# Assessment of Machine Learning algorithms for predicting autoignition and ignition delay time in microscale supercritical water oxidation process

Deewakar Sharma, Carole Lecoutre-Chabot, Fabien Palencia, Olivier Nguyen, Arnaud Erriguible, Samuel Marre

# Assessment of Machine Learning algorithms for predicting autoignition and ignition delay time in microscale supercritical water oxidation process

Deewakar Sharma[a], Carole Lecoutre[a], Fabien Palencia[a], Olivier Nguyen[a], Arnaud Erriguible[a,b], Samuel Marre[a]*

[a] Univ. Bordeaux, CNRS, Bordeaux INP, ICMCB – UMR5026, F-33600 Pessac, France.

[b] Univ. Bordeaux, CNRS, Bordeaux INP, I2M – UMR5295, F-33600 Pessac, France.

**Abstract**

With recent advancements in space technology, there is a need to develop technologies to ensure a sustainable environment for human survival. Among these, treatment of human and organic waste aboard manned space missions is a challenging task for which supercritical water oxidation using hydrothermal flames has been proposed as a possible solution. The critical step in readily adopting this technology from established ground-based setups is scaling the process to microscale. In addition to the challenge of physical realization of the microreactors at these high pressure and temperature ($P > 22\ MPa, T > 350\ C$) conditions, the need to explicitly analyze the process dynamics at microscale is inevitable owed to the size of the reactors under consideration, the physics being significantly different from meso/mini scale systems. One of the primary objectives is to identify the operating physical parameters for which formation of hydrothermal flames can be obtained. Before proceeding with an expensive computational or experimental approach to determine the exact ignition map, an initial estimate based on physical arguments can help in providing insights into the process. We address this problem using homogeneous ignition calculations to develop machine learning models to predict autoignition as well as ignition delay time. The ingenuity of the work lies in defining autoignition criteria in relation to flow time scales expected

at microscale. Various classification models were trained and tested for predicting autoignition and regression models were demonstrated to predict the ignition delay time. While predicting autoignition is a straightforward process, a two-step approach is proposed for ignition delay time. Finally, how machine learning can be used more explicitly, particularly for understanding and designing efficient microreactors, is presented which highlights that machine learning approach is not merely restricted to prediction but can also have real implications on improving the process as a whole.

*Keywords: Supercritical water oxidation, hydrothermal flames, machine learning, flame ignition, microscale.*

## 1. Introduction

Over the recent years, there has been an unprecedented growth in the space industry and it is envisioned to have deep space missions involving humans in the near future. It is thus inevitable to develop technologies which can aid in sustaining life aboard these missions. Among these, treatment of organic and human waste and converting them back to reusable resources is identified as one of the essential technologies. The generated by-products can be reused in some form thereby reducing the burden of carrying additional resources during space launches. Tapping on the potential of supercritical water oxidation (SCWO) with hydrothermal flames, this process has been proposed to meet the aforementioned technological requirements [1, 2]. Hydrothermal flames are the flames which exist in supercritical (near critical) water due to autoignition of organic matter. The phenomenon is attributed to reduction in autoignition temperature of organic matter, which also acts as a fuel, in supercritical water conditions ( $P > 22.1\,MPa, T > 374\,C$ ). While water is polar at ambient conditions, it becomes non-polar at supercritical conditions. This facilitates dissolution of several non-polar gases such as oxygen, nitrogen, as well as organic matter. With both organic matter and oxygen dissolved in supercritical water (SCW), it provides a uniform medium for the oxidation of organic matter. In these conditions, when the

concentration of organic matter (fuel) exceeds a certain limit, $i.e.$ threshold concentration at a given pressure and temperature, the oxidation reaction leads to autoignition and thus the formation of hydrothermal flames. The motivation to use hydrothermal flames for SCWO are two folds. Firstly, it permits the injection of reactants at lower temperature than the critical point of water which circumvents the problems of clogging in the inlet section of the reactor – one of the major drawbacks of SCWO [3]. Secondly, a higher heat generated due to hydrothermal flames promote a faster reaction and decomposition of waste, recalcitrant molecules, etc. This process has been successfully demonstrated for ground-based applications [4-7]. An important constraint in readily adapting the designs and understanding of hydrothermal flames from ground-based applications to the space sector is the size of reactors under consideration. Most of the reactors for ground applications are at meso/mini scale with the volume of reactor being of the order of a few liters. However, with volume and weight being a limitation in space applications, it is inevitable to have a device with similar functionalities but which is compact and light in weight. In pursuit of advancing the technology of SCWO using hydrothermal flames for space applications, we have proposed to carry out this process at microscale [8]. We define this process as $\mu SCWO - H$. The feasibility to realize such microreactors (made of sapphire) which can withstand such harsh pressure and temperature conditions along with being chemically compatible with SCW has already been demonstrated [8]. This was made possible due to patented developed technology to etch sapphire [9]. In addition to fulfilling the criteria of compact design, working at microscale augments the SCWO process capabilities by tapping the advantages of a classical microfluidic system, such as further reductions in reaction time, more uniformity in the reactions, etc.

The need to develop this technology, as evident from the above discussion, necessitates an explicit study of hydrothermal flames at microscale owing to differences in the physics with respect to meso/mini scale reactors. One of the prime reasons can be ascribed to a high surface area to volume ratio in a typical microsystem resulting in a higher heat transfer, which can eventually render a different behavior to

hydrothermal flames when compared to meso/mini scale systems. Though, it can be argued that for utility in space, the absence of any gravity can result in negligible heat loss due to natural convection, analysis on the ground being the first step before moving to space applications necessitates such an investigation. The second difference arises due to the small reactor volume which implies a large amount of heat being generated in a smaller region. Consequently, the effect of heat constrained within the system will significantly impact the flame dynamics. Thirdly, owing to low flow rates encountered at microscale, the operating condition is primarily expected to be in laminar regime as compared to the majority of the pertinent literature operating in turbulent regime for hydrothermal flames, for example in [4, 6].

In order to develop this technology, it is vital to identify operating regimes (flow rates, fuel & oxidizer concentration, operating temperature, pressure) for which autoignition and thus, the formation of hydrothermal flames can be attained. While development of an exact ignition map can be achieved from experiments or high-fidelity numerical simulations, understanding autoignition conditions based on homogeneous 0D calculations can aid in narrowing down the operational window to search upon the optimum conditions. Homogeneous ignition analysis involves solving the reaction kinetics in time assuming a zero-dimensional reactor using the reaction scheme for a given fuel under consideration. The conventional utility of this approach is limited, such as predicting autoignition parameters (ignition delay, maximum heat release etc.) for given set of operating conditions. However, it holds enormous potential in the context of the present problem, which could be tapped by using data-driven models. For instance, upon successful training the machine learning models, these can be used to predict the desired output parameters for new operating conditions. Further, direct mapping of field parameters to ignition related parameters can help to better understand the formation of ignition kernel, flame dynamics, design of reactors, etc.

Machine learning over the recent years has garnered significant attention in diverse fields of physical systems – fluids, thermal, and combustion. Brunton *et al.* [10] have presented an overview on how various

machine learning methodologies can find their applicability in the various areas of fluid mechanics, addressing a wide range of problems, such as modeling and controlling fluid flows. The authors described how optimization performance as well as convergence time can be improved by posing several objectives in fluid mechanics as optimization and regression problems. Brenner *et al.* [11] and Brunton [12] in their recent articles have further provided insights into advancing fluid mechanics using machine learning. Machine learning models have also been used to predict thermodynamic properties of pure fluids and mixtures, equilibrium compositions in liquid-vapor flash calculations, heat transfer coefficients, etc. [13-18]. In the context of combustion, Zheng *et al.* [19] reviewed the recent progress and applicability of machine learning in combustion studies. The authors highlighted several key pertinent sub-domains where machine learning could have significant impact, such as detection of combustion oscillations due to thermoacoustic, investigating physiochemical properties of fuels and subsequently designing a new generation of fuels, and reconstruction of cellular surface of gaseous detonation, to name a few. A more comprehensive review has been recently presented by Zhou *et al.* [20]. The authors highlighted developing relations between input parameters (temperature, pressure, species etc.) and output parameters of interest (chemical kinetics etc.) for combustion simulations using various machine learning models. Here again, feasibility of fuel design, predicting physical properties of fuel (such as density, heating value), their ignition delay time, flash point, etc. using neural networks was presented. In addition, the authors addressed the challenges of combining machine learning with combustion research and aptly mentioned that one of the key challenge lies in judiciously framing the physical problem in the framework of machine learning. This not only requires in depth understanding of the machine learning aspects, such as data selection and pre-processing, appropriate choice of machine learning models, tuning the models, etc., but most importantly, an in-depth understanding of the physical problem. In the context of the current problem, relevant work in auto-ignition is scarce and only a few of them can be found in literature. Pan *et al.* [21] used *Support Vector Machine* model to predict autoignition temperature of organic fuels

based only on their molecular structure. The relationship was developed between autoignition temperature in terms of descriptors which were selected using a genetic algorithm. This algorithm is known to mimic natural selection evolutionary processes primarily, selection, gene crossover, and mutation to obtain global optima solutions. The first step comprises of initially creating a population, which is a set of individuals with each individual representing the solution that is to be solved. In order to decide which individuals will be selected from the population, a fitness function is defined to evaluate the fitness score. Using the fitness score, two pairs of individuals are selected with individuals having a higher fitness score are more likely to be selected. Subsequently, offspring's are created from genes (here gene represents variables on which individuals are dependent) after selecting a crossover point. The offsprings are continually added to the population and the algorithm terminates when convergence is attained, $i.e.$ when no new offspring significantly different from existing ones can be formed. A similar work has been reported by Suleiman *et al.* [22] where using molecular weight along with the number of carbon, hydrogen, and oxygen atoms as parameters, the authors were able to predict autoignition temperature of organic energetic compounds using hybrid support vector regression. More recently, Shah *et al.* [23] used *Random Forest* and deep learning algorithms to predict autoignition and flame properties (flame speed, fuel octane ratings, ignition delay time ) in multicomponent fuels. Both the methodologies were found to have high predictability even using a small data set. Cui *et al.* [24] developed models for ignition delay using Back propagation neural network for n-butane/hydrogen mixtures. In addition, genetic algorithm optimized back propagation model was also developed and performance of both were compared over a wide range of operating conditions (pressures varying from 20 –25 bar, temperature varying from 722 - 987 K, equivalence ratio from 0.5 - 15) and the performance metric (average correlation coefficient) in case of the latter was found to be superior. Lehn *et al.* [25] used neural networks to predict the laminar burning velocity using pressure, temperature, fuel/air ratio, and twelve molecular groups as the input variables. Using sensitivity analysis to analyze the impact of functional groups, the laminar

burning velocity was found to increase with unsaturation, which was coherent with observations in literature. Cui *et al.* [26] used back propagation neural network to predict the ignition delay time in surrogate fuels where data for training and validation was generated using 0-D ignition calculations. An interesting result reported was comparison of computational time using 0D simulation and trained back propagation neural network. While the former took $28\ s$ for a single calculation, 176 cases were evaluated in $3.2\ s$ using the latter highlighting significant gain using neural networks. More recently, Huang *et al.* [27] studied the impact of hydrogen addition on the ignition delay time for aviation fuel using artificial neural network (ANN). It was found that a large local relative error occurs when the ignition delay time was very small. In order to improve the performance, a sub-ANN model was developed by training on the subset of data points corresponding to the poor predictions from the initial ANN. Thus, if the predicted ignition delay time from the initial ANN was below the selected threshold of $10^3 \mu s$, the sub-ANN was used to update the output value.

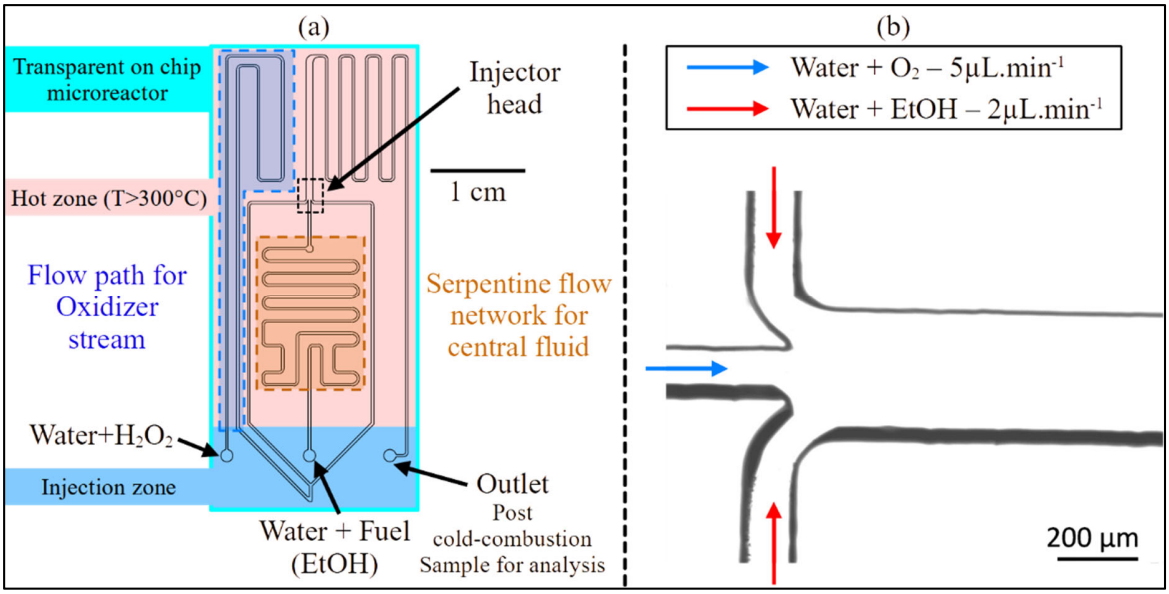
In this work, we demonstrate the use of machine learning models to predict the autoignition characteristics of ethanol as a model fuel for the formation of hydrothermal flames at microscale. Owing to microscale dimensions under considerations, the ingenuity of the current work lies in defining autoignition by accounting for residence (flow) time and subsequently developing machine learning models to predict the output parameters – possibility of autoignition and ignition delay time. Furthermore, we also highlight how these data driven models can be used to predict zone of formation of the ignition kernel from real physical data, such as CFD simulations. This can eventually be used for understanding the dynamics of hydrothermal flames as well as designing better microreactors. The prime focus of the article is thus to exhibit how by using a simple homogeneous ignition calculation in conjunction with machine learning models, initial estimate of autoignition conditions and corresponding ignition delay time can be predicted for microscale SCWO process. We begin by describing the physical problem and means of data generation. Subsequently, various machine learning algorithms are applied

on the data to predict the autoignition condition as well as ignition delay time. Finally, we illustrate how this approach can have physical implications for better understanding of the process dynamics for development of this technology.

## 2. Problem description

The physical process to be investigated was described in the previous section. In order to provide readers with more insights into the microfluidic process of SCWO, a schematic of the microreactor design proposed for the aforementioned application is shown in Figure 1 (a).

The microreactor consists of two inlets feeding the system with oxidizer ($H_2O$ + $H_2O_2$) and fuel ($H_2O$ + ethanol). Two serpentine microchannels help preheating the two fluids (and ensure for the oxidizing fluid the decomposition of $H_2O_2$ in $H_2O$ and $O_2$) before they are put in contact in the injector. While details have been presented elsewhere [8] , we present an example illustrating phenomenon of phase-change of water at high pressure and temperature (Figure 1(b)) conditions supporting the feasibility to realize such extreme conditions at microscale. For the current problem, the focus lies in the injection zone where fuel and oxidizer streams mix with each other. The schematic of this current physical process under consideration (injection zone) can thus be described as shown in Figure 2 and the long-term objective is to obtain a sustainable hydrothermal flame in the presented microreactor.




183

184 Figure 1: (a) Schematic of the microreactor system proposed to be used for $\mu SCWO - H$. (b) Microscopy image of
185 the injector head microfabricated in sapphire.

186 The elementary design considerations as well as experimental observations [8] provided insights into how
187 the considered physical dimensions (microscale reactor) could implicitly impact the phenomenon of
188 autoignition in the current system and thus finds relevance in the context of the present problem. This
189 can be explained as follows. Upon injection of the fuel (ethanol in the present case) and oxidizer into the
190 reactor (microchannel), the species mix with each other resulting in a chemical reaction.

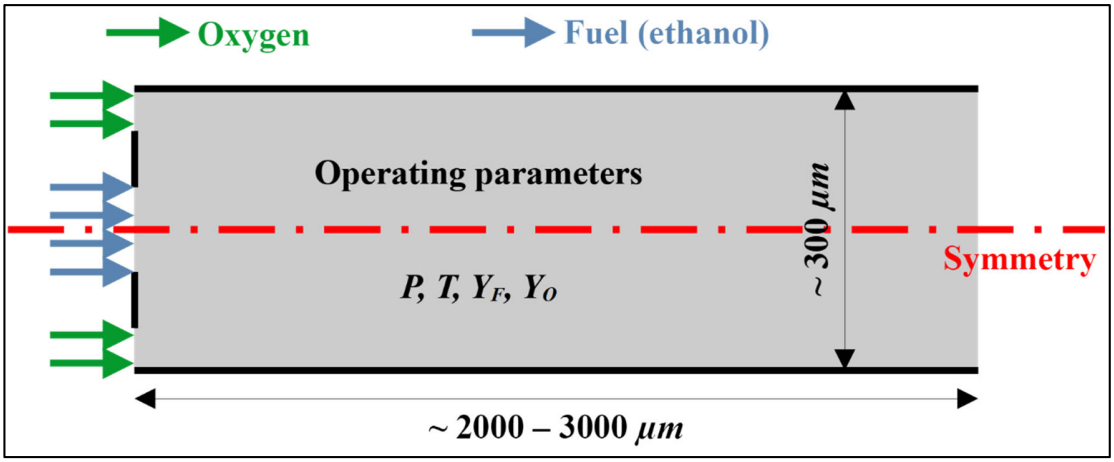



Figure 2: Schematic of microscale reactor for $\mu SCWO - H$.

Even though the reaction between fuel and oxidizer is omnipresent in these operating conditions, autoignition will only occur under certain conditions governed by the coupled phenomena, hydrodynamics, thermodynamics and chemical kinetics of the system. An important characteristic of the autoignition is the ignition delay, which is defined as the time taken by a reacting mixture of a given composition to auto-ignite. The ignition delay can thus be interpreted as the time when the first ignition kernel may be expected to form. This ignition delay in the current problem can be related to the constraint on the permissible autoignition conditions due to the scale of the reactor under consideration. It is evident that in order to have hydrothermal flame at microscale, the ignition kernel needs to be formed within the reactor channel. This implies that the ignition delay for a given mixture composition at a given pressure and temperature should be less than the time taken by this reacting mixture element to flow through the reactor microchannel. Failure to meet this criterion implies that even though the mixture is ignitable, it will not serve the purpose for the given physical system under investigation. This is illustrated in Figure *3* where we define the flow time as $t_f = L/u$ with $L$ being the length of the channel and $u$ the axial velocity.

The above explanation thereby substantiates presenting the schematic of the microreactor with physical dimensions in the present homogeneous reaction/0D reactor study.

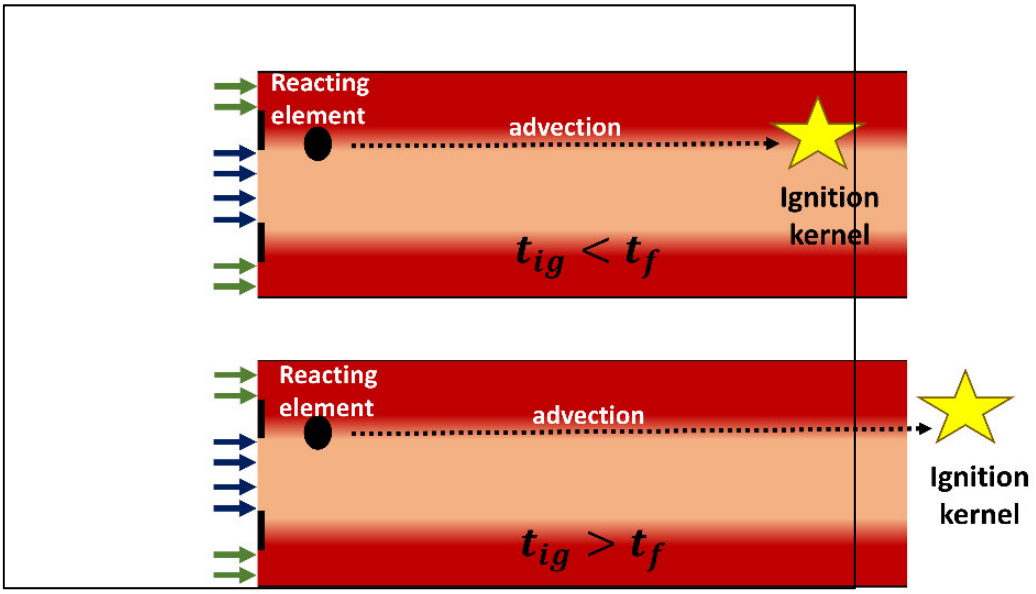



Figure 3: Schematic illustration of how ignition delay $(t_{ig})$ being larger than flow residence time $(t_f)$ implies failure of autoignition in $\mu SCWO - H$ application.

Thus, despite considering homogeneous calculation in the present context, reference to the physical system and hydrodynamics is self-explanatory as it will govern the criterion for deciding whether or not we have autoignition. This forms one of the major differences in identifying autoignition conditions at microscale in comparison to meso/mini scale reactors. Before proceeding further, it is worth mentioning that in a real scenario where hydrodynamics will be considered, the ignition limits (which will be presented below) will be modified owing to several local dynamics playing their role. However, the utility of the current approach lies in providing an initial estimate on the limits along with physical implications it can have as discussed in §5.

## 3. Methodology

### *3.1 Data generation*

In order to develop a machine learning model, having reliable and sufficient data is of prime importance. In the present work, data is generated using homogeneous ignition calculations based on a single step reaction for oxidation of ethanol in SCW conditions as given in [8]:

$$C_2H_5OH + 3O_2 \rightarrow 2CO_2 + 3H_2O \quad (1)$$

with reaction rate evaluated as,

$$Q_{EtOH} = -10^{17.23} \exp\left(\frac{-E_a}{RT}\right)[EtOH]^{1.34}[O_2]^{0.55} \quad (2)$$

The activation energy in the above expression is, $E_a = 214 \frac{kJ}{mol}$ [28]. The governing equations of conservation of mass, species, and energy were solved in time using a simple first order Euler scheme. As the physical system under consideration is at microscale, the relative change in pressure is expected to be very small when compared to operating pressure which is greater than $22.1\ MPa$. Thus, homogeneous calculations in the current work correspond to a constant pressure reactor system with density varying in the system. Appendix 1 in the supplementary information presents the governing equations used for data generation. The parameters (independent variables) considered were the mass fraction of ethanol $(Y_f)$, mass fraction of oxygen $(Y_o)$, the temperature $(T)$, and the pressure $(P)$. Table 1 presents the range of each parameter considered in the present work. The upper limit for fuel and oxygen concentration arises due to the validity of reaction rate up to these limits as mentioned in [8].

Table 1: Range of input parameters considered

| **Parameter** | **Range** |
|---|---|
| Pressure $(P)$ | $225 - 250\ bar$ |
| Temperature $(T)$ | $350 - 450\ [C]$ |
| Fuel (ethanol) $(Y_F)$ | $0.5 - 4.5\ [\ in\ percent]$ |

| | |
|---|---|
| Oxidizer $(Y_o)$ | $0.5 - 9.5$ [ $in\ percent$] |

Each operating point, denoted by $[Y_{f,i}, Y_{o,i}, T_i, P_i]$ for $i^{th}$ data point, was randomly generated using Python. The data set consists of 15000 data points. The entire data set, termed as data set 1, comprised 15000 data points and was used for training, validation, and testing various models. It is intuitive to expect that once the models are trained, these can further be used to predict the output (autoignition or not, ignition delay time) for new conditions not included in this data set, such as those coming from experiments. This could further aid in testing the trained machine learning models. However, in the absence any such data at present, another data set comprising 5000 data points was created which can be considered as an example of data from experiments. We term this data set as external data set and this is used only for testing the trained models. The objective of doing this is to further test the robustness of the trained models. For each data point, the time taken for autoignition, $i.e.$, ignition delay time, was evaluated as the time when a sudden jump in the heat release rate is observed as shown for one case in Figure 4.

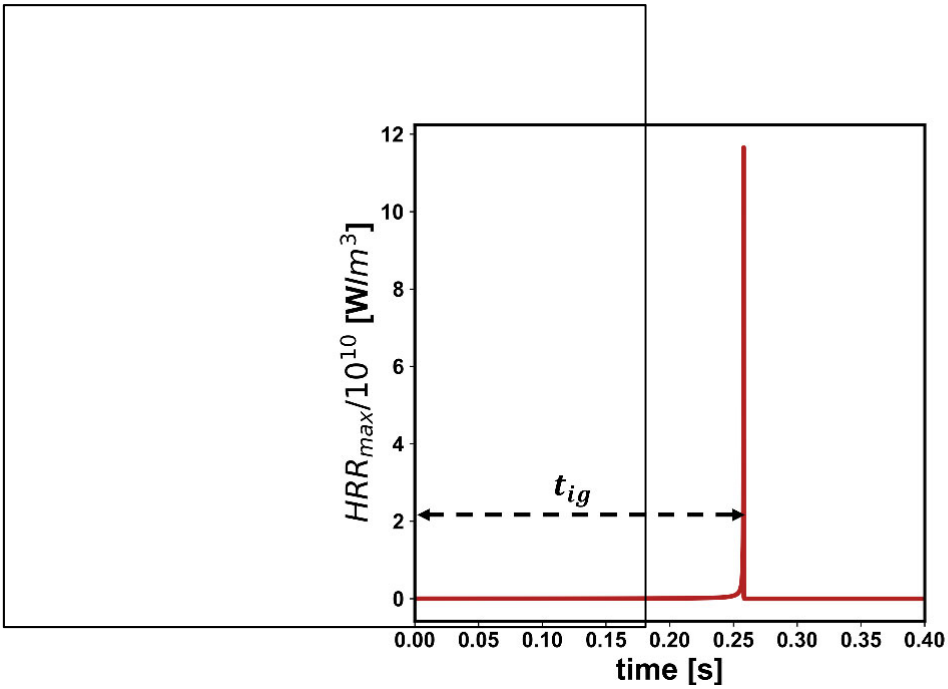



Figure 4: Representative case illustrating how ignition delay time was defined. The plot is for $P = 250\ bar, T = 400\ C, Y_f = 2\ \%, Y_o = 9.5\ \%$

The output or dependent variables of interest are whether we have autoignition or not $(I_g)$, and if yes, what is the ignition delay time $(t_g)$. While the former is a categorical variable, the latter represents a continuous output variable. As per previous discussion, categorizing $I_g$ will depend on flow or hydrodynamic time scales under consideration. Based on preliminary investigation on design aspects as presented in [8] , $1s$ is chosen as the upper limit in the present case. This corresponds to flow rates in the range $5 - 100\mu l/min$ for channel dimensions of $\sim 300\mu m$ width and $\sim 20\ \mu m$ etch depth and length $\sim 2500\ \mu m$. The time limit presented is thus the maximum tentative flow time in these conditions which nevertheless can be adapted for different flow conditions in the future. Based on this criterion, the two data sets comprised of $52\%$ and $53.2\ \%$ cases corresponding to autoignition as shown in Figure 5, which represents a well distributed data.

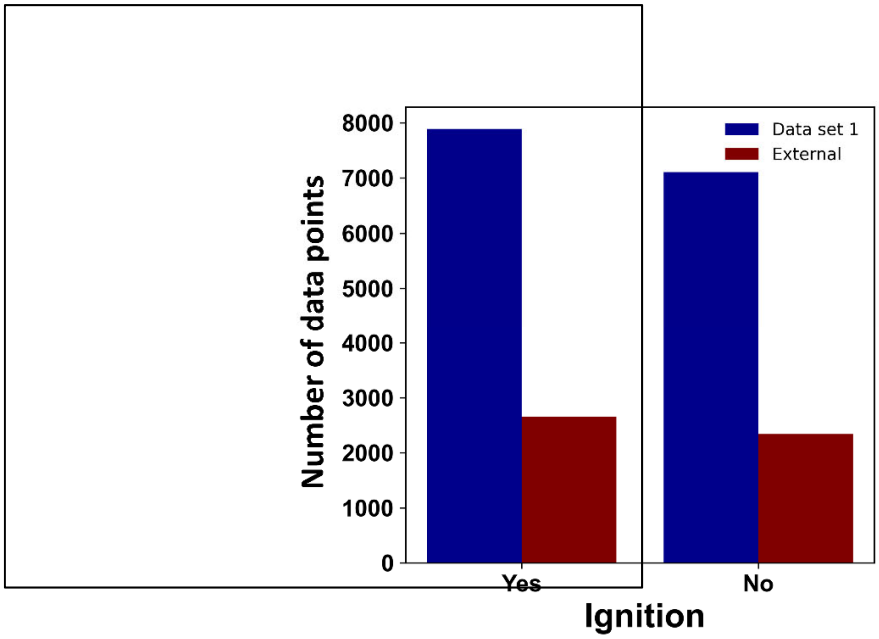



Figure 5: Bar graph illustrating number of data points in the data sets (data set 1 and external) which correspond to conditions of autoignition (blue bar) and which do not (dark red bar).

### *3.2 Data treatment & preprocessing*

One of the initial steps before developing machine learning models is data preprocessing which includes several aspects such as checking for any duplicate entries. Both the data sets, data set 1 and external data set, were checked if any duplicate entries of operating conditions $[Y_f, Y_o, T, P]$ were present and none was found. In terms of the range of values of the individual parameters, there exists a large variation in their magnitudes. Here mass fractions, owing to their definition, lie between $0$ and $1$, while temperature $(350\ C - 450\ C)$ and pressure $(225\ bar$ to $250\ bar)$ are $O(10^2)$. Training the models directly on this data can result in model inaccuracies due to bias towards features with larger values. Several standard scaling methods, such as normalization (scaling between minimum and maximum value) and standardization (scaling such that data exhibits properties of Gaussian distribution with zero mean and unit variance), exist to address this issue. The primary objective of this step is to ensure that all the

features are of the same order. In the present work, however, we resort to scaling based on physical aspects of the problem – operating conditions. It is evident from the aforementioned range of the variables that scaling is required primarily for temperature and pressure. As in supercritical water oxidation, the critical point of water ($T_{c,w} = 374\ C$) serves as a landmark to define the process, we scale the feature temperature ($T$) as $T = (T - T_{c,w})/100$. With temperatures varying from $350\ C$ – $450\ C$, this results in $T$ being the same order as mass fractions. Similarly, pressure, which in the current scenario varies from $22.5 -\ 25\ MPa$ ($225\ bar -\ 250\ bar$), was scaled by 100. These scaling ensures that all the variables (features) are now same $\sim O(1)$ in the current problem. The motivation to opt for the proposed scaling is its simplicity in implementation. This will be particularly useful when using these models with external data from physical analysis, such as from experiments or entire 2D/3D space from CFD simulations, wherein the physical variables can be obtained in the proposed scaled manner and thus be directly fed into the models for predicting autoignition and ignition delay time rendering ease of using machine learning methodology for real process. For output parameters, $I_g$ was set to 1 when we had autoignition else it was assigned the value of 0. In addition to identifying whether or not we have autoignition, predicting the ignition delay time is also a parameter of interest. One of the key challenges is how to describe ignition delay time when the input operating parameters corresponded to no autoignition condition. This is because, in a physical sense, no autoignition implies that despite having left the reacting mixture element for considerable long duration, we do not observe any sudden increase in HRR. Thus, a very large value could be used to define no autoignition. However, in the present context no autoignition is identified when $t_{ig} > t_f$ and thus, in principle, any value greater than $t_f$ could be assigned to $t_{ig}$ . One of the primary concerns with this approach is that many data points with this same value could create a bias towards this assigned value. Thus, errors may creep in during training and subsequent testing. In order to minimize this, we assign value 0 to these instances. It is to be mentioned that assigning 0 does not have any physical meaning and is just an indicator of no-autoignition event. It may be argued

303 that this may also create a bias towards lower values of ignition delay time, and this is what will be
304 explored in the subsequent section and an appropriate strategy to predict ignition delay time will be
305 discussed.

306 *3.3 Machine learning models and performance criteria*

307 The first part of the presented problem represents a classification problem wherein the objective lies to
308 predict whether we have autoignition. Various supervised machine learning models were trained to
309 predict the autoignition ($I_g$) for a given set of input variables. These include *logistic regression (Logistic)*,
310 *Decision Tree (DT)* [29], *Random Forest (RF)* [30], *Support Vector Machine Classifier (SVC)* [31], *k-Nearest*
311 *neighbors (kNN)* [32], and *AdaBoost classifier (Ada)*. The motivation to use different models was to analyze
312 which model could well capture the desired trend and be subsequently used for further analysis. The
313 models were implemented in Scikit-learn [33] in Python 3.9. The performance of each model is known to
314 depend on several parameters and values of these parameters need to be tuned to have their optimum
315 values, the process known as *hyperparameter tuning*. The objective of this process is primarily to ensure
316 that the model achieves a global minimum, which may otherwise be skipped when considering the default
317 values. These optimal parameters were obtained by using the *GridSearchCV* function in Python using an
318 8-fold cross validation. A similar methodology was adopted for predicting ignition delay time. Since
319 ignition delay time is a continuous variable, here regression models were used, namely *Ridge regression*
320 *[34]*, *Decision Tree regressor*, *Random Forest regressor*, *k-nearest neighbor regressor*, *Adaboost*, and
321 *Gradient boost regressor (GB)*. While details of each model can be found in the cited reference, we briefly
322 describe each model in simple terms.

323 - **Linear models:** Ridge and logistic models form a part of generalized linear models. Logistic
324 regression is a classification algorithm used to predict the probability of an event. It is one of the
325 easiest models to implement, train, and test. The model assumes there is no or very minimal

multicollinearity between the independent variables. The probability of an event $(p)$ is converted to odds (ratio of success and failure probability, $p/(1-p)$ ) on which a logit transformation is applied. This is known as log of odds and is described by $\log\left(\frac{p}{1-p}\right) = \beta_0 + \sum_{i=1}^{k} \beta_i X_i$. The coefficients $(\beta_i)$ are obtained by maximizing the log of likelihood. ***Ridge regression*** is a form of a linear regression model where the model is penalized using sum of squared of the weights (coefficients) in order to prevent overfitting. Here, the residual sum of squares can be evaluated given by $\sum_{i=1}^{N}(y_n - \hat{y}_n)^2 + \lambda \sum_{i=1}^{k} \beta_i^2$ , where $\hat{y}_n = \sum_{i=1}^{k} \beta_i X_i$ and $\lambda$ is a parameter which governs the severity of the penalization.

- **Tree based models:** *Decision Tree*, *Random Fores*t, *AdaBoost*, and *GradientBoost* models fall under the category of Tree Based models. In these models, a tree like structure is developed based on certain conditions applied on the input variables and a decision is made to predict the output. Thus, the prediction can be obtained using simple if-else conditions rendering these models quite intuitive to interpret. These kinds of models can be used both for classification as well as regression problems.

  ***Decision Tree*** is the simplest model in this category. Starting with the base of the tree, also termed as the root node, the data is split into branches leading to decision nodes. The terminal node where a decision is made, are called the leaves of the tree. Usually, the tree is not allowed to grow to its full depth to prevent overfitting and some parts of the tree are removed, the process being termed as pruning.

  ***Random Forest*** comprises of several decision tress operating collectively to predict the output. The trees are developed by bootstrapping the data and each tree predicts an output. Subsequently, the output with maximum number of votes is taken to be the model prediction. Random Forest forms a part of ensemble models, $i.e.$ the models where several models (trees in this case) are generated and the output is governed collectively from these models. While

Random Forest operates in parallel, averaging out error over all trees, ***AdaBoost*** and ***GradientBoost*** work sequentially to reduce the errors. In these models, the errors in the previous models (trees) are corrected in the successive models. In *Adaboost*, only a single split is permissible in each tree resulting in two leaf nodes and the trees are called stumps. Initially, equal weight is assigned to each data set and the weights are adjusted while developing subsequent stumps. This is attained by assigning a higher weight to incorrectly classified data point in order to correct in subsequent classification. Unlike *Adaboost*, no limitation is posed on the number of leaves and splits in *GradientBoost* algorithm. The peculiarity of latter method is that leaf nodes predict residuals and with addition of tree, the residuals are reduced.

- **k-Nearest Neighbor** It is a form of non-parametric ($i.e.$ it does not assume any form of mapping function between input and output) and supervised learning classifier which uses proximity between data set to classifications. "$k$" signifies how many neighbors will be used to predict the output. The proximity is calculated in terms of distance metrics, the most common being the Euclidean distance. These models are easy to implement and have very few parameters to hyper-tune. However, it fails to perform well with the increase in dimensionality of the data.
- **Support Vector Machine (Classifier)** This method can be used both for classification and regression analysis, the former being used in the current study. It sorts the data into respective classes by defining margins / hyperplanes between the classes. In the case of linearly separable data, the best decision boundary (hyperplane) is identified as the one which has the largest distance from the classes. The points nearest to the hyperplane, termed as support vectors, are used to evaluate the distance. In the case of non-linearly separable data, data is transformed to higher dimensions using kernels, such as polynomial, radial basis, in order to define linear decision boundary.

Several metrics exist to quantify the performance of machine learning models [35], such as the absolute error ($MAE$), the mean squared error ($RMSE$), and the coefficient of determination ($R^2$) for regression problems and accuracy, F1-score, etc., for classification problems. In the present case, performance of classification models was evaluated in terms of accuracy, which is defined as the ratio of correct predictions to total predictions while $R^2$ was opted for regression analysis and can be defined mathematically by (where symbols have their usual meanings),

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(\hat{y}_n - y_n)^2}{\sum_{i=1}^{N}(y_n - \bar{y}_n)^2} \quad (3)$$

### *3.4 Training, cross validation, and testing of models*

The various machine learning models were built on data set 1 by dividing it into training and testing subsets in the ratio of $9{:}\,1$. Here, $10\ \%$ of the data was reserved for final testing (termed as final test data henceforth) and none of these data points were used at any stage during the training/validation process to make sure that performance metrics obtained for each model are fair by all means.

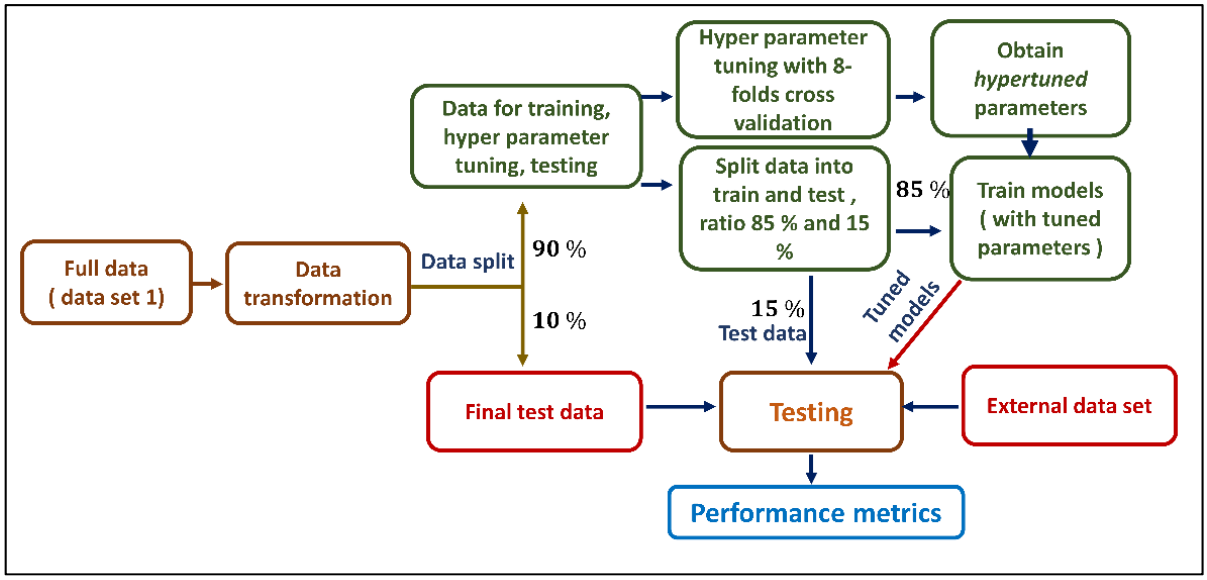



Figure 6: Schematic of procedure adopted in the current work for training, validation, and testing machine learning models.

The other 90 % of the data served two purposes. Firstly, it was used to obtain hyper-tuned parameters using *GridsearchCV* function with 8-folds cross-validation. Hypertuned parameters for all the models are presented in Appendix 2. Once the tuned parameters were obtained, this data set was split into train and test in the ratio of 85 % and 15 % for training and testing, respectively. The latter is termed as test data henceforth. Subsequently, models were trained with best preforming parameters and tested on this 15% data (test data). Finally, these models were tested on the initial 10 % data (final test data) in addition to the external data. A schematic of this procedure is illustrated in Figure 6.

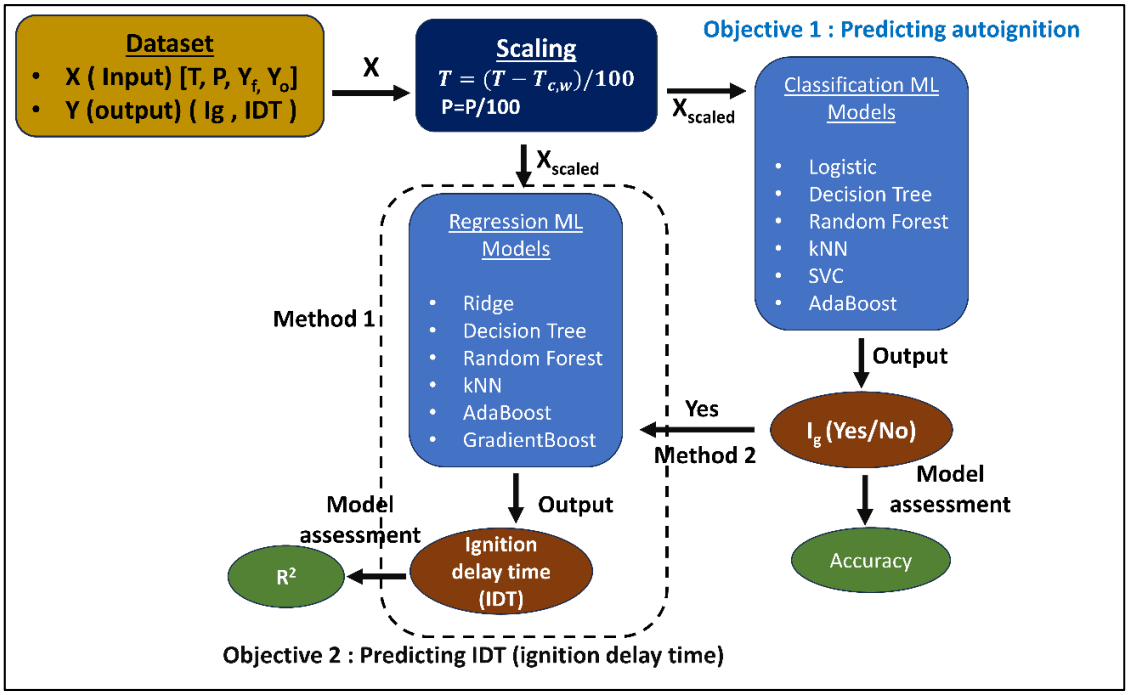



Figure 7: Schematic of the flow process for predicting autoignition and ignition delay time from input variables. For predicting ignition delay time, two methods are proposed (see text below). Models are evaluated based on accuracy as well as $R^2$ for autoignition and ignition delay time prediction, respectively.

As mentioned previously, the two objectives of interests are to predict whether there is autoignition and ignition delay time. Figure 7 Illustrates a schematic of these two objectives highlighting the inputs, predicted variables, and performance metrics. While training/testing the models for whether we have ignition or not is pretty straightforward, it is not so evident for ignition delay time. The complexity arises because in the entire data, there are cases which do not correspond to autoignition and for which the values have been set to zero. Here, two different approaches are possible. Firstly, models can be trained (and hyper-tuned) using the entire data set $i.e.$ including zero as ignition time for no autoignition cases. This implies that predicting ignition time will be independent of whether we have ignition or not. In the second approach, the models for predicting ignition time can be trained only using the real ignition delay

time data, $i.e.$ data points corresponding to autoignition cases. These methodologies lead to two different possibilities of predicting the output variable on an entirely new data set (external data set). For the sake of clarity, these are termed as method 1 (M1), where models are developed on the entire data set, and method 2 (M2), where models are developed only using the actual ignition delay time. The advantage of former lies in its simplicity to predict the output as in case of the latter, two models need to be developed and used, one for predicting autoignition and second for predicting the ignition time where the outcome of first model (classification) will have influence on the overall outcome of the second model (regression). However, the first method could be prone to more error due to manual bias introduced. We thereby test both these approaches and compare the performance metrics of the models, which are presented in the next section.

## 4 Results and discussions

In this section, the performance of various machine learning models is presented. However, before that, we do some data analysis of the results from homogeneous ignition calculation to understand how the data behaves and if the output yields meaningful results. This will further aid in ensuring that proposed implications of using machine learning models can be successfully applied as described in §5.

### *4.1. Data Analysis*

We first try to seek the effect of various independent parameters on whether we have ignition or not. It can be seen from Figure 8 that with increase in temperature, the likelihood of autoignition increases. Here, for the sake of clarity, only $20\%$ of the data points from data set 1 were plotted by selecting every $5^{th}$ data point. The trend in Figure 8 can be explained as increase in temperature represents a higher heat release and thus a faster reaction rate, which results in a smaller autoignition time. This is further supported from the ignition delay plot as shown in Figure 9. Further, both the outputs seem to be affected more by the oxidizer (oxygen) percentage as compared to fuel concentration while there is a very little

impact of pressure. This can be inferred from Figure 8 and Figure 9 where the trend for autoignition as well as ignition delay remains nearly flat with fuel percentage while it exhibits a non-linear trend (nearly exponential) with oxygen percentage.

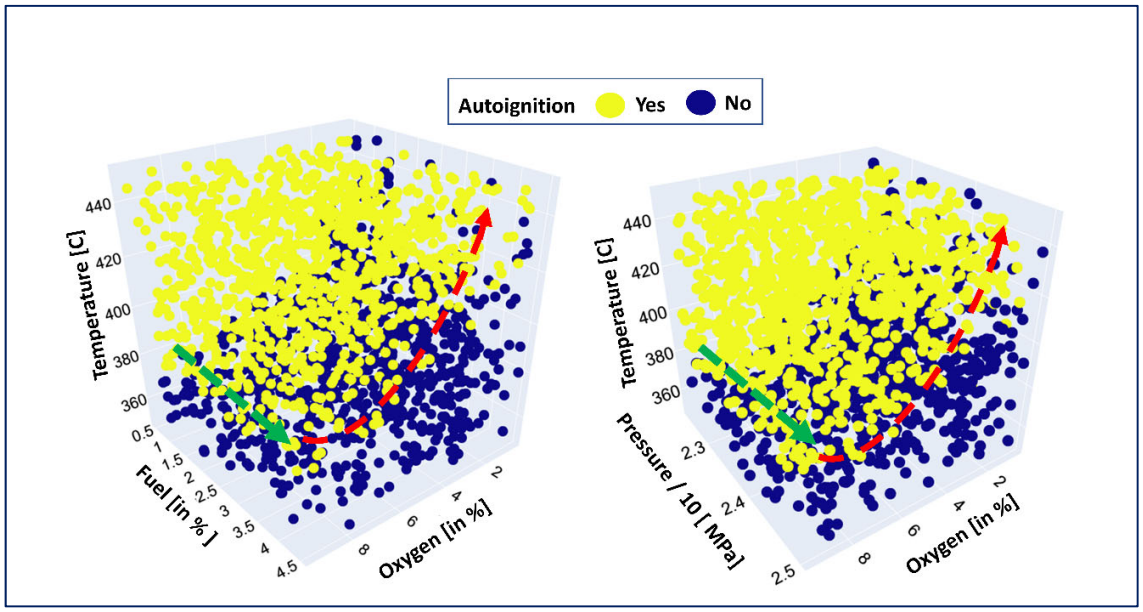



Figure 8: Scatter plot illustrating dependence of various input parameters on output variable $(I_g)$. Green and red dashed lines illustrate the trend of ignition likelihood with fuel and oxidizer concentration, respectively (left) and pressure and oxidizer concentration (right) with temperature.

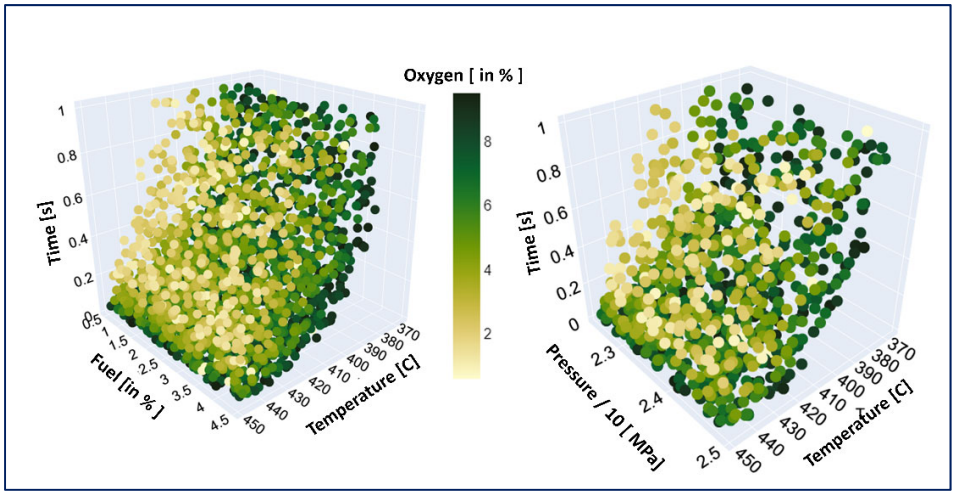



Figure 9: Ignition delay plot as function of various parameters.

From a process engineering perspective, this trend is thereby advantageous as we have the possibility of autoignition without much varying of the fuel concentration. This implies that we can nearly keep this parameter constant when attempting to seek autoignition conditions in real experiments and focus more on adapting the oxidizer concentration.

### *4.2 Performance of various machine learning models*

Different models presented in the above section were first tested for the classification problem of predicting autoignition. Figure 10 illustrates the performance metrics of different models for predicting $I_g$. The metrics presented for each model correspond to three different test data sets as was explained in §3.4. In most of the cases, the performance metrics for all the three test cases remain nearly the same illustrating that the models were trained, have a small variance and were not overfitted. While the performance metrics of all the models seem to be quite satisfactory, *Random Forest* is able to make predictions most accurately. The performance is slightly better than *Decision Tree*, which as a similar approach, but *Random Forest* goes a step further to bootstrap random samples and eventually predicts the outcome for the tree with highest votes. Averaging over all the trees subsequently averages out the

variance in each tree thereby optimizing the bias and variance of the model in this classification problem. The performance of the support vector machine classifier *(SVC)* is also comparable with that of *Random Forest*. This can be ascribed to very clear distinguishment between the classification variable, $i.e.$ whether there will be autoignition or not, and this model is known to perform quite well in such cases. Further, this can also be attributed to no overlapping classes in the present case as each instance of input parameter is assigned only a single class.

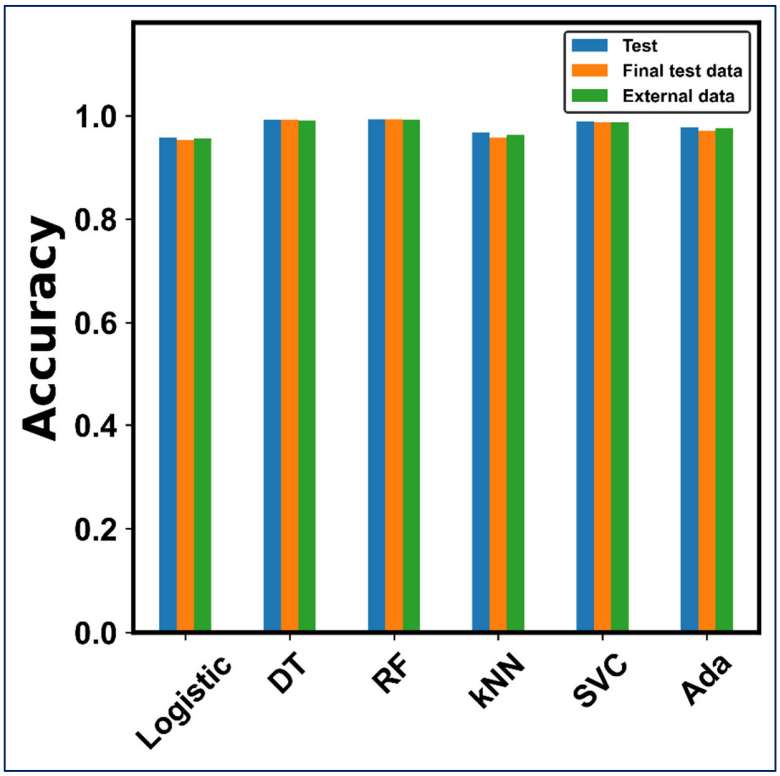



Figure 10: Accuracy of different models for the classification problem of identifying whether we have autoignition or not. For abbreviations, refer to the table after the conclusion.

The second part of the problem is to predict the ignition delay time. As was explained previously, here two different approaches are followed. We first present the results with method 1, $i.e.$ where ignition time is predicted independently of whether we have ignition or not. Figure 11 shows the performance metrics of various models used in this case. Here, it can be seen that except *Random Forest* most of the

models perform quite poorly in predicting the ignition time. The objective to illustrate these not so convincing results is to highlight that if not dealt with and applied appropriately, despite their enormous potential, machine learning models can result in very poor outcomes. One possible reason in the current case could arise due to training the models with exact numerical value and expecting the prediction to be exact, which is very unlikely especially within the limits of numerical accuracy. One means to improve the performance could be to classify the ignition delay time within certain ranges to make bins. This would reduce the error as instead of predicting the exact value, we have a certain tolerance to the exact value. This could also explain why *Random Forest* performs better as compared to other models. Nevertheless, the poor performance of most of the models could primarily arise due to imposition of zero as ignition time for no autoignition cases. These cases, which even in training set seem to have incurred manual bias, further worse the model performance owing to likelihood of predicted values being close to zero which otherwise would resemble no autoignition condition.

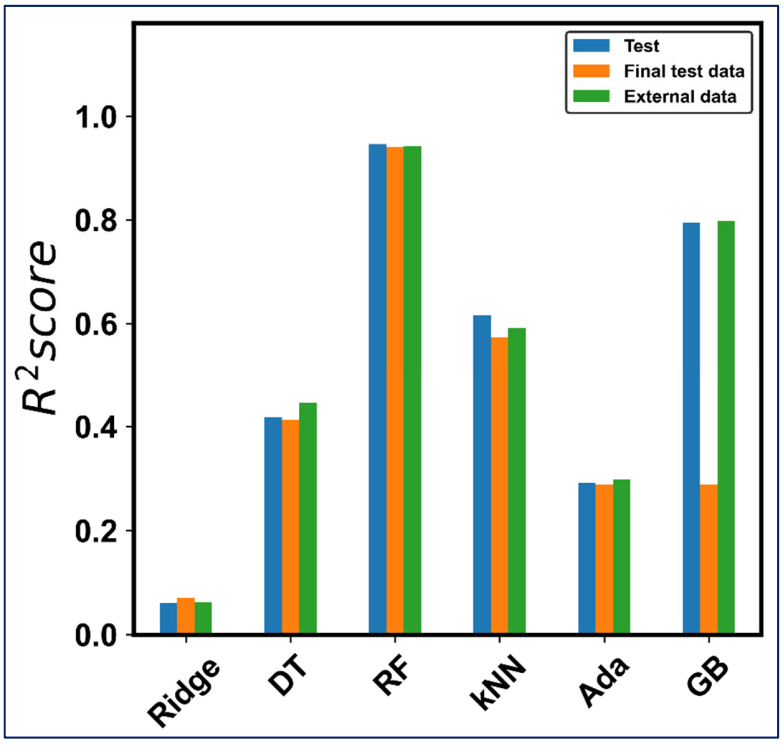



Figure 11: $R^2$ values for different models for predicting ignition delay using method 1 as described in the text.

In order to be coherent with physical reasoning that ignition delay is to be predicted only when autoignition is predicted, we test method 2 as was described previously. The models in this approach are trained using only the data points which correspond to autoignition and thus all the cases corresponding to $I_g = 0$ are excluded. This results in $7891$ data points from data set 1. The rest of the procedure to define the training/validation, testing, and final test data is the same as in the previous case and explained in Figure 6. The only difference was that the number of data points were reduced. Hyper tuning of the parameters was performed as in the previous case using 8-fold cross-validation. Figure 12 shows performance metrics of different models on various test data sets. Here, for testing the external data, data points corresponding to only autoignition were used which comprised 2659 data points from the total 5000.

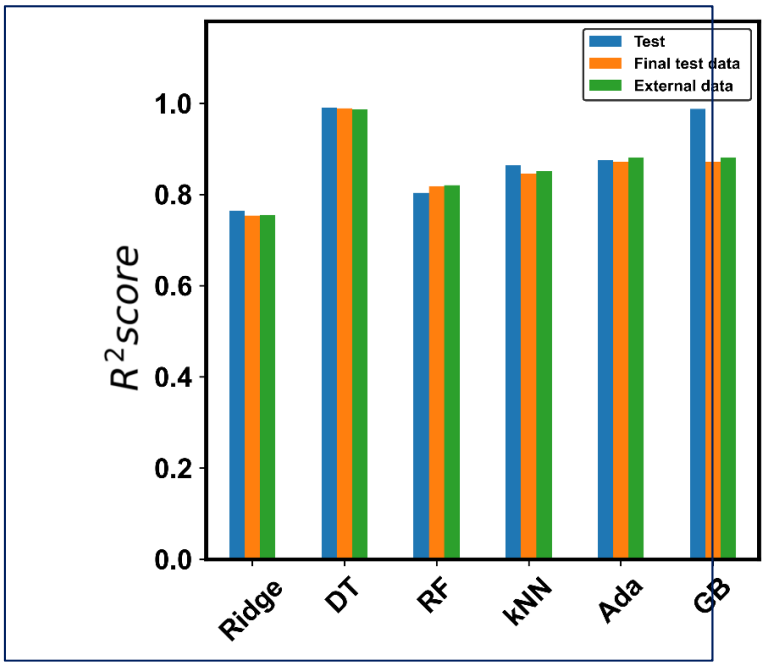
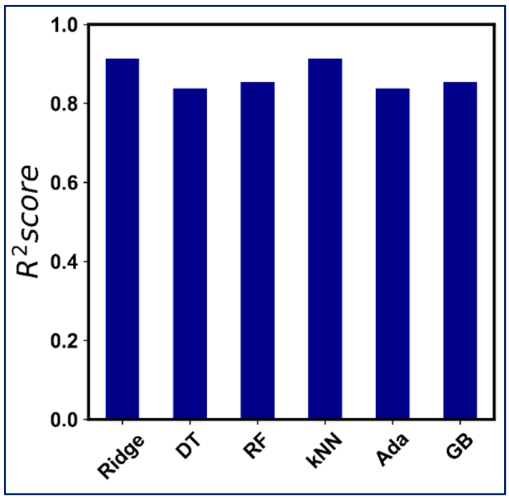



Figure 12: $R^2$ values for different models for predicting ignition delay using method 2 as described in the text.

Comparing the performance of various models using method 1 ( where we used the entire data to predict autoignition irrespective of autoignition and as shown in Figure 10), the performance of all the models is significantly better using method 2 as can be seen in Figure 12 ($i.e.$ predicting ignition delay time in a two-step process, firstly predicting autoignition and if found to occur, only then predict the ignition delay time). This shows that there was a substantial effect of inaccuracies that arose due to accounting for imposed ignition delay time in no autoignition cases. It may be argued that using some other value instead of zero could improve the performance using method 1. However, this may not make a large difference because there will be a significant percentage of the population with the same (hypothetical) ignition time delay which introduces manual bias in the data set itself. Therefore, in order to predict the ignition delay, the second approach seems to be more reasonable, given the condition that autoignition is well known. Thus, if the operating parameter is known to yield autoignition, we can calculate the ignition delay time quite precisely with various regression models as shown in Figure 12. However, in such case, attention must be paid as the error in the output would be a combined error due to two models, firstly due to predicting autoignition and subsequently in ignition delay time.

As a final step, we tested this approach to check how well the machine learning models as presented so far could find their utility in a real application. Here, the external data set serves as the test case while a more realistic case is presented in the next section. The input data (external data) is first tested for whether we have ignition or not. Subsequently, the data points for which autoignition is predicted are passed as input to the models predicting the ignition delay time. Let us define this output as $V_1$. For the remaining data points, $i.e.$ for which no autoignition was predicted, the corresponding output ignition delay time was set to zero to be coherent with how non ignition is defined in input data. Let us define this output as $V_2$. It is to be noted here that zero as ignition time serves no purpose for model development and is just a numerical value to complete the output vector. These two sets $(V_1, V_2)$ are combined to obtain the final predicted ignition delay time output $(\hat{y})$ and this is eventually compared with true values

520 $(y)$. As *Random Forest* had the best performance for predicting autoignition, this model was used in the
521 initial step of predicting for autoignition. Subsequently, all the aforementioned models were tested, and
522 their performance metrics are shown in Figure 13. It can be seen that there is a significant improvement
523 over the model performance as compared to method 1. We further present a comparison between actual
524 values and predicted ones for three different models. This is shown in Figure 14 where in an ideal case all
525 the points should like on $y = x$ straight line.




526

527 Figure 13: $R^2$ values for different models for predicting ignition delay for external data set.

This thereby presents that a two-model sequential approach needs to be followed to predict the desired output parameters with satisfactory accuracy.

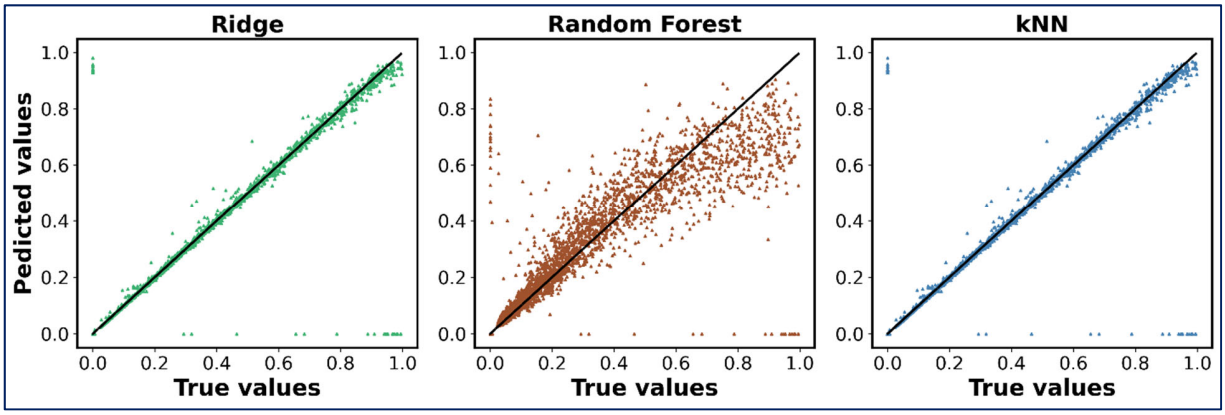



Figure 14: Comparison between actual and predicted values of ignition delay on external data set using methods as described in the text using three different models as mentioned therein.

As a final step, we finally analyzed the importance of various features on the model outputs. While this was briefly presented in the above section based on graphical visualization, here we used feature importance metrics available in *Random Forest* algorithm. Figure 15 presents the feature importance metrics in case of autoignition as well as ignition delay. It can be observed that temperature plays one of the major roles in governing the outcome of both the parameters followed by oxygen mass fraction. While there is a small difference in the extent to which these features are important, the overall trend remains the same and is coherent with physical understanding of the physical process considered in the present work.

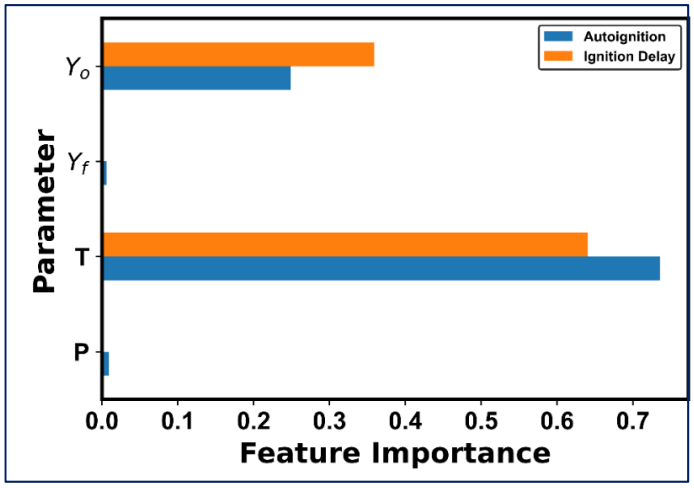



Figure 15: Feature importance metrics for autoignition and ignition delay time.

So far, we have highlighted how homogenous ignition calculations coupled with tentative flow time scale can be used to build data-driven machine learning models, which when trained judiciously can predict whether we can have autoignition as well as ignition delay time for a given set of operating parameters. However, a more important question which arises here is how these models can be further used to develop understanding as well improving the SCWO process at microscale. We discuss this aspect in next section.

## 5 Potential implication of ML models

In this section, we present a perspective on how these models and the approach in general can be used to better understand and improve upon SCWO process apart from the default utility in predicting the autoignition as well as ignition delay time. One of utilities of this approach is that the trained models can be used to map the flow field (in terms of temperature, fuel concentration, etc.) at a given time from CFD simulations to predict where in the flow domain autoignition is likely to occur and what could be the corresponding ignition delay time. This can provide insights into the probable region in the domain where ignition kernel could form, an essential and important aspect in understanding the flame dynamics. In order to explain this in a more qualitative way, we perform CFD simulations for the schematic shown in

Figure 2. Owing to symmetry in the domain (as highlighted in the figure) only half of the domain is simulated. The modeling approach is similar to the one described in [36] and is not been described here in detail as the prime objective is to demonstrate the described implication without any quantitively interpretations. Figure 16 shows an example of the generated 2D field of fuel and oxygen concentration along with temperature contours. It is to be mentioned that since the objective lies in predicting the autoignition condition, we are not concerned about what happens after the ignition using this methodology. The input data is fed to the machine learning model following which regions where ignition is likely to happen is identified. For the given configuration, it is likely that ignition will happen in regions dominated by oxidizer concentration which is expected. Mapping the ignition delay time on this contour plot shows regions with small ignition delay time thereby highlighting the likely regions for the formation of an ignition kernel.

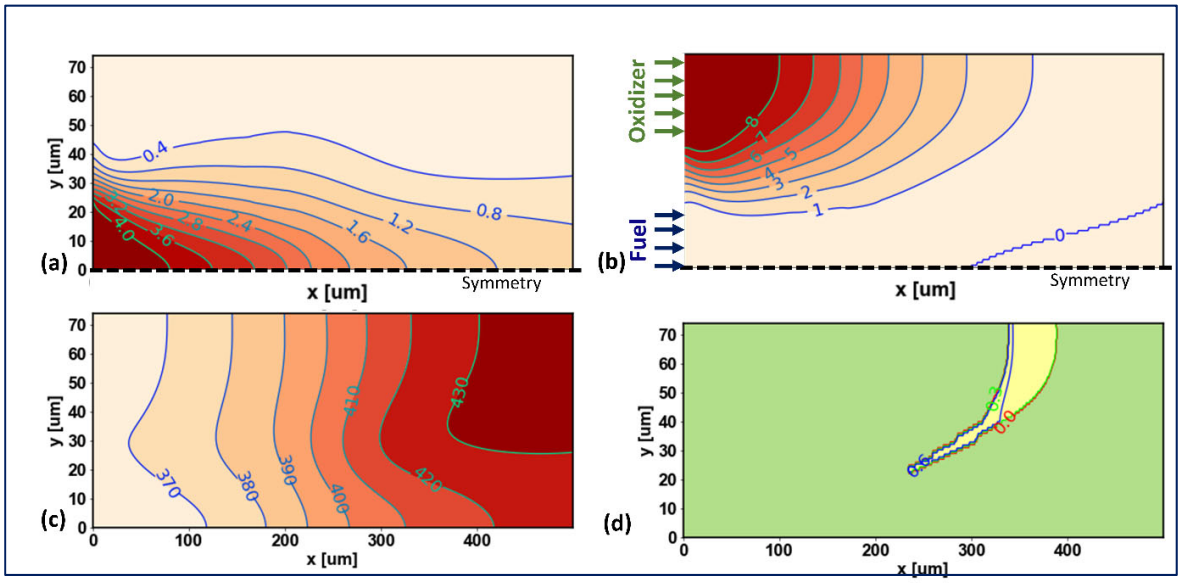



Figure 16: Illustration of how ML models can assist in predicting ignition kernel region. 2D field generated from simulation. (a) Fuel mass percentage (b) Oxygen mass percentage (c) Temperature [in C] (d) Machine learning model prediction of autoignition zone (yellow) and ignition delay time.

Further implications of this methodology are briefly described as follows. In a real scenario, autoignition will depend on how the flow field evolves and thus hydrodynamics is bound to play an important role. When the fuel and the oxidizer are injected into the reactor microchannel, the temperature of the mixture element will start to increase due to the reaction as the fluid element advects downstream. Thus, even if we started with conditions corresponding to probability of no ignition as per homogeneous calculation, the conditions can eventually change as the fluid element is advected and thus we can have conditions leading to autoignition later in time. In such a scenario, it will be intuitive to have the design of reactors, such as extending flow paths, using bluff bodies, etc. to alter hydrodynamics and mixing dynamics, which ensure that such conditions are attained as close as possible to the inlet region. Thus, rather than running a large number of simulations or experiments, mapping the simulation fields from several designs to predict ignition as illustrated ~~previously~~ can significantly aid in designing efficient microreactors. Furthermore, a similar mapping can be done for ignition delay time. This can provide insights for better understanding the impact of hydrodynamics on the autoignition phenomenon. Besides, the methodology can be extended in analyzing the impact of various process parameters, such as minimum inlet temperature at which ignition can occur, minimum fuel and oxidizer concentration for wide range of physical dimensions of the microreactor. A more important implication would lie in 3D analysis of the system. Owing to the small scale of the system, running a 3D simulation can be computationally very expensive. This may lead to limiting the parametric space of various geometric parameters to be explored and thus constraint optimum microreactor design. However, extending the methodology from 2D systems, an initial estimate can be made in 3D design configurations. Thus, the implications of the current approach of developing machine learning models are not only restricted to the trivial application of predicting autoignition conditions but can significantly improve the understanding and design considerations of microreactor in $\mu SCWO - H$. The presented approach will be used in the future with real simulation data to understand the dynamics of hydrothermal flames at microscale.

## 6 Conclusion and perspectives.

Machine learning based models were trained and tested to predict autoignition for a given set of operating parameters (in terms of pressure, temperature, and fuel and oxidizer concentration) leading to the formation of hydrothermal flames for supercritical water oxidation at microscale for its application in the space industry. The autoignition criteria was defined in relation to residence time of the fluid/reacting element in the microreactor. Thus, despite homogeneous reaction cases which may always yield autoignition, this was restricted in the present case owing to limitation by flow time scales involved. Among several classification models, *Random Forest* and *Support Vector Machine* were able to predict outcomes with high accuracy. Subsequently, regression models were used to predict ignition delay time where a two-step sequential strategy consisting of first predicting autoignition followed by ignition delay time for the corresponding cases. The developed approach was further tested on an example of a simple model system with 2D simulation to highlight the potential of using Machine Learning models beyond simple prediction of autoignition and ignition delay time. Several further implications and utility of the machine learning methodology were presented in the context of present problem highlighting how it can assist in understanding the onset of hydrothermal flames at microscale and eventually design efficient microreactors. As a future perspective, it is intended to couple the hydrodynamics of jets or co-axial flow to account for more intricate hydrodynamic time scales into the machine learning models. This will further improve the predictability of these models.

**Acknowledgements**

The authors are grateful to the French space agency CNES for funding the postdoctoral position of Deewakar Sharma and for financial support awarded through GdR MFA.

**Abbreviations used for various Machine learning models**

| *Abbreviation* | *Name* |
|---|---|
| Ridge | Ridge regression model |
| DT | Decision Tree |
| RF | Random Forest |
| kNN | k-nearest neighbor |
| Ada | AdaBoost |
| GB | Gradient Boost |
| SVC | Support vector classifier |

620

621 **References**


622 [1] Hicks MC, Hegde UG, Fisher JW. Investigation of Supercritical Water Phenomena for Space and
623 Extraterrestrial Application. NASA; 2012.
624 [2] Hicks MC, Hegde UG, Kojima JJ. Hydrothermal ethanol flames in Co-flow jets. The Journal of
625 Supercritical Fluids 2019;145:192-200.
626 [3] Bermejo MD, Cocero MJ. Supercritical water oxidation: A technical review. AIChE Journal
627 2006;52(11):3933-51.
628 [4] Serikawa RM, Usui T, Nishimura T, Sato H, Hamada S, Sekino H. Hydrothermal flames in
629 supercritical water oxidation: investigation in a pilot scale continuous reactor. Fuel
630 2002;81(9):1147-59.
631 [5] Augustine C, Tester JW. Hydrothermal flames: From phenomenological experimental
632 demonstrations to quantitative understanding. The Journal of Supercritical Fluids 2009;47(3):415-
633 30.
634 [6] Sierra-Pallares J, Teresa Parra-Santos M, García-Serna J, Castro F, José Cocero M. Numerical
635 modelling of hydrothermal flames. Micromixing effects over turbulent reaction rates. The Journal
636 of Supercritical Fluids 2009;50(2):146-54.
637 [7] Wellig B, Weber M, Lieball K, Príkopský K, Rudolf von Rohr P. Hydrothermal methanol diffusion
638 flame as internal heat source in a SCWO reactor. The Journal of Supercritical Fluids 2009;49(1):59-
639 70.
640 [8] Deewakar Sharma, Arnaud Erriguible, Olivier Nguyen, Carole Lecoutre, Yves Garrabos, et al.. A
641 preliminary investigation of microreactor designs for supercritical water oxidation using
642 hydrothermal flames for space applications. *13th International Symposium on Supercritical Fluids*,
643 ISASF (International Society for the Advancement of Supercritical Fluids), May 2022, Montréal,
644 Canada. 2 p. ⟨hal-04129176⟩
645 [9] A. Cario, C. Fauveau, O. Nguyen, C. Lecoutre, Y. Garrabos, Marre. S. Microreacteurs en saphir
646 (CNRS et CNES). 2020.
647 [10] Brunton SL, Noack BR, Koumoutsakos P. Machine Learning for Fluid Mechanics. Annual Review of
648 Fluid Mechanics 2020;52(1):477-508.

[11] Brenner MP, Eldredge JD, Freund JB. Perspective on machine learning for advancing fluid mechanics. Physical Review Fluids 2019;4(10):100501.
[12] Brunton SL. Applying machine learning to study fluid mechanics. Acta Mechanica Sinica 2021;37(12):1718-26.
[13] Zhang T, Li Y, Li Y, Sun S, Gao X. A self-adaptive deep learning algorithm for accelerating multi-component flash calculation. Computer Methods in Applied Mechanics and Engineering 2020;369:113207.
[14] Liu Y, Hong W, Cao B. Machine learning for predicting thermodynamic properties of pure fluids and their mixtures. Energy 2019;188:116091.
[15] Zhu B, Zhu X, Xie J, Xu J, Liu H. Heat Transfer Prediction of Supercritical Carbon Dioxide in Vertical Tube Based on Artificial Neural Networks. Journal of Thermal Science 2021;30(5):1751-67.
[16] Sun F, Xie G, Song J, Li S, Markides CN. Thermal characteristics of in-tube upward supercritical CO2 flows and a new heat transfer prediction model based on artificial neural networks (ANN). Applied Thermal Engineering 2021;194:117067.
[17] Lashkarbolooki M, Vaferi B, Shariati A, Zeinolabedini Hezave A. Investigating vapor–liquid equilibria of binary mixtures containing supercritical or near-critical carbon dioxide and a cyclic compound using cascade neural network. Fluid Phase Equilibria 2013;343:24-9.
[18] Wang K, Luo J, Wei Y, Wu K, Li J, Chen Z. Practical application of machine learning on fast phase equilibrium calculations in compositional reservoir simulations. Journal of Computational Physics 2020;401:109013.
[19] Zheng Z, Lin X, Yang M, He Z, Bao E, Zhang H, et al. Progress in the Application of Machine Learning in Combustion Studies. ES Energy & Environment 2020;9:1-14.
[20] Zhou L, Song Y, Ji W, Wei H. Machine learning for combustion. Energy and AI 2022;7:100128.
[21] Pan Y, Jiang J, Wang R, Cao H, Cui Y. Predicting the auto-ignition temperatures of organic compounds from molecular structure using support vector machine. Journal of Hazardous Materials 2009;164(2):1242-9.
[22] Suleiman MA, Owolabi TO, Adeyemo HB, Olatunji SO. Modeling of autoignition temperature of organic energetic compounds using hybrid intelligent method. Process Safety and Environmental Protection 2018;120:79-86.
[23] Shah N, Zhao P, DelVescovo D, Ge H. Prediction of Autoignition and Flame Properties for Multicomponent Fuels Using Machine Learning Techniques. SAE International; 2019.
[24] Cui Y, Wang Q, Liu H, Zheng Z, Wang H, Yue Z, et al. Development of the ignition delay prediction model of n-butane/hydrogen mixtures based on artificial neural network. Energy and AI 2020;2:100033.
[25] vom Lehn F, Cai L, Copa Cáceres B, Pitsch H. Exploring the fuel structure dependence of laminar burning velocity: A machine learning based group contribution approach. Combustion and Flame 2021;232:111525.
[26] Cui Y, Liu H, Wang Q, Zheng Z, Wang H, Yue Z, et al. Investigation on the ignition delay prediction model of multi-component surrogates based on back propagation (BP) neural network. Combustion and Flame 2022;237:111852.
[27] Huang Y, Jiang C, Wan K, Gao Z, Vervisch L, Domingo P, et al. Prediction of ignition delay times of Jet A-1/hydrogen fuel mixture using machine learning. Aerospace Science and Technology 2022;127:107675.
[28] Koido K, Hirosaka K, Kubo T, Fukayama M, Ouryouji K, Hasegawa T. Numerical study on premixed hydrothermal combustion in tube reactor. Combustion Theory and Modelling 2009;13(2):295-318.
[29] Quinlan JR. Induction of decision trees. Machine Learning 1986;1(1):81-106.
[30] Breiman L. Random Forests. Machine Learning 2001;45(1):5-32.

[31] Cortes C, Vapnik V. Support-vector networks. Machine Learning 1995;20(3):273-97.
[32] Cover T, Hart P. Nearest neighbor pattern classification. IEEE Transactions on Information Theory 1967;13(1):21-7.
[33] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research;12:2825-30.
[34] Hoerl AE, Kennard RW. Ridge Regression: Biased Estimation for Nonorthogonal Problems. Technometrics 2000;42(1):80-6.
[35] Hossin M, Sulaiman MN. A Review on Evaluation Metrics for Data Classification Evaluations. International Journal of Data Mining & Knowledge Management Process 2019;5(2):1-11.
[36] Zhang F, Marre S, Erriguible A. Mixing intensification under turbulent conditions in a high pressure microreactor. Chem Eng J 2020;382:122859.