International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Agri-Eco Predict: Minimizing Carbon Intensity in Ensemble Prediction Model for Agricultural Product Price

R. Ragunath[1], R. Rathipriya[2*]

Submitted: 06/05/2024 **Revised**: 19/06/2024 **Accepted**: 26/06/2024

Abstract: Computational loads of Ensemble prediction models (EPMs) are discussed in this paper with a special emphasis on their carbon footprints or carbon emissions. It is purposefully carried out to determine and reduce the carbon emissions caused by EPMs for forecasting Agricultural product prices (APP). Random Forest (RF) and Gradient Boosting (GBR), as well as combinations of both with an adaptive weighted strategy, were included in the experimental study that assessed energy consumption and carbon emissions of EPMs. Carbon emissions were significantly reduced while maintaining prediction accuracy through optimising these models on CPU and T4 GPU platforms. For instance, for optimized RF models on CPU; there was a decline in carbon emission from 3.576e^{-07} kgCO_2e to 1.793e^{-07} kgCO_2e, while Mean Squared Error (MSE) improved from 3.014 to 2.189 respectively. Similarly, after optimization, GBR models on GPU no longer changed their carbon footprint but changed MSE significantly. The findings indicated that it is possible to mitigate the carbon output without affecting the accuracy of predictions using hyperparameter optimization based EPM.

***Keywords**: Agricultural Product Prices, Carbon Footprint, Ensemble Prediction Models, Green AI, Energy Consumption, Sustainability*

1. Introduction

Accurate price forecasts are important in the agricultural industry, shaping decisions from farm to market. Popular forecasting models such as Ensemble prediction models (EPMs) have been preferred by many for APP due to their reliability and high accuracy. However, they come with computational costs resulting in significant energy consumption contributing to global warming. The objective of this paper is to explore approaches for assessing and mitigating the carbon intensity of EPMs so as to make them more environmentally sustainable.

1.1 Ensemble Prediction Model

It compiles predictions from several basic prediction models to create one single predicted output. The mathematical formulation of an EPM can be illustrated by the equation: Let {h1(x),h2(x),…,hM(x)} be M base prediction models, where hi(x) represents the prediction of the ith model for an input x The final ensemble prediction y^(x) is given by: y^(x)=f(h1(x),h2(x),…,hM(x)) where 'f' is a function that combines the predictions of the base models. The specific form of 'f 'depends on the type of ensemble method used. These ensemble methods falls into three common types: A popular example of bagging is Random Forests. What bagging does is generate multiple versions of a predictor by training on different random subsets of the training data and then aggregates their predictions.

- Popular boosting algorithms include AdaBoost, Gradient Boosting Machines (GBM), and XGBoost. Boosting is an ensemble method that sequentially trains models, each correcting the errors of its predecessor.
- Stacking is an ensemble method that combines multiple models by using another model (meta-learner) to make the final prediction and learns how to best combine the base models.

1.2 Measuring Carbon Intensity

The following methods are used for measuring the carbon intensity of EPMs:

- **Monitoring CPU/GPU Usage:** Done using system performance tools;
- **Power Consumption:** Kilowatt hour (kWh) measured using power meters or estimated from hardware specifications;
- **Execution Time:** The total time taken by the model to run predictions.

1.3 Carbon Emission Factors

Energy consumption data is transformed into carbon emissions by employing regional specific emission factors (kg CO_2e per kWh), which consider the local energy mix.

This can be represented in equation 1.:

[1*]*Ph.D Research Scholar, Periyar University, Salem- 11. Email: rrcspu@gmail.com*
[2]*Assistant Professor, Periyar University, Salem- 11. Email: rrcspu@gmail.com*
**** Corresponding Author**: R. Rathipriya*
**Email: rathipriyar@gmail.com*

$$Carbon\ Emission(kg\ CO_2e = Energy\ Consumption\ (kwh) \times Emission\ Factor\ (kg\ CO_2e\ per\ kwh)$$ ----------(1)

1.4 Reducing Carbon Intensity using Model Optimization Techniques

- **Algorithmic Improvements:** This approach helps to reduce the computational complexity of the algorithm.
- **Efficient Coding Practices:** Implementing code that avoids unnecessary computations.
- **Model Pruning:** Reducing model size by removing excessive elements while maintaining the same level of performance.

1.5 Hardware and Infrastructure

- **Efficient Hardware:** The use of energy-efficient hardware, such as low-power CPUs and GPUs.
- **Cloud Computing:** Using cloud services that rely on renewable energy sources and offer expandable computer resources.
- **Edge Computing:** Placing models on edge devices to minimize data transfer and central processing demand.

1.6 Scheduling and Workload Management

- **Task Scheduling:** Executing computational problems when less power is required during off peak periods.
- **Dynamic Scaling**: Changing computing power based on workload prevents oversupplying.

The upcoming sections of this work are prepared as follows: Section 2 delivers essential background information. Section 3 outlines the proposed research work, and Section 4 explores the results of the proposed work and engages in detailed discussions of it. In conclusion, Section 5 presents a comprehensive study of this work.

2 Review Literature

EPMs aggregate the outputs of several base models to ensure improvement in prediction accuracy and robustness. RF and GBR are the most applied techniques in agricultural forecasting, since they turn out to be very efficient for treating large datasets with complex nonlinear relationships. Research works of Breiman, 2001, and Friedman, 2001 showed that RF and GBR work very well for different agriculture predictive tasks.

Another important factor in improving model performance is the optimization of the hyperparameters of these models. Much research has focused on methods like grid search, random search, and Bayesian optimization. Bergstra and Bengio, 2012, demonstrated that random search often proves to be more efficient than grid search, while Snoek et al., 2012 indicated that Bayesian optimization represents a principled way to find the optimal hyperparameters. These optimization techniques are accordingly of crucial importance to ensure both accuracy and computational efficiency for EPM methods.

Training and deployment for EPMs are very computationally intensive; hence, they contribute significantly to energy usage and carbon emission. According to Strubell et al. (2019), carbon emissions from machine learning models have been on the rise. Henderson et al. (2020) showed that it is also becoming ever more urgent to develop sustainable practice methods that would be in a position to measure and then reduce energy use by these models. Some research on the carbon footprint of machine learning models has noted the use of energy-efficient hardware such as GPUs and TPUs, providing higher performance per watt than traditional CPUs. In addition, efficient techniques such as model pruning, scaling, and efficient training algorithms reduce energy consumption by up to 100-fold without compromising model accuracy (Han et al., 2015; Jacob et al., 2018).

Sustainable and Green AI works on the conservation of the environment within the realms of Artificial Intelligence and Machine Learning. It deals with the consumptions of energy and the toll on the environment from AI-related technologies, not limited to data centers, machine learning algorithms, and the training of models itself (Rey 2024, Huertas-García 2023). This includes the use of machine learning to speed up processes involved in establishing a circular economy by developing environmentally sensitive machine learning models for use in industries, and an application of AI in the real-time monitoring of territory and environmental resources Huertas-García 2022. Other research investigates how AutoML influences energy consumption and proposes strategies through which improving the energy efficiency of computational processes involved within AutoML can be achieved Castellanos-Nieves 2023; Rey 2024; Tornede 2021. Specifically, these very frameworks and methodologies will contribute to an extended field of Green AI, supporting the development of environmentally friendly AI and machine learning systems.

According to the literature, EPM optimization should not be limited to predictive accuracy but also include energy efficiency and carbon emissions reduction. While RF and GBR proved efficient for agricultural forecasting, their computational requirements should not come at the expense of the environment. Green AI can thus be attained through not only hyperparameter tuning but also by fusing with energy-efficient hardware.

3. Proposed Work

3.1 Design Goal and Problem Statement

The main objective of this study is to reduce the computation complexity inherent in ensemble prediction models used for agricultural product price forecasting in order to reduce energy consumption and therefore carbon emission without jeopardizing the accuracy of the forecast. So far, EPMs have been reliable models compared with base models used today in agricultural price forecasting; however, their large energy consumption and carbon footprint make them unsustainable in the long term.

This, therefore, requires optimization strategies that reduce computational complexity without reducing EPMs' predictive performance. This includes quantifying current computational demands, using hyperparameter optimization methods, and evaluating their impact on energy efficiency and prediction accuracy. The goal of this work is thus to establish scalable and sustainable practices for developing and using EPMs in agricultural economics.

3.2 Optimizing Hyperparameters in Ensemble Prediction Models for Agricultural Product Price Forecasting

First, different ensemble prediction models such as Random Forest (RF), Gradient Boosting (GBR) and Ensemble of both using adaptive weighted strategy are initialized. Initial hyperparameter settings for these models are defined. Functions are developed to calculate the mean squared error (MSE), which measures how accurate the predictions are, and to calculate carbon emissions based on models' energy consumption, to understand their environmental impact. The models are optimized by testing different hyperparameters to find the best settings that result in the lowest MSE. This process is repeated for each model until the best performance is achieved. After optimization, the energy consumption of the models is measured on CPU and GPU platforms. Using this energy consumption data, carbon emissions are calculated. Finally, the optimized models are evaluated by comparing their prediction accuracy and carbon emissions, highlighting improvements in efficiency and sustainability. The detailed steps of the proposed work are given in Pseudocode 1.

Pseudocode 1: Agri-Eco Predict: Optimizing EPMs for Agricultural Product Price Forecasting

```
Step 1: Initialize models: RF (Random Forest), GBR (Gradient Boosting), Ensemble model
        Initialize hyperparameters: RF_params, GBR_params
Step 2: function calculate_MSE(y_true, y_pred):
        MSE = (1/N) * sum((y_true - y_pred)^2)
        return MSE
        function calculate_carbon_emissions(energy_consumption):
        carbon_emissions = energy_consumption * carbon_factor
        return carbon_emissions
Step 3: for each model in [RF, GBR, Ensemble]:
        best_MSE = infinity
    best_hyperparameters = None
    for hyperparameters in model.hyperparameter_space:
        model.set_params(hyperparameters)
        model.train(training_data, training_labels)
        predictions = model.predict(validation_data)
        MSE = calculate_MSE(validation_labels, predictions)
        if MSE < best_MSE:
            best_MSE = MSE
            best_hyperparameters = hyperparameters
        model.set_params(best_hyperparameters)
Step 4: energy_consumption_CPU = measure_energy_CPU(model, data)
        energy_consumption_GPU = measure_energy_GPU(model, data)
Step 5: carbon_emissions_CPU = calculate_carbon_emissions(energy_consumption_CPU)
        carbon_emissions_GPU = calculate_carbon_emissions(energy_consumption_GPU)
Step 6: for each platform in [CPU, GPU]:
    for each model in [RF, GBR, Ensemble]:
        model.train(training_data, training_labels)
        predictions = model.predict(test_data)
```

```
        MSE = calculate_MSE(test_labels, predictions)
        if platform == CPU:
            energy_consumption = measure_energy_CPU(model, test_data)
        else:
            energy_consumption = measure_energy_GPU(model, test_data)
            carbon_emissions = calculate_carbon_emissions(energy_consumption)
 Step 7:  Return ({platform} {model.name},{MSE}, {carbon_emissions}) as output
Note:
Equations:
Mean Squared Error (MSE): MSE = (1/N) * Σ(y_true - y_pred)^2
Carbon Emissions: carbon_emissions = energy_consumption * carbon_factor
Where,
N: Number of data points
carbon_factor: Conversion factor from energy consumption to carbon emissions
```

4. Results and discussions

Data were sourced from **data.gov.in** and pre-processed for EPMs modeling. It includes predictor variables such as year, average rainfall, and month, and, with the target variable being crop's wholesale price index value. The crop chosen for this study is paddy. Addressing the missing value problem, crucial for accurate model outcomes, appropriate imputation techniques were applied to replace missing values. Table 1 presents the hyperparameter setting of RF and GBR models.

Table 1: Hyperparameters Details for EPM

Regression Model	Hyperparameters	Optimal Values
Random Forest Regression (RFR)	n_estimators	Integer: 25
	max_depth	Integer: 10
GBoost Regression(GBR)	learning_rate	Float: 0.3
	max_depth	Integer:10

Table 2: Comparative summary of the initial and optimized EPMs (CPU)

Model	Initial Mean Squared Error (MSE)	Initial Carbon Emissions (kg CO_2e)	Optimized Mean Squared Error (MSE)	Optimized Carbon Emissions (kg CO_2e)
Random Forest (RF)	3.014	$8.5e^{-07}$	2.189	$3.8e^{-07}$
Gradient Boosting (GBR)	3.177	$2.9e^{-07}$	3.205	$2.9e^{-07}$
Ensemble (RF + GBR)	2.962	$3.2e^{-07}$	2.676	$3.1e^{-07}$

Table 2 presents a clear comparison between the initial and optimized models run CPU platform for Mean Squared Error (MSE) and Carbon Emissions. It is observed that there is an improvement MSE reduction and carbon emission reduction for all models. Table 3 shows a clear comparison between the initial and optimized models run T4 GPU platform for Mean Squared Error (MSE) and Carbon Emissions. It is noted that there is an improvement MSE reduction and carbon emission reduction for all models. Both CPU and GPU platforms show improvements in the reduction of MSE and Carbon emissions after hyperparameter optimization for all models. The extension of improvement differ between CPU and T4 GPU platforms, with the T4 GPU platforms showing more significant improvements in MSE.

Table 3: Comparative summary of the initial and optimized EPMs (T4 GPU)

Model	Initial Mean Squared Error (MSE)	Initial Carbon Emissions (kg CO_2e)	Optimized Mean Squared Error (MSE)	Optimized Carbon Emissions (kg CO_2e)
Random Forest (RF)	3.014	$3.6e^{-07}$	2.189	$1.8e^{-07}$
Gradient Boosting (GBR)	3.177	$2.4e^{-07}$	3.205	$1.9e^{-07}$
Ensemble (RF + GBR)	2.962	$1.9e^{-07}$	2.676	$1.6e^{-07}$

4.1 Observations and Discussions

The following sub-section offers an elaborate examination of the outcomes derived from the research, centering on the influence of hyperparameter optimization on model accuracy and carbon footprint. The research was carried out to assess the efficiency of ensemble prediction models (EPMs) when operating on both CPU and T4 GPU systems. Its objective was to unveil the predictive capabilities and sustainability of these models in forecasting APP. The results underscore enhancements in predictive accuracy and diminished carbon emissions. The subsequent points encapsulate the main findings:

4.1.1. Model Accuracy Improvement (MSE Reduction):

- **CPU Models:** Reduction in mean-squared error upon optimization for all models, RF, GBR, and Ensemble, proves an increase in predictive performance.
- **T4 GPU Models:** Although there has been an surge in MSE for GBR upon optimization, RF and Ensemble models depict a reduced MSE, which reflects that the model has generalized better.

4.1.2. Carbon Emissions Reduction:

- **CPU Models**: Optimised models give off less CO_2 emissions than their original counter-parts. Most conspicuously, the GBR model represents the largest drop in emissions, hence ascertaining efficiency with optimisation.
- **T4 GPU Models:** Although there are variations of carbon emission amongst models, on the whole computation on the GPU platform seems to retain a low emission; hence, the environment-friendly side over the CPU.

4.1.3. Platform-dependent Observations:

- **CPU vs. T4 GPU**: T4 GPU generally generates fewer CO_2 emissions per computation compared to a CPU, making it more relevant to green AI development.
- Tasks that are model-specific: These optimization results are not consistent; some of the models tendencies in their MSE values surprisingly change. This calls for progressive monitoring and adjustments in the ways of optimization so that models either remain or improve in performance.

4.1.4. Efficiency and Sustainability:

- Enhanced optimization of EPMs would not only lead to greater accuracy of the models, but also to sustainability through lesser energy consumption and lower carbon footprint.
- As such, efficient strategies shall be implemented for training and deploying models that help minimize the environmental impact while maximizing computational performance.

5. Conclusion:

This research work explored the computational loads and carbon emissions of Ensemble Prediction Models (EPMs) for predicting agricultural product prices (APP). The empirical experiments were conducted using Random Forest (RF), Gradient Boosting (GBR), and their adaptive weighted ensemble model. The energy usage and carbon footprints of these EPMs on both CPU and T4 GPU platforms were evaluated using crop dataset. These results highlighted the potential to decrease carbon emissions and enhance model performance by fine-tuning hyperparameters in these models. This paves the way to develop sustainable AI applications in agricultural prediction. To move forward this research, it is recommended to explore the integration of advanced hardware such as TPUs, optimize multi-objective functions for accuracy and carbon footprint reduction, and evaluate the long-term environmental impacts of optimized EPMs across various sectors beyond agriculture.

Reference:

[1] Ali, A. H. (2023). Green AI for Sustainability: Leveraging Machine Learning to Drive a Circular Economy. Babylonian Journal of Artificial Intelligence, 2023, 15–16. https://doi.org/10.58496/BJAI/2023/004

[2] Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. Journal of Machine Learning Research.

[3] Breiman, L. (2001). Random forests. Machine Learning. Volume 45,Page 5-32, url={https://api.semanticscholar.org/CorpusID:89141

[4] Castellanos-Nieves, D. (2023). Improving Automated Machine-Learning Systems through Green AI, Appl. Sci. 2023, 13, 11583. https://doi.org/10.3390/app132011583

[5] Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. Ann. Statist. 29 (5) 1189 - 1232, October 2001. https://doi.org/10.1214/aos/1013203451.

[6] Han, S., Pool, J., Tran, J., & Dally, W. (2015). Learning both weights and connections for efficient neural network. Advances in Neural Information Processing Systems.

[7] Henderson, P., Hu, J., Romoff, J., Brunskill, E., Jurafsky, D., & Pineau, J. (2020). Towards the systematic reporting of the energy and carbon footprints of machine learning. Journal of Machine Learning Research.

[8] Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., ... & Kalenichenko, D. (2018). Quantization and training of neural networks for efficient integer-arithmetic-only inference. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

[9] Khaki, S., & Wang, L. (2019). Crop yield prediction using deep neural networks. Frontiers in Plant Science.

[10] Pantazi, X. E., Moshou, D., Alexandridis, T., Whetton, R. L., & Mouazen, A. M. (2016). Wheat yield prediction using machine learning and advanced sensing techniques. Computers and Electronics in Agriculture.

[11] Schmidt, D., Marques, M. R., Botti, S., & Marques, M. A. (2019). Recent advances and applications of machine learning in solid-state materials science. npj Computational Materials.

[12] Snoek, J., Larochelle, H., & Adams, R. P. (2012). Practical Bayesian optimization of machine learning algorithms. Advances in Neural Information Processing Systems.

[13] Strubell, E., Ganesh, A., & McCallum, A. (2019). Energy and policy considerations for deep learning in NLP. Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics.