Vol. 1, Issue 11 | Nov 2026 | Open Access — Research Paper

CARBON FOOTPRINT AI – AN INTELLIGENT EMISSION MONITORING AND REDUCTION PLATFORM USING DIGITAL TWIN, MULTI-AGENT REINFORCEMENT LEARNING AND EXPLAINABLE AI

Md Nausheen, Md Mujaheed, K. Karthik
Department of CSE(AI&ML), CMR Technical Campus, Hyderabad, Telangana, India

***Abstract*—**Organisations are asked to cut their carbon emissions but most tools only report past figures give one number with no explanation or are too technical for everyday users. This project presents Carbon Footprint AI a web platform that predicts the daily carbon emission of a facility from ten operational inputs such as electricity, fuel, water, waste and travel. A machine learning model rando forest, gradient boosting or xgboost whichever scores the highest r^2 makes the prediction. The same model drives a digital twin a virtual copy of the facility where users test changes such as less fuel or more solar power before making them and view the result in an interactive 3D factory. Four reinforcement learning agents energy, water, fuel and waste trained with proximal policy optimization recommend actions that lower emissions and cost. shap values show why the model gave each prediction. We built the system in Python with streamlit and sqlite with user accounts guided data entry csv and excel upload and pdf reports. The facility data is synthetic so its accuracy only shows that the pipeline works. On real country level data the same models scored $R^2 = 0.9985$ on a random split but only 0.73 to 0.94 when whole countries were held out which is the more honest figure. Overall the platform gives a complete and explainable first tool for emission planning and its main next step is testing on real facility data.

I. INTRODUCTION

Climate change is one of the biggest problems of our time and organisations such as factories campuses and offices are now expected to measure and reduce the carbon dioxide they release. Doing this well is hard. Emission figures usually come from audits and spreadsheets that are prepared weeks after the activity happened so managers cannot see today emissions or test what would lower them.

Machine learning can estimate emissions from data that a facility already has for example meter readings fuel records and headcount. A digital twin can then be used to ask whatif questions without touching real equipment. reinforcement learning can go one step further and suggest which actions to take. Even so many such tools return a number with no reason. Managers who must justify a decision do not trust a black box and people without a technical background find the inputs confusing.

Carbon footprint ai fills this gap. A user logs in enters a day of facility data or uploads a file and receives a predicted emission a low moderate or high level a whatif simulation recommended actions and an explanation. Each input comes with help text that says what it means and where to find the value. Missing values are never silently replaced by zero and every value is labelled as measured estimated or sample data.

The aim of this work is not to replace a professional carbon audit. It gives a clear starting point so that a user can see roughly where emissions come from and what might reduce them. This paper also reports the limits we found when we tested the system. In particular the facility data is synthetic and a random train and test split on real data gave a much higher score than a split that held out whole countries. We think this finding is important and honest and it shapes our plan for future work.

II. RELATED WORK

A. Machine learning for emission estimation

Tree-based ensemble models are widely used for regression on tabular data. random forests average many decision trees to reduce overfitting gradient boosting builds trees one after another so that each corrects the previous errors and xgboost is an efficient implementation of this idea We use all three because they handle mixed numeric inputs without much tuning and are fast enough for an interactive dashboard. Public datasets such as the Our World in Data CO_2 collection make it possible to test emission models on real national records although not at the level of a single facility.

B. Digital twins and simulation

A digital twin is a virtual copy of a physical system that is kept in step with real data and used to test changes safely. Twins can be physics based or data driven. Our is data driven the trained prediction model plays the role of the simulator so a change to an input immediately gives an updated emission. This keeps the twin simple and means there is a single prediction engine in the whole system.

C. Reinforcement learning and explainable ai

In reinforcement learning an agent learns by trying actions and receiving rewards. proximal policy optimization is a popular and stable algorithm for this and stable baselines3 provides a tested implementation. When several agents act on one shared situation they can be trained so that each learns its own policy while sharing a common reward. A separate problem is trust. shap assigns each input a contribution to one prediction using shapley values from game theory and for tree models it can be computed exactly and quickly. Combining prediction simulation recommendation and explanation in one tool that non experts can use is what carbon footprint ai tries to do.

III. SYSTEM ARCHITECTURE

Carbon footprint ai uses a modular design in which each part does one job and all parts communicate through a central

streamlit application. This makes the system easier to build, test and extend. Fig. 1 shows how the parts connect.

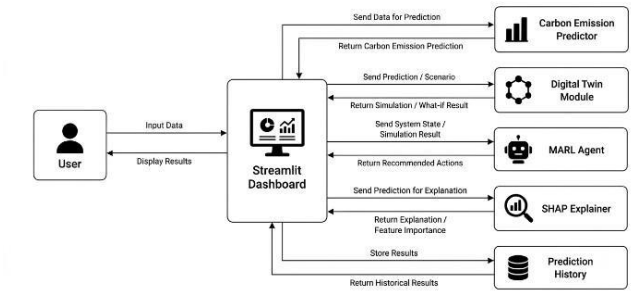


Fig. 1: System architecture of Carbon Footprint AI

A. User layer

This is the web interface. A user registers and logs in then enters facility data by hand uploads a csv or excel file or loads sample data. Results appear on the same page as numbers charts and the 3D view.

B. Streamlit application

The application checks that a user is logged in routes each request to the right module, validates inputs and keeps the session state. We chose streamlit because it is lightweight and connects a python machine learning model directly to a web page.

C. Prediction engine

Inputs are scaled and passed to the best trained regressor which returns the predicted emission in kg co_2 equivalent per day. The result is shown with a low, moderate or High level.

D. Digital Twin engine and 3D view

The twin holds the facility state applies a change and returns the new predicted emission. The 3d view draw five departments production, energy, water, fuel and waste Management and colours them using values returned by this same engine.

E. Multi agent module

Four trained PPO agents look at the twin state and each recommend one action for their area.

F. Explainability module

shap values are computed for the model and shown as a bar chart a waterfall plot a force plot and a global importance chart.

G. Account and history store

User accounts and each user prediction history are stored in a sqlite database. Passwords are stored only as salted hashes and history is always read for the logged in user only.

H. Report generator

The latest prediction the input values the recommended actions and an estimated rupee value of the emission cut are compiled into a downloadable pdf.

IV. METHODOLOGY

The system turns operating data into a prediction a simulation a set of recommendations and an explanation in the steps below.

A. Data collection and preprocessing

Public facility-level emission data does not exist so we generated a synthetic dataset of 10,000 facility days. Each row has ten inputs: electricity (kwh/day), water (litres/day), fuel (litres/day), renewable energy (kwh/day), waste (kg/day), distance travelled (km/day), employees, production (units/day), temperature (degreeC) and humidity (%). The target daily Carbon Emission was calculated from these inputs with controlled random noise. About 1% of the cells 100 per column 1,000 in total were missing and were filled with the column median. The inputs are scaled with a standardscaler, and the data is split 80/20 into training and test sets. For user data every row is checked for missing non numeric and out-of-range values and invalid rows are rejected and reported not filled in.

B. Machine learning models

We trained a random forest, a gradient boosting regressor and when installed, xgboost. They were compared on the test set using mean absolute error root mean squared error and the r^2 score and the model with the highest r^2 was saved automatically with its scaler.

C. Digital Twin simulation

The twin applies a percentage change to one input and asks the trained model for the new emission. Five standard scenarios are provided +10% electricity −10% fuel +10% renewable energy −10% waste and −10% travel. The user can also change production electricity water, fuel and waste by up to ±50% in the 3d view.

The model predicts one total emission not department emissions. To colour the departments each department's emission is estimated from the model's sensitivity raise that department input by 10%, take the change in predicted emission divided by 10% of the baseline and multiply by the baseline emission. Colours are relative red when a department is at least 66% of the largest baseline department yellow from 33% to 66% and green below that. These are estimates not meter readings.

D. Multi agent reinforcement learning

We wrote a custom multi agent gymnasium environment around the twin. four agents energy water, fuel and waste each choose one of five actions do nothing reduce usage optimize equipment use renewable energy or recycle waste. All agents receive the same reward 1.0 × emission reduction + 0.5 × cost reduction so they are encouraged to cooperate. An episode lasts 50 steps. The agents are trained one after another for three rounds of 20,000 timesteps each using ppo.

E. Explainability

A shap tree explainer attributes each prediction to the ten inputs showing which pushed the emission up and which pushed it down.

V. CORE MODULES AND IMPLEMENTATION

A. Technology stack

The system is written in python. Scikit learn and xgboost provide the regressors gymnasium and stable baselines3 the environment and ppo shap the explanations streamlit and plotly the dashboard three.js the 3D view sqlite the storage and reportlab the pdf reports.

B. Facility data entry

A guided three-step form collects facility information resource use and production and environment data. Each input has a help note giving its meaning where to find the value and an example and can be marked as an estimate. The predict button stays disabled until every value is present and valid. A csv or excel file can be uploaded instead with an automatic guess of

which column feeds which input that the user can correct. A demo mode loads clearly labelled sample data that is never saved to the history.

C. Accounts and per-user history

Registration checks the email format rejects duplicate emails and usernames and stores a salted scrypt hash of the password never the password itself. All database queries are parameterised. Each prediction is saved with the user id and the history page loads only that user rows. This is a sensible student project design and has not had a professional security review.

D. Prediction, twin and recommendations

These modules follow Section 4. The twin and the 3d view both call the single trained model so there is no second prediction engine.

E. Personal calculator

A separate tab turns everyday answers commute electricity bill flights into the model inputs. Its inputs lie outside the range of the facility data, so its output is a rough guide only.

F. Reports

The pdf report lists the predicted emission the inputs the agents recommendations and the rupee value of the emission cut at a carbon price the user can set. This value is an estimate not a utility bill saving.

Fig. 2 shows the interactive 3d digital twin at the baseline setting. Each department is coloured by its estimated relative emission and the sliders change production energy Water fuel and waste by a percentage.

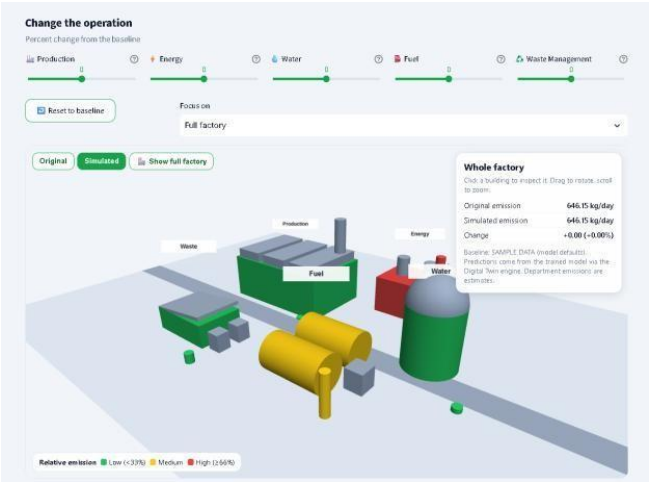


Fig. 2: 3d digital twin view at the baseline (sample data, 646.15 kg co_2-eq/day).

VI. RESULTS AND DISCUSSION

A. Prediction performance on synthetic data

Table I gives the test set results. gradient boosting had the highest r^2 and was selected automatically as the best model.

TABLE I
RESULTS ON THE SYNTHETIC TEST SET (20% SPLIT).

Model	MAE	RMSE	R^2
Random forest	31.95	48.01	0.9398
Gradient boosting	27.31	41.91	0.9541
xgboost	27.28	42.98	0.9517

Because the target was produced from a known formula plus noise a high r^2 here shows that the models can recover that structure. It does not show that they would work on a real facility.

B. Check on real data

To test the same pipeline on real observations we trained the three models on the our world in data co_2 dataset. After removing regional aggregates keeping years from 1990 and dropping rows with missing values 3 166 country year records from 108 countries remained. The inputs were population gdp energy per capita coal oil gas and cement co_2, and primary energy consumption and the target was total co_2. Table II shows the result of a random 80/20 split.

TABLE II
REAL NATIONAL DATA, RANDOM 80/20 SPLIT.

Model	MAE (Mt)	RMSE (Mt)	R^2
Random forest	10.30	44.51	0.9985
Gradient boosting	14.16	53.04	0.9978
xgboost	16.24	86.13	0.9943

To get a more honest picture we repeated the random forest test but kept all rows of a country either in training or in testing so the model had to predict countries it had never seen. Table III shows that the score dropped clearly.

TABLE III
RANDOM FOREST, RANDOM SPLIT VERSUS UNSEEN COUNTRIES.

Test	R^2
Random split	0.9985
Countries held out, seed 1	0.8926
Countries held out, seed 2	0.9382
Countries held out, seed 3	0.7301
Countries held out, mean	0.854

This gap between a random split and a country hold out is a common issue. In a random split other years of the same country are in the training set so the model partly recognises each country. The inputs also include coal oil gas and cement co_2 which are parts of the total being predicted. The experiment therefore shows that the pipeline works on real data, but the 0.9985 figure should not be read as forecasting skill.

C. Digital Twin scenarios

Table IV shows the five standard scenarios applied to one baseline facility profile whose predicted emission was 646.15 kg co_2-eq/day.

TABLE IV
EFFECT OF WHAT-IF SCENARIOS ON DAILY EMISSION (BASELINE 646.15 KG CO_2-EQ).

Scenario	New	Change	%
+10% electricity	680.18	+34.02	+5.27
−10% fuel	630.26	−15.89	−2.46
+10% renewable	639.29	−6.86	−1.06
−10% waste	646.04	−0.11	−0.02
−10% travel	645.15	−1.00	−0.16

Every effect has the direction we expected more electricity raises emissions while less fuel or more renewable energy lowers them. Fuel reduction had the largest benefit of the five. Waste and travel changes had very small effects at this baseline which follows from their small weight in the data formula.

D. Agent recommendations and explanations

The four agents were run on the same baseline facility used in the twin tests. The Energy agent chose the water agent chose the fuel agent chose and the waste agent chose. Together these actions reduced the predicted emission from before to after kg co_2-eq per day a reduction. The final mean episode rewards were energy water fuel waste. As the agents were trained on a simulator built from synthetic data these results show that the method works but have not been confirmed on a real facility.

E. Overall discussion

The pipeline from data entry through prediction simulation recommendation explanation and report works from an engineering standpoint. However the facility results depend on synthetic data and the department emissions in the 3d view are model based estimates. rather than hiding these points we report them they show clearly what must be validated before the system could support real decisions.

VII. FUTURE WORK

The most important next step is to train and test the models on real facility data for example from smart meters and to evaluate them with splits that match real use such as holding out whole facilities or whole months. Department level metering would allow real department emissions in the 3d view instead of estimates.

The rupee value can be replaced with actual savings by using the electricity tariff and fuel price. The personal calculator should be rebuilt on published emission factors. Before public deployment the account system needs a professional security review and the database can be moved to postgresql. Finally the agent should be tested on a pilot facility to measure whether their recommendations reduce emissions in practice.

REFERENCES

[1] S. M. Lundberg and S.-I. Lee, “A unified approach to interpreting model predictions,” in Proc. Advances in Neural Information Processing Systems (NeurIPS), 2017, pp. 4765–4774.

[2] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal policy optimization algorithms,” arXiv:1707.06347, 2017.

[3] L. Breiman, “Random forests,” Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.

[4] J. H. Friedman, “Greedy function approximation: A gradient boosting machine,” Annals of Statistics, vol. 29, no. 5, pp. 1189–1232, 2001.

[5] T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” in Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, 2016, pp. 785–794.

[6] A. Raffin et al., “Stable-Baselines3: Reliable reinforcement learning implementations,” Journal of Machine Learning Research, vol. 22, no. 268, pp. 1–8, 2021.

[7] H. Ritchie, M. Roser, and P. Rosado, “CO2 and greenhouse gas emissions,” Our World in Data. [Online]. Available: https://github.com/owid/co2-data

[8] M. Grieves and J. Vickers, “Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems,” in Transdisciplinary Perspectives on Complex Systems, Springer, 2017, pp. 85–113.