Assessing the Effectiveness of Feature Selection and Boosted Classifiers on Prediction of Chronic Kidney Disease, Thyroid Cancer, and Rice Species Classification. A Reproducibility Paper.

Griffith Potts
University College London
London, United Kingdom

Harry Mellett
University College London
London, United Kingdom

Kacper Buksa
University College London
London, United Kingdom

Landon Rutledge
University College London
London, United Kingdom

Abstract

This study aims to reproduce and critically evaluate the findings of a previous machine learning paper that used feature selection and boosted classifiers to predict Chronic Kidney Disease with high accuracy. Reproducing such studies is essential for validating their reliability, identifying potential biases or errors, and assessing whether the models generalize well to new, more complex datasets. By testing the original methodology on contemporary datasets, this project also examines the model's real-world applicability and robustness. Due to the dataset characteristics and lack of documentation of the original study, this reproducibility paper initially aimed for a degree of reproducibility consistent with either R2 or R3[1]. This paper follows the original methodology as closely as possible however experiments and variation with the imputation, information gain threshold and hyperparameter tuning were required. When reproducing the results on the CKD dataset [2], similar metrics were achieved to the original paper, with most models improving from using the reduced feature set. Similar to the original paper, standard and boosted decision trees presented the best metrics, further reinforcing the overall success at reproduction of results. The discussion highlights that while boosting improved Decision Trees, it led to potential instability and overfitting in linear models like AdaBoost-LR, particularly on small or imbalanced datasets. These findings emphasize the importance of dataset characteristics, transparent methodology, and model-specific tuning when evaluating generalizability and robustness in machine learning studies.

Keywords

Information Gain, Logistic Regression, Decision Tree, Support Vector Machine, AdaBoost, Accuracy, Precision, Sensitivity, F-Measure

1 Introduction

1.1 Background of the Original Paper

This paper focuses on effectively detecting Chronic Kidney Disease (CKD), utilizing the publicly available CKD dataset [2]. The study employs the Information Gain (IG) technique, which ranks and selects the most relevant features in the CKD dataset. The purpose of IG is to rank the influence of features that do not help to predict the target variable. The attributes that have the least amount of influence on the target variable can be removed from the training dataset in order to improve the performance of the algorithms in detecting CKD [3]. The machine learning models used in this study are Logistic Regression (LR), Decision Tree (DT), and Support Vector Machine (SVM). This study also boosts each model using the Adaptive Boosting (AdaBoost) technique. AdaBoost focuses on improving the accuracy of an ensemble [4]. The accuracy is improved by creating a refined classifier from a series of weak classifiers that enhances the classification performance of each model by reducing overfitting and improving the accuracy [3]. The three ensemble models produced are AdaBoost-LR, AdaBoost-SVM, and AdaBoost-DT. The three boosted classifiers are benchmarked against the standard LR, SVM, and DT models.

1.2 Results of the Original Paper

In terms of performance comparison between the different models, the study evaluates the impact of feature selection and AdaBoost on model performance using these metrics: accuracy, precision, sensitivity, and F-measure. The results shown in the study include the performance evaluation of all models trained on the full feature set versus the performance evaluation of all models using the reduced feature set (selected using IG). The AdaBoost-DT model was the best performing model, achieving a perfect performance (100% accuracy, precision, sensitivity, and F-measure). The study also offers a performance comparison between its AdaBoost-DT model and between other relevant CKD prediction models in recent literature, including: RF + LR, CHIRP Method, XGBoost, Cost-sensitive RF, and LSVM + SMOTE.

Some of the main findings in this study include: the impact of feature selection (IG), performance of boosted models versus standard models, and the best performing model - AdaBoost-DT. The findings in this study show that feature selection improved model accuracy and reduced computational complexity. The most significant CKD features were retained, while the less relevant features were removed. The Adaptive Boosted models consistently outperformed the standard models. For example, the standard DT model achieved 94% accuracy in classifying the reduced feature set, while the AdaBoost-DT model achieved 100% accuracy in classifying that same feature set. The AdaBoost-DT model was also shown to outperform the other CKD prediction models.

1.3 Purpose of Reproduction Study

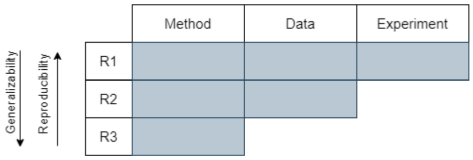
1.3.1 Objectives and Rationale. Reproducing machine learning (ML) studies is a fundamental practice to ensure the validity, reliability, and robustness of the proposed models and methodologies [5] First, the original results must be verified through following the methodology described in the study. This is to ensure the reported results are accurate, repeatable, and not coincidental. In some ML studies, results can be influenced by random initialization, dataset biases, or implementation-specific optimizations [6]. ML models should also work consistently across different environments, datasets, and implementations. A model that fails to reproduce well may indicate overfitting, improper validation procedures, or dataset-specific biases [6]. During the process of reproducing ML studies, potential errors or biases can be found and corrected. ML models can sometimes amplify biases present in datasets or contain hidden implementation errors that are missed by the researchers. Reproduction helps to detect such biases or errors by reevaluating model performance from different perspectives. Independent reproduction improves scientific rigor by ensuring that findings are not due to one-time cherry-picked results. In the study this group is looking at, the researchers claim their AdaBoost-DT model achieves a 100% score in each of the performance evaluation categories. Statements such as this in published papers should stand out, as it is very challenging to achieve 100% accuracy with a model. This group has will determine whether or not the reported results of 100% accuracy hold true while reproducing this study. Direct reproducibility is not possible with this study, as the code and minor steps in the methodology were not noted in the published study.

1.3.2 Evaluating Algorithms on Contemporary Datasets. There is also a focus on the replicability of this study, as the models and methods introduced in this study should be able to be applied to newer datasets. It is important for ML models to be tested on new datasets in order to ensure their relevance and real-world applicability, as well as to identify any flaws in the original study. For example, the original method applied by the researchers could have data leakage or variables could be improperly controlled, delivering misleading results [7]. The ML models also need to be tested to ensure generalization to unseen data. Models can overfit to specific datasets, which produces results that are seemingly positive. However, testing the model on a new dataset will test the generalization and how robust the model is in working with a new dataset [7]. In order for an ML model to truly be robust, it should be able to effectively operate on new, unseen datasets. This group looked to replicate the methodology introduced by the researchers, who used the CKD dataset, on two new datasets: UC Irvine's (UCI) Machine Learning Repository Rice (Cammeo and Osmancik) dataset (05/10/2019) and UCI's Differentiated Thyroid Cancer Recurrence dataset (30/10/2023). The CKD dataset is relatively small, so the group made sure to select at least one dataset that is larger than the CKD dataset. In selecting a larger dataset, this group will be able to simulate a more real-world examination of the model, as this more realistic data is often more diverse and noisier. Testing the model on a bigger dataset could potentially help identify bottlenecks and help optimize the model for efficiency. It is also important to test for bias in the original dataset, as some datasets used for ML research can have a demographic imbalance or under-represented categories [8]. By applying new datasets to these algorithms, it can be seen whether or not the algorithms do contain large amounts of bias. The model should perform fairly across different groups or datasets [8]. If the model does not perform fairly across different groups, misclassification can take place, leading to improper results from the model.

2 Methodology

2.1 Reproduction Strategy

2.1.1 Degrees of Reproducibility. As outlined by Semmelrock et al. [9], reproducibility in the area of health/life science is of critical importance, however data is provided often hard to access or contains many missing values. It is therefore important to understand the degree to which a study is able to reproduce a paper. Gunderson and Kjensmo [1] describe three degrees of reproducibility, R1, R2 and R3. R1 describes experiment reproducibility, where exactly the same results are produced, utilizing the same implementation techniques, hyperparameters, and software. R2, data reproducibility, requires that almost the same results are produced when the same data are used, but with a different implementation. For Method reproducibility (R3) to be achieved, both an alternative implementation of the methodology and different data can be used, but must produce the same results or findings. This is summarised in Figure 1.



	Method	Data	Experiment
R1	▨	▨	▨
R2	▨	▨	
R3	▨		

Figure 1: Degrees of Reproducibility. Taken from Semmelrock et al.

The original dataset that is used in this study, and its associated paper are discussed in this study, and are found to lack details surrounding hyperparameters, software used, and imputation techniques. Therefore, this reproducibility paper is limited to either an R2 or R3 degree of reproducibility.

2.1.2 Data Sources. The dataset used in the original study [2] is that of the CKD dataset sourced from the UC Irvine Machine Learning Repository. It consists of 24 features and 400 records, however each feature is missing a varying number of records throughout the dataset, which could prove problematic and require specific imputation techniques. An objective of this study is to test algorithms and technique robustness by applying these to alternative datasets. Although sharing a similar clinical context, the differential thyroid cancer recurrence dataset [10] presents challenges different from the original dataset. Containing 16 features and 383 records it is a dataset with a slightly lower dimensionality than the CKD dataset. This can be used to compare the effectiveness of information gain on data with an already low dimensionality. This dataset is highly unbalanced with 275 negative classifications and 108 positive classifications. While the CKD dataset is slightly unbalanced, the methodology will be tested by this highly unbalanced dataset.

Another alternative dataset was the Rice dataset [11]. This is centered around the classification of rice types (cammeo or osmancik) and presents the methodology with a completely different topic area. This dataset has a much lower dimensionality at 7 features; however, it has a much higher record count at 3810. This will test the methodology by assessing whether the information gain is still effective when applied to a dataset with an already low count of features. Furthermore, since the dataset requires no imputation and has a high record count, it will help to understand to what extent the method can perform with a dataset for which it was not initially designed.

2.1.3 *Pre-processing and imputation.*

The original dataset [2] required the most pre-processing and imputation as there were many missing values. The following cleaning process was applied to the CKD dataset:

(1) Convert .arrf file to a Pandas dataframe.
(2) Remove empty columns and records.
(3) Correct inconsistent entries, for example "\tnotCKD" to "notCKD".
(4) Convert "?" to NaN for missing values.
(5) Impute missing values.

Dataset imputation was not specified in the original study, so this study implemented its own methodology. Each missing value was identified under each feature column. If the feature was categorical, the missing values were replaced with the mode value. If the feature was numerical, the median value was used to replace the missing values. The original study [3] details that all categorical values were converted to binary and a Min-Max scaling applied to all data excluding binary types and 'age'. This was applied to all three datasets.

2.1.4 *Information Gain.*

Known as the curse of dimensionality, a dataset with high dimensionality can act counter-intuitively, where an increase in features can increase the noise and complexity of any given model [12]. Therefore, feature engineering is an important step in pre-processing to reduce complexity and increase performance. Information Gain (IG) is a feature reduction technique used in the original study [3]. IG measures the extent to which a feature reduces entropy, thus the ability to predict the target variable. This is computed as:

$$IG(X|Y) = H(X) - H(X|Y)$$

where X and Y are random variables, $H(X)$ the entropy for X and $H(X|Y)$ the conditional entropy for X and Y [3]. Once the IG values are computed for each feature, a threshold is set where any features below the threshold are filtered out creating a new reduced dataset. The original study set the threshold as 'pe', however it is noted that this is an arbitrary value. In order to apply IG to other datasets, an alternative methodology was needed to derive the threshold. Prasetiyowati, Maulidevi and Surendro [13] suggest a technique that uses the standard deviation of the IG values to set a threshold. This was implemented for the alternative datasets and compared against the results of the 'pe' threshold for the original dataset. The datasets were first split into a standard 70/30 train-test split. The reduced datasets were then created post-split.

2.1.5 *Implementation of Standard Algorithms.*

The algorithms used in the original methodology include DT, SVM and LR. The code to implement these algorithms and their respective hyperparameters were not provided by the original paper, therefore this reproducibility paper used Scikit-Learn classifiers[14]. As the validation methodology was not specified in the original paper, this study implemented a 5-fold cross validation on each possible hyperparameter combination for each algorithm. This used GridSearchCV from Scikit-Learn[15]. Five-fold cross validation shuffles and splits the training set into four training folds and one test fold. The model is then fitted five times, with the test fold iterating through positions of the folds. With each implementation, performance metrics such as F-measure or accuracy are calculated and an average taken. This process is summarised in Table 1. This process was performed on each hyperparameter combination for both full and reduced datasets to identify the most effective.

Table 1: K-fold cross validation where K=5.

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Split 1	Test	Train	Train	Train	Train
Split 2	Train	Test	Train	Train	Train
Split 3	Train	Train	Test	Train	Train
Split 4	Train	Train	Train	Test	Train
Split 5	Train	Train	Train	Train	Test

Table 2: Hyperparameter values used during K-fold cross validation of each model

Model	Hyperparameter	Values
DT	criterion	entropy
	max_depth	2,3,4,5,6,7,8,9,10
	min_samples_split	2,5,10
	min_samples_leaf	1,2,4,8
	max_features	None, sqrt, log2
SVM	C	0.1, 1, 10, 100, 1000
	kernel	linear, rbf, poly
	gamma	scale, 0.01, 0.001
	degree	2, 3
LR	C	0.001, 0.01, 0.1, 1, 10, 100
	penalty	l1, l2
	solver	liblinear
	max_iter	100, 500, 1000, 2000

The hyperparameters used for each model during optimisation and k-fold cross validation are summarised in Table 2. The models were then tested on their respective unseen test sets, and the below performance metrics were calculated, as per the original methodology. The formula used are outlined below, with TP and TN as

True Positive and True Negative values, and FP and FN as False Positive and False Negative values.

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

$$Sensitivity = \frac{TP}{TP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Fmeasure = \frac{2 * precision * recall}{precision + recall}$$

In order to calculate TP, TN, FP and FN, a confusion matrix was generated for each model using Scikit-Learn's confusion matrix function[16]. The workflow from data sourcing to standard model implementation is summarised in Figure 2.

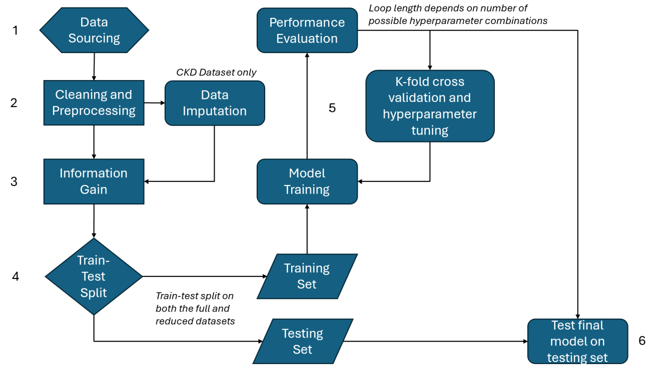
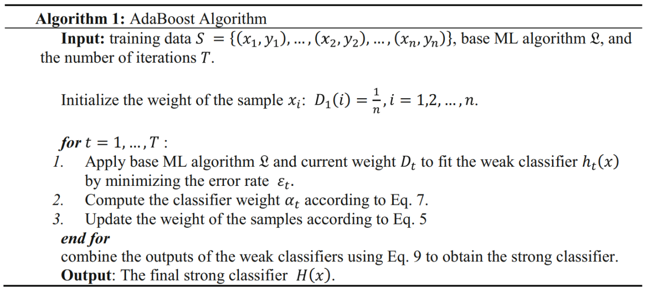

1
Data Sourcing
2
Cleaning and Preprocessing
CKD Dataset only
Data Imputation
3
Information Gain
4
Train-Test Split
Train-test split on both the full and reduced datasets
Training Set
Testing Set
Model Training
5
K-fold cross validation and hyperparameter tuning
Performance Evaluation
Loop length depends on number of possible hyperparameter combinations
Test final model on testing set
6

Figure 2: General workflow from data sourcing (1) to standard algorithm implementation (6).

2.1.6 Boosting the models. Once the standard algorithms were implemented, the original methodology applied the adaptive boosting technique, known as Ada-boost, to each algorithm. Boosting is a model ensemble technique that aims to improve the overall accuracy of a classifier. It does this by combining multiple weak classifiers by weighted voting. Introduced by Freund and Schapire [17], Ada-boost is one of the most popular boosting models. It adaptively adjusts to the errors of the weak hypothesis and samples training examples based on weights that determine the probability of being selected for training by the classifier[17]. The algorithm is summarised in Figure 3.

The original study does not specify the number of estimators, or the learning rate applied to the Ada-boost algorithms. This study utilised the AdaBoostClassifier function from Scikit-Learn [18] to implement the Ada-boost algorithm, selecting the maximum number of estimators as 100 and a learning rate of 0.01. Once the boosting was complete, it was noticed that the boosted LR model was overfitting and predicting all negative values. In order to mitigate this, class weight balancing was applied to the imbalanced datasets including CKD and Thyroid datasets. Results were then compared between ablations.



```
Algorithm 1: AdaBoost Algorithm
    Input: training data $S = \{(x_1, y_1), ..., (x_2, y_2), ..., (x_n, y_n)\}$, base ML algorithm $\mathfrak{L}$, and
  the number of iterations $T$.

    Initialize the weight of the sample $x_i$: $D_1(i) = \frac{1}{n}, i = 1,2, ..., n$.

    for $t = 1, ..., T$ :
   1.   Apply base ML algorithm $\mathfrak{L}$ and current weight $D_t$ to fit the weak classifier $h_t(x)$
        by minimizing the error rate $\varepsilon_t$.
   2.   Compute the classifier weight $\alpha_t$ according to Eq. 7.
   3.   Update the weight of the samples according to Eq. 5
    end for
    combine the outputs of the weak classifiers using Eq. 9 to obtain the strong classifier.
    Output: The final strong classifier $H(x)$.
```

Figure 3: Figure showing the Ada-boost Algorithm[3].

2.2 Challenges and Modifications

Overall, the paper did not provide sufficient detail to be able to reproduce the methodology exactly as it was originally done. To start, the paper did not provide any code or pseudocode to follow. The scikit-learn was package was used to complete tasks such as Min-Max Scaling, Information Gain, and implementing the algorithms, but it is unknown whether or not the researchers also used scikit-learn. The methodology section provided in the paper is fairly comprehensive, however, certain details were also left out of that. In terms of preprocessing for the CKD dataset, the authors of the paper mention converting categorical features to numeric and applying Min-Max Scaling, however they do not explain their methodology for handling missing values in the dataset. The first modification made to the original methodology was applying imputation to the CKD dataset in order to ensure there would be no missing values. The IG threshold set by the researchers in the paper was 'pe', and no reason was provided as to why this threshold was set at this feature. With two new datasets being tested on these models, the group did not want to set arbitrary IG threshold values for those datasets. Instead, the standard deviation of the IG data-frame was taken to determine the threshold for which features to retain. On top of an arbitrary IG threshold with no explanation attached, the researchers also did not specify model hyperparameters. Therefore, the group decided to utilize the gridsearch technique to find the optimal parameter values. This ensured the models would be using the hyperparameter combination which yields the best model performance. Finally, it was determined that two of the datasets used were unbalanced - CKD and Thyroid, resulting in improper classification for those two datasets. The paper did not make any mention of this issue, and the group had to make sure the dataset was balanced to avoid overfitting, particularly for the Logistic Regression model.

3 Results

To assess the effectiveness of information gain (IG) feature selection for classification, model performance was assessed using accuracy, precision, sensitivity and F-measure metrics. Additionally, the Receiver Operating Characteristics curves (ROC) were used to visually compare the classification performances and to identify any potential overfitting or unrealistically high results.

As the original paper [3] did not provide a justification for the chosen IG threshold (*'pe'*), this study tested two approaches:

(1) Replication of the original study, using the same arbitrary threshold (*'pe'*).

(2) An alternative approach, using the standard deviation of IG values was used as the threshold [13].

3.1 Original Dataset Results

3.1.1 Hyperparameter Tuning.

Grid Search was applied to the Decision Tree (DT), Support Vector Machine (SVM) and Logistic Regression (LR) models, the help find the best parameters for each model, shown in Table 3.

Table 3: Best hyperparameters found through Grid Search.

Model	**Full Feature Set**
DT	criterion='entropy', max_depth=5, min_samples_leaf=8
SVM	C=100, degree=2, kernel='linear'
LR	C=10, penalty='l1', solver='liblinear'

Model	**Reduced Feature Set**
DT	criterion='entropy', max_depth=3, min_samples_leaf=8
SVM	C=100, degree=2, kernel='linear'
LR	C=100, penalty='l2', solver='liblinear'

3.1.2 Original Arbitrary Feature Selection Threshold.

The original study [3] applied an arbitrary information gain (IG) threshold, *'pe'*, to select features for classification from the CKD dataset [2]. This resulted in 18 features that exceeded the *'pe'* threshold to be used to form the reduced feature set. Figure 4 shows the IG values of all features in descending order, resulting with the 18 features above the *'pe'* value forming the reduced dataset.

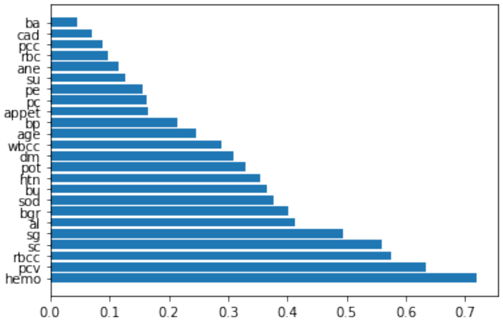


Figure 4: Information Gain values for all features in the CKD dataset.

To assess the impact of this feature selection method, the six classification models were trained on the full feature set (Table 4) and the reduced feature set (Table 5). The performance was evaluated using accuracy, sensitivity, precision and F-measure, with the addition of recall (used for F-measure calculation). Both the full dataset and the reduced feature set models have obtained similar metrics to what the original paper has documented, leading to the conclusion that the reproduced models performed on the same level as the original study.

Table 4: Model performance on the full dataset.

Model	Accuracy	Sensitivity	Precision	Recall	F-Measure
DT-Full	0.939	0.977	0.875	0.977	0.923
SVM-Full	0.974	1.000	0.935	1.000	0.966
LR-Full	0.982	1.000	0.956	1.000	0.977
AdaDT-Full	1.000	1.000	1.000	1.000	1.000
AdaSVM-Full	0.982	1.000	0.956	1.000	0.977
AdaLR-Full	0.640	0.093	0.667	0.093	0.163

Table 5: Model performance on the reduced dataset using the *'pe'* threshold.

Model	Accuracy	Sensitivity	Precision	Recall	F-Measure
DT-Reduced	0.947	1.000	0.878	1.000	0.935
SVM-Reduced	0.991	1.000	0.977	1.000	0.989
LR-Reduced	0.982	1.000	0.956	1.000	0.977
AdaDT-Reduced	1.000	1.000	1.000	1.000	1.000
AdaSVM-Reduced	0.974	1.000	0.935	1.000	0.966
AdaLR-Reduced	0.956	1.000	0.896	1.000	0.945

To further assess classification performance, Receiver Operating Characteristic (ROC) curves were visualised on all trained reduced feature set models (Figure 5). The Area Under Curve (AUC) values quantify the model's performances in distinguishing between the classification values. All models have achieved an AUC of over 0.99, which indicates further success of reproducing the original paper's results.

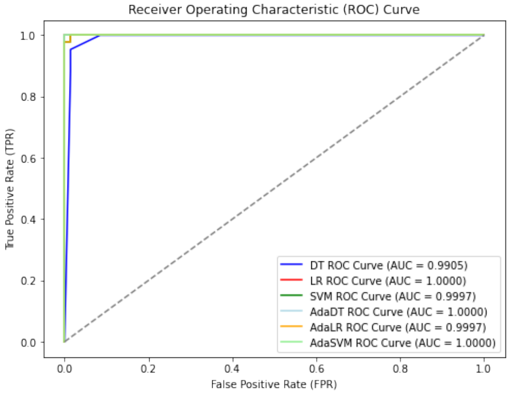


Figure 5: ROC curves for models trained on reduced dataset, includes AUC values.

3.1.3 Alternative Feature Selection Threshold: Standard Deviation.

Using the new methodology for the Information Gain feature selection threshold [13], the threshold for the CKD dataset was increased to only include features with an IG value above $\sigma = 0.187$. This was determined by calculating the standard deviation of the IG values, as described in Section 2.1.4. This increases the IG feature selection threshold to 'appet', removing nine features instead of the six features removed in the original study, resulting in a reduced dataset with 15 features.

Similar to the original method, the standard deviation threshold method was evaluated by repeating the training of the six models, completed on both the full and reduced feature dataset. The results for the alternative reduced alternative feature set in Table 6.

Table 6: Model performance on the reduced CKD dataset using the SD methodology for determining IG threshold.

Model	Accuracy	Sensitivity	Precision	Recall	F-Measure
DT-Reduced	0.947	1.000	0.878	1.000	0.935
SVM-Reduced	0.974	1.000	0.935	1.000	0.966
LR-Reduced	0.965	1.000	0.915	0.915	0.956
AdaDT-Reduced	1.000	1.000	1.000	1.000	1.000
AdaSVM-Reduced	0.974	1.000	0.935	1.000	0.966
AdaLR-Reduced	0.939	1.000	0.860	1.000	0.925

For further model performance evaluation, ROC curves were also generated for all models trained on the reduced dataset (Figure 6). The ROC curves help confirm that the models were capable of CKD prediction, with the AUC values all being consistently above 0.99. These findings solidify that the standard deviation threshold approach is effective in feature selection, while being a well documented and reproducible technique to be used for the remaining datasets.

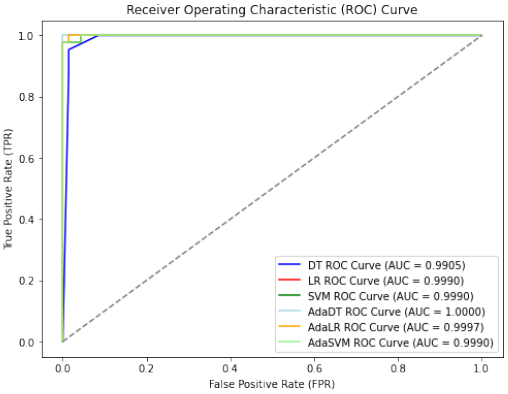


Figure 6: ROC curves for models trained on the reduced dataset using the standard deviation IG threshold.

3.2 Modern Dataset Results

3.2.1 *Overview.*

This section presents the experimental results obtained from evaluating multiple classification models across two modern datasets: The Differentiated Thyroid Cancer Reoccurrence dataset and the Rice Classification dataset. The same methodology applied to the CKD dataset, using ST to derive the IG threshold value, is applied to the modern datasets.

3.2.2 *Differentiated Thyroid Cancer Reoccurrence Dataset Results.*

The thyroid dataset consists of 16 features and contains 383 records. This is similar to the CKD dataset in that it has many features but relatively few records. It is also very unbalanced, with only 28 percent of records listing thyroid cancer reoccurrence as 'Yes'. Due to this imbalance, the F-Measure is the primary metric used for evaluating the model's performance on the data. Unlike the CKD dataset, the thyroid dataset does not require any imputation of values. Using the previously defined methodology for IG threshold selection, the reduced thyroid dataset consists of seven features, with nine features below the threshold. After the 70:30 train-test split is performed, the six classification methods are applied to the full (Table 7) and reduced (Table 8) thyroid datasets.

Table 7: Model performance on the full thyroid dataset.

Model	Accuracy	Sensitivity	Precision	Recall	F-Measure
DT	0.957	0.875	0.966	0.875	0.918
SVM	0.939	0.875	0.903	0.875	0.889
LR	0.913	0.844	0.844	0.844	0.844
AdaDT	0.957	0.938	0.909	0.938	0.923
AdaSVM	0.843	0.531	0.850	0.531	0.654
AdaLR	0.722	0.000	0.000	0.000	0.000

Table 8: Model performance on the reduced thyroid dataset.

Model	Accuracy	Sensitivity	Precision	Recall	F-Measure
DT-Reduced	0.965	0.906	0.967	0.906	0.935
SVM-Reduced	0.922	0.781	0.926	0.781	0.847
LR-Reduced	0.922	0.844	0.871	0.844	0.857
AdaDT-Reduced	0.965	0.969	0.912	0.969	0.939
AdaSVM-Reduced	0.826	0.406	0.929	0.406	0.565
AdaLR-Reduced	0.913	0.750	0.923	0.750	0.828

In the full thyroid dataset, Ada-DT achieves the highest F-Measure (0.923), followed by DT (0.918) and SVM (0.889). The Ada-LR model performs the worst, calculating a low F-Measure (0.000) and low accuracy (0.722). This is after balancing was implemented. Without balancing of the data, the LR model predicts all negatives, with no true or false positive values identified. Generally, boosting does not improve performance, with two of the three boosted models performing worse than their standard counterpart.

In the reduced thyroid dataset, Ada-DT (0.939) and DT (0.935) again perform the best, followed by LR (0.857). The boosting of SVM in the reduced dataset causes a significant decrease in accurate classification, dropping from 0.847 to 0.565.

The ROC curves for models trained on the reduced dataset in Figure 7 show how well the classification models separate the data across all thresholds, all of which have an AOC of 0.93 or above.

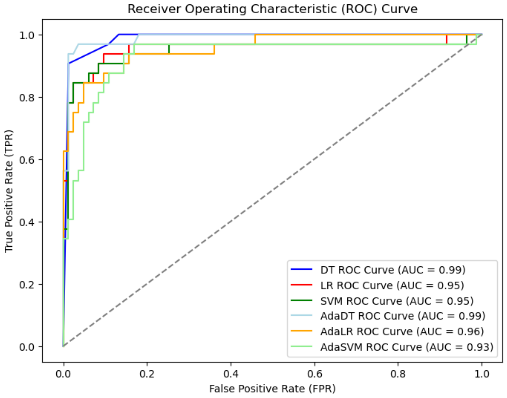


Figure 7: ROC curves for models trained on the reduced thyroid dataset

3.2.3 Rice Classification Dataset Results.

This dataset consists of seven features and contains 3,810 records. Unlike the CKD dataset, the rice dataset does not require imputation and is relatively balanced (43:57), therefore accuracy is the best metric for evaluating its performance instead of F-Measure, which is used in the previous datasets. Using the new methodology for the selection threshold of Information Gain features, the IG value threshold for the rice dataset removes two features, creating a reduced rice dataset with five features. After the 70:30 train-test split is performed, the six classification methods are applied to the full (Table 9) and reduced (Table 10) rice datasets.

Table 9: Model performance on the full rice dataset.

Model	Accuracy	Sensitivity	Precision	Recall	F-Measure
DT-Full	0.919	0.954	0.903	0.954	0.928
SVM-Full	0.927	0.941	0.926	0.941	0.933
LR-Full	0.933	0.952	0.927	0.952	0.939
AdaDT-Full	0.923	0.952	0.911	0.952	0.931
AdaSVM-Full	0.929	0.926	0.943	0.926	0.935
AdaLR-Full	0.930	0.954	0.921	0.954	0.937

Table 10: Model performance on the reduced rice dataset.

Model	Accuracy	Sensitivity	Precision	Recall	F-Measure
DT-Reduced	0.926	0.952	0.915	0.952	0.933
SVM-Reduced	0.930	0.949	0.925	0.949	0.937
LR-Reduced	0.927	0.950	0.918	0.950	0.934
AdaDT-Reduced	0.929	0.950	0.922	0.950	0.936
AdaSVM-Reduced	0.927	0.931	0.934	0.931	0.933
AdaLR-Reduced	0.547	1.000	0.547	1.000	0.707

In the full rice dataset, the LR classification method performs the best, with an accuracy of 0.933. This is followed by Ada-LR (0.930) and Ada-SVM (9.292). Boosting slightly improve DT and SVM performance, but the change in accuracy is minimal (0.002-0.004 increase) and decreases LR accuracy by 0.003. The SVM model performs the best (0.930) in the reduced dataset, followed by Ada-DT (0.929). Boosting improves the DT classification method, but performs slightly worse with SVM. The boosted LR method is an outlier, as it performs significantly worse than every other method.

Overall, the rice dataset maintained a strong classification performance likely because of its large record size. The large volume of data also likely improved precision throughout the methods, with accuracies in both datasets performing within 0.014 of each other. However, Ada-LR has struggled in the reduced dataset, potentially due to overfitting or due to its lower sensitivity to the selected features. The ROC curves of the models trained on the reduced dataset in Figure 8 indicate an accurate classification of data into their thresholds, with AUCs of all the models over 0.94.

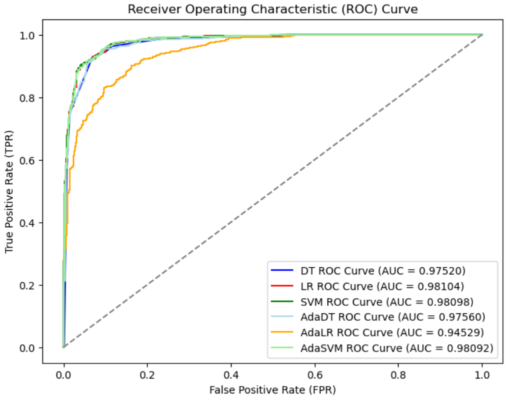


Figure 8: ROC curves for models trained on the reduced thyroid dataset

4 Discussion

4.1 Interpretation of Results

The reproduction closely replicated the original study's results in the context of the CKD dataset, particularly for the AdaBoost-DT model, which consistently achieved 100% accuracy, precision, sensitivity, and F-measure (Table 4). However, a significant divergence emerged in the performance of AdaBoost-LR, which drastically underperformed in several configurations. For instance, on the full Thyroid dataset, AdaBoost-LR returned an F-measure score of 0.000 (Table 7), failing to predict any positive cases, despite having class balancing enabled. A review of the confusion matrix and prediction outputs confirmed that the model classified all test samples as the majority class.

This behavior is an indicator of overfitting. The use of boosting may have applied variations or noise in the training data leading to poor generalization. The combination of an inherently linear base model and an imbalanced dataset likely contributed to this breakdown, as logistic regression is sensitive to poorly separated classes without strong regularization [19].

In contrast, AdaBoost-DT showed consistency across all datasets, validating the studies original claims that boosting significantly improves Decision Tree classifiers. This is because Decision Trees, while prone to overfitting, benefit from AdaBoost's iterative correction of errors [20].

4.2 Insights from Modern Datasets

Testing the reproduced methodology on different datasets - the Rice (Cammeo and Osmancik) and Differentiated Thyroid Cancer Recurrence datasets - provided critical insights into the generalization of the models beyond the original CKD data.

The Rice dataset, being large (3,810 samples) and relatively clean with only seven features, is ideal for assessing scalability and baseline performance in low-noise settings. All models performed well on this dataset, with accuracy generally above 90% (Table 9). Standard SVM on the reduced dataset achieved 93% accuracy (Table 10), while AdaBoost-SVM reached 92.7% (Table 10), showing weaker classification from boosting. This suggests that for large, low-dimensional datasets with minimal noise, boosting offers diminishing returns, particularly when base classifiers already perform well. Additionally, AdaBoost-LR dropped in performance from 93% (Table 9) on the full dataset to 54.7% on the reduced dataset (Table 10), indicating that even the removal of two features significantly altered its capacity to find a useful decision boundary. This shows that boosting a linear classifier in a low-dimensional space can make it brittle, especially if informative features are removed.

Conversely, the Thyroid dataset is smaller (383 rows), contains 16 features, and exhibits class imbalance. Here, performance varied widely depending on the model and whether feature selection was applied. For example, standard SVM had an F-measure score of 0.847 (Table 8), while AdaBoost-SVM dropped to 0.565 (Table 8) on the reduced dataset - despite boosted models generally being expected to outperform their non-boosted counterparts. This counterintuitive result reflects overfitting from boosting on a small, imbalanced dataset, where misclassified samples are repeatedly up-weighted in ways that reinforce noise rather than correct errors.

An interesting positive outlier was AdaBoost-LR on the reduced Thyroid dataset, which improved from 0.000 (full dataset, Table 7) to 0.828 (reduced dataset, Table 8) in F-measure. This shows that feature selection can correct instability caused by irrelevant or misleading features, especially in linear models. However, this trend did not generalize - on the Rice dataset, reducing features hurt AdaBoost-LR performance (Table 9). This also shows the model may have been drastically over-fitted to the full Thyroid dataset, leading to incorrect results.

Across both new datasets, performance trends showed that:

1. Data size, class balance, and feature quality significantly affect the stability and effectiveness of ensemble models.
2. Boosting is most reliable when applied to tree-based learners (e.g., DT), and its benefits on SVM and LR vary widely depending on data conditions.
3. Feature selection can sometimes enhance or degrade performance, depending on whether the removed features are redundant or essential.

4.3 Comparative Analysis

The side-by-side comparison of model performance across the three datasets highlighted several important patterns and differences.

The CKD dataset, which is relatively small (400 rows) and moderately balanced, provided results very close to those in the original study. Most notably, AdaBoost-DT achieved perfect classification (100%). The minimal number of features (15 after standard deviation IG-based reduction) likely made the dataset amenable to both linear and non-linear models. The effectiveness of boosting here validates the original paper's findings. However, this performance potentially reflects dataset simplicity and cleanliness, not just model strength. These conditions are not representative of more complex, noisy real-world scenarios.

In contrast, the Thyroid dataset showed greater divergence between model types and preprocessing configurations:

1. Reducing the feature set from 16 to 7 features via IG substantially improved performance for AdaBoost-LR (F-measure = 0.828, Table 7), suggesting that removal of noisy or irrelevant features allowed the model to properly classify the dataset.
2. AdaBoost-SVM, however, performed worse of the reduced dataset (Table 8), indicating that SVM may depend on complex feature interactions that were lost during feature selection.

The Rice dataset provided another contrast. Here, the class distribution was well-balanced, and most models performed well. However, the difference between full and reduced feature sets produced some anomalies:

1. AdaBoost-DT showed stable performance (92.3% on full, Table 9), 92.9% on reduced, Table 10).
2. AdaBoost-LR, once again, dropped sharply from 93% (full, Table 9) to 54.7% (reduced, Table 10), showing that even minor reductions in a low-dimensional space can impact a model that relies on linear separability.

Some conclusions can be drawn from this analysis:

1. Boosting improves tree-based classifiers (DT) across all datasets, especially in low-sample settings like CKD or Thyroid.
2. The impact of feature selection is dataset- and model-dependent. It helps LR when removing noisy features (such as on Thyroid) but hurts it when removing even a few features (Rice).
3. Performance differences arise from a combination of factors, including: model type (linear vs. non-linear), dataset size and dimensionality, feature quality and class balance, and sensitivity of boosting algorithms to noise [21].

4.4 Limitations

Several limitations affected the reproducibility and interpretation of results:

1. Lack of Original Code or Detailed Methodology: The original study did not provide source code, exact preprocessing steps, or hyperparameters. Key choices like IG thresholds or learning rates were either arbitrary or undocumented, requiring informed assumptions during reproduction.
2. Class Imbalance and Overfitting: The Thyroid dataset, in particular, suffered from class imbalance, which exposed overfitting in the boosted LR model. Although class weights were added, performance issues remained. This suggests a need for more advanced balancing techniques like SMOTE or cost-sensitive loss functions [22].

(3) Inadequate Justification for IG Thresholds: While the original study used a static threshold ('pe'), our application of the standard deviation-based threshold was more systematic but may still be suboptimal across datasets with vastly different data distributions.
(4) AdaBoost Implementation Fragility: AdaBoost is designed to improve weak learners, but not all base classifiers benefit equally. Our results indicate that AdaBoost-LR is prone to collapse under certain conditions, especially with limited or imbalanced data, or in the case where there aren't many weak classifiers.

5 Conclusion

5.1 Summary of Key Findings

This study reproduced and extended the original work by Mienye et al. (2021), confirming effectiveness of AdaBoost-DT for classification across multiple datasets. It achieved perfect classification on the CKD dataset, and strong performance on both the Thyroid and Rice datasets. The study also highlighted the limitations of boosting for linear models like LR, where overfitting, instability, or collapse occurred under certain conditions. Information Gain was useful for feature selection, particularly in smaller datasets, but its effectiveness diminished on low-dimensional data like the Rice dataset.

5.2 Reflections on the Process

The reproduction process provided valuable insight into the importance of clear and detailed documentation in machine learning research. The original paper did not include source code, specific hyperparameter settings, or detailed steps for preprocessing (e.g., handling of missing values, IG thresholding), which made direct reproduction difficult. As a result, our group had to make assumptions about:

(1) The handling of missing data (e.g., using median imputation).
(2) The exact threshold used for feature selection via Information Gain.
(3) The internal configuration of AdaBoost (e.g., number of estimators, base learners).

Despite these gaps, we were able to reconstruct the methodology closely and produce comparable results, demonstrating that replication is possible with well-understood ML frameworks, but that it is also fragile when methodological transparency is limited. Applying the models to modern datasets also offered a deeper understanding of model behavior in real-world scenarios. The Rice dataset revealed that boosted models are robust and scalable, while the Thyroid dataset highlighted potential weaknesses when facing class imbalance or feature sparsity, especially for SVM and LR. This process also emphasized that reproducibility and replicability are not the same. While we could replicate the methodology, exact reproducibility was not achievable without source code or more detailed experimental logs. This highlights the need for the ML community to adopt better reproducibility standards, including publishing code and data pipelines.

5.3 Future Work

Building on the findings of this project, there are a few other directions for future research:

(1) Alternative Ensemble Methods: More advanced or optimized ensemble techniques could be explored, such as XGBoost [23], which could potentially outperform AdaBoost in real-world applications due to better handling of missing data, regularization, and speed.
(2) Robust Feature Selection Techniques: The use of Information Gain worked well in this study, but with more time, experimentation with various IG thresholds could be conducted to see how many features could be removed before performance is impacted. Other methods like Recursive Feature Elimination [24] could be explored, as they might yield more optimal subsets, especially for high-dimensional data.
(3) Class Imbalance Handling: Results from the Thyroid dataset show that performance suffers in the case of class imbalance. Future work could include data resampling techniques, such as SMOTE [22], to improve model robustness on imbalanced datasets.
(4) Model Explainability and Fairness: For real-world use, especially in healthcare, interpretability is crucial. Incorporating SHAP values [25] can help explain how features influence predictions. Evaluating the fairness of models across different subgroups in the data (e.g., gender, age, etc.) would be important for ethical deployment.

References

[1] Gundersen OE, Kjensmo S. State of the art: Reproducibility in artificial intelligence. In: Proceedings of the AAAI conference on artificial intelligence. vol. 32; 2018. .
[2] Rubini L, Soundarapandian P, Eswaran P. Chronic Kidney Disease; 2015. DOI: https://doi.org/10.24432/C5G020. UCI Machine Learning Repository.
[3] Mienye ID, Obaido G, Aruleba K, Dada OA. Enhanced prediction of chronic kidney disease using feature selection and boosted classifiers. In: International conference on intelligent systems design and applications. Springer; 2021. p. 527-37.
[4] Subasi A. Chapter 3 - Machine learning techniques. In: Subasi A, editor. Practical Machine Learning for Data Analysis Using Python. Academic Press; 2020. p. 91-202. Available from: https://www.sciencedirect.com/science/article/pii/B9780128213797000035.
[5] Dilmegani C. Reproducible AI: Why it Matters & How to Improve it [2025]?; 2025. Available from: https://research.aimultiple.com/reproducible-ai/.
[6] Heil BJ, Hoffman MM, Markowetz F, Lee SI, Greene CS, Hicks SC. Reproducibility standards for machine learning in the Life Sciences. Nature Methods. 2021 Aug;18(10):1132–1135.
[7] Desai A, Abdelhamid M, Padalkar NR. What is Reproducibility in Artificial Intelligence and Machine Learning Research?; 2024. Available from: https://arxiv.org/html/2407.10239v1#bib.
[8] van Giffen B, Herhausen D, Fahse T. Overcoming the pitfalls and perils of algorithms: A classification of machine learning biases and mitigation methods. Journal of Business Research. 2022;144:93-106. Available from: https://www.sciencedirect.com/science/article/pii/S0148296322000881.
[9] Semmelrock H, Kopeinik S, Theiler D, Ross-Hellauer T, Kowald D. Reproducibility in Machine Learning-Driven Research. arXiv preprint. 2023.
[10] Borzooei S, Tarokhian A. Differentiated Thyroid Cancer Recurrence; 2023. DOI: https://doi.org/10.24432/C5632J. UCI Machine Learning Repository.
[11] Rice (Cammeo and Osmancik); 2019. DOI: https://doi.org/10.24432/C5MW4Z. UCI Machine Learning Repository.
[12] Reddy GT, Reddy MPK, Lakshmanna K, Kaluri R, Rajput DS, Srivastava G, et al. Analysis of Dimensionality Reduction Techniques on Big Data. IEEE Access. 2020;8:54776-88.
[13] Prasetiyowati MI, Maulidevi NU, Surendro K. Determining threshold value on information gain feature selection to increase speed and prediction accuracy of random forest. Journal of Big Data. 2021;8(1):84.
[14] Scikit-Learn. Scikit-Learn: Machine Learning in Python; 2025. Accessed: March 25, 2025. Available from: https://scikit-learn.org/stable/.

[15] Scikit-Learn. GridSearchCV; 2025. Accessed: March 25, 2025. Available from: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html.

[16] Scikit-Learn. confusionmatrix; 2025. Accessed: March 25, 2025. Available from: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html.

[17] Freund Y, Schapire RE. A decision-theoretic generalization of on-line learning and an application to boosting. Journal of computer and system sciences. 1997;55(1):119-39.

[18] Scikit-Learn. AdaBoostClassifier; 2025. Accessed: March 25, 2025. Available from: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html.

[19] Zhang L, Geisler T, Ray H, Xie Y. Improving logistic regression on the imbalanced data by a novel penalized log-likelihood function. Journal of Applied Statistics. 2021 Jun;49(13):3257–3277.

[20] Misra S, Li H. Chapter 9 - Noninvasive fracture characterization based on the classification of sonic wave travel times. In: Misra S, Li H, He J, editors. Machine Learning for Subsurface Characterization. Gulf Professional Publishing; 2020. p. 243-87. Available from: https://www.sciencedirect.com/science/article/pii/B9780128177365000090.

[21] Bognár L, Fauszt T. Factors and conditions that affect the goodness of machine learning models for predicting the success of learning. Computers and Education: Artificial Intelligence. 2022;3:100100. Available from: https://www.sciencedirect.com/science/article/pii/S2666920X22000558.

[22] Elreedy D, Atiya AF. A Comprehensive Analysis of Synthetic Minority Oversampling Technique (SMOTE) for handling class imbalance. Information Sciences. 2019;505:32-64. Available from: https://www.sciencedirect.com/science/article/pii/S0020025519306838.

[23] Chen T, Guestrin C. XGBoost: A Scalable Tree Boosting System. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD '16. New York, NY, USA: Association for Computing Machinery; 2016. p. 785–794. Available from: https://doi.org/10.1145/2939672.2939785.

[24] Chen Xw, Jeong JC. Enhanced recursive feature elimination. In: Sixth International Conference on Machine Learning and Applications (ICMLA 2007); 2007. p. 429-35.

[25] Meng Y, Yang N, Qian Z, Zhang G. What Makes an Online Review More Helpful: An Interpretation Framework Using XGBoost and SHAP Values. Journal of Theoretical and Applied Electronic Commerce Research. 2021;16(3):466-90. Available from: https://www.mdpi.com/0718-1876/16/3/29.