Machine Learning for CO_2 Emission Prediction using Parallel Computing

Hafizah Farhah Saipan Saipol, Syarifah Zyurina Nordin
Malaysia-Japan International Institute of Technology, Universiti Teknologi Malaysia, 54100 Kuala Lumpur
Email: hafizah.farhah@utm.my, szyurina@utm.my

INTRODUCTION

Carbon dioxide (CO_2) emissions are the primary driver of global climate change. Human activities such as burning fossil fuels and land-use changes have raised atmospheric CO_2 from 278 ppm in the 18th century to 420 ppm in 2024 which leads to a 1.1°C rise in global temperature since 1880. The ongoing increase aggravates global warming, heightening climate-related hazards such as high temperatures, rising sea levels, and loss of biodiversity. Climate change mitigation requires accurate and timely CO2 emission forecasting for policy decisions. Machine learning models have emerged as powerful tools for emissions prediction, which offer superior accuracy compared to traditional statistical methods [1]. However, computational efficiency remains a critical bottleneck in operational deployment, particularly when evaluating multiple algorithms or conducting hyperparameter optimization [2].

PROBLEM STATEMENT

Sequential training of multiple ML models for emissions prediction is computationally expensive and time intensive. With increasing availability of multi-core processors, there exists significant potential for performance optimization through parallel computing architectures. Limited works specifically addresses parallel optimization for CO2 emissions prediction models.

RESEARCH OBJECTIVES

The objectives of this research is to:

a) Implement parallel processing framework for training ML models
b) Evaluate parallel performance compared to sequential performance
c) Validate parallelization maintains prediction accuracy.

METHODOLOGY

1. Dataset	1. Source: OWID CO_2 emission dataset 2. Temporal coverage: 1990 – 2023 (34 years) 3. Geographic focus: 8 Southeast Asian Countries 4. Features used: 72 variables including emissions metrics, energy consumption, economic indicators 5. Target variable: Total CO2 emissions (Mt)
2. Data Preprocessing	1. Missing Value treatment: Dropped columns with >20% missing data; applied median imputation for remaining gaps 2. Feature Selection: Random Forest filtering (threshold >0.01), reducing to 23 features. 3. Normalization: StandardScaler for distance-based algorithms (SVR)
3. Machine Learning Models	Five supervised models were developed i) Random Forest: Ensemble robustness ii) Gradient Boosting: Sequential error correction iii) Neural Network: Baseline linear model with regularization iv) Extra Trees: Randomized ensemble approach and feature importance capability v) Support Vector Regression: non-ensemble baseline, allowing comparison with tree-based and neural models
4. Parallel Implementation	Framework: Python 3.10 with joblib 1.3.2 Backend: Loky (process-based parallelism via joblib) Hardware: 4-core CPU, 16GB RAM (Kaggle cloud-based platform)
5. Performance Metrics	Computational efficiency: i) Execution time (T): Total training time (sec) ii) Speedup (S): $S = \frac{T_{seq}}{T_{parallel}}$ iii) Parallel efficiency (Eff): $Eff = \frac{S}{p} * 100\%$ Prediction accuracy are based on: • Mean Absolute Error (MAE) • Root Mean Squared Error (RMSE) • Coefficient of Determinant (R^2) • Mean Absolute Percentage Error (MAPE)

RESULTS

Table 1. Model performance metrics

Model	Parameters (Tuned)	MAE	RMSE	R^2	MAPE
Neural Network (Large)	Hidden_layers = (200, 100, 50) Max_iteration=1000 Learning rate=0.001	3.24	5.38	0.9989	1.77%
Extra Trees (High)	n_estimators=1000, max_depth=20	5.18	9.68	0.9964	2.47%
Random Forest (High)	n_estimators=1000, max_depth=20	7.25	12.33	0.9941	4.85%
Random Forest (Med)	n_estimators=500, max_depth=20	8.05	14.06	0.9923	5.58%
Gradient Boosting (High)	n_estimators=1000, max_depth=7	8.48	14.10	0.9923	7.23%

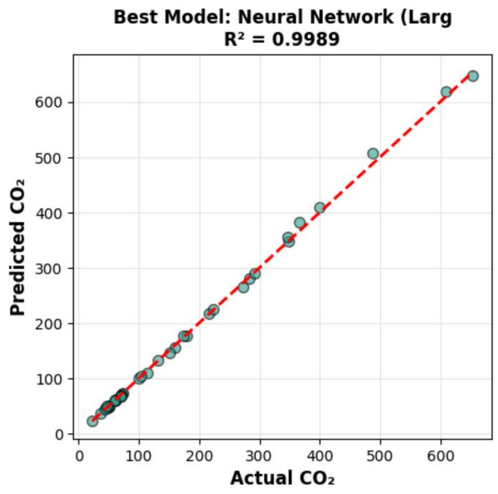

Fig. 1. Actual vs predicted CO2

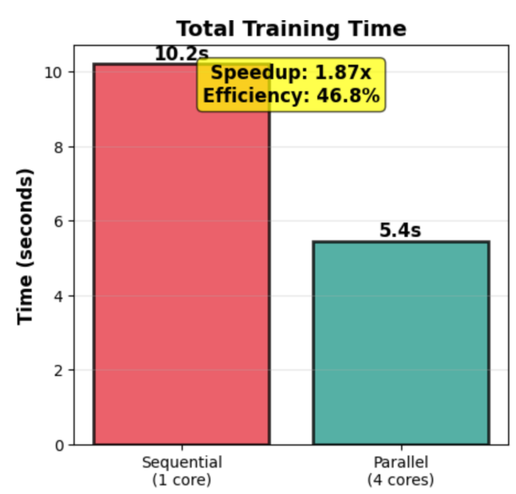

Fig. 2. Sequential and parallel performance

Table 2. Performance comparison

Details	Performance
Sequential time	10.25 sec
Parallel time	5.45 sec
Time saved	4.80 sec (46.8%)
Speedup	1.88x
Parallel efficiency	47.0%
CPU cores used	4
Dataset size	148 training samples
Models trained	9

CONCLUSION

This study demonstrates that parallel computing significantly enhances the efficiency of machine learning models for CO2 emissions prediction without compromising model accuracy. By utilizing Kaggle-based multi-core environment with four CPUs, the parallel execution achieved a 1.88x speedup and 47% parallel efficiency which reduce total computation time by 46.8% compared to sequential execution. Among the nine trained models, the Neural Network (Large) achieved the best predictive accuracy with R^2 = 0.9889 and MAPE = 1.77%, followed closely by ensemble-based models such as Extra Trees, and Random Forest. To conclude, as climate modeling demands increase and datasets grow larger, parallel optimization will become essential rather than optional. This research could provide a solution for accelerating environmental ML applications while maintaining scientific rigor.

REFERENCES

[1] Jin, Y. and Sharifi, A., 2025. Machine learning for predicting urban greenhouse gas emissions: A systematic literature review. Renewable and Sustainable Energy Reviews, 215, p.115625.
[2] Hassanpouri Baesmat, K., Farrokhi, Z., Chmaj, G. and Regentova, E.E., 2025. Parallel Multi-Model Energy Demand Forecasting with Cloud Redundancy: Leveraging Trend Correction, Feature Selection, and Machine Learning. Forecasting, 7(2), p.25.

ACKNOWLEDGEMENTS

We would like to thank MJIIT for the continuous support.