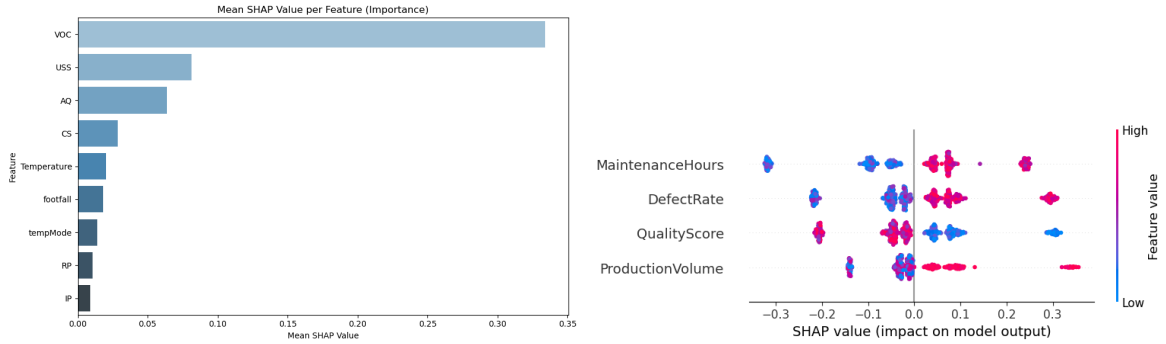


**Figure 3:** SHAP importance rankings for around 30% features (datasets A and B).

one-time ranking may risk overlooking subtle interactions or shifts in relevance that emerge as the feature set evolves.

### SHAP Importance vs. Single-Feature Predictive Power

In this part of the study, we examined the previously mentioned effectiveness of Random Forest models when using only a single feature. Additionally, we compared the classification performance of individual features with their corresponding SHAP values. These comparisons are visualized in the plots below.

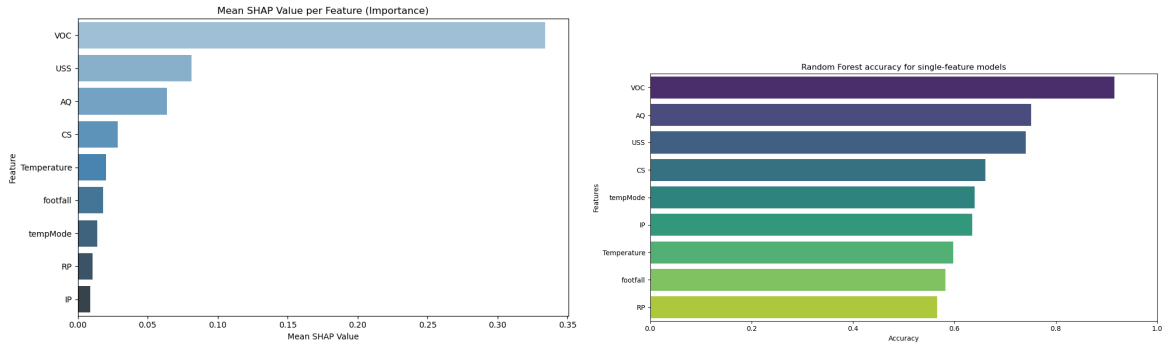


**Figure 4:** Dataset A SHAP importance ranking and single-feature model accuracy ranking.

For two out of the three datasets, the models achieved high accuracy using almost any single feature, demonstrating the strength of Random Forests in capturing useful patterns from minimal input. Interestingly, features with the highest individual accuracy had the highest SHAP values, which means that that SHAP-based ranking consistently pointed us toward the most informative features.

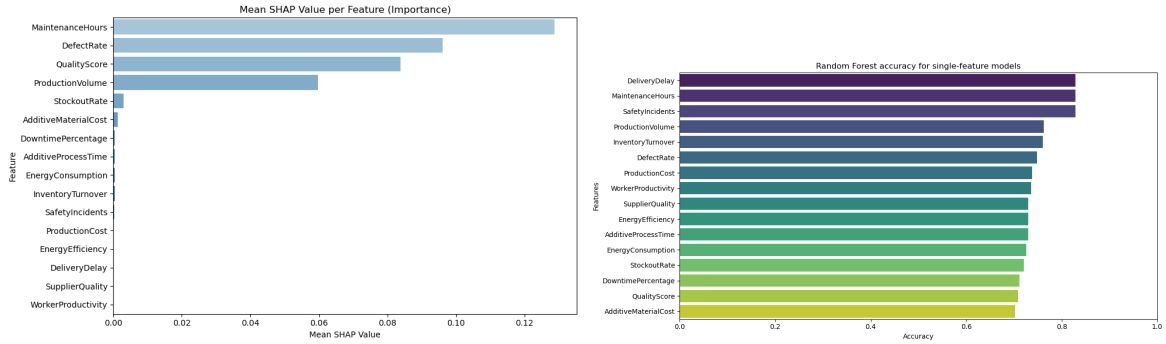


**Figure 5:** Dataset B SHAP importance ranking and single-feature model accuracy ranking.

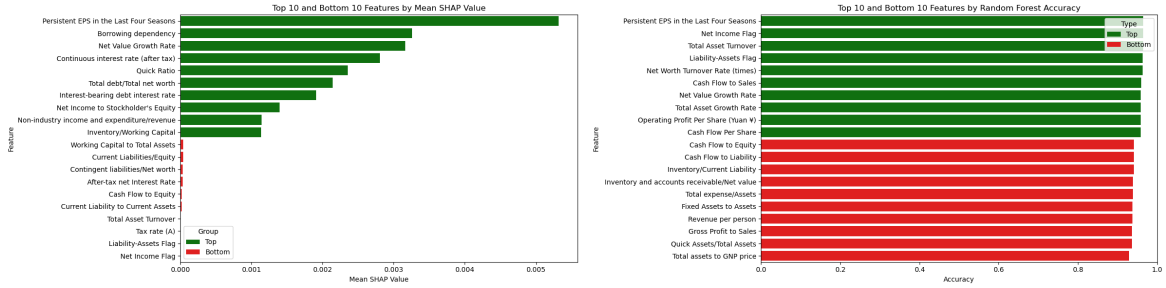


**Figure 6:** Dataset C SHAP importance ranking and single-feature model accuracy ranking (10 best and 10 worst ranked).

# 4. Conclusion

This study investigated the effects of SHAP-based feature selection on the performance and efficiency of Random Forest models. The experimental results indicate that eliminating features with low SHAP importance enables substantial model simplification while maintaining comparable predictive performance.

The results show that SHAP values offer a powerful and interpretable metric for ranking features by their contribution to model output. In many cases, a substantial portion of features could be eliminated with little to no loss in classification accuracy. This highlights SHAP's potential not only as an explanatory tool but also as a practical aid in dimensionality reduction and model optimization. Additionally, we found that while the top SHAP-ranked features tend to remain stable during the reduction process, features with lower importance often shift in rank as the feature set changes. This behavior suggests a degree of redundancy among less important features, highlighting the value of a more context-aware approach to feature selection rather than relying solely on a static ranking from the full feature set. We also looked at how individual features perform when used in isolation. In one of the datasets, every feature could achieve high accuracy score, and in the other two, a few features performed similarly well. This might suggest that certain features dominate the prediction process. It also points to the strength and flexibility of Random Forests, which seem capable of extracting useful patterns even from minimal input. However, relying on one or two features increases the risk of overfitting and reduces robustness to noise or distributional shifts. Our findings emphasize the importance of selecting a small, diverse set of complementary features to ensure stable and reliable performance. Several directions remain open for further exploration. One promising extension is to apply the proposed methodology to multi-class classification tasks, where feature interactions and importance may behave differently. Additionally, a deeper investigation into the dynamics of SHAP values—especially their stability and variability across training runs or under distributional shifts—could offer further insight into their reliability as a feature selection signal. In conclusion, the results underscore the potential of SHAP not only as an interpretability framework but also as a systematic tool for feature reduction and model

optimization. However, retraining after feature removal involves trade-offs, including resource costs and potential changes in model behavior. Careful consideration is needed before applying such optimization in practice. Taken together, our findings encourage a more dynamic and iterative approach to feature selection, one that considers not only static importance scores but also how those scores evolve during the pruning process. SHAP-based evaluation, when used thoughtfully, provides valuable insights into feature relevance, interaction, and redundancy—enabling practitioners to design more efficient and interpretable models without sacrificing accuracy.

## Declaration on Generative AI

The author has not employed any Generative AI tools.

## References


[1] D. Połap, A. Jaszcz, Decentralized medical image classification system using dual-input cnn enhanced by spatial attention and heuristic support, Expert Systems with Applications 253 (2024) 124343.

[2] A. Jaszcz, D. Połap, Aimm: Artificial intelligence merged methods for flood ddos attacks detection, Journal of King Saud University-Computer and Information Sciences 34 (2022) 8090–8101.

[3] K. Prokop, Feature fusion using deep learning modules and fuzzy c-means for medical disease recognition, in: 2025 IEEE International Conference on Fuzzy Systems (FUZZ), IEEE, 2025, pp. 1–6.

[4] A. Zielonka, A. Sikora, M. Woźniak, Trust mechanism fuzzy rules intelligent car real-time diagnostic system, in: 2023 IEEE International Conference on Fuzzy Systems (FUZZ), IEEE, 2023, pp. 1–8.

[5] T. Bury, A. Kacprzak, P. Żerdziński, Soft inference as a voting mechanism in k-nearest neighbors clustering algorithm, in: International Conference on Information and Software Technologies, Springer, 2023, pp. 309–318.

[6] M. Pleszczyński, Using selected heuristic algorithms in solving nonlinear differential equations, in: 2024 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2024, pp. 1–6.

[7] P. Dutta, S. Paul, A. Kumar, Chapter 25 - comparative analysis of various supervised machine learning techniques for diagnosis of covid-19, in: S. L. Tripathi, V. E. Balas, S. Mohapatra, K. B. Prakash, J. Nayak (Eds.), Electronic Devices, Circuits, and Systems for Biomedical Applications, Academic Press, 2021, pp. 521–540.

[8] R. Marimuthu, S. N. Shivappriya, M. N. Saroja, Chapter 14 - a study of machine learning algorithms used for detecting cognitive disorders associated with dyslexia, in: H. D. Jude (Ed.), Handbook of Decision Support Systems for Neurological Disorders, Academic Press, 2021, pp. 245–262.

[9] B. Grillone, S. Danov, A. Sumper, J. Cipriano, G. Mor, A review of deterministic and data-driven methods to quantify energy efficiency savings and to predict retrofitting scenarios in buildings, Renewable and Sustainable Energy Reviews 131 (2020) 110027.

[10] H. Wang, Y. Liu, B. Zhou, C. Li, G. Cao, N. Voropai, E. Barakhtenko, Taxonomy research of artificial intelligence for deterministic solar power forecasting, Energy Conversion and Management 214 (2020) 112909.

[11] J. Zhang, Impact of an improved random forest-based financial management model on the effectiveness of corporate sustainability decisions, Systems and Soft Computing 6 (2024) 200102.

[12] E. Mosca, F. Szigeti, S. Tragianni, D. Gallagher, G. Groh, Shap-based explanation methods: a review for nlp interpretability, in: Proceedings of the 29th international conference on computational linguistics, 2022, pp. 4593–4603.

[13] R. Iranzad, X. Liu, A review of random forest-based feature selection methods for data science education and applications, International Journal of Data Science and Analytics (2024) 1–15.