**Experimentation of a New Approach Based on Ensemble Learning Estimator to Maximize Accuracy**

#

# Abstract

In this article, we introduce a novel approach based on an ensemble learning estimator specifically designed to optimize the predictive accuracy of supervised classification models. This contribution, both theoretical and methodological, relies on the strategic combination of multiple heterogeneous learning algorithms (decision trees, boosting methods, SVMs, etc.) orchestrated through a meta-model. The resulting architecture, named MaxEnsForest, is aimed at enhancing the robustness, accuracy, and generalization capacity of traditional models.

Within this framework, we present the results of an extensive series of experiments conducted on several benchmark datasets to evaluate the performance of MaxEnsForest under diverse conditions. The study highlights the individual contributions of each component within the architecture, as well as the impact of integrated optimization strategies such as GridSearchCV, feature importance analysis, and performance visualization through robust evaluation metrics.

This work seeks to establish a rigorous transition from theoretical design to solid experimental validation, empirically demonstrating the relevance and superiority of MaxEnsForest compared to conventional ensemble learning techniques.

Moreover, this research proposes an optimized ensemble learning architecture centered around a Grand Estimator, designed to maximize prediction accuracy while ensuring stability, robustness, and resilience to data variability.

***Keywords:****Ensemble learning, Ensemble estimator, Meta-model, Predictive accuracy, Generalization performance, , Hyperparameters, MaxEnsForest.*

# 2. Introduction

**Ensemble learning has demonstrated its ability to improve the performance of individual models by combining multiple base learners. However, optimizing predictive accuracy remains a challenge, particularly in the face of model variance and data heterogeneity. As highlighted in our previous work, we introduce MaxEnsForest, a novel hybrid approach that leverages the strengths of multiple supervised algorithms, integrated through a meta-model structured as a lightweight neural network.**

**Ensemble methods such as Random Forest, AdaBoost, Gradient Boosting, and Extremely Randomized Trees have become standard approaches for solving complex supervised classification problems. Their success is largely attributed to their ability to combine multiple base models (weak learners) to reduce variance, mitigate bias, and improve the overall robustness of the final model.**

**Nevertheless, despite their empirically strong performance, these methods have several notable limitations :**

* The prediction aggregation strategy is often fixed (e.g., majority vote, weighted average), limiting adaptability.
* They frequently fail to fully exploit the complementarity and diversity among models.
* Hyperparameter tuning is commonly performed manually or through random search, which may lead to suboptimal configurations and hinder generalization.
* Most ensemble methods struggle to integrate heterogeneous algorithms such as SVMs or neural networks due to structural differences in prediction outputs.
* The increasing volume and complexity of modern datasets often render traditional learning methods insufficient for achieving reliable and generalizable predictions.
* Existing ensemble techniques, though effective, still exhibit limitations in terms of accuracy, stability, and sensitivity to hyperparameters particularly when dealing with imbalanced classes or noisy data.

No method, to date, guarantees consistently optimal predictive accuracy. *MaxEnsForest* is proposed as a dynamic solution that combines multiple strategies to address these limitations, with a central focus on accuracy optimization.

In light of these observations, several key research questions arise :

* How can we design an ensemble architecture that is both flexible and adaptive, capable of adjusting to various data types and classification tasks ?
* What mechanisms can effectively leverage the complementarity of heterogeneous models within an ensemble framework ?
* How can we ensure both the stability and interpretability of predictions generated by an ensemble model ?
* What optimization strategy can be implemented to efficiently tune both hyperparameters and the meta-model ?
* How does *MaxEnsForest* perform across different datasets compared to conventional ensemble methods such as Random Forest, AdaBoost, and Gradient Boosting ?
* What tools and techniques can be employed to evaluate and interpret the results, in order to demonstrate the generalization capabilities and overall effectiveness of the *MaxEnsForest* model ?

These questions form the foundation of our study, which aims to explore a new ensemble learning approach *MaxEnsForest* that integrates various base learners (Decision Trees, Boosting methods, SVMs, etc.) under the supervision of a neural network-based hyper-model.

# 3. Proposed Methodology

The methodology adopted for the development and evaluation of MaxEnsForest follows a structured approach, organized into several complementary phases.

Initially, a synthetic dataset was generated using the make\_classification function from the scikit-learn library. This dataset allows for controlled complexity by including informative and redundant features, as well as imbalanced classes, in order to simulate a realistic environment for supervised learning.

The second phase involves designing the architecture of the MaxEnsForest algorithm. This model is a heterogeneous ensemble that combines several base classifiers (Random Forest, SVM, AdaBoost, Gradient Boosting, etc.), whose outputs are aggregated through a meta-model. The meta-model may consist of either a lightweight neural network or a linear model. This aggregation strategy aims to maximize diversity while minimizing overall error.

Subsequently, fine-tuning of hyperparameters is conducted using exhaustive grid search (GridSearchCV), both for the base learners and the meta-model.

Finally, the model’s performance is assessed using multiple, complementary evaluation metrics, including accuracy, weighted precision, F1-score, and the confusion matrix, in order to provide a comprehensive view of the model’s behavior.

## **3.1 Architecture**

New Data

 …

Training stage

 *Predictions* *Predictions*

 . . . . . .

 . . . . . .

 . . . . . .

 Homogeneous classification mode Heterogeneous classification mode

**Hyper-Estimator**

**MaxEnsForest**

 Finales Predictions

## **Fig 1- An optimized ensemble learning architecture referred to as a Grand Estimator named MaxEnsForest (Mabumbi et al.2025)**

Symbol Interpretation:

|  |  |
| --- | --- |
|     | : The different errors of each learning level. |
|    | : Large ensemble classifiers. |
|  | : The different weights of the large ensemble estimators. |
|  | : Mode predictions of homogeneous estimators. |
|  | : Mode predictions of heterogeneous estimators. |
|   | : The different homogeneous and heterogeneous estimators. |
|   | : Meta-classifier prediction. |

K-Fold1 and K-Fold2: The different parts of the DATASET divided into two to reduce over-fitting and under-fitting errors.

## **3.2 Algorithm Operation**

 **Algorithm : « *MaxEnsForest* »**

**Input** :

* Training data *:*
* *(m ,…,M ) :* The number of iterations
* *P:* The number of estimators
* *:* The different weights of the large ensemble classifiers
* *:* Excellent estimator of each training level.
* *:* Error of each training level.

**Output** : A set of estimators

***Step 1: Learn the homogeneous and heterogeneous classifiers***

Learn a new estimator based on the set D

***Step 2: Learn the large classifiers***

 Learn a new estimator based on its predictions

1. Initialize the weights := , *i* = 1 *,..., n*
2. Calculate

**For *m* = *1* to *M***:

 Fit the weak rule on the sample of estimators weighted by the weights *,...,* resulting from this fit.

1. Calculate the error rate of  **:**
2. Calculate the weights of the resulting estimators **:**
3. Readjust the weights of the estimators: **:**

 ***Step 3 :*** Construct a new dataset ***D***

The construction of a new dataset will contain the predictions of the first level

**Step 4 :** Repeat step (2)

**Return**

# 4. Experimentation

## **4.1 Datasets**

The experimentation is conducted using standard datasets (*make\_classification*, *Breast Cancer*) available in the *scikit-learn* library. These are supervised classification datasets commonly used for testing, training, and evaluating artificial intelligence algorithms in the field of machine learning. The datasets undergo preprocessing, including standardization and an 80/20 split into training and testing sets, depending on the dataset characteristics.

**4.2 Implementation in the Anaconda Environment (Jupyter Notebook)**

****

**Fig 2-Library Imports**

****

**Fig 3- Dataset Generation, Splitting, and Definition of Base Models**

****

**Fig 4-Optimization, Meta-Feature Initialization, and Cross-Validation**

****

**Fig 5- Training and Evaluation of MaxEnsForest**

****

**Fig 6- Visualization of Results in Graphical Form**

**4.3 Metrics**

In the context of evaluating our approach, we employed a set of robust evaluation metrics to ensure a comprehensive analysis of the model’s performance. The selected indicators are as follows :

* **Overall Accuracy:** Measures the rate of correct predictions across all classes;
* **Weighted Average Precision:** Accounts for class imbalances by assigning a weight proportional to the frequency of each class;
* **Confusion Matrix :** Provides a detailed visualization of classification errors among different categories;
* **Weighted F1-Score :** Offers a balance between precision and recall, particularly useful in contexts where false positives and false negatives have critical implications. This combination of metrics allows us to finely evaluate not only the overall performance of the model but also its ability to maintain consistency on imbalanced or complex datasets.

## **4.4 Results**

### **a) Evaluation of the overall accuracy of the MaxEnsForest model with the make\_classification dataset.**

1. **Weighted Average Precision**

Summary of errors by model :

DecisionTree - Average error: 0.1200, Standard deviation : 0.0257

RandomForest - Average error: 0.0750, Standard deviation : 0.0079

GradientBoostingClassifier - Average error: 0.0750, Standard deviation : 0.0209

AdaBoost - Average error: 0.0925, Standard deviation : 0.0170

SVM - Average error: 0.1350, Standard deviation : 0.0215

Epoch 1/20

**Fig 7** **Accuracy of the MaxEnsForest model**

**25/25** ━━━━━━━━━━━━━━━━━━━━ **1s** 4ms/step - accuracy: 0.8022 - loss: 0.5658

Epoch 2/20

**25/25** ━━━━━━━━━━━━━━━━━━━━ **0s** 4ms/step - accuracy: 0.9271 - loss: 0.2472

Epoch 3/20

**25/25** ━━━━━━━━━━━━━━━━━━━━ **0s** 4ms/step - accuracy: 0.9084 - loss: 0.2756

Epoch 4/20

**25/25** ━━━━━━━━━━━━━━━━━━━━ **0s** 3ms/step - accuracy: 0.9373 - loss: 0.2204

Epoch 5/20

**25/25** ━━━━━━━━━━━━━━━━━━━━ **0s** 4ms/step - accuracy: 0.9379 - loss: 0.2137

Epoch 6/20

**25/25** ━━━━━━━━━━━━━━━━━━━━ **0s** 4ms/step - accuracy: 0.9285 - loss: 0.2373

Epoch 7/20

**25/25** ━━━━━━━━━━━━━━━━━━━━ **0s** 3ms/step - accuracy: 0.9245 - loss: 0.2335

Epoch 8/20

**25/25** ━━━━━━━━━━━━━━━━━━━━ **0s** 4ms/step - accuracy: 0.9406 - loss: 0.2195

Epoch 9/20

**25/25** ━━━━━━━━━━━━━━━━━━━━ **0s** 4ms/step - accuracy: 0.9363 - loss: 0.1973

Epoch 10/20

**25/25** ━━━━━━━━━━━━━━━━━━━━ **0s** 3ms/step - accuracy: 0.9349 - loss: 0.2276

Epoch 11/20

**25/25** ━━━━━━━━━━━━━━━━━━━━ **0s** 4ms/step - accuracy: 0.9438 - loss: 0.2119

Epoch 12/20

**25/25** ━━━━━━━━━━━━━━━━━━━━ **0s** 3ms/step - accuracy: 0.9257 - loss: 0.2427

Epoch 13/20

**25/25** ━━━━━━━━━━━━━━━━━━━━ **0s** 5ms/step - accuracy: 0.9462 - loss: 0.2016

Epoch 14/20

**25/25** ━━━━━━━━━━━━━━━━━━━━ **0s** 3ms/step - accuracy: 0.9259 - loss: 0.2363

Epoch 15/20

**25/25** ━━━━━━━━━━━━━━━━━━━━ **0s** 4ms/step - accuracy: 0.9280 - loss: 0.2431

Epoch 16/20

**25/25** ━━━━━━━━━━━━━━━━━━━━ **0s** 3ms/step - accuracy: 0.9293 - loss: 0.2355

Epoch 17/20

**25/25** ━━━━━━━━━━━━━━━━━━━━ **0s** 3ms/step - accuracy: 0.9218 - loss: 0.2535

Epoch 18/20

**25/25** ━━━━━━━━━━━━━━━━━━━━ **0s** 3ms/step - accuracy: 0.9444 - loss: 0.2016

Epoch 19/20

**25/25** ━━━━━━━━━━━━━━━━━━━━ **0s** 3ms/step - accuracy: 0.9392 - loss: 0.2101

Epoch 20/20

**25/25** ━━━━━━━━━━━━━━━━━━━━ **0s** 3ms/step - accuracy: 0.9364 - loss: 0.2016

**4/4** ━━━━━━━━━━━━━━━━━━━━ **0s** 21ms/step

***Final prediction of MaxEnsForest : 0.9500***



**Fig 8**-**Evaluation of the Overall Accuracy of the MaxEnsForest Model**



**Fig 9**- **Confusion Matrix**



**Fig 10**-**Calculation of the Error at Each Learning Level**





**Fig 11**- **Error at Each Learning Level**





**Fig 12**-**Calculation of the Error of Our MaxEnsForest Model**

**7. Approach Optimization**

The effectiveness of the MaxEnsForest model relies not only on its heterogeneous ensemble structure, integrating multiple base estimators (decision trees, Random Forest, SVM, AdaBoost, GBT, etc.), but also on a rigorous optimization phase that enhances its robustness and generalization.

Initially, the dataset is split into multiple parts, referred to as K-Fold1, K-Fold2, K-Fold3, etc., and a hyperparameter search is performed using the GridSearchCV method. This search is applied hierarchically, first to the base models, then to the meta-model. This strategy allows exhaustive exploration of optimal parameter combinations for each sub-model while minimizing the risk of overfitting.

Next, a feature importance analysis is conducted to identify the most discriminative attributes. This step strengthens the model’s interpretability and reduces complexity by eliminating less relevant features using measures such as Gini importance, linear SVM coefficients, or SHAP (Shapley Additive Explanations) for advanced versions.

Finally, stratified cross-validation techniques are integrated throughout the pipeline to ensure reliable and balanced evaluation, especially in the presence of imbalanced classes. This interactive optimization approach, coupled with a modular architecture, makes MaxEnsForest a powerful, interpretable ensemble model suitable for complex classification use cases.

In summary, the approach is enhanced by :

* **GridSearchCV:** Automated hyperparameter optimization

optimized\_models = {}

for name, model in base\_models.items():

 grid = GridSearchCV(model, param\_grid[name], cv=3, scoring='accuracy', n\_jobs=-1)

 grid.fit(x\_train, y\_train)

 print(f"Meilleurs paramètres pour {name}: {grid.best\_params\_}")

 optimized\_models[name] = grid.best\_estimator\_

* **Feature Importance :** Selection of significant variables.
* ***Feature Importance***



**Fig 13**- **Importance of variables (Random Forest)**

### **b) Evaluation and Comparison of the Overall Accuracy of MaxEnsForest with Other Algorithms on the "Breast Cancer" Dataset.**



**Fig 14- Evaluation and Comparison of MaxEnsForest**

**

**Fig 15- Creation of the MaxEnsForest Class**

**

**Fig 16 Importing Dataset and Base Models**

**

**Fig 17 Data Normalisation**

**

**Fig 18-Formatting Results**



**Fig 19-Visualisation of Results**



**Fig 20- Model Comparison**

MaxEnsForest outperforms all baseline models, confirming the relevance of the combination via meta-learning.

# 8. Analysis and Discussion

MaxEnsForest fully leverages the synergy between the diversity of base models and the nonlinear learning capacity provided by the meta-learning neural network. The latter enables effective modeling of complex interactions among the outputs of different estimators, thereby enhancing the overall predictive power of the ensemble.

Experimental results demonstrate the robustness of MaxEnsForest on the make\_classification dataset, achieving an overall accuracy of 95%. The weighted F1-score reflects an optimal balance between precision and recall, a critical metric in real-world scenarios where the impact of false positives and false negatives can be significant. The confusion matrix confirms a clear and distinct class separation, with minimal misclassifications for both majority and minority classes.

Compared to individual base models, MaxEnsForest exhibits superior stability and adaptability, resulting in a significant performance gain. This improvement validates the hypothesis that a structured and complementary combination of diverse algorithms can substantially optimize predictive performance.

Furthermore, in an evaluation conducted on the Breast Cancer dataset, MaxEnsForest outperforms not only all standalone base algorithms but also advanced stacking approaches that combine Random Forest and AdaBoost, as illustrated in the figure above.

# 9. Conclusion

The MaxEnsForest approach represents a substantial advancement in machine learning by synergistically combining the strengths of heterogeneous ensemble methods within a modular and optimizable architecture.

Through the strategic integration of diverse base estimators including Random Forest, SVM, AdaBoost, among others and the deployment of a meta-aggregation model, MaxEnsForest demonstrates exceptional generalization capabilities, particularly in scenarios characterized by complex, noisy, or imbalanced data distributions.

Empirical evaluations on both synthetic and real-world datasets validate its superior accuracy, enhanced stability, and increased robustness to distributional shifts, outperforming traditional baseline models.

Moreover, interpretability is significantly enhanced through advanced techniques such as SHAP and feature importance analysis, making the model well-suited for industrial and safety-critical applications where transparency is paramount.

Consequently, the experimental validation of MaxEnsForest establishes it as a pivotal contribution to ensemble learning, facilitating the development of higher-performing predictive models while minimizing learning errors.

# 8. Perspectives

Looking ahead, several promising directions can be explored to further enhance the impact and applicability of the MaxEnsForest ensemble learning framework :

* **Integration with distributed computing frameworks** (such as Apache Spark, Dask, or Ray), to ensure scalability across large-scale datasets and meet the demands of big data environments;
* **Automation of the learning pipeline** using cutting-edge AutoML and meta-learning tools, enabling MaxEnsForest to autonomously adapt to diverse datasets, problem types, and hyperparameter configurations ;
* **Hybridization with deep learning architectures**, for instance by leveraging neural networks as meta-ensembles or incorporating pre-trained models to capture high-level feature representations ;
* **Deployment in embedded or edge environments**, through memory optimization and low-latency inference strategies, facilitating use in resource-constrained systems;
* **Integration into intelligent decision-making systems** (e.g., Industry 4.0, cybersecurity, predictive medicine, algorithmic finance), where high accuracy, robustness, and interpretability are non-negotiable.

As such, MaxEnsForest paves the way for a new generation of intelligent, adaptive, and explainable models capable of meeting the increasingly complex challenges posed by modern artificial intelligence systems.

Disclaimer (Artificial intelligence)

Option 1:

Author(s) hereby declare that NO generative AI technologies such as Large Language Models (ChatGPT, COPILOT, etc.) and text-to-image generators have been used during the writing or editing of this manuscript.

Option 2:

Author(s) hereby declare that generative AI technologies such as Large Language Models, etc. have been used during the writing or editing of manuscripts. This explanation will include the name, version, model, and source of the generative AI technology and as well as all input prompts provided to the generative AI technology

Details of the AI usage are given below:

1.

2.

3.

# References

1. Dietterich, T. G. (2000). Ensemble Methods in Machine Learning. In : Multiple Classifier Systems. MCS 2000.
2. Breiman, L. (2001). Random Forests. Machine Learning, 45(1), 5-32.
3. Kuncheva, L. I. (2004). Combining Pattern Classifiers: Methods and Algorithms. Wiley-Interscience.
4. Rokach, L. (2010). Ensemble-based classifiers. Artificial Intelligence Review, 33(1), 1-39.
5. Brown, G. (2004). Diversity in Neural Network Ensembles. PhD Thesis, University of Birmingham.
6. Fernández-Delgado, M., Cernadas, E., Barro, S., & Amorim, D. (2014). Do we need hundreds of classifiers to solve real world classification problems? Journal of Machine Learning Research, 15, 3133-3181.
7. Lakshminarayanan, B., Pritzel, A., & Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. Advances in Neural Information Processing Systems, 30.
8. He, H., & Garcia, E. A. (2009). Learning from imbalanced data. IEEE Transactions on Knowledge and Data Engineering, 21(9), 1263-1284.
9. Tang, B., Li, S., & Zhao, C. (2024). Predicting the Performance of Students Using Deep Ensemble Learning.
10. Journal of Intelligence, 12(12), 124.
11. Zhao, S., Zhou, D., Wang, H., Chen, D., & Yu, L. (2025). Enhancing Student Academic Success Prediction Through Ensemble Learning and Image-Based Behavioral Data Transformation. Applied Sciences, 15(3),
12. Liu, N., Wang, J., Zhu, Y., Wan, L., & Li, Q. (2024). Improving imbalance classification via ensemble learning based on two-stage learning. Frontiers in Computational Neuroscience, 17, 1296897.
13. Dinsa, E. F., Das, M., Abebe, T. U., et al. (2024). Automatic categorization of medical documents in Afaan Oromo using ensemble machine learning techniques. Discover Applied Sciences, 6, 575.
14. Mukhlashin, P. A. R., Fitrianto, A., Soleh, A. M., & Wan Muhamad, W. Z. A. (2023). Ensemble learning with imbalanced data handling in the early detection of capital markets. Journal of Accounting and Investment, 24(2), 600-617.
15. Huang, D., Liu, Z., & Wu, D. (2024). Research on Ensemble Learning-Based Feature Selection Method for Time-Series Prediction. Applied Sciences, 14(1), Article 40.
16. Al-Huqail, A. A., Islam, Z., & Al-Harbi, H. F. (2024). An ML-Based Ensemble Approach for the Precision Classification of Mangroves, Trend Analysis, and Priority Reforestation Areas in Asir, Saudi Arabia. Sustainability, 16(23), 10355.
17. Raza, K. (2019). Improving the prediction accuracy of heart disease with ensemble learning and majority voting rule. In U-Healthcare Monitoring Systems (pp. 179-196). Elsevier.
18. Bandi, R., Likhit, M. S. S., Reddy, S. R., Bodla, S. R., & Venkat, V. S. (2023). Voting classifier-based crop recommendation. SN Computer Science, 4(5), 516.
19. Kumar, M., Singhal, S., Shekhar, S., Sharma, B., & Srivastava, G. (2022). Optimized stacking ensemble learning model for breast cancer detection and classification using machine learning. Sustainability, 14(21),13998.
20. Mungoli, N. (2023). Adaptive Ensemble Learning : Boosting Model Performance through Intelligent Feature Fusion in Deep Neural Networks. arXiv preprint arXiv:2304.02653. Sipper, M. (2023). Classy Ensemble : A Novel Ensemble Algorithm for Classification. arXiv preprint arXiv:2302.10580
21. Rosales, R., Popov, P., & Paulitsch, M. (2023). Evaluation of Confidence-based Ensembling in Deep Learning Image Classification. arXiv preprint arXiv:2303.03185.
22. Nam, G., & Lee, J. (2024). Ex Uno Pluria: Insights on Ensembling in Low Precision Number Systems. arXiv preprint arXiv:2411.14860.Zhou, X. (2023).Imbalanced ensemble learning leveraging a novel data-level diversity metric. Computers & Industrial Engineering.
23. Mabumbi, L. N., Katalay, P. K., Lubanzadio, R. K., Lutumba, G. K., Mbala, G. N., Kiala, A. K. W., ... & Salu, G. K. (2025). New Approach Based on the Ensemble Learning Estimator to Maximize Accuracy. Journal of Advances in Mathematics and Computer Science, 40(3), 35-41.