Modelling methane emissions from rice paddies using machine learning

1st Abira Sengupta
School of Compution
University of Otago
Dunedin, New Zealand
enggabira0609@gmail.com

2nd Fathima Nuzla Ismail
Dept. of Mathematics
State University of New York at Buffalo
USA
fathima.nuzla.ismail@gmail.com

2024 39th International Conference on Image and Vision Computing New Zealand (IVCNZ) | 979-8-3315-1877-6/24/$31.00 ©2024 IEEE | DOI: 10.1109/IVCNZ64857.2024.10794457

***Abstract*—Natural methane (CH_4) emissions from wet ecosystems are an important part of today's global warming. Climate affects the exchange of CH_4 between ecosystems and the atmosphere by influencing CH_4 production and oxidation. The net CH_4 exchange depends on ecosystem hydrology and vegetation characteristics. This study focuses on investigating methane emissions specifically from rice paddies in West Bengal, India. The focus is on applying machine learning models to predict methane emissions based on various factors, including wind, temperature, precipitation, and pressure. In this study, we used data from the Copernicus Atmosphere Monitoring Service (CAMS), specifically the CAMS global greenhouse gas reanalysis (EGG4). We applied machine learning models, such as Support Vector Regression, Random Forest, Adaptive Boosting, XGBoost, and Multi-Layer Perceptron, and optimized their *hyperparameters* using the *Optuna* framework in Python. To assess the performance of these models, we used 10-fold cross-validation, which showed that the Multi-Layer Perceptron outperformed the others. Furthermore, this study highlights the relevance of *hyperparameter* adjustment in enhancing model accuracy and finding significant features, which is very useful in environmental monitoring applications.**

***Index Terms*—CAMS, Methane, Machine learning, Optuna, Hyper-parameter Optimisation.**

I. Introduction

Methane is one of the most significant greenhouse gases in the Earth's atmosphere. It can absorb infrared light 15–30 times better than carbon dioxide. As a result, it directly contributes to global warming and climate change [1]. Its concentration has been increasing at the rate of about 1% per year [2]. Methane is not only a significant greenhouse gas but also has an impact on the chemistry and oxidation capacity of the atmosphere. For example, it can change the amount of ozone present in the troposphere layer and act as a sink for chlorine but as a source of hydrogen and water vapour in the stratosphere [2].

The main sources of methane are wetlands, paddy fields, ruminants, biomass burning, etc. Wetland rice fields have recently been identified as a major source of atmospheric methane. Methane emissions from rice field have been influenced by water management, nitrogen, fertilizer use, organic input and rice varieties [3]. The anaerobic fermentation of soil organic matter occurs when the oxygen supply from the atmosphere is cut off to the soil by the flooded rice field. One of the main byproducts of anaerobic fermentation is methane. Through rice plant roots and stems, as well as by diffusion and ebullition, it is released from submerged soils into the atmosphere [2]. The best estimates of methane sources are summarised in Table I, where flooded rice fields emit 50 Tg/yr of methane annually[1]. Methane measurements were initiated under various conditions of paddy fields in West Bengal, India from 1989 onwards [1].

Several techniques, such as Decision Trees (DT) [4], Random Forest (RF) [5], Artificial Neural Network (ANN) [6], Logistic Regression (LR) [7], and Convolutional Neural Networks (CNN) [8], have been used to assess methane emissions from rice field prediction. In most of these cases, the proposal of new methodologies involved the empirical comparison of the performance of the models when applied to methane emissions prediction. However, less attention has been paid to efficient methods for establishing optimal *Hyper-Parameter (HP)* values for model generation and assessing the importance of these HPs on model learning.

Therefore, the main goal of this study is to propose a framework that adopts a well-known HP tuning method to obtain the values required for the optimal performance of an ML model. To this end, we employ the *Optuna Hyper-Parameter Optimization (HPO)* framework [9] to help us not only obtain optimal ML models but also provide insight into the contribution of each hyper-parameter in ML model learning for regression tasks.

II. Materials and Methods

A. Description of the data set

1) Study Area: The entire state of West Bengal, India, is the subject of the study (Figure 1). According to the State-wise Rice Productivity Analysis[2], rice cultivation is spread across 18 districts in West Bengal, divided into different productivity categories.

- The high rice productivity group, with yields exceeding 2500 kg/ha, includes the districts of Burdwan, Birbhum, Nadia, and Hooghly.

[1] 1 Tg = 1 million tons [2].
[2] https://drdpat.bih.nic.in/PA-Table-25-West%20Bengal.htm

979-8-3315-1877-6/24/$31.00 ©2024 IEEE

TABLE I: Estimated sources of methane [2]

Natural	
Wetlands	**120**
Lakes, rivers	20
Oceans	10
Termites	10
Total	160
Anthropogenic	
Mining, processing and use of coal, oil and natural gas	100
Enteric fermentation	80
Flooded ricefields	**50**
Biomass burning	30
Landfills	30
Animal waste	30
Domestic sewage	20
Total	340

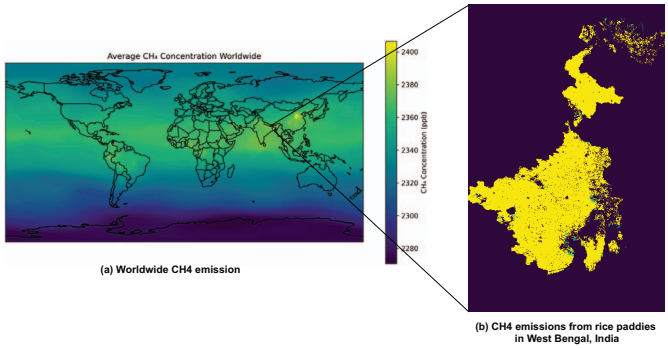


Fig. 1: The Copernicus Atmosphere Monitoring Service (CAMS) global inversion-optimised greenhouse gas fluxes and concentrations over worldwide from 2003 to 2020.

- The medium productivity group, with yields between 2000 to 2500 kg/ha, consists of districts like 24 Parganas, Murshidabad, Bankura, Malda, Midnapur, Dinajpur, and Howrah.

Remote sensing is essential for modeling natural hazards as it delivers real-time, high-resolution data over large areas, improving the ability to analyse and predict events such as floods, wildfires, and earthquakes [10]. Table II presents the district-wise details of cropped areas in each of the flood hazard zones for West Bengal, as prepared by the National Remote Sensing Centre India space research organisation, the Department of Space Government of India. According to this study[3] Hooghly, Murshidabad, Nadia, Paschim Medinipur, Purba Barddhaman districts have the maximum cropped area affected by flood in the years 2000-2020. A map of West Bengal's 18 districts' rice paddies is shown in Figure 1 (b).

2) Data: This dataset is taken from the ECMWF Atmospheric Composition Reanalysis, especially the CAMS global greenhouse gas reanalysis (EGG4), which spans the period from 2003 to 2020.[4]. It focusses on long-lived greenhouse gases like carbon dioxide (CO_2) and methane (CH_4). Emissions and natural fluxes at the surface play an important influence in the atmospheric evolution of these gases.

The chemical loss of CH_4 is characterised by a climatological loss rate, and surface emissions are sourced from several databases. The analysis uses a 4D-Var assimilation approach to assimilate data over a 12-hour period, accounting for the precise timing of observations and model changes inside the assimilation window.

This dataset offers worldwide, three-dimensional, time-consistent fields of atmospheric composition (AC), including chemical species, aerosols, and greenhouse gases like CH_4. It has a temporal resolution of three hours and is organised in a gridded manner with a spatial resolution of $0.75° \times 0.75°$. The data is organised into multiple vertical layers, including surface levels, total columns, model levels, and pressure levels. For our investigation, we received the dataset in NetCDF format. Methane concentrations around the world are shown in Figure 1(a).

This study used twenty-five features from the West Bengal datasets. The key features include *Surface Net Solar and Thermal Radiation, Clear Sky* measured in W/m^2, which represents the balance between incoming and outgoing solar radiation, as well as the net thermal radiation exchange under clear sky conditions. The *CH_4 Column-Mean Molar Fraction* (metric ppb or ppm), measures CH_4 concentration in a vertical air column. *CH_4 Surface Fluxes* (Metric g CH_4/m^2/day) estimate the rate of CH_4 emission. It helps to identify the source, such as wetlands. We used this feature as the target

TABLE II: District-Wise cropped area affected (in Hectares) in different flood hazard zones

District	Very Low	Low	Moderate	**High**	**Total**
ALIPUR DUAR	762	676	0	0	1438
BANKURA	18744	1045	0	0	19789
BIRBHUM	40069	11700	2004	70	53843
DAKSHIN DINAJPUR	49871	11792	24	0	61687
HOOGHLY	66888	44611	8068	**819**	120387
HOWRAH	29869	16246	3796	0	49911
MALDA	50409	46662	3611	0	100682
MURSHI DABAD	129631	36059	15196	**4959**	185845
NADIA	182091	24504	1627	24	208245
NORTH 24 PARGANAS	49043	6271	144	0	55457
PASCHIM ME-DINIPUR	141638	56538	11806	**3638**	213620
PURBA BARD-DHAMAN	152250	36380	8564	**1298**	198491
PURBA ME-DINIPUR	102706	51400	4264	11	158381
SOUTH 24 PARGANAS	59901	5174	11	0	65086
UTTAR DI-NAJPUR	54776	13906	1285	0	69967

[3]https://ndma.gov.in/sites/default/files/PDF/FHA/WB_FloodHazardAtlas.pdf

[4]https://ads.atmosphere.copernicus.eu/datasets/cams-global-ghg-reanalysis-egg4?tab=documentation

variable in this study.

Additional features include the *10m U-Component and V-Component of Wind* (m/s) (east-west and north-south speed). The 2m Dewpoint Temperature signifies the temperature at which air becomes saturated and dew forms at $2m$ height (metric C°). *2m Temperature* (air temperature at $2m$ above the surface). The *Boundary Layer Height* (m) represents the height of the atmospheric boundary layer. *Convective Inhibition* (J/kg) and *Convective Precipitation* (mm) measure energy for convection and precipitation from convection, respectively. *High Cloud Cover* (%) feature is the fraction of the sky covered by high-altitude clouds. *Mean Sea Level Pressure* (hPa) represents the atmospheric pressure at sea level.

The variables include *Potential Evaporation* (mm), *Skin Reservoir Content* (m^3/m^2) for water stored on soil or vegetation, *Skin Temperature* (C°), *Surface Sensible Heat Flux* (W/m^2), and water metrics like Total Column Water (kg/m^2) and *Total Column Water Vapour*. Every feature provides information about the interactions between the surface and atmosphere in the area under study.

3) Data cleansing: The type of ML algorithm and the task it has been applied to inform the pre-processing of the data. The *XGBC*, *RF*, *LGBM*, and *AdaBoost* algorithms did not require data normalisation. However, for the *SVM* and *MLP* algorithms, the data was standardised to a mean of zero and a standard deviation. Furthermore, to handle missing values, correcting inaccuracies and removing duplicates we used Python libraries. We dropped 'Convective precipitation', 'Precipitation type', 'Total precipitation', 'Large-scale precipitation', and 'Total cloud cover' due to a high proportion of null values.

4) Experimental Design: As one objective of this work was to establish better hyper-parameters we adopted the Python Optuna framework [9]. We adopted Optuna as it claims to be an agnostic framework that is not tied to any particular machine learning or deep learning framework. To obtain a more realistic evaluation of ML model performance given the HPs selected by the Optuna framework, 10-Fold Cross-Validation (CV)[5] was used to assess the test performance of each ML model generated across each trial.

To elaborate, for each type of ML algorithm, we ran it over 80 trials, thus generating 80 10-Fold CV ML models. The average Root Mean Square Error (RMSE) value for regression obtained from 10-Fold CV was used as the basis for the Tree-structured Parzen Estimator (TPE) HPO method [11] to adjust the HPs for the next ML model training. Out of these 80 candidate ML models, the best ML model selected was based on the lowest average RMSE depending on if the model was developed for regression respectively. Performance metrics using 10-Fold CV were obtained from this optimal ML model of RMSE and Mean Absolute Error (MAE) for regression [8].

All experiments used the MLP, RF, and AdaBoost algorithms from the Python `scikit-learn` library [12]. The Python XGBoost implementation was adopted from [13] and [14] respectively and conducted on an Intel i7-13700 PC desktop system with 32 GB of RAM running Windows 10.

[5]10-Fold Cross-Validation is a robust technique used to evaluate a model's performance by splitting the dataset into 10 folds. It ensures that every data point is used for both training and testing.

B. Brief description of each of the ML algorithms

In this section, we present our ML model-based methods for methane emission prediction. Machine learning techniques have made a substantial contribution to the efficient outcomes of classification and prediction systems in recent years. While machine learning relies solely on inventory data, it is not dependent on expert knowledge. In this study, we adopted six ML algorithms for methane emission prediction: Random Forest (RF), Adaptive Boosting (AdaBoost), Extreme Gradient Boosting (XGBoost), and Multi-Layer Perceptron (MLP), summarised below:

1) Random Forest: Decision Trees (DTs) are another type of universal function approximator that falls within the category of supervised learning technique [15]. DTs can be applied to both classifications where predicted responses are discrete and regression if expected responses are continuous problems.

A popular extension of Decision Trees is the Random Forest (RF), an ensemble model composed of numerous separately trained DTs [16]. In an RF model, each component tree makes a prediction regarding the classification of the input data; the class receiving the most votes becomes the final classification. RFs can also perform regression; in this case, the final result is obtained by averaging the outputs of the individual trees. The fundamental principle behind the RF technique involves selecting a random subset of features at each node of every tree, while bagging resamples of the initial set of data points to select samples for training each component tree. As the number of trees in the forest increases, the generalization error converges to a limit [5].

2) Adaptive Boosting: Among all the theoretically provable boosting techniques, the most successful one in practical applications has been Adaptive Boosting (AdaBoost). It combines the predictions of several weak learners to create a powerful classifier or regressor.

Its success can be attributed to two things: first, it is very simple; second, AdaBoost has a feature called "adaptivity" that other boosting algorithms do not have [17], [18]. AdaBoost automatically adapts to the strengths of the weak hypotheses generated by the weak learner. It works by increasing the weight of observations that were previously misclassified. This can, in principle, reduce the classification error leading to a high level of precision.

3) Extreme Gradient Boosting : Extreme Gradient Boosting, or XGBoost is a scalable and extremely effective gradient-boosting algorithm that is frequently used for machine learning applications, including regression and classification. XGBoost, as a classifier, is an enhanced implementation of the gradient boosting framework that is optimised for performance and speed [19]. It improves on conventional boosting techniques by maximising model accuracy and computing speed. The XGBoost classifier constructs an ensemble of decision trees sequentially, with each tree aiming to rectify the mistakes of previous ones. Large-scale classification jobs benefit greatly

from its regularisation approaches, which reduce over-fitting. It also has several features, such as support for parallel computation, tree pruning, and handling of missing data. As a regressor, XGBoost is a more advanced version in terms of speed and accuracy. An ensemble of regression trees is constructed by the XGBoost regressor, and each tree is trained to reduce the residual errors of the trees that came before it. It improves generalisation ability by introducing regularisation to control model complexity. XGBoost is also very helpful for large-scale regression issues because it has features like missing value handling, tree pruning, and efficient parallel processing. Regression problems in a variety of applications have come to rely on XGBoost due to its adept handling of huge datasets and complex patterns.

4) Multi-layer Perceptron: Multi-layer perceptrons (MLPs) are neural networks that process a single input with multiple independent weights by combining many neuron units in parallel. To accommodate generic functions, additional degrees of freedom can be provided by adding a second layer of hidden neuron units. Simple classification and regression issues can be resolved with MLPs. In a classification work, for example, the output is the expected class for the input data; in a regression work, on the other hand, the output is the regressed value for the input data. MLP can distinguish data that is not linearly separable or separable by a hyperplane. MLP networks are flexible, general-purpose, nonlinear models made up of several units arranged into multiple layers [20].

C. Results and Discussion

Hyperparameter tuning plays a vital role in influencing feature importance in machine learning by enhancing model performance and identifying the most impactful variables. Correctly adjusting hyperparameters improves the model's capacity to accurately assess feature relevance, resulting in greater predictive accuracy and interpretability. This is important in modeling natural hazard-related predictions [21], [22]. Table III shows details of the HPs used for producing the optimal ML models for regression.

Results of models built using the CAMS WestBengal data set as shown in Table IV that this time the better-performing ML model is *MLP* across all metrics. It has the lowest **RMSE** and **MAE** values and the highest R^2 score compared with other ML models of this work, whereas RF is the least-performing model. We then extracted SHAP values to check the impact of the features on the ML model[6]. Longitude, latitude, and time of year were the most important features according to most of the models.

Figure 2 and Figure 3 presented which *HPs* contribute most to each model learning and performance. *C* is the most important hyperparameter in the *SVR* model (Figure 2). For the *RF* model, the *minimum_sample_split* contributes the most (38%), followed by *maximum_depth* (25%), while the *criterion* contributes only 1% (Figure 2). In the *DTR* model, *max_depth* and *min_samples_leaf* contribute 45% and 30%, respectively, whereas max features and criterion contribute less than 1% (Figure 2). In the *AdaBoost* model, there are two essential hyperparameters, in which *n_estimators* accounting for 59% (Figure 3). *Eta* and *gamma* contribute 20% and 19%, respectively, to the *XGBoost* model, while the *objective* and *eval_metric* contribute less than 1% each (Figure 3). In the *MLPR* model, the *learning_rate_init* has the greatest impact (49%), with the *solver* and *activation* contributing 1% and less than 1%, respectively (Figure 3).

[6]https://github.com/abira-sengupta/Methane-Rice-paddies

TABLE III: Models and HPs for regression.

Model	Hyper-parameters
SVR	*C* = 0.737379, *degree* = 16, *gamma* = 'auto', *kernel* = 'rbf'
DTR	*max_features* = ['sqrt', 'log2'], *max_depth* = 2, 10, *min_samples_split* = 2, 20, *min_samples_leaf*=1, 20, *splitter* = ['best', 'random'], *criterion* = ['squared_error', 'absolute_error', 'friedman_mse', 'poisson']
RF	*criterion* = 'poisson', *max_depth* = 2, *max_samples* = 0.651752, *min_samples_leaf* = 0.189317, *min_samples_split* = 0.449878, *n_estimators* = 23
Ada Boost	*learning_rate* = 0.014532, *n_estimators* = 110
XGBR	*eta* = 0.027458, *eval_metric* = 'rmse', *gamma* = 0.771812, *max_depth* = 5, *max_leaves* = 5, *min_child_weight* = 1, n_estimators = 99
MLP	*activation* = 'logistic', *alpha* = 0.000842, *hidden_layer_sizes* = 14, *learning_rate* = 'adaptive', *learning_rate_init* = 0.003694, *max_iter* = 265, *momentum* = 0.071646, *solver* = 'sgd'

TABLE IV: Regression model test results using 10-Fold CV.

Model	RMSE	MAE	R2
SVR	0.3554±0.0742	0.3305±0.0286	0.6472±0.0334
DTR	0.6694±0.1272	0.5783±0.0600	0.3318±0.0679
RF	0.784±0.1223	0.6299±0.0457	0.2163±0.0338
AdaBoost	0.1219±0.0114	0.2817±0.0153	0.8769±0.0114
XGBR	0.0477±0.0099	0.1488±0.0124	0.9524±0.0057
MLP	**0.0091±0.0024**	**0.0739±0.0088**	**0.9905±0.0037**

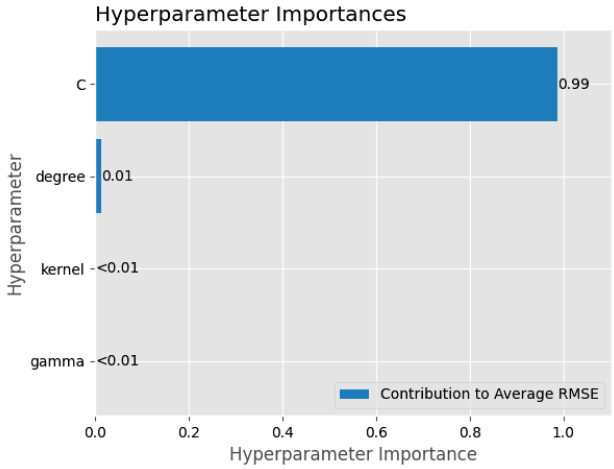
III. Threats to Validity and Limitations

We acknowledge that there are some uncontrolled factors that might have impacted the results reported in our study.

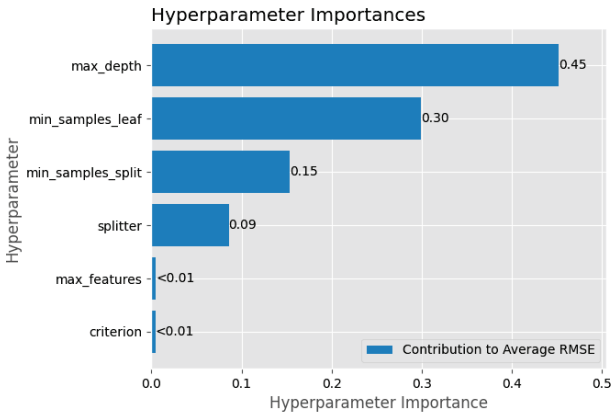
For example, the ML models performed well for regression tasks, but only on small datasets from a small part of India and during a limited time period. Additionally, single-level emissions meteorological data, single-level chemical vertical integrals, and single-level radiation data were downloaded from the CAMS global greenhouse gas reanalysis (EGG4) source. We did not choose multi-level meteorological data, hence we do not have any pressure level or model-level datasets. The impact on model performance, when the analysis is scaled to a complete country using multi-level data, is a question for ongoing research.

Furthermore, several features include null values, thus we removed those columns during data cleaning to improve the ML models' efficacy. In the future, it is expected that data produced by more recent climate forecasting models will mitigate this problem.

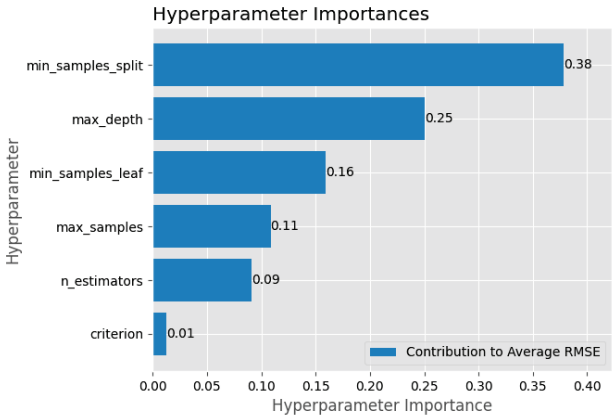
Additionally, adequate resolution of location-specific granular data will improve model performance by including more



SVR

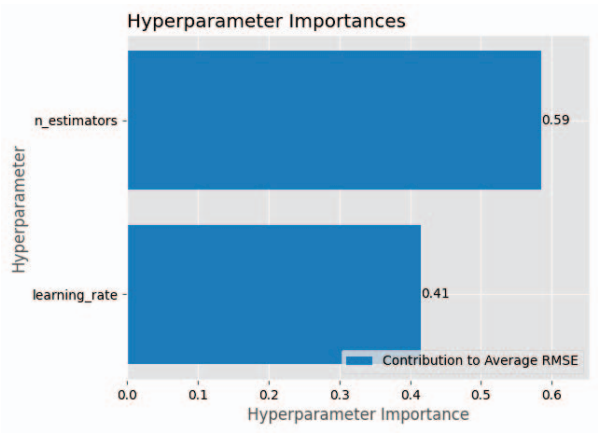


DT

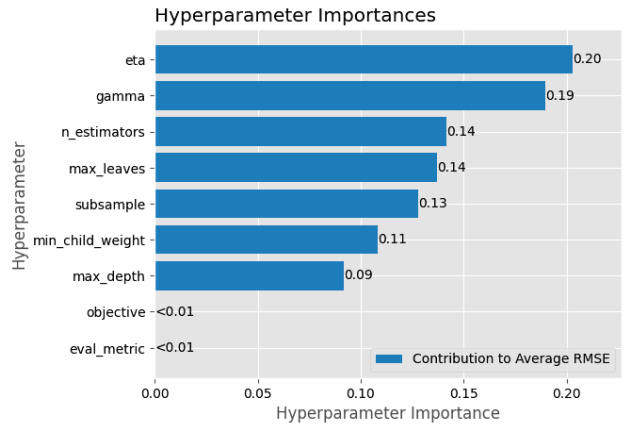


RFR

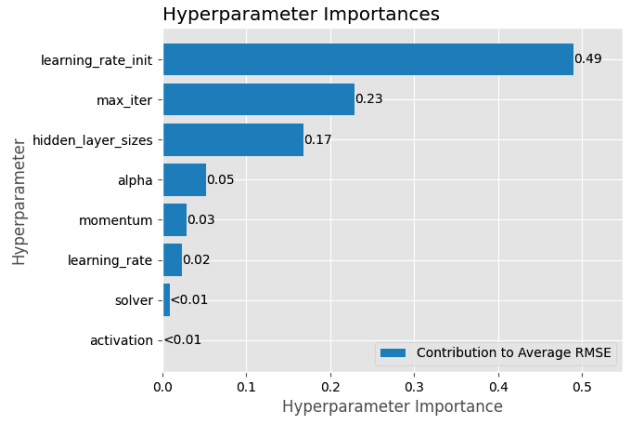
Fig. 2: HP importances for the SVR, RF and DTR models.



ABR



XGBR



MLPR

Fig. 3: HP importances for the ABR, XGBR and MLPR models.

precise location-based data obtained from handheld sensors or wireless sensor networks. Higher resolution satellite or drone-based photos could also help to acquire correct fluxes or concentration of methane emissions but potentially incur a higher financial cost to obtain.

Another risk was that the ML models created would be at risk of overfitting. Our methodology addressed this possible risk by using a 10-fold CV when determining the optimum HPs for a certain ML model type. Using CV was intended to lower the danger of over-fitting and improve the ML model's capacity to generalise to new data.

Finally, the ML models were created using currently available Python modules that implement existing learning techniques. In addition to limiting the use of other suitable ML methods for methane emission analysis, the dependence on these Python libraries neglected to take non-ML-based approaches into consideration. Despite this, our methodology was regularly deployed across many geographic areas, enabling model re-usability. Future testing of our framework in various environmental and geographical contexts can help to validate our study strategy.

IV. Conclusion

In this work, we have proposed a framework that combines current research on generating optimal ML models to predict methane emissions from rice paddies in West Bengal, India. Various machine-learning techniques were employed to estimate methane emissions based on critical factors such as wind, temperature, precipitation, and pressure.

Major findings of our work identified which HPs were the most influential in generating ML models and highlighted that the MLP regression would be a more appropriate ML model for predicting CH_4 emissions among the other models and underscore the importance of selecting appropriate machine learning algorithms and optimizing hyperparameters to enhance predictive performance.

Furthermore, in this study, a selected dataset from CAMS global greenhouse gas reanalysis (EGG4) highlights the significance of particular variables, such as CH_4 surface fluxes to identify sources and measured methane emissions from a specific region. These insights can help policymakers and agricultural stakeholders design more targeted interventions to mitigate methane emissions and contribute to climate change mitigation efforts. Future research could expand this work by integrating more granular data and considering additional environmental factors.

References

[1] N. Purkait, M. Sengupta, S. De, and D. Chakrabarty, "Methane emission from the rice fields of west bengal over a decade," *89.60. Fe; 92.60. Sz; 92.70. Cp*, 2005.

[2] H.-U. Neue, "Methane emission from rice fields," *Bioscience*, vol. 43, no. 7, pp. 466–474, 1993.

[3] J. Wang, H. Akiyama, K. Yagi, and X. Yan, "Controlling variables and emission factors of methane from global rice fields," *Atmospheric Chemistry and Physics*, vol. 18, no. 14, pp. 10419–10431, 2018.

[4] F. Abid and N. Izeboudjen, "Predicting Forest Fire in Algeria Using Data Mining Techniques: Case Study of the Decision Tree Algorithm," in *Advanced Intelligent Systems for Sustainable Development (AI2SD'2019) Volume 4-Advanced Intelligent Systems for Applied Computing Sciences*, pp. 363–370, Springer, 2020.

[5] L. Breiman, "Random Forests," *Machine learning*, vol. 45, pp. 5–32, 2001.

[6] H. Liang, M. Zhang, and H. Wang, "A Neural Network Model for Wildfire Scale Prediction Using Meteorological Factors," *IEEE Access*, vol. 7, pp. 176746–176755, 2019.

[7] P. de Bem, O. de Carvalho Júnior, E. Matricardi, R. Guimarães, and R. Gomes, "Predicting wildfire vulnerability using logistic regression and artificial neural networks: a case study in Brazil," *International Journal of Wildland Fire*, vol. 28, no. 1, pp. 35–45, 2018.

[8] S. Bhatt and U. Chouhan, "An enhanced method for predicting and analysing forest fires using an attention-based CNN model," *Journal of Forestry Research*, vol. 35, no. 1, p. 67, 2024.

[9] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A Next-generation Hyperparameter Optimization Framework," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD'19, (New York, NY, USA), p. 2623–2631, Association for Computing Machinery, 2019.

[10] F. N. Ismail and S. Amarasoma, "One-class classification-based machine learning model for estimating the probability of wildfire risk," *Procedia Computer Science*, vol. 222, pp. 341–352, 2023. International Neural Network Society Workshop on Deep Learning Innovations and Applications (INNS DLIA 2023).

[11] J. Bergstra, D. Yamins, and D. D. Cox, "Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures," in *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, ICML'13, (Atlanta, GA, USA), p. I–115–I–123, JMLR.org, 2013.

[12] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[13] T. Chen and C. Guestrin, "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), pp. 785—-794, Association for Computing Machinery, 2016.

[14] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "LightGBM: A highly efficient gradient boosting decision tree," *Advances in Neural Information Processing Systems*, vol. 30, pp. 3146–3154, 2017.

[15] L. Breiman, J. Friedman, R. Olshen, and C. J. Stone, *Classification and Regression Trees*. Routledge, 2017.

[16] S. Suthaharan, *Machine Learning Models and Algorithms for Big Data Classification: Thinking with Examples for Effective Learning*. No. 36 in Integrated Series in Information Systems, Springer Publishing Company, Incorporated, 1st ed., 2015.

[17] D. L. Shrestha and D. P. Solomatine, "Experiments with AdaBoost.RT, an Improved Boosting Scheme for Regression," *Neural computation*, vol. 18, no. 7, pp. 1678–1710, 2006.

[18] Y. Freund and R. E. Schapire, "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting," *Journal of computer and system sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[19] K. K. Raghunath, V. V. Kumar, M. Venkatesan, K. K. Singh, T. Mahesh, and A. Singh, "XGBoost Regression Classifier (XRC) Model for Cyber Attack Detection and Classification Using Inception V4," *Journal of Web Engineering*, vol. 21, no. 4, pp. 1295–1322, 2022.

[20] M. Khalil Alsmadi, K. B. Omar, S. A. Noah, and I. Almarashdah, "Performance Comparison of Multi-layer Perceptron (Back Propagation, Delta Rule and Perceptron) algorithms in Neural Networks," in *2009 IEEE International Advance Computing Conference*, pp. 296–299, IEEE, 2009.

[21] F. N. Ismail, B. J. Woodford, S. A. Licorish, and A. D. Miller, "An assessment of existing wildfire danger indices in comparison to one-class machine learning models," *Natural Hazards*, 2024.

[22] F. N. Ismail, A. Sengupta, B. J. Woodford, and S. A. Licorish, " A Comparison of One-Class Versus Two-Class Machine Learning Models for Wildfire Prediction in California," *Data Science and Machine Learning*, vol. 1943, pp. 239–253, 2024.