AFRC
AMERICAN FLAME RESEARCH COMMITTEE
INDUSTRIAL COMBUSTION SYMPOSIUM

M ENGINEERING

Using Machine Learning Algorithms to Predict Steam-Assisted Flare Performance

Manikandan Pandiyan[1], Jenna Stolzman[1], and Margaret Wooldridge[1, 2]

[1]*Department of Mechanical Engineering, University of Michigan - Ann Arbor*
[2]*Department of Aerospace Engineering, University of Michigan - Ann Arbor*
{mpandiya, stolzmaj, mswool}@umich.edu

Abstract

The United States Environmental Protection Agency (EPA) reports that 32% of global methane emissions originate from the oil and gas industry. Current EPA regulations require that oil and gas flares maintain a minimum heating value, along with other requirements, to ensure a combustion efficiency greater than 96.5%. It is important to understand how various operational factors, such as the heating value, the exit velocity, and the composition of the flare gas, affect performance. Large-scale flare studies are extremely costly and time consuming, and important inputs may be overlooked during manual data analysis. Instead, machine learning (ML) models are more economically advantageous for studying the various flare operating conditions and the relationships of operating parameters with combustion efficiency. Existing data for flares are largely available via the United States EPA website and various other reports. The objective of this work was to create an ML algorithm that can predict flare performance effectively and accurately using existing steam-assist flare data. To meet the project objective, a supervised ML framework was developed that employed several methods, such as data cleaning and feature engineering, to improve the predictive performance of steam-assist flares. Steam-assisted flare data were used for the study due to the significant role of steam-assist in reducing visible smoke emissions from flares. Steam-assisted flare data were converted to inputs consisting of Reynolds number, momentum flux ratio, steam-to-fuel ratio, dilution ratio, and net heating value of the combustion zone to predict combustion efficiency. An evaluation of several different ML models was performed using several metrics such as the mean square error, the mean absolute error, and the coefficient of determination values (R^2). The trained ML models were determined to be trustworthy (i.e., with good predictive performance) for combustion efficiency with R^2 values greater than 0.90 for some ML approaches. Specifically, the linear-based algorithms yielded the lowest quality models with R^2 values less than 0.77 for the training and testing datasets, and the algorithms designed for non-linear data fitting performed well with the Categorical Boosting model yielding the best performance with R^2 values of over 0.93 for the testing and training datasets. The results demonstrate the utility of ML methods for predicting flare performance and provide useful tools to guide researchers, government agencies, and other stakeholders in understanding the major factors affecting flare performance.

Keywords: steam-assist, machine learning, combustion efficiency, experimental flare data, flare performance, flare design optimization

1 Introduction & Motivation

Industrial flares are essential safety devices that relieve pressure in oil and gas fields, process pipelines, and storage vessels during normal, start-up, shutdown, and malfunction situations [1–3]. Flares vary in design based on operating parameters such as flow rate, pressure level, composition, etc., and may include a method of air- or steam-assist to improve combustion characteristics [4, 5]. In 2018, it was found that there were more than 78,000 flare units within the Permian, Eagle Ford, and Bakken regions in the U.S. with a collective capacity to flare more than 3,500 million cubic feet of natural gas per day [6]. It is assumed within industry and government that flares meeting U.S. Environmental Protection Agency (EPA) design specifications, such as a minimum net heating value, achieve 98% methane destruction efficiency. However, recent reports, such as the work by Plant et al. [1], determined that flaring releases up to five times more methane to the atmosphere compared with EPA estimates in the Permian, Eagle Ford and Bakken regions. They attributed an equivalent efficiency of 91.1% destruction of methane due to unlit flares and inefficient combustion. Clearly, there is a need to understand the various operating inputs that affect flare performance in order to improve the effectiveness of these important technologies.

Large-scale flare studies have been conducted over the past several decades, beginning in the 1980s with the pioneering work by McDaniel and Tichenor [7] and Pohl and Soelberg [8]. It was found that the heating value and velocity of the flare gas and the amount of steam used for steam-assist flares were the main factors influencing combustion efficiency. Specifically, the combustion efficiency for steam-assisted flares decreased only when the flare gas heating value was below 300 British Thermal Units per standard cubic foot (BTU/scf) [7]. Furthermore, McDaniel and Tichenor found that flaring low-energy gasses at high exit velocities could also result in reduced combustion efficiencies [7]. These large-scale studies helped shape the EPA Federal Regulation 40 CFR 63.670 [9], which regulates flaring. Specifically, the minimum net heating value of the combustion zone (NHVcz) must be greater than 270 BTU/scf, which helps ensure a destruction efficiency greater than 98%. Large-scale flare studies continued into the early 2010s with Allen and Torres [10] where they confirmed earlier conclusions that the most impactful variables affecting combustion efficiency are the heating value, the amount of steam used, and the velocity of the flare gas. The major drawbacks with large-scale flare studies are their high costs and lengthy execution time. An added concern is that important flare operational parameters may be overlooked during manual data analysis because of the large volume of data collected (e.g., hundreds of gas composition measurements as a function of time for each test condition). At the same time, flare data are also oftentimes non-linear and sparse (e.g., missing some parameters such as smoke or particulate emissions), adding complexity to traditional data analysis methods. Fortunately, machine learning (ML) tools are built to handle such datasets and have proven to be useful for identifying the relationships between operating parameters and flare performance.

Recently, ML models have been applied to existing flare data for predictive modeling of combustion efficiency (%CE) and the percent opacity (%Opacity) of flares under various operating conditions. Alphones et al. [11] used Response Surface Modeling (RSM) and Sigmoid models to develop predictive models of steam- and air-assisted flares based on experimental flare study data from the years 1983 to 2016. The quadratic RSM and Sigmoid models expressed %CE and %Opacity as a function of operating variables and yielded $R^2 > 0.90$ and 0.87, respectively. Damodara et al. [12] developed artificial neural networks (ANNs) based- %CE and %Opacity prediction models using the Levenberg-Marquardt backpropagation algorithm and demonstrated $R^2 > 0.95$ for both air-assisted and steam-assisted flare data, and showed that the experimental flare study data from the years 1983 to 2014 were in good agreement with the models' outputs. Recently, a zone-based modeling strategy was proposed to predict %CE and %Opacity of steam-assisted flares by Lou et al. [13]. First, the experimental flare data were divided into two zones on the basis of the carbon-to-hydrogen ratio. Zone-based predictive models using Random Forests and Catboost algorithms were developed to demonstrate superior model performance even when extreme values were presented (i.e., all original data were included in the model development and testing).

The random forest and neural networks algorithms have shown the best performance in predicting the combustion efficiency of flares thus far. However, few studies have tested other ML models. The objective of this study was to determine if further improvements in ML methods could be achieved for flare data. The approach used existing steam-assisted flare data and transformed the most important variables into features to predict combustion efficiency. Then, multiple ML algorithms were used with the dataset to determine the most suitable model for predicting combustion efficiency. Metrics such as the mean square error, the mean absolute error, and the coefficient of determination (R^2) were used to compare between models. To the authors' knowledge, different boosting ensemble learning methods have not been tested on existing flare data. These models have exceptional generalization and regularization capabilities, which may prove superior for these types of non-linear and sparse datasets. Additionally, they have the ability to handle skewed distributions, enabling better performance when dealing with abnormal flare conditions (such as conditions that fall outside of Federal Regulations). The potential combination of boosting algorithms with feature engineering techniques could yield promising results in accurately predicting combustion efficiency.

The remainder of this paper is organized as follows. Section 2 provides an overview of the methods used to collect data, create features, and "clean" the data to develop ML models. Section 3 explains the proposed ML framework for predictive modeling of steam-assisted industrial flare data. It also describes the steps involved in building an ML model. Section 4 provides the findings and a detailed comparative analysis of the ML algorithms. Section 5 summarizes the results of the study and provides recommendations for future research in the field of industrial flare design.

2 Data Collection, Pre-Processing, & Feature Engineering

The data for flares are largely available through the United States EPA website and various other reports. For the current work, data were first collected from various sources [2, 6, 13–17] that include important variables in flaring, such as flow rate, exit velocity, composition and heating value of the flare gas, cross wind speed, and combustion and destruction efficiencies. There are a total of 425 data points between seven datasets. The raw data variables, such as flare gas exit velocity, were transformed into features, such as Reynolds number, to be used as input to the ML models. The as-reported variables from the datasets are listed in Table 1. The input features based on the as-reported data are summarized in Table 2. The main output variables used to assess flare performance are combustion and destruction removal efficiency (% CE and % DRE).

Table 1: Raw Data Variables

Variable	Description
ρ_f	Density of flare gas (kg/m3)
ρ_{air}	Density of air (kg/m3) at standard conditions (P = 1 atm and T = 298 K)
ρ_{air}	Density of air (kg/m3) at standard conditions (P = 1 atm and T = 298 K)
ρ_{air}	Density of air (kg/m3) at standard conditions (P = 1 atm and T = 298 K)
v_f	Exit velocity of the flare gas (m/s)
v_{air}	Velocity of cross wind (m/s)
D_{eff}	Effective diameter of the flare (m)
μ_f	Dynamic viscosity of the flare gas (Pa-s), estimated using the major components of the flare gas
$\dot{m}_{diluent}$	Mass flow rate of the diluent (nitrogen or carbon dioxide) (kg/s)
$\dot{m}_{fuel}$	Mass flow rate of flare gas (kg/s)
$\dot{m}_{steam}$	Mass flow rate of steam (kg/s)
NHV_{CZ}	Net heating value of the combustion zone gas (BTU/scf)
$\#C$	Carbon number of hydrocarbon in the flare gas
$\#H$	Hydrogen number of hydrocarbon in the flare gas
$X_{C_xH_y}$	%Volume of hydrocarbon in the flare gas

Table 2: Input features used for ML models

Features	Equation
Reynolds number, Re	$Re = \frac{\rho_f v_f D_{eff}}{\mu_f}$
Momentum flux ratio, MFR	$MFR = \frac{\rho_{air} v_{air}^2}{\rho_f v_f^2}$
Dilution ratio, D	$D = \frac{\dot{m}_{diluent}}{\dot{m}_{fuel}}$
Steam ratio, S	$S = \frac{\dot{m}_{steam}}{\dot{m}_{fuel}}$
Net heating value of the combustion zone gas, NHVcz	$NHV_{CZ} = \frac{NHV_{VG} * Q_{VG} + Q_f * NHV_f + Q_p * NHV_p}{Q_{VG} + x_a * Q_a + x_s * Q_s + Q_p}$
Carbon-to-hydrogen ratio, CHR	$CHR = \frac{\sum \#CX_{C_xH_y}}{\sum \#HX_{C_xH_y}}$

Statistical analysis of the experimental data was conducted and the results are presented in Figure 1. Specifically, Figure 1 shows the distribution of the experimental steam flare data for both independent variables (features) and dependent variables (outputs). The box plots show the non-uniform distribution of the experimental data and the regions of sparsity. Figure 2 depicts the pairwise relationships between the variables in the steam-assist flare dataset. Each scatter plot in the matrix represents the relationship between two variables, with one variable plotted on the x-axis and the other variable plotted on the y-axis. The plots suggest that there are some statistically significant correlations between the independent variables, but none of them is particularly strong. Therefore, it can be said that there is no multicollinearity problem with the results of these data. The pairwise distribution plots also reveal outliers present in the data (i.e., individual points that are far from the main cluster of points).

The scatter plot matrix includes asterisk notation to indicate the significance level of the results. A single asterisk represents statistical significance at the 0.05 level (2-tailed), indicating a confidence level 95%. Two asterisks represent significance at the 0.01 level, with a confidence level 99%, while three asterisks indicate significance at the 0.001 level, with a confidence level of 99.9%, making it the highest level of significance. As shown in Figure 2, most of the data show little correlation when considered in pairwise relationships. Some key exceptions include the relationships between CE and DE. In particular, there is a strong correlation between CE and DE ($R > 0.86$). This supports the use of CE as a surrogate for DE, which is valuable, as DE is less frequently reported.

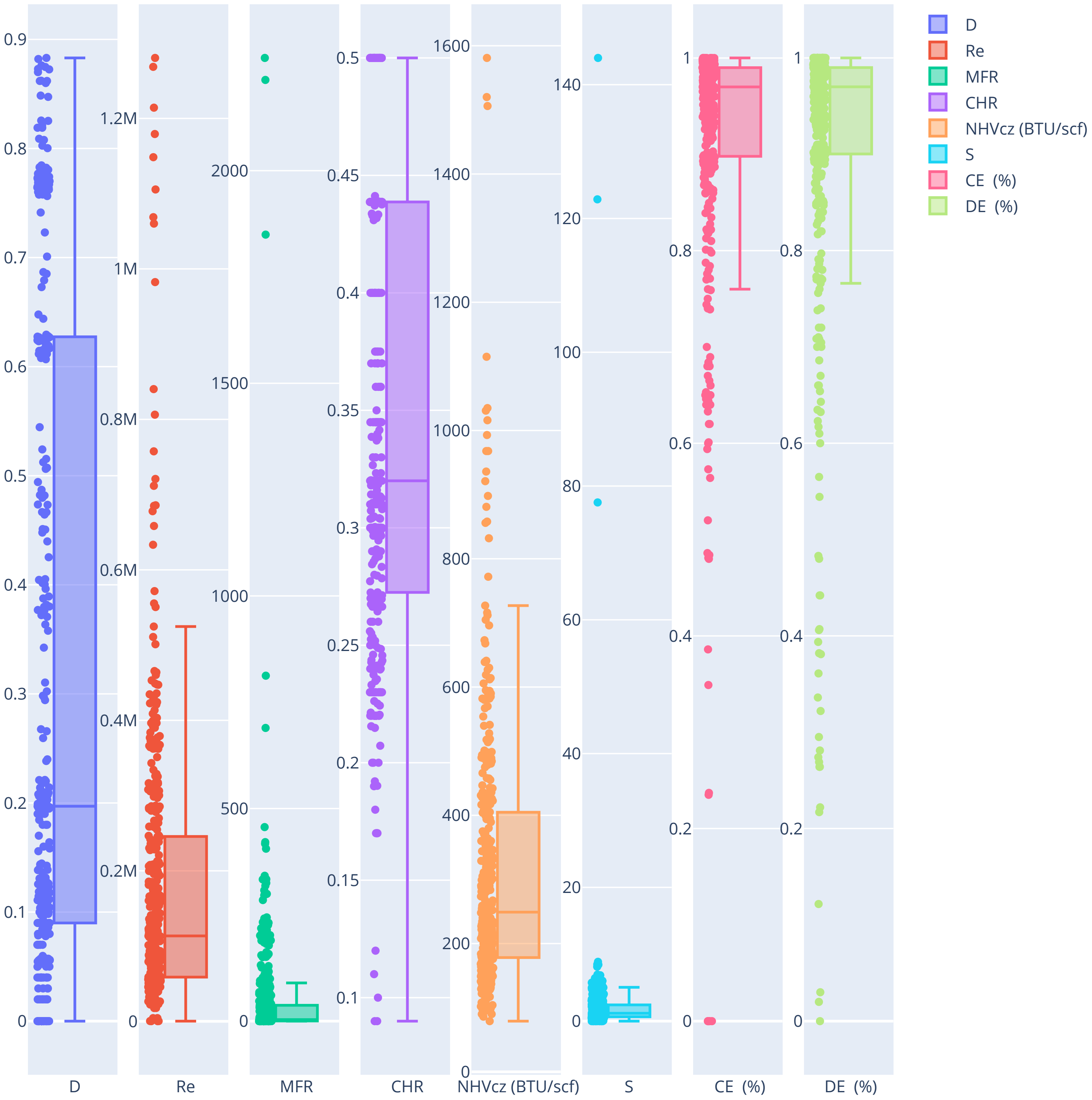


Figure 1: Results of statistical analysis of steam-assist flare data: Distribution of features and performance data (i.e., box plots)

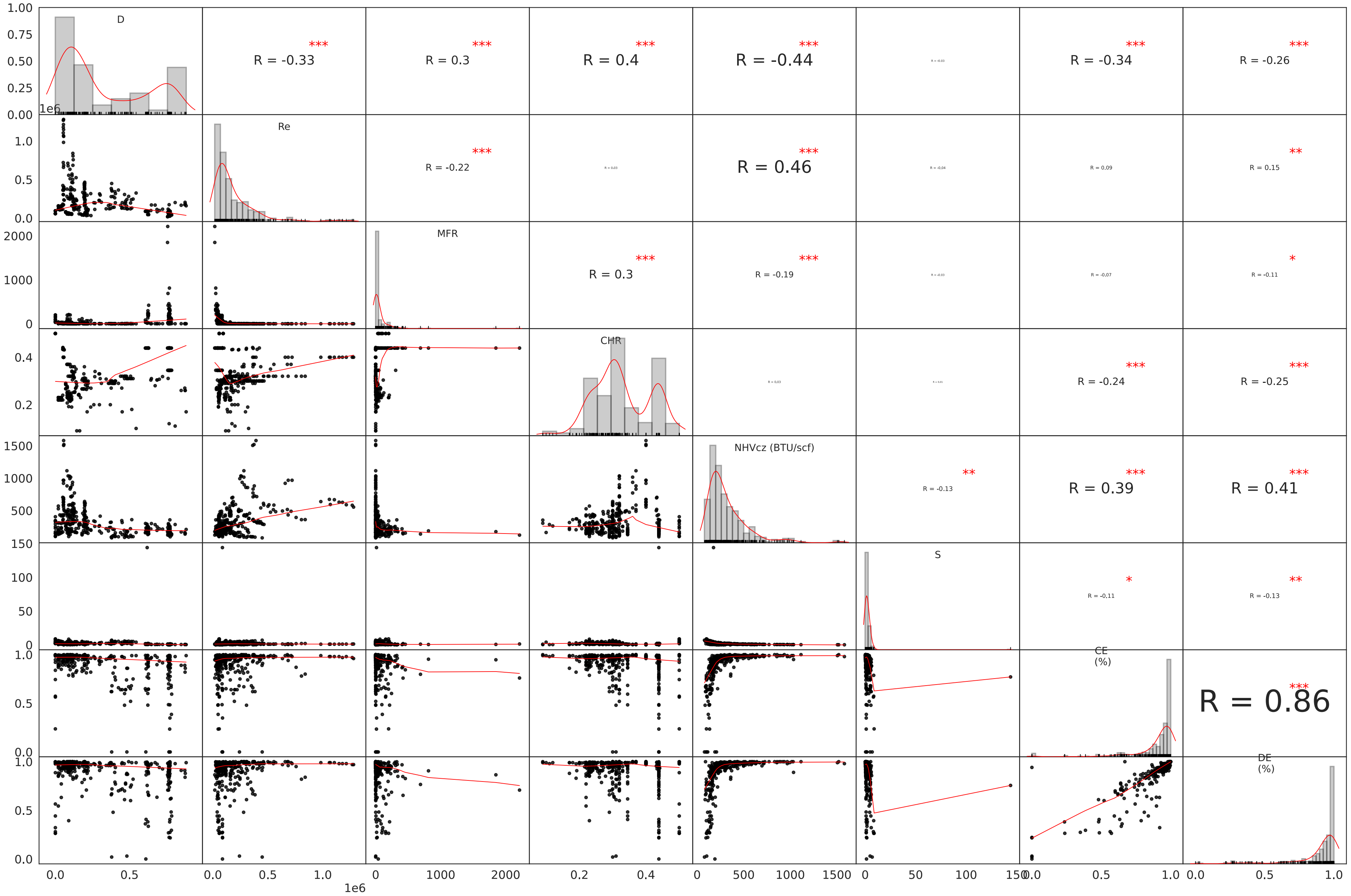


Figure 2: Pairwise distribution of features and performance data

3 Methodology

The research methodology was carried out as illustrated in Figure 3. There were several stages in the framework, which are given as follows. The raw data were collected from various sources. Preprocessing of the data was the next step to prepare the experimental steam-assist flare data for modeling and analysis by cleaning, transforming, and structuring. Next, feature engineering (the process of extracting and transforming features or independent variables from experimental data to prepare for modeling and analysis) was applied. This step requires domain expertise and statistical tools to create features that will help ML algorithms better understand and learn from the data. Feature engineering involves various tasks, including choosing relevant variables, handling missing values, scaling or normalizing variables, and creating new variables through transformations or combinations of existing variables. Data cleaning techniques are then performed to improve the performance results of the ML model. Specifically, in this work the following steps were taken: null rows were removed, scaling and normalization to the features was conducted, and variable transformation techniques were adopted. The fully pre-processed data were then employed for the predictive modeling. The features that were extracted and preprocessed from the raw data were then used as inputs to the supervised ML algorithms. The primary objective of this step was to improve the accuracy of ML algorithms by providing high-quality input data.

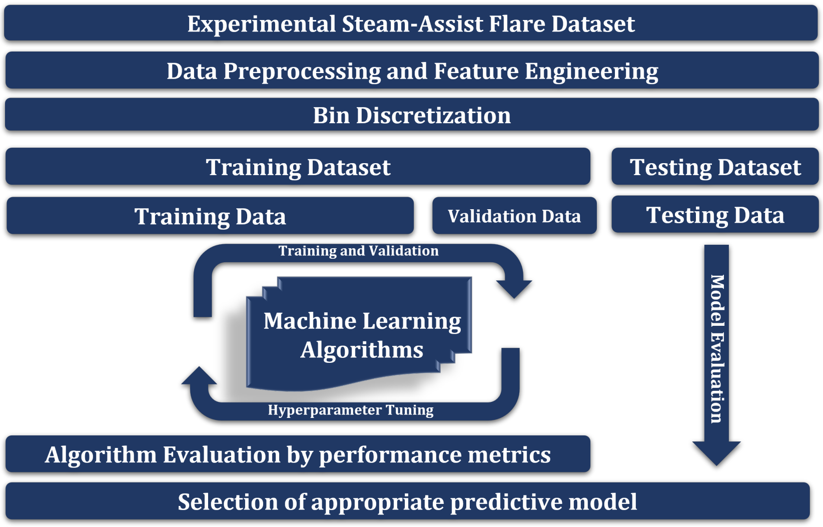

Experimental Steam-Assist Flare Dataset
Data Preprocessing and Feature Engineering
Bin Discretization
Training Dataset
Testing Dataset
Training Data
Validation Data
Testing Data
Training and Validation
Machine Learning Algorithms
Hyperparameter Tuning
Model Evaluation
Algorithm Evaluation by performance metrics
Selection of appropriate predictive model

Figure 3: Machine Learning Model Development Process

Bin discretization is a crucial step in handling non-uniform experimental data for ML model development. By dividing the dataset into distinct bins, outliers can be preserved, allowing for the identification of non-uniform and non-linear features, which is valuable for modeling the flaring process under different operating conditions. Therefore, by using bin discretization, different models can be built for each bin according to the flaring process. To assess the performance of ML models, the flare data are split into two sets: a training dataset and a testing dataset, with a ratio of 85:15 respectively. The data are assigned from the various sources for steam assist flares, and to prevent any source bias, the data are shuffled before being split. This ensures that each source contributes to both the model training and testing datasets.

3.1 Machine Learning Algorithms

Machine Learning is a subset of Artificial Intelligence (AI) that incorporates several learning paradigms such as supervised learning, unsupervised learning, and reinforcement learning. With the emergence of "Big Data" in many industrial sectors, the adoption of ML algorithms can improve the efficiency of data analysis and processing for quick decision making [18]. In this work, several algorithms were explored for flare predictive analytics, which included multiple linear regression (MLR), support vector regression (SVR), random forests (RF), gradient boosting (GradBoost), extreme gradient boost (XGBoost), and categorical boost (CatBoost). The key attributes of the different approaches are presented here.

3.1.1 Multiple Linear Regression

Regression analysis is a widely accepted statistical approach to predict the relationship between one or more independent variables (i.e. predictors) and the dependent variables (i.e. predicted values). Multiple linear regression (MLR) is mainly used to understand the change in the dependent variable y when there is a change in the independent variables p $(x_1, x_2, \ldots, x_p)$. MLR can be represented as follows:

$$y_i = f(x_{i1}, x_{i2}, ..., x_{ip}) \tag{1}$$

$$y_i = \beta_0 + \beta_1 * x_{i1} + \beta_2 * x_{i2} + + \beta_k * x_{ik} \tag{2}$$

where,
β_i are the model fit coefficients.

3.1.2 Support Vector Regression

Support vector regression (SVR) is an effective method for both linear and non-linear regression types, and is commonly used for curve fitting and prediction. In the standard formulation of support vector regression, Vapnik's ϵ-insensitive cost function is employed:

$$\gamma_x(e) = \text{Cmax}(0, |e| - \varepsilon), \quad C > 0.$$

In which an error $e = y - \bar{y}$ up to ε is not penalized; otherwise, it will incur in a linear penalization. The "C penalization factor," the "insensitive zone," and the "kernel parameter" are the three hyperparameters that must be defined in the SVR implementation. In this work, grid search and k-fold cross-validation procedures were performed to tune these parameters. More details on SVR can be found in the literature [19].

3.1.3 Random forests

The random forest (RF) algorithm [20] is not only useful in regression analysis, but also performs well in feature selection. RF integrates several decision trees into an algorithm by including the concept of ensemble learning. The prediction for a new observation is then obtained by combining the predicted values derived from each individual tree in the forest. 'The number of trees', 'the minimum number of observations at the terminal node', and 'the number of suitable features to split' are the three main parameters for RF algorithms. There are comprehensive mathematical explanations for RFs in the literature [21].

3.1.4 Gradient Boosting Machines

Gradient boosting machines (GBMs) [22] are a type of supervised machine learning model that uses an ensemble of decision trees to generate an overall prediction $D(x)$. This is expressed mathematically as:

$$D(x) = d_{tree1}(x) + d_{tree2}(x) + \cdots$$

where each $d_{tree_i}(x)$ represents the prediction of the i^{th} individual decision tree in the ensemble.
There are three important tuning parameters in a GBM model that include the maximum number of trees 'ntree', the maximum number of interactions between the independent values 'tree depth' and 'learning rate' [23]. In this work, the general parameters used in the development of the 'GBM' model were identified.

3.1.5 Extreme Gradient Boosting

The extreme gradient boost algorithm (XGBoost) [24] also follows the principle of the gradient boost machine algorithm. While XGBoost requires many parameters, model performance depends highly on the optimum combination of parameters. The process of the XGBoost algorithm is: consider a dataset with n_m features and an n_n number of instances $DS = \left\{(x_i, y_i) = i = 1 \ldots \pi, x_i \in \mathbb{R}^{n_m}, y_j \in \mathbb{R}\right\}$. By reducing the loss and regularization goal, one should determine which set of functions works best.

$$\mathscr{L}(\phi) = \sum_i l\left(y_i, \phi\left(x_i\right)\right) + \sum_k \Omega\left(f_k\right),$$

where l denotes the loss function, f_k represents the (k-th tree) to solve the previous equation, while Ω is a measure of the model's complexity, this prevents overfitting of the model [25].

3.1.6 Categorical Boosting

Categorical boosting (CatBoost) is another library of gradient boosting to reduce prediction shift during model training [26]. The CatBoost technique, when compared with other machine learning algorithms, requires only a modest amount of data training and can handle a variety of data types, including categorical features. For more details on the CatBoost algorithm, resources are available in the literature [27].

3.2 Hyperparameter Optimization

Hyperparameter optimization is an important step in the ML model development process. It involves finding the optimal values for the hyperparameters of the ML algorithms, which can significantly impact the model performance [28]. Hyperparameters are the parameters of ML algorithms that are set before model training. The hyperparameter optimization process involves searching over a range of possible hyperparameters using techniques such as grid search, random search, or Bayesian optimization [29, 30]. The optimal set of hyperparameters is then selected on the basis of the performance of the model on the validation set. In this work, hyperparameter optimization was implemented using grid search.

3.3 Performance Evaluation Metrics

Various performance evaluation metrics were used to assess the effectiveness of the models developed in this work. The metrics were the mean squared error (MSE), the mean absolute error (MAE), the coefficient of determination (R^2), and the adjusted R^2 values computed for the ML algorithms during the predictive modeling of each in the test dataset (i.e., the unseen data). The square root of the MSE is the standard deviation of the differences between the predictions of the model and the experimental values:

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

The mean absolute error is the average of the errors between the predicted and actual values:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|(y_i - \hat{y}_i)|$$

The closer the MSE and MAE values are to 0, the better the model. R^2 represents the proportion of variance of target (dependent variable) that has been explained by the independent variables in the model. (R^2) values range between 0 and 1, where 1 represents a perfect model and 0 a poor model:

$$\mathrm{R}^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

Adjusted R^2 is a modified version of R^2 that considers the number of predictors (independent variables) in a given model:

$$\mathrm{R}^2{}_{\text{adjusted}} = 1 - \left[\frac{\left(1 - R^2\right)(n-1)}{n-k-1}\right]$$

A higher R^2 value shows a good match between the predicted values and actual values, while the adjusted R^2 value allows for comparison between models.

4 Results & Discussion

The following process was applied to assess the different machine learning models. Bin discretization was used to address the non-uniform data. Using the NHV_{cz} (BTU/scf) EPA requirement, all steam-assist flare data were divided into two distinct bins to avoid the removal of outliers. Then, different models were built for the bins. For each bin, the flare data were further divided into two sets: a training dataset for model training and a testing dataset for model evaluation, with a ratio of 85:15. The training dataset was divided into two parts: a training set and a validation set. A K-fold cross-validation (CV) (K = 5) was used. In 5-fold CV, the training set is split into five subsets, of which four subsets are for model training, and the fifth subset assesses the generalization of the trained ML model using the performance metrics determined from the model training. This process was repeated five times, each time choosing different subsets for validation. Grid-search-based hyperparameter tuning was utilized to find optimal hyperparameters that can generalize the model well. The grid-tuning method searches the grid for each hyperparameter and evaluates each ML algorithm based on the performance metrics (R^2, MAE, etc.). This method finds the best combination of hyperparameter values for each algorithm using a well-defined grid.

The algorithms implemented included Random Forest, GradBoost, SVR, XGBoost, and Catboost. The hyperparameters for the Random Forest and GradBoost are related to the number of decision trees and the learning rate. For SVR, the hyperparameters control the type of kernel, the degree of the polynomial kernel, and the regularization parameter. For XGBoost, the hyperparameters include the learning rate, the maximum tree depth, and the number of trees. Lastly, Catboost includes the L2 regularization term and the learning rate as hyperparameters. By tuning these hyperparameters, better performance is achieved for each of these models on each bin. Once the best hyperparameters for each ML algorithm were chosen by the grid search approach and the 5-fold CV, the test dataset (i.e., unseen data) was used to evaluate the model performance of the best trained model of each algorithm. MSE, MAE, and R^2 were calculated to assess the performance of the ML models.

% CE Models

Tables 3 and 4 provide the MSE, MAE, R^2 and adjusted R^2 values for the different ML algorithms for the training and testing datasets for predicting %CE (Table 3) and %DE (Table 4). Figure 4 shows the comparisons of actual and predicted values for the top six models for %CE and %DE. From the results of the %CE training dataset, it is found that linear-based models, MLR and SVR, have limited performance compared with the more advanced algorithms. MLR yielded an R^2 of 0.7638 with the training dataset, indicating around 76% of the variance in the %CE was captured by the model. SVR performed better with an R^2 of 0.8976, but appeared to suffer from overfitting, as the difference between the training and testing R^2 values was substantial. On the other hand, the ensemble-based algorithms, RF, GradBoost, and XGBoost, demonstrated superior performance. GradBoost and XGBoost yielded very high R^2 values of 0.9939 and 0.9974 with the training dataset, respectively, showcasing the ability to fit the data well. Finally, Catboost outperformed all other algorithms, achieving an of $R^2 = 1.0$ on the training dataset, indicating a perfect fit to the training dataset.

Similar to the performance of the algorithms in the training dataset, the MLR and SVR methods showed lower predictive capability compared with the other algorithms with the testing dataset, with testing R^2 values of 0.7267 and 0.6570, respectively. RF yielded relatively good performance for the testing data with an R^2 of 0.8630, demonstrating the ability to generalize and reproduce the "unseen" testing data. GradBoost and XGBoost also performed well, with R^2 values of 0.8931 and 0.9170, respectively, demonstrating strong predictive capabilities. However, Catboost remained the top performer with the testing dataset, with an R^2 value of 0.9344. Catboost also exhibited very low MAE and MSE, indicating precise predictions. Overall, the ensemble-based algorithms, especially Catboost, proved to be highly effective in predicting the %CE on unseen steam-assist flare data.

% DE Models

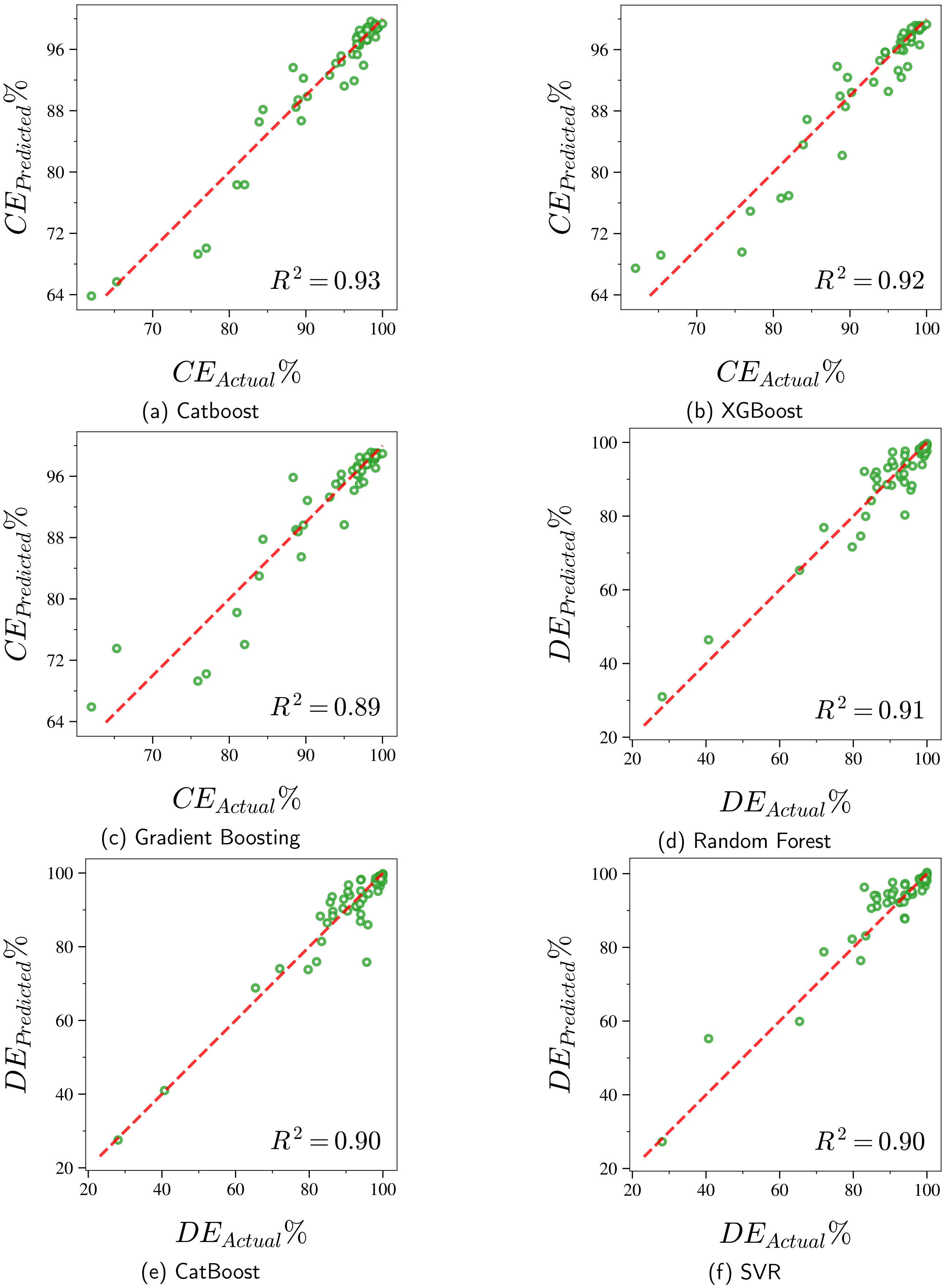
From the results presented in Table 4, it is observed that the performance patterns for predicting %DE were similar to the %CE model results. This was expected given the strong correlation between %CE and %DE observed in the data previously (Figure 2). The linear-based models, MLR and SVR, exhibited moderate performance, with SVR outperforming MLR. RF performed well, with high R^2 values of 0.9301 and 0.9073 on the training and testing datasets, respectively. GradBoost did not reach the same performance level as RF. XGBoost showed strong performance with an R^2 of 0.9774 in the training dataset and a slightly lower value for the testing dataset. As seen with %CE, Catboost was the best performer, with an almost perfect R^2 value of 0.9995 on the training dataset and a highly accurate R^2 value of 0.8977 on the testing dataset. Catboost also yielded the lowest MAE and MSE with both datasets indicating exceptional predictive capabilities. Overall, the ensemble-based algorithms, especially Catboost and XGBoost, proved to be highly effective in predicting %DE.

Table 3: Performance Summary of Different % CE Models

Algorithms	R^2	Adjusted R^2	MAE	MSE
% CE (Training dataset)				
MLR	0.7638	0.7594	3.5622	30.2776
SVR	0.8976	0.8957	1.7149	13.1282
RF	0.9723	0.9718	1.0713	3.5481
GradBoost	0.9939	0.9938	0.5133	0.7716
XGBoost	0.9974	0.9973	0.4304	0.3374
CatBoost	1.0000	1.0000	0.0038	0.0000
% CE (Testing dataset)				
MLR	0.767	0.6903	2.8048	19.4209
SVR	0.6570	0.6113	2.6237	24.3745
RF	0.8630	0.8447	2.0470	9.7389
GradBoost	0.8931	0.8788	1.6658	7.5969
XGBoost	0.9170	0.9059	1.5728	5.8980
CatBoost	0.9344	0.9257	1.3697	4.6619

Table 4: Performance Summary of Different % DE Models

Algorithms	R^2	Adjusted R^2	MAE	MSE
% DE (Training dataset)				
MLR	0.6904	0.6846	4.5067	62.1064
SVR	0.8294	0.8262	2.6061	34.2149
RF	0.9301	0.9288	2.0233	14.0126
GradBoost	0.8365	0.8334	3.5791	32.7992
XGBoost	0.9774	0.9769	1.3532	4.5425
CatBoost	0.9995	0.9995	0.2366	0.1013
% DE (Testing dataset)				
MLR	0.7217	0.6846	4.8265	50.8419
SVR	0.8956	0.8816	2.9388	19.0827
RF	0.9073	0.8949	2.9479	16.9461
GradBoost	0.8106	0.7853	4.4022	34.6090
XGBoost	0.8339	0.8118	3.5141	30.3462
CatBoost	0.8977	0.8841	2.8224	18.6901



(a) Catboost

(b) XGBoost

(c) Gradient Boosting

(d) Random Forest

(e) CatBoost

(f) SVR

Figure 4: Results for %CE (a) – (c) and %DE (d) – (f) for the top performing ML methods. The dot symbols visually compare the model's predictions to the experimental data. The dashed line is a reference line for perfect prediction, where the predicted values exactly match the experimental values.

5 Conclusion & Future Work

In this work, a machine learning framework was developed that employed several methods, such as data preprocessing, cleaning, and feature engineering, to improve the predictive performance of steam-assist flares. The ML framework consisted of a set of supervised ML algorithms, which included multiple linear regression, support vector regression, random forests, gradient boosting, extreme gradient boost, and categorical boost methods. The results showed some of the ML models had excellent generalization and regularization capabilities, which may prove superior to other ML methods for this type of non-linear and sparse dataset. Several ensemble-based algorithms, especially CatBoost, proved to be highly effective in predicting combustion and destruction removal efficiency (% CE and % DRE) in unseen steam-assist flare data. These models can be used to improve flare design and performance, e.g., via integration into the design tools.

6 Acknowledgment

The information, data, or work presented herein was funded in part by the Advanced Research Projects Agency-Energy (ARPA-E), U.S. Department of Energy, under Award Number DE-AR0001534. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof. We also thank the U-M Mechanical Engineering Department for their financial support.

7 References

[1] Genevieve Plant, Eric A Kort, Adam R Brandt, Yuanlei Chen, Graham Fordice, Alan M Gorchov Negron, Stefan Schwietzke, Mackenzie Smith, and Daniel Zavala-Araiza. Inefficient and unlit natural gas flares both emit large quantities of methane. *Science* 377, 1566–1571 (2022).

[2] Marielle Saunois, Ann R Stavert, Ben Poulter, Philippe Bousquet, Josep G Canadell, Robert B Jackson, Peter A Raymond, Edward J Dlugokencky, Sander Houweling, and Prabir K Patra. The global methane budget 2000–2017. *Earth system science data* 12, 1561–1623 (2020).

[3] US Environmental Protection Agency. *Parameters for properly designed and operated flares*. 2012.

[4] Yousheng Zeng, Jon Morris, and Mark Dombrowski. Validation of a new method for measuring and continuously monitoring the efficiency of industrial flares. *Journal of the Air & Waste Management Association* 66, 76–86 (2016).

[5] Darcy J Corbin and Matthew R Johnson. Detailed expressions and methodologies for measuring flare combustion efficiency, species emission rates, and associated uncertainties. *Industrial & Engineering Chemistry Research* 53, 19359–19369 (2014).

[6] World Bank. *Global Gas Flaring Tracker 2022*. Tech. rep. The World Bank, 2022.

[7] Marc McDaniel and Bruce A Tichenor. Flare efficiency study (1983).

[8] JH Pohl and NR Soelberg. *Evaluation of the efficiency of industrial flares: flare head design and gas composition. Final report, October 1983-December 1984*. Tech. rep. Energy and Environmental Research Corp., Irvine, CA (USA), 1985.

[9] United States Environmental Protection Agency. *Cost Reports and Guidance for Air Pollution Regulations*. Tech. rep. United States Environmental Protection Agency, 2021.

[10] David T Allen and Vincent M Torres. TCEQ 2010 flare study final report. *The University of Texas at* (2011).

[11] Arokiaraj Alphones, Vijaya Damodara, Anan Wang, Helen Lou, Xianchang Li, Christopher B Martin, Daniel H Chen, and Matthew R Johnson. Response surface modeling and setpoint determination of steam-and air-assisted flares. *Environmental Engineering Science* 37, 246–262 (2020).

[12] Vijaya Durga Damodara, Arokiaraj Alphones, Daniel H Chen, Helen H Lou, Christopher Martin, and Xianchang Li. Flare performance modeling and set point determination using artificial neural networks. *International Journal of Energy and Environmental Engineering* 11, 91–109 (2020).

[13] Helen H Lou, Jian Fang, Huilong Gai, Richard Xu, and Sidney Lin. A novel zone-based machine learning approach for the prediction of the performance of industrial flares. *Computers & Chemical Engineering* 162, 107795 (2022).

[14] Rustam Abubakirov, Ming Yang, and Nima Khakzad. A risk-based approach to determination of optimal inspection intervals for buried oil pipelines. *Process Safety and Environmental Protection* 134, 95–107 (2020).

[15] Héctor Cañas, Josefa Mula, Manuel Dıaz-Madroñero, and Francisco Campuzano-Boların. Implementing industry 4.0 principles. *Computers & industrial engineering* 158, 107379 (2021).

[16] Kanwar Devesh Singh, Preeti Gangadharan, Daniel H Chen, Helen H Lou, Xianchang Li, and Peyton Richmond. Computational fluid dynamics modeling of laboratory flames and an industrial flare. *Journal of the Air & Waste Management Association* 64, 1328–1340 (2014).

[17] Flawn Williams and Robert V Percival. Flaring in Texas: A Comprehensive Government Failure. *Texas Environmental Law Journal* 51, 1–31 (2020).

[18] Shrikant Tiwari, Prasenjit Chanak, and Sanjay Kumar Singh. A Review of the Machine Learning Algorithms for Covid-19 Case Analysis. *IEEE Transactions on Artificial Intelligence* 4, 44–59 (2023).

[19] Harris Drucker, Christopher J Burges, Linda Kaufman, Alex Smola, and Vladimir Vapnik. Support vector regression machines. *Advances in neural information processing systems* 9 (1996).

[20] Leo Breiman. Random forests. *Machine learning* 45, 5–32 (2001).

[21] Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Machine learning* 63, 3–42 (2006).

[22] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, 1189–1232 (2001).

[23] Max Kuhn and Kjell Johnson. *Applied predictive modeling*. Vol. 26. Springer, 2013.

[24] Tianqi Chen and Carlos Guestrin. “Xgboost: A scalable tree boosting system”. *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. 2016, 785–794.

[25] Erman Çakıt and Metin Dağdeviren. Predicting the percentage of student placement: A comparative study of machine learning algorithms. *Education and Information Technologies* 27, 997–1022 (2022).

[26] Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. CatBoost: unbiased boosting with categorical features. *Advances in neural information processing systems* 31 (2018).

[27] Vahid Azizi and Guiping Hu. “Machine learning methods for revenue prediction in google merchandise store”. *Smart Service Systems, Operations Management, and Analytics: Proceedings of the 2019 INFORMS International Conference on Service Science*. Springer. 2020, 65–75.

[28] Matthias Feurer and Frank Hutter. Hyperparameter optimization. *Automated machine learning: Methods, systems, challenges*, 3–33 (2019).

[29] Li Yang and Abdallah Shami. On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing* 415, 295–316 (2020).

[30] M Geetha, P Manikandan, and Jovitha Jerome. “Soft computing techniques based optimal tuning of virtual feedback PID controller for chemical tank reactor”. *2014 IEEE Congress on Evolutionary Computation (CEC)*. IEEE. 2014, 1922–1928.