Progress in Energy and Combustion Science 91 (2022) 101010

Contents lists available at ScienceDirect

Progress in Energy and Combustion Science

journal homepage: www.elsevier.com/locate/pecs

Combustion machine learning: Principles, progress and prospects

Matthias Ihme [*,a,b], Wai Tong Chung [a], Aashwin Ananda Mishra [b]

[a] Department of Mechanical Engineering, Stanford University, Stanford, CA 94305, USA
[b] SLAC National Accelerator Laboratory, Menlo Park, CA 94025, USA

ARTICLE INFO

Keywords:
Machine learning
Data-driven methods
Combustion

ABSTRACT

Progress in combustion science and engineering has led to the generation of large amounts of data from large-scale simulations, high-resolution experiments, and sensors. This corpus of data offers enormous opportunities for extracting new knowledge and insights—if harnessed effectively. Machine learning (ML) techniques have demonstrated remarkable success in data analytics, thus offering a new paradigm for data-intense analyses and scientific investigations through combustion machine learning (CombML). While data-driven methods are utilized in various combustion areas, recent advances in algorithmic developments, the accessibility of open-source software libraries, the availability of computational resources, and the abundance of data have together rendered ML techniques ubiquitous in scientific analysis and engineering. This article examines ML techniques for applications in combustion science and engineering. Starting with a review of sources of data, data-driven techniques, and concepts, we examine supervised, unsupervised, and semi-supervised ML methods. Various combustion examples are considered to illustrate and to evaluate these methods. Next, we review past and recent applications of ML approaches to problems in combustion, spanning fundamental combustion investigations, propulsion and energy-conversion systems, and fire and explosion hazards. Challenges unique to CombML are discussed and further opportunities are identified, focusing on interpretability, uncertainty quantification, robustness, consistency, creation and curation of benchmark data, and the augmentation of ML methods with prior combustion-domain knowledge.

1. Introduction

1.1. Motivation

Progress in the field of combustion science and engineering is inexorably linked to data. Perhaps most relevant are fundamental databases of thermochemical properties that have been painstakingly compiled from measurements of thousands of chemical compounds, mixtures, and reaction systems [1–3]. These data have been fundamental to the specification of thermodynamic quantities, chemical kinetic rates, and the evaluation of equilibrium states. Experimental data for reacting flow systems have contributed to our understanding of the coupling between heat-release, fluid dynamics, species conversion, and flame structure in turbulent reacting flows [4]. While early measurements were largely limited to single-point measurements from thermocouples, hot-wires, and sample probes at modest acquisition rates, recent advances in imaging techniques, high-energy laser sources, and high-speed data acquisition systems have enabled planar, tomographic, and simultaneous multi-species measurements at data rates in excess of 400 GB/s [5,6]. Sensors are another important source of data for control, advanced prognostics, and health monitoring in gas turbines, furnaces, and other energy-conversion systems. With the increasing complexity of such systems, the number of sensors has been growing in proportion, so that modern aircraft gas turbines are estimated to generate between 2 GB and 2 TB of data during a transatlantic flight [7–9].

The detection, prevention, and mitigation of wildfires is another important area that heavily relies on observational data, which are primarily generated from satellites, aerial monitoring, and patrol. In particular, satellites provide continuous measurements of vegetation density, smoke emissions, moisture, aerosols, surface temperature, and other meteorological data at acquisition rates in excess of 100 GB/day [10–15]. Heterogeneous data from these sources at varying spatiotemporal resolution are continuously processed for active fire detection, wildland fire management, and mapping of fire severity [16,17].

Apart from sensing, observations, and experimental measurements, computational simulations are another significant source of data. Specifically, the increasing availability of computational resources has enabled remarkable advances in numerical simulations of turbulent reacting flows at increasingly higher fidelity, spatial resolution, and

* Corresponding author.
E-mail address: mihme@stanford.edu (M. Ihme).

https://doi.org/10.1016/j.pecs.2022.101010
Received 16 March 2021; Accepted 24 March 2022

Nomenclature

Abbreviations

ANN	Artificial neural network
CNN	Convolutional neural network
MLP	Multilayer perceptron
GAN	Generative adversarial network
LSTM	Long short-term memory
RNN	Recurrent neural network
SVM	Support vector machine
ODE	Ordinary differential equation
ML	Machine learning
CombML	Combustion machine learning
Sci(Eng)ML	Scientific (and engineering) machine learning
PDF	Probability density function
PMF	Probability mass function
MLE	Maximum likelihood estimation
QoI	Quantity of interest
GA	Genetic algorithm
PCA	Principal component analysis
RL	Reinforcement learning
RANS	Reynolds-averaged Navier-Stokes
LES	Large-eddy simulation
DNS	Direct numerical simulation
SGS	Subgrid scale
MARS	Multivariate adaptive spline regression
LEM	Linear-eddy model
GEP	Gene expression programming
ELM	Extreme learning machine
BNN	Bayesian neural network

Greek Symbols

$\delta(\zeta)$	Dirac function
$\dot{\omega}$	Chemical source term
ρ	Density
τ	Viscous stress tensor
θ	Model parameters
σ	Sigmoidal function
ϕ	Thermochemical state

Roman Symbols

$\mathbf{E}$	Expectation
X	Feature sample space
Y	Target sample space
$\mathbf{V}$	Variance
N	Number, samples
C	Molar species concentration vector
p	Pressure
u	Velocity
q	Heat flux
j	Diffusive mass flux
e_t	Total specific energy
e	Specific sensible and chemical energy
I	Identity matrix
K	Model complexity
Pr	Probability
$p_x(X)$	Probability density function of continuous random quantity x
$p_x^*(X)$	Fine-grained distribution of quantity x
$P_x(X)$	Probability mass function of discrete random quantity x
M	Dimension of feature space, input data
E	Error measure/objective function
f	Hypothesis learned from data
F	Target function encapsulated by data
x	Input data (features)
y	Output data (target)
w	Model parameters, weight coefficients
b	Bias coefficient
i	Impurity measure

Symbols and Parameters

$\mathscr{E}$	Generalization error
$\mathscr{H}$	Hypothesis set
$\mathscr{O}$	Order
$\mathscr{L}$	Dataset or sample space
$\mathscr{X}$	Set of input data, features
$\mathscr{Y}$	Set of output data, target
$\mathscr{P}$	Set of model parameters
$\mathscr{M}$	Low-dimensional manifold

Subscripts

D	Spatial dimension
l	Labeled data
M	Mesh
S	Species
U	Independent solution variables
u	Unlabeled data

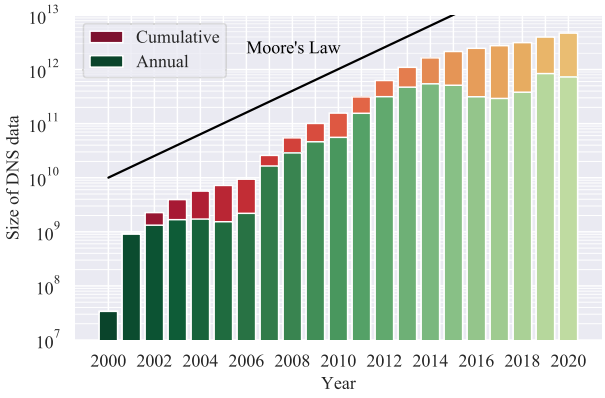
physical complexity. This is illustrated in Fig. 1, showing data from a survey of more than 200 published direct numerical simulation (DNS) studies on turbulent reacting flows over the last two decades [18–239]. Details about these DNS studies are provided as supplementary material.

The annual and cumulative data (Fig. 1a) are computed in terms of degree of freedom per timestep as $N_M \times N_U$ (with N_M as the mesh size) and $N_U = N_S + N_D + 1$ as the total number of solution variables (with N_S and N_D the number of species and the spatial dimension, respectively). Considering that each DNS is typically evolved over $\mathscr{O}(10^6)$ timesteps to capture the relevant flame dynamics and $\mathscr{O}(10^2)$ instantaneous solution fields are stored for analysis, more than 10 PB of structured data are available for analysis.

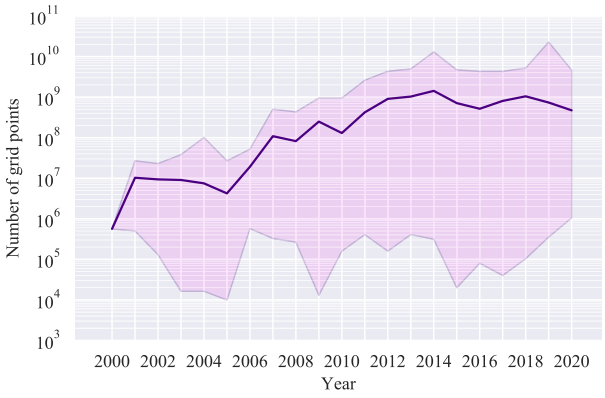
This nearly continuous growth in data (Fig. 1a) is not only linked to advances in the scalability and availability of high-performance computing systems but also to the maturation of combustion-simulation tools that have enabled the consideration of increasingly more complex combustion problems; over the last 20 years, the average mesh size increased by more than three orders of magnitude (Fig. 1b). The availability of computational resources has also allowed researchers to increase the chemical complexity; it is now possible to perform reactive DNS studies by considering upward of $\mathscr{O}(30)$ species in three-dimensional geometries (Fig. 1c), which has enabled representations of highly relevant chemical processes involving low-temperature radical chemistry, soot, pollutant formation, and multiphase combustion.

1.2. Data, information, and knowledge

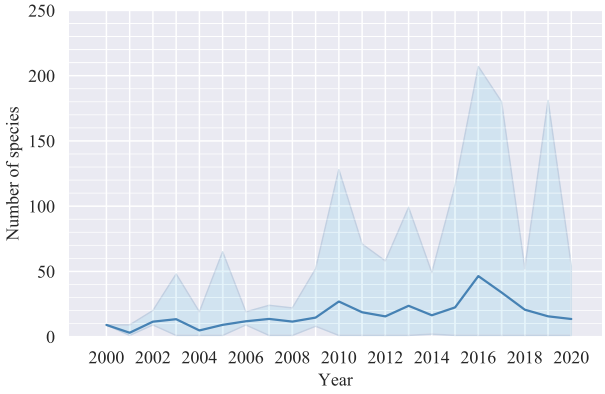
With the continuous growth in data volume, the managing and processing of data plays a critical role in converting data into information that can eventually contribute to the generation of knowledge. Commonly, a hierarchical view is taken to distinguish among data, information, and knowledge [240,241] (Fig. 2). Data are created from observations, experiments, simulations, and theory in the form of raw facts, descriptors, or numerical values. Data management is primarily concerned with the collection, curation, culling, organization, and transfer of data. Data processing involves transformation into



(a) Cumulative data from DNS.



(b) Annual growth in mesh resolution.



(c) Annual growth in number of chemical species.

Fig. 1. Analysis of more than 200 published DNS studies [18–239] over the last two decades, showing (a) annual and cumulative data (degree of freedom per timestep) and evaluation of (b) mesh size and (c) number of chemical species. The shaded region indicates the maximum and minimum quantity, and the solid line is the mean value.

information through analysis, processing, and manipulation as well as combination with other data, models, and theory. Knowledge, in turn, is derived from information through application, the generation of new insights, and the development of ideas. This data-transformation process introduces several challenges pertaining to data management and analysis. Various methods have been established to process and analyze data [242,243]. However, traditional techniques that rely on statistical methods, data reduction, and visualization are expected to reach their limits in the presence of today's growing data volume and increasing physical complexity.

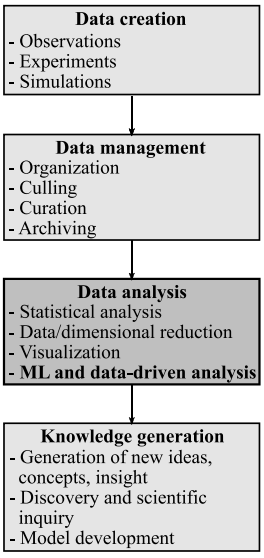


Fig. 2. Hierarchical view of data conversion and transformation to information through data analysis and knowledge generation. This review is concerned with machine learning and data-driven analysis.

Renewed interest in conjunction with recent progress in machine learning (ML) offers new opportunities for data analysis in combustion applications. In addition to the aforementioned increasing data volume (see Fig. 1), key enabling factors for increasing the utilization of data-driven methods include advances in computational resources, custom ML hardware (such as graphics/tensor processing units and application-specific integrated circuits), and data storage technologies for large datasets, improved accessibility to ML techniques through open-source software libraries [244–247], and the flexibility of ML techniques for a wide range of applications involving pattern recognition, regression, classification, clustering, dimensional reduction, and control.

1.3. Knowledge-discovery paradigms

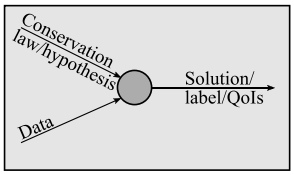
Our current understanding of combustion and thermofluid flows has largely been derived from physics-based principles. This genesis stands in contrast to data-driven approaches in which a hypothesis or a set of rules in the form of an explanatory model are derived from data (Fig. 3). In traditional physics-based approaches (Fig. 3a), conservation laws (or physics-based rules) that are derived from first principles and data in the form of model parameters, boundary conditions, and initial states are supplied to a computational model or mathematical expression, which outputs the solution in the form of realizations or response functions.

Thermodynamic principles, conservation laws, and constitutive relations are derived from first principles to describe the evolution of chemically reacting flows and combustion systems. At the continuum level, the set of governing equations for conservation of momentum, species, and energy can be written in the form [248]:

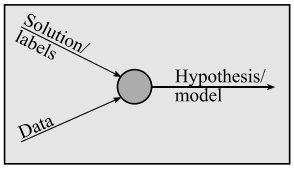
$$\partial_t U + \nabla\cdot F(U) - \nabla\cdot Q(U,\nabla U) = S(U)\,, \tag{1}$$

where $U \in \mathbb{R}^{N_U}$ is the conservative state vector, $F \in \mathbb{R}^{N_U\times N_D}$ is the inviscid flux, $Q \in \mathbb{R}^{N_U\times N_D}$ is the viscous-diffusive flux, and $S \in \mathbb{R}^{N_U}$ is the vector of source terms. The individual terms in Eq. (1) can be expanded as:

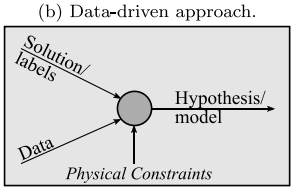
$$U = \begin{pmatrix} \rho u \\ C \\ \rho e_t \end{pmatrix}, \tag{2a}$$



(a) Physics-based approach.



(b) Data-driven approach.



(c) Knowledge-guided data-driven approach.

Fig. 3. Schematic comparing knowledge-discovery paradigms of (a) physics-based approaches, (b) data-driven approaches, and (c) hybrid knowledge-guided data-driven approaches. QoIs, quantities of interest.

$$F = \begin{pmatrix} \rho u \otimes u + p\mathrm{I} \\ C \otimes u \\ u(\rho e_t + p) \end{pmatrix}, \tag{2b}$$

$$Q = \begin{pmatrix} \tau \\ -j \\ \tau \cdot u - q \end{pmatrix}, \tag{2c}$$

$$S = \begin{pmatrix} 0 \\ \dot{\omega} \\ 0 \end{pmatrix}, \tag{2d}$$

where ρ is the density, $u \in \mathbb{R}^{N_D}$ is the velocity vector, $C \in \mathbb{R}^{N_S}$ is the vector of molar species concentrations, $e_t = e + |u|^2/2$ is the specific total energy, e is the specific sensible and chemical energy, p is the pressure, and $\dot{\omega} \in \mathbb{R}^{N_S}$ is the vector of chemical source terms. The constitutive relations describing the viscous stress tensor $\tau \in \mathbb{R}^{N_D \times N_D}$, the diffusive mass flux $j \in \mathbb{R}^{N_S \times N_D}$, and the heat flux vector $q \in \mathbb{R}^{N_D}$ are typically represented by Newton's law, multicomponent or mixture-averaged diffusion models, and Fourier's relation, respectively [249]. The system of equations, Eq. (1), is closed with a state equation, which is here written in implicit form:

$$g(p, e, C) = 0\,. \tag{3}$$

Physical principles for the conservation of mass and species require

$$W^{\mathsf{T}} C = \rho\,, \quad W^{\mathsf{T}} \dot{\omega} = 0\,, \quad W^{\mathsf{T}} j = 0\,, \tag{4}$$

where $W \in \mathbb{R}^{N_S}$ is the vector of molecular weights of all species. Secondary conservation principles for entropy, kinetic energy, and other derived quantities are obtained through manipulation of Eq. (1) [248].

While Eq. (1) provides an exact description of combustion-physical processes at the continuum level, direct solution or experimental evaluation become infeasible for the following reasons. First, the spatio-temporal scales associated with large-scale flow dynamics, turbulence, scalar mixing, chemical reactions, and heat release span several orders of magnitude, making it infeasible to resolve all scales. Second, the chemical complexity, which requires considering a large number of species and elementary reaction steps, limits the detailed simulation of combustion systems involving complex and multicomponent transportation fuels [250]. Third, incomplete knowledge of thermodynamic response functions, transport properties, rate coefficients, and other constitutive models limits predictive accuracy. Fourth, the chaotic behavior of turbulent flows amplifies small flow-field perturbations; the exponential growth of these perturbations restricts the time horizon over which these flows can be accurately predicted and scales with the inverse of the maximum Lyapunov exponent [251,252].

Addressing these issues has been the subject of active research and significant progress has been made. In particular, low-pass filtering techniques are commonly employed for separating large-scale processes that evolve on resolved scales and processes that occur on numerically unresolved scales [253–256]. Closure models in the form of algebraic or differential equations for turbulence/chemistry coupling, turbulent stresses, and turbulent transport have been developed using physical arguments. Efficient chemical reduction techniques [257–261] and reduced-manifold methods [262–268] are now well established for constructing compact kinetic mechanisms and low-dimensional combustion manifolds with controlled accuracy and dimensionality of the reduced state-vector $N_{S'}$, with $N_{S'} \ll N_S$. In addition, the consideration of sensitivities, uncertainties, and inter-species dependencies through data-centric methods (encapsulated in the Process Informatics Model (PrIMe) [269], multi-dataset optimization approaches [270], and Bayesian inference and uncertainty quantification [271–274]) have resulted in major advances in the development of chemical kinetic mechanisms and thermodynamic properties with ever-increasing levels of accuracy. These data-centric approaches rely on rules in the form of prior information (such as physics-based principles and conservation laws) or assumptions about model parameters and a statistical description of the agreement of the model with the data. Data from experiments and other sources are then used to train model parameters, given available information.

In contrast to physics-based approaches, data-driven approaches are not constrained by physical principles, which enables applications to a wider range of problems. However, they rely on large datasets from which hypotheses and models are inferred (Fig. 3b). They offer attractive alternatives over traditional physics-based approaches because hypotheses about the form and structure of physical processes, universal properties, and relationships (such as conservation principles, constitutive relations, material-frame invariance, and symmetries) are not required to describe the system under investigation. Additionally, human experts are limited in their ability to extract structures and knowledge from complex combustion data. In contrast, given adequate computing hardware, data-driven methods can learn and extract complex structures from extremely large sets of high-dimensional data.

However, despite their ability to adapt to various problems, purely data-driven approaches are not expected to be truly predictive [275]. The primary reasons for this expectation are the lack of the resulting models to obey universal properties and fundamental governing laws that are intrinsic to the combustion-physical system, the lack of sufficient data to fully parameterize complex thermofluid systems, and difficulties in generalizing these models to scenarios on which they have not been trained. Therefore, knowledge-guided data-driven approaches have been developed [276–279] (Fig. 3c). These approaches are frequently referred to as "physics-informed," "physics-guided," or "physics-aware" methods. However, since the field of combustion encompasses disciplines other than physics and these hybrid approaches enable the accommodation of other scientific and engineering knowledge, it is more appropriate to adopt this generic nomenclature in the context of combustion machine learning (CombML). These hybrid methods bridge the gap between purely physics-based and data-driven methods. Knowledge-guided data-driven approaches encode prior knowledge, physical constraints, and mathematical operators into the ML model to achieve consistently accurate predictions and

generalization. This knowledge guides the general structure of the model, while the data provide structural refinement and parameterization of the model. Other benefits of these hybrid models include improved accuracy when dealing with incomplete and incorrect data, less data required for training, and improved generalizability and interpretability [277,280,281]. Similar to other approaches, knowledge-guided data-driven methods are frameworks that are versatile for a wide range of physical problems.

With relevance to the application of these approaches to combustion science and engineering, a variety of factors, requirements, and constraints determine the selection of a particular paradigm. Chief among them are the amount of available data and the underlying knowledge of the combustion system under consideration (Fig. 4). For example, the discovery of thermochemical response functions from comprehensive measurements of chemical compounds in the absence of fundamental knowledge for describing complex species properties is an ideal application for data-driven methods. In contrast, the lack of multidimensional and species-resolved measurements of turbulent combustion processes in propulsion systems demands a knowledge-guided and data-driven approach in which the construction of data-driven models is augmented by physical constraints and conservation principles to obtain robust model predictions.

1.4. Terminologies and definitions

This section introduces common terminologies and definitions that will be used throughout this article [282–286].

Artificial intelligence The field of study dealing with enabling computer-based systems to perform sophisticated tasks, such as automated reasoning, language translation, or visual perception.

Channels Channels, commonly used in the context of convolutional neural networks, refer to additional dimensions in the network architecture to account for specific aspects, such as colors, in the input data.

Classification A mathematical mapping from unlabeled instances to discrete classes. Here, the output of the model is categorical.

Clustering The task of grouping objects of a set into distinct groups, so that objects assigned to the same group are more similar than objects in different groups.

Combustion machine learning (CombML) Application of ML techniques to combustion.

Deep learning Deep learning is a class of ML in which multilayer representations are utilized to extract hierarchical features from a complex input; common deep learning models are based on neural network architectures and include CNNs, recurrent neural networks (RNNs), generative adversarial networks (GANs), and deep belief networks.

Dimensionality reduction Transformation of the representation of data from a high-dimensional space into a lower-dimensional space, such that the transformed data retain the most useful information from the original dataset.

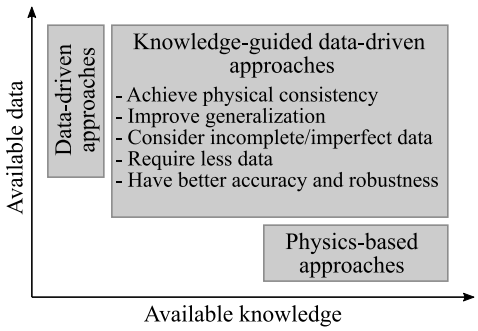


Fig. 4. Schematic of knowledge-discovery paradigms considering available data and prior knowledge. Figure adapted from [280].

Features and labels In supervised learning, an algorithm is trained with data points consisting of inputs and a corresponding output. Inputs are typically referred to as features, whereas outputs are typically referred to as labels. These terms are used interchangeably within this article.

Generalizability The ability of a ML model to make predictions with comparable accuracy for new and unseen samples taken from the distribution for which it was trained. Generalizability is often regarded as the key metric of effective learning.

Gradient descent An iterative approach to minimizing a given metric (usually in the form of a loss function) by estimating the gradient of the metric with respect to the parameters of the model, conditioned upon the training data. Model parameters are then adjusted by an increment along the direction of the maximal decrease of the metric.

Hyperparameter Hyperparameters are parameters that control the learning process of an ML algorithm. Examples include the learning rate for the optimizer, the number of layers or neurons in a neural network, and the regularization rate. Hyperparameters are traditionally ascertained using human expertise, intuition, or calibration with a validation dataset. Hyperparameters differ from model parameters that are determined using the training dataset and specify the ML model, such as the weights and biases in a neural network.

Hypothesis A hypothesis is a specific function $f : \mathscr{X} \rightarrow \mathscr{Y}$ that is ascertained to describe the target function $F : \mathscr{X} \rightarrow \mathscr{Y}$ that is encapsulated by the data; here, $\mathscr{X}$ denotes the set of input data and $\mathscr{Y}$ corresponds to the set of output data.

Inductive bias Inductive biases of a learning algorithm represent the set of assumptions in the approach that is not inferred from the training data. Inductive biases can arise due to the choice of the model class, the objective function, and/or the optimization strategy.

Learning problem A learning problem is generally concerned with improving the performance (or reducing a loss function) of executing a certain task through a training experience using a limited set of data.

Learning rate The size of the change in the model parameters for every gradient-descent iteration is determined by the gradient multiplied by a scalar, the learning rate. The learning rate is equivalent to the under-relaxation factor used in computational physics.

Loss/cost/objective function The function being optimized during ML training, which minimizes the error measure of an ML model. The loss function is typically used to refer to the minimized function of a single data point, while cost function refers to the averaged loss function of the dataset. These terms are subsets of the objective function. In this article, we use these four terms interchangeably.

ML ML is a branch of artificial intelligence that specifically focuses on enabling computer-based systems to infer predictive models from data. The notion of "learning" expresses the ability of such an algorithm to progressively improve its performance in a specific task by processing data and information. Traditionally, algorithms do not rely on explicit rules from domain experts for how to perform the task. However, a growing branch of ML demonstrates that learning can be improved by merging ML algorithms with domain knowledge.

ML algorithm An ML algorithm is a procedure that operates on data to identify a specific hypothesis from a set of candidates that optimizes the performance in representing the target function through training experience. A large number of ML algorithms have been developed for a wide range of applications; they differ in the representation of the hypothesis set, the procedure for selecting a hypothesis, and the exposure of the training data to the model.

ML model An ML model is the output of the ML algorithm, consisting of a hypothesis and model parameters that are employed for making predictions about data not seen during training.

Overfitting Overfitting describes the performance of a model that matches the training data so exactly that it exhibits significantly inferior performance on new data not seen during training.

Regression A mathematical mapping from unlabeled instances to a continuous range, or a metric space.

Regularization The set of approaches to adding information to ameliorate an ill-posed problem or to reduce the potential for overfitting. Classically, in ML applications, regularization is applied via a penalty term in the loss function that penalizes more complex models. This penalty forces the optimization procedure to select simpler functions that fit the training data well, as opposed to more complex functions that are prone to overfitting.

Reinforcement learning (RL) An approach to ML in which the goal is to learn an optimal policy that maximizes the return (or reward) accrued by an agent while interacting with an environment. The policy is a mapping from states of the environment to actions to be taken by the agent.

Scientific and engineering machine learning (SciEngML) Application of ML to problems encountered in scientific and engineering domains, by incorporating physical principles, conservation laws, and other constraints to construct interpretable ML models of multiscale and multiphysicochemical systems from sparse, low-fidelity, and heterogeneous data. Scientific ML was originally introduced to specifically address scientific discovery using ML techniques for scientific data [287]. To include engineering- and combustion-specific applications, we believe that SciEngML and CombML, respectively, are more fitting descriptors.

Semi-supervised learning Semi-supervised learning intersects supervised and unsupervised methods by learning from a combination of labeled and unlabeled data, $\{(x,y)_i, x_j\}_{i,j=1}^{N_l,N_u}$, typically with $N_u \ll N_l$. For example, semi-supervised learning methods have been applied in RL and statistical classification.

Supervised learning Supervised learning encompasses methods that learn from a collection of labeled data, $\{(x,y)_i\}_{i=1}^{N_l}$ to predict an outcome $\widehat{y}$ for an input $\widehat{x}$, through application of a learned hypothesis $f(\widehat{x})$. Common applications of supervised learning include classification and regression problems.

Underfitting Underfitting describes the performance of a model that exhibits poor predictive ability on the training data itself, as a consequence of its inability to capture the complexity of the data.

Unsupervised learning Unsupervised learning is concerned with extracting knowledge from unlabeled data $\{x_i\}_{i=1}^{N_u}$; it is commonly utilized for dimensional reduction and clustering.

1.5. Objective and outline

The objective of this article is to review recent progress and discuss open challenges in CombML techniques with specific application to combustion science and engineering. This article seeks to address a broad readership that includes those new to combustion and/or ML, domain experts in ML who wish to develop and apply ML methods to combustion problems, as well as engineers and researchers that are trained in traditional combustion and would like to acquire knowledge in ML methods.

In Section 2 we introduce the mathematical background and principles that are the foundation of many ML methods. This section is intended for readers that are not familiar with statistical methods and readers that would like to refresh their knowledge about statistical learning methods related to CombML. Section 3 is concerned with ML algorithms; we follow the convention of distinguishing among supervised, unsupervised, and semi-supervised methods. Recognizing that ML combines a wide range of learning techniques that are highly versatile and applicable to various problems, this section reviews the most common techniques and draws connections to combustion-specific applications. To this end, we complement the exposure of these techniques with representative examples of canonical combustion problems. These examples highlight salient feature of various ML algorithms and—by including the source code and data as supplementary material and in a GitHub repository [288]—provide a tutorial for readers interested in further exploration. With this background, applications of these methods to combustion problems of direct scientific and engineering interest are examined in Section 4. This section is structured into three separate topics: (i) fundamental combustion investigations, (ii) applications to propulsion and energy-conversion systems, and (iii) fire and explosion hazards, accidents, and safety management. Following the review of recent progress in CombML applications, in Section 5 we discuss research opportunities, open research questions, and outstanding challenges to successfully adopting and adapting ML techniques for combustion. A summary of this article is provided in Section 6.

With the rapid proliferation of ML into various disciplines of science and engineering, the field of combustion has greatly benefited from advances in related areas. Therefore, the interested reader is referred to the following monographs and reviews that provide additional information about ML methods and their application to science and engineering. In particular, the textbook by Bishop [282] is an accessible introduction to a wide range of ML techniques. For a more rigorous treatment of ML methods see Murphy [283], which is based on probability theory and model-based approaches. The monographs by Hastie et al. [289] and Goodfellow et al. [286] cover complementary subjects on statistical learning and deep learning and are recommended for readers with a background in statistics. Applying ML techniques to fluid mechanics was reviewed by Brunton et al. [290], while Jain et al. [291] provided a scoping review of ML methods for wildfire science and management. Other reviews that deliver a perspective on the infusion of data-intensive ML methods into various scientific and engineering areas, the use of deep learning for discovering features and structures from data, as well as the embedding of physical principles and knowledge into ML techniques are by Jordan and Mitchell [285], LeCun et al. [284], and Karniadakis et al. [277].

2. Mathematical background

This section provides relevant background information for statistical learning techniques and discusses key differences between physics-based computational approaches and data-driven methods (Fig. 3). Next, we outline a general supervised ML model in order to illustrate key concepts. By considering the example of linear regression, we compare and contrast ML to classical linear algebra and statistics viewpoints. With this foundation, various ML algorithms and applications to a wide range of combustion problems are discussed in Sections 3 and 4.

2.1. Probabilistic analysis

Readers familiar with turbulence have encountered probabilistic concepts in the context of frequentist interpretation, which is concerned with the probabilistic analysis of stochastic processes. In this interpretation, statistical analysis tools are employed to extract physical trends such as mean, variance, and higher moments in order to describe a random process. As such, probabilities are regarded as statistical representations of a process that can be generated from long-term sampling. In contrast, Bayesian interpretation associates probabilities with quantification of the uncertainty of an event or a hypothesis. It is therefore more general and statistical data are more commonly interpreted through a Bayesian viewpoint.

We distinguish between a probability density function (PDF) and a probability mass function (PMF). The PDF of a continuous random variable x is denoted by $p_x(X)$ and quantifies the probability of x taking on a particular value [253,292]:

$$p_x(X)dX = \Pr\{X \leq x < X + dX\} \,, \tag{5}$$

where X denotes the sample space variable and the probability Pr is defined as

$$\Pr\{X_a \leq x < X_b\} = \int_{X_a}^{X_b} p_x(X)dX \,. \tag{6}$$

Here, we follow convention employed in the combustion literature and introduce the subscript x to denote the random variable. The properties of a PDF require non-negativity and normalization, which can be written as:

$$p_x(X) \geq 0 \ , \tag{7a}$$

$$\int_{-\infty}^{\infty} p_x(X) dX = 1 \ . \tag{7b}$$

Extending Eq. (5) to multiple continuous random variables, x and y, results in a joint PDF $p_{x,y}(X,Y)$. The marginal PDF of x is then obtained by integration over the sample space Y:

$$p_x(X) = \int_{-\infty}^{\infty} p_{x,y}(X,Y) dY \ , \tag{8}$$

and the extension to more than two variables directly follows.

The conditional PDF $p_{x|y}(X|Y)$ provides the probability of a continuous random variable x for a particular value of y. The conditional PDF is computed from the product rule:

$$p_{x,y}(X,Y) = p_{x|y}(X|Y) p_y(Y) \ . \tag{9}$$

Rearranging Eq. (9) and using the symmetry of a joint PDF $p_{x,y}(X,Y) = p_{y,x}(Y,X)$ gives

$$p_{y|x}(Y|X) = \frac{p_{x|y}(X|Y) p_y(Y)}{p_x(X)} \ , \tag{10}$$

which is referred to as Bayes' theorem and provides a relationship between conditional probabilities.

Probabilities of random processes are often approximated by analytic functions [293]. The Gaussian (or normal) distribution is one of the most common PDFs for representing continuous random variables. To represent the distribution of a continuous positive variable, the lognormal or Gamma distribution is used. To represent the probability of bounded continuous variables, commonly encountered in representing chemical species or mixing processes, the beta-distribution is employed. Other distributions for modeling multiscalar mixing processes and for considering high-order statistical moments are the Dirichlet distribution, the bivariate beta distribution, and the statistically most-likely distributions.

Equation (10) is the foundation of Bayesian inference; the distributions are interpreted in the sense of uncertainties and hypothesis testing [282,294]. More specifically, this approach is employed in inferring specific quantities, model parameters, or a hypothesis from available data and observations. Denoting the parameters as θ and the available data by $\mathscr{L}$, Bayes' theorem can then be written as

$$p_{\theta|\ell}(\Theta|\mathscr{L}) = \frac{p_{\ell|\theta}(\mathscr{L}|\Theta) p_\theta(\Theta)}{p_\ell(\mathscr{L})} \ , \tag{11}$$

where $p_\theta(\Theta)$ is the prior probability that is constructed from assumptions in the absence of data and $p_\ell(\mathscr{L})$ is the marginal likelihood function that is evaluated from the data. The likelihood function $p_{\ell|\theta}(\mathscr{L}|\Theta)$ is computed from the data given the parameters and quantifies the probability of the data for different specifications of the parameter vector θ. From this formulation, the posterior probability $p_{\theta|\ell}(\Theta|\mathscr{L})$ can be evaluated, which quantifies the probability of the parameters Θ given the observations.

For the analysis of data that are obtained from infrequent measurements or sparse sampling, it is more appropriate to consider the data as discrete random variables. The distribution of a sequence of N random discrete samples $\{x_1, x_2, \ldots, x_N\} = \{x_i\}_{i=1}^N$ can then be represented by the fine-grained distribution

$$p_x^*(X) = \frac{1}{N} \sum_{i=1}^{N} \delta(X - x_i) \ , \tag{12}$$

which is a discrete distribution with equal weights; $\delta(\zeta)$ is the Dirac function. The asterisk denotes that Eq. (12) is a random distribution as it depends on random samples, x_i, that are obtained from measurements or are generated from a continuous distribution using, for instance, an acceptance-rejection method [295]. Fig. 5 illustrates the construction of a fine-grained PDF from a continuous distribution function.

Discrete random processes are analyzed using PMFs. A PMF can be derived from the fine-grained distribution by multiplying Eq. (12) with a constant function $\mathbb{I}(X) = 1$ and integrating over the sample space:

$$\int_{-\infty}^{\infty} p_x^*(X) dX = \frac{1}{N} \sum_{i=1}^{N} \int_{-\infty}^{\infty} \mathbb{I}(X) \delta(X - x_i) dX \ , \tag{13a}$$

$$= \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}(X = x_i) \ , \tag{13b}$$

$$= \sum_{i=1}^{N} P_x(X_i) \ , \tag{13c}$$

where we used the sifting property of the Dirac function and define $P_x(X_i) = \frac{1}{N} \mathbb{I}(X = x_i)$ as the PMF of the random discrete variable x_i. It directly follows that the PMF satisfies the following properties:

$$0 \leq P_x(X_i) \leq 1 \quad \forall X_i \ , \tag{14a}$$

$$\sum_{i=1}^{N} P_x(X_i) = 1 \ . \tag{14b}$$

It is common to omit the index i and write the PMF as $P_x(X)$.

The joint PMF of two discrete random variables is then written as $P_{x,y}(X,Y)$ from which the marginal PMF is determined as

$$P_x(X) = \sum_Y P_{x,y}(X,Y) \ . \tag{15}$$

In ML applications it is common to work with sparse sample data and PMFs such as the binomial distribution function and the Poisson distribution [293].

Statistical quantities are obtained by taking moments of the PDF or PMF. The mean value or expectation of a function $g(X)$ is computed as

$$\mathbf{E}(g) = \begin{cases} \int_{-\infty}^{\infty} g(X) p_x(X) dX \ , \\ \sum_X g(X) P_x(X) \ . \end{cases} \tag{16}$$

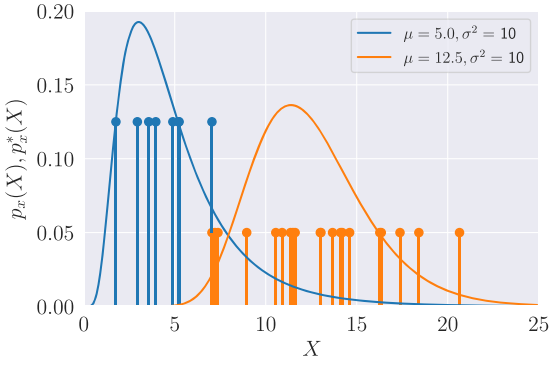


Fig. 5. Construction of a fine-grained distribution (symbols) through sampling from a lognormal continuous distribution (solid line) for two conditions, with mean μ and variance σ^2.

The variance of $g(X)$ is given as:

$$\begin{aligned}\mathbf{V}(g) &= \mathbf{E}\left([g-\mathbf{E}(g)]^2\right) \\ &= \begin{cases} \int_{-\infty}^{\infty}[g(X)-\mathbf{E}(g)]^2 p_x(X)dX\ , \\ \sum_X [g(X)-\mathbf{E}(g)]^2 P_x(X)\ . \end{cases}\end{aligned} \tag{17}$$

When working with multidimensional data that combine features with different scales, it is common to normalize these quantities by defining a standardized random variable with zero mean and unity variance:

$$x^* = \frac{x-\mathbf{E}(x)}{\sqrt{\mathbf{V}(x)}}\ . \tag{18}$$

2.2. Statistical modeling and estimation

A statistical model is represented by $(\mathscr{L}, \{P_{x|\theta}\}_{\theta\in\mathscr{P}})$, where $\mathscr{L}$ denotes the sample space from which the data are sampled, $\{P_{x|\theta}\}_{\theta\in\mathscr{P}}$ is a parameterized family of probability distributions defined on $\mathscr{L}$, and $\mathscr{P}$ denotes the set of model parameters. In statistical modeling, it is assumed that the data are generated by a process that can be well approximated by a member of this parameterized family. As an example, the output of an experiment may denote high-temperature ignition or low-temperature ignition. As this space of outcomes is binary, we define our sample space as $\mathscr{L}\in\{0,1\}$, where 0 represents low-temperature ignition and 1 indicates high-temperature ignition. Given this sample space, we can define the parameterized family of PDFs as a Bernoulli distribution:

$$P_{x|\theta}(X|\theta) = \theta^X(1-\theta)^{1-X}\ . \tag{19}$$

Once a statistical model is defined, the data are used to estimate the parameters of the distribution. Statistics that estimate the true population parameter, θ, are called estimators and are denoted as $\widehat{\theta}$. An ideal estimator should exhibit certain properties, such as consistency, low bias, or low variance [289]. While many estimators exist, the maximum likelihood estimation (MLE) is commonly employed in ML; in the next section we discussed it in more detail.

2.3. Maximum likelihood estimation

MLE is a method for estimating the parameters in a distribution by minimizing the "difference" to the true (unknown) distribution. The likelihood, L, is the joint probability distribution of the data conditioned upon the parameterized probability distribution

$$L(\theta|\mathscr{L}) = P_{x|\theta}(x_1=X_1, x_2=X_2, \ldots, x_N=X_N|\theta)\ . \tag{20}$$

If the samples are independent, then Eq. (20) simplifies to $L(\theta|\mathscr{L}) = \prod_{i=1}^{N} P_{x|\theta}(x_i=X_i|\theta)$. This independence is not necessary to apply MLE. Note that the likelihood is a function of the parameters, θ, conditioned upon the data, $\mathscr{L}$. Additionally, the likelihood is not a probability distribution; thus, it may not, for instance, integrate to unity over the range of parameters. With this formalism, the maximum likelihood estimator is defined as:

$$\widehat{\theta}_{\text{MLE}} = \underset{\theta\in\mathscr{P}}{\arg\max}\ L(\theta|\mathscr{L})\ . \tag{21}$$

Using this approach, the inference problem is converted to an optimization problem. This task can be carried out with tools from optimization, such as steepest descent. The maximum likelihood estimator exhibits desirable properties in that it is consistent, unbiased, and invariant under transformation. Additionally, under some regularity conditions, the maximum likelihood estimator is normally distributed with a mean at the true population parameter and a variance inversely proportional to the estimated Fischer information [296].

In practice, it is more convenient to maximize the logarithm of the likelihood (the log likelihood), $l(\theta|\mathscr{L}) = \ln(L(\theta|\mathscr{L}))$. Since the logarithm is a monotonic function, this maximization does not affect the estimator, but it often simplifies subsequent calculations.

2.4. Statistical learning

ML is an algorithmic approach for constructing models that predict the outcome, or response, of a system through inference from collected data [283]. In supervised learning, the most common approach, the data consist of a set of observations,

$$\begin{aligned}\mathscr{L} &= \left\{(x,y)_1, (x,y)_2, \ldots, (x,y)_N\right\}\ , \\ &= \left\{(x,y)_i\right\}_{i=1}^{N},\end{aligned} \tag{22}$$

where $x\in\mathscr{X}$ represents the input state or features and $y\in\mathscr{Y}$ is the corresponding outcome or target. Here, $\mathscr{X}$ denotes the set of input data and $\mathscr{Y}$ is the set of output data. Each pair $(x,y)_i$ is referred to as an instance or a sample. These observation samples are employed during the training of the ML model that is then applied to data that was not seen during training. The dimension of the input state can be a scalar $x\in\mathbb{R}$ (such as pressure or temperature), a vector of thermochemical quantities $x\in\mathbb{R}^M$ (such as the vector of species concentrations C), or a multidimensional tensor of different flame images $x\in\mathbb{R}^{M\times M}$. Similarly, the output can be a scalar, a vector, or a higher-dimensional tensor. ML relies on the inherent assumption of inferring an underlying relation between input and output, which can be written as:

$$F: \mathscr{X}\rightarrow\mathscr{Y}\ , \tag{23}$$

where F is the target function that is embedded in the data.

Consider a general supervised ML algorithm (Fig. 6), which we discuss in detail because supervised learning algorithms are prominent and because all of them largely adhere to this architecture. At the inception, we have a physics-based relation between input features and output target F. The exact form of this relation is typically unknown and is evident only through the set of observations $\mathscr{L}$ given by Eq. (22).

Error measures and training error The objective of an ML algorithm is to identify from a set of hypotheses, $\mathscr{H}$, a particular hypothesis, f, that expresses the unknown relation between features and targets. Different ML algorithms, which are further discussed in Section 3, correspond to different hypothesis sets that are represented by their candidate functions, f_j (Fig. 6). The performance of a candidate hypothesis in approximating the target function is quantified through an error measure, E. An iterative optimization method uses the error measure to systematically test various candidate functions from the hypothesis set to determine an optimal model.

In this context it is important to recognize the difference between ML and curve fitting. Although both methods operate on data, curve fitting is fundamentally concerned with identifying a set of model parameters such that a prescribed function provides a best fit to available data. In contrast, ML is a more general approach in that it is concerned with finding a particular function that can generalize over unobserved data that are sampled from the same distribution. As such, the key difference between curve fitting and ML lies in the approach: curve fitting focuses on the data, while ML focuses on the underlying process embedded in the data.

The functional form of the error measure depends on user-specific requirements for accuracy and quantities of interest (QoIs), as represented by the ML model. For example, for classification problems, a popular error measure is

$$E(Y,f(X)) = \frac{1}{N}\sum_{i=1}^{N} L(y_i,f(x_i))\ , \tag{24}$$

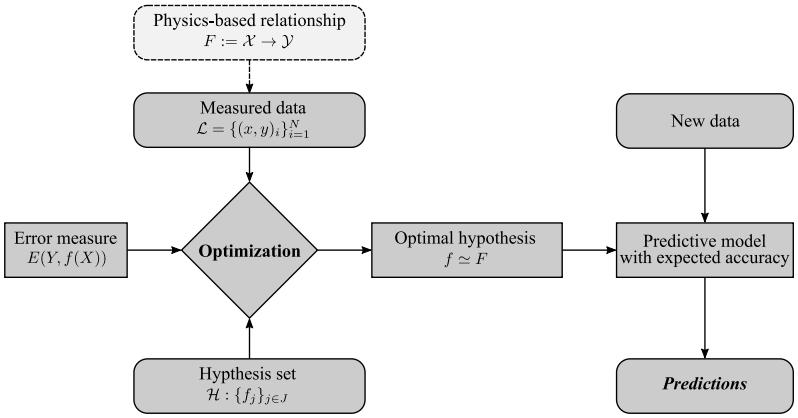


Fig. 6. Schematic of a general supervised learning algorithm.

where the zero-one loss function $L(a,b)$ is defined as

$$L(a,b) = \begin{cases} 0 & \text{if } a = b\,, \\ 1 & \text{if } a \neq b\,, \end{cases} \tag{25}$$

which measures the occurrence of misclassification averaged over the learning set for the candidate function f. Another loss function for binary classification problems is the binary cross entropy loss,

$$E(Y,f(X)) = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(f(x_i)) + (1-y_i)\log(1-f(x_i))], \tag{26}$$

which is equivalent to the (averaged) negative of the log likelihood, assuming that the target variables are sampled from a Bernoulli distribution whose success probability is modeled by f.

For regression problems, a commonly employed error measure is the mean squared error,

$$E(Y,f(X)) = \frac{1}{N}\sum_{i=1}^{N} \| \, y_i - f(x_i) \, \|^2\,, \tag{27}$$

which estimates the square of the distance between the predictions and the true target averaged over the learning set. The minimization of the mean squared error is equivalent to minimizing the negative log likelihood, assuming that the target variable is sampled from a Gaussian distribution with a constant variance.

Other error measures can be considered for certain applications, such as the mean absolute error (or the L_1-norm), which is more stable for data with significant outliers [289]. The confusion matrix [297] is another tool for assessing errors in supervised classification problems. For regression tasks, the Huber loss metric or quantile losses are commonly employed; a comprehensive discussion of these loss functions and their effects appears in Natekin and Knoll [298].

The dataset $\mathcal{L} = \{(x,y)_i\}_{i=1}^N$ consists of samples from the joint distribution $p_{x,y}(X,Y)$. While error estimates over the entire dataset may be useful for optimization, they may not accurately reflect the performance of the model with new and unseen data. This ability is referred to as generalization, and is a central objective in ML [282]. The expected prediction error of the final ML model, the generalization error, is defined as an expectation over the probability distribution of the feature and target pairs,

$$\mathscr{E}(f) = \mathbf{E}_{p(X,Y)}\{E(Y,f(X))\}\,, \tag{28}$$

and the expectation, introduced in Eq. (16), given as

$$\mathbf{E}_{p(X,Y)}\{E(Y,f(X))\} = \int\int p_{x,y}(X,Y)E(Y,f(X))dXdY\,. \tag{29}$$

While Eq. (28) quantifies the prediction error over all possible feature-target pairs, the joint PDF is typically not available due to sample variability or incomplete measurements, thereby introducing bias into the evaluation of the generalization error. Therefore, the generalization error is estimated through re-substitution of the training data by which $p_{x,y}(X,Y)$ is replaced by an empirical distribution,

$$\mathscr{E}(f) \simeq \mathbf{E}_{(X,Y)\in\mathcal{L}}\{E(Y,f(X))\}\,. \tag{30}$$

Re-substitution is commonly employed in curve fitting but results in optimistic estimates, as it relies on the same data that are used during learning. Hence, it is common practice in ML to divide the dataset into a training set and a testing set, $\mathcal{L} = \mathcal{L}_{\text{train}} \cup \mathcal{L}_{\text{test}}$. The testing set is only used for estimating the generalization error,

$$\mathscr{E}(f) \simeq \mathbf{E}_{(X,Y)\in\mathcal{L}_{\text{test}}}\{E(Y,f(X))\}\,, \tag{31}$$

providing a more reliable estimate of the model performance. The relative sizes of the training set and the testing set depend on the problem under consideration, the hypothesis set, and other factors [299]. Standard practice, based on the Pareto principle, suggests that the training set includes 80% of the learning dataset and the testing set includes the remaining 20% of the data.

Optimization The process of learning an ML model involves determining a set of model parameters. Model parameters in ML models represent coefficients in regression models, synaptic weights in neural networks, or likelihood parameters in classification models. General-solution techniques are considered for ML to accommodate various learning algorithms, error measures, and hypotheses. This accommodation is generally achieved by framing the solution of finding the optimal set of model parameters as an optimization problem, subject to minimizing the error over the training set. Gradient descent is commonly employed as an iterative method for finding the local minima of differentiable functions with respect to coefficients θ,

$$\theta^{n+1} = \theta^n - \alpha\left.\frac{\partial E}{\partial \theta}\right|_{\theta=\theta^n}\,, \tag{32}$$

where the hyperparameter α is introduced to control the learning rate. Optimization algorithms from the family of gradient descent methods, which are frequently employed and readily accessible through open software libraries, include classical batch gradient descent, stochastic gradient descent, adaptive moment estimation, and root-mean square propagation [286,300,301]. Optimizers such as the latter two incorporate adaptive learning rates; learning-rate scheduling overcomes the issue of problem-specific hyperparameter tuning. Alternatives to these methods include higher-order methods such as Newton's methods [302] and meta-heuristic methods [303] such as genetic algorithms (GAs) or simulated annealing. These methods offer advantages for optimizing in large search spaces, finding global optima, and carrying out

gradient-free optimization.

Hyperparameter optimization Hyperparameters determine the learning rate and convergence of a ML algorithm. The evaluation of these hyperparameters is typically performed in prior or iterative simulations in which models are trained with various hyperparameters and an independent dataset is used to assess the models' performance. To prevent potential overfitting as a result of iterating on the same dataset, the learning set is split into three parts: a training set, a testing set, and an additional validation set, viz., $\mathscr{L} = \mathscr{L}_{\text{train}} \cup \mathscr{L}_{\text{test}} \cup \mathscr{L}_{\text{valid}}$, typically in 80:10:10 or 60:20:20 splits. The validation set is used to find satisfactory values for the hyperparameters. Traditionally, hyperparameter optimization is carried out using randomized grid searches [304]. With advances in algorithms and computational resources, approaches like sequential model-based global optimization and Bayesian optimization have gained popularity [305]. We extend this discussion on hyperparameters through practical applications in Section 3.

2.5. Model error and bias-variance decomposition

To explore concepts underlying underfitting and overfitting, we consider the bias-variance decomposition [306]. Assuming that the joint probability distribution of the features and the target quantities, $P_{x,y}(X, Y)$, is available, the best possible model can be determined by minimizing the generalization error, independent of any dataset. This idealization is referred to as Bayes' model and is represented as $f_B(X)$. The error in this model arises due to noise and random deviations in the dataset. This error is irreducible and represents the lowest possible error that any ML model can attain on the dataset.

The generalization error for a single model $f_l(X)$, learned over a single dataset $\mathscr{L}_l \subset \mathscr{L}$, can be expressed as:

$$\mathscr{E}(f_l) = \mathbf{E}_{(X,Y)\in\mathscr{L}_l}\{E(Y,f_l(X))\}\ , \tag{33}$$

where the mean squared error, Eq. (27), is used as the error measure. However, we note that this error depends on the learning set used to formulate the model. If we take the expectation over all learning sets for the generalization error, this formulation can be decomposed as:

$$\begin{aligned}\mathbf{E}_{(X,Y)\in\mathscr{L}}\{\mathscr{E}(f)\} \quad &= \underbrace{E(Y,f_B(X))}_{\text{Noise}} + \underbrace{\left(f_B(X) - \mathbf{E}_{(X,Y)\in\mathscr{L}}\{f(X)\}\right)^2}_{\text{Bias}} \\ &+ \underbrace{\mathbf{E}_{(X,Y)\in\mathscr{L}}\left[\left(\mathbf{E}_{(X,Y)\in\mathscr{L}}\{f(X)\} - f(X)\right)^2\right]}_{\text{Variance}},\end{aligned} \tag{34}$$

showing that the expected value of the generalization error can be broken down into noise, bias, and variance. The noise term in Eq. (34) represents the limit on accuracy and is independent of the model and the learning dataset, providing a theoretical lower bound on the generalization error for any ML model. The bias term in Eq. (34) measures the difference between the average prediction of models of a selected family generated over distinct datasets and the prediction of Bayes' model. As Bayes' model is the idealized best possible model, departure from it should lead to higher errors, as seen in the decomposition of Eq. (34). The last term in Eq. (34) measures the variance of predictions of ML models learned from all possible learning datasets. Generally, high model bias is indicative of underfitting, while high variance relates to overfitting, as illustrated by the following example.

Consider measurements of the laminar flame speed for methane (CH_4)/air mixtures (Fig. 7). For the learning datasets, we randomly sample ten points from the data with white noise. Three polynomial regression models are fitted using such learning sets: up to 1st order, 3rd order, and 5th order. In each plot, the light gray curves show the predictions of the regression model learned over distinct learning datasets, randomly sampled for each curve. The average over these curves, which approximates $\mathbf{E}_{(X,Y)\in\mathscr{L}}\{E(Y,f(X))\}$, is the bold dark curve. For regression up to 1st order, there is little variance in any of the individual model predictions from the averaged prediction (Fig. 7a). Thus, the 1st order models have low variance. However, significant difference between the averaged prediction and the actual function is observable—meaning that the 1st order models have high bias. The situation where models exhibit low variance but high bias is referred to as underfitting. Underfitting of trained models may occur when the hypothesis set, $\mathscr{H}$, is not expressive enough to capture the signal in the data.

In contrast, for regression considering up to 5th order polynomial representations (Fig. 7c), the averaged prediction over all individual models is reasonably close to the true signal. Therefore, the 5th order models have low bias. However, there is significant variation between the predictions of individual 5th order models: they are very sensitive to the choice of data points in the learning set. Hence, the 5th order models have high variance. The situation where the models have high variance but low bias is referred to as overfitting. Overfitting of trained models may occur when the hypothesis set, $\mathscr{H}$, is overly expressive so that, in addition to accounting for the signal in the data, the trained models start to account for the stochastic noise in the data. Finally, for regression up to 3rd order (Fig. 7b), the averaged prediction over all individual models is close to the analytic solution and the variance between individual models is also low. These models have low bias and low variance.

Model selection The discussion about the expressive performance of a model leads to the concept of model selection, or choosing the hypothesis set, for the problem and the dataset available [322]. Based on the complexity of the functions in the hypothesis set, algorithms can be arranged hierarchically. This complexity or expressiveness can be measured using the Vapnik-Chervonenkis (VC) dimension [323] (Fig. 8). However, for realistic problems, the choice of the hypothesis set is based on the nature of the problem, the amount and type of data available, the accuracy desired, the cost of training, the storage required by the trained model, and other constraints. An overarching concern in the choice of the hypothesis set, especially in CombML, is to ensure that it is at least as complex as the problem under consideration. Currently, the robust selection of a hypothesis set and the corresponding ML algorithm is still in its infancy; selection is largely guided by experience.

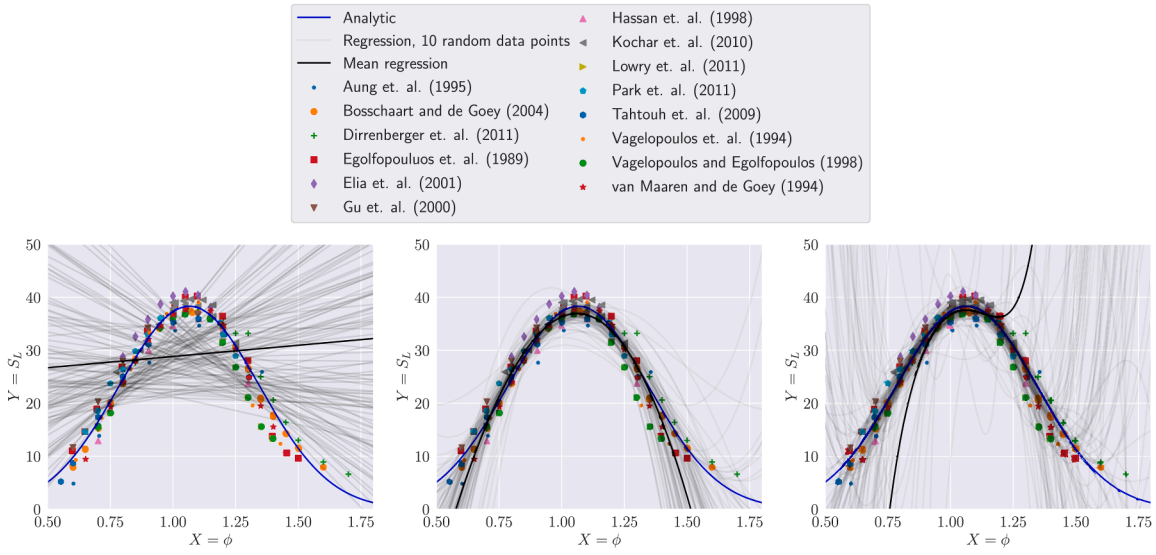
A common procedure in ML applications is to start with a certain degree of overfitting and correct from there. This strategy ensures that the hypothesis set is at least as complex as what is possible to be inferred from the data. Techniques to limit overfitting include cross-validation, early stopping, and regularization. Regularization strategies attempt to reduce overfitting by biasing the optimization procedure to select simpler candidate functions in the hypothesis set; simpler models are not able the represent the high-frequency noise that is superimposed on the signal (Fig. 7). To reduce overfitting, a penalty (or regularization) term is appended to the loss function during the training procedure. Considering the mean squared error in Eq. (27), the resulting expression can then be written as:

$$E(Y,f(X)) = \frac{1}{N}\sum_{i=1}^{N} \| y_i - f(x_i) \|^2 + \lambda g(\theta)\ , \tag{35}$$

where the hyperparameter λ is determined using the validation set and the regularization function $g(\theta)$ is evaluated from the model parameters in the hypothesis set.

2.6. Uncertainties

Two broad classes of uncertainties require consideration (Fig. 9): epistemic uncertainties and aleatoric uncertainties. Epistemic uncertainties arise due to the lack of knowledge of the dynamics of the system under consideration, or an inability to express its dynamic behavior using models. Such epistemic uncertainties are important for instance when learning from small datasets or sparse training data. This situation is commonly encountered in combustion applications due to limited experimental access, low data-acquisition rates, or intermittency of the combustion system under investigation. For such conditions, it is essential to utilize domain knowledge to structure the learning



(a) 1^{st} order regression. (b) 3^{rd} order regression. (c) 5^{th} order regression.

Fig. 7. Bias-variance decomposition: Fitting of laminar flame-speed measurements for CH_4/air mixtures at ambient conditions [307–320] with polynomial regression. Empirical correlation by Gülder [321] (blue) is given as $S_L(\phi) = W\phi^{\eta}\exp\{-\zeta(\phi-\sigma)^2\}$ with $W = 42.2$ cm/s, $\eta = 0.15$, $\zeta = 5.18$, and $\sigma = 1.075$. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

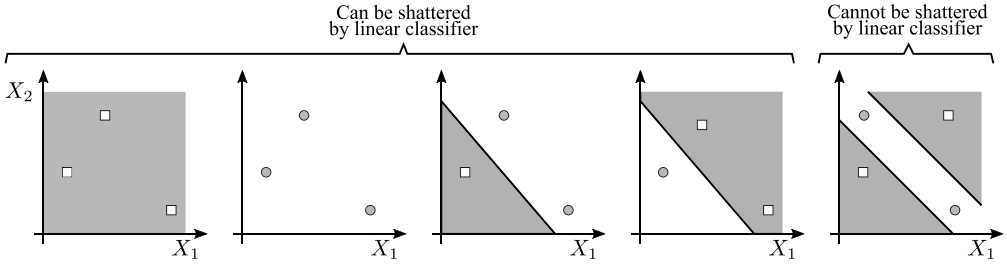


Fig. 8. Illustration of VC dimension for binary classification problem. A linear classifier can only perfectly classify, i.e. shatter, at most $d = 3$ points in a binary classification problem. Hence, the VC dimension is $d + 1 = 4$.

algorithm appropriately.

Epistemic uncertainties can be further divided into parameter uncertainties and structural uncertainties, pertaining to uncertainties associated with the model parameters and the form of the optimal model selected from the hypothesis space, respectively. In contrast, aleatoric uncertainties are often referred to as irreducible or stochastic uncertainties. In combustion applications, these uncertainties may arise due to noise in the training data, the projection of data onto a low-dimensional thermochemical state space, or the absence of important features in the data. Aleatoric uncertainties can be divided into homoscedastic uncertainties and heteroscedastic uncertainties. Homoscedastic uncertainties are uniform across all inputs in the range, while heteroscedastic uncertainties vary over the input space.

Uncertainties are traditionally demarcated into statistical and systematic uncertainties. Statistical uncertainties refer to errors that may be quantified by statistical analysis over a series of measurements. In contrast, systematic uncertainties occur due to models and theory—and cannot be treated the same as statistical uncertainties. In this regard, epistemic uncertainties are associated with systematic uncertainties and statistical uncertainties are aleatoric.

2.7. Dataset shifts

Traditional physics-based approaches (Fig. 3) that are derived from

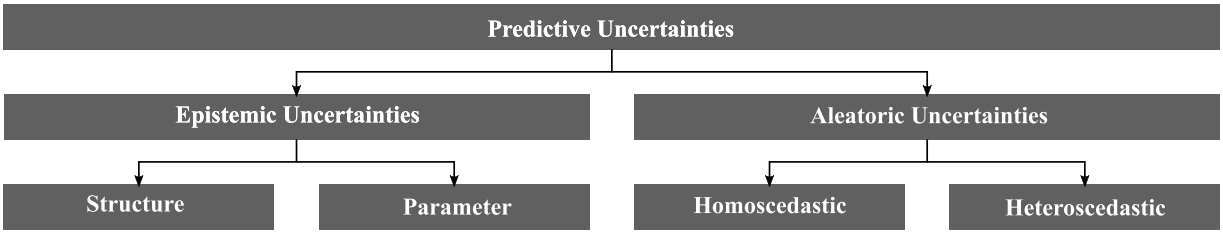


Fig. 9. Uncertainties in ML applications.

fundamental physical principles are applicable to conditions and scenarios that comply with the underlying assumption of the physical model. In contrast, the mappings learned by data-driven models are contingent upon the training data. Thus, ML models are typically only applicable in the range of their training data. An inherent assumption in ML is that the samples to be predicted should be drawn from the same joint probability distribution (defined over the features and the target) as the training dataset. Samples where this condition is met are referred to as in-sample instances; samples where this condition is violated are referred to as out-of-distribution instances. A change in the joint distributions of the training data compared to those of the prediction samples is termed dataset shift [324]. Dataset shifts can appear in different forms:

- Covariate shift: Shift in the distribution over the feature space without changing the conditional distribution of the labels
- Label shift (or prior probability shift): Shift in the distribution over the target space without changing the conditional distribution of the features
- Concept drift: Change in the statistical properties of the target quantities over time due to a changing system behavior not considered in the model
- Open set recognition: Occurrence of new classes (in a classification problem) in the target space of the prediction sample, which were absent in the training data.

Dataset shifts can lead to the deterioration of the model's predictive accuracy. This is compounded by the fact that many ML algorithms and state-of-the-art models produce overly confident predictions for out-of-distribution samples. In practical CombML application, a good ML model should attempt to be robust to dataset shift. This can be particularly challenging when dealing with time-dependent combustion problems. A common instance of dataset shift in CombML applications manifests when the ML model is trained and evaluated on simulation data, but is deployed in realistic applications. In many cases, such a dataset shift may be obligated as labeled training data may only be available from simulations.

3. ML algorithms

3.1. Overview of ML techniques

ML methods can be categorized into three types: supervised, unsupervised, and semi-supervised learning (Fig. 10). This section provides an overview of various ML methods, specifically focusing on underlying mathematical principles and discussing salient features by considering illustrative combustion examples of selected methods from each category. Applications of these methods to combustion science and engineering are reviewed in Section 4.

Section 3.2 discusses supervised learning algorithms, which are currently the most common class of ML methods with relevance to combustion. Supervised learning requires that the training data are labeled, consisting of N tuples of inputs and labels $\{(x,y)_i\}_{i=1}^{N}$. Examples of supervised learning in combustion applications include classification techniques for selecting particular chemical-kinetic mechanisms, constitutive relations, and combustion models that generalize with available data. Regression techniques extend learning methods to continuous outputs and can be applied to fitting thermodynamic response functions, rate coefficients, and subgrid closures.

In Section 3.3, we discuss unsupervised or descriptive learning. These techniques operate on unlabeled learning data and are employed in knowledge discovery, dimensional reduction, and the identification of latent variables in data. As such, these techniques can be attractive for the construction of low-dimensional combustion manifolds, reduced chemical mechanisms, and the identification of parametric dependencies in complex combustion environments. Principal component analysis (PCA; Section 3.3.2) is one example of a commonly employed unsupervised learning method that has been used for identifying low-dimensional combustion manifolds.

Supervised and unsupervised learning algorithms may be differentiated by their reliance on labeled data or the lack thereof. The intersection of these approaches results in a category of methods that learn from both labeled and unlabeled data. Such semi-supervised learning approaches are attractive for the analysis of incomplete measurements and dealing with missing data, which is commonly encountered in combustion applications in which only a few thermochemical quantities can be measured. In addition, semi-supervised techniques in RL can be applied for optimal control of combustion systems in the presence of noise and the absence of descriptive models. With relevance to describing the dynamics of a combustion system, sequence models and generative approaches are particularly attractive because they enable the consideration of data sequences and can be utilized to generate low-order models through the abstraction of high-fidelity simulations. Section 3.4 discusses technical details behind these semi-supervised learning methods.

Recognizing that purely data-driven ML models can exhibit deficiencies [325,326] in accurately capturing complex physicochemical processes in combustion applications, Section 3.5 discusses methods and

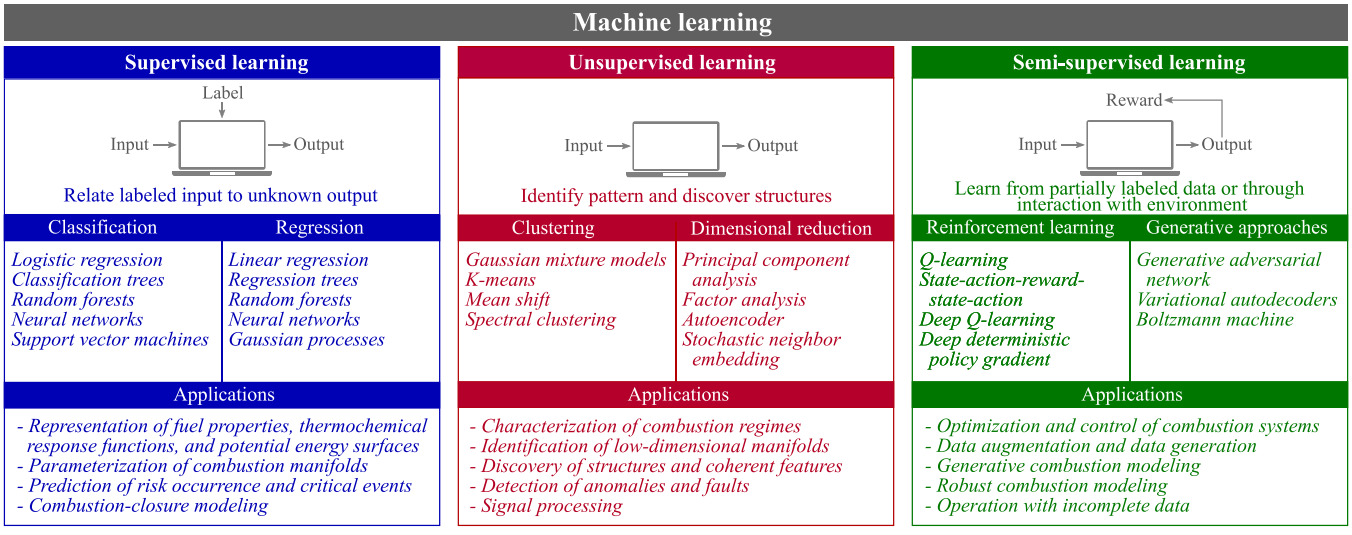


Fig. 10. Classification of ML techniques with examples and applications of corresponding ML methods.

frameworks for combining prior knowledge and physical information from combustion science and engineering with ML.

3.2. Supervised learning

Supervised learning is concerned with inferring a hypothesis in the form of an ML model from a labeled dataset that maps the input data to the output. Supervised learning provides an algorithmic framework for learning the model parameters from the data by minimizing an objective function

$$\arg\max_{\theta\in\mathscr{P}} E(Y,f(X,\theta)) \ , \tag{36}$$

where we expose the implicit dependence of the hypothesis f on the model parameters θ for clarity.

Supervised learning algorithm differ in their prespecification of the hypothesis set, model architecture, and the optimization method for selecting model parameters.

3.2.1. Logistic regression

Perhaps one of the simplest algorithms in supervised learning is logistic regression [327]. Logistic regression is commonly employed for binary classification of learning data $\mathscr{L}=\{(x,y)_i\}_{i=1}^N$ with $y_i\in\{\psi_1,\psi_2\}$ (where $\psi_1=0$ and $\psi_2=1$) corresponding to two classes that are mutually exclusive and exhaustive. Thus, logistic regression predicts the probability of the output $\widehat{y}=f(x)$ being a member of class ψ_1. By introducing the posterior probability of class ψ_1, this formulation can be written as:

$$\widehat{Y}(X)=P(Y=\psi_1|X)=\sigma\big(w^\mathsf{T}X+b\big) \tag{37}$$

or $\widehat{Y}(X)=1-P(Y=\psi_1|X)$ since both classes are mutually exclusive and exhaustive.

The prediction of logistic regression for binary classification is executed in two discrete steps (Fig. 11a). The first step, scalarization, involves the reduction of the M-dimensional feature vector $X\in\mathbb{R}^M$ to a scalar quantity $Z=w^\mathsf{T}X+b$ with weights $w\in\mathbb{R}^M$ and bias $b\in\mathbb{R}$. In the second step, Z is passed through an activation function, which introduces nonlinearities to the algorithm through a sigmoid function:

$$\sigma(Z)=\frac{1}{1+\exp\{-Z\}} \quad \text{with} \quad Z=w^\mathsf{T}X+b \ . \tag{38}$$

The model parameters $\theta=(w^\mathsf{T},b)^\mathsf{T}$ are determined by minimizing an error function; logistic regression considers the likelihood function $L(y|\theta)=\prod_{i=1}^N\widehat{y}_i^{y_i}(1-\widehat{y}_i)^{1-y_i}$ (Section 2.3). Taking the negative logarithm gives the cross-entropy error function, and dividing by the number of samples N results in the cross-entropy error (Eq. (26)).

The nonlinearity in the sigmoidal function requires an iterative method to determine the model parameters. The convex form of the error function ensures a unique solution and gradient descent (Eq. (32)) or an iteratively reweighted least squares method can be employed [328]. The Jacobian, $\partial_\theta E$, can be determined in analytic form by applying the chain rule:

$$\frac{\partial E}{\partial\theta}=\frac{\partial E}{\partial\sigma}\frac{\partial\sigma}{\partial Z}\frac{\partial Z}{\partial\theta} \ , \tag{39}$$

with the derivative of the sigmoidal function given as $\partial_Z\sigma(Z)=\sigma(Z)(1-\sigma(Z))$.

Although logistic regression is commonly employed for binary classification, the extension to multiclass classification problems is obtained by writing the posterior distribution in Eq. (37) as a soft-max function for K distinct classes (Fig. 11b):

$$P(Y=\psi_k|X)=\widehat{Y}_k(X)=s(Z) \ , \tag{40}$$

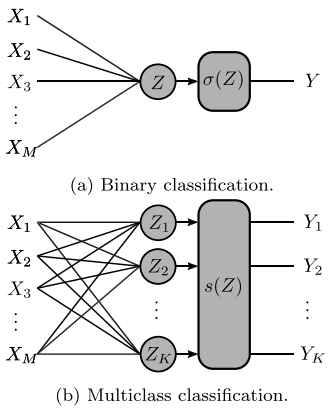


Fig. 11. Schematic of a logistic regression algorithm with an M-dimensional feature vector for (a) binary classification and (b) multiclass classification with K classes.

where ψ_k is a binary vector of zeros except for element k and $s(Z)$ is the soft-max function,

$$s(Z_k)=\frac{\exp\{Z_k\}}{\sum_{j=1}^K\exp\{Z_j\}} \ . \tag{41}$$

The cross-entropy error function is then written as:

$$E(\theta_1,\theta_2,\ldots,\theta_K)=-\frac{1}{N}\sum_{i=1}^N\sum_{k=1}^K y_{ik}\log(\widehat{y}_{ik}) \ , \tag{42}$$

with $\widehat{y}_{ik}=\widehat{y}_k(x_i)$. In Section 3.2.6 we examine this multiclass logistic regression for demarcating distinct ignition regimes. Note that the number of model parameters scales linearly with the dimensionality of the feature space and the number of classes, making this algorithm computationally tractable for high-dimensional input spaces.

In Section 2.4, we indicated that ML algorithms are concerned with identifying a target hypothesis from a hypothesis set. This is not the case for logistic regression, where the hypothesis is predetermined by the transfer function and the ML model is fully determined by specification of the model parameters. In Section 3.2.4, we see that binary logistic regression can be considered as a fundamental building block of a neuron in a neural network in which the input signal is passed through a transfer function. Section 4.3.1 discusses applications and extensions of logistic regression models for predicting fire occurrence and for risk assessment.

3.2.2. Decision trees

Logistic regression relies on the assumption of linear separability of the feature space, which limits application to problems with complex decision boundaries. Tree-based methods [329,330], such as decision trees (also referred to as classification and regression trees), overcome this issue and can represent arbitrarily complex relationships by recursively partitioning the feature space into hypercuboids [283]. In addition, this intuitive algorithm provides a high degree of interpretability.

A decision tree consists of a set of rules that defines a partition over the feature space. Consider a multiclass classification problem with K distinct classes, $Y\in\{\psi_1,\psi_2,\ldots,\psi_K\}$. Given a set of features X that are sampled from a space Ω, the decision tree generates a model, $f(X)=\psi_k$, that partitions the domain Ω into non-empty subregion $\Omega^f_{\psi_k}$ such that $\Omega=\cup_{k=1}^K\Omega^f_{\psi_k}$ with $\Omega^f_{\psi_l}\cap\Omega^f_{\psi_m}=\varnothing$ for $l\neq m$. Here, the superscript f denotes that these partitions were generated by the function f.

Fundamentally, a tree is a graph consisting of vertices and directed edges, $G=(V,E)$. An edge E connects the parent vertex V_k to its child vertex V_l. The first vertex is referred to as the root; it has no parent vertex. All other vertices are connected by edges. Terminal vertices are

called leaves.

In a binary classification problem (Fig. 12), a decision tree seeks to classify each two-dimensional data sample $x_i = (x_1, x_2)_i$ into two distinct categories, $Y \in \{\psi_1, \psi_2\}$. The leaf nodes are shaded to indicate the classification of the point. Initially, the tree consists of a singleton leaf and all data samples are assigned to class ψ_1 to minimize the error (Fig. 12a). In the first branch (Fig. 12b), the tree partitions the X_1–X_2 space into two subsets that correspond to $\Omega^f_{\psi_1} : X_1 < \xi_1$ and $\Omega^f_{\psi_2} : X_1 \geq \xi_2$. In the second step (Fig. 12c), another branch is created to refine the partition. To classify a new sample via such a tree-based model, the sample is traversed down the tree and is allocated to the correct partition at every internal vertex until it terminates at a leaf that is assigned to the class defined by the vertex.

Let us consider how tree-based models learn from data. Following the formulation of Breiman et al. [330], we introduce an impurity measure, $i(V)$, to assess the quality of possible branches that are represented by a vertex V. For each vertex, a branch is identified that maximizes the reduction in the impurity. The impurity decreases due to a binary split that divides a vertex V into left and right children, V_L and V_R, is given by

$$\Delta i(s, V) = i(V) - \frac{N_{V_L}}{N_V} i(V_L) - \frac{N_{V_R}}{N_V} i(V_R), \tag{43}$$

where N_V is the number of samples (from the training set) assigned originally to node V, and N_{V_L} and N_{V_R} are the number of samples assigned to vertex V_L and V_R, respectively. For classification problems, a popular impurity measure is the impurity function based on the Gini index [331]:

$$i_G(V) = \sum_{k=1}^{K} P(\psi_k|V)(1 - P(\psi_k|V)), \tag{44}$$

where K is the number of classes and $P(\psi_k|V)$ denotes the probability of a point in the sample being assigned to class ψ_k conditioned upon the split represented by node ψ, explicitly N_{ψ_k}/N_V. At every step of growing a decision tree from data, different splits over different features at a given terminal node are evaluated. The split leading to the maximum decrease in the impurity measure is chosen to partition the (formerly terminal) node into additional child nodes. This process continues until no further decrease in the impurity measure is possible.

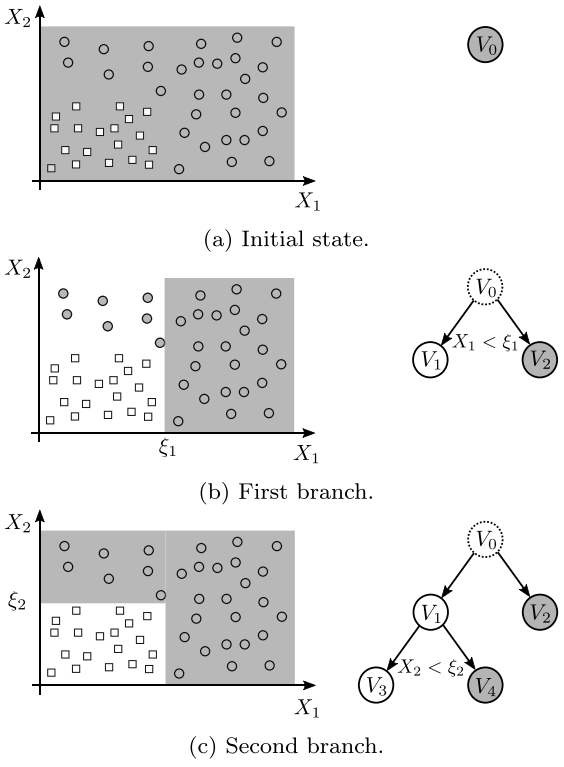


Fig. 12. Stepwise construction of a decision tree for a two-dimensional feature space with binary classification.

The process of growing a tree from data is a greedy algorithm [332]: it makes the locally optimal choice at each step. This approach makes decision-tree models prone to overfitting if allowed to grow without restriction. Various approaches prevent overfitting by regularization, including stopping splitting when a terminal node has only $N_{\min}$ samples form the learning set and constraining the tree to be of a certain maximum depth, $d_{\max}$. These scenarios require learning of the hyperparameters $N_{\min}$ or $d_{\max}$, which is commonly done using a validation dataset or cross-validation (Section 2.4).

Decision trees offer ease-of-use and accurate predictions, especially with simple tabular data. This learning algorithm also provides the additional benefit of model interpretability (Section 5.2). At the basic level, this can be embodied via feature importance scores, provided by the mean decrease impurity (MDI) measure for all the features in the input set [333]. Here, the importance of a feature is given by aggregating the weighted decrease in variance for all the nodes where the specific feature is used as the criterion for partitioning the feature space. Such measures of model interpretability provide insight into the underlying rationale learned by the model during training and can lead to high confidence in the model. Similarly, such interpretability measures can lead to data-driven discovery of new relationships between input features and targets, making this ML method suitable for combustion problems that can benefit from fundamental insights such as in model discovery and feature selection.

3.2.3. Random forests

Section 3.2.2 showed that decision trees are prone to overfitting, which is reflected by low bias but high variance. Therefore, it is desirable to consider an approach that retains the low bias of decision-tree models but reduces their variance. Random forests are an ensemble method that accomplishes this goal. By combining a collection of decision trees into a random forest, an ensemble model is created that has lower variance than the individual constituents while maintaining low bias. This is reflected in the concept of the "strength of weak learnability" [334], which considers an ensemble of weak models that are (largely) independent and deliver far superior predictions [335]. The variance of the ensemble model is directly proportional to the correlation between individual models in the ensemble. Thus, the more uncorrelated our individual models are, the lower the variance of the ensemble model. To inject this decorrelation between individual decision trees into the random forest, two concepts are commonly utilized:

- Bagging [336]: Bagging (or bootstrap aggregating) is an approach to create different ML models from the same dataset. The first step generates multiple new training subsets by sampling from the original dataset, uniformly and with replacement. Each of these sampled datasets can be used to train an ML model. The final prediction is chosen by aggregating the predictions of these individual models. In random forests, each individual tree is exposed to such a bootstrap sample of the original training dataset from which to learn, ensuring that every tree learns from a different dataset and imparts a level of decorrelation to the trees.
- Randomly subsampling over the features [337]: In random forests, for splitting at each node, the trees must determine the best split over random subsamples of the features. This approach introduces additional decorrelation between the trees in the ensemble.

Random forests possess the benefits of decision trees in accuracy, ease-of-use, and interpretability of decision trees, and are not as prone to

overfitting. This makes this ML method suitable for data that do not require special representation. Similar to other ensemble approaches, the computational cost scales with the number of decision trees that constitute the ensemble. Often, very complex problems can require more than thousands of trees [336] for good prediction, which can pose challenges for real-time predictions and limited hardware (such as in combustion-control applications).

3.2.4. Neural networks

In Section 3.2.1, we noted that logistic regression has inherent inabilities to replicate nonlinear decision boundaries, limiting its utility for complex physics-based classification tasks. For such undertakings, we must rely on more expressive algorithms, such as neural networks.

The most straightforward neural network architecture, often referred to as a multilayer perceptron (MLP) or fully connected, feedforward neural network, consists of an arrangement of individual logistic regression units, termed neurons, in a network of hierarchical layers (Fig. 13). The output $\widehat{Y}$ of neuron l is computed as

$$\widehat{Y} = \sigma(Z) \quad \text{with} \quad Z = \sum_{i=1}^{M} w_{l,i} X_i + b_l \,, \tag{45}$$

where σ is the transfer or activation function of neuron l (Fig. 13 b). For neural networks, the sigmoid function, tanh function, and ReLU (Rectified Linear Unit) functions are the preferred activation functions for introducing nonlinearities. Sigmoid functions are also used in logistic regression (Section 3.2.1) and can be convenient for training via backpropagation due to the differentiability of the activation function (Eq. (38)) and its derivative. However, because of the slow convergence during training when using the sigmoid function, ReLU functions (which are clipped linear functions) are sometimes preferred. In a fully connected neural network, the outputs from the preceding layer act as composite features for every neuron in the succeeding layer. As such, a fully connected neural network defines a mapping from the space of the input layer to that of the output layer.

Deep fully connected networks have been applied to problems with a variety of inputs and outputs, such as scalars, images, and sequences (Section 4.1.3). However, in many such applications, fully connected neural networks encounter limitations. Therefore, specialized neural networks that ameliorate these limitations have been developed for specific applications. For problems involving high-dimensional tensorial inputs and outputs (spatial/image data), CNNs are more appropriate. CNNs consist of convolutional blocks, pooling layers, and a fully connected network (Fig. 14a). A convolutional block typically combines a convolution layer, an activation function, and batch normalization. This architecture allows the fully connected layers to process image data without the need for vectorizing the image, thereby preserving non-local information. This can result in higher prediction accuracy and easier training when dealing with multi-dimensional simulations and flame imaging data.

In computer vision and digital signal processing, a filter is a function that operates on a local neighborhood of a pixel to generate a result [338]. Filtering an image involves application of such a filter over the entire image. Such filters can be applied to images (or multidimensional tensors) to denoise and resize images, or to extract features like texture and edges from the image. Convolutional layers consists of multiple correlation filters. The simplest filters replace the corresponding pixel in the output by the average or maximum value of the pixel's neighborhood in the input. These replacements correspond to the pooling operations used in CNNs (Fig. 15a), where a max-pooling operation is employing using a 2×2 filter, with the filter moving across two rows and columns (strides). Depending on the problem, filter sizes and strides can be chosen via hyperparameter tuning or by the following intuition: kernels which are too large will result in large information losses, while kernels which are too small will result in low sharing of information with neighboring pixels. Using large strides has the same effect as

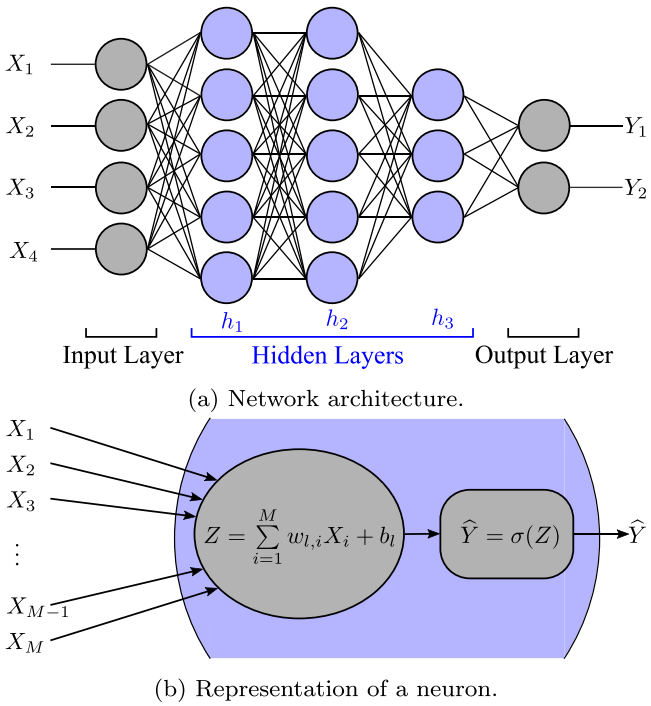


Fig. 13. Schematic of a fully connected, feedforward neural network. (a) Network architecture of a multilayer perceptron, consisting of an input layer with four input channels $X_{i=1,\ldots,4}$, two output channels Y_1 and Y_2, and three hidden layers with 5, 5, and 3 neurons, respectively. (b) Representation of an operation on a neuron l as a logistic regression with a weighted summation over input state and application of transfer function σ.

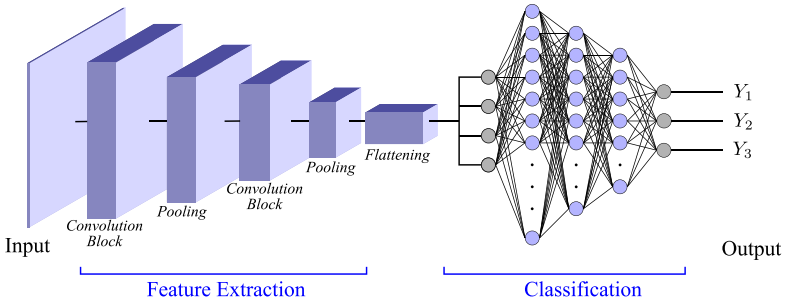
downsampling the spatial data. A more complicated filtering operation occurs when the filtered result is a weighted combination of pixels over a small neighborhood of pixels:

$$g(i,j) = \sum_{k=0}^{K} \sum_{l=0}^{L} f(i+k, j+l) h(k,l) \,. \tag{46}$$

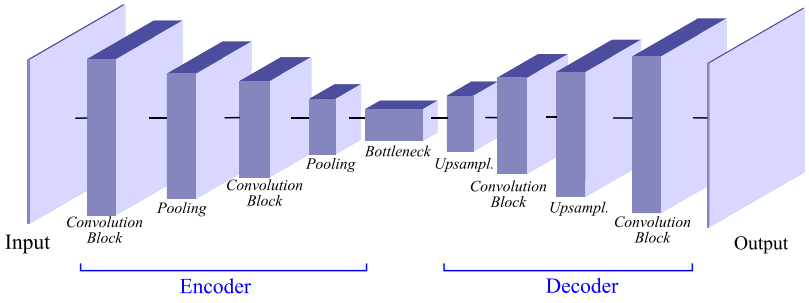
The entries of the weight kernel or mask $h(k,l)$ are referred to as the filter coefficients and K and L are the widths of the filter. This application is the correlation operator, given by $g = f^{*}h$. Such linear correlation (or cross-correlation) filtering is referred to as convolution in reference to neural networks (Fig. 15b).

These deep learning layers can be arranged together to form versatile architectures. One example of this involves arranging convolutional layers into an autoencoder network [339], which is an un- or semi-supervised learning approach, depending on its utilization. An autoencoder is a deep neural network that is broken up into two main sections: the encoder and the decoder (Fig. 14b). The encoder reduces the data field into a set of parameters that describe the variance seen in the input. The last layer in the encoder is often called the bottleneck, which outputs a compressed form of the original input data known as latent variables, and is treated as input to the decoder. The decoder takes the features in the bottleneck and regenerates the input or a segmentation. Autoencoders are closely related to another dimensionality reduction technique, PCA (Section 3.3.2), and can even be viewed as a nonlinear generalization of principal component analysis PCA [340]. This kind of architecture can find many practical uses in reducing the dimensions of combustion chemistry (Section 4.1.3).

A loss of non-local information can also occur when using fully connected networks with sequential data (e.g., temporal data describing flame dynamics, transient ignition events, or engine cycle operation). Additionally, in many sequence modeling problems, the length of the input and output are not fixed, which can lead to problems when using

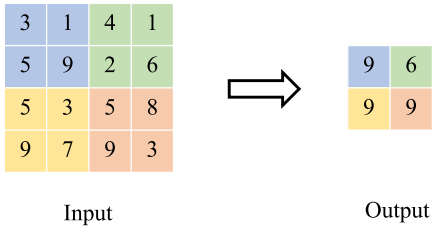


(a) CNN classification.

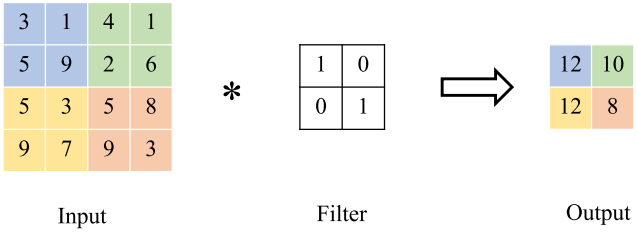


(b) CNN autoencoder.

Fig. 14. Schematic of commonly employed CNN architectures. (a) CNN classifier combining feature extraction and fully connected neural network for classification. (b) CNN autoencoder with encoder and decoder. The convolutional block combines a convolution layer, an activation function, and batch normalization.



(a) Max-pooling with a 2×2 filter and a stride of 2.



(b) Convolution with a 2×2 filter and a stride of 2.

Fig. 15. Operations in a CNN: (a) pooling and (b) convolution.

fully connected networks. Similar to image data, sequences can be very long, leading to large computational cost. For instance, a single-layer network with 1,000 neurons that accepts a time history with 10,000 measurements as input leads to over a million weights and biases. RNNs [341] represent a network architecture that overcomes these challenges by employing a hidden state to maintain the relationship between past and future inputs (Fig. 16). To reduce the number of weights and biases, the same operation with the same weights is applied to each element in the input sequence, which gives the network its recurrent moniker:

$$A_t = f_a(W_a A_{t-1} + W_x X_t + b_a) \,, \tag{47a}$$

$$Y_t = f_y\left(W_y A_t + b_y\right) \,, \tag{47b}$$

with activation A_t, output Y_t, function $f_{\{a,y\}}$, weights $W_{\{a,x,y\}}$, biases $b_{\{a,y\}}$, and the subscript t denoting the timestep. Numerous types of RNN exist, such as one-to-one, many-to-one, and many-to-many. A traditional fully connected network can be viewed as a one-to-one RNN with one sequence and one output, while a many-to-many RNN has many sequences and many outputs (Fig. 16). A popular RNN architecture is the long short-term memory (LSTM) architecture, which incorporates logic gates to regulate information within the network [342].

3.2.5. Support vector machines (SVMs)

An SVM [343,344] is a non-probabilistic algorithm that forms decision boundaries within linearly separable data. In a binary classification problem, these decision boundaries are formed by solving for the maximum distance between points closest to the decision boundary (Fig. 17). These points are called support vectors.

SVMs employ hyperplanes for classifying and regressing data. Consider a binary classification problem for N data points of feature x_{ij} defined in an M-dimensional space. In this classification problem, the decision boundary is an optimal hyperplane for separating data points into two classes $y_i \in \{\psi_1, \psi_2\}$ (where $\psi_1 = -1$ and $\psi_2 = 1$). The hyperplane dividing the two classes—the decision boundary—can be expressed as:

$$\sum_{i=1}^{M} w_i X_i + b \; = 0 \,, \tag{48}$$

where the weights $w \in \mathbb{R}^M$ form a vector normal to the hyperplane and b is the bias coefficient. The hyperplanes coinciding with the support vectors on both sides of the decision boundary, known as margins, can be expressed as:

$$\sum_{i=1}^{M} w_i X_i + b \; = Y_{\text{supp}} = \pm 1 \,, \tag{49}$$

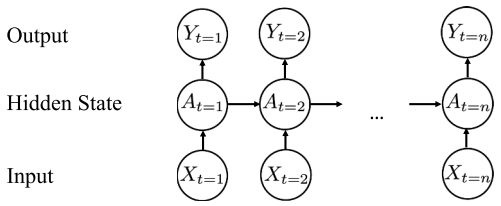


Fig. 16. Schematic of a many-to-many RNN.

where the distance between the two margins can be expressed as $2/\| w \|$. For a hard-margin problem, where data points are not allowed to fall within the margin, training the SVM thus involves solving the following optimization problem using the method of Lagrange multipliers:

$$\arg\min_{w_j,b_i} \frac{1}{2}\sum_{j=1}^{M} w_j^2 - \sum_{i=1}^{N}\sum_{j=1}^{M} \lambda_i \left[y_i \left(w_j x_{ij} + b_i \right) - 1 \right] , \tag{50}$$

where λ is the Lagrange multiplier. The first term of this loss function minimizes the distance between the two margins, while the second enforces a constraint such that no point falls within the margin. Note that the second term in the loss function can be modified to allow for a soft-margin SVM, where some points are allowed to fall within the margins.

As Eq. (48) suggests, this form of SVM can only be applied to linear classification problems. However, SVMs can be extended to nonlinear problems by transforming the data points to a higher-dimensional space, rendering the data linearly separable. This transformation is usually accomplished with the use of inner products. However, since the computational cost can become intractable with large datasets, the inner product is approximated using a kernel function in practical algorithms, a process known as kernelizing. The radial basis function kernel, constructed based on the Euclidean distance of vectors, is a popular kernel for nonlinear data.

This section has focused on describing the employment of SVMs for classification problems. However, this method can be directly extended to regression problems by considering samples of a continuous target variable, instead of the binary classes discussed here. SVMs offer ease-of-use and low computational complexity (Section 3.2.6), and are thus suited in computationally-restricted applications such as in real-time prediction of combustion phenomena.

3.2.6. Application examples

Let us examine and contrast the supervised learning algorithms discussed in the previous sections in the context of a combustion application. For this, we consider predicting the ignition behavior of an *n*-dodecane/air mixture. The learning data are generated by performing zero-dimensional homogeneous reactor simulations with Cantera [345] using a compact skeletal mechanism [346] for various initial temperatures and *n*-dodecane ($C_{12}H_{26}$) mole fractions at a constant pressure of 20 bar. The simulations are advanced until $t = 1$ s, which corresponds to the maximum timescale for alkane low-temperature chemistry under typical engine conditions [347]. Isocontours of adiabatic flame temperature and CH_2O mass fraction, a marker for low temperature combustion, as a function of $C_{12}H_{26}$ mole fraction and temperature are shown in Fig. 18.

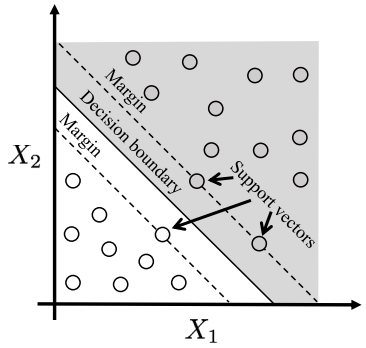


Fig. 17. Hard margin SVM applied to a two-dimensional binary classification problem.

A specified number of randomly sampled points within the raw data is used to generate the training set for a multiclass classification problem. Following the definitions of low- and high-temperature chemistry by Ju et al. [347], data points where the adiabatic temperature exceeds 1050 K are labeled as high-temperature chemistry and conditions in which reactions occur below 1050 K and with CH_2O exceeding 1% of the maximum CH_2O level are labeled as low-temperature chemistry; all other points are labeled as no ignition. Various learning algorithms are trained to classify these three conditions: regions dominated by (i) high-temperature chemistry, (ii) low-temperature chemistry, and (iii) no ignition.

The contours in Fig. 19a,f depict the generated classes within the learning data, while the corresponding scatter points represent sampled points used for training the ML algorithms. In the present problem, we illustrate the behavior of the learning algorithms trained with 2% and 60% of the generated data. While the accuracy of the classifiers increases with the training data, the accuracy of the ML algorithms tends to plateau when sufficient data is provided during training (Fig. 20a). Low model complexity of logistic regression results in relatively low test accuracy of 0.89. With sufficient training data (Fig. 19f), all the nonlinear classifiers (Fig. 19h,i,j) predict the ignitability of *n*-dodecane mixtures accurately ($\sim$ 96% test accuracy), with feedforward neural networks providing the highest classification accuracy. However, other ML algorithms such as the random forest can outperform neural networks for small datasets (less than 5% of the learning data), as shown in Fig. 20a.

Under certain scenarios, the accuracy of the classifiers can decrease with increasing data size, as seen with the SVM (Fig. 20a). This phenomenon can be attributed to the SVM's sensitivity of the hyperparameters to the size of the datasets; the SVM's hyperparameters were optimized for $\sim$ 20% of the learning dataset. When applying logistic regression multiclass classifiers (Fig. 19b,h), the probability $\widehat{Y} = P_{y|x}(Y = \psi_1|X)$ of a sample belonging to a particular class ψ_1 is still evaluated in a binary fashion. However, if $\widehat{Y} = P_{y|x}(Y = \psi_2|X)$ or $\widehat{Y} = P_{y|x}(Y = \psi_3|X)$ exceeds $\widehat{Y} = P_{y|x}(Y = \psi_1|X)$, then the sample point would be assigned to ψ_2 or ψ_3, respectively. Hence, only three linear decision boundaries will be formed during classification.

The case with 2% learning data demonstrates supervised learning in representative combustion problems in which simulations or experiments can be costly, yielding small datasets. Since the classifier accuracy is highly dependent on the size of the training data (Fig. 20a), all nonlinear classifiers (Fig. 19c,d,e) demonstrate some flaws when trained with 2% learning data. For example, the sharp boundaries predicted by the random forest (Fig. 19d), in contrast to the smooth boundaries produced by SVMs and neural networks, are artefacts of recursively partitioning the feature space during training. In many applications, decision trees and random forests can face issues when extrapolating outside of the training set due to this recursive partitioning; partitions for labels outside the training set are not well defined, which can result in clipped predictions. In order to improve the accuracy of ML methods with the sparse and small datasets encountered in many scientific problems, knowledge-guided ML approaches (Section 3.5), which have largely been applied to neural networks, have been proposed to

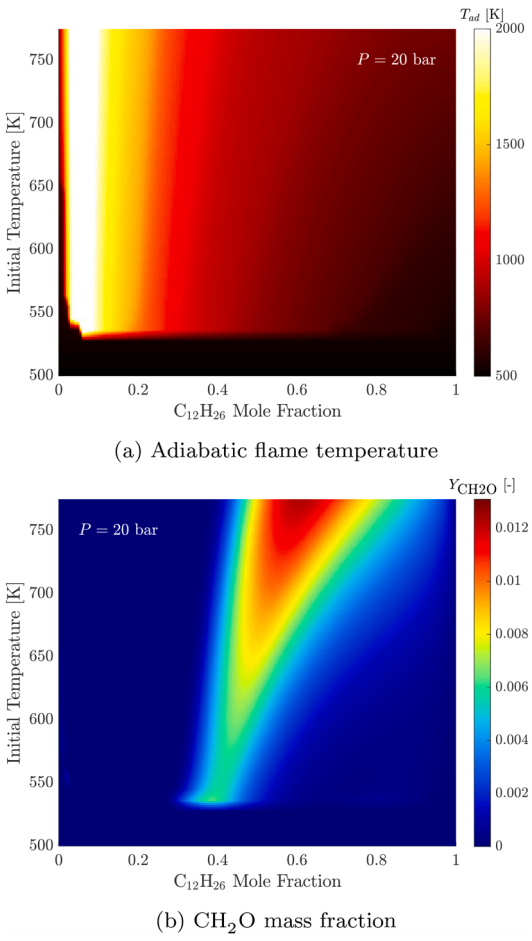


Fig. 18. Isocontours of (a) adiabatic flame temperature and (b) CH_2O mass fraction of a zero-dimensional homogeneous reactor with $C_{12}H_{26}$ at 20 bar.

circumvent this issue by embedding domain knowledge into ML architectures.

In addition to classifier accuracy, another aspect to consider when selecting the appropriate ML algorithm involves the computational cost (Fig. 20b). Logistic regression requires the least computational costs during training and prediction, due to the algorithm's simplicity. Due to the algorithmic complexity of gradient descent methods, training neural networks is typically costlier than other supervised learning algorithms. This cost can be ameliorated with the use of specialized ML hardware (graphics/tensor processing unit). However, the largest computational costs from using neural networks typically arise from the cost of high-dimensional hyperparameter search.

While a common practice in supervised learning involves splitting learning datasets into training and testing sets with a ratio of 80:20 (Section 2.4), an alternative practice involves splitting learning sets into a training, validation, and testing set. Here, we split the learning data into a 60:20:20 split, respectively. This additional validation set is used for searching ML hyperparameters, which can have a significant effect on the model predictions as shown in Fig. 20c by the difference in validation accuracy that arises from varying the two hyperparameters within SVM models using a radial basis function kernel. Within this ML model, C is associated with the cost factor of the soft margin, while γ is related to the variance of the data. Hyperparameter searches can be done through an exhaustive grid search, which traverses through the entire space of possible hyperparameters. This is only feasible in certain algorithms such as SVMs, decision trees, and random forests due to the small number of hyperparameters. The main hyperparameter within decision trees involves selecting a stopping criteria for the depth of the tree during training, while random forests possess an additional hyperparameter that determines the number of decision trees. Hyperparameter search for neural networks involves a large number of dimensions (such as learning rate, hidden layer size, and training batch size) which can grow even larger for complex deep learning architectures. In this situation, an exhaustive grid search is computationally not feasible, and low cost strategies—such as a random search (which samples the hyperparameter space randomly) or Bayesian search [348] (which samples the hyperparameter space with quantified uncertainties)—are typically used. These hyperparameter and architecture optimization strategies can require deep expertise and have spawned an entire field of research known as automated ML [349].

3.2.7. Algorithm selection and inductive biases

With the maturation of ML methods and their widespread availability, we can now choose supervised learning algorithms that best fit specific problems. However, at present the selection of certain algorithms has remained pragmatic as it is often determined by a combination of the data format and the volume of the data (Section 3.2.6). With respect to the volume of training data available, different rules of thumb have been offered. For instance, it has been recommended that for training a CNN for image classification, about a thousand representative images are required for each class [350]. For scalar data, the VC dimension of the model can be used to estimate the sample-complexity bounds or the relationship between data volume and generalization error, as previously discussed in Section 2.5. With respect to data format, a commonly-held belief may be that for tabular/scalar input data, one can utilize random forests, boosted trees, or densely connected neural networks. Convolutional architectures are recommended for image data formats, while RNNs are commonly used for sequence data. While these are useful guiding principles as shown in Section 4, they are not universal. For instance, one can use data augmentation to train complex multi-class classifiers with a few hundred training samples. In many cases, transfer learning can be used to train accurate models with just tens of training samples. Introducing domain knowledge in the learning procedure, for instance by using hard or soft constraints, can enable models to be trained with even fewer samples still.

To guide the formulation and training of data-driven models for combustion applications, we outline the concept of inductive bias [351,352]. Inductive bias refers to the set of assumptions that each model uses to generalize beyond the training data. With relevance to combustion applications, this can be illustrated by tasks involving the creation of a data-driven model for a chemical kinetic mechanism or the construction of a closure model for the subgrid scalar dissipation rate. For this, we can consider two diametrically different approaches. In the first approach, we may explicitly define the mathematical structure and the learning algorithm needs to determine the model-describing coefficients as a function of the input features. This principle is encapsulated by data-centric approaches and empirical correlation models (Fig. 7). The second approach may be to use a fully connected neural network for this modeling task. Here, the structure of the model form is not defined and the ML algorithm is free to approximate the model response as an arbitrary function of the input features. The key difference in these two approaches is the degree of inductive bias. The first approach introduces a high degree of inductive bias in the learning process. This reduces the amount of computational resources and data required to train the model. However, it also reduces the degree of freedom of the data-driven model (with respect to the space of functions it can explore) and may lead to underfitting. The second approach has a significantly lower degree of inductive bias. The model can explore over a larger hypothesis set to arrive at the optimal hypothesis. However, the model would require much more data and computational resources for training and may be prone to overfitting.

The aforementioned approaches are representations of two different ML paradigms: ML with features that are hand-engineered with domain knowledge and an end-to-end ML design philosophy. While end-to-end

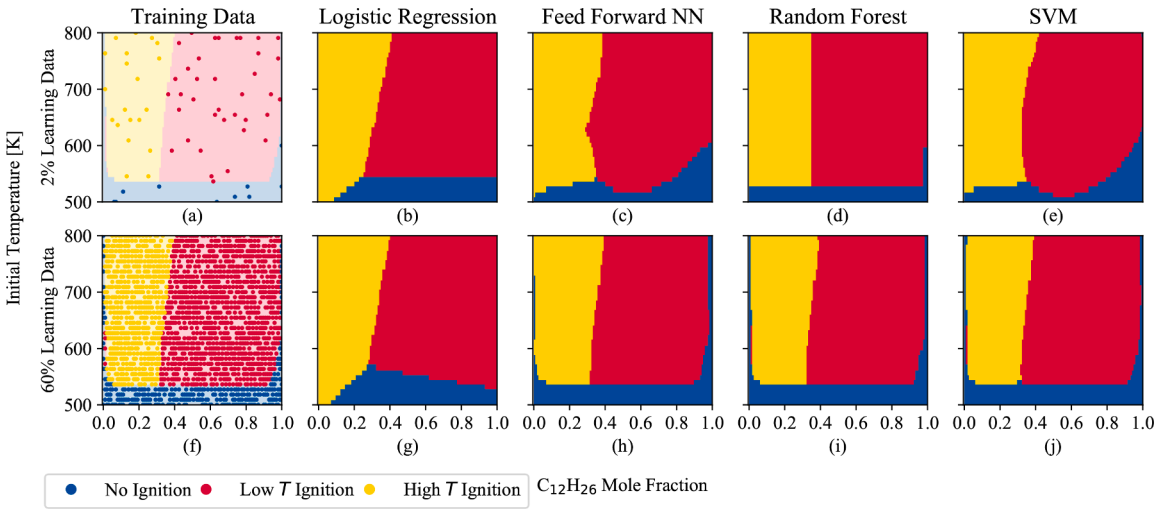


Fig. 19. Comparison of various supervised learning algorithms for classifying the ignition behavior of an *n*-docecane/air mixture using a training set with 2% (top) and 60% samples (bottom). NN, neural network.

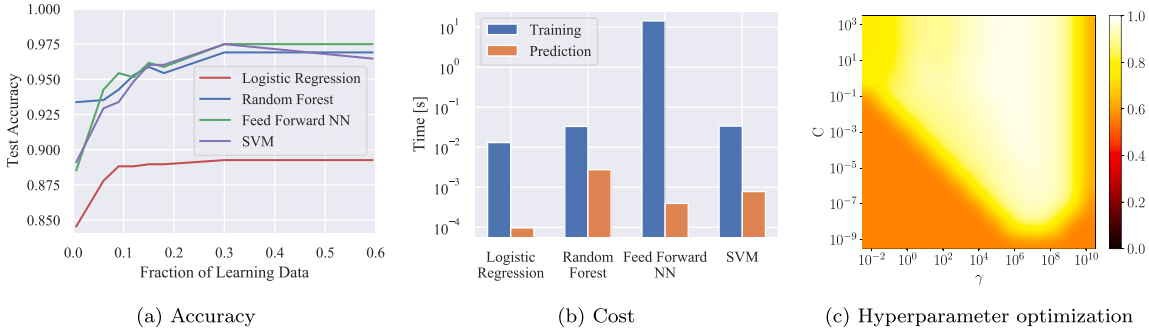


Fig. 20. Different factors for choosing an ML algorithm: (a) prediction accuracy, (b) training and prediction cost, and (c) hyperparameter optimization for SVM model. NN, neural network.

deep-learning-based approaches have been successful in various data-analytic fields [353,354], they have significant limitations for tasks involving learning from small amounts of data, reasoning about structured data, and/or generalizing beyond the training conditions [355]. As further discussed in Section 3.5, in combustion science and engineering we have substantial domain knowledge that can be used to guide the choice of learning approaches.

At this juncture, we discuss the inductive biases of specific models and strategies, while illustrating their use with examples from combustion applications:

Decision trees and random forests For decision tree based models, the partitions are axis-aligned hyper-rectangles. Thus, decision tree models have a bias towards axis-aligned decision surfaces. This can be observed in Fig. 19d, where decision boundaries are composed entirely of vertical and horizontal lines due to sparse data availability around the decision boundaries. Ensembling approaches such as bootstrap aggregation used in random forests, are useful in reducing this inductive bias.

SVMs Since SVMs are tuned based on a maximum-margin criterion, they introduce a preference bias in the training data. The points defining the boundary, that is the support vectors, are given substantial importance in the formulation of the classifier. Thus, due to this bias, SVM classifiers are sensitive to any outliers near the margin and insensitive to data density beyond the margins. This can be observed in Fig. 19e, where sparse data lead to inaccurate predictions of the decision boundary.

Fully connected neural networks As we had illustrated earlier, fully connected layers have very weak inductive biases. Herein, all the neurons are connected to the units in the preceding and succeeding layers.

CNNs Convolutional layers are equivariant to spatial translations. When coupled with pooling layers, they are approximately translation invariant. Thus, convolutional based feature extraction is not affected by the absolute position of the feature in the feature map. Bereft of this inductive bias, a model would need training examples with the features located at different positions in the feature map to approximate this invariance. In some cases, this translation invariance may hinder model performance, specifically wherein the absolute positions of the features are important for the solution. Furthermore, convolutional layers introduce a relational bias of locality. Thus, the key features for a filter are in close proximity, determined by the size of the filter. As the receptive field of a filter is smaller than the entire input, there is an assumption that there is a strong correlation between adjacent input pixels. It should be emphasized that CNNs by themselves are not invariant to transformations like rotations or reflections, and variants have been developed to adhere to such invariances [356,357].

RNNs Recurrent layers introduce a temporal invariance in the set of solutions. Thus, the outcome of a sequence of events is the same, unchanged by any time translation of the sequence of events. Additionally, RNNs also introduce a preferential bias for locality due to their Markovian assumptions.

Regularization approaches The inductive bias of popular regularization approaches may be viewed as a preference bias towards the simplest solution. However, different forms of regularization lead to varying biases. L_1-regularization introduces an inductive bias towards sparser

solutions that utilize fewer features. L_2-regularization introduces a preference bias towards solutions whose parameters have small magnitudes. Convex combination of L_1- and L_2-regularization, such as in elastic nets [358], treat the features as groups carrying similar information, and preferentially select some groups of features over others. Dropout based regularization introduces a bias reducing co-adaptation by controlling the Rademacher complexity [359].

3.3. Unsupervised learning

Unsupervised learning involves the use of algorithms for problems with unlabeled data, where a true solution or ground truth is not easily distinguishable. These algorithms are typically used in clustering and dimensional-reduction problems. Unsupervised learning algorithms have a long history across numerous fields; the most popular algorithms, PCA and *k*-means, have been developed since the early and mid twentieth century [360,361], respectively. After providing an overview of key concepts of *k*-means and PCA (Section 3.3.1 and 3.3.2), we examine both methods in application to a combustion dataset in Section 3.3.3. The interested reader is referred to Ghahramani [362] for further discussions of unsupervised learning. The book by Celebri and Aydin [363] describes state-of-the-art algorithms for unsupervised learning.

Clustering algorithms assign sets of similar data points to different groups without any prior knowledge. These algorithms can be categorized into two broad classes [364,365]: (i) hierarchical clustering and (ii) partitional clustering. Hierarchical clustering seeks to form clusters by iteratively building a hierarchy of data. Clustering is performed either through divisive clustering (where the data start as one cluster and are subdivided into new clusters that move down the hierarchy of data) or agglomerative clustering (where every point starts as a cluster and is merged into a new cluster that moves up the hierarchy). Popular algorithms in this category include single-link [366] and complete-link [367] algorithms. In contrast, partitional clustering methods split data into clusters without forming any hierarchies, usually by solving an objective function. *K*-means is the most well-known algorithm in this category (Section 3.3.1). An extensive discussion of the subcategories within hierarchical and partitional clustering is provided by Jain et al. [364,365].

Dimensional-reduction techniques can solve problems where raw data are especially noisy, irrelevant, and/or difficult to store. Hence, these techniques have found widespread use in numerous fields for preprocessing, compressing, and visualizing data. In ML applications, these techniques are typically used for feature extraction and feature selection. Feature extraction involves preprocessing high-dimensional raw data into useful low-dimensional features. Examples of feature extraction in combustion include the construction of a reaction-progress variable from a set of chemical species, the identification of combustion regimes, and the construction of low-dimensional combustion manifolds. A classic algorithm used in feature extraction is PCA (Section 3.3.2). Reviews of feature extraction are provided by Ding et al. [368] and Khalid et al. [369].

Feature selection is typically used to improve the performance of ML models by discarding irrelevant feature subsets in large and noisy raw data [370]. Feature-selection techniques can be split into three broad categories: (i) wrapper methods, (ii) filter methods, and (iii) embedded methods. Wrapper methods divide the feature space into subsets and search for the optimal combination of features by directly testing the performance of the ML model. Filter methods, which are low cost, simply evaluate features based on a predefined criterion. Popular criteria include minimum redundancy—maximum relevance [371] and Relief [372]. Embedded methods involve feature selection methods that are employed *in situ* with learning algorithms, for example the regularization in neural networks. Feature selection is surveyed in Li et al. [370].

3.3.1. K-means clustering

K-means clustering subdivides a dataset of N points into K clusters, where K is a user-defined parameter. This clustering process is typically carried out by solving an optimization problem based on a cost function J derived from the Euclidean distance between any data point x_i and the center point/centroid μ_k of a cluster k:

$$J = \sum_{i=1}^{N} \sum_{k=1}^{K} w_{ik} \| x_i - \mu_k \|^2 . \tag{51}$$

Finding the global minimum for this optimization problem is computationally difficult. However, heuristics that target local minima are in widespread use. The most popular of these heuristics is Lloyd's algorithm [373], which involves randomly initializing a centroid and then iteratively solving Eq. (51) until local convergence by assigning data points to the closest cluster by setting $w_{ik} = 1$ for the minimum Euclidean distance $\sum_{i=1}^{N}\sum_{k=1}^{K} \| x_i - \mu_k \|^2$ and setting $w_{ik} = 0$ otherwise. Then, the positions of the centroids μ_k are updated by minimizing J from evaluating its gradient with respect to w_{ik}.

While significant advances have been made in developing various *k*-means clustering algorithms, it remains a subject of active research due to its popularity and computational difficulty, with a focus on improving *k*-means heuristics [374,375].

3.3.2. PCA

PCA is one of the most widely applied methods for dimensional reduction [376–378]. PCA is a mathematical approach for revealing preferential directions in multidimensional datasets through the identification of correlations in state space. The outcome of PCA is the identification of a transformed coordinate system along the direction of maximum data variation, enabling the elimination of less important dimensions while retaining the primary data structure. PCA algorithms typically involve three steps [268]. In the first step, a matrix consisting of the dataset $x \in \mathbb{R}^{N\times M}$ is transformed into a symmetric covariance matrix $S \in \mathbb{R}^{M\times M}$:

$$S = \frac{1}{N-1} x^{\mathrm{T}} x . \tag{52}$$

Second, the covariance matrix is decomposed to a matrix containing the eigenvalues $\Lambda \in \mathbb{R}^{M\times M}$ and a matrix containing the principal components (or eigenvectors) $Q \in \mathbb{R}^{M\times M}$:

$$S = Q\Lambda Q^{\mathrm{T}} , \tag{53}$$

Last, the principal component score matrix $P \in \mathbb{R}^{N\times M}$ is obtained by multiplying the original dataset with the eigenvector matrix:

$$P = xQ . \tag{54}$$

With this, a low-dimensional approximation to the data is obtained by only considering the first m eigenvectors (with $m < M$):

$$x \simeq x_m = P_m Q_m^{\mathrm{T}} , \tag{55}$$

where x_m approximates x by only considering the first m eigenvectors of S, $P_m \in \mathbb{R}^{N\times m}$ is the truncated score matrix, and $Q_m \in \mathbb{R}^{M\times m}$ is the truncated matrix of principal components [268].

Since its conception [360], PCA has been one of the most widely used methods in numerous fields, include combustion (as further discussed in Section 4.1.3), and various adaptations of the original method are still being constructed to target specific problems. For example, Yi et al. [379] improved on the robustness of the PCA algorithm through modifications to the construction of the covariance matrix. Lu et al. [380] developed a robust PCA algorithm that can be applied to tensors with rank larger than two. The review by Jolliffe and Cadima [378] describes various PCA adaptations.

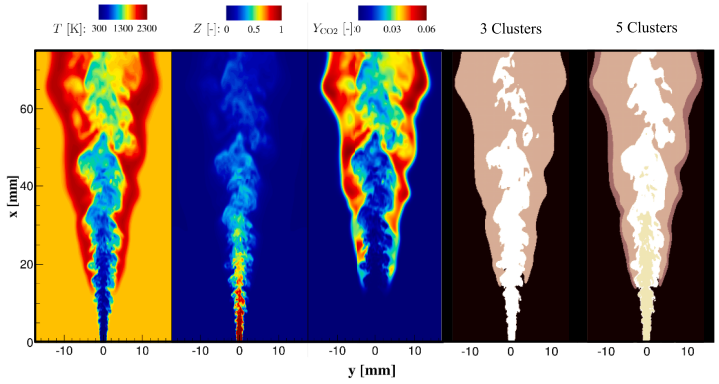
Within combustion modeling, PCA has been a popular ML method

due to the linearity of the principal components, which ensures interpretability and ease of use for further analysis. Another reason for the popularity of PCA is its maneuverable nature: errors from dimensional reduction can be controlled via careful selection of the number of principal components to retain. Further applications in combustion, particularly for the identification of low-dimensional combustion manifolds and combustion modeling are discussed in Section 4.1.3.

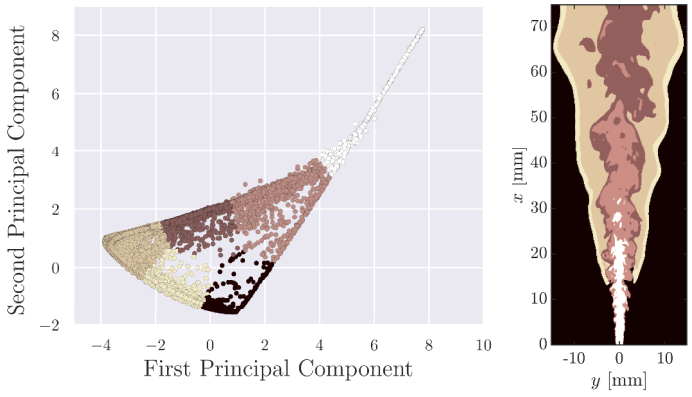
3.3.3. Application examples

In this section, we illustrate the application of unsupervised learning algorithms to simulation data of a jet-in-hot-coflow experiment [382]. In this configuration, CH_4 is supplied through an inlet of diameter $D = 1.5$ mm. The central fuel jet is pulsed and reaches a steady-state velocity of 178 ms^{-1}. A hot coflow is provided from a lean H_2/air mixture ($\phi = 0.465$), which is supplied from a 75×75 mm^2 square section that surrounds the central jet. *K*-means and PCA are applied to a single snapshot from a large-eddy simulation (LES) calculation [381] (Fig. 21a). The algorithm was used to form three and five clusters from a seven-dimensional feature space consisting of temperature, mixture fraction, and five major species mass fractions, $\mathscr{F} = \{T, Z, Y_{CO_2}, Y_{H_2O}, Y_{CO}, Y_{CH_4}, Y_{O_2}\}$. With three clusters, *k*-means distinguishes among regions of reactants, products, and oxidizer (Fig. 21a). Using five clusters allows the *k*-means algorithm to distinguish the reaction zone and to subdivide the reactant region into a low-temperature fuel region and a intermediate-temperature region (Fig. 21a). Here, two principal components were reduced from the seven-dimensional LES dataset $\mathscr{F}$ (Fig. 21b), and *k*-means was applied on the principal components. The right panel in Fig. 21b shows the clustered dataset mapped to physical space. Six clusters are required for the algorithm to identify the reaction front—compared to five clusters when applying *k*-means to the raw dataset (Fig. 21a). Nonetheless, the results of the two clustering approaches are similar, demonstrating that the fidelity of the dataset is sufficiently maintained after PCA.

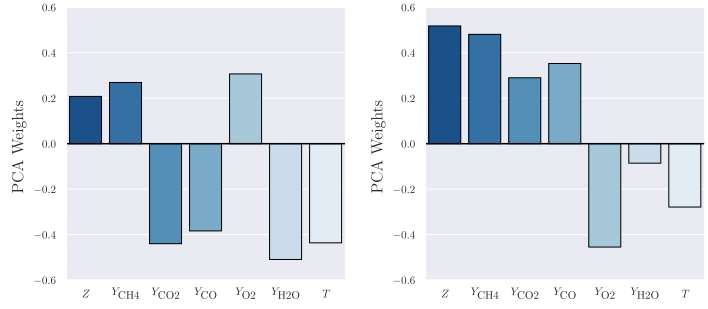
The first principal component is heavily weighted in the negative direction with combustion products and temperature (Fig. 21c). The intermediate and complete combustion products correspond with



(a) Instantaneous scalar fields and *k*-means clustering with $K = \{3, 5\}$.



(b) Application of *k*-means (with $K = 6$ clusters) to two-dimensional principal component space (left) and transformation to physical space (right).



(c) Weights of first (left) and second (right) principal components from (b).

Fig. 21. Application of unsupervised learning to LES of a jet-in-hot-coflow [381].

clusters where the first principal component is negative (Fig. 21b). The second principal component is heavily weighted in the positive direction with mixture fraction and CH_4, and in the negative direction with O_2 and temperature (Fig. 21b). The low-temperature fuel region corresponds with clusters where the second principal component exceeds a value of 2 (Fig. 21b).

Overall, this example illustrates the utility of combining unsupervised learning techniques in order to analyze complex combustion processes and to enable the reduction of the thermochemical state space, thus isolating physical processes.

3.4. Semi-supervised learning

This section introduces key ideas of semi-supervised learning strategies. These methods enable the utilization of unlabeled and labeled data. Because of the paucity of labeled data, which is commonly encountered in combustion applications, these methods are particularly attractive for analyzing incomplete and missing data. Given the promise of generative methods to overcome issues in supervised learning, here we discuss generative approaches in CombML (Section 3.4.1). In addition, due to the utility of RL techniques in intelligent control, we will discuss this method in detail (Section 3.4.2) and demonstrate its utility in an example application aimed at flame stabilization (Section 3.4.3).

3.4.1. Generative approaches

Most algorithms that we have discussed so far fall in the category of discriminative algorithms [383]. These algorithms formulate a mapping from the space of features to the space of targets, $F: \mathscr{X} \rightarrow \mathscr{Y}$ (Eq. (23)). Once a discriminative model is trained, it takes the features characterizing a new sample as an input and outputs the predicted target for the sample. In certain cases, the discriminative algorithm may model the probability of the target classes conditioned upon the input features, $p_{y|x}(Y|X)$. In contrast, generative algorithms define a mapping from the space of target classes to the space of features, $F: \mathscr{Y} \rightarrow \mathscr{X}$ [383]. In essence, the generative algorithm models the probability distribution over the space of functions conditioned upon a specific class, $p_{x|y}(X|Y)$. Common examples of generative algorithms include naïve Bayes classifiers and Gaussian mixture models [383]. Two generative approaches have gained popularity in the scientific community in the recent past: variational autoencoders [384] and GANs [385].

Owing to their wide-ranging applications, here we further discuss GANs. While GANs were originally proposed as semi-supervised learning [385], this architecture has also been applied to unsupervised learning problems in image generation [386]. In addition to classical GANs covered here, other variants are being developed and applied through modification of the objective function, such as in the Wasserstein GAN [387], or through changing complex deep learning architectures, such as in the super-resolution [388], which have been applied in CombML as will be discussed Section 4. More information on GAN variants and applications outside of CombML are provided in recent surveys [389,390].

A GAN generates synthetic data that can mimic real data provided during training. This is done in the classical GAN via an architecture consisting of two separate models [391]: a generator and a critic (also called the discriminator). The generator model outputs synthetic data from input features extracted randomly from a prior probability distribution, while the critic model determines whether the generated sample resembles the real training data through classification.

A typical training loop is shown in Fig. 22. In every step of training, the generator model takes inputs Z that have been randomly extracted from a latent space $p_z(Z)$ to generate synthetic data $\widehat{G}(Z)$. Typically, this latent space is represented by a Gaussian distribution [247]. At the same time, these generated synthetic instances are evaluated by the critic model, which outputs a value $\widehat{C}(\widehat{G}(Z))$, which can be interpreted as the probability that $\widehat{G}(Z)$ belongs to the training data. The generator model aims to produce synthetic data that mimic real data and thus, aims to increase the error of the critic model. Hence, this training procedure can be expressed as minimizing $\log[1-\widehat{C}(\widehat{G}(Z))]$.

The critic model learns to differentiate between real training data X and generated synthetic data $\widehat{G}(Z)$. In a classification problem, the critic model thus seeks to maximize the probability that X belongs to training data and minimize the probability that $\widehat{G}(Z)$ belongs to the training data. This training procedure can be expressed as maximizing $\log[\widehat{C}(X)]$ and $\log[1-\widehat{C}(\widehat{G}(Z))]$. Thus the objectives of both generator and critic models can be thus expressed as a minimization-maximization function:

$$\min_{\widehat{G}}\max_{\widehat{C}} E = \min_{\widehat{G}}\max_{\widehat{C}}\left(\mathbf{E}_x\{\log[\widehat{C}(X)]\} + \mathbf{E}_z\{\log[1-\widehat{C}(\widehat{G}(Z))]\}\right), \tag{56a}$$

$$= \min_{\widehat{G}}\max_{\widehat{C}}\left(\int_{-\infty}^{\infty} p_x(X)\log[\widehat{C}(X)]dX + \int_{-\infty}^{\infty} p_z(Z)\log[1-\widehat{C}(\widehat{G}(Z))]dZ\right), \tag{56b}$$

$$= \min_{\widehat{G}}\max_{\widehat{C}}\int_{-\infty}^{\infty}\left(p_x(X)\log[\widehat{C}(X)] + p_g(X)\log[1-\widehat{C}(\widehat{G}(X))]\right)dX, \tag{56c}$$

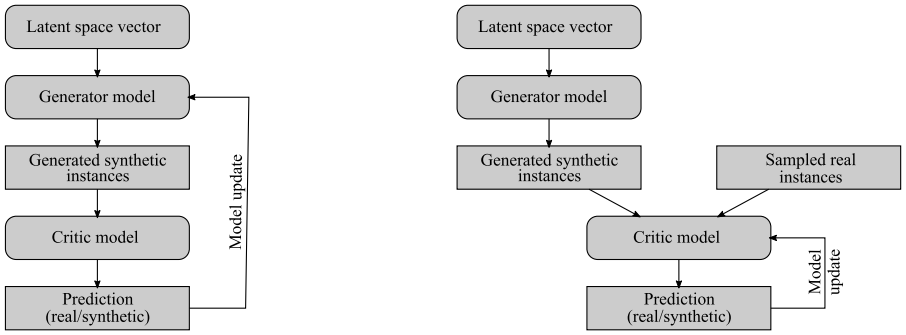
which is minimized/maximized via gradient descent/ascent during training. Note that the training data X has a probability distribution $p_x(X)$, and that the generated synthetic data $\widehat{G}(Z)$ is obtained from features Z which were extracted from $p_z(Z)$, which results in a distribution of synthetic data in the form of $p_g(X)$. As shown by the minimization-maximization optimization, this methodology recasts training as a two-player zero-sum game in which the generator and the critic compete with each other—hence "adversarial". This minimization-maximization problem can be seen as a saddlepoint optimization, which can lead to unstable training and difficulties in convergence. Fortunately, modern guidelines [386] for stable architectures, which embrace batch-normalization layers and Leaky ReLU activation while avoiding pooling layers, can be applied when designing a GAN for CombML applications, as will be further discussed in Section 4.1.4.

For an optimal critic model and fixed generator model, the classical GAN optimization problem is reexpressed as [385]:

$$\min_{\widehat{G}} E = \min_{\widehat{G}}\left[\mathscr{D}_{KL}\left(p_x\|\frac{p_x+p_g}{2}\right) + \mathscr{D}_{KL}\left(p_g\|\frac{p_x+p_g}{2}\right) - 2\log(2)\right], \tag{57a}$$

$$= 2\min_{\widehat{G}}\mathscr{D}_{JS}(p_x \parallel p_g) - 2\log(2), \tag{57b}$$

where $\mathscr{D}_{KL}(p \parallel q)$ and $\mathscr{D}_{JS}(p \parallel q)$ is the Kullback-Leibler (KL) divergence and Jensen-Shannon (JS) divergence, respectively, which are measures for the statistical distance between two probability distributions p and q. Other popular choices of statistical distances within turbulent reacting flows include the Mahalanobis [392] and Wasserstein [243] distances. In fact, some GAN architectures, such as the Wasserstein GAN [387], utilize the Wasserstein distance in their objective functions in place of the KL/JS divergences. Important properties of the JS divergence is that it is always non-negative $\mathscr{D}_{JS}(p \parallel q) \geq 0$, and only zero $\mathscr{D}_{JS}(p \parallel q) = 0$ when $p = q$. Thus, Eq. (57) is important for demonstrating that when a global optimum of the objective function is achieved, the distribution of synthetic data from the generator model becomes identical to the distribution of the training data $p_g = p_x$. Thus, the GAN optimization problem can be seen as increasing the resemblance of p_g with p_x through minimizing their statistical distances.



(a) Generator model. (b) Critic model.

Fig. 22. Schematic outlining a single training iteration for (a) generator model and (b) critic model in a GAN.

3.4.2. Reinforcement learning

RL optimizes interactions between an agent and its environment over time [393]. Consider the control of a gas turbine as a RL problem in which the gas-turbine actuator, such as fuel-flow rate or air-mass flow rate, constitutes the agent; the state of the environments (combustor exit temperature or emissions) is provided by sensors (Fig. 23). At the initial timestep $n = 0$, the sensors feed information about state s_0 to the gas-turbine actuators. The actuators then change the conditions within the gas turbine, yielding a new state s_1 and a corresponding reward function r_1 that provide feedback arising from the actuators' action for future actions. This iterative sequence of states, actions, and rewards is collected into a trajectory $\tau = ([s,a]_0, [r,s,a]_1, \ldots, [r,s,a]_n)$ that can be represented in episodic or continuous form. For simplicity, we restrict this discussion to the stepwise form. The instantaneous reward r from each increasing step accumulates into a long-term return function $R = \sum_{n=0}^{\infty}\gamma^n r_n$, with a discount factor $\gamma \in [0,1]$ that decreases the instantaneous reward from the far future. Formally, RL problems involve finding the optimal policy π^* that maximizes the expected return for all states:

$$\pi^* = \arg\max_{\pi} \mathbf{E}(R|\pi) \,, \tag{58}$$

where the policy π maps the actuators' actions and the system states. If actuators are following policy π at timestep n, then $\pi(a|\xi)$ is the probability that the action taken is $a_n = a$ if the state is $s_n = \xi$.

Most RL algorithms are based on the value function method, which finds the two essential functions for measuring long-term return:

$$V^{\pi}(s_n) = \mathbf{E}(R|s_n,\pi), \tag{59a}$$

$$Q^{\pi}(s_n,a_n) = \mathbf{E}(R|s_n,a_n,\pi). \tag{59b}$$

The state-value function $V^{\pi}(s_n)$ can be thought of as the expected long-term return starting from a state s_n and following a policy π. If the dynamics that relates state s_n to a future state s_{n+1} is unknown, then the quality (also known as action value) function $Q^{\pi}(s_n,a_n)$ becomes a more useful measure. $Q^{\pi}(s_n,a_n)$ measures expected long-term return starting from a state s_n, taking an action a_n, and thereafter following a policy π.

In value-function methods, RL shifts toward maximizing the value function in order to find the optimal policy:

$$V^* = \arg\max_{\pi} V^{\pi}(s_n) \,, \tag{60a}$$

$$Q^* = \arg\max_{\pi} Q^{\pi}(s_n,a_n) \,. \tag{60b}$$

This optimization problem can be solved by estimating V^{π} through repeated sampling of generated trajectories using Monte Carlo methods, enabling the agent to learn from experience. Another method involves exploiting the Markov property (where a future state relies solely on the current state and action) of RL problems to express Q^{π} as a recursive Bellman equation [394]:

$$Q^{\pi}(s_n,a_n) = \mathbf{E}_{n+1}[r_{n+1} + \gamma Q^{\pi}(s_{n+1},a_{n+1})] \,, \tag{61}$$

which can be solved as an optimal-control problem using dynamic programming [395].

Q-learning [396] is a popular algorithm that combines concepts from Monte Carlo and dynamic programming methods. It relies on iteratively updating values of Q for a state-action pair with:

$$Q_{n+1}(s_n,a_n) \leftarrow Q_n(s_n,a_n) + \beta\left[r_{n+1}(s_n,a_n) + \gamma \max_a Q_n(s_{n+1},a) - Q_n(s_n,a_n)\right], \tag{62}$$

where β is the learning rate. Q is initialized randomly for the initial episode. An episode is a sequence of timesteps that terminates when a condition is met. For each episode, the procedure involved in Q-learning is:

1. Initialize the state.
2. Choose actions for the corresponding state either (i) from policies derived from Q or (ii) randomly.
3. Observe the reward and new state.
4. Update Q using Eq. (62) for a single step s_n and action taken a_n.
5. Proceed to the next timestep and repeat steps 2 to 4.

Note that two possible actions can be taken in step 2 in order to allow the agent to (i) explore through a random action or (ii) exploit the information learned through Q. The choice of the action is controlled by a user-defined parameter ϵ that dictates the ratio of exploration to exploitation steps—an ϵ-greedy policy. The subsequent episode utilizes Q from the previous episode. In Section 3.4.3, Q-learning is applied to an example involving a well-stirred reactor.

When solving these problems, the state-action-reward space is typically expressed in a tabular manner. Hence, traditional RL problems can suffer from the curse of dimensionality: the computational cost rises exponentially with an increasing number of variables. Supervised-learning methods such as neural networks (Section 3.2.4) can be used to approximate functions in order to ameliorate computational bottlenecks and to handle multidimensional input data (enabling the consideration of realistic systems). Moreover, recent developments in deep learning have enabled deep neural networks to process high-dimensional raw input data directly, without any form of feature engineering. These advances have resulted in a new branch of work known as deep RL; popular algorithms include deep Q networks, deep deterministic policy gradients, and related algorithms [397].

A recent review of the application of RL to controlling industrial processes identified several benefits and drawbacks of RL control versus traditional control methods [398]; these benefits and drawbacks can be extended to combustion control. Combustion control is typically concerned with optimizing multiple objectives relating to performance and emissions while managing complex combustion phenomena, such as

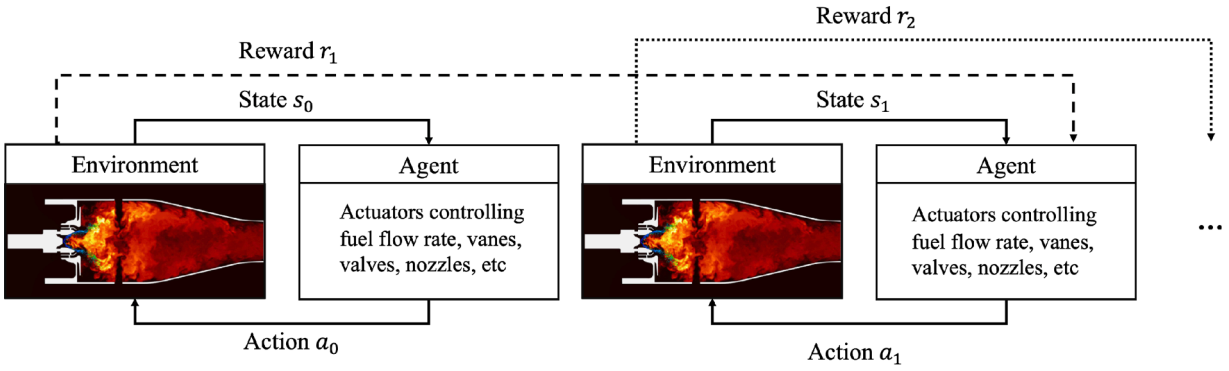


Fig. 23. Iterative loop describing the control of a gas turbine as an RL problem.

blowoff and instabilities. In this context, RL control has enormous potential because these methods are model-free (they can deal with many transient and complex combustion phenomena) and general (control can directly be manipulated by changing the reward function).

Despite the enormous potential, there are many reasons [398,399] why RL has so far been limited to academic studies. Until the initial demonstration of deep RL [354], these methods were largely restricted to solving fully observable problems with discrete state-activation spaces, excluding the application of these methods to many control problems. For example, controlling a gas turbine requires the use of sensors, which only provides partial observability because information about the entire flowfield remains unknown. In addition, the actions are performed by actuators that determine the continuous values of fuel and oxidizer flow-rates, yielding a continuous action space. While both of these issues have been resolved by advancements in deep RL, several problems still prohibit technical adoption: (i) data inefficiency, as upward of $\mathscr{O}(10^5)$ training samples are needed, (ii) out-of-sample performance, as RL behavior outside training conditions is unpredictable, and (iii) interpretability, as safety measures are difficult to integrate with black-box methods (Section 5). These challenges are currently the subject of active research by the ML community, and must also be considered when tackling combustion problems.

3.4.3. Application examples

In order to explore Q-learning, we consider the control of a quasi-steady well-stirred reactor governed by the following equation [400]:

$$T = T_{\text{in}}\left\{1 + \frac{HV\, Y_{F,\text{in}}\text{Da}\, \exp\left(-\frac{E_a}{RT_{\text{in}}}\right)}{c_p T_{\text{in}}\left[1 + \text{Da}\, \exp\left(-\frac{E_a}{RT_{\text{in}}}\right)\right]}\right\}, \tag{63}$$

with temperature T, fuel heating value HV, fuel mass fraction Y_F, Damköhler number Da, activation energy E_a, and gas constant R. The subscript 'in' refers to inlet quantities. In this example, an inlet fuel temperature $T_{\text{in}} = 300$ K is prescribed. CH_4 is chosen as fuel, corresponding to $HV = 55$ MJ/kg and $c_p = 2.26$ kJ/(kg K), with an inlet fuel mass fraction of $Y_{F,\text{in}} = 0.1$. An activation energy $E_a = 20$ MJ/mol describes the CH_4 reaction. A solution of this equation can be expressed as an S-curve (Fig. 24a). The top branch represents a stable flame regime, the middle branch represents an unstable flame regime, and the bottom branch represents inert mixtures.

In this problem, we are interested in applying Q-learning to ensure that the reactor operates within a specific temperature range in the unstable regime (Fig. 24, horizontal dashed lines). An episode terminates when the control fails (the minimum and maximum temperature of the S-curve are reached) or the total number of timesteps, $n = 600$, has elapsed. The variable learning rate decreases logarithmically with an increasing number of episodes. The initial learning exploration rate is set to 1.0, which corresponds to completely random actions.

The control procedure follows the steps in Section 3.4.2. First, Q is initialized as 0 for all state-action pairs. The reactor is initialized with a temperature on the unstable branch near the inert branch (Fig. 24b). An action is taken by increasing or decreasing the inlet mass flow rate, which in turn changes Da. We observe the new state (temperature) resulting from the action (Da) and observe the corresponding instantaneous reward (Table 1a). We discretize the temperature into three relevant states: temperature above the target range, temperature below the target range, and temperature within the target range. While the reactor is being controlled through the inlet mass flow rate (and hence Da), it is more convenient to express the action as the change in temperature resulting from the change in Da for the state-action-reward table. A high instantaneous reward of 1 occurs if the reactor operates within the target range $T^{n+1} \in [T^*_{\text{min}}, T^*_{\text{max}}]$, an intermediate reward of 0.8 occurs if the reactor operates outside the target range and approaches the target condition, and there is no reward for all other conditions.

Q is updated with Eq. (62) using the values in Table 1a. Q is typically expressed in a similar table known as a Q-table (Table 1b). The largest Q for the state $T^{n+1} < T^*_{\text{min}}$ is observed for action $T^{n+1}(\text{Da}^n) > T^n$ (Table 1b), which instructs the reactor to increase the temperature. In contrast, the largest Q for the state $T^{n+1} > T^*_{\text{min}}$ is observed for action $T^{n+1}(\text{Da}^n) < T^n$, which instructs the reactor to lower the temperature. Finally, when the temperature is within the target range (state $T^{n+1} \in [T^*_{\text{min}}, T^*_{\text{max}}]$), the reactor temperature is incentivized to remain unchanged.

When we employ Q-learning control on a well-stirred reactor, all episodes are initialized at the same temperature and mass flow rate. The state trajectories for a selected set of episodes exhibit robust convergence to the target conditions after approximately 30 episodes (Fig. 24b). During the 50$^{\text{th}}$ episode, the reactor operates close to the target range after timestep $n = 150$ (Fig. 24b). Overall, although this problem is relatively simplistic, it illustrates the ability of RL techniques to explore complex state relations that are encountered at stability boundaries and blowout conditions.

3.5. Integration of knowledge with ML

As ML methods proliferate within various research communities, more applications and modifications of ML are being made in order to achieve a variety of scientific objectives. As noted in Section 1.5, numerous texts [277,280,281] discuss the integration of ML methods

with the sciences. Here we explore how ML methods in Section 3.2 to 3.4 can be applied and modified to meet various objectives of combustion science and engineering.

3.5.1. Modifying ML for combustion science and engineering

Despite the continuing accumulation of data (Fig. 1a), cleanly labeled and structured data are still lacking within the combustion community. In fields tied to the natural sciences and engineering, including combustion research, scientific datasets are limited because (i) they can only be accessed by their owners and collaborators, (ii) they require significant preprocessing before being input to learning algorithms, and/or (iii) hard constraints can be present in scientific problems. These conditions differ significantly from fields that have applied ML techniques successfully: as example, the accessibility of computer vision techniques was significantly aided by the establishment of the ImageNet database [350], which contains tens of millions of labeled and sorted images.

By leveraging domain knowledge, the reliance on large corpora of labeled data can be reduced by embedding theoretical constraints within CombML methods and data, thereby ensuring that the trained ML model adheres to theoretical principles. Such knowledge-guided ML has been developed in order to improve upon the prediction performance, sample efficiency, and theoretical consistency of learning algorithms. Since SciEngML in general—and CombML in particular—have criteria that differ from those of conventional ML applications in data analytics, many of these algorithms have been extended to ensure interpretability and stability of ML predictions.

Given that supervised learning methods (Section 3.2) have received substantially more attention from various ML communities, it is no surprise that most of the developments behind knowledge-guided ML have largely been relevant to supervised learning methods. Many of the concepts that are applicable to combustion research were first pioneered in the related field of fluid mechanics [290]; applications in combustion itself remain nascent. Various components of supervised learning can be modified to formulate knowledge-guided ML methods, including loss functions, data, ML architectures, and model output/labels (Fig. 25, based on the general supervised learning algorithm from Fig. 6).

Regarding the modification of loss functions to knowledge-guided loss functions (Fig. 25, Option 1), in Section 3.2 we outlined supervised learning techniques in which training is guided by minimizing a loss function that measures the discrepancy between model predictions and data in some norm. In this training procedure, the algorithm is expected to infer latent relationships between input features and the target via complex correlations. However, a significant portion of this training process involves rediscovering the effects of physical constraints that are already known. This prior knowledge may be in the form of conservation principles, symmetry constraints, or invariance properties. In some cases, the ML algorithm may learn these relationships imperfectly, leading to poor generalization performance and a loss of trust. This lack of generalizability can be detrimental when scientific ML models violate fundamental conservation principles or break constitutive relations for transport properties, reaction rates, or secondary conservation principles.

Table 1
Q-learning applied to stabilizing the temperature of a partially well-stirred reactor.

State/Action	$T^{n+1}(\mathrm{Da}^n) > T^n$	$T^{n+1}(\mathrm{Da}^n) < T^n$	$T^{n+1}(\mathrm{Da}^n) = T^n$
$T^{n+1} < T^*_{\min}$	0.8	0	0
$T^{n+1} > T^*_{\max}$	0	0.8	0
$T^{n+1} \in [T^*_{\min}, T^*_{\max}]$	1	1	1

(a) Instantaneous rewards for a state-action pair.

State/Action	$T^{n+1}(\mathrm{Da}^n) > T^n$	$T^{n+1}(\mathrm{Da}^n) < T^n$	$T^{n+1}(\mathrm{Da}^n) = T^n$
$T^{n+1} < T^*_{\min}$	999.833	997.189	997.452
$T^{n+1} > T^*_{\max}$	998.671	999.788	998.628
$T^{n+1} \in [T^*_{\min}, T^*_{\max}]$	999.836	999.194	999.837

(b) *Q*-table for $n = 600$ at the 99th episode.

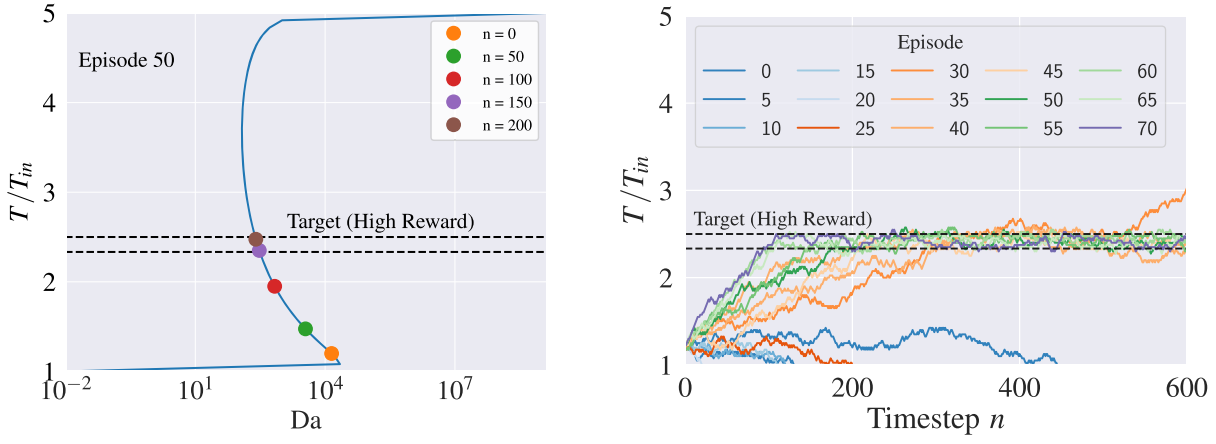
In the framework of theoretical principles, the constrained optimization problem can be expressed as

$$\arg\min_{\theta\in\mathscr{P}} \sum_{i=1}^{N} L(Y_i), \quad \text{subject to} \quad g(Y,\theta) = 0\,, \tag{64}$$

where Y_i are model predictions, θ is the unknown parameter state or a set of functional relationships, L is the loss function used for optimization, and $g(Y,\theta)$ represents knowledge-guided constraints. These constraints can consist of equality- or inequality-based constraints, holonomic or non-holonomic relationships, and may even be described via partial differential equations. In practice, these knowledge-guided constraints are weakly enforced [289], typically by introducing a regularization term [401]:

$$\arg\min_{\theta} \sum_{i=1}^{N} L(Y_i) + \lambda g(Y_i,\theta), \tag{65}$$

where the regularization coefficient λ is a hyperparameter whose value can be set using cross-validation (Section 2.5 and Eq. (35)).



(a) S-curve and solution for episode 50

(b) Trajectories

Fig. 24. *Q*-learning control of a well-stirred reactor: (a) S-curve and discrete instances at various timesteps for episode 50 and (b) selected trajectories for distinct episodes.

Using knowledge-guided constraints in regularization indicates a preference for functions that approximately adhere to physics constraints during training. Due to the weak enforcement of knowledge-guided constraints, g_k is written as:

$$\arg\min_{\theta} \sum_{i=1}^{N} L(Y_i) + \lambda g_k(Y_i, \theta). \tag{66}$$

In this scenario, labeled data may not even be required, as the knowledge-guided constraints can lead to the selection of functions from a hypothesis set. For example, this approach has been followed in computational physics, where the solutions of deterministic partial differential equations can be learned by deep neural networks via constraints [402,403]. In flow physics, these methods are exemplified by physics-informed neural networks [404], where g_k contains constraints related to the partial differential equations, initial conditions, and/or boundary conditions.

Regarding data (Fig. 25, Option 2), numerous approaches to augment training and testing datasets have improved the data efficiency and accuracy of scientific ML (Fig. 25). One of the most popular methods for improving the performance of ML algorithms, even outside the sciences, involves the use of feature selection and extraction methods [368,370] (Section 3.3). Within the context of scientific modeling, performing feature selection on datasets can range from straightforward approaches such as selecting the relevant inputs based on domain expertise [392] to more systematic approaches such as applying nonlinear transformations on inputs to match the target governing equations [405].

Regarding ML architecture (Fig. 25, Option 3), in Section 3.2.4 we discussed how modifications to the architecture of conventional feed-forward neural networks can result in CNNs and RNNs, which are better tailored to handle complex visual and sequential data, respectively. Similarly, the architecture of ML algorithms can also be altered to suit the needs of scientific modeling. For fields related to the combustion sciences and engineering, one of the first architectures tailored for fluid modeling was the tensor-basis neural network proposed by Ling et al. [406]. In that work, the network architecture was modified to guarantee Galilean invariance of the predicted tensor components, which was achieved by merging the outputs of two networks.

For the broader context of scientific and engineering modeling, neural ordinary differential equations (ODEs) [407] are another excellent case of knowledge-guided architecture. Recall that Eq. (47) contains the general equation for a sequence of hidden states A_t for RNNs in terms of weights W and biases b. Consider a special case of a one-to-many RNN where the hidden states are expressed as:

$$A_t = A_{t-1} + f(A_{t-1}, W, b)\ . \tag{67}$$

Neural ODEs seek to represent the hidden states by a continuous ODE:

$$\frac{dA(t)}{dt} = f(A(t), W, b, t)\ , \tag{68}$$

permitting the use of an ODE solver to evaluate the gradients of the loss function during back-propagation. The end result is a model with better stability in solving dynamical systems than RNNs [407]. Other examples entail the approximation of differential operators via convolutions [408] and enforcing conservation principles to ensure that the solution is divergence-free and fulfills Galilean invariance [409].

Finally, regarding model output/labels (Fig. 25, Option 4), we discuss methods of embedding the output of ML models with theoretical models. One popular example involves modeling most problems with conventional numerical methods and governing equations, while modeling a small aspect of the problem (such as closure models, model coefficients, and interpolated variables) with ML. These applications typically aim to develop data-driven models that are more accurate and cost-efficient than conventional modeling approaches. The development of ML-based closure models has been a particularly rich research area within the combustion modeling community, which will be further discussed in Section 4.

3.5.2. *Applying ML to combustion science and engineering*

Table 2 highlights three typical combustion-scientific objectives [277,280,281] that can be addressed with ML, along with the relevant applications to combustion. Given the massive developments in regression methods in supervised learning, especially in deep learning (Section 3.2.4), much of the focus in SciEngML is on improving scientific and engineering models, either by modeling correction terms or by replacing governing equations directly. In contrast, unsupervised learning methods, such as clustering methods (Section 3.3.1) and PCA (Section 3.3.2), have been used within various scientific communities to aid analysis and post-processing of data. However, the distinctions between supervised and unsupervised learning for specific scientific problems are not exact; they merely serve as a broad categorization. For instance, semi-supervised learning methods such as GANs (Section 3.4.1) can improve models, while supervised learning methods such as symbolic regression can be applied to discovering governing equations, as will be discussed in Section 4.1.4.

Specific applications of ML in combustion science and engineering—such as modeling combustion closure and identifying combustion manifolds—are explored in detail in Section 4. Note that a small portion of ML research is currently dedicated to modifying deep learning methods for solving partial and ordinary differential equations. Since this field is rather nascent, and is more pertinent to the broader scope of

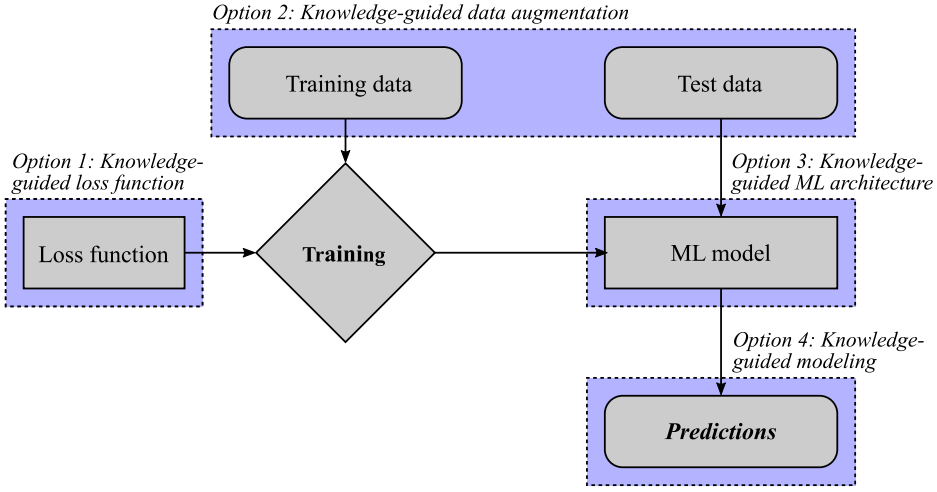


Fig. 25. Strategies for modifying various aspects of supervised learning in knowledge-guided ML. Blue boxes highlight components that can be modified.

Table 2
Scientific objectives and applications of CombML.

Scientific objectives	CombML application	Typical ML solution
Augmenting or substituting scientific models	Modeling of combustion thermochemistry; Parameterization of combustion manifolds; Combustion closure modeling	Supervised learning
Scientific discovery and categorization	Identification of combustion-controlling features and physical quantities; Identification of low-dimensional manifolds; Characterization of combustion regimes; Methods for analysis; Discovery of structures and coherent features from sensors and measurements	Unsupervised learning
Methods for solving differential equations	Improving numerical simulations	Supervised learning

various computational sciences, we point the interested reader to related articles in the field [410,411].

4. Applications

To examine the application of various ML methods to combustion, we consider three topical areas. Section 4.1 examines the application of CombML to problems pertaining to fundamental combustion problems, including the representation of thermochemical properties, the construction of chemical mechanisms, the identification and parameterization of combustion manifolds, and the formulation of combustion closure models. Applications discussed in this section largely employ supervised and unsupervised learning methods that were discussed in Section 3.2 and 3.3, respectively. Section 4.2 reviews CombML for applications to propulsion and energy-conversion systems, specifically examining CombML techniques for surrogate modeling, fault detection, data analysis, and control. Apart from supervised and unsupervised learning methods, semi-supervised learning techniques will find applications for control and optimization in this field. Section 4.3 is concerned with reviewing CombML applications for fire and explosion hazards, accidents and risk management. This field of applications is particularly challenging for CombML methods due to the requirements on the accurate representation of rare extreme events and the need for interpretability of the resulting ML models.

4.1. ML for fundamental combustion investigations

Numerical simulations of reacting flows are a critical component in the analysis, design, and optimization of propulsion and energy-conversion systems (Section 4.2) as well as in assessing risk and mitigating accidental fires and combustion hazards (Section 4.3). The governing equations that describe these flows have been established from physics-based principles. In this section, we discuss progress and outline opportunities for utilizing ML techniques and data-driven approaches for combustion modeling and predictions.

The main challenges in solving the governing equations, Eq. (1), arise from key issues pertaining to the chemical complexity and the wide range of spatiotemporal scales in combustion [256]. Methods have been developed for obtaining compact descriptions of reaction chemistry in order to replace the state vector U in Eq. (1) with $V \in \mathbb{R}^{N_V}$ where $N_V \ll N_U$, resulting in:

$$\partial_t V + \nabla\cdot F(V) - \nabla\cdot Q(V,\nabla V) = S(V)\ , \tag{69}$$

where $V = (\rho u^\mathsf{T}, \rho\psi^\mathsf{T})^\mathsf{T}$ and $\psi \in \mathbb{R}^{N_\psi}$ parameterizes a low-dimensional manifold that approximates the thermochemical state vector

$$\phi \simeq \widehat{\phi} = \mathscr{M}(\psi)\ . \tag{70}$$

The vector ψ may include a subset of species mass fractions, derived quantities (for example mixture fraction or reaction progress), or other flow field-describing quantities such as strain rate or scalar dissipation rate. While different manifolds share similar representations for $\widehat{\phi}$, the structure of the transported quantities ψ and the functional relation of the manifold representation $\mathscr{M} : \mathbb{R}^{N_\psi} \to \mathbb{R}^{N_{\widehat{\phi}}}$ exhibit considerable variations. Since $N_\psi \ll N_\phi$, the utilization of a manifold can significantly reduce the computational cost. Various manifold techniques have been developed [256] that differ in terms of chemistry manifolds [257–259,261,263,412], reaction-transport manifolds [264,266,267,413,414], thermodynamic manifolds [415–417], and empirical manifolds [268,418]. Because of their flexibility and generality, ML techniques have been employed for parameterizing thermochemical properties, for developing chemical-kinetic models, and for developing low-dimensional combustion manifolds. These applications are discussed in Section 4.1.1 to 4.1.3.

ML techniques have been employed to address the wide range of scales, spanning the system-level device scale to the smallest Kolmogorov length scale, reaction zone thickness, and thermoviscous sublayer thickness. As a consequence, resolving all scales remains infeasible for most applications; statistical representations using Reynolds-averaged Navier-Stokes (RANS) and LES methods are commonly employed [254,255]. In LES, a low-pass filter,

$$\overline{\phi}(x,t) = \int \phi(x-\xi,t;\Delta)G(\xi;x)d\xi\ , \tag{71}$$

is applied to the governing equations, Eq. (69), to separate the resolved scales from the subgrid scales (SGS), taking the following form:

$$\begin{aligned}\partial_t\overline{V} + \nabla\cdot F(\overline{V}) - \nabla\cdot Q(\overline{V},\nabla\overline{V}) = \underbrace{\overline{S(V)}}_{S_{\mathrm{TCI}}}\\ +\nabla\cdot\underbrace{(F(\overline{V}) - \overline{F(V)})}_{F_{\mathrm{SGS}}}\\ -\nabla\cdot\underbrace{(Q(\overline{V},\nabla\overline{V}) - \overline{Q(V,\nabla V)})}_{Q_{\mathrm{SGS}}}\end{aligned} \tag{72}$$

where the SGS contributions are collected on the right-hand side and represent the couplings among turbulence and reaction chemistry, turbulent stresses, and turbulent transport. Traditionally, closure models have been derived from physical principles, calibration, and empirical knowledge [255,419]. In Section 4.1.4, we discuss CombML approaches for constructing closure models of these SGS terms.

4.1.1. Regression of thermochemical properties

Historically, regression analyses have played a key role in evaluating thermochemical properties—such as formation enthalpy, standard entropy, and heat capacity—from experimental data. Perhaps the best known method is Benson's group additivity [420]. In this method, a thermochemical property of a compound is expressed as a linear combination of contributions from its functional groups:

$$\phi = \sum_{i=1}^{N_{\mathrm{groups}}} \alpha_i\varphi_i\ , \tag{73}$$

where φ_i denotes the partial contribution to the property from the i^{th} group and the coefficient α_i is the number of groups in the compound. The partial contributions are determined by fitting to experimental data. As such, this method can be considered as an early but highly successful

learning technique. With the advent of ML, new avenues for regressing thermochemical data have been explored and various methods are now well established.

Using quantitative structure-property relationship (QSPR) approaches, group-additivity methods have now been generalized by considering a broader range of molecular descriptors [421–423], including constitutional properties (number of atoms and bonds as well as molecular weight), structural information (molecular topology, fragments, and functional groups), as well as geometric (moments of inertia, shadow descriptors) and quantum-chemical properties (such as dipole moments, orbital structure representation, or ionization potential) [422,423].

The representation of nonlinear functional relations, the consideration of a higher-dimensional feature space, and the discovery of latent spaces have led to rapid adaptation of ML techniques in this field. In particular, because of the ability to approximate complex and highly nonlinear functions, feedforward neural networks and other supervised learning methods have found widespread applications for regressing various thermodynamic and physical properties, including melting/boiling/flash points, viscosity, vapor pressure, critical-point properties, formation enthalpies, standard entropy, octane and cetane ratings, sooting propensity, and more [424–433]. Data generated from high-throughput screening experiments and first-principles methods enable the generation of comprehensive datasets that cover a wide range of fuel classes, hydrocarbons, and oxygenated compounds as well as multicomponent, synthetic, and biomass-based fuels. In particular, ML-based regression techniques have been established for discovering the structural complexities of biofuel compounds and for exploring fuel properties to tailor cleaner-burning biofuels [434].

In an example of a supervised learning framework for regressing the thermochemical property of the yield sooting index from high-throughput screening experiments (Fig. 26), molecular descriptors were generated for each compound in the experimental database [435]. A series of preprocessing steps populated descriptor matrix with missing values and normalized the descriptors. Next, a recursive feature elimination strategy was employed: features were removed by iteratively fitting a SVM regressor and removing less-important descriptors. The key features identified through this process constitute inputs to a neural network that was trained to predict the yield sooting index (Fig. 26). These feature selection and extraction steps are typically required to achieve optimal performance when using traditional ML methods, and can often be aided by unsupervised learning methods (Section 3.3).

While feedforward neural networks provide accurate predictions, they lack interpretability (Section 5.2) and cannot contribute in generating fundamental insight. Miraboutalebi et al. [436] demonstrated that random forests feature importance (Section 3.2.3) could be employed to identify the most important fuel components in determining the cetane number of a biofuel blend. In another investigation, Kessler et al. [437] compared feedforward networks, graph neural networks (a type of deep learning architecture [438]) and multivariate equations for predicting sooting propensity. While graph neural networks and multivariate equations lacked predictive accuracy when compared to feedforward networks, these methods were shown to provide fundamental insight on the relationship between molecular structures and sooting behavior. Another benefit from applying graph-based approaches comes from their greater flexibility for discovering latent features from complex molecular structures, where substructural features and property relationships are directly learned from atom-level features of molecular-graph representations [439,440]. This kind of work is related to a subfield known as representation learning [441], which aims at tailoring ML algorithms to specific data structures.

Apart from considering molecular-structure information, other investigations have utilized first-principles simulations in the construction of CombML models. For example, Rupp et al. [442] employed kernel ridge regression as a regularization method to model the atomization energy from information in the Hamiltonian about nuclear charges and Coulomb forces. Cross-validation for a training set involving 7000 molecules showed that the mean absolute error reduces to $\sim$ 10 kcal/mol, which is comparable to mean-field electronic structure theory [443]. Subsequent work by Hansen et al. [444] evaluated various ML algorithms that included support vector regression, neural networks, and *k*-nearest neighbor algorithms, achieving three-fold improvements in accuracy compared to Rupp et al. [442]. The importance of selecting models, optimizing hyperparameters, and representing physical properties and invariant representations of the molecular structure were recognized as main factors that improve accuracy. To address the variability in the predictions from different CombML models that are trained on the same data, consensus models have been constructed that are derived from weighting model predictions using strict, majority, or probability-based consensus methods [445].

With relevance to reducing numerical errors from model approximations, hybrid approaches have been developed in which first-principles calculations are augmented with neural networks trained with experimental data to constrain the computational models [446–448]. In contrast to such weakly coupled approaches, a self-evolving learning model was developed by Li et al. [449] that combines an active learning machine with automatic first-principles calculations to improve the accuracy of computing thermochemical data by exploring the molecular structure of the chemical system. This active learning method is a type of semi-supervised learning strategy (Section 3.4) that uses a third source of information, such as first principles calculations, to augment an existing chemical database.

In summary, the extensive databases for thermochemical properties are ideally suitable for applications of data-driven methods to extract complex property relations [450]; they are expected to replace traditional group-additivity methods. However, the lack of interpretability remains an outstanding research issue towards generating fundamental insight. As such, knowledge-guided and interpretable ML methods will be key-enabling techniques for establishing these methods as routine analysis tools for screening and fuel characterization. In addition, the strong dependency of the model performance makes the optimization of hyperparameters and the development of robust procedures for training critical; establishing community knowledge in the efficient construction of ML models is therefore necessary to accelerate the transition of these methods into practical applications. While traditional ML methods have been widely employed for feature extraction, emerging and specialized network architectures, such as graph neural networks, are poised to offer new opportunities to directly learn from complex and heterogeneous thermochemical properties and molecular structures, without the need for feature engineering.

4.1.2. *Chemical kinetics and chemistry adaptation*

The paucity of experimental data and the chemical complexity are the main challenges in constructing and utilizing detailed kinetic mechanisms and transport models for multidimensional combustion simulations. Various ML algorithms have been explored to address these challenges. These strategies can be categorized as (i) ML techniques for constructing surrogate models to facilitate fast sampling of the thermochemical state space, (ii) ML methods for constructing reduced mechanisms, (iii) ML approaches for optimizing rate parameters, and (iv) ML-based approaches for inferring chemical principles and reaction pathways from data.

CombML has been employed to construct surrogate models and reduced chemical mechanisms. Compared to established chemical-reduction strategies that are solely based on analytic principles [250,451], ML techniques offer greater flexibility and adaptability to specific problems. For example, Li et al. [452] developed an ML-based surrogate formulation to facilitate rapid sensitivity analysis of large chemical systems. In this formulation, a feedforward neural network was trained by sampling the output of a detailed chemical model over uncertain model parameters at a specific operating condition. Shallow neural networks with one hidden layer were used for the computationally

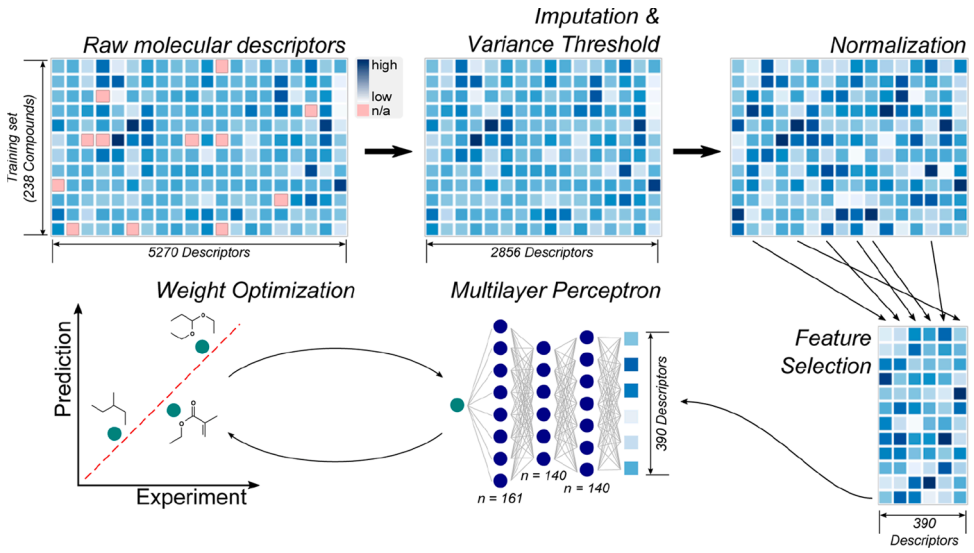


Fig. 26. Overview of a data-driven supervised learning framework for regressing thermochemical properties from raw molecular descriptors. Reduction of molecular descriptors is achieved through a series of analysis steps that involve imputation of missing data, variance thresholding, and normalization before feature selection is employed for training a feedforward neural network. Reprinted with perimission from [435]. Copyright 2017 American Chemical Society.

efficient evaluation of the sample space. The output of the network was fed into a random-sampling high-dimensional model representation for global sensitivity analysis. As such, this formulation provides a computationally efficient way to evaluate model sensitivities. Similar surrogate modeling formulations have been employed for accelerating Bayesian analyses of combustion kinetic models and for performing parametric sensitivities of ignition-delay predictions [453–456].

ML techniques have also been explored to develop and optimize chemical mechanisms for large hydrocarbon fuels, transportation fuels, and fuel blends. Many of these developments take advantage of the hierarchical structure of chemical mechanisms [457,458] or are built on recently developed hybrid-chemistry concepts [459,460]. In particular, these concepts are based on the principle that the high-temperature oxidation of large-hydrocarbon fuels can be represented by two steps: fast pyrolysis followed by oxidation of the pyrolysis fragments. Measurements of pyrolysis products are commonly used to constrain the kinetic parameters in the lumped pyrolysis model [459]. Instead of hypothesizing a particular pyrolysis model, Ranade et al. [461,462] introduced a two-step regression approach to describe the pyrolysis chemistry from measured data; shallow neural networks with limited expressiveness were employed to determine the reaction rates of measured species by fitting concentration profiles to experimental data. A feedforward neural network was then employed to relate the nonlinear reaction rates to concentrations of measured species during the pyrolysis stage. This model was combined with a foundational chemistry model to represent the reaction chemistry of the pyrolysis fragments, yielding effective chemistry reduction schemes for large hydrocarbon fuels. Further opportunities to extend this approach arise because often only a subset of chemical species are measured, the reaction pathways in the pyrolysis model are decoupled from the fragment-oxidation model, and the ML model can be augmented with knowledge about physical principles and conservation laws.

Inspired by this ML-based hybrid chemistry model, a data-driven model for chemistry acceleration was developed by Alqahtani and Echekki [463] in which reaction rates of representative species were modeled using feedforward neural networks and the remaining species were represented using a compact foundational chemistry model. An network classifier was employed to identify species that were represented by different chemical submodels.

By considering the oxidation of *n*-pentanol, Chang et al. [464] developed a procedure for constructing a skeletal mechanism that combines chemical submodels, optimization of reaction-rate parameters, and sensitivity analysis. The chemical mechanism that was constructed by combining a skeletal pyrolysis mechanism, a reduced C_2-C_3 submechanism, and a detailed $H_2/CO/C_1$ model was optimized against species measurements from jet-stirred-reactor experiments and ignition-delay measurements from shock-tube and rapid compression machine experiments using a GA. Uncertainties of the resulting mechanisms were assessed using polynomial chaos expansions.

A self-adaptive differential evolution algorithm was employed by Cheng et al. [465] to optimize chemical kinetic models using ignition-delay-time measurements from shock-tube experiments. While several statistical regression methods have been employed to infer rate parameters [274,466], interestingly, this approach used a differential evolution algorithm [467,468] to learn the preexponential factors and low-pressure limits for a subset of elementary reactions that were identified through prior sensitivity analysis. In this study, evolutionary algorithms enabled the self-adaptation of model parameters from previous experience to improve the fitness of the model by minimizing the error in the predicted ignition delay with respect to measured data.

The discovery of physical principles through data-driven techniques has been an area of growing interest, and significant progress has been made in developing CombML methods for uncovering dynamic processes, conservation principles, and kinematic relations [284,407,469–473]. These developments include the extension of the sparse identification of nonlinear dynamics method for determining rate coefficients of reaction networks from noisy data [474], the use of mixed-integer linear programming for finding reactions in chemical mechanisms that are consistent with steady-state concentration profiles [475], and the postulation of a general differential model that is exposed to mathematical and statistical tests to reduce the model to a subset of reactions [476]. Other approaches encapsulate Arrhenius expressions and other physical constraints into neural networks to map reaction rates, facilitating interpretability, and the quantification of rate parameters [477].

In summary, CombML has been demonstrated as being viable for augmenting chemical-kinetic mechanisms and for representing simplified chemical systems; yet the representation of entire mechanisms that are representative of complex transportation fuels remains an outstanding issue that requires embedding physico-chemical principles, devising ML architectures that map well to the intrinsic stiffness, complex reaction pathways, and scale separation, as well as considering

Bayesian ML methods for uncertainty quantification and robust simulations in CombML applications.

4.1.3. Identification and parameterization of combustion manifolds

To reduce the computational cost associated with the evaluation of thermodynamic properties, chemical species, and reaction rates, low-dimensional manifold representations are frequently employed in combustion simulations. Unsupervised learning techniques for dimensional reduction (most notably PCA) and supervised learning for regressing thermochemical manifolds are ML-based methods for addressing these problems.

Manifold identification In addition to methods that are founded on physical and theoretical principles for constructing low-dimensional manifolds [256], data-driven techniques have seen considerable success in combustion applications. Most notable is the application of PCA (Section 3.3.2) to data from experiments and simulations to identify low-dimensional state-space representations, the reduction of chemical-kinetic mechanisms, and the construction of combustion models. Maas and Thévenin [478] employed PCA to analyze species correlations in a turbulent non-premixed hydrogen-air flame. Parente et al. [268] established PCA as a method for automatically identifying low-dimensional manifolds in flames and their corresponding parameterization through the selection of optimal reaction variables. However, the application of PCA as a global method for identifying a compact representation of thermochemical measurements from turbulent flames revealed deficiencies attributed to the nonlinear thermochemical state space and to the fact that different regions in the flame map to different regions in composition space. To overcome these deficiencies, local PCA methods were developed. In these methods, data were separated into clusters that were either preselected based on physical principles (considering mixture-fraction conditioning) or identified with an unsupervised partitioning algorithm (using vector quantization). Application to experimental data showed that local PCA methods provide substantial improvements in identifying reduced state-space representations with lower reconstruction errors [268]. Parente et al. [479] successfully employed local PCA to relate experimentally observed modifications in the flame structure and emissions to the oxygen-dilution in jet-in-hot-coflow flames that operate in the MILD (Moderate or Intense Low-oxygen Dilution) combustion regime. PCA showed that the major species were associated with the first cluster, whereas the principal components in the second cluster were strongly correlated with radical and intermediate species of OH and CO, demonstrating the selectivity of local PCA for combustion analysis.

Further improvements in PCA performance for heterogeneous datasets were made by Coussement et al. [480], who introduced a kernel-density weighted PCA for optimally sampling from heterogeneous data. Mirgolbabaei and Echekki [481] proposed a kernel PCA method to regularize complex thermochemical state spaces. This method introduces a nonlinear mapping that transforms the original data to a higher-dimensional feature space to which linear PCA is applied, with high compression potential. A similar concept of transforming the input data to a different feature space was explored using an autoencoder for identifying nonlinear principal components [482].

PCA methods have been combined with nonlinear methods for parameterizing thermochemical manifolds, since plane PCA manifolds inadequately represent the nonlinear composition space in combustion applications [483]. Biglari and Sutherland [484] used multivariate adaptive spline regression (MARS) [485] to parameterize low-dimensional manifolds as a function of a reduced set of principal components. In this approach, PCA was utilized to identify a set of optimal bases; MARS was then employed iteratively to select a set of basis functions that minimizes the regression error. An application of this approach is illustrated in Fig. 27, comparing the parameterization of the OH mass fraction as a function of the first two principal components [484]. The PCA presentation of Y_{OH} by a PCA hyperplane is not able to capture the nonlinear dependence on the principal components. Other nonlinear regression models for mapping thermochemical state spaces to a reduced set of principal components are feedforward neural networks [486,487], support vector regression [488], and Gaussian process regression [488,489]. These methods will be discussed below with specific focus on neural networks for parameterizing multidimensional manifolds.

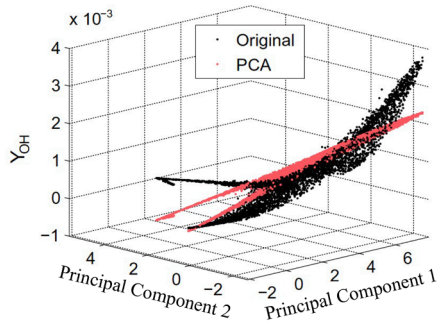
The potential of PCA for combustion modeling has been recognized. Sutherland and Parente [418] were the first to propose a principal component scoring approach that directly solves transport equations for the principal components; the state-space variables were constructed from the principal components. Other PCA-based modeling approaches considered solving for a reduced set of state-space variables and reconstructing the remaining variables from the principal components [490,491]. D'Alessio et al. [492] employed PCA in an adaptive-chemistry approach to identify local clusters in the simulation domain that can be represented by specialized reduced chemical kinetic mechanisms. PCA-based combustion models for reacting-flow simulations were explored by considering various combustion problems that include a perfectly stirred reactor [488,489], one-dimensional turbulence configurations [487,493], unsteady two-dimensional flame configurations [490,491], and turbulent flames [494].

With the success of these PCA-based methods for the automatic identification of low-dimensional manifolds and the construction of combustion models come various opportunities to improve these purely data-driven methods. In particular, integrating scientific and engineering knowledge (Section 3.5) could help to address the interpretability of the principal components in relation to local combustion-physical processes as well as the consideration of nonlinear relationships among data, data normalization, and sensitivity to outliers that must be considered when dealing with experimental and sparse datasets [377,479,482].

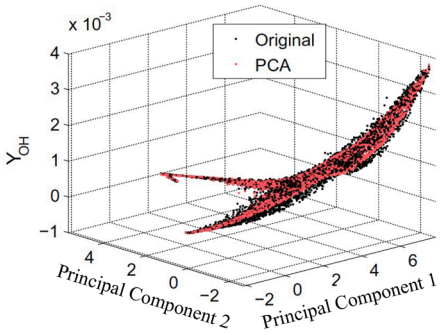
Manifold parameterization In parameterization techniques, the thermodynamic state space is either precomputed in its entirety or dynamically computed during the simulation and stored for further reuse. Such techniques include *in situ* adaptive tabulation [495], piecewise polynomial approximation [496], mapping methods [497,498], and tabulation techniques [499–504]. While tabulation techniques such as conventional structured look-up tables are often employed to parameterize the composition space, interpolation order, data access, and storage requirements are major limitations to the scalability of this procedure. In particular, the memory allocation that is required to store these tables grows exponentially with dimensionality as $\mathscr{O}(10^{N_\psi})$, thereby limiting their utilization for low-dimensional manifold parameterizations with $N_\psi \lesssim 4$. This limitation imposes challenges due to, first, the reduced memory footprint of emerging compute architectures and, second, the need to incorporate more complex combustion-physical phenomena such as heat-loss effects [505–510], emissions [511–513], multistream flows [514–518], transient processes [501,519,520], and dispersed-phase combustion [521–527]; higher-dimensional manifold representations are thus required, rendering these tabulation techniques unfeasible.

The benefits of supervised learning techniques in overcoming deficiencies of traditional tabulation techniques have been recognized by the combustion community. In particular, feedforward neural networks have been widely applied for chemistry approximations and the representation of thermochemical manifolds.

Neural networks were first introduced by Christo et al. [325] to represent the reaction chemistry in transported PDF combustion simulations. Training data were generated through statistical mapping: small-scale PDF/Monte-Carlo calculations were performed to populate the composition space that is encountered in the application. A feedforward network architecture with two hidden layers, having an equal number of neurons with sigmoidal activation function, was employed. The input features consisted of mixture fraction and reaction progress variable; the output corresponded to the production rate of progress variable. Model parameters were trained with a back-propagation



(a) Linear reconstruction using PCA.



(b) Nonlinear reconstruction using MARS.

Fig. 27. Low-dimensional manifold parameterization of Y_{OH} from two principal components using (a) PCA and (b) MARS. Reprinted from [484], Copyright 2012, with permission from Elsevier.

algorithm with adaptive learning rates to improve convergence. A nonlinear transformation was applied to the input data to ensure that the training data were sampled from a more uniform distribution. This work was extended to more complex chemical systems involving three- and five-step reduced mechanisms for H_2/CO_2 [528]. The network performance with respect to generalization, computational cost, and memory requirements was examined through parametric studies by changing the number of input/output channels and network hyperparameters. While significant reductions in memory requirements were already evident for a small number of input variables, benefits in the computational efficiency over conventional look-up tables increased with the chemical complexity and dimensionality of the input state.

Instead of representing the combustion manifold through a network, Blasco et al. [529] embedded the temporal evolution of a homogeneous reactor system into a feedforward neural network. In this approach, a concurrent network specialization was employed (Fig. 28) in which one network was used to represent the chemical mapping, $Y(t+\delta t) = \mathscr{A}_1(Y(t))$ (Fig. 28a), and the second network represented the thermodynamic state, $\{\rho, T\}(t+\delta t) = \mathscr{A}_2(Y(t))$ (Fig. 28b), where $\mathscr{A}_i$ denotes a specific ANN. Shallow networks with at most two hidden layers were considered for representing the chemical and thermodynamic states and the input state was described by the species composition. Issues pertaining to generalization and sensitivity to the training data were addressed by biasing the training set away from the equilibrium composition. This treatment can be considered as an early form of knowledge-guided ML (Section 3.5). Separate networks for different time increments δt were generated. Analysis of the network performance showed that the accuracy in representing the chemical system is species-dependent: highly reactive and minor species that evolve on fast chemical timescales incur larger errors. Sample selection and partition of the composition further improved the accuracy of this approach. These issues were addressed in a subsequent work [530] in which the input state was augmented by including the timestep size δt and partitioning the composition space to represent each subdomain with feedforward neural networks of lower architectural complexity. In this context we note that recent developments of RNNs (Section 3.2.4) offer interesting opportunities to consider transient effects that are considered in the chemical system.

Self-organizing maps were utilized for automatic subdivision of the composition space [531,532]. The partition into subdomains of reduced topographical complexity enabled the utilization of shallow networks with fewer hidden layers. Another approach was proposed by Chen et al. [533] in which an *in situ* adaptive tabulation-generated thermochemical state-space representation was divided into two-dimensional subdomains that were fitted to feedforward neural networks with the goal of improving the linear approximations and mitigating the storage requirements. While appreciable reductions in memory were reported as a consequence of the network parameterization through high-dimensional transfer functions, the accuracy strongly depended on the network architecture and the comprehensiveness of the training data (also shown in Section 3.2.6).

Accuracy issues of network representations were addressed by developing a method for identifying optimal feedforward network architectures [534,535]. This method utilized a generalized mixed-variable pattern search [536] to optimize the number of neurons, hidden layers, transfer functions, and connectivity, with subsequent extensions including the optimization of hyperparameters in the transfer function [537]. In this study, the automatic generation of optimal network architectures showed that deep networks with dense connectivities beyond neighboring layers can significantly improve the descriptive accuracy of networks. Quantitative comparisons of optimal network performance against conventional tabulation techniques demonstrated significantly higher knowledge density, which was defined as the ratio between accuracy and memory requirement [537]. These studies demonstrate an early attempt within combustion research in automated ML [349], a subfield dedicated towards hyperparameter and architecture optimization strategies, which has since gained a large interest due to the proliferation of deep learning methods. Owing to the high dimension of the search spaces, traditional approaches for hyperparameter optimization—such as manual tuning, grid search, and random search—do not scale well. However, in the recent past, better procedures such as tree-structured Parzen estimators [538], Bayesian optimization [348], and hyperbands [539] have shown effective and efficient determination of optimal neural network hyperparameters. The availability of these methods in open-source libraries [540] provides a potential solution to address this issue.

Neural networks have been employed in unsteady turbulent combustion simulations. Flemming et al. [541] and Kempf et al. [542] performed LES of a turbulent jet flame in which separate networks were used to represent individual thermochemical quantities. These networks were trained from steady flamelet solutions that were filtered to account for turbulence/chemistry coupling. Standard feedforward networks with two hidden layers and an output layer were used.

Filtered steady-state flamelet solutions were also employed [535] to construct optimal neural networks for simulating a bluff-body swirl-stabilized flame. Specialized optimal feedforward neural networks for each thermochemical quantity consisted of up to four hidden layers with eight neurons per layer. Results from these simulations are presented in Fig. 29. The instantaneous temperature field (Fig. 29a) illustrates the turbulent flow field in this bluff-body flame. Comparison of radial profiles for mean and root mean square of mixture fraction between simulations using optimal neural networks and conventional look-up tables (denoted by $\mathscr{T}$; Fig. 29b) show comparable results and overall good agreement with experimental data.

Sen and Menon [543] employed standalone simulations from a linear-eddy model (LEM) [544,545] to train feedforward neural networks. LEMs provide a parametric description of an unsteady turbulent mixing process in a one-dimensional domain [545]. *A priori* and *a posteriori* applications to turbulent flames [543,546] showcased the ability

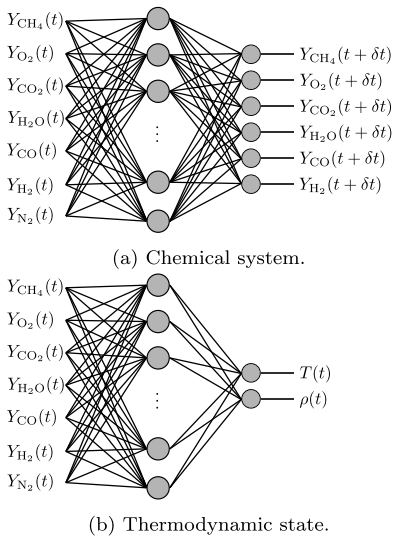


Fig. 28. Concurrent network specialization for representing the temporal evolution of (a) a chemical system and (b) a thermodynamic state. Adapted from [529], Copyright 1998, with permission from Elsevier.

of these LEM-trained feedforward neural networks to describe turbulent combustion and underscored the importance of considering micro-mixing during network training. To achieve acceptable accelerations over direct integration techniques, smaller networks with fewer connectivities were preferred over deeper networks with better fitness. The generality of the network parameterization was addressed by considering combustion conditions outside the training base. Although acceptable performance was reported [543,546], there are ongoing opportunities (Section 5.4) to address the prevailing issues of these regression-based data-driven techniques in extrapolating as well as ensuring scalar-boundedness and conservation.

A PCA network-based chemistry tabulation approach was explored by Dalakoti et al. [547], with various canonical flame configurations used to train the PCA-network model. *A priori* investigations of the model's application to a DNS of a turbulent lifted jet flame was performed, and deficiencies pertaining to mass conservation and the accurate representation of minor species were addressed.

Chatzopoulos and Rigopoulos [548] and Franke et al. [549] extended the methodology of Blasco et al. [531] that combined self-organizing maps with feedforward neural networks for optimal regression of the thermochemical composition space in applications to RANS and LES calculations of turbulent flames. By considering a generic training set, the networks showed a small capacity for extrapolation, but accurate predictions were challenging when the target predictions deviated too far from the training set. In the absence of very large datasets, this limitation in extrapolation is typical, and has spawned growing interest in constrained learning approaches that consider domain knowledge (Section 3.5). With the goal of applying this method in LES-PDF simulations of a turbulent flame, Franke et al. [549] constructed the composition space from solutions of the unsteady flamelet equations that were computed for various strain-rate conditions in order to capture conditions encountered in turbulent-flame simulations. The thermochemical composition space was represented by 400 subdomains, which were designated with a clustering algorithm, with identical network architectures consisting of two hidden layers with 30 neurons per layer. Application of this regression method to unsteady LES-PDF simulations was in excellent agreement with a finite-rate LES-PDF simulation.

Wan et al. [233] employed neural networks for regressing chemical source terms for DNS computations. In this approach, the low-dimensional reaction-diffusion manifold was constructed from stochastic micro-mixing simulations in which the reaction chemistry was represented by a reduced chemical mechanism. A standard feedforward neural network architecture was used to map the input vector (consisting of species mass fractions and temperature) to the chemical source terms. Physical constraints were not explicitly incorporated in the network and mass conservation was enforced during the simulation. Appreciable reductions in cost with good accuracy in a DNS application were reported.

Modeling internal combustion engines that require the consideration of large chemical mechanisms, Owoyele et al. [550] utilized feedforward neural networks for regressing solutions from an unsteady flamelet model by accounting for variations in pressure, residence time, and turbulence/chemistry interaction through a presumed PDF model. A knowledge-guided group-multi-target network approach was proposed: each group of species that were highly correlated are represented by a common network with multiple output nodes. Training (via backward propagation) these multiple networks is more straightforward than training a single network that outputs every species, while avoiding laboriously training one network for each species. An alternative approach for representing complex combustion manifolds is the modular connectionist architecture in which expert networks that constitute the architecture compete to learn specific subsets of the training data [551,552].

Together, these examples highlight the maturity of supervised learning techniques for representing complex combustion manifolds. The ability of networks to represent complex relationships between the input and target spaces offers alternatives to commonly employed tabulation techniques, which are traditionally limited by memory requirements. However, despite successes, so far these methods have not seen widespread application to combustion simulations due to a lack of robust control of network fitness, interpretability, and enforcing physical principles. Therefore, the utilization of knowledge-guided data-driven approaches (Section 3.5) for embedding constraints on physical properties, boundedness, and conservation principles offer promising opportunities for transitioning these methods into practical application. Similarly, details about the network architecture, the hyperparameter search, and the training procedure are not always reported, which hampers the development of a knowledge base within our combustion community. This can be strengthened by creating public datasets to promote the development of open-source CombML algorithms, similar to AlexNet [553], VGGNet [554], or ResNet [555] for image recognition.

4.1.4. *Turbulent combustion closure modeling*

CombML has been employed for developing data-driven models for SGS contributions and turbulence/chemistry interaction in filtered turbulent reacting flow equations (Eq. (72)). Many of these developments have been based on data-driven models of non-reacting turbulent flows

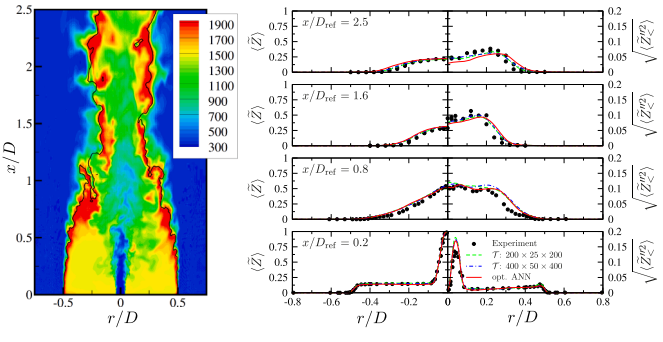


Fig. 29. LES of a bluff-body swirl-stabilized flame using optimal feedforward neural networks to represent the thermochemical state space. (a) Instantaneous temperature field. (b) Measured and computed mean and root mean square mixture fraction profiles. Reprinted from [535], Copyright 2009, with permission from Elsevier.

[392,406,556–560]. In contrast to the use of feedforward neural network architectures with thermochemical vectors in Section 4.1.3, convolutional architectures are preferred when modeling turbulent closure due to their suitability in learning from spatial data.

Turbulence/chemistry interaction An open research issue in turbulent combustion is the modeling of the filtered chemical source term $\overline{S(V)}$ in Eq. (72). Studies have employed supervised and unsupervised learning techniques to model this term. Deep neural networks were shown in an *a priori* study by Chen et al. [561] to outperform algebraic models at large filter sizes when employed as a model for the joint filtered density functions of mixture fraction and progress variable from DNS of MILD combustion.

Lapeyre et al. [562] employed a CNN for regressing the SGS flame surface density in premixed turbulent flames. Instead of using information about the local velocity field and gradients of the reaction progress variable, in this supervised learning approach only the filtered progress variable was used as the input feature field. The training data were generated from filtered DNS results that were then sampled onto a coarse mesh to train a U-Net, a popular CNN architecture. *A priori* tests on the same DNS configuration with impulsively changing inlet conditions showed good performance and improved accuracy over physics-based closure models, demonstrating the ability of CNNs to extract structural information and representative features. CNNs were also explored in *a priori* studies for performing deconvolution tasks and approximating the progress variable variance of filtered DNS data of a freely propagating turbulent premixed flame [563]. While this approach showed promise in modeling turbulence/chemistry interaction, the authors noted that a large amount of data is required to train their deep learning approach to ensure good performance in making out-of-distribution predictions. As will be discussed in Section 5.1, the assembly of a public database would address this issue, while also providing means of comparing different CombML methods and strategies on a common dataset. Other potential CombML research opportunities include improving prediction accuracy by exploring superresolution methods [279,564,565] for learning the latent space as an unsupervised learning task, as well as encoding of regularization conditions [566,567] and physical constraints (Section 3.5) for improving out-of-distribution predictions.

Ranade and Echekki [568], 569] proposed a data-driven method for constructing the joint scalar PDF of the thermochemical state space from experimental data of multiscalar measurements. Their method combined PCA (Section 3.3.2) for parameterizing the composition space with multidimensional kernel density estimation to generate a scalar PDF. The parameterization of the resultant PDF was learned from measurement data. This method was evaluated in *a priori* studies and *a posteriori* RANS simulations of the Sandia piloted jet flame configuration under various conditions. Henry de Frahan et al. [570] evaluated various ML algorithms for representing joint PDFs of mixture fraction and progress variable; their study considered a single snapshot of a DNS from a lean premixed low-swirl burner. The efficacy of a random forest, a fully connected neural network, and a variational autoencoder was examined by performing *a priori* analysis on different regions within the same flame. Results showed that the fully connected network was most accurate and that the employment of the variational autoencoder, a generative method, did not improve predictive accuracy. This contrasts the conclusions from Bode et al. [571] where a Wasserstein GAN, another generative method, was shown to outperform a CNN in capturing smalls-scale structures in scalar transport within a turbulent flow. These mixed findings demonstrate the necessity of a sufficiently complex benchmark dataset that can be employed by the CombML community for comparing various ML methods.

SGS transport and mixing Of equal importance to the description of turbulence/chemistry interaction is the modeling of the SGS turbulent transport and SGS molecular diffusion flux, F_{SGS} and Q_{SGS}, that appear in Eq. (72).

The discovery of model forms for the SGS stress tensor in turbulent premixed flames was explored by Schoepplein et al. [572]. In this approach, generic functional expressions for the SGS stress tensor, $\tau_{ij}^{\mathrm{SGS}} = \overline{\rho u_i u_j} - \overline{\rho}\widetilde{u}_i\widetilde{u}_j$ and the SGS kinetic energy $k^{\mathrm{SGS}} = \tau_{kk}^{\mathrm{SGS}}$ of the form

$$\tau_{ij}^{\mathrm{SGS}} \overset{\mathrm{M}}{=} \overline{\rho}\sum_{\alpha} G_\alpha(I_1, I_2, \ldots) T_{ij}^{\alpha} \,, \tag{74a}$$

$$k^{\mathrm{SGS}} \overset{\mathrm{M}}{=} \overline{\rho}\sum_{\alpha} C_\alpha I_\alpha \,, \tag{74b}$$

were considered, with T_{ij}^{α} being the basis functions and I^{α} the invariants of the stress tensor. Gene expression programming (GEP) [573] was employed to determine the scalar coefficients G_α and C_α. Physical constraints and mathematical invariances were explicitly incorporated into the resulting ML models. Interestingly, this method was able to discover the functional form of tensor-diffusivity models that have been developed from physical arguments. This is illustrated in Fig. 30, showing quantitative comparisons of the GEP-derived models of the SGS turbulence kinetic energy for different LES-filter ratios and two algebraic SGS models [572]. These GEP-derived models provide results that are comparable to those obtained from Clark's tensor diffusivity model [574], emphasizing the ability of ML techniques to discover physics-based models for complex reacting flow environments.

Chung et al. [575] applied sparse symbolic regression in conjunction with feature selection via the random-forest feature importance score for discovering algebraic models of SGS stresses in transcritical non-premixed flames; velocity and its spatial derivatives were extracted from filtered DNS and used as features for random forests trained to predict SGS stresses. Feature importance scores were extracted from the random forests to identify potential candidate variables for sparse regression, which resulted in a derived model similar to Clark's formulation. This work demonstrates that interpretable supervised learning algorithms can generate insights in turbulent combustion modeling.

Yellapantula et al. [577] modeled the filtered and SGS dissipation rate of the reaction progress variable in turbulent premixed flames. Filtered data from DNS of a planar turbulent premixed flame under various conditions were used to train neural networks, and *a priori* tests were performed to examine the ability of a feedforward neural network to predict the filtered scalar dissipation rate over a range of filter widths (Fig. 31). While the ML model reproduced small-scale features for small filter ratios, increasing the filter width resulted in progressively increasing discrepancies in the predictions of the trained model (Fig. 31b).

CNNs and feedforward neural networks have been employed for parameterizing unresolved stresses, filtered source terms, and SGS transport of the reaction progress variable from filtered DNS data of premixed jet flames [578,579]. While these studies reported good agreement in modeling filtered DNS data, the parameterization with respect to primary quantities omits the consideration of hidden states, providing opportunities for future generalization to a broader class of flames and operating conditions.

In many of these studies, supervised learning algorithms were typically trained on filtered DNS data and then tested *a priori*. Without careful treatment, the ML SGS models can be unstable due to the accumulation of small errors over a large number of timesteps when evaluated *a posteriori* [559]. One way of dealing with this issue is with knowledge-guided ML (Section 3.5). For example, Bode et al. [279] employed a physics-informed enhanced super-resolution GAN (PIESRGAN) to model SGS stresses in both *a priori* and *a posteriori* *n*-dodecane spray flame simulations. Fig. 32 shows that a physics-informed loss term, which ensures continuity, is optimized alongside the adversarial and accuracy loss terms. This GAN architecture also maintains residual-residual dense block layers (RRDB), from its namesake, the ERSGAN [580], which has been designed to overcome stability issues [386] in classical GAN (Section 3.4.1) while ensuring high predictive

accuracy.

Complex coupling between the liquid phase and the gaseous flowfield in turbulent spray combustion is a major obstacle in developing first principles-based models. To addressing these challenges, Yao et al. [581], 582] employed feedforward neural networks to model the PDF and scalar dissipation rate in turbulent spray flames. Filtered results from a series of carrier-phase DNS with various operating conditions (equivalence ratio, temperature, and Stokes number) were used to train neural networks in order to represent the scalar dissipation rate in terms of an eight-dimensional feature-input set (mixture fraction, eddy viscosity, turbulent dissipation rate, molecular diffusion, density, droplet evaporation rates, slip velocity, and droplet number density). Beyond the network regression, feature-importance ranking was employed, indicating that only a few quantities are significant for the trained model. Information extracted from such data-driven methods is of broader significance as it can enable deeper understanding of the underlying physics and guide the formulation of physics-based models (Section 5.2).

Reduction of computation cost Another benefit of CombML is the resulting reduction in the computational cost of combustion simulations. A common approach is to replace the expensive computation of thermochemical properties and reaction rates with ML regression models (Section 4.1.3). An alternative strategy involves incorporating classification and clustering methods into simulations for alleviating computational bottlenecks within conventional modeling methods. For example, Liang et al. [583] proposed a dynamic cell clustering technique that incorporates clustering algorithms for accelerating multidimensional simulations of internal combustion engines. This clustering approach partitions the computational domain into multiple zones, each of which consists of its own set of temperatures and compositions; thus, only one detailed chemistry system of ODEs needs to be solved per zone in each global advancement timestep. The chemistry reduction problem can therefore be framed as a clustering problem that can be solved using *k*-means (Section 3.3.1). This approach was further improved by using clustering algorithms that are more suitable to spatial data such as the bounding-box *k*-means; for example, the KIVA simulations of Perini [584] achieved a 50% to 75% reduction in simulation cost with reasonable agreement with unclustered simulations [585] (Fig. 33). This

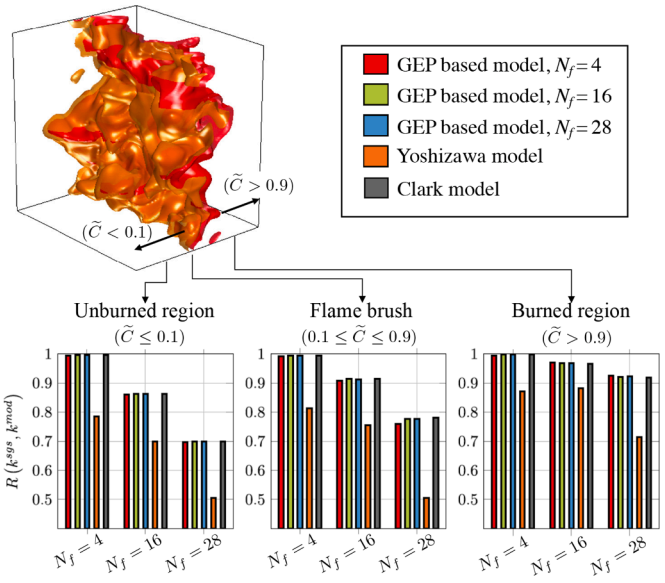


Fig. 30. Analysis of DNS results of a planar premixed turbulent flame in the thin reaction zones regime, showing comparisons of Pearson correlations for GEP-derived models and two algebraic models due to Yoshizawa [576] and Clark [574] for three LES filter ratios N_f. Adapted from [572], Copyright 2018, with permission from Elsevier.

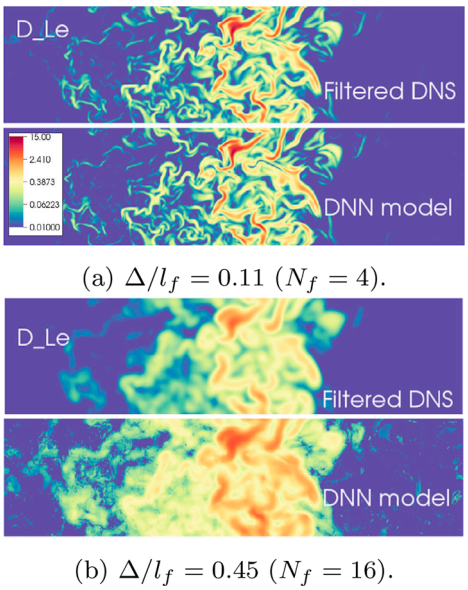


Fig. 31. Predictions of the normalized subgrid scalar dissipation rate of progress variable $\widetilde{\chi}_C/\chi_{\text{lam}}$ from a feedforward neural network for a lean premixed C_7H_{16}/air flame (Case D_{Le}), comparing results for two filter ratios N_f. Δ, filter size; l_f, laminar flame thickness. Reprinted from [577], Copyright 2020, with permission from Elsevier.

concept has also been successfully extended to particle-based PDF simulation methods [586].

Classification algorithms can also be applied to reduce the computational cost in finite-rate combustion simulations. By combining concepts from the Pareto-efficient combustion framework [381,587,588] and RANS uncertainty predictions [392], Chung et al. [589] used random forests (Section 3.2.3) to assign combustion submodels of different cost and fidelity to the same LES domain. Results showed a 20% reduction in the computational cost and good agreement with LES using finite-rate chemistry (Fig. 34). In this approach, feature-importance selection was used to determine controlling quantities that appropriately map combustion submodels to be consistent with the underlying flow representation. In addition, formulation as a classification problem using a random-forest model provides a high degree of interpretability and approximation errors made by the ML algorithm are limited by the predictive capability of the lowest-performing submodel. While the potential of this approach was established in a canonical burner configuration, applications to complex combustion configuration require the consideration of deep learning architectures for representing more extensive candidate combustion submodels, the direct control of solution errors, and the consideration of nonlinear and transient effects on submodel selection.

To reduce the computational cost of evaluating stiff chemical source terms in reacting-flow simulations, Lapointe et al. [590] developed a data-driven method for selecting a stiff-chemistry ODE solver in operator-splitting schemes. In this approach, feedforward neural networks were used as a classifier to select an optimal ODE solver that is employed locally to advance the stiff chemical-kinetic ODEs. These neural networks were trained from thermochemical states that represent 0D, partially stirred reactors and 1D laminar flames to predict the CPU time and error of the ODE solvers for a given thermochemical input state. During the simulation, the ODE solver with the lowest computational cost subject to user-specific tolerances is employed at each grid point and timestep. Benchmark tests and three-dimensional simulations determined that the data-driven selection of an optimal ODE integrator can provide speedups of more than a factor of three compared to default ODE integrators—without significant reduction in accuracy. Importantly, accurate network predictions of error and cost require training

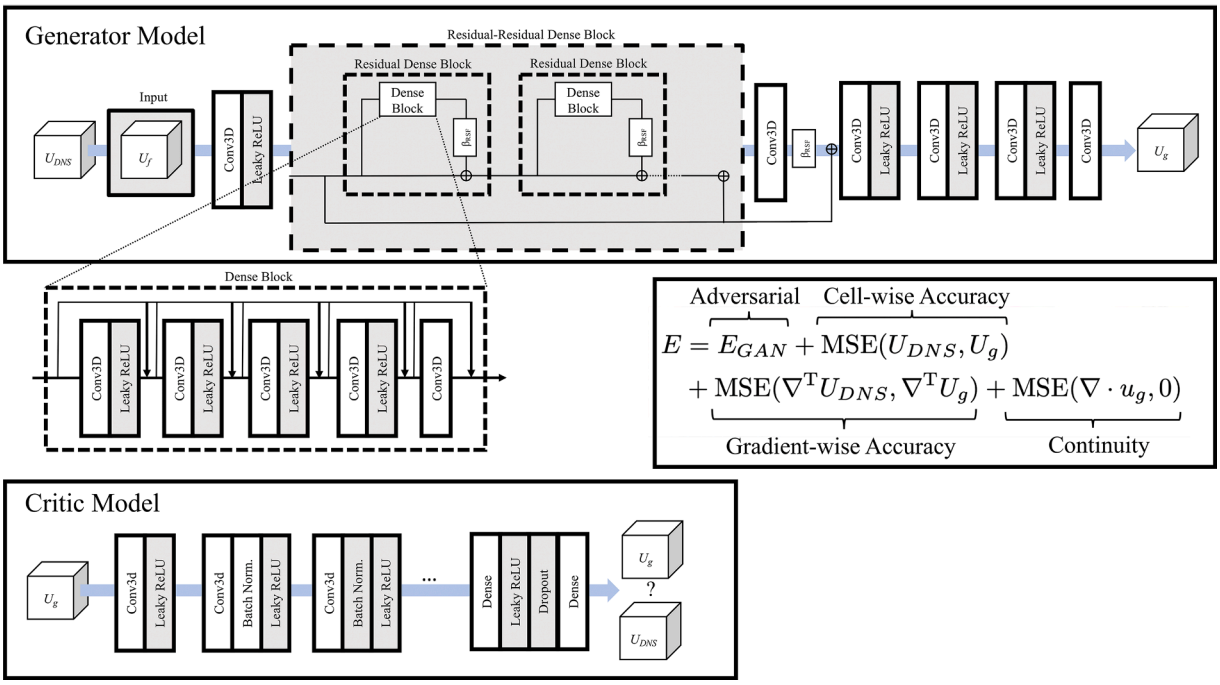


Fig. 32. Physics-Informed Enhanced Super Resolution Generative Adversarial Network (PIESRGAN) architecture. Adapted from [279] (CC BY 4.0).

with data that are generated from mechanisms and fuels considered in the simulation.

In summary, applications of supervised learning has produced encouraging results for modeling SGS terms and for parameterizing PDFs, often outperforming physics-based closure models in *a priori* studies. However, caution is warranted for extrapolating these findings to *a posteriori* applications, and further advances are necessary to examine the performance of these data-driven models that are typically generated for specific operating conditions in practical applications. Addressing the limitations of regression models, GANs and RL offers the ability to generate predictions for out-of-distribution conditions, and knowledge-guided ML provides avenues for incorporating physical principles and other fundamental consistencies that are necessary for enabling robust combustion simulations. Similarly, we have shown that deep learning methods are promising for combustion modeling; yet it remains unclear whether CNNs and RNNs are sufficient for representing highly complex thermochemical state relations in combustion simulations, or if bespoke and novel network architectures are needed to cope with the complexity encountered in CombML.

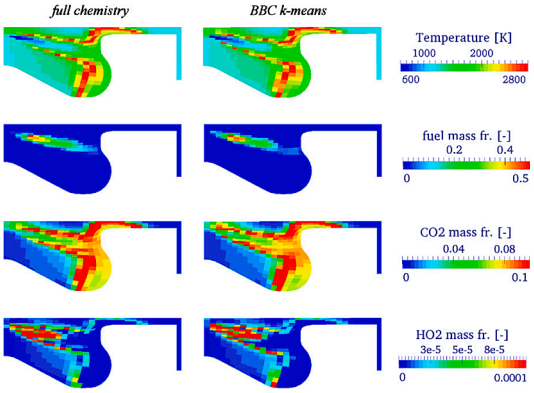


Fig. 33. Comparison of flow fields from KIVA simulations employing full chemistry and bounding-box clustering *k*-means. Reprinted from [584], Copyright 2013, with permission from Elsevier.

4.2. ML for propulsion and energy-conversion systems

In this section, we discuss applications of ML algorithms to propulsion and energy-conversion systems. Kalogirou [591] reviewed early adaptations of learning algorithms for modeling, controlling, and diagnosing power-generation systems and automotive engines. Since then, learning algorithms have been extended to broad areas, including system-level surrogate modeling (Section 4.2.1), intelligent fault detection (Section 4.2.3), data analytics (Section 4.2.2), and intelligent control (Section 4.2.4).

4.2.1. Surrogate models of system-level behavior

Surrogate models are representations of complex combustion-systems; they are often employed in combustor design, control, and calibration to circumvent costly high-fidelity simulations and experimental instrumentation. In contrast to the fundamental combustion investigation (Section 4.1) that consider low-dimensional combustion manifolds, the surrogate models that are examined here can be viewed as reduced-order models—metamodels [592]—of the system-level behavior of propulsion devices. This kind of surrogate modeling can be particularly useful in scenarios in which high-fidelity simulations and experimental testing require tremendous labor or computational resources, such as when optimizing combustion-system design and control parameters. This is a particularly rich area of research for automotive engines due to the relatively low cost of experimental setups and measurement techniques. In contrast, there are limited studies of surrogate modeling of gas turbines [593,594], furnaces, and power plants [595–597].

In traditional engine calibration, steady-state engine performance and emission metrics—gained through laborious experiments with varying control parameters implemented on a single engine—are stored as look-up tables within engine control units. Due to the low cost and acceptable accuracy of ML methods, early work on engine calibration sought to replace these engine-calibration tables with deterministic regression methods such as neural networks [598,599]. Since these neural networks capture nonlinearities more effectively than look-up tables, feedforward neural networks were determined to reduce the

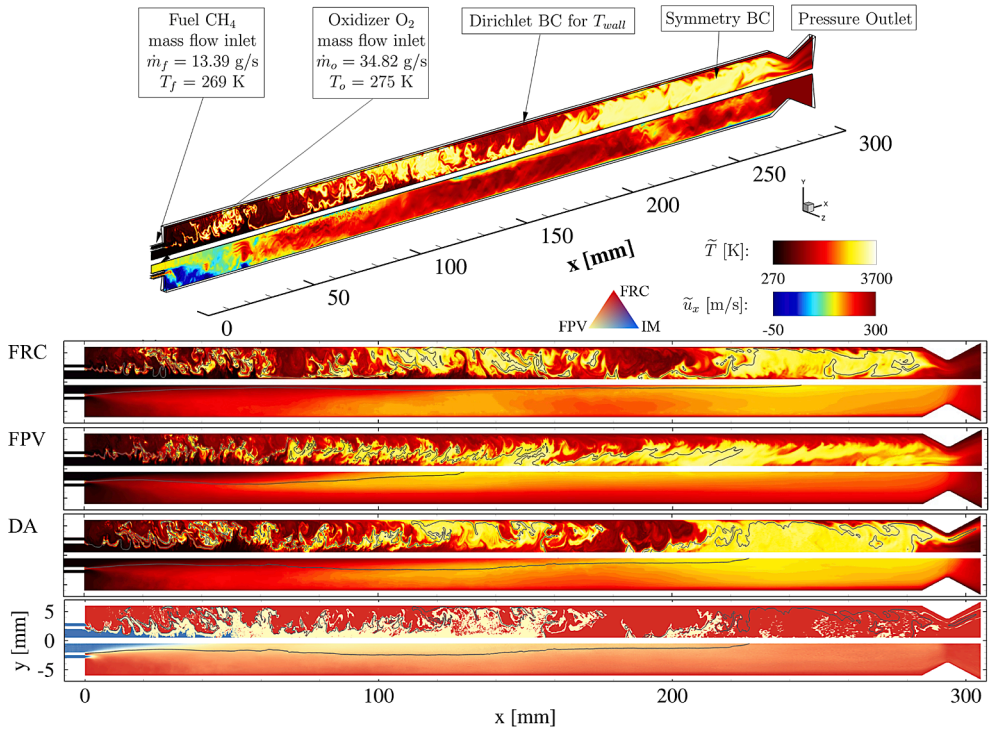


Fig. 34. Simulation of a rocket combustor, comparing instantaneous and time-averaged flow fields from monolithic finite-rate chemistry (FRC) simulation, a two-scalar flamelet/progress variable (FPV) model, and data-assisted (DA) LES, with DA-combustion submodel assignment at the bottom. Reprinted from [589], Copyright 2021, with permission from Elsevier.

number of engine measurements required to generate engine-calibration maps [600]. With the increasing popularity of biofuels and fuel additives, these ML algorithms are under investigation as a way to handle the added dimensionality in calibration tables that arises from variations in fuel composition [601,602]. These surrogate models could also be employed to improve other aspects of engine calibration, as highlighted by the use of SVMs to reduce the number of experimental measurements needed to explore novel optimization algorithms for engine calibration [603].

Different ML algorithms deliver different benefits for different objectives of engine surrogate modeling. Berger et al. [604] reported that, through their ability to quantify uncertainties, Bayesian ML methods such as Gaussian process regression can still be applied in order to generate more robust and trustworthy predictions. In the case of virtual sensors that process raw sensor outputs to deliver real-time quantities, surrogate models must accurately predict time-dependent quantities. While conventional feedforward neural networks have been employed [605,606], more suitable architectures such as RNNs were proposed as early as 2010 for modeling NO [607]. Another study [608] employed time-delay neural networks, applying a moving window on time histories to obtain input quantities in order to predict NO_x and smoke emissions.

However, since most engine surrogate models run on resource-limited edge devices such as engine control units, the search for low-cost algorithms has led to a significant research focus on the development of extreme learning machines (ELMs) [609,610]. These networks consist of a single hidden layer with weights that are randomly initialized, guaranteeing low computational complexity and avoiding the cost of iterative tuning. Since the datasets collected from engine experiments can be small, iterative hyperparameter tuning can be crucial for improving the prediction accuracy of engine surrogate models, as demonstrated in the study by Shamshirband et al. [611], comparing the performance of SVMs tuned by several distinct optimization algorithms.

In an investigation in which surrogate models were employed in order to optimize biofuel composition in an engine, Wong et al. [612] established that ELMs trained on sparse and small datasets could be more accurate than traditional ML methods such as SVMs. Vaughan and Bohac [613] proposed a novel ELM architecture—the weighted ring ELM—in order to accurately predict the combustion timing of a homogeneous charge compression engine by using engine parameters and recent updates from previous engine cycles as inputs. Kernel ELMs, which use a predefined kernel function to replace random initialization weights and biases, were investigated by Silitonga et al. [614] as a more reliable and stable alternative to conventional ELMs.

Many previous investigations focused on small datasets obtained from a single engine model. For example, an ELM engine model was trained on a dataset of 24 samples from a single engine [615]. As such, constructing a dataset for a benchmark engine case in order to objectively evaluate the aforementioned methods could be easily implemented (Section 5.1). However, this approach could be challenging for problems that require larger and/or more complex datasets, such as design optimization. For example, Moiz et al. [616] employed thousands of high-fidelity simulations of in-cylinder flow fields to train an ensemble ML approach (consisting of a blend of linear regression, neural networks, regression trees, and SVMs) in order to map engine design parameters and key engine outputs. In a later study, the same group [617] applied a GA optimization scheme to outputs from this engine surrogate modeling approach in order to optimize the piston bowl geometry of a compression ignition engine.

Outside of design optimization, CombML for engine surrogate modeling can be summarized as the development of cost-effective methods for edge devices, such as engine control units. In contrast to the fundamental applications discussed in Section 4.1, knowledge-guided ML can be challenging to implement since it is more difficult to embed physical constraints that can define entire systems. Instead, CombML methods that can quantify uncertainties (such as Bayesian ML) can be utilized to ensure robust behavior. Since some of these methods can be more computationally expensive than their deterministic counterparts, further developments will be needed for them to operate on edge devices.

4.2.2. Data processing and analysis for scientific discovery

As sensor and simulation technologies advance, datasets generated through measurements and simulations can become more complex—and thus more cumbersome to analyze. Under these conditions, unsupervised and semi-supervised approaches have become popular in aiding the analysis and processing of data for investigating of propulsion and energy systems. For example, Petrarolo et al. [618] used *k*-means for clustering flame images from a hybrid rocket combustor into several groups, generating insights into the corresponding short-term combustion dynamics. *K*-means clustering was also applied by Cao et al. [619] for aiding the analysis of large-scale coherent structures from velocity measurements in in-cylinder flows. Xiao et al. [620] employed a density-based clustering algorithm to identify aluminum agglomerates within solid propellant combustion data; their calculations employed the discrete element method. Analysis of aluminum agglomerates returned results that were consistent with experimental measurements. Nakaya et al. [621] determined that the Gaussian process latent variable method is superior to PCA for studying flame-vortex interactions, thermoacoustic instabilities, and transitional behaviors from high-speed images of an experimental setup.

Deep learning methods can be particularly convenient for scientific analysis due to their high accuracy and their ability to accept complex datasets without preprocessing. Wan et al. [623] directly fed images from Raman, Rayleigh, and CO laser-induced fluorescence scattering experiments of a multi-regime burner to identify combustion regimes with 85% accuracy. Liu et al. [622] applied a deep belief network, a deep learning method that automatically extracts nonlinear features, for developing a soft-sensor system from combustion images from a charge-coupled device camera in order to predict and monitor the oxygen content of a heavy fuel furnace in real time (Fig. 35). However, deep learning methods are typically opaque and provide little scientific insight, as intermediate processes in making a prediction automatically from raw data are difficult to analyze in complex architectures. Nevertheless, developments in interpretable deep learning suggest a promising path toward automatic scientific discovery. For example, the SciNet autoencoder structure [624] was shown to automatically extract physical laws from several physical sample problems.

Similar attempts in interpretable ML are gaining traction in combustion research. Barwey et al. [625] applied a CNN to map OH planar laser-induced fluorescence images with particle image velocimetry to reconstruct the velocity fields of a model gas-turbine combustor [626] under various operating conditions. This deep learning approach was accurate for predictions in attached flame regions but performance was insufficient in detached flame regions. Traditional interpretable learning algorithms such as random forests (Section 3.2.3) have also been used to identify important physical quantities that affect combustion phenomena, highlighting the benefits of transparent ML methods for scientific applications. Feature-importance scores were extracted from random forests to identify flame topologies and preignition quantities that relate to engine cycle-to-cycle variations from both experimental [627] and computational [628] databases.

4.2.3. Intelligent fault detection

Propulsion and energy systems typically consist of many components that are subject to faults that can arise from mechanical hardware failure, engine misfire, and thermoacoustic instabilities, among others. Traditional ML algorithms, especially feature-extraction and classification algorithms, can be used to detect system faults [629] through the procedures shown in Fig. 36. Before training, noisy experimental data are typically preprocessed using feature-extraction methods. These noisy experimental data commonly include time series of various measurements such as vibration, pressure, chemiluminescence, and thermocouple data. Supervised learning algorithms, in the form of classification methods, are trained to output the presence and type of faults. This approach was employed by Yadav and Kalra [630] who applied a Fourier transform scheme on automotive engine acoustic signals, which were then fed into a neural network in order to identify faults on various engine parts. A similar work [631] applied wavelet transform schemes to engine pressure signals, while another study [632] employed independent component analysis for the vibration signals of a marine diesel engine.

Studies comparing fault detection methods, such as that by Jafarian et al. [633] (which compares the effectiveness of *k*-nearest neighbors, neural networks, and SVMs as classifiers), can still generate insights into the effectiveness of these strategies. However, ML methods can have numerous hyperparameters, while the choice of feature-extraction methods can significant affect the performance of the detection system. As such, the evaluation and comparison of fault-detection strategies could benefit from an established benchmark dataset (Section 5.1). ML algorithms can still be selected based on specific user requirements. SVMs may offer superior generalization with small datasets typical of fault-detection problems [634]. As will be discussed in Section 5.3, Bayesian methods can provide uncertainty quantification alongside predictions, but they can be computationally expensive. In order to overcome this challenge, Wong et al. [635] proposed the use of sparse Bayesian ELMs, which inherit the low computational costs of conventional single-layered ELMs, in order to identify faults in a four-cylinder engine. More interpretable methods such as decision trees (Section 3.2.2) and random forests (Section 3.2.3) can provide insights through feature ranking, as demonstrated in several studies [636,637] of engine-misfire detection.

Neural networks are powerful tools for fault detection, especially with complex deep learning architectures. Convolutional and recurrent layers in these networks enable the direct use of complex datasets such as visual images and time-series data as feature sets. Kuzhagaliyeva et al. [638] reported that deep LSTM networks make accurate (75%) predictions of engine preignition when fed with time-series data extracted from low-cost pressure sensors. In the same work, 1D convolutional networks produced more accurate (79%) predictions when fed with datasets preprocessed via PCA.

Stacked autoencoders [639] and deep belief networks [640] learn directly from complex noisy raw data, removing the need for feature-extraction steps. Yan and Yu [641] used a deep stacked autoencoder network for combustor anomaly detection with a database of gas turbine thermocouple measurements. When used with two-dimensional convolutional networks, stacked autoencoders can also

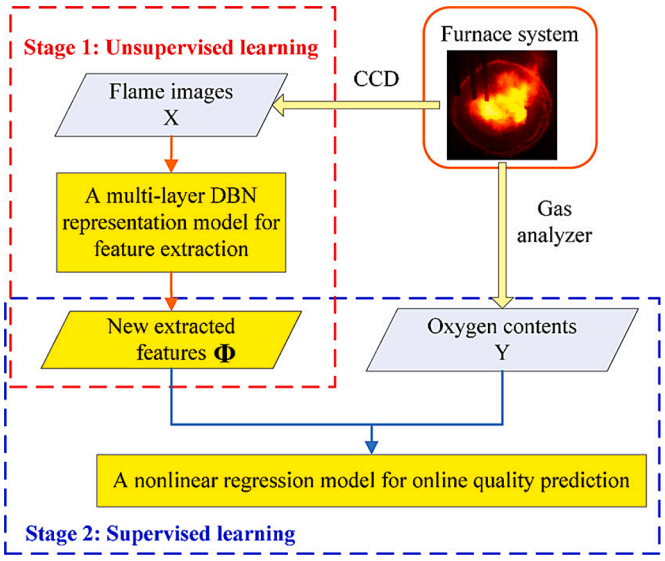


Fig. 35. Deep learning method incorporating a deep belief network (DBN) applied to predict and monitor oxygen content in real time with a charge-coupled device (CCD) camera in a heavy fuel furnace. Reprinted with permission from [622]. Copyright 2017 American Chemical Society.

be employed to monitor combustion in gas turbines and furnaces directly from flame images, as reported by Akintayo et al. [642] and Han et al. [643], respectively. These architecture are especially useful for challenges arising from the complex nature of real-world data, which tend to be poorly labeled or even unlabeled.

In summary, deep learning methods can offer more direct avenues for processing sensor data, without any prerequisite feature engineering steps. Fault detection systems are often deployed to manage failure of critical components. As such, any ML method used in production must be able to make robust predictions, either by quantifying uncertainties or through interpretable CombML. Note that fault-detection data are generated from limited failures in otherwise robust engineering systems, meaning there is a great deal of information about healthy systems but little about faulty systems. Traditional strategies [644] for dealing with these class imbalances involve augmenting the data, either through undersampling large classes or oversampling small classes, or by adding penalty to the loss functions. Class imbalance can also benefit from emerging concepts such as weakly supervised learning [645]. Further discussions on the application of CombML to the modeling of rare events pertaining to fire and explosion hazards, accidents and safety managements is given in Section 4.3.

4.2.4. *Intelligent control*

The intelligent control of combustion systems is commonly performed using model-free or model-based methods. RL methods are a particularly popular model-free method for optimally controlling nonlinear stochastic and deterministic problems.

Traditional RL is typically restricted to simulations and laboratory setups due to restrictions in processing multidimensional data, the computational cost of evaluating policies and value functions, and the inability to handle continuous data. Malikopoulos et al. [646] applied Q-learning with a discrete representation of the state-action space to control injection timing and turbocharger actuators that optimize key engine outputs (performance and emissions) in a diesel engine model. In order to circumvent these restrictions under practical and complex configurations that require continuous state-action spaces, neural networks can be used to approximate the state-action spaces. In particular, Schaefer et al. [647] explored various methods for combining RNNs with traditional RL methods such as Q-learning for stabilizing gas-turbine operations at high load conditions by considering 20 distinct operating points. Xue et al. [648] applied Q-learning, with three neural networks, on a physical engine test bench to optimize engine fuel economy during engine idle control.

The idea of combining function approximators (typically in the form of neural networks) to overcome the limitations of traditional RL has been explored within combustion systems as far back as 2001 [649]. By employing Q-learning with four neural network agents, Stephan et al. [649] developed one of the first RL control schemes; it controlled air distribution and air consumption, to be applied to a physical thermal plant. However, early RL control schemes employing neural networks were typically unstable and could not guarantee convergence in many practical applications [650]. Seminal work by Mnih et al. [354] is well-known for spurring a renewed interest in integrating neural networks with RL, resulting in a subset of methods known as deep RL. These methods differ from early works in their employment of complex deep learning architectures such as CNNs as well as their increased robustness from employing a replay buffer, which stores experiences from an RL agent over many episodes. Since these methods have only been developed recently, they have not been investigated extensively for combustion control. Cheng et al. [651] used a synchronous neural episodic control approach that employed CNNs and LSTM networks to consider 40 operating points in order to control air volume, fuel content, oxygen, and feedwater flow in a coal-fired boiler. Henry de Frahan et al. [652] presented the first work to apply deep RL for optimizing efficiency and emissions in an internal combustion engine. In contrast, model-based methods incorporate surrogate models to predict future behavior and adjust actuators accordingly. While stationary combustion systems are instrumented with sensors and their operation is augmented by computational models for online control with access to many learning algorithms, the control units of automotive engines were constrained by bandwidth, sensors, and compute power. As such, many demonstrations of complex algorithms for automotive engines are performed in laboratory settings that cannot yet be replicated in real-world automotive applications. While the development of custom hardware for deep learning algorithms shows promise for a new kind of control unit, the search for low-cost algorithms for engine control has led to a significant focus on low-cost approaches, such as the ELMs [609] discussed in Section 4.2.1 in the context of surrogate modeling of propulsion systems.

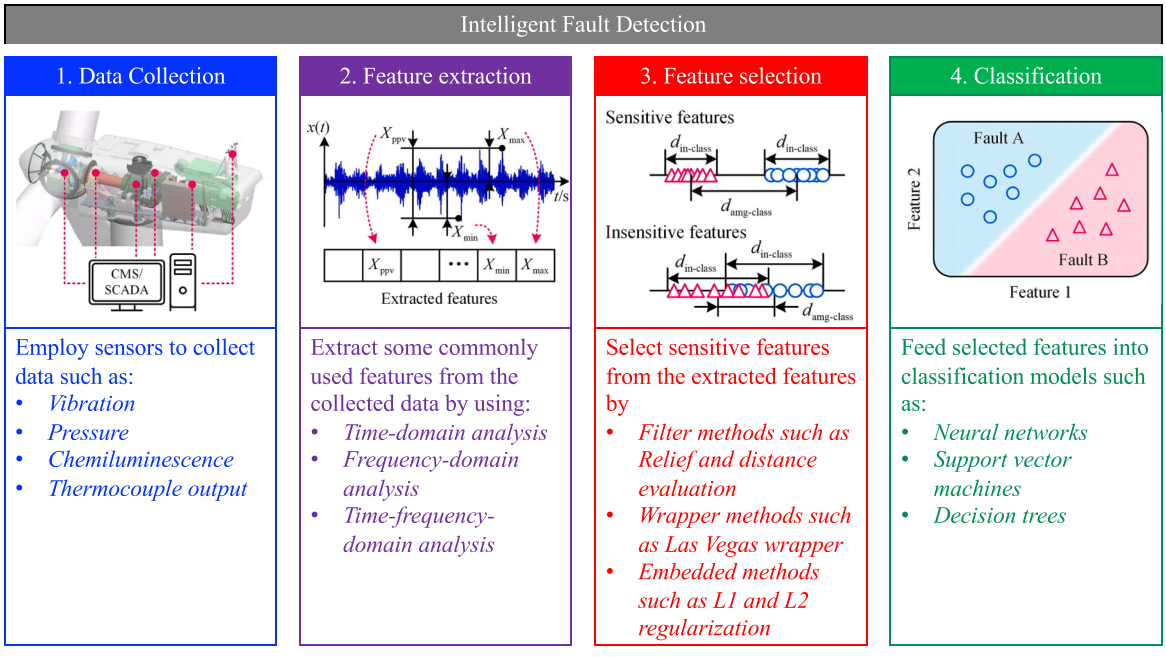


Fig. 36. Steps in machine-fault diagnosis. Adapted from [629], Copyright 2020, with permission from Elsevier.

In summary, robust intelligent control via RL has only become feasible for combustion experiments in recent years, due to the developments of deep RL. As such, these methods are still not feasible for commercial applications. This is largely caused by the lack of confidence in RL-control schemes under unseen conditions, especially in safety-critical applications, and by the large amount of online training required for current deep RL methods to converge. While the former can be addressed by interpretable ML, which would enable understanding of model behavior even in out-of-distribution conditions, the latter is currently being addressed by trends in offline RL [653], a RL paradigm that trains on previously collected data, without additional online data collection.

4.3. ML for fire and explosion hazards, accidents, and safety management

This section is concerned with discussing applications of ML methods to problems involving fire and explosion hazards, accidents, and safety management. Hazards describe conditions, combustible materials, and actions that might cause the onset of a fire or its increase in size and/or severity. If not attended, fire and explosion hazards can cause injuries, loss of life, property damage, and/or environmental impacts [654]. Examples include accidental gas explosions in fuel-storage systems and reactors [655,656], dust explosions in coal mines and grain elevators [657], wildfires [658–660], compartment fires [661–664], and fire and explosion accidents in fuel storage tanks [665], nuclear reactors [666], and batteries [667–669].

Compared to problems involving fundamental combustion investigations (Section 4.1) and energy-conversion systems (Section 4.2), key-distinguishing features are that these fire and explosion accidents are rare extreme events. These events are characterized by the inherently stochastic nature, complex dynamic response, and high intrinsic dimensionality [670]. Observations of fire and explosion accidents are often scarce and the available data are zero-heavy, nonstationary, and limited in spatiotemporal resolution [671]. Because of the consequences of fire and explosions accidents, reliable predictions impose unique challenges for CombML. In the following, we examine important aspects, specifically focusing on the ML application to predicting wildfire occurrence and wildfire dynamics (Section 4.3.1) and the modeling of rare explosion and fire events (Section 4.3.2).

4.3.1. Wildfire occurrence and wildfire-spread dynamics

Wildfires are rare events; however, they can have significant environmental and economic impacts. With the changing climate, the frequency and severity of wildfires are expected to increase [672], putting substantial stress on fire management and authorities to mitigate the risk of wildfires. To guide fire-preparedness and response actions, critical factors are predictions for fire occurrence, fire frequency, fire growth rate, size and intensity, and duration. The ignition of wildfires are caused by anthropogenic and natural processes. These ignition processes are inherently stochastic and ignition probabilities depend on ignition intensity, fuel properties, and the local environment. Logistic regression has been commonly used to predict ignition probabilities and the risk of wildfire occurrence due to their interpretability in probabilistic problems [671]. In contrast, the growth of the fire following the ignition can be described by physical and empirical models for fire-spread behavior [673–677]. The prediction of the fire-spread dynamics introduces other challenges pertaining to the lack of first-principles understanding of key combustion phenomena, the large variability in fuel composition, and the coupling to weather and topography. In the following, we discuss ML methods for predicting fire occurrence and fire-spread behavior.

Wildfire occurrence Physical models that are based on first principles for predicting wildfire-risk occurrence remain elusive, and supervised learning methods, such as logistic regression, neural networks, SVM, and other classification models (Section 3.2), have been widely adopted. These models consider dependencies on ignition potential, including fuel properties, weather, fire danger indices, lightning activities, topography, and anthropogenic effects [678]. A common approach is to model the spatiotemporal fire occurrence as a likelihood in which dependencies on covariates are represented by a conditional intensity function that is approximated by a Bernoulli probability of a fire occurrence [671,679]. Early investigations by Martell et al. [680] created a logistic model to predict the daily occurrence of anthropogenic forest fires using historical fire data that were collected over 17 years in Northern Ontario. Wotton and Martell [681] extended this work by accounting for effects of fuel-moisture content and other meteorological variables considering data from nearly 2.7×10^5 lightning strikes over 13 fire seasons in the Ontario fire management region. Analysis showed a regional dependence of the predictors with the duff moisture content and lightning strike being the most significant fire-ignition predictors. In order to consider nonlinear relationships between the probability of the fire occurrence and various explanatory variables, logistic generalized additive models were developed [679,682]. Preisler et al. [682] used observational data of 11 explanatory variables (including weather conditions, elevation, and fire danger indices) at a resolution of 1 km^2 to predict the spatiotemporal probability of the fire occurrence. Results from this model prediction in the region of Oregon are illustrated in Fig. 37, showing the observation of fire data (Fig. 37(a,b)) that were used for the model construction, and predictions of the probability of the fire ignition on two different days (Fig. 37(c,d)).

Apart from logistic regression models, other ML methods have been applied to fire-occurrence predictions, including neural networks [683–688], SVMs [687,689], and random forests [690,691]. The first application of feedforward neural networks for predicting anthropogenic wildfire occurrence in Alberta (Canada) was demonstrated by Vega-Garcia et al. [683]. In this study, network architectures with a single hidden layer and a varying number of neurons with up to 20 input features and two output channels were considered. Interestingly, comparisons with results from logistic models showed only marginal improvements of network predictions. This study also recognized early issues with the computational cost required for constructing neural-network models, limited interpretability compared to random forests and logistic regression models, and sensitivity to the network architecture. The impact of the network architecture on the accuracy for predicting monthly fire occurrence was examined by Dutta et al. [688] considering different networks, including feedforward, time-delay, and RNNs, showing that the latter provide improved predictions of spatial patterns of fire incidences. With relevance to the reliable prediction of fire occurrences for practical application to fire management, it was recognized that a critical step in the construction of these models is the evaluation of the goodness-of-fit and cross-validation to examine the model generalizability [678,692].

Wildfire-spread dynamics Computational models for predicting fire dynamics play a critical role for wildland fire management. Although physics-based models have supported the analysis of wildfire dynamics and prescribed fires [693], real-time predictions of large-scale fires and their behavior over several days largely rely on empirical or semi-empirical formulations. In these models, the rate of fire spread is commonly represented in algebraic or probabilistic form [674,675,694,695]:

$$s_\mathrm{r} = s_\mathrm{r}(u_\mathrm{w}, \alpha_\mathrm{w}, \psi_\mathrm{s}, \phi_\mathrm{F}, \ldots) \,, \tag{75a}$$

$$p_\mathrm{r} = p_\mathrm{r}(u_\mathrm{w}, \alpha_\mathrm{w}, \psi_\mathrm{s}, \phi_\mathrm{F}, \ldots) \,, \tag{75b}$$

where s_r is the rate of spread and p_r is a transition probability with dependencies on input variables for wind speed u_w, wind direction α_w, local slope of the terrain ψ_s, and model parameters ϕ_F that describe fuel properties such as fuel density, fuel category, heat content, fuel depth, and moisture content. These models have been constructed using laboratory measurements and field experiments that consider a relatively narrow range of conditions and specific fuel models [694,696].

However, because of difficulties in accurately characterizing fuel properties, variability in topography, and dynamically changing weather and environmental conditions over the course of the fire event, these models lack generalizability and are limited in their ability to predict realistic fires. To overcome these issues, data-driven approaches seek to correct for uncertainties by dynamically adjusting the input variables, physical models, and/or model parameters to maximize agreement between observations and predictions [697–699]. Learning of the input variables or model parameters is accomplished during a preprocessing stage or by assimilating observations in real time [700].

Abdalhaq et al. [701] and Rodríguez et al. [702], 703] developed a two-stage prediction method that combines a calibration stage with a prediction stage for dynamic wildfire simulations. A cellular automaton was used to describe the fire dynamics; values for the independent input data to the fire-spread model were learned during the calibration stage using a GA. For this task, a population of N individuals was created, each of which represented a particular set of input variables for wind speed, wind direction, slope, moisture, and vegetation. The fitness of each individual in this population was evaluated by computing the difference between the predicted fire map and the observations. Genetic operators for selection, crossover, and mutation were applied to generate a new population that was evolved for a specified number of generations. Parameters for the individual (or group of individuals) with the best fitness were used in the subsequent prediction stage. Observational data for fire perimeter and other information (such as topography, vegetation, or meteorology data from satellites, aerial imaging, or weather stations at discrete time intervals ranging from several minutes to several hours) were injected into the calibration stage to constrain input variables. This method can therefore be considered as a loosely coupled assimilation approach for state estimation in dynamic flow simulations in which unknown or uncertain input variables are continuously updated to capture observations [704,705]. Subsequent investigations extended this ML approach to real-time simulations by utilizing decision trees for run-time scheduling and real-time simulations [706–710] as well as

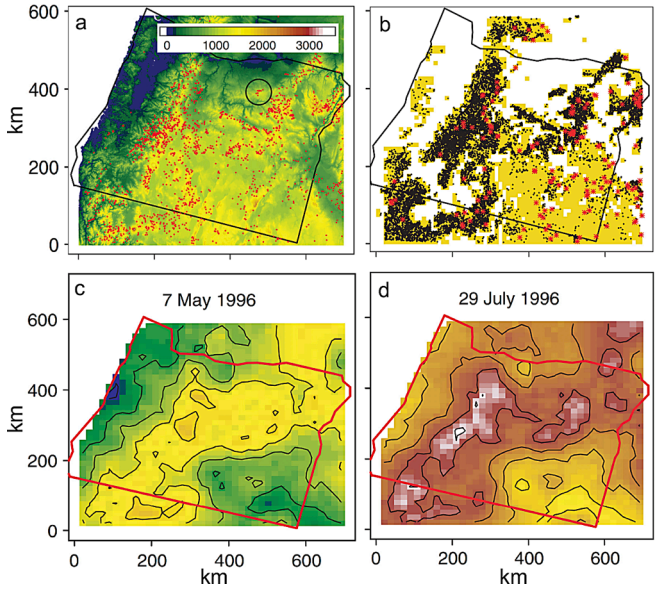


Fig. 37. Application of logistic generalized additive models to predict fire occurrence in the region of Oregon: (a) elevation map with red dots indicating locations of federal fires in 1996, (b) locations of all federal fires between 1989 and 1996 (black dots), yellow show federal lands, and red stars indicate locations of large fires (> 1000 ac), (c,d) predicted fire ignition probabilities showing distinct seasonal and spatial variability (probability increases from blue to green, red, and pink). Adapted from [682], Copyright 2004, with permission from CSIRO. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

considerations of uncertainties of the GA for fire-spread predictions [711,712]. Applications to synthetic data from realistic fire scenarios showed that this ML method improves predictions in the presence of dynamically changing environments, constituting a viable approach for augmenting missing or incomplete information in simulations. However, the dependency of this method on low-fidelity models, the decorrelation of time sequences, and the projection of latent processes on the input space limit application to practical problems.

Instead of regressing input variables from data, Ascoli et al. [713] employed GAs to learn unknown fuel-model parameters that describe fuel load, fuel density, fuel depth, heat content, and extinction moisture. This approach builds on prior work [714] in which GAs were used to estimate material properties from bench-scale experiments. Observations for wildfire-spread rate under various weather conditions, burn intensities, fuel moisture, and fuel mixtures were used to train and test a variety of fuel models that conformed to Rothermel's formulation [694]. The performance of these fuel models was evaluated using goodness-of-fit metrics for mean squared error, mean absolute error, mean bias error, and t-tests. Without extensive hyperparameter optimization, the GA used a population size of 50 individuals with 80% crossing probability, 10% mutation probability, and 5% elitism; up to 14 model parameters were represented in the feature set. Results from this study [713] showcased the viability of agent-based ML techniques for learning fuel-model parameters to improve fire-spread predictions. While this study focused on parameter calibration, it holds promise for learning entire fuel models via more advanced ML techniques.

Feedforward neural networks were employed by Chetehouna et al. [715] to map slope, wind speed, and fuel-moisture content to physical and topological fire parameters for rate-of-spread, flame height, and flame angle. A shallow network, consisting of one hidden layer with five neurons, was used to represent the data. The comparison of expressiveness against physical and semi-empirical models revealed similar performance. However, significant discrepancies arose when the network was used to extrapolate to new conditions. Further, the small dataset limited the ability of the network to learn complex state relations, which can lead to overfitting. As such, further improvements may be achieved by introducing physical constraints (Section 3.5) and aggregating data that span a wider range of conditions (Section 5.1) to make this regression task viable for realistic wildfire conditions.

Ntinas et al. [698] developed a data-driven approach that combines fuzzy logic with a differential evolution algorithm [467] to learn the transition functions in a cellular-automaton model from real wildfire observations, including the effects of fire suppression by firefighters. The transition function describing the state of each cell and its interaction with its neighborhood was described using fuzzy logic to account for fire intensity, wind, slope, and vegetation type. The parameters in this high-dimensional model space that encode the transfer functions were learned in a training process by advancing an ensemble of 10^5 individual model simulations over the time interval of an observed fire sequence. The fitness of the model was evaluated using Jaccard similarity [716], which measures the agreement between observed and predicted fire maps. Simulations of a wildfire occurrence during the 2004 drought season in the northern part of Sardinia, Italy [717] are shown in Fig. 38, illustrating a sequence of simulated burn areas over the course of the fire evolution. Comparisons of results from the data-driven model with observations of the fire perimeter (solid lines in Fig. 38) and benchmark results from a semi-empirical fire-simulator [696] and a cellular automaton [718] indicated substantial improvements by the ML model.

Instead of utilizing evolutionary optimization to incorporate knowledge from wildfire observations, Subramanian and Crowley [719], 720] tackled the problem of learning the local fire-spread rate as a solution of a Markov decision process. The basis of this model was a cellular automaton in which the fire was treated as an agent responding to its local environment. RL (Section 3.4.2) was used to learn the transition policy for advancing the fire in each cell over a prescribed time interval $\Delta\tau$, which typically spanned several hours. The reward function

was biased toward long-term reward for accurately reproducing the target observation at the end of the time interval. As such, this approach seeks to determine the integrated flame spread $\int_t^{t+\Delta\tau} p_r(\xi)d\xi$ to capture the fire perimeter at $t+\Delta\tau$, but does not reproduce the transient dynamics that is necessary for time-accurate fire-spread simulations. However, the flexibility of RL allows to incorporate physical constraints and conditions to make this approach viable for transient fire-spread predictions. Several RL algorithms were examined, including classical approaches (value iteration, policy iteration, and Q-learning), deep RL (asynchronous advantage actor-critic), and Monte Carlo tree search; the authors concluded that the latter two algorithms performed better for predicting intermediate and future fire spread than the other evaluated algorithms.

Supervised learning was used by Zheng et al. [721] to learn local transition rules in a cellular automaton model. In this method, the local ignition probability in each lattice cell was represented using an ELM. The ELM was trained using data from five historical fire sites that were mapped to the cellular-automaton lattice. Data on local vegetation and topography that were extracted from satellite measurements were used as inputs and the binary burned/unburned ignition probability was used as output. An analysis of the performance of this method uncovered overall good agreement with acceptable predictions for the burned area. In the future, extending the input-feature set to account for heterogeneous fuel composition, introducing physical knowledge into the ML model, and exploiting recent network developments should increase the generalizability and expressiveness of this approach.

The utilization of conceptually simple neural networks with shallow feedforward architectures has dominated supervised-learning applications in wildfires. While these models are robust and sufficiently flexible for dealing with the complexities of wildfire applications, recent advances in deep neural networks offer new opportunities for dealing with large areal image analysis, analyzing spatiotemporal dynamic behavior, and incorporating physical constraints and uncertainties into the model. In the rest of this section, we explore recent progress in utilizing deep neural networks for predicting wildfire behavior using RNNs, CNNs, LSTMs, and Bayesian neural networks (BNNs).

Kozik et al. [722] developed an ML-based wildfire model in which the fire behavior was represented by an RNN and a Kalman filter accelerated the learning process and accounted for uncertainties in the physical parameters, model representation, and observations. Unlike in cellular-automaton models, in this formulation the entire fire region was represented by a cylindrical arrangement of cells; each cell constituted a neuron. Along each polar direction, a sparsely connected RNN was constructed consisting of all neurons within an elliptic region (an "indicatrix") that contribute to the heat flux at the location of the fire. The total heat flux was then computed by integrating over all neuronal heat-flux contributions that were weighted by the activation function within this region. Unknown parameters that describe the geometry of the ellipsoid depend on the wind velocity, slope, and fuel properties and were determined from a sequence of successive observations. Uncertainties in the model and errors in the observed data were considered using a Kalman-filtering approach for parameter estimation during RNN training.

Instead of utilizing a Kalman filter for propagating uncertainties to the RNN model, Khakzad [723] utilized a dynamic Bayesian network. Instead of generating temporal dependencies between inputs and output states—as in RNNs—dynamic Bayesian networks provide probabilistic relationships. The fire-spread behavior in a wildland-industrial interface was modeled by representing the topography, vegetation, and storage tanks on a discrete lattice, which was mapped to a dynamic Bayesian network. Neighboring lattice cells were connected and the most probable fire pathways—and thereby the wildfire risk—were determined as the product of local ignition probability, fire-spread probability, and fire-intensity response. The probability for fire spread was modeled in analytic form, and daily forecast data from a fire-behavior model for

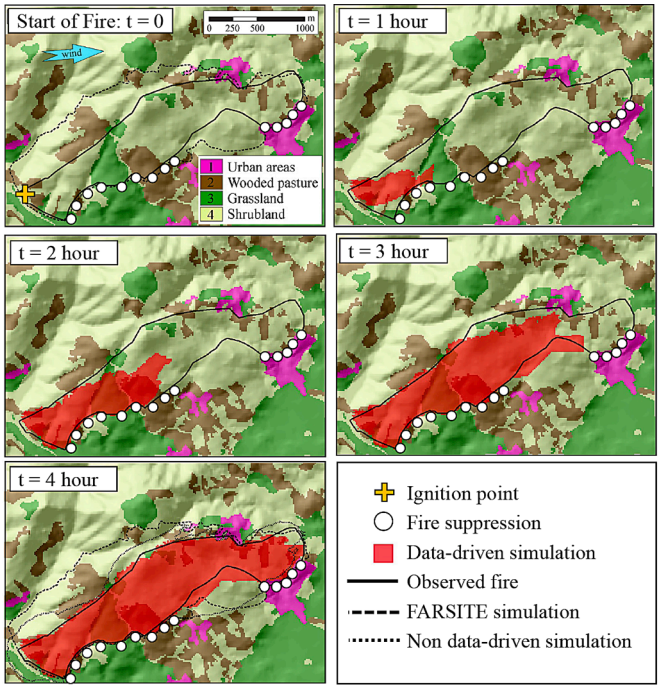


Fig. 38. Application of a data-driven method to simulate a wildfire sequence in northern Sardinia, Italy [717]. Upper left, vegetation, ignition location, and reported fire perimeter. Remaining panels, sequence of simulated burn areas using a data-driven model at the indicated time points. Also shown are comparisons with reference simulations from a semi-empirical fire simulator (FARSITE) and a cellular automaton with parametric transition function (non data-driven simulation) at the final state of the fire (lower left). Adapted from [698], Copyright 2017, with permission from Elsevier.

wind, rate of spread, and fire intensity were used to relate the model to representative weather conditions. Although this model invoked several simplifications for representing the fire-spread rate and discrete burning states, Bayesian approaches such as this one offer attractive opportunities for quantitative risk assessment by learning transition and ignition probabilities from observational data.

In the context of risk assessment, Radke et al. [724] employed CNNs to evaluate the ignition probability of regions around an existing fire perimeter as the fire advanced over 24 h. The CNN was trained with observations from historical data of fire perimeters (at 24-h intervals) and atmospheric data for pressure, temperature, wind speed/direction, precipitation, and humidity (at hourly intervals). The CNN model improved upon predictions of fire simulators, demonstrating the ability of ML methods to extract fire-spread patterns from wildfire observations. Limited observational data, extended intervals between consecutive measurements, and the need to consider topography, vegetation, and fuel properties suggest that further advances in these techniques are in store for risk assessment [725].

By addressing the computational cost of wildfire simulations, recent studies have investigated the feasibility of constructing ML models from data in order to represent the spatiotemporal evolution of wildfire behavior [724,726,727]. For example, Hodges and Lattimer [726] considered CNNs for simulating burn maps in 6-h time intervals over a 24-h wildfire event; the learning data were generated from a semi-empirical fire simulator [696]. The set of simulation data contained relevant complexities including heterogeneous topography, varying canopy, moisture content, wind, and fuel composition. Simulated burn maps and data fields were collected at time intervals of 6 h and were downsampled to images of 50×50 pixels at a spatial resolution of 1 km/pixel. The input state to the CNN consisted of 13 image channels representing fuel properties, moisture parameters, wind velocity, elevation, and the initial burn-map; the two output channels

described probabilities of burned and unburned states at the next time increment, from which the burn map after 6 h was constructed. A relatively complex CNN architecture was used consisting of six hidden layers: two convolution blocks, two downsampling stages, one fully connected layer, and one transpose convolution block. The relatively high dropout (a regularization method) and need for shuffling in order to mitigate overfitting suggest that hyperparameter optimization (Section 3.2.6) could improve network performance. Results for burn-map predictions at 6-h time increments were overall in good agreement with the simulation at substantially reduced run times. Deficiencies in the ability to capture small-scale features point to a need to extend this method for predicting localized ignition events associated with smoldering and/or spotting.

Instead of correlating data sequences at temporally segregated intervals, Burge et al. [727] trained a convolutional LSTM network for the time-accurate simulation of wildfire dynamics over a sequence of consecutive timesteps. Their ML model was trained on simulated wildfire data generated by a cellular automaton and a semi-empirical fire--spread model was employed to represent spatiotemporally varying heat accumulation. The computational domain was represented by 100×100 cells (with representative resolution of ~ 10 m/cell) and simulations were performed that cover a wide range of wildfire scenarios, considering dynamically changing wind, complex topography, spatially varying moisture, and realistic vegetation density. Extensive hyperparameter optimization was performed to identify an optimal architecture for predicting the advancement of the fire front at the next timestep, corresponding to a time interval of ~ 5 min. This model returned predictions of transient wildfire dynamics at four time instances for the active burn area and the burned region (Fig. 39). With advancing time, errors in predicting the correct flame location accumulated but remained localized over the entire fire sequence, which is representative of a 4–8 h fire-spread development (Fig. 39). Considering the complexity of this simulated fire, which is representative of realistic wildfire scenarios, these results underscore the potential of RNNs to learn dynamics encapsulated in empirical fire-spread models as well as future opportunities for application to dynamic fire predictions.

In summary, the abundance of observational data in conjunction with difficulties in constructing reliable physics-based models for wildfire behavior has led to remarkable forays in exploring various ML methods across wildfire applications. While significant progress has been made in employing ML for various learning tasks, embracing these techniques for practical applications remains limited due the lack of interpretability, the need for uncertainty quantification, and the ability of out-of-distribution predictions. Therefore, enormous opportunities arise for Bayesian deep learning methods that are guided by physical principles and available knowledge.

4.3.2. *Explosions, accidental fires, and rare events*

Explosions and accidental fires in compartments and enclosures are other areas where ML offers enormous opportunities for generating fundamental insight, developing improved models, and enabling quantitative risk assessment. However, major challenges for these CombML applications are the rarity of these events, dealing with sparse and incomplete observations, the high sensitivity to changing environmental conditions, and the complex chain of causal events that lead to these catastrophic outcomes. Because of these complexities, comparatively little work has been done on applying ML methods to explosions and accidental fires; advances in data-driven methods are needed to impact these areas of applications. In the following, we review recent progress on traditional data-driven ML application and discuss emerging system-dynamics based data-driven methods that utilize observational data for rare-event modeling.

An active area of interest in CombML is the explosion risk analysis in hydrogen fueling stations and nuclear reactors. This analysis involves two steps—the first being the utilization of descriptive models or observations to create data for various explosion scenarios; the second step is the construction of exceedance frequency maps using probabilistic analysis. In these applications, data-driven methods were largely employed for constructing response functions to describe explosion scenarios in order the reduce computational cost [728]. The consideration of uncertainties that arise from environmental conditions, ignition location, overpressure, and other parameters was only recently explored through the utilization of BNNs [729]. In this approach, BNNs were utilized to sample explosion scenarios and to guide the selection of simulation conditions to reduce model uncertainties. A particular strength of this Bayesian method is the control of model uncertainties, which is critical for reliable risk evaluations.

With relevance to the detection of precursor events that trigger instabilities, various approaches have been developed that combine dynamic system analysis with CombML [730,731]. The key idea of these approaches consists in analyzing continuous time sequences of observational quantities using well establish methods such as recurrent plots [732], complexity-entropy causality [733], or early warning criteria [734,735], and the application of supervised and unsupervised learning (such as SVM, ANNs, and *k*-means) to detect precursor events. So far, these methods have been demonstrated in applications to thermoacoustic instabilities, their extension offers promising opportunities for detecting precursors that control detonation and rare ignition events. Data-driven classification was employed to isolate features that demarcate decision boundaries for ignition and detonation [736]. In this work, logistic regression was applied to 485 data that are representative of different ignition configurations, including weak ignition in shock tubes, mild ignition in rapid compression machines, and cellular detonations, showing that only a small fraction of less than three parameters is sufficient to classify the data.

Further improving our ability to predict rare fire events and to detect precursors that trigger explosions is expected to benefit greatly from recent developments of data-driven methods that are constrained by dynamical principles [737–741]. In these methods, a time sequence of data is decomposed using time-delay embedding or other scale-separation methods to represent the time sequence in a high-dimensional state space that is partitioned to separate quasi-linear and non-linear dynamics. While these methods have been largely demonstrated in applications to idealized flow systems, they are also applicable to rare-event analysis of explosions and accidental fires.

5. Open research issues and opportunities

Section 4 reviewed applications of CombML techniques to a wide range of problems, including fundamental combustion investigations (Section 4.1), propulsion and energy-conversion systems (Section 4.2), and safety-critical problems pertaining to fire hazards and risk management (Section 4.3). While these applications have had initial success in solving scientific and engineering problems, several open research issues require addressing within the context of SciEngML and CombML [287]. Before proceeding with discussing these research issues, we identify several CombML research opportunities that will greatly benefit from advances in ML:

Digital twins and life-cycle management Digital twins are general concepts that integrate multiphysics, multiscale, probabilistic simulations of a combustion system using available physical models, sensor updates, and other information to mirror the behavior of its corresponding hardware counterpart [742]. In contrast to the component-level analysis that is typically targeted in reacting-flow calculations (Section 4.1), digital twins often describe entire engineering systems in order to monitor the behavior and health of a virtual counterpart under various parametric trajectories. As such, they offer new perspectives for enabling fuel-flexible operation of propulsion systems, responding to environmental changes, and the health monitoring of power-generation systems. In Section 4.1 and 4.2 we discussed the integration of CombML with both high-fidelity simulations and system-level modeling, respectively, which can result in accurate and

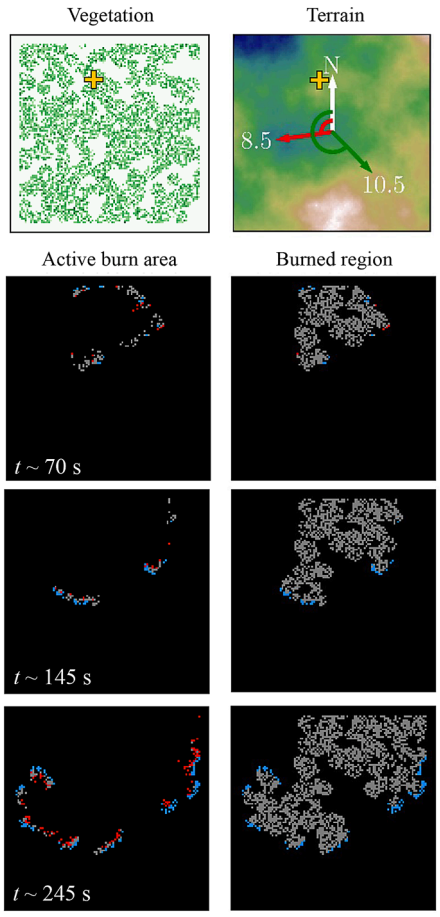


Fig. 39. Convolutional LSTM for predicting transient wildfire spread of a representative wildfire scenario. (top) Vegetation and terrain with changing wind direction (red/green arrows) and ignition point (cross). (bottom) Visualization of three predictions from a single fire sequence showing classification errors of the active burn area and the burned region. Red, false positives; blue, false negatives; black and gray, properly classified cells. Adapted from [727]. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

real-time simulations required for digital twins. Specific technologies relevant to anomaly detection and health monitoring were discussed in Section 4.2.3; the discussion of data-compression algorithms in Section 4.1.3 can also be broadly applied to the compression of sensor data and time-history data, which are typically integrated with simulation capabilities within the digital-twin paradigm. Given the maturity of various regression and classification techniques discussed here, it is within reach to assess various CombML approaches in order to compare strategies for these distinct aspects of digital-twin technologies through benchmark datasets (Section 5.1), if sufficiently complete datasets of specific combustion system could be assembled.

Discovery of sustainable and tailored fuel solutions As explored in Section 4.1.1 and 4.1.2, ML has enjoyed remarkable success in predicting fuel properties and improving chemical kinetic mechanisms. Addressing the need for carbon-free and renewable energy solutions, the investigation of alternative fuel sources has been a subjective of active research that would greatly benefit from CombML to integrate the vast domain knowledge and complex behaviors that arise from novel fuels, fuel additives, and biofuels. One interesting challenge is to develop new ML architectures that optimally handle the complex datasets in this field. Similar to the use of CNNs with spatially structured data and RNNs with sequential data (Section 3.2.4), the use of novel ML architectures such as graph neural networks has already demonstrated initial successful (Section 4.1.1) in incorporating chemical molecular structures as well as chemical relationships, offering new opportunities for the rapid screening of fuel candidates and the tailoring of fuels for specific applications.

Combustion control in challenging conditions As discussed in Section 4.2.4, traditional RL methods were popular for controlling combustion systems in the 2000s; now, deep RL offers a robust framework for intelligent control in many of these systems (Section 4.2.4). Examples include the stable operation of ultra-lean and low-emission combustion concepts, the rapid response to changing operating conditions in high-speed propulsion systems (such as scramjets, rotating detonation engines, and rocket motors), and the control of multiphase and high-pressure combustion systems. However, these methods can still be data-inefficient—they often require more than 10^5 training steps for convergence, which is too costly for many realistic control problems. While advances in deep RL and computing technology will eventually overcome these issues, the largest concern is likely the opacity and lack of interpretability (Section 5.2): the outputs of many learning algorithms can suffer the issue of poor trustworthiness due to difficulties in examining their internal mechanisms. As such, there is a critical need for interpretable CombML methods in these safety-critical control and risk-management applications.

5.1. Benchmark datasets and metrics

Section 1.1 provided an overview about the data that has been generated in various combustion fields, and Fig. 1 analyzed the amount of data that has been created from detailed numerical simulations alone. Currently, the infrastructure for accessing this data by the broader combustion community is still in its infancy. Beyond the need for developing this infrastructure, a key practice that our combustion community can emulate from fields with mature ML applications is the creation of benchmark datasets for a variety of combustion problems as well as a set of common, relevant metrics to evaluate CombML performance. For example, ML research in the field of high-energy physics has benefited tremendously from benchmark datasets [743,744]; initiatives have led to the creation and curation of new datasets [745] with community-wide standards for model evaluation. The availability of these data has encouraged researchers from outside the domain to investigate the data, often leading to novel and improved methodologies [746,747]. As another illustration, the proliferation of deep learning was aided by the performance of CNNs in the ImageNet large-scale visual recognition challenge [350,748]. Hence, the creation of well-documented and widely accessible benchmark datasets germane to critical challenges in the combustion sciences and engineering is expected to strongly promote the maturation of CombML applications. Collaborative efforts such as the Workshop on Turbulent Non-premixed Flames [4] and the Engine Combustion Network [749], which provide large corpora of well-documented experiments for model validation, can pave the way toward the effective usage of CombML methods that can lead to highly accurate reduced-order models (Section 4.2.1), improved data-driven combustion closures (Section 4.1.4) and the generation of new scientific insight (Section 4.2.2). In addition, government-established public databases such as the Co-optima Fuel Database [750], a central repository for data on the chemical properties of hundreds of neat fuels and fuel blends, will empower researchers to easily access a benchmark dataset for evaluating CombML methods for discovering the properties of novel fuel blends as well as for developing and optimizing chemical kinetic mechanisms for specific applications (Section 4.1.1 and 4.1.2). The necessity of a CombML database has been reflected throughout Section 4: while previous CombML applications have compared the performance of traditional ML algorithms such as SVMs, feedforward neural networks, and classification and regression trees, a fair comparison of accuracy and generalizability of novel and diverse modern deep learning-based architectures requires substantially more effort and can benefit immensely from a community-based effort.

5.2. Interpretability and explainability

Critical requirements for CombML are interpretability and explainability. Interpretability refers to the ability to comprehend the outcome of a model and to understand the causality between input and response [751,752]. Related to interpretability is explainability, which considers the causal relation of the response of the ML model to its internal architecture, operators, and weights. In traditional physics-based combustion models (Fig. 3a), each term in the model represents a specific physical mechanism or its interaction with other terms. Consequently, such models have a high degree of interpretability. In contrast, common data-driven models (Fig. 3b) attempt to infer relationships using complex cross-correlations inherent to the data. While simple models, such as linear or logistic regression, are inherently interpretable, complex models, such as deep neural networks, are not. The lack of interpretability may not be of concern for problems that bear low risk or are fully characterized. However, the ability to explain the behavior of a model is particularly relevant for combustion applications involving safety or reliability, such as ML-based fire-risk assessment (Section 4.3). In these cases, the interpretability of CombML models is crucial for controlling and monitoring critical engineering systems under potentially high-risk conditions. Apart from this direct practical relevance, interpretability is also valuable during model development and is critical for assessing model properties such as generalizability, fidelity, consistency, and stability [753].

Progress has been made in developing tools for evaluating interpretability by assessing the predictive and descriptive accuracies and relevance of ML models. Such tools can be model-specific (they may work for only a particular model class) or model-agnostic (they work across model types). A popular technique for deriving explanations from trained CNN models is layer-wise relevance propagation [754,755], which identifies important features according to a trained model via a backward pass through the architecture. The backward pass is a sequential relevance-redistribution procedure in which neurons that made the highest contribution to the succeeding layer are assigned the highest relevance score. When carried out backward until the input layer, this procedure assigns a relative importance to each pixel in the input. While layer-wise relevance propagation is a stable and effective technique for classification, it does not translate to regression problems very well.

Another useful approach is Shapley additive explanations [756], which is applicable to both classification and regression tasks. This approach estimates Shapley values [757] over features in order to derive explanations; it assumes that features are independent and that the model is locally linear. This formulation is similar to local interpretable model-agnostic explanations (LIME) [758], which utilizes a local surrogate model for the sample prediction under consideration. To date, these and other [752,756,758–760] techniques have found only limited application in CombML, creating opportunities for extending their applicability in order to facilitate the interpretability of CombML models.

5.3. Quantifying uncertainties of CombML models

By extending the discussion on data uncertainties in Section 2.6, CombML models exhibit various degrees of predictive uncertainties, which may arise due to inadequate or noisy training data, out-of-sample instances, inopportune choices of the model or hyperparameters, and/or the nature of the error minimization. These uncertainties are exacerbated by the lack of interpretability in complex CombML models (Section 5.2). These uncertainties can result in poor engineering decisions that may prove hazardous in safety and reliability-critical applications (Section 4.3). Since quantified uncertainties can act as direct measures for reliability, the assessment of model uncertainties is crucial for combustion systems in potentially hazardous conditions, for example digital twins and ML-based control algorithms.

Quantification of uncertainties requires probabilistic approaches in CombML applications. For simpler problems, approaches such as Bayesian regression provide accurate mean predictions along with estimates of the variability of predictions, which may be expressed as confidence or prediction intervals. Gaussian process regression-based modeling [761] is a popular approach for constructing nonparametric and interpretable models that provide mean predictions along with estimates of predictive uncertainty. While Gaussian process regression has been successfully applied to many scientific problems, a deficiency is their scaling with data volume. In Gaussian process models, the inference time grows cubically with the number of observations, as this requires the inversion of the dense covariance matrix. For complex tasks that require numerous training examples to learn from, this makes Gaussian processes computationally prohibitive [762]. To improve the scalability, approximate and hybrid methods have been proposed, including global approximations [763], utilizing sparse kernels [764], or using low-rank approximations via kernel interpolation [765].

Another promising alternative involves forming a Bayesian counterpart of ML algorithms by representing weights and biases as random variables with associated prior probability distributions, instead of representing these parameters deterministically. Upon gathering the data, these parameters are converted to the posterior distribution based on Bayes' theorem (Section 2.1). Most classical ML models, such as linear regression and logistic regression, have Bayesian counterparts—in this case Bayesian linear regression and Bayesian logistic regression. Bayesian Neural Networks (BNNs) are the Bayesian equivalent for deterministic neural networks [766,767] that combine the benefits of predictive accuracy of traditional deep learning models with the uncertainty estimation of probabilistic models. However, despite remarkable advances [768–770], persistent shortcomings of BNNs currently limit their application to complex problems. In particular, uncertainty estimates obtained from BNNs are strongly dependent on the selected inference algorithm. Additionally, the training time can be orders of magnitude longer than those of deterministic neural networks, often requiring the training of multiple networks [771].

Alternatively, non-probabilistic approaches can be used to quantify uncertainties. For instance, quantile loss-based methods have been used to account for aleatoric uncertainties in trained models. While quantile regression may be carried out with a variety of parametric [772] and non-parametric [773] models, their application in deep learning via quantile regression neural networks [774] is gaining popularity. Classical deterministic neural networks assume that the noise in the data is Gaussian and homoscedastic (Section 2.6). However, these assumptions may not be valid in combustion datasets, where the target data may be severely skewed or strictly bounded. In such cases, a robust alternative is to estimate the point predictions of different quantiles using sets of quantile neural networks.

At the current state of development, Gaussian process based models offer accuracy with respect to their mean predictions, reliable estimates of prediction uncertainty, along with robustness to out-of-distribution instances. Therefore, further exploration of these models offer interesting opportunities for CombML.

5.4. Evaluating out-of-distribution predictions

For restricted datasets, CombML models can be successful in applications involving interpolation. However, such deterministic models exhibit shortcomings for conditions outside the range of observations for which the model has been trained. When queried on such out-of-distribution samples, models are expected to extrapolate—rather than interpolate—within the range of training data. In the context of combustion-closure models (Section 4.1.4), a neural network-based model for predicting the turbulence/chemistry interaction may deteriorate substantially beyond the range of flow conditions in the training data, for flames that operate with different fuels, or if the model is applied to combustion regimes that are governed by different principles.

In such cases, complex models such as neural networks tend to make predictions that are often erroneous, with high predictive confidence [775,776].

One of the underlying causes of this lack of extrapolation for ML models (or their inability to generalize beyond their training data) is overfitting (Section 2.5). Owing to the greedy nature of the optimization process, the model attempts to mimic both the physics-based information and the conceivable artefacts in the data in order to reduce the error. This behavior may not be obvious in the model-evaluation phase if the testing data lie within the range of the training data. In such scenarios, model regularization is typically employed (Eq. (65)). Augmenting physical relations or conservation principles as soft constraints would aid in the formulation of knowledge-based models that can potentially generalize beyond their training data (Section 3.5).

An alternate strategy deals with determining the applicability of trained models for new samples on which to make predictions. Measures like the Wasserstein metric [243] or the Mahalonobis distance [392] can be employed to gauge the applicability of the trained model for new samples. Another approach to overcome limitations in data availability involves the use of generative methods (Section 3.4.1), such as GANs and variational autoencoders. These methods require less data as they are typically designed to learn from the probabilistic distributions of the training data rather than their individual instances, thus allowing for greater data efficiency. While these methods are typically used for generating synthetic data, studies [279,570] in turbulence modeling indicate improvements in generalizability, when using generative learning methods instead of supervised learning. Finally, one may rely on Bayesian models (Section 5.3), which progressively increase their predictive uncertainty as the samples venture beyond the range of the training data.

5.5. Integrating domain knowledge in CombML

Unlike empirical disciplines, combustion science is founded on physical concepts that encapsulate conservation, invariant principles, and reciprocity relations. Purely relying on traditional data-driven ML methods in combustion applications comes at the risk of omitting this scientific domain knowledge. In traditional ML applications, model tuning involves optimization of an error metric in target space that captures certain quantities of interest. Such modeling pipelines do not guarantee that the final model will adhere to physical constraints and conservation principles. As an example, the representation of individual species contained in a chemical mechanism through specialized neural networks provides benefits in representing highly non-linear reaction manifolds (Section 4.1.3). However, training individual networks based solely on a traditional loss metric may introduce issues in ensuring overall species and mass conservation. Therefore, effectively leveraging domain knowledge to complement ML methods is expected to lead to more reliable and robust CombML models that are computationally less expensive and engender a higher degree of trust for their deployment in combustion applications.

5.6. Computational complexity and accuracy

A factor often neglected in assessing CombML methods is the computational complexity of the ML algorithm, which is determined by the sample size N, the dimensionality of the feature space M, and the model complexity K (represented by the number of neurons and layers, support vectors, or hierarchical levels) and expressed as $\mathscr{O}(N^\alpha M^\beta K^\gamma)$ in which the exponents α, β, and γ depend on the particular algorithm employed [285,777]. Depending on the application, the complexity can change widely. For instance, when considering data from 3D simulations, the sample size is typically small ($N \sim \mathscr{O}(10)$) but the dimensionality of the feature space scales with the degrees of freedom and the dimension of the state vector so that $M \sim N_M N_U$ (Fig. 1c). In contrast, health monitoring of gas turbines or sampling the thermochemical state in a combustion system at discrete spatial locations results in a small feature space with $M \sim \mathscr{O}(10)$ but large samples ($N \sim \mathscr{O}(10^6)$). As such, iterative approaches for optimizing the performance of a CombML model and applying it to combustion problems can result in computational complexity that rapidly outgrows available resources; efficient learning algorithms are then necessary to expedite training and testing. This can be an important computational bottleneck for *in situ* calculations in high-fidelity simulations and for edge-computing devices used for control systems.

Related to complexity is the accuracy of the model in fitting the training data and generalizing to new data. Controlling accuracy is key in scientific and engineering simulations. More complex CombML models do not necessarily improve the accuracy because they are prone to overfitting in the presence of insufficient or noisy data. The relation among the size of the training data and a model's complexity and accuracy has been analyzed theoretically. For example, probably approximately correct learning is a framework for analyzing ML methods [778] that utilizes the VC dimension [323] (Section 2.5) as a measure of the expressive capacity of an ML model. Results from this analysis, although rather conservative for practical applications, provide bounds on the minimum data size for meeting certain model accuracies. Extending these results and connecting the concept of accuracy to CombML will be necessary in order to construct compact models and to assess computational complexity and data needs for meeting accuracy requirements in application to combustion science and engineering.

6. Summary

In this article, we have reviewed ML techniques for applications in combustion science and engineering. Beyond traditional applications for regression and dimensionality reduction, we explored the versatility of CombML to areas of control, optimization, discovery, and modeling. Many of these applications are currently explored through *a priori* tests and idealized problems. Successful adaptation of CombML techniques holds exciting promise for significantly advancing the current state of combustion science and engineering. Advances in the field of combustion have largely been made through physics-driven inquiry with a need for detailed understanding of the underlying physical mechanisms in conjunction with systematic model validation and uncertainty assessment. Incorporating these aspects into data-driven techniques is expected to lower barriers to broader adaptation to safety-critical combustion applications such as detonation, fire safety, emission control, and combustion stability.

We have highlighted opportunities and open research issues pertaining to the interpretability and explainability of CombML models, uncertainty quantification through the application of probabilistic and non-probabilistic approaches to data-driven models, and physics-constrained learning. Recent advances in these areas were discussed, as were tools for interpreting deep learning methods in various combustion applications.

Combustion science and engineering is a data-rich field. While substantial amounts of data have been accumulated, the collection and curation of these data in the form of benchmark data is expected to significantly accelerate the infusion of ML techniques into various areas related to combustion. In addition, best practices for sharing software, data pipelines, learning data, and procedures for model evaluations will benefit the greater CombML community.

While ML has demonstrated initial success in combustion science and engineering, even more exciting opportunities and breakthroughs should emerge by integrating combustion-domain knowledge and knowledge-guided methods into ML techniques. The holistic combination of data-driven methods with physical insights will impact all areas of combustion science and technology, ranging from knowledge discovery, data-assisted modeling and simulation techniques, *in situ* control

and optimization strategies, data-driven screening of alternative fuels, as well as applications to safety and risk assessment under consideration of uncertainties and safety margins.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We gratefully acknowledge financial support from AFOSR under award FA9550-21-1-0077, from the DOE Office of Basic Energy Sciences under award DE-SC0022222, from NASA under award NNX15AV04A, and from the DOE National Nuclear Security Administration under award DE-NA0003968. MI appreciates support from the DFG Mercator Fellowship (SPP1980) and the Alexander von Humboldt Foundation.

Supplementary material

The source codes and data for all example applications, discussed in Section 3, are available through a GitHub repository at: https://github.com/IhmeGroup/CombML_Tutorials.

The database of the DNS configurations that were analyzed in Fig. 1 is provided as supplementary material (`dnsDatabaseSummary.xlsx`) as an Excel spreadsheet. This spreadsheet contains relevant information, cataloged by combustion configuration, showing key characteristics for mesh resolution, chemical complexity, fuel, and number of parametric configurations studied, citation, and DOI. To provide this information in compact form, we report dimensionally averaged mesh sizes for DNS studies that were performed with different spatial resolutions. 2D simulation are indicated by omitting the third mesh-resolution (N_z).

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.pecs.2022.101010

References

[1] Chase Jr MW. NIST-JANAF Thermochemical tables (journal of physical and chemical reference data monographs). 4th. New York: American Chemical Society; 1998.
[2] Shen VK, Siderius DW, Krekelberg WP, Hatch HW. NIST standard reference simulation website, NIST standard reference database number 173. Tech Rep. National Institute of Standards and Technology; 2017.
[3] Ruscic B., Bross D.H.. Active thermochemical tables (ATcT). 2021. https://atct.anl.gov.
[4] Barlow R.S.. TNF workshop: International workshop on measurement and computation of turbulent flames. 1996. https://tnfworkshop.org.
[5] Driscoll JF, Chen JH, Skiba AW, Carter CD, Hawkes ER, Wang H. Premixed flames subjected to extreme turbulence: some questions and recent answers. Prog Energy Combust Sci 2020;76:100802.
[6] Miller JD, Slipchenko M, Meyer TR, Jiang N, Lempert WR, Gord JR. Ultrahigh-frame-rate OH fluorescence imaging in turbulent flames using a burst-mode optical parametric oscillator. Opt Lett 2009;34(9):1309–11.
[7] Garg S, Schadow K, Horn W, Pfoertner H, Stiharu I. Sensor and actuator needs for more intelligent gas turbine engines. Turbo Expo: Power for Land, Sea, and Air. 2010. p. 155–67.
[8] Vivekanandarajah A.. How airlines are flying high with aviation data analytics. 2018. https://seleritysas.com/blog/2018/11/17/flying-high-aviation-data-analytics/.
[9] Domek P.. Big data in aviation cleared for takeoff. 2019. https://spotlightvalley.com/big-data-aviation.
[10] Justice CO, Townshend JRG, Vermote EF, Masuoka E, Wolfe RE, Saleous N, Roy DP, Morisette JT. An overview of MODIS land data processing and product status. Remote Sens Environ 2002;83:3–15.
[11] Justice CO, Giglio L, Korontzi S, Owens J, Morisette JT, Roy D, Descloitres J, Alleaume S, Petitcolin F, Kaufman Y. The MODIS fire products. Remote Sens Environ 2002;83(1):244–62.
[12] Schroeder W, Oliva P, Giglio L, Csiszar IA. The new VIIRS 375 m active fire detection data product: algorithm description and initial assessment. Remote Sens Environ 2014;143:85–96.
[13] Justice CO, Román MO, Csiszar I, Vermote EF, Wolfe RE, Hook SJ, et al. Land and cryosphere products from suomi NPP VIIRS: overview and status. J Geophys Res Atmos 2013;118(17):9753–65.
[14] Schmit TJ, Griffith P, Gunshor MM, Daniels JM, Goodman SJ, Lebair WJ. A closer look at the ABI on the GOES-R series. Bull Am Meteorol Soc 2017;98(4):681–98.
[15] Roy DP, Wulder MA, Loveland TR, Woodcock CE, Allen RG, Anderson MC, et al. Landsat-8: science and product vision for terrestrial global change research. Remote Sens Environ 2014;145:154–72.
[16] Gorelick N, Hancher M, Dixon M, Ilyushchenko S, Thau D, Moore R. Google earth engine: planetary-scale geospatial analysis for everyone. Remote Sens Environ 2017;202:18–27.
[17] NASA Earthdata. Wildfire data toolkit. 2020. https://earthdata.nasa.gov/learn/toolkits/wildfires.
[18] Chen JH, Im HG. Stretch effects on the burning velocity of turbulent premixed hydrogen/air flames. Proc Combust Inst 2000;28:211–8.
[19] de Bruyn Kops SM, Riley JJ, Kosály G. Direct numerical simulation of reacting scalar mixing layers. Phys Fluids 2001;13(5):1450–65.
[20] Im HG, Chen JH. Effects of flow strain on triple flame propagation. Combust Flame 2001;126:1384–92.
[21] Jiang X, Luo KH. Direct numerical simulation of transitional noncircular buoyant reactive jets. Theor Comput Fluid Dyn 2001;15(3):183–98.
[22] Jiang X, Luo KH. Direct numerical simulation of the near field dynamics of a rectangular reactive plume. Int J Heat Fluid Flow 2001;22(6):633–42.
[23] Bell JB, Day MS, Grcar JF. Numerical simulation of premixed turbulent methane combustion. Proc Combust Inst 2002;29:1987–93.
[24] Echekki T, Chen JH. High-temperature combustion in autoigniting non-homogeneous hydrogen/air mixtures. Proc Combust Inst 2002;29:2061–8.
[25] Im HG, Chen JH. Preferential diffusion effects on the burning rate of interacting turbulent premixed hydrogen-air flames. Combust Flame 2002;131:246–58.
[26] Mizobuchi Y, Tachibana S, Shinio J, Ogawa S, Takeno T. A numerical analysis of the structure of a turbulent hydrogen jet lifted flame. Proc Combust Inst 2002;29:2009–15.
[27] Tanahashi M, Nada Y, Ito Y, Miyauchi T. Local flame structure in the well-stirred reactor regime. Proc Combust Inst 2002;29:2041–9.
[28] Cazan R, Menon S. Direct numerical simulation of sandwich and random-packed propellant combustion. AIAA Pap 2003–5082 2003.
[29] Echekki T, Chen JH. Direct numerical simulation of autoignition in non-homogeneous hydrogen-air mixtures. Combust Flame 2003;134:169–91.
[30] Jiang X, Luo KH. Dynamics and structure of transitional buoyant jet diffusion flames with side-wall effects. Combust Flame 2003;133:29–45.
[31] Lange M. Massively parallel DNS of flame kernel evolution in spark-ignited turbulent mixtures. In: Krause E. and Jäger W, editor. High Performance Computing in Science and Engineering '02. Springer; 2003. p. 425–38.
[32] Pantano C, Sarkar S, Williams FA. Mixing of a conserved scalar in a turbulent reacting shear layer. J Fluid Mech 2003;481:291–328.
[33] Tanahashi M, Nada Y, Tsukinari S, Saitoh T, Miyauchi T, Choi G. Local flame structure of turbulent premixed flames – DNS and CH/OH PLIF. Proc Symp Smart Control Turbul 2003;4:81–91.
[34] Vervisch L, Hauguel R, Domingo P. Direct numerical simulation (DNS) of premixed turbulent V-flames. AIAA Pap 2003–4497 2003.
[35] Bell JB, Day MS, Rendleman CA, Woosley SE, Zingale M. Direct numerical simulations of Type Ia supernovae flames. II. The Rayleigh–Taylor instability. Astrophys J 2004;608(2):883–906.
[36] Hawkes ER, Chen JH. Direct numerical simulation of hydrogen-enriched lean premixed methane-air flames. Combust Flame 2004;138:242–58.
[37] Lou H, Miller RS. On ternary species mixing and combustion in isotropic turbulence at high pressure. Phys Fluids 2004;16(5):1423–38.
[38] Mehravaran K, Jaberi FA. Direct numerical simulation of transitional and turbulent buoyant planar jet flames. Phys Fluids 2004;16(12):4443–61.
[39] Pantano C. Direct simulation of non-premixed flame extinction in a methane–air jet with reduced chemistry. J Fluid Mech 2004;514:231–70.
[40] Papalexandris MV. Numerical simulation of detonations in mixtures of gases and solid particles. J Fluid Mech 2004;507:95–142.
[41] Sripakagorn P, Mitarai S, Kosály G, Pitsch H. Extinction and reignition in a diffusion flame: Adirect numerical simulation study. J Fluid Mech 2004;518:231–59.
[42] Sutherland JC. Evaluation of mixing and reaction models for large-eddy simulation. The University of Utah; 2004. Ph.D. thesis.
[43] Vervisch L, Hauguel R, Domingo P, Rullaud M. Three facets of turbulent combustion modelling: DNS of premixed V-flame, LES of lifted nonpremixed flame and RANS of jet-flame. J Turbul 2004;5:N4.
[44] Viggiano A, Magi V. A 2-D investigation of *n*-heptane autoignition by means of direct numerical simulation. Combust Flame 2004;137:432–43.

[45] Bell JB, Day MS, Shepherd IG, Johnson MR, Cheng RK, Grcar JF, et al. Numerical simulation of a laboratory-scale turbulent v-flame. Proc Natl Acad Sci USA 2005;102(29):10006–11.
[46] Domingo P, Vervisch L, Payet S, Hauguel R. DNS of a premixed turbulent V flame and LES of a ducted flame using a FSD-PDF subgrid scale closure with FPI-tabulated chemistry. Combust Flame 2005;143:566–86.
[47] Domingo P, Vervisch L, Réveillon J. DNS analysis of partially premixed combustion in spray and gaseous turbulent flame-bases stabilized in hot air. Combust Flame 2005;140:172–95.
[48] Gashi S, Hult J, Jenkins KW, Chakraborty N, Cant S, Kaminski CF. Curvature and wrinkling of premixed flame kernels-comparisons of OH PLIF and DNS data. Proc Combust Inst 2005;30:809–17.
[49] Grcar JF, Glarborg P, Bell JB, Day MS, Loren A, Jensen AD. Effects of mixing on ammonia oxidation in combustion environments at intermediate temperatures. Proc Combust Inst 2005;30:1193–200.
[50] Hawkes ER, Chen JH. Evaluation of models for flame stretch due to curvature in the thin reaction zones regime. Proc Combust Inst 2005;30:647–55.
[51] Michioka T, Kurose R, Sada K, Makino H. Direct numerical simulation of a particle-laden mixing layer with a chemical reaction. Int J Multiph Flow 2005;31(7):843–66.
[52] Reveillon J, Vervisch L. Analysis of weakly turbulent dilute-spray flames and spray combustion regimes. J Fluid Mech 2005;537:317–47.
[53] Sutherland JC, Smith PJ, Chen JH. Quantification of differential diffusion in nonpremixed systems. Combust Theory Model 2005;9(2):365–83.
[54] Sankaran R, Im HG, Hawkes ER, Chen JH. The effects of non-uniform temperature distribution on the ignition of a lean homogeneous hydrogen-air mixture. Proc Combust Inst 2005;30:875–82.
[55] Thévenin D. Three-dimensional direct simulations and structure of expanding turbulent methane flames. Proc Combust Inst 2005;30:629–37.
[56] van Oijen JA, Bastiaans RJM, Groot GRA, de Goey LPH. Direct numerical simulations of premixed turbulent flames with reduced chemistry: Validation and flamelet analysis. Flow Turbul Combust 2005;75:67–84.
[57] Wang Y. Direct numerical simulation of non-premixed combustion with soot and thermal radiation. University of Maryland; 2005. Ph.D. thesis.
[58] Wang Y, Rutland CJ. DNS Study of the ignition of *n*-heptane fuel spray under high pressure and lean conditions. J Phys Conf Ser 2005;16:124–8.
[59] Wu Y, Haworth DC, Modest MF, Cuenot B. Direct numerical simulation of turbulence/radiation interaction in premixed combustion systems. Proc Combust Inst 2005;30:639–46.
[60] Zingale M, Woosley SE, Rendleman CA, Day MS, Bell JB. Three–dimensional numerical simulations of Rayleigh–Taylor unstable flames in Type Ia supernovae. Astrophys J 2005;632:1021–34.
[61] Mizobuchi Y, Shinjo J, Ogawa S, Takeno T. A numerical study on the formation of diffusion flame islands in a turbulent hydrogen jet lifted flame. Proc Combust Inst 2005;30:611–9.
[62] Bell JB, Day MS, Almgren AS, Lijewski MJ, Rendleman CA, Cheng RK, Shepherd IG. Simulation of lean premixed turbulent combustion. J Phys Conf Ser 2006;46:1–15.
[63] Chen JH, Hawkes ER, Sankaran R, Mason SD, Im HG. Direct numerical simulation of ignition front propagation in a constant volume with temperature inhomogeneities: I. Fundamental analysis and diagnostics. Combust Flame 2006;145:128–44.
[64] Hawkes ER, Chen JH. Comparison of direct numerical simulation of lean premixed methane-air flames with strained laminar flame calculations. Combust Flame 2006;144:112–25.
[65] Hawkes ER, Sankaran R, Pébay PP, Chen JH. Direct numerical simulation of ignition front propagation in a constant volume with temperature inhomogeneities: II. Parametric study. Combust Flame 2006;145:145–59.
[66] Sankaran R, Hawkes ER, Chen JH, Lu T, Law CK. Direct numerical simulations of turbulent lean premixed combustion. J Phys Conf Ser 2006;46:38–42.
[67] Bell JB, Cheng RK, Day MS, Shepherd IG. Numerical simulation of Lewis number effects on lean premixed turbulent flames. Proc Combust Inst 2007;31:1309–17.
[68] Bell JB, Day MS, Grcar JF, Lijewski MJ, Driscoll JF, Filatyev SA. Numerical simulation of a laboratory-scale turbulent slot flame. Proc Combust Inst 2007;31:1299–307.
[69] Deshmukh KV, Haworth DC, Modest MF. Direct numerical simulation of turbulence-radiation interactions in homogeneous nonpremixed combustion systems. Proc Combust Inst 2007;31:1641–8.
[70] Hawkes ER, Sankaran R, Sutherland JC, Chen JH. Scalar mixing in direct numerical simulations of temporally evolving plane jet flames with skeletal CO/H_2 kinetics. Proc Combust Inst 2007;31:1633–40.
[71] Lignell DO, Chen JH, Smith PJ, Lu T, Law CK. The effect of flame structure on soot formation and transport in turbulent nonpremixed flames using direct numerical simulation. Combust Flame 2007;151:2–28.
[72] Sankaran R, Hawkes ER, Chen JH, Lu T, Law CK. Structure of a spatially developing turbulent lean methane-air Bunsen flame. Proc Combust Inst 2007;31:1291–8.
[73] Aspden AJ, Bell JB, Day MS, Woosley SE, Zingale M. Turbulence–flame interactions in Type Ia supernovae. Astrophys J 2008;689(2):1173–85.
[74] Bell JB, Cheng RK, Day MS, Beckner VE, Lijewski MJ. Interaction of turblence and chemistry in a low-swirl burner. J Phys Conf Ser 2008;125:012027.
[75] Chakraborty N, Hawkes ER, Chen JH, Cant RS. The effects of strain rate and curvature on surface density function transport in turbulent premixed methane-air and hydrogen-air flames: A comparative study. Combust Flame 2008;154:259–80.
[76] Lignell DO, Chen JH, Smith PJ. Three-dimensional direct numerical simulation of soot formation and transport in a temporally evolving nonpremixed ethylene jet flame. Combust Flame 2008;155:316–33.
[77] Tanahashi M, Sato M, Shimura M, Miyauchi T. DNS and combined laser diagnostics of turbulent combustion. J Therm Sci Technol 2008;3(3):391–409.
[78] Bisetti F, Chen JY, Chen JH, Hawkes ER. Differential diffusion effects during the ignition of a thermally stratified premixed hydrogen-air mixture subject to turbulence. Proc Combust Inst 2009;32:1465–72.
[79] Chen JH, Choudhary A, de Supinski B, DeVries M, Hawkes ER, Klasky S, Liao WK, Ma KL, Mellor-Crummey J, Podhorszki N, Sankaran R, Shende S, Yoo CS. Terascale direct numerical simulations of turbulent combustion using S3D. Comput Sci Discov 2009;2(1):015001.
[80] Day MS, Bell JB, Cheng RK, Tachibana S, Beckner VE, Lijewski MJ. Cellular burning in lean premixed turbulent hydrogen-air flames: Coupling experimental and computational analysis at the laboratory scale. J Phys Conf Ser 2009;180:012031.
[81] Day M, Bell J, Bremer P-T, Pascucci V, Beckner V, Lijewski M. Turbulence effects on cellular burning structures in lean premixed hydrogen flames. Combust Flame 2009;156:1035–45.
[82] Grcar JF, Bell JB, Day MS. The Soret effect in naturally propagating, premixed, lean, hydrogen-air flames. Proc Combust Inst 2009;32:1173–80.
[83] Lee UD, Yoo CS, Chen JH, Frank JH. Effects of h_2o and NO on extinction and re-ignition of vortex-perturbed hydrogen counterflow flames. Proc Combust Inst 2009;32:1059–66.
[84] Lignell DO, Hewson JC, Chen JH. A-priori analysis of conditional moment closure modeling of a temporal ethylene jet flame with soot formation using direct numerical simulation. Proc Combust Inst 2009;32:1491–8.
[85] Lu T, Law CK, Yoo CS, Chen JH. Dynamic stiffness removal for direct numerical simulations. Combust Flame 2009;156:1542–51.
[86] Yoo CS, Sankaran R, Chen JH. Three-dimensional direct numerical simulation of a turbulent lifted hydrogen jet flame in heated coflow: Flame stabilization and structure. J Fluid Mech 2009;640:453–81.
[87] Yoo CS, Chen JH, Frank JH. A numerical study of transient ignition and flame characteristics of diluted hydrogen versus heated air in counterflow. Combust Flame 2009;156:140–51.
[88] Chakraborty N, Swaminathan N. Effects of Lewis number on scalar dissipation transport and its modeling in turbulent premixed combustion. Combust Sci Tech 2010;182:1201–40.
[89] Chakraborty N, Rogerson JW, Swaminathan N. The scalar gradient alignment statistics of flame kernels and its modelling implications for turbulent premixed combustion. Flow Turbul Combust 2010;85:25–55.
[90] Gruber A, Sankaran R, Hawkes ER, Chen JH. Turbulent flame-wall interaction: A direct numerical simulation study. J Fluid Mech 2010;658:5–32.
[91] Kerkemeier SG. Direct numerical simulation of combustion on petascale platforms: Applications to turbulent non-premixed hydrogen autoignition. ETH Zurich; 2010. Ph.D. thesis.
[92] Lee D, Huh KY. Statistically steady incompressible DNS to validate a new correlation for turbulent burning velocity in turbulent premixed combustion. Flow Turbul Combust 2010;84:339–56.
[93] Lee UD, Yoo CS, Chen JH, Frank JH. Effect of NO on extinction and re-ignition of vortex-perturbed hydrogen flames. Combust Flame 2010;157:217–29.
[94] Lu T, Yoo CS, Chen JH, Law CK. Three-dimensional direct numerical simulation of a turbulent lifted hydrogen jet flame in heated coflow: A chemical explosive mode analysis. J Fluid Mech 2010;652:45–64.
[95] Malkeson SP, Chakraborty N. A priori direct numerical simulation assessment of algebraic models of variances and dissipation rates in the context of Reynolds-averaged Navier-Stokes simulations for low Damköhler number partially premixed combustion. Combust Sci Tech 2010;182:960–99.
[96] Neophytou A, Mastorakos E, Cant RS. DNS of spark ignition and edge flame propagation in turbulent droplet-laden mixing layers. Combust Flame 2010;157:1071–86.
[97] Poludnenko AY, Oran ES. The interaction of high-speed turbulence with flames: Global properties and internal flame structure. Combust Flame 2010;157:995–1011.
[98] Xia J, Luo KH. Direct numerical simulation of inert droplet effects on scalar dissipation rate in turbulent reacting and non-reacting shear layers. Flow Turbul Combust 2010;84:397–422.
[99] Yu H, Wang C, Grout RW, Chen JH, Ma KL. In situ visualization for large-scale combustion simulations. IEEE Comput Graph Appl 2010;30:45–57.
[100] Aspden AJ, Day MS, Bell JB. Characterization of low Lewis number flames. Proc Combust Inst 2011;33:1463–71.
[101] Aspden AJ, Day MS, Bell JB. Lewis number effects in distributed flames. Proc Combust Inst 2011;33:1473–80.
[102] Aspden AJ, Day MS, Bell JB. Turbulence-flame interactions in lean premixed hydrogen: transition to the distributed burning regime. J Fluid Mech 2011;680:287–320.
[103] Day MS, Bell JB, Gao X, Glarborg P. Numerical simulation of nitrogen oxide formation in lean premixed turbulent $H_2/O_2/N_2$ flames. Proc Combust Inst 2011;33:1591–9.
[104] Day MS, Gao X, Bell JB. Properties of lean turbulent methane-air flames with significant hydrogen addition. Proc Combust Inst 2011;33:1601–8.
[105] Grout RW, Gruber A, Yoo CS, Chen JH. Direct numerical simulation of flame stabilization downstream of a transverse fuel jet in cross-flow. Proc Combust Inst 2011;33:1629–37.
[106] Hamlington PE, Poludnenko AY, Oran ES. Interactions between turbulence and flames in premixed reacting flows. Phys Fluids 2011;23(12):125111.

[107] Hawkes ER, Sankaran R, Chen JH. Estimates of the three-dimensional flame surface density and every term in its transport equation from two-dimensional measurements. Proc Combust Inst 2011;33:1447–54.
[108] Lignell DO, Chen JH, Schmutz HA. Effects of Damköhler number on flame extinction and reignition in turbulent non-premixed flames using DNS. Combust Flame 2011;158:949–63.
[109] Moureau V, Domingo P, Vervisch L. From large-eddy simulation to direct numerical simulation of a lean premixed swirl flame: Filtered laminar flame-PDF modeling. Combust Flame 2011;158:1340–57.
[110] Poludnenko AY, Gardiner TA, Oran ES. Spontaneous transition of turbulent flames to detonations in unconfined media. Phys Rev Lett 2011;107:054501.
[111] Tanaka S, Shimura M, Fukushima N, Tanahashi M, Miyauchi T. DNS of turbulent swirling premixed flame in a micro gas turbine combustor. Proc Combust Inst 2011;33:3293–300.
[112] Yoo CS, Lu T, Chen JH, Law CK. Direct numerical simulations of ignition of a lean *n*-heptane/air mixture with temperature inhomogeneities at constant volume: parametric study. Combust Flame 2011;158:1727–41.
[113] Yoo CS, Richardson ES, Sankaran R, Chen JH. A DNS study on the stabilization mechanism of a turbulent lifted ethylene jet flame in highly-heated coflow. Proc Combust Inst 2011;33:1619–27.
[114] Bisetti F, Blanquart G, Mueller ME, Pitsch H. On the formation and early evolution of soot in turbulent nonpremixed flames. Combust Flame 2012;159:317–35.
[115] Day M, Tachibana S, Bell J, Lijewski M, Beckner V, Cheng RK. A combined computational and experimental characterization of lean premixed turbulent low swirl laboratory flames: i. methane flames. Combust Flame 2012;159:275–90.
[116] Grout RW, Gruber A, Kolla H, Bremer P-T, Bennett JC, Gyulassy A, Chen JH. A direct numerical simulation study of turbulence and flame structure in transverse jets analysed in jet-trajectory based coordinates. J Fluid Mech 2012; 706:351–83.
[117] Gruber A, Chen JH, Valiev D, Law CK. Direct numerical simulation of premixed flame boundary layer flashback in turbulent channel flow. J Fluid Mech 2012; 709:516–42.
[118] Hamlington PE, Poludnenko AY, Oran ES. Intermittency in premixed turbulent reacting flows. Phys Fluids 2012;24(7):075111.
[119] Hawkes ER, Chatakonda O, Kolla H, Kerstein AR, Chen JH. A petascale direct numerical simulation study of the modelling of flame wrinkling for large-eddy simulations in intense turbulence. Combust Flame 2012;159:2690–703.
[120] Knudsen E, Richardson ES, Doran EM, Pitsch H, Chen JH. Modeling scalar dissipation and scalar variance in large eddy simulation: Algebraic and transport equation closures. Phys Fluids 2012;24(5):055103.
[121] Kolla H, Grout RW, Gruber A, Chen JH. Mechanisms of flame stabilization and blowout in a reacting turbulent hydrogen jet in cross-flow. Combust Flame 2012; 159:2755–66.
[122] Luo Z, Yoo CS, Richardson ES, Chen JH, Law CK, Lu T. Chemical explosive mode analysis for a turbulent lifted ethylene jet flame in highly-heated coflow. Combust Flame 2012;159:265–74.
[123] Richardson ES, Chen JH. Application of PDF mixing models to premixed flames with differential diffusion. Combust Flame 2012;159:2398–414.
[124] Shan R, Yoo CS, Chen JH, Lu T. Computational diagnostics for *n*-heptane flames with chemical explosive mode analysis. Combust Flame 2012;159:3119–27.
[125] Bell JB, Day MS, Lijewski MJ. Simulation of nitrogen emissions in a premixed hydrogen flame stabilized on a low swirl burner. Proc Combust Inst 2013;34: 1173–82.
[126] Chatakonda O, Hawkes ER, Aspden AJ, Kerstein AR, Kolla H, Chen JH. On the fractal characteristics of low Damköhler number flames. Combust Flame 2013; 160:2422–33.
[127] Wang H, Luo K, Fan J. Direct numerical simulation and conditional statistics of hydrogen/air turbulent premixed flames. Energy Fuels 2013;27(1):549–60.
[128] Yoo CS, Luo Z, Lu T, Kim H, Chen JH. A DNS study of ignition characteristics of a lean isooctane/air mixture under HCCI and SACI conditions. Proc Combust Inst 2013;34:2985–93.
[129] Attili A, Bisetti F, Mueller ME, Pitsch H. Formation, growth, and transport of soot in a three-dimensional turbulent non-premixed jet flame. Combust Flame 2014; 161:1849–65.
[130] Bhagatwala A, Chen JH, Lu T. Direct numerical simulations of HCCI/SACI with ethanol. Combust Flame 2014;161:1826–41.
[131] Gruber A, Salimath PS, Chen JH. Direct numerical simulation of laminar flame-wall interaction for a novel H_2-selective membrane/injector configuration. Int J Hydrog Energy 2014;39:5906–18.
[132] Kolla H, Hawkes ER, Kerstein AR, Swaminathan N, Chen JH. On velocity and reactive scalar spectra in turbulent premixed flames. J Fluid Mech 2014;754: 456–87.
[133] Nambully S, Domingo P, Moureau V, Vervisch L. A filtered-laminar-flame PDF sub-grid-scale closure for LES of premixed turbulent flames: II. Application to a stratified bluff-body burner. Combust Flame 2014;161:1775–91.
[134] O'Brien J, Urzay J, Ihme M, Moin P, Saghafian A. Subgrid-scale backscatter in reacting and inert supersonic hydrogen-air turbulent mixing layers. J Fluid Mech 2014;743:554–84.
[135] Aspden AJ, Day MS, Bell JB. Turbulence-chemistry interaction in lean premixed hydrogen combustion. Proc Combust Inst 2015;35:1321–9.
[136] Attili A, Bisetti F, Mueller ME, Pitsch H. Damköhler number effects on soot formation and growth in turbulent nonpremixed flames. Proc Combust Inst 2015; 35:1215–23.
[137] Bansal G, Mascarenhas A, Chen JH. Direct numerical simulations of autoignition in stratified dimethyl-ether (DME)/air turbulent mixtures. Combust Flame 2015; 162:688–702.
[138] Bhagatwala A, Sankaran R, Kokjohn S, Chen JH. Numerical investigation of spontaneous flame propagation under RCCI conditions. Combust Flame 2015; 162:3412–26.
[139] Bhagatwala A, Luo Z, Shen H, Sutton JA, Lu T, Chen JH. Numerical and experimental investigation of turbulent DME jet flames. Proc Combust Inst 2015; 35:1157–66.
[140] Bisetti F, Sarathy SM, Toma M, Chung SH. Stabilization and structure of *n*-heptane tribrachial flames in axisymmetric laminar jets. Proc Combust Inst 2015;35:1023–32.
[141] Bruno C, Sankaran V, Kolla H, Chen JH. Impact of multi-component diffusion in turbulent combustion using direct numerical simulations. Combust Flame 2015; 162:4313–30.
[142] Day M, Tachibana S, Bell J, Lijewski M, Beckner V, Cheng RK. A combined computational and experimental characterization of lean premixed turbulent low swirl laboratory flames II. Hydrogen flames. Combust Flame 2015;162:2148–65.
[143] Gruber A, Kerstein AR, Valiev D, Law CK, Kolla H, Chen JH. Modeling of mean flame shape during premixed flame flashback in turbulent boundary layers. Proc Combust Inst 2015;35:1485–92.
[144] Jozefik Z, Kerstein AR, Schmidt H, Lyra S, Kolla H, Chen JH. One-dimensional turbulence modeling of a turbulent counterflow flame with comparison to DNS. Combust Flame 2015;162:2999–3015.
[145] Karami S, Hawkes ER, Talei M, Chen JH. Mechanisms of flame stabilisation at low lifted height in a turbulent lifted slot-jet flame. J Fluid Mech 2015;777:633–89.
[146] Kim SO, Luong MB, Chen JH, Yoo CS. A DNS study of the ignition of lean PRF/air mixtures with temperature inhomogeneities under high pressure and intermediate temperature. Combust Flame 2015;162:717–26.
[147] Kitano T, Tsuji T, Kurose R, Komori S. Effect of pressure oscillations on flashback characteristics in a turbulent channel flow. Energy Fuels 2015;29(10):6815–22.
[148] Krisman A, Hawkes ER, Talei M, Bhagatwala A, Chen JH. Polybrachial structures in dimethyl ether edge-flames at negative temperature coefficient conditions. Proc Combust Inst 2015;35:999–1006.
[149] Lapointe S, Savard B, Blanquart G. Differential diffusion effects, distributed burning, and local extinctions in high Karlovitz premixed flames. Combust Flame 2015;162:3341–55.
[150] Lyra S, Wilde B, Kolla H, Seitzman JM, Lieuwen TC, Chen JH. Structure of hydrogen-rich transverse jets in a vitiated turbulent flow. Combust Flame 2015; 162:1234–48.
[151] Minamoto Y, Kolla H, Grout RW, Gruber A, Chen JH. Effect of fuel composition and differential diffusion on flame stabilization in reacting syngas jets in turbulent cross-flow. Combust Flame 2015;162:3569–79.
[152] Miyata E, Fukushima N, Naka Y, Shimura M, Tanahashi M, Miyauchi T. Direct numerical simulation of micro combustion in a narrow circular channel with a detailed kinetic mechanism. Proc Combust Inst 2015;35:3421–7.
[153] Nikolaou ZM, Swaminathan N. Direct numerical simulation of complex fuel combustion with detailed chemistry: Physical insight and mean reaction rate modeling. Combust Sci Tech 2015;187:1759–89.
[154] Poludnenko AY. Pulsating instability and self-acceleration of fast turbulent flames. Phys Fluids 2015;27:014106.
[155] Sankaran R, Hawkes ER, Yoo CS, Chen JH. Response of flame thickness and propagation speed under intense turbulence in spatially developing lean premixed methane-air jet flames. Combust Flame 2015;162:3294–306.
[156] Savard B, Bobbitt B, Blanquart G. Structure of a high Karlovitz n-C_7H_{16} premixed turbulent flame. Proc Combust Inst 2015;35:1377–84.
[157] Vié A, Franzelli B, Gao Y, Lu T, Wang H, Ihme M. Analysis of segregation and bifurcation in turbulent spray flames: A 3D counterflow configuration. Proc Combust Inst 2015;35:1675–83.
[158] Xin YX, Yoo CS, Chen JH, Law CK. A DNS study of self-accelerating cylindrical hydrogen-air flames with detailed chemistry. Proc Combust Inst 2015;35:753–60.
[159] Grogan KP, Ihme M. Weak and strong ignition of hydrogen/oxygen mixtures in shock-tube systems. Proc Combust Inst 2015;35:2181–9.
[160] Aspden AJ, Day MS, Bell JB. Three-dimensional direct numerical simulation of turbulent lean premixed methane combustion with detailed kinetics. Combust Flame 2016;166:266–83.
[161] Attili A, Bisetti F, Mueller ME, Pitsch H. Effects of non-unity Lewis number of gas-phase species in turbulent nonpremixed sooting flames. Combust Flame 2016; 166:192–202.
[162] Bobbitt B, Lapointe S, Blanquart G. Vorticity transformation in high Karlovitz number premixed flames. Phys Fluids 2016;28:015101.
[163] Burali N, Lapointe S, Bobbitt B, Blanquart G, Xuan Y. Assessment of the constant non-unity Lewis number assumption in chemically-reacting flows. Combust Theory Model 2016;20(4):632–57.
[164] Gao Y, Shan R, Lyra S, Li C, Wang H, Chen JH, Lu T. On lumped-reduced reaction model for combustion of liquid fuels. Combust Flame 2016;163:437–46.
[165] Krisman A, Hawkes ER, Talei M, Bhagatwala A, Chen JH. Characterisation of two-stage ignition in diesel engine-relevant thermochemical conditions using direct numerical simulation. Combust Flame 2016;172:326–41.
[166] Lapointe S, Blanquart G. Fuel and chemistry effects in high Karlovitz premixed turbulent flames. Combust Flame 2016;167:294–307.
[167] Minamoto Y, Chen JH. DNS of a turbulent lifted DME jet flame. Combust Flame 2016;169:38–50.
[168] Towery CAZ, Poludnenko AY, Urzay J, O'Brien J, Ihme M, Hamlington PE. Spectral kinetic energy transfer in turbulent premixed reacting flows. Phys Rev E 2016;93:053115.
[169] Urbano A, Selle L, Staffelbach G, Cuenot B, Schmitt T, Ducruix S, Candel S. Exploration of combustion instability triggering using large eddy simulation of a multiple injector liquid rocket engine. Combust Flame 2016;169:129–40.

[170] Wang H, Hawkes ER, Chen JH. Turbulence-flame interactions in DNS of a laboratory high Karlovitz premixed turbulent jet flame. Phys Fluids 2016;28:095107.
[171] Abdelgadir A, Rakha IA, Steinmetz SA, Attili A, Bisetti F, Roberts WL. Effects of hydrodynamics and mixing on soot formation and growth in laminar coflow diffusion flames at elevated pressures. Combust Flame 2017;181:39–53.
[172] Aspden AJ, Bell JB, Day MS, Egolfopoulos FN. Turbulence-flame interactions in lean premixed dodecane flames. Proc Combust Inst 2017;36:2005–16.
[173] Belhi M, Lee BJ, Bisetti F, Im HG. A computational study of the effects of DC electric fields on non-premixed counterflow methane-air flames. J Phys D: Appl Phys 2017;50(49):494005.
[174] Chi C, Janiga G, Abdelsamie A, Zähringer K, Turányi T, Thévenin D. DNS study of the optimal chemical markers for heat release in syngas flames. Flow Turbul Combust 2017;98:1117–32.
[175] Gauding M, Dietzsch F, Goebbert JH, Thévenin D, Abdelsamie A, Hasse C. Dissipation element analysis of a turbulent non-premixed jet flame. Phys Fluids 2017;29(8):085103.
[176] Hamlington PE, Darragh R, Briner CA, Towery CAZ, Taylor BD, Poludnenko AY. Lagrangian analysis of high-speed turbulent premixed reacting flows: Thermochemical trajectories in hydrogen-air flames. Combust Flame 2017;186:193–207.
[177] Krisman A, Hawkes ER, Chen JH. Two-stage autoignition and edge flames in a high pressure turbulent jet. J Fluid Mech 2017;824:5–41.
[178] Krisman A, Hawkes ER, Talei M, Bhagatwala A, Chen JH. A direct numerical simulation of cool-flame affected autoignition in diesel engine-relevant conditions. Proc Combust Inst 2017;36:3567–75.
[179] O'Brien J, Towery CAZ, Hamlington PE, Ihme M, Poludnenko AY, Urzay J. The cross-scale physical-space transfer of kinetic energy in turbulent premixed flames. Proc Combust Inst 2017;36:1967–75.
[180] Richardson ES, Chen JH. Analysis of turbulent flame propagation in equivalence ratio-stratified flow. Proc Combust Inst 2017;36:1729–36.
[181] Savard B, Blanquart G. Effects of dissipation rate and diffusion rate of the progress variable on local fuel burning rate in premixed turbulent flames. Combust Flame 2017;180:77–87.
[182] Wang H, Hawkes ER, Chen JH. A direct numerical simulation study of flame structure and stabilization of an experimental high Ka CH_4/air premixed jet flame. Combust Flame 2017;180:110–23.
[183] Wang H, Hawkes ER, Chen JH, Zhou B, Li Z, Aldén M. Direct numerical simulations of a high Karlovitz number laboratory premixed jet flame – an analysis of flame stretch and flame thickening. J Fluid Mech 2017;815:511–36.
[184] Wang H, Hawkes ER, Zhou B, Chen JH, Li Z, Aldén M. A comparison between direct numerical simulation and experiment of the turbulent burning velocity-related statistics in a turbulent methane-air premixed jet flame at high Karlovitz number. Proc Combust Inst 2017;36:2045–53.
[185] Bisetti F, Abdelgadir A, Steinmetz SA, Attili A, Roberts WL. Self-similar scaling of pressurised sooting methane/air coflow flames at constant Reynolds and Grashof numbers. Combust Flame 2018;196:300–13.
[186] Borghesi G, Krisman A, Lu T, Chen JH. Direct numerical simulation of a temporally evolving air/n-dodecane jet at low-temperature diesel-relevant conditions. Combust Flame 2018;195:183–202.
[187] Doan NAK, Swaminathan N, Minamoto Y. DNS of MILD combustion with mixture fraction variations. Combust Flame 2018;189:173–89.
[188] Gruber A, Richardson ES, Aditya K, Chen JH. Direct numerical simulations of premixed and stratified flame propagation in turbulent channel flow. Phys Rev Fluids 2018;3:110507.
[189] Jaravel T, Riber E, Cuenot B, Pepiot P. Prediction of flame structure and pollutant formation of Sandia flame D using large eddy simulation with direct integration of chemical kinetics. Combust Flame 2018;188:180–98.
[190] Kim J, Bassenne M, Towery CAZ, Hamlington PE, Poludnenko AY, Urzay J. Spatially localized multi-scale energy transfer in turbulent premixed combustion. J Fluid Mech 2018;848:78–116.
[191] MacArt JF, Grenga T, Mueller ME. Effects of combustion heat release on velocity and scalar statistics in turbulent premixed jet flames at low and high Karlovitz numbers. Combust Flame 2018;191:468–85.
[192] Rieth M, Kempf AM, Kronenburg A, Stein OT. Carrier-phase DNS of pulverized coal particle ignition and volatile burning in a turbulent mixing layer. Fuel 2018;212:364–74.
[193] Wang H, Hawkes ER, Savard B, Chen JH. Direct numerical simulation of a high Ka CH_4/air stratified premixed jet flame. Combust Flame 2018;193:229–45.
[194] Aditya K, Gruber A, Xu C, Lu T, Krisman A, Bothien MR, Chen JH. Direct numerical simulation of flame stabilization assisted by autoignition in a reheat gas turbine combustor. Proc Combust Inst 2019;37:2635–42.
[195] Aspden AJ, Day MS, Bell JB. Towards the distributed burning regime in turbulent premixed flames. J Fluid Mech 2019;871:1–21.
[196] Bénard P, Lartigue G, Moureau V, Mercier R. Large-eddy simulation of the lean-premixed PRECCINSTA burner with wall heat loss. Proc Combust Inst 2019;37:5233–43.
[197] Dalakoti DK, Krisman A, Savard B, Wehrfritz A, Wang H, Day MS, Bell JB, Hawkes ER. Structure and propagation of two-dimensional, partially premixed, laminar flames in diesel engine conditions. Proc Combust Inst 2019;37:1961–9.
[198] Fu Y, Yu C, Yan Z, Li X. DNS analysis of the effects of combustion on turbulence in a supersonic H2/air jet flow. Aerosp Sci Technol 2019;93:105362.
[199] Govindaraju PB, Jaravel T, Ihme M. Coupling of turbulence on the ignition of multicomponent sprays. Proc Combust Inst 2019;37:3295–302.
[200] Jaravel T, Labahn J, Sforzo B, Seitzman J, Ihme M. Numerical study of the ignition behavior of a post-discharge kernel in a turbulent stratified crossflow. Proc Combust Inst 2019;37:5065–72.
[201] Lipkowicz JT, Wlokas I, Kempf AM. Analysis of mild ignition in a shock tube using a highly resolved 3D-LES and high-order shock-capturing schemes. Shock Waves 2019;29(4):511–21.
[202] Luca S, Attili A, Lo Schiavo E, Creta F, Bisetti F. On the statistics of flame stretch in turbulent premixed jet flames in the thin reaction zone regime at varying Reynolds number. Proc Combust Inst 2019;37:2451–9.
[203] Ma PC, Wu H, Jaravel T, Bravo L, Ihme M. Large-eddy simulations of transcritical injection and auto-ignition using diffuse-interface method and finite-rate chemistry. Proc Combust Inst 2019;37:3303–10.
[204] Poludnenko AY, Chambers J, Ahmed K, Gamezo VN, Taylor BD. A unified mechanism for unconfined deflagration-to-detonation transition in terrestrial chemical systems and Type 1a supernovae. Science 2019;366(6465):eaau7365.
[205] Savard B, Hawkes ER, Aditya K, Wang H, Chen JH. Regimes of premixed turbulent spontaneous ignition and deflagration under gas-turbine reheat combustion conditions. Combust Flame 2019;208:402–19.
[206] Whitman SHR, Towery CAZ, Poludnenko AY, Hamlington PE. Scaling and collapse of conditional velocity structure functions in turbulent premixed flames. Proc Combust Inst 2019;37:2527–35.
[207] Xu C, Poludnenko AY, Zhao X, Wang H, Lu T. Structure of strongly turbulent premixed *n*-dodecane-air flames: direct numerical simulations and chemical explosive mode analysis. Combust Flame 2019;209:27–40.
[208] Denker D, Attili A, Luca S, Bisetti F, Gauding M, Pitsch H. Dissipation element analysis of turbulent premixed jet flames. Combust Sci Tech 2019;191(9):1677–92.
[209] Abdelsamie A, Kruis FE, Wiggers H, Thévenin D. Nanoparticle formation and behavior in turbulent spray flames investigated by DNS. Flow Turbul Combust 2020;105:497–516.
[210] Aoki K, Shimura M, Park J, Minamoto Y, Tanahashi M. Response of heat release rate to flame straining in swirling hydrogen-air premixed flames. Flow Turbul Combust 2020;104:451–78.
[211] Bambauer M, Hasslberger J, Klein M. Direct numerical simulation of the Richtmyer–Meshkov instability in reactive and nonreactive flows. Combust Sci Tech 2020;192(11):2010–27.
[212] Benekos S, Frouzakis CE, Giannakopoulos GK, Bolla M, Wright YM, Boulouchos K. Prechamber ignition: An exploratory 2-D DNS study of the effects of initial temperature and main chamber composition. Combust Flame 2020;215:10–27.
[213] Brearley P, Ahmed U, Chakraborty N, Klein M. Scaling of second-order structure functions in turbulent premixed flames in the flamelet combustion regime. Fluids 2020;5(89):1–12.
[214] Chabane AM, Truffin K, Angelberger C. Direct numerical simulation of catalytic combustion in a meso-scale channel with non-planar walls. Combust Flame 2020;222:85–102.
[215] Cifuentes L, Sellmann J, Wlokas I, Kempf A. Direct numerical simulations of nanoparticle formation in premixed and non-premixed flame–vortex interactions. Phys Fluids 2020;32(9):093605.
[216] Dalakoti DK, Savard B, Hawkes ER, Wehrfritz A, Wang H, Day MS, Bell JB. Direct numerical simulation of a spatially developing *n*-dodecane jet flame under spray a thermochemical conditions: flame structure and stabilisation mechanism. Combust Flame 2020;217:57–76.
[217] Dave HL, Chaudhuri S. Evolution of local flame displacement speeds in turbulence. J Fluid Mech 2020;884:A46.
[218] Denker D, Attili A, Boschung J, Hennig F, Gauding M, Bode M, Pitsch H. Dissipation element analysis of non-premixed jet flames. J Fluid Mech 2020;905:A4.
[219] Domingo-Alvarez P, Bénard P, Moureau V, Lartigue G, Grisch F. Impact of spray droplet distribution on the performances of a kerosene lean/premixed injector. Flow Turbul Combust 2020;104:421–50.
[220] Falkenstein T, Kang S, Cai L, Bode M, Pitsch H. DNS study of the global heat release rate during early flame kernel development under engine conditions. Combust Flame 2020;213:455–66.
[221] Falkenstein T, Rezchikova A, Langer R, Bode M, Kang S, Pitsch H. The role of differential diffusion during early flame kernel development under engine conditions–part I: Analysis of the heat-release-rate response. Combust Flame 2020;221:502–15.
[222] Fillo AJ, Schlup J, Beardsell G, Blanquart G, Niemeyer KE. A fast, low-memory, and stable algorithm for implementing multicomponent transport in direct numerical simulations. J Comput Phys 2020;406:109185.
[223] Gao X. Direct numerical simulation of mixing and combustion under canonical shock turbulence interaction. University of Southern California; 2020. Ph.D. thesis.
[224] Haghiri A, Talei M, Brear MJ, Hawkes ER. Sound generation by turbulent premixed flames. J Fluid Mech 2018;843:29–52.
[225] Jiang J, Wu T, Li H, Sun M, Zhang B. Analysis of turbulent transport characteristic in hydrogen diffusion flames using direct numerical simulation. Numer Heat Transf; A: Appl 2020;78(4):125–39.
[226] Kim SH, Su Y. Front propagation formulation for large eddy simulation of turbulent premixed flames. Combust Flame 2020;220:439–54.
[227] Klein M, Herbert A, Kosaka H, Böhm B, Dreizler A, Chakraborty N, Papapostolou V, Im HG, Hasslberger J. Evaluation of flame area based on detailed chemistry DNS of premixed turbulent hydrogen-air flames in different regimes of combustion. Flow Turbul Combust 2020;104:403–19.

[228] Luong MB, Pérez FEH, Im HG. Prediction of ignition modes of NTC-fuel/air mixtures with temperature and concentration fluctuations. Combust Flame 2020;213:382–93.
[229] Ma M-C, Talei M, Sandberg RD. Direct numerical simulation of turbulent premixed jet flames: Influence of inflow boundary conditions. Combust Flame 2020;213:240–54.
[230] Malkeson SP, Wacks DH, Chakraborty N. Statistical behaviour and modelling of fuel mass fraction dissipation rate transport in turbulent flame-droplet interaction: A direct numerical simulation study. Flow Turbul Combust 2020;105:237–66.
[231] Ozel-Erol G, Hasslberger J, Klein M, Chakraborty N. A direct numerical simulation analysis of turbulent V-shaped flames propagating into droplet-laden mixtures. Int J Multiph Flow 2020;133:103455.
[232] Towery CAZ, Poludnenko AY, Hamlington PE. Detonation initiation by compressible turbulence thermodynamic fluctuations. Combust Flame 2020;213:172–83.
[233] Wan K, Barnaud C, Vervisch L, Domingo P. Chemistry reduction using machine learning trained from non-premixed micro-mixing modeling: Application to DNS of a syngas turbulent oxy-flame with side-wall effects. Combust Flame 2020;220:119–29.
[234] Wang Z, Wang H, Luo K, Fan J. Direct numerical simulation of particle-laden turbulent boundary layers without and with combustion. Phys Fluids 2020;32:105108.
[235] Wu B, Roy SP, Zhao X. Detailed modeling of a small-scale turbulent pool fire. Combust Flame 2020;214:224–37.
[236] You J, Yang Y. Modelling of the turbulent burning velocity based on Lagrangian statistics of propagating surfaces. J Fluid Mech 2020;887:A11.
[237] Zhong L, Zhang X, Zhou L, Liu C, Wei H. Direct numerical simulation of flame propagation and deflagration to detonation transition in confined space with different perforated plate positions. Combust Sci Tech 2021;193(16):2907–34.
[238] Zhou T, Zhao P, Ye T, Zhu M, Tao C. Direct numerical simulation of low temperature reactions affecting *n*-dodecane spray autoignition. Fuel 2020;280:118453.
[239] Chung WT, Ma PC, Ihme M. Examination of diesel spray combustion in supercritical ambient fluid using large-eddy simulations. Int J Engine Res 2020;21(1):122–33.
[240] Alavi M, Leidner DE. Review: Knowledge management and knowledge management systems: conceptual foundations and research issues. Manag Inf Syst Q 2001;25(1):107–36.
[241] Springmeyer RR, Blattner MM, Max NL. A characterization of the scientific data analysis process. Proc Conf Vis 1992;3:235–42.
[242] National Research Council. Transforming combustion research through cyberinfrastructure. Washington, DC: The National Academies Press; 2011.
[243] Johnson R, Wu H, Ihme M. A general probabilistic approach for the quantitative assessment of LES combustion models. Combust Flame 2017;183:88–101.
[244] Pedregosa F, Varoquaux G, Gramfort A, Michel et al V. Scikit-learn: machine learning in python. J Mach Learn Res 2011;12:2825–30.
[245] Chollet F.. Keras. 2015. https://github.com/fchollet/keras.
[246] Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, et al. **PyTorch**: an imperative style, high-performance deep learning library. Adv Neural Inform Process Syst 2019;32:8024–35.
[247] Abadi M., Agarwal A., Barham P., Brevdo E., Chen Z., Citro C., Corrado G.S., Davis A., Dean J., Devin M., Ghemawat S., Goodfellow I., Harp A., Irving G., Isard M., Jia Y., Jozefowicz R., Kaiser L., Kudlur M., Levenberg J., Mané D., Monga R., Moore S., Murray D., Olah C., Schuster M., Shlens J., Steiner B., Sutskever I., Talwar K., Tucker P., Vanhoucke V., Vasudevan V., Viégas F., Vinyals O., Warden P., Wattenberg M., Wicke M., Yu Y., Zheng X.. TensorFlow: Large-scale machine learning on heterogeneous systems. 2015. https://www.tensorflow.org.
[248] Williams FA. Combustion Theory. CRC Press; 1985.
[249] Giovangigli V. Multicomponent Flow Modeling. Boston: Birkhäuser; 1999.
[250] Lu T, Law CK. Toward accommodating realistic fuel chemistry in large-scale computations. Prog Energy Combust Sci 2009;35:192–215.
[251] Nastac G, Labahn JW, Magri L, Ihme M. Lyapunov exponent as a metric for assessing the dynamic content and predictability of large-eddy simulations. Phys Rev Fluids 2017;2:094606.
[252] Mohan P, Fitzsimmons N, Moser RD. Scaling of Lyapunov exponents in homogeneous isotropic turbulence. Phys Rev Fluids 2017;2:114606.
[253] Pope SB. PDF methods for turbulent reactive flows. Prog Energy Combust Sci 1985;11:119–92.
[254] Peters N. Turbulent Combustion. Cambridge University Press; 2000.
[255] Pitsch H. Large-eddy simulation of turbulent combustion. Annu Rev Fluid Mech 2006;38:453–82.
[256] Pope SB. Small scales, many species and the manifold challenges of turbulent combustion. Proc Combust Inst 2013;34:1–31.
[257] Lu T, Law CK. A directed relation graph method for mechanism reduction. Proc Combust Inst 2005;30:1333–41.
[258] Pepiot-Desjardins P, Pitsch H. An efficient error-propagation-based reduction method for large chemical kinetic mechanisms. Combust Flame 2008;154:67–81.
[259] Niemeyer KE, Sung C-J, Raju MP. Skeletal mechanism generation for surrogate fuels using directed relation graph with error propagation and sensitivity analysis. Combust Flame 2010;157:1760–70.
[260] Sun W, Chen Z, Gou X, Ju Y. A path flux analysis method for the reduction of detailed chemical kinetic mechanisms. Combust Flame 2010;157:1298–307.
[261] Jaravel T, Wu H, Ihme M. Error-controlled kinetics reduction based on non-linear optimization and sensitivity analysis. Combust Flame 2019;200:192–206.
[262] Peters N. Laminar diffusion flamelet models in non-premixed turbulent combustion. Prog Energy Combust Sci 1984;10:319–39.
[263] Maas U, Pope SB. Simplifying chemical kinetics: Intrinsic low-dimensional manifolds in composition space. Combust Flame 1992;88:239–64.
[264] Gicquel O, Darabiha N, Thevenin D. Laminar premixed hydrogen/air counterflow flame simulations using flame prolongation of ILDM with differential diffusion. Proc Combust Inst 2000;28:1901–8.
[265] van Oijen JA, Lammers FA, de Goey LPH. Modeling of complex premixed burner systems by using flamelet-generated manifolds. Combust Flame 2001;127:2124–34.
[266] Pierce CD, Moin P. Progress-variable approach for large-eddy simulation of non-premixed turbulent combustion. J Fluid Mech 2004;504:73–97.
[267] Ihme M, Cha CM, Pitsch H. Prediction of local extinction and re-ignition effects in non-premixed turbulent combustion using a flamelet/progress variable approach. Proc Combust Inst 2005;30:793–800.
[268] Parente A, Sutherland JC, Tognotti L, Smith PJ. Identification of low-dimensional manifolds in turbulent flames. Proc Combust Inst 2009;32:1579–86.
[269] Frenklach M. Transforming data into knowledge—process informatics for combustion chemistry. Proc Combust Inst 2007;31:125–40.
[270] Ruscic B, Pinzon RE, Morton ML, von Laszevski G, Bittner SJ, Nijsure SG, et al. Introduction to Active Thermochemical Tables: Several "key" enthalpies of formation revisited. J Phys Chem A 2004;108(45):9979–97.
[271] Najm HN, Debusschere BJ, Marzouk YM, Widmer S, Le Maître OP. Uncertainty quantification in chemical systems. Int J Numer Meth Engrg 2009;80(6–7):789–814.
[272] Tomlin AS. The role of sensitivity and uncertainty analysis in combustion modelling. Proc Combust Inst 2013;34:159–76.
[273] Braman K, Oliver TA, Raman V. Bayesian analysis of syngas chemistry models. Combust Theory Model 2013;17(5):858–87.
[274] Wang H, Sheen DA. Combustion kinetic model uncertainty quantification, propagation and minimization. Prog Energy Combust Sci 2015;47:1–31.
[275] Willcox KE, Ghattas O, Heimbach P. The imperative of physics-based modeling and inverse theory in computational science. Nat Comput Sci 2021;1:166–8.
[276] Raissi M, Yazdani A, Karniadakis GE. Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations. Science 2020;367(6481):1026–30.
[277] Karniadakis GE, Kevrekidis IG, Lu L, Perdikaris P, Wang S, Yang L. Physics-informed machine learning. Nat Rev Phys 2021;3:422–40.
[278] Kashinath K, Mustafa M, Albert A, Wu J-L, Jiang C, Esmaeilzadeh S, et al. Physics-informed machine learning: Case studies for weather and climate modelling. Phil Trans R Soc A 2021;379(2194):20200093.
[279] Bode M, Gauding M, Lian Z, Denker D, Davidovic M, Kleinheinz K, Jitsev J, Pitsch H. Using physics-informed enhanced super-resolution generative adversarial networks for subfilter modeling in turbulent reactive flows. Proc Combust Inst 2021;38:2617–25.
[280] Karpatne A, Atluri G, Faghmous JH, Steinbach M, Banerjee A, Ganguly A, et al. Theory-guided data science: A new paradigm for scientific discovery from data. IEEE Trans Knowl Data Eng 2017;29(10):2318–31.
[281] Willard J, Jia X, Xu S, Steinbach M, Kumar V. Integrating scientific knowledge with machine learning for engineering and environmental systems. arXiv Preprint 2021;2003.04919.
[282] Bishop CM. Pattern Recognition and Machine Learning. Springer; 2006.
[283] Murphy KP. Machine Learning: A Probabilistic Perspective. MIT Press; 2012.
[284] LeCun Y, Bengio Y, Hinton G. Deep learning. Nature 2015;521:436–44.
[285] Jordan MI, Mitchell TM. Machine learning: Trends, perspectives, and prospects. Science 2015;349:255–60.
[286] Goodfellow IJ, Bengio Y, Courville A. Deep Learning. MIT Press; 2016.
[287] Baker N, Alexander F, Bremer T, Hagberg A, Kevrekidis Y, Najm H, Parashar M, Patra A, Sethian J, Wild S, Willcox K, Lee S. Basic research needs for scientific machine learning: Core technologies for artificial intelligence. Tech. Rep. U. S. Department of Energy, Advanced Scientific Computing Research; 2019.
[288] Chung W.T., Ihme M.. CombML tutorials. 2021. https://github.com/IhmeGroup/CombML_Tutorials.
[289] Hastie T, Tibshirani R, Friedman J. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. 2nd. Springer; 2009.
[290] Brunton SL, Noack BR, Koumoutsakos P. Machine learning for fluid mechanics. Annu Rev Fluid Mech 2020;52:477–508.
[291] Jain P, Coogan SCP, Subramanian SG, Crowley M, Taylor S, Flannigan MD. A review of machine learning applications in wildfire science and management. Environ Rev 2020;28:478–505.
[292] Haworth DC. Progress in probability density function methods for turbulent reacting flows. Prog Energy Combust Sci 2010;36:168–259.
[293] Olkin I, Gleser LJ, Derman C. Probability Models and Applications. 2nd. World Scientific; 2019.
[294] Bernardo JM, Smith AFM. Bayesian Theory. John Wiley & Sons, Inc.; 1994.
[295] Rubinstein RY, Kroese DP. Simulation and the Monte Carlo Method. John Wiley & Sons, Inc.; 2008.
[296] Ly A, Marsman M, Verhagen J, Grasman RPPP, Wagenmakers E-J. A tutorial on Fisher information. J Math Psychol 2017;80:40–55.
[297] Davis J, Goadrich M. The relationship between precision-recall and ROC curves. Proc Int Conf Mach Learn 2006:233–40.
[298] Natekin A, Knoll A. Gradient boosting machines, a tutorial. Front Neurorob 2013;7(21):1–21.
[299] Guyon I. A scaling law for the validation-set training-set size ratio. AT&T Bell Lab 1997:1–11.
[300] Ruder S. An overview of gradient descent optimization algorithms. arXiv Preprint 2016;1609.04747.

[301] Kingma DP, Ba J. Adam: A method for stochastic optimization. Proc. Int. Conf. Learn. Repr.. 2015.
[302] Fletcher R. Practical Methods of Optimization. John Wiley & Sons, Inc.; 2013.
[303] Boyd S, Boyd SP, Vandenberghe L. Convex Optimization. Cambridge University Press; 2004.
[304] Bergstra JS, Bengio Y. Random search for hyper-parameter optimization. J Mach Learn Res 2012;13:281–305.
[305] Bergstra JS, Bardenet R, Bengio Y, Kégl B. Algorithms for hyper-parameter optimization. Adv Neural Inform Process Syst 2011;24:2546–54.
[306] Geman S, Bienenstock E, Doursat R. Neural networks and the bias/variance dilemma. Neural Comput 1992;4(1):1–58.
[307] Aung KT, Tseng L-K, Ismail MA, Faeth GM. Response to comment by S. C. Taylor and D. B. Smith on "Laminar burning velocities and Markstein numbers of hydrocarbon/air flames". Combust Flame 1995;102:526–30.
[308] Bosschaart KJ, de Goey LPH. The laminar burning velocity of flames propagating in mixtures of hydrocarbons and air measured with the heat flux method. Combust Flame 2004;136(3):261–9.
[309] Dirrenberger P, Le Gall H, Bounaceur R, Herbinet O, Glaude P-A, Konnov A, Battin-Leclerc F. Measurements of laminar flame velocity for components of natural gas. Energy Fuels 2011;25(9):3875–84.
[310] Egolfopoulos FN, Cho P, Law CK. Laminar flame speeds of methane-air mixtures under reduced and elevated pressures. Combust Flame 1989;76:375–91.
[311] Elia M, Ulinski M, Metghalchi M. Laminar burning velocity of methane-air-diluent mixtures. J Eng Gas Turbines Power 2000;123:190–6.
[312] Gu XJ, Haq MZ, Lawes M, Woolley R. Laminar burning velocity and Markstein lengths of methane-air mixtures. Combust Flame 2000;121:41–58.
[313] Hassan MI, Aung KT, Faeth GM. Measured and predicted properties of laminar premixed methane/air flames at various pressures. Combust Flame 1998;115:539–50.
[314] Kochar YN, Vaden SN, Lieuwen TC, Seitzman JM. Laminar flame speed of hydrocarbon fuels with preheat and low oxygen content. AIAA Pap 2010-778 2010.
[315] Lowry W, de Vries J, Krejci M, Petersen E, Serinyel Z, Metcalfe W, Curran H, Bourque G. Laminar flame speed measurements and modeling of pure alkanes and alkane blends at elevated pressures. J Eng Gas Turbines Power 2011;133(9):091501.
[316] Park O, Veloo PS, Liu N, Egolfopoulos FN. Combustion characteristics of alternative gaseous fuels. Proc Combust Inst 2011;33:887–94.
[317] Tahtouh T, Halter F, Mounaïm-Rousselle C. Measurement of laminar burning speeds and Markstein lengths using a novel methodology. Combust Flame 2009;156:1735–43.
[318] Vagelopoulos CM, Egolfopoulos FN, Law CK. Further considerations on the determination of laminar flame speeds with the counterflow twin-flame technique. Symp (Int) Combust 1994;25:1341–7.
[319] Vagelopoulos CM, Egolfopoulos FN. Direct experimental determination of laminar flame speeds. Symp (Int) Combust 1998;27:513–9.
[320] van Maaren A, de Goey LPH. Stretch and the adiabatic burning velocity of methane-and propane-air flames. Combust Sci Tech 1994;102:309–14.
[321] Gülder ÖL. Correlations of laminar combustion data for alternative S.I. engine fuels. SAE Techn Pap 841000 1984.
[322] Abu-Mostafa YS, Magdon-Ismail M, Lin H-T. Learning from Data: A Short Course. AMLBook; 2012.
[323] Vapnik VN, Chervonenkis AY. On the uniform convergence of relative frequencies of events to their probabilities. In: Vovk V, Papadopoulos H, Gammerman A, editors. Measures of Complexity: Festschrift for Alexey Chervonenkis. Springer; 2015. p. 11–30.
[324] Quionero-Candela J, Sugiyama M, Schwaighofer A, Lawrence ND. Dataset Shift in Machine Learning. MIT Press; 2009.
[325] Christo FC, Masri AR, Nebot EM, Pope SB. An integrated PDF/neural network approach for simulating turbulent reacting systems. Proc Combust Inst 1996;26:43–8.
[326] Novati G, de Laroussilhe HL, Koumoutsakos P. Automating turbulence modelling by multi-agent reinforcement learning. Nat Mach Intell 2021;3(1):87–96.
[327] Kleinbaum DG, Klein M. Logistic Regression: A Self-Learning Text. 3rd. Springer; 2010.
[328] Green P. Iteratively reweighted least squares for maximum likelihood estimation, and some robust and resistant alternatives. J R Statist Soc B 1984;46(2):149–92.
[329] Morgan JN, Sonquist JA. Problems in the analysis of survey data, and a proposal. J Am Stat Assoc 1963;58(302):415–34.
[330] Breiman L, Friedman J, Stone CJ, Olshen RA. Classification and Regression Trees. CRC Press; 1984.
[331] Gini C. Variabilità E Mutabilità: Contributo Allo Studio Delle Distribuzioni E Delle Relazioni Statistiche. Bologna, Tipogr. di P. Cuppini; 1912.
[332] Cormen TH, Leiserson CE, Rivest RL, Stein C. Introduction to Algorithms. MIT Press; 2009.
[333] Louppe G. Understanding random forests: From theory to practice. Université de Liège; 2014. Ph.D. thesis.
[334] Schapire RE. The strength of weak learnability. Mach Learn 1990;5:197–227.
[335] Zhou Z-H. Ensemble Methods: Foundations and Algorithms. CRC Press; 2012.
[336] Breiman L. Random forests. Mach Learn 2001;45:5–32.
[337] Amit Y, Geman D, Wilder K. Joint induction of shape features and tree classifiers. IEEE Trans Pattern Anal Mach Intell 1997;19(11):1300–5.
[338] Szeliski R. Computer Vision: Algorithms and Applications. Springer; 2010.
[339] Kramer MA. Nonlinear principal component analysis using autoassociative neural networks. AIChE J 1991;37:233–43.
[340] Hinton GE, Salakhutdinov RR. Reducing the dimensionality of data with neural networks. Science 2006;313(5786):504–7.
[341] Rumelhart DE, Hinton GE, Williams RJ. Learning representations by back-propagating errors. Nature 1986;323:533–6.
[342] Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput 1997;9(8):1735–80.
[343] Boser BE, Guyon IM, Vapnik VN. A training algorithm for optimal margin classifiers. Proc Annu Workshop Comput Learn Theor 1992;5:144–52.
[344] Cortes C, Vapnik V. Support-vector networks. Mach Learn 1995;20(3):273–97.
[345] Goodwin D.G., Speth R.L., Moffat H.K., Weber B.W.. Cantera: An object-oriented software toolkit for chemical kinetics, thermodynamics, and transport processes. https://www.cantera.org; 2021.
[346] Yao T, Pei Y, Zhong B-J, Som S, Lu T, Luo KH. A compact skeletal mechanism for *n*-dodecane with optimized semi-global low-temperature chemistry for diesel engine simulations. Fuel 2017;191:339–49.
[347] Ju Y, Reuter CB, Yehia OR, Farouk TI, Won SH. Dynamics of cool flames. Prog Energy Combust Sci 2019;75:100787.
[348] Snoek J, Larochelle H, Adams RP. Practical Bayesian optimization of machine learning algorithms. Adv Neural Inform Process Syst 2012;25:2951–9.
[349] Hutter F, Kotthoff L, Vanschoren J, editors. Automated Machine Learning : Methods, Systems, Challenges. Springer; 2019.
[350] Deng J, Dong W, Socher R, Li L-J, Li K, Fei-Fei L. ImageNet: A large-scale hierarchical image database. IEEE Conf Comput Vision Pattern Recognit 2009:248–55.
[351] Mitchell TM. The need for biases in learning generalizations. CBM-TR-117. Computer Science Department, Rutgers University; 1980.
[352] Baxter J. A model of inductive bias learning. J Artif Intell Res 2000;12:149–98.
[353] Sutskever I, Vinyals O, Le QV. Sequence to sequence learning with neural networks. Adv Neural Inf Process Syst 2014;27.
[354] Mnih V, Kavukcuoglu K, Silver D, Graves A, Antonoglou I, Wierstra D, et al. Playing Atari with deep reinforcement learning. arXiv Preprint 2013;1312.5602.
[355] Lake BM, Ullman TD, Tenenbaum JB, Gershman SJ. Building machines that learn and think like people. Behav Brain Sci 2017;40:e253.
[356] Cohen T, Weiler M, Kicanaoglu B, Welling M. Gauge equivariant convolutional networks and the icosahedral CNN. Int Conf Mach Learn 2019:1321–30.
[357] Bronstein MM, Bruna J, LeCun Y, Szlam A, Vandergheynst P. Geometric deep learning: Going beyond Euclidean data. IEEE Signal Process Mag 2017;34(4):18–42.
[358] Zou H, Hastie T. Regularization and variable selection via the elastic net. J R Stat Soc B 2005;67(2):301–20.
[359] Mianjy P, Arora R. On convergence and generalization of dropout training. Adv Neural Inf Process Syst 2020;33.
[360] Pearson K. On lines and planes of closest fit to systems of points in space. Lond Edinb Dubl Phil Mag 1901;2(11):559–72.
[361] Steinhaus H. Sur la division des corp matériels en parties. Bull Acad Polon Sci 1956;4(12):801–4.
[362] Ghahramani Z. Unsupervised learning. In: Bousquet O, von Luxburg U, Rätsch G, editors. Advanced Lectures on Machine Learning: ML Summer Schools 2003, Canberra, Australia, February 2003, Tübingen, Germany, August 2003, Revised Lectures. Springer; 2004. p. 72–112.
[363] Celebri ME, Aydin K, editors. Unsupervised Learning Algorithms. Springer; 2016.
[364] Jain AK, Murty MN, Flynn PJ. Data clustering: A review. ACM Comput Surv 1999;31(3):264–323.
[365] Jain AK. Data clustering: 50 years beyond K-means. Pattern Recognit Lett 2010;31:651–66.
[366] Florek K, Łukaszewicz J, Perkal J, Steinhaus H, Zubrzycki S. Sur la liaison et la division des points d'un ensemble fini. Colloq Math 1951;2(3–4):282–5.
[367] Lance GN, Williams WT. A general theory of classificatory sorting strategies 1. Hierarchical systems. Comput J 1967;9(4):373–80.
[368] Ding S, Zhu H, Jia W, Su C. A survey on feature extraction for pattern recognition. Artif Intell Rev 2012;37:169–80.
[369] Khalid S, Khalil T, Nasreen S. A survey of feature selection and feature extraction techniques in machine learning. Sci Inf Conf 2014:372–8.
[370] Li J, Cheng K, Wang S, Morstatter F, Trevino RP, Tang J, et al. Feature selection: A data perspective. ACM Comput Surv 2017;50(6):1–45.
[371] Ding C, Peng H. Minimum redundancy feature selection from microarray gene expression data. J Bioinform Comput Biol 2005;3(2):185–205.
[372] Kira K, Rendell LA. The feature selection problem: Traditional methods and a new algorithm. Proc Nat Conf Artif Intell 1992;10:129–34.
[373] Lloyd SP. Least squares quantization in PCM. IEEE Trans Inf Theory 1982;28(2):129–37.
[374] Gupta S, Kumar R, Lu K, Moseley B, Vassilvitskii S. Local search methods for *k*-means with outliers. Proc VLDB Endow 2017;10(7):757–68.
[375] Ahmadian S, Norouzi-Fard A, Svensson O, Ward J. Better guarantees for *k*-means and euclidean *k*-median by primal-dual algorithms. Annu IEEE Symp Found Comput Sci 2017;58:61–72.
[376] Jackson JE. A User's Guide to Principal Components. John Wiley & Sons, Inc.; 1991.
[377] Jolliffe IT. Principal Component Analysis. 2nd. Springer; 2002.
[378] Jolliffe IT, Cadima J. Principal component analysis: A review and recent developments. Phil Trans R Soc A 2016;374(2065):20150202.
[379] Yi S, Lai Z, He Z, Cheung Y, Liu Y. Joint sparse principal component analysis. Pattern Recognit 2017;61:524–36.
[380] Lu C, Feng J, Chen Y, Liu W, Lin Z, Yan S. Tensor robust principal component analysis with a new tensor nuclear norm. IEEE Trans Pattern Anal Mach Intell 2020;42(4):925–38.

[381] Douasbin Q, Ihme M, Arndt C. Pareto-efficient combustion framework for predicting transient ignition dynamics in turbulent flames: Application to a pulsed jet-in-hot-coflow flame. Combust Flame 2021;223:153–65.
[382] Arndt CM, Papageorge MJ, Fuest F, Sutton JA, Meier W, Aigner M. The role of temperature, mixture fraction, and scalar dissipation rate on transient methane injection and auto-ignition in a jet in hot coflow burner. Combust Flame 2016; 167:60–71.
[383] Ng AY, Jordan MI. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. Adv Neural Inform Process Syst 2002;14: 605–10.
[384] Kingma DP, Welling M. Auto-encoding variational Bayes. Proc. Int. Conf. Learn. Repr.. 2014.
[385] Goodfellow IJ, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y. Generative adversarial networks. arXiv Preprint 2014; 1406.2661.
[386] Radford A, Metz L, Chintala S. Unsupervised representation learning with deep convolutional generative adversarial networks. Proc Int Conf Learn Repr 2016.
[387] Arjovsky M, Chintala S, Bottou L. Wasserstein generative adversarial networks. Int Conf Mach Learn 2017;70:214–23.
[388] Ledig C, Theis L, Huszár F, Caballero J, Cunningham A, Acosta A, Aitken A, Tejani A, Totz J, Wang Z, Shi W. Photo-realistic single image super-resolution using a generative adversarial network. Proc IEEE Conf Comput Vision Pattern Recognit 2017:105–14.
[389] Saxena D, Cao J. Generative adversarial networks (GANs): Challenges, solutions, and future directions. ACM Comput Surv 2021;54(3):1–42.
[390] Wang Z, She Q, Ward TE. Generative adversarial networks in computer vision: A survey and taxonomy. ACM Comput Surv 2021;54:1–38.
[391] Creswell A, White T, Dumoulin V, Arulkumaran K, Sengupta B, Bharath AA. Generative adversarial networks: An overview. IEEE Signal Process Mag 2018;35: 53–65.
[392] Ling J, Templeton J. Evaluation of machine learning algorithms for prediction of regions of high Reynolds-averaged Navier-Stokes uncertainty. Phys Fluids 2015; 27(8):085103.
[393] Sutton RS, Barto AG. Reinforcement Learning: An Introduction. 2nd. MIT Press; 2018.
[394] Baird L. Residual algorithms: Reinforcement learning with function approximation. Mach Learn Proc 1995;12:30–7.
[395] Bellman R. Dynamic Programming. Princeton University Press; 1957.
[396] Watkins CJCH. Learning from delayed rewards. University of Cambridge, King's College; 1989. Ph.D. thesis.
[397] François-Lavet V, Henderson P, Islam R, Bellemare MG, Pineau J. An introduction to deep reinforcement learning. Found Trends Mach Learn 2018;11(3–4): 219–354.
[398] Nian R, Liu J, Huang B. A review on reinforcement learning: Introduction and applications in industrial process control. Comput Chem Eng 2020;139:106886.
[399] Dulac-Arnold G, Levine N, Mankowitz DJ, Li J, Paduraru C, Gowal S, Hester T. Challenges of real-world reinforcement learning: definitions, benchmarks and analysis. Mach Learn 2021;110:2419–68.
[400] Turns SR. An Introduction to Combustion: Concepts an Applications. 3rd. McGraw-Hill; 2012.
[401] van Leeuwen T, Herrmann FJ. A penalty method for PDE-constrained optimization in inverse problems. Inverse Probl 2015;32:015007.
[402] Psichogios DC, Ungar LH. A hybrid neural network-first principles approach to process modeling. AIChE J 1992;38(10):1499–511.
[403] Lagaris IE, Likas A, Fotiadis DI. Artificial neural networks for solving ordinary and partial differential equations. IEEE Trans Neural Netw 1998;9(5):987–1000.
[404] Shengze C, Zhiping M, Zhicheng W, Minglang Y, Karniadakis GE. Physics-informed neural networks (PINNs) for fluid mechanics: A review. arXiv Preprint 2021;2105.09506.
[405] Zobeiry N, Humfeld KD. A physics-informed machine learning approach for solving heat transfer equation in advanced manufacturing and engineering applications. Eng Appl Artif Intell 2021;101:104232.
[406] Ling J, Kurzawski A, Templeton J. Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. J Fluid Mech 2016;807:155–66.
[407] Chen TQ, Rubanova Y, Bettencourt J, Duvenaud D. Neural ordinary differential equations. Adv Neural Inform Process Syst 2018;31:6572–83.
[408] Long Z, Lu Y, Ma X, Dong B. PDE-Net: Learning PDEs from data. Proc Int Conf Mach Learn 2018;80:3208–16.
[409] Kim B, Azevedo VC, Thuerey N, Kim T, Gross M, Solenthaler B. Deep fluids: A generative network for parameterized fluid simulations. Comput Graph Forum 2019;38(2):59–70.
[410] Raissi M, Karniadakis GE. Hidden physics models: Machine learning of nonlinear partial differential equations. J Comput Phys 2018;357:125–41.
[411] Raissi M, Perdikaris P, Karniadakis GE. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J Comput Phys 2019;378:686–707.
[412] Lam SH, Goussis DA. The CSP method for simplifying kinetics. Int J Chem Kinet 1994;26:461–86.
[413] van Oijen JA, de Goey LPH. Modelling of premixed laminar flames using flamelet-generated manifolds. Combust Sci Tech 2000;161:113–37.
[414] Bykov V, Maas U. The extension of the ILDM concept to reaction-diffusion manifolds. Combust Theory Model 2007;11(6):839–62.
[415] Keck JC, Gillespi D. Rate-controlled partial-equilibrium method for treating reacting gas-mixtures. Combust Flame 1971;17:237–41.
[416] Ren Z, Goldin GM, Hiremath V, Pope SB. Reduced description of reactive flows with tabulation of chemistry. Combust Theory Model 2011;15(6):827–48.
[417] Hiremath V, Ren Z, Pope SB. A greedy algorithm for species selection in dimension reduction of combustion chemistry. Combust Theory Model 2010;14 (5):619–52.
[418] Sutherland JC, Parente A. Combustion modeling using principal component analysis. Proc Combust Inst 2009;32:1563–70.
[419] Echekki T, Mastorakos E, editors. Turbulent Combustion Modeling: Advances, New Trends and Perspectives. Springer; 2010.
[420] Benson SW, Buss JH. Additivity rules for the estimation of molecular properties. thermodynamic properties. J Chem Phys 1958;29:546–72.
[421] Todeschini R, Consonni V. Molecular Descriptors for Chemoinformatics. 2nd. Wiley-VCH; 2009.
[422] Katritzky AR, Kuanar M, Slavov S, Hall CD, Karelson M, Kahn I, et al. Quantitative correlation of physical and chemical properties with chemical structure: Utility for prediction. Chem Rev 2010;110(10):5714–89.
[423] Le T, Epa VC, Burden FR, Winkler DA. Quantitative structure–property relationship modeling of diverse materials properties. Chem Rev 2012;112(5): 2889–919.
[424] Tetteh J, Suzuki T, Metcalfe E, Howells S. Quantitative structure–relationships for the estimation of boiling point and flash point using a radial basis function neural network. J Chem Inf Comput Sci 1999;39:491–507.
[425] Katritzky AR, Stoyanova-Slavova IB, Dobchev DA, Karelson M. QSPR Modeling of flash points: An update. J Mol Graphics Modell 2007;26(2):529–36.
[426] Gharagheizi F, Alamdari RF, Angaji MT. A new neural network–goup contribution method for estimation of flash point temperature of pure components. Energy Fuels 2008;22(3):1628–35.
[427] Saldana DA, Starck L, Mougin P, Rousseau B, Pidol L, Jeuland N, Creton B. Flash point and cetane number predictions for fuel compounds using quantitative structure property relationship (QSPR) methods. Energy Fuels 2011;25(9): 3900–8.
[428] Saldana DA, Starck L, Mougin P, Rousseau B, Ferrando N, Creton B. Prediction of density and viscosity of biofuel compounds using machine learning methods. Energy Fuels 2012;26(4):2416–26.
[429] Saldana DA, Starck L, Mougin P, Rousseau B, Creton B. On the rational formulation of alternative fuels: melting point and net heat of combustion predictions for fuel compounds using machine learning methods. SAR QSAR Environ Res 2013;24(4):259–77.
[430] Mitchell JBO. Machine learning methods in chemoinformatics. WIREs Comput Mol Sci 2014;4:468–81.
[431] de Oliveira FM, de Carvalho LS, Teixeira LSG, Fontes CH, Lima KMG, Câmara ABF, Araújo HOM, Sales RV. Predicting cetane index, flash point, and content sulfur of diesel–biodiesel blend using an artificial neural network model. Energy Fuels 2017;31(4):3913–20.
[432] Abdul Jameel AG, van Oudenhoven V, Emwas A-H, Sarathy SM. Predicting octane number using nuclear magnetic resonance spectroscopy and artificial neural networks. Energy Fuels 2018;32(5):6309–29.
[433] Yalamanchi KK, van Oudenhoven VCO, Tutino F, Monge-Palacios M, Alshehri A, Gao X, Sarathy SM. Machine learning to predict standard enthalpy of formation of hydrocarbons. J Phys Chem A 2019;123(38):8305–13.
[434] Ardabili S, Mosavi A, Várkonyi-Kóczy AR. Systematic review of deep learning and machine learning models in biofuels research. In: Várkonyi-Kóczy AR, editor. Engineering for Sustainable Future. Springer; 2020. p. 19–32.
[435] St. John PC, Kairys P, Das DD, McEnally CS, Pfefferle LD, Robichaud DJ, Nimlos MR, Zigler BT, McCormick RL, Foust TD, Bomble YJ, Kim S. A quantitative model for the prediction of sooting tendency from molecular structure. Energy Fuels 2017;31(9):9983–90.
[436] Miraboutalebi SM, Kazemi P, Bahrami P. Fatty acid methyl ester (FAME) composition used for estimation of biodiesel cetane number employing random forest and artificial neural networks: A new approach. Fuel 2016;166:143–51.
[437] Kessler T, St. John PC, Zhu J, McEnally CS, Pfefferle LD, Mack JH. A comparison of computational models for predicting yield sooting index. Proc Combust Inst 2021;38:1385–93.
[438] Wu Z, Pan S, Chen F, Long G, Zhang C, Yu PS. A comprehensive survey on graph neural networks. IEEE Trans Neural Netw Learn Syst 2021;32(1):4–24.
[439] Coley CW, Barzilay R, Green WH, Jaakkola TS, Jensen KF. Convolutional embedding of attributed molecular graphs for physical property prediction. J Chem Inf Model 2017;57:1757–72.
[440] Grambow CA, Pattanaik L, Green WH. Deep learning of activation energies. J Phys Chem Lett 2020;11(8):2992–7.
[441] Bengio Y, Courville A, Vincent P. Representation learning: A review and new perspectives. IEEE Trans Pattern Anal Mach Intell 2013;35(8):1798–828.
[442] Rupp M, Tkatchenko A, Müller K-R, von Lilienfeld OA. Fast and accurate modeling of molecular atomization energies with machine learning. Phys Rev Lett 2012;108:058301.
[443] Kotliar G, Savrasov SY, Haule K, Oudovenko VS, Parcollet O, Marianetti CA. Electronic structure calculations with dynamical mean-field theory. Rev Mod Phys 2006;78:865–951.
[444] Hansen K, Montavon G, Biegler F, Fazli S, Rupp M, Scheffler M, von Lilienfeld OA, Tkatchenko A, Müller K-R. Assessment and validation of machine learning methods for predicting molecular atomization energies. J Chem Theory Comput 2013;9(8):3404–19.
[445] Valsecchi C, Grisoni F, Consonni V, Ballabio D. Consensus versus individual QSARs in classification: Comparison on a large-scale case study. J Chem Inf Model 2020;60:1215–23.
[446] Hu L, Wang X, Wong L, Chen G. Combined first-principles calculation and neural-network correction approach for heat of formation. J Chem Phys 2003;119: 11501–7.

[447] Wu J, Xu X. The X1 method for accurate and efficient prediction of heats of formation. J Chem Phys 2007;127:214105.
[448] Sun J, Wu J, Song T, Hu L, Shan K, Chen G. Alternative approach to chemical accuracy: A neural networks-based first-principles method for heat of formation of molecules made of H, C, N, O, F, S, and Cl. J Phys Chem A 2014;118(39):9120–31.
[449] Li Y-P, Han K, Grambow CA, Green WH. Self-evolving machine: A continuously improving model for molecular thermochemistry. J Phys Chem A 2019;123(10):2142–52.
[450] Wu Z, Ramsundar B, Feinberg E, Gomes J, Geniesse C, Pappu AS, et al. Moleculenet: A benchmark for molecular machine learning. Chem Sci 2018;9:513–30.
[451] Analysis of kinetic reaction mechanisms. In: Turányi T, Tomlin AS, editors. Analysis of Kinetic Reaction Mechanisms. Springer; 2015.
[452] Li S, Yang B, Qi F. Accelerate global sensitivity analysis using artificial neural network algorithm: Case studies for combustion kinetic model. Combust Flame 2016;168:53–64.
[453] An J, He G, Qin F, Li R, Huang Z. A new framework of global sensitivity analysis for the chemical kinetic model using PSO-BPNN. Comput Chem Eng 2018;112:154–64.
[454] Wang J, Zhou Z, Lin K, Law CK, Yang B. Facilitating Bayesian analysis of combustion kinetic models with artificial neural network. Combust Flame 2020;213:87–97.
[455] Messerly RA, Rahimi MJ, St. John PC, Luecke JH, Park J-W, Huq NA, Foust TD, Lu T, Zigler BT, McCormick RL, Kim S. Towards quantitative prediction of ignition-delay-time sensitivity on fuel-to-air equivalence ratio. Combust Flame 2020;214:103–15.
[456] Han W, Sun Z, Scholtissek A, Hasse C. Machine learning of ignition delay times under dual-fuel engine conditions. Fuel 2021;288:119650.
[457] Ranzi E, Frassoldati A, Grana R, Cuoci A, Faravelli T, Kelley AP, Law CK. Hierarchical and comparative kinetic modeling of laminar flame speeds of hydrocarbon and oxygenated fuels. Prog Energy Combust Sci 2012;38:468–501.
[458] Curran HJ. Developing detailed chemical kinetic mechanisms for fuel combustion. Proc Combust Inst 2019;37:57–81.
[459] Wang H, Xu R, Wang K, Bowman CT, Hanson RK, Davidson DF, et al. A physics-based approach to modeling real-fuel combustion chemistry – I. Evidence from experiments, and thermodynamic, chemical kinetic and statistical considerations. Combust Flame 2018;193:502–19.
[460] Xu R, Wang K, Banerjee S, Shao J, Parise T, Zhu Y, et al. A physics-based approach to modeling real-fuel combustion chemistry – II. Reaction kinetic models of jet and rocket fuels. Combust Flame 2018;193:520–37.
[461] Ranade R, Alqahtani S, Farooq A, Echekki T. An ANN based hybrid chemistry framework for complex fuels. Fuel 2019;241:625–36.
[462] Ranade R, Alqahtani S, Farooq A, Echekki T. An extended hybrid chemistry framework for complex hydrocarbon fuels. Fuel 2019;251:276–84.
[463] Alqahtani S, Echekki T. A data-based hybrid model for complex fuel chemistry acceleration at high temperatures. Combust Flame 2021;223:142–52.
[464] Chang Y, Jia M, Niu B, Xu Z, Liu Z, Li Y, Xie M. Construction of a skeletal oxidation mechanism of *n*-pentanol by integrating decoupling methodology, genetic algorithm, and uncertainty quantification. Combust Flame 2018;194:15–27.
[465] Cheng J, Zou C, Lin Q, Liu S, Wang Y, Liu Y. High-temperature oxidation of methyl isopropyl ketone: A shock tube experiment and a kinetic model. Combust Flame 2019;209:376–88.
[466] Frenklach M. Systematic optimization of a detailed kinetic model using a methane ignition example. Combust Flame 1984;58:69–72.
[467] Storn R, Price K. Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces. J Glob Optim 1997;11:341–59.
[468] Qin AK, Huang VL, Suganthan PN. Differential evolution algorithm with strategy adaptation for global numerical optimization. IEEE Trans Evol Comput 2009;13(2):398–417.
[469] Bongard J, Lipson H. Automated reverse engineering of nonlinear dynamical systems. Proc Natl Acad Sci USA 2007;104(24):9943–8.
[470] Schmidt M, Lipson H. Distilling free-form natural laws from experimental data. Science 2009;324(5923):81–5.
[471] Brunton SL, Proctor JL, Kutz JN. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. Proc Natl Acad Sci USA 2016;113(15):3932–7.
[472] Rudy SH, Brunton SL, Proctor JL, Kutz JN. Data-driven discovery of partial differential equations. Sci Adv 2017;3(4):e1602614.
[473] Champion K, Lusch B, Kutz JN, Brunton SL. Data-driven discovery of coordinates and governing equations. Proc Natl Acad Sci USA 2019;116(45):22445–51.
[474] Hoffmann M, Fröhner C, Noé F. Reactive SINDy: Discovering governing reactions from concentration data. J Chem Phys 2019;150:025101.
[475] Langary D, Nikoloski Z. Inference of chemical reaction networks based on concentration profiles using an optimization framework. Chaos 2019;29(11):113121.
[476] Burnham SC, Searson DP, Willis MJ, Wright AR. Inference of chemical reaction networks. Chem Eng Sci 2008;63(4):862–73.
[477] Ji W, Deng S. Autonomous discovery of unknown reaction pathways from data by chemical reaction neural network. J Phys Chem A 2021;125(4):1082–92.
[478] Maas U, Thévenin D. Correlation analysis of direct numerical simulation data of turbulent non-premixed flames. Symp (Int) Combust 1998;27:1183–9.
[479] Parente A, Sutherland JC, Dally BB, Tognotti L, Smith PJ. Investigation of the MILD combustion regime *via* Principal Component Analysis. Proc Combust Inst 2011;33:3333–41.
[480] Coussement A, Gicquel O, Parente A. Kernel density weighted principal component analysis of combustion processes. Combust Flame 2012;159:2844–55.
[481] Mirgolbabaei H, Echekki T. Nonlinear reduction of combustion composition space with kernel principal component analysis. Combust Flame 2014;161:118–26.
[482] Mirgolbabaei H, Echekki T, Smaoui N. A nonlinear principal component analysis approach for turbulent combustion composition space. Int J Hydrog Energy 2014;39:4622–33.
[483] Yang Y, Pope SB, Chen JH. Empirical low-dimensional manifolds in composition space. Combust Flame 2013;160:1967–80.
[484] Biglari A, Sutherland JC. A filter-independent model identification technique for turbulent combustion modeling. Combust Flame 2012;159:1960–70.
[485] Friedman JH. Multivariate adaptive regression splines. Ann Stat 1991;19(1):1–67.
[486] Mirgolbabaei H, Echekki T. A novel principal component analysis-based acceleration scheme for LES–ODT: An a priori study. Combust Flame 2013;160:898–908.
[487] Echekki T, Mirgolbabaei H. Principal component transport in turbulent combustion: A posteriori analysis. Combust Flame 2015;162:1919–33.
[488] Isaac BJ, Thornock JN, Sutherland J, Smith PJ, Parente A. Advanced regression methods for combustion modelling using principal components. Combust Flame 2015;162:2592–601.
[489] Malik MR, Isaac BJ, Coussement A, Smith PJ, Parente A. Principal component analysis coupled with nonlinear regression for chemistry reduction. Combust Flame 2018;187:30–41.
[490] Coussement A, Gicquel O, Parente A. MG-local-PCA method for reduced order combustion modeling. Proc Combust Inst 2013;34:1117–23.
[491] Isaac BJ, Coussement A, Gicquel O, Smith PJ, Parente A. Reduced-order PCA models for chemical reacting flows. Combust Flame 2014;161:2785–800.
[492] D'Alessio G, Parente A, Stagni A, Cuoci A. Adaptive chemistry via pre-partitioning of composition space and mechanism reduction. Combust Flame 2020;211:68–82.
[493] Biglari A, Sutherland JC. An a-posteriori evaluation of principal component analysis-based models for turbulent combustion simulations. Combust Flame 2015;162:4025–35.
[494] Malik MR, Obando Vega P, Coussement A, Parente A. Combustion modeling using Principal Component Analysis: A posteriori validation on Sandia flames D, E and F. Proc Combust Inst 2021;38:2635–43.
[495] Pope SB. Computationally efficient implementation of combustion chemistry using *in situ* adaptive tabulation. Combust Theory Model 1997;1(1):41–63.
[496] Tonse SR, Moriarty NW, Brown NJ, Frenklach M. PRISM: Piecewise reusable implementation of solution mapping. An economical strategy for chemical kinetics. Isr J Chem 1999;39:97–106.
[497] Ribert G, Gicquel O, Darabiha N, Veynante D. Tabulation of complex chemistry based on self-similar behavior of laminar premixed flames. Combust Flame 2006;146:649–64.
[498] Veynante D, Fiorina B, Domingo P, Vervisch L. Using self-similar properties of turbulent premixed flames to downsize chemical tables in high-performance numerical simulations. Combust Theory Model 2008;12(6):1055–88.
[499] Jones WP, Kollmann W. Multi-scalar pdf transport equations for turbulent diffusion flames. In: Durst F, Launder BE, Lumley JL, Schmidt FW, Whitelaw JH, editors. Turbulent Shear Flows 5. Springer; 1987. p. 296–309.
[500] Xia G, Li D, Merkle CL. Consistent properties reconstruction on adaptive Cartesian meshes for complex fluids computations. J Comput Phys 2007;225:1175–97.
[501] Ihme M, See YC. Prediction of autoignition in a lifted methane/air flame using an unsteady flamelet/progress variable model. Combust Flame 2010;157:1850–62.
[502] Liu Z, Liang J, Pan Y. Construction of thermodynamic properties look-up table with block-structured adaptive mesh refinement method. J Thermophys Heat Trans 2014;28(1):50–8.
[503] Lee S, Devaud C. Application of conditional source-term estimation to two turbulent non-premixed methanol flames. Combust Theory Model 2016;20(5):765–97.
[504] Bode M, Collier N, Bisetti F, Pitsch H. Adaptive chemistry lookup tables for combustion simulations using optimal B-spline interpolants. Combust Theory Model 2019;23(4):674–99.
[505] Hossain M, Jones JC, Malalasekera W. Modelling of a bluff-body nonpremixed flame using a coupled radiation/flamelet combustion model. Flow Turbul Combust 2001;67:217–34.
[506] Fiorina B, Baron R, Gicquel O, Thevenin D, Carpentier S, Darabiha N. Modelling non-adiabatic partially premixed flames using flame-prolongation of ILDM. Combust Theory Model 2003;7:449–70.
[507] Ketelheun A, Kuenne G, Janicka J. Heat transfer modeling in the context of large eddy simulation of premixed combustion with tabulated chemistry. Flow Turbul Combust 2013;91:867–93.
[508] Proch F, Kempf AM. Modeling heat loss effects in the large eddy simulation of a model gas turbine combustor with premixed flamelet generated manifolds. Proc Combust Inst 2015;35:3337–45.
[509] Ma PC, Wu H, Ihme M, Hickey J-P. Nonadiabatic flamelet formulation for predicting wall heat transfer in rocket engines. AIAA J 2018;56:2336–49.
[510] Zips J, Traxinger C, Pfitzner M. Time-resolved flow field and thermal loads in a single-element GOx/GCH_4 rocket combustor. Int J Heat Mass Transf 2019;143:118474.
[511] Ihme M, Pitsch H. Modeling of radiation and NO formation in turbulent non-premixed flames using a flamelet/progress variable formulation. Phys Fluids 2008;20:055110.
[512] Mueller ME, Pitsch H. LES model for sooting turbulent nonpremixed flames. Combust Flame 2012;159:2166–80.
[513] Perakis N, Haidn OJ, Ihme M. Investigation of CO recombination in the boundary layer of CH_4/O_2 rocket engines. Proc Combust Inst 2021;38:6403–11.

[514] Hasse C, Peters N. A two mixture fraction flamelet model applied to split injection in a DI Diesel engine. Proc Combust Inst 2005;30:2755–62.
[515] Ihme M, See YC. LES flamelet modeling of a three-stream MILD combustor: analysis of flame sensitivity to scalar inflow conditions. Proc Combust Inst 2010; 33:1309–17.
[516] Ihme M, Zhang J, G H, Dally B. Large-eddy simulation of a jet-in-hot-coflow burner operating in the oxygen-diluted combustion regime. Flow Turbul Combust 2012;89:449–64.
[517] Chen Y, Ihme M. Large-eddy simulation of a piloted premixed jet burner. Combust Flame 2013;160:2896–910.
[518] Perry BA, Mueller ME. Joint probability density function models for multiscalar turbulent mixing. Combust Flame 2018;193:344–62.
[519] Pitsch H, Riesmeier E, Peters N. Unsteady flamelet modeling of soot formation in turbulent diffusion flames. Combust Sci Tech 2000;158:389–406.
[520] Ameen MM, Kundu P, Som S. Novel tabulated combustion model approach for lifted spray flames with large eddy simulations. SAE Int J Engines 2016;9: 2056–65.
[521] Baba Y, Kurose R. Analysis and flamelet modelling for spray combustion. J Fluid Mech 2008;612:45–79.
[522] Ge H-W, Gutheil E. Simulation of a turbulent spray flame using coupled PDF gas phase and spray flamelet modeling. Combust Flame 2008;153:173–85.
[523] Franzelli B, Fiorina B, Darabiha N. A tabulated chemistry method for spray combustion. Proc Combust Inst 2013;34:1659–66.
[524] Franzelli B, Vie A, Ihme M. On the generalisation of the mixture fraction to a monotonic mixing-describing variable for the flamelet formulation of spray flames. Combust Theory Model 2015;19:773–806.
[525] Wen X, Debiagi P, Stein OT, Kronenburg A, Kempf AM, Hasse C. Flamelet tabulation methods for solid fuel combustion with fuel-bound nitrogen. Combust Flame 2019;209:155–66.
[526] Vascellari M, Xu H, Hasse C. Flamelet modeling of coal particle ignition. Proc Combust Inst 2013;34:2445–52.
[527] Watanabe J, Yamamoto K. Flamelet model for pulverized coal combustion. Proc Combust Inst 2015;35:2315–22.
[528] Christo FC, Masri AR, Nebot EM. Artificial neural network implementation of chemistry with PDF simulation of H_2/CO_2 flames. Combust Flame 1996;106: 406–27.
[529] Blasco JA, Fueyo N, Dopazo C, Ballester J. Modelling the temporal evolution of a reduced combustion chemical system with an artificial neural network. Combust Flame 1998;113:38–52.
[530] Blasco JA, Fueyo N, Larroya JC, Dopazo C, Chen JY. Single-step time-integrator of a methane-air chemical system using artificial neural networks. Comput Chem Eng 1999;23(9):1127–33.
[531] Blasco JA, Fueyo N, Dopazo C, Chen JY. A self-organizing-map approach to chemistry representation in combustion applications. Combust Theory Model 2000;4:61–76.
[532] Ranade R, Li G, Li S, Echekki T. An efficient machine-learning approach for PDF tabulation in turbulent combustion closure. Combust Sci Tech 2019;193(7): 1258–77.
[533] Chen J-Y, Blasco J, Fueyo N, Dopazo C. An economical strategy for storage of chemical kinetics: Fitting *in situ* adaptive tabulation with artificial neural networks. Proc Combust Inst 2000;28:115–21.
[534] Ihme M, Marsden AL, Pitsch H. Generation of optimal artificial neural networks using a pattern search algorithm: Application to approximation of chemical systems. Neural Comput 2008;20(2):573–601.
[535] Ihme M, Schmitt C, Pitsch H. Optimal artificial neural networks and tabulation methods for chemistry representation in LES of a bluff-body swirl-stabilized flame. Proc Combust Inst 2009;32:1527–35.
[536] Audet C, Dennis Jr JE. Pattern search algorithms for mixed variable programming. SIAM J Optimiz 2000;11(3):573–94.
[537] Ihme M. Construction of optimal artificial neural network architectures for application to chemical systems: Comparison of generalized pattern search method and evolutionary algorithm. In: Hui CLP, editor. Artificial Neural Networks Application. IntechOpen; 2011. p. 125–50.
[538] Nguyen H-P, Liu J, Zio E. A long-term prediction approach based on long short-term memory neural networks with automatic parameter optimization by Tree-structured Parzen Estimator and applied to time-series data of NPP steam generators. Appl Soft Comput 2020;89:106116.
[539] Li L, Jamieson K, DeSalvo G, Rostamizadeh A, Talwalkar A. Hyperband: A novel bandit-based approach to hyperparameter optimization. J Mach Learn Res 2018; 18:1–52.
[540] McGibbon RT, Hernández CX, Harrigan MP, Kearnes S, Sultan MM, Jastrzebski S, et al. Osprey: Hyperparameter optimization for machine learning. J Open Source Softw 2016;1(5):1–34.
[541] Flemming F, Sadiki A, Janicka J. LES using artificial neural networks for chemistry representation. Prog Comput Fluid Dyn 2005;5(7):375–85.
[542] Kempf A, Flemming F, Janicka J. Investigation of lengthscales, scalar dissipation, and flame orientation in a piloted diffusion flame by LES. Proc Combust Inst 2005;30:557–65.
[543] Sen BA, Menon S. Linear eddy mixing based tabulation and artificial neural networks for large eddy simulations of turbulent flames. Combust Flame 2010; 157:62–74.
[544] Kerstein AR. A linear-eddy model of turbulent scalar transport and mixing. Combust Sci Tech 1988;60:391–421.
[545] Kerstein AR. Linear-eddy modelling of turbulent transport. Part 6. Microstructure of diffusive scalar mixing fields. J Fluid Mech 1991;231:361–94.
[546] Sen BA, Hawkes ER, Menon S. Large eddy simulation of extinction and reignition with artificial neural networks based chemical kinetics. Combust Flame 2010; 157:566–78.
[547] Dalakoti DK, Wehrfritz A, Savard B, Day MS, Bell JB, Hawkes ER. An *a priori* evaluation of a principal component and artificial neural network based combustion model in diesel engine conditions. Proc Combust Inst 2021;38: 2701–9.
[548] Chatzopoulos AK, Rigopoulos S. A chemistry tabulation approach via rate-controlled constrained equilibrium (RCCE) and artificial neural networks (ANNs), with application to turbulent non-premixed $CH_4/H_2/N_2$ flames. Proc Combust Inst 2013;34:1465–73.
[549] Franke LLC, Chatzopoulos AK, Rigopoulos S. Tabulation of combustion chemistry via artificial neural networks (ANNs): Methodology and application to LES-PDF simulation of Sydney flame L. Combust Flame 2017;185:245–60.
[550] Owoyele O, Kundu P, Ameen MM, Echekki T, Som S. Application of deep artificial neural networks to multi-dimensional flamelet libraries and spray flames. Int J Engine Res 2020;21(1):151–68.
[551] Jacobs RA, Jordan MI, Nowlan SJ, Hinton GE. Adaptive mixtures of local experts. Neural Comput 1991;3(1):79–87.
[552] Owoyele O, Kundu P, Pal P. Efficient bifurcation and tabulation of multi-dimensional combustion manifolds using deep mixture of experts: An *a priori* study. Proc Combust Inst 2021;38:5889–96.
[553] Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. Commun ACM 2017;60(6):84–90.
[554] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. Proc Int Conf Learn Repr. 2015. p. 1–14.
[555] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. IEEE Conf Comput Vision Pattern Recognit 2016:770–8.
[556] Duraisamy K, Iaccarino G, Xiao H. Turbulence modeling in the age of data. Annu Rev Fluid Mech 2019;51:357–77.
[557] Tracey B, Duraisamy K, Alonso JJ. Application of supervised learning to quantify uncertainties in turbulence and combustion modeling. AIAA Pap 2013-259 2013.
[558] Wang JX, Xiao H. Data-driven CFD modeling of turbulent flows through complex structures. Int J Heat Fluid Flow 2016;62:138–49.
[559] Wu J-L, Xiao H, Paterson E. Physics-informed machine learning approach for augmenting turbulence models: A comprehensive framework. Phys Rev Fluids 2018;3:74602.
[560] Maulik R, San O, Jacob JD, Crick C. Sub-grid scale model classification and blending through deep learning. J Fluid Mech 2019;870:784–812.
[561] Chen ZX, Iavarone S, Ghiasi G, Kannan V, D'Alessio G, Parente A, Swaminathan N. Application of machine learning for filtered density function closure in MILD combustion. Combust Flame 2021;225:160–79.
[562] Lapeyre CJ, Misdariis A, Cazard N, Veynante D, Poinsot T. Training convolutional neural networks to estimate turbulent sub-grid scale reaction rates. Combust Flame 2019;203:255–64.
[563] Nikolaou ZM, Chrysostomou C, Vervisch L, Cant S. Progress variable variance and filtered rate modelling using convolutional neural networks and flamelet methods. Flow Turbul Combust 2019;103:485–501.
[564] Dong C, Loy CC, He K, Tang X. Image super-resolution using deep convolutional networks. IEEE Trans Pattern Anal Mach Intell 2016;38(2):295–307.
[565] Fukami K, Fukagata K, Taira K. Super-resolution reconstruction of turbulent flows with machine learning. J Fluid Mech 2019;870:106–20.
[566] Wang Q, Ihme M. Regularized deconvolution method for turbulent combustion modeling. Combust Flame 2017;176:125–42.
[567] Wang Q, Ihme M. A regularized deconvolution method for turbulent closure modeling in implicitly filtered large-eddy simulation. Combust Flame 2019;204: 341–55.
[568] Ranade R, Echekki T. A framework for data-based turbulent combustion closure: *A priori* validation. Combust Flame 2019;206:490–505.
[569] Ranade R, Echekki T. A framework for data-based turbulent combustion closure: *A posteriori* validation. Combust Flame 2019;210:279–91.
[570] Henry de Frahan MT, Yellapantula S, King R, Day MS, Grout RW. Deep learning for presumed probability density function models. Combust Flame 2019;208: 436–50.
[571] Bode M, Gauding M, Göbbert JH, Liao B, Jitsev J, Pitsch H. Towards prediction of turbulent flows at high Reynolds numbers using high performance computing data and deep learning. ISC High Perform Comput 2018:614–23.
[572] Schoepplein M, Weatheritt J, Sandberg R, Talei M, Klein M. Application of an evolutionary algorithm to LES modelling of turbulent transport in premixed flames. J Comput Phys 2018;374:1166–79.
[573] Ferreira C. Gene expression programming: A new adaptive algorithm for solving problems. Complex Syst 2001;13:87–129.
[574] Clark RA, Ferziger JH, Reynolds WC. Evaluation of subgrid-scale models using an accurately simulated turbulent flow. J Fluid Mech 1979;91:1–16.
[575] Chung WT, Mishra AA, Ihme M. Interpretable data-driven methods for subgrid-scale closure in LES for transcritical LOX/GCH4 combustion. Combust Flame 2021.In press
[576] Yoshizawa A. Statistical theory for compressible turbulent shear flows, with the application to subgrid modeling. Phys Fluids 1986;29(7):2152–64.
[577] Yellapantula S, Perry BA, Grout RW. Deep learning-based model for progress variable dissipation rate in turbulent premixed flames. Proc Combust Inst 2021; 38:2929–38.
[578] Seltz A, Domingo P, Vervisch L, Nikolaou ZM. Direct mapping from LES resolved scales to filtered-flame generated manifolds using convolutional neural networks. Combust Flame 2019;210:71–82.

[579] Nikolaou ZM, Chrysostomou C, Minamoto Y, Vervisch L. Evaluation of a neural network-based closure for the unresolved stresses in turbulent premixed V-flames. Flow Turbul Combust 2020;106:331–56.
[580] Wang X, Yu K, Wu S, Gu J, Liu Y, Dong C, Qiao Y, Loy CC. ESRGAN: Enhanced super-resolution generative adversarial networks. Eur Conf Comput Vis 2018: 63–79.
[581] Yao S, Wang B, Kronenburg A, Stein OT. Modeling of sub-grid conditional mixing statistics in turbulent sprays using machine learning methods. Phys Fluids 2020; 32(11):115124.
[582] Yao S, Wang B, Kronenburg A, Stein OT. Conditional scalar dissipation rate modeling for turbulent spray flames using artificial neural networks. Proc Combust Inst 2021;38:3371–8.
[583] Liang L, Naik CV, Puduppakkam K, Wang C, Modak A, Meeks E, Ge HW, Reitz R, Rutland C. Efficient simulation of diesel engine combustion using realistic chemical kinetics in CFD. SAE Techn Pap 2010-01-0178 2010.
[584] Perini F. High-dimensional, unsupervised cell clustering for computationally efficient engine simulations with detailed combustion chemistry. Fuel 2013;106: 344–56.
[585] Torres DJ, Trujillo MF. KIVA-4: An unstructured ALE code for compressible gas flow with sprays. J Comput Phys 2006;219(2):943–75.
[586] Perlman C, Frojd K, Seidel L, Tuner M, Mauss F. A fast tool for predictive IC engine in-cylinder modelling with detailed chemistry. SAE Techn Pap 2012-01-1074 2012.
[587] Wu H, See YC, Wang Q, Ihme M. A Pareto-efficient combustion framework with submodel assignment for predicting complex flame configurations. Combust Flame 2015;162:4208–30.
[588] Wu H, Ma PC, Jaravel T, Ihme M. Pareto-efficient combustion modeling for improved CO-emission prediction in LES of a piloted turbulent dimethyl ether jet flame. Proc Combust Inst 2019;37:2267–76.
[589] Chung WT, Mishra AA, Perakis N, Ihme M. Data-assisted combustion simulations with dynamic submodel assignment using random forests. Combust Flame 2021; 227:172–85.
[590] Lapointe S, Mondal S, Whitesides RA. Data-driven selection of stiff chemistry ODE solver in operator-splitting schemes. Combust Flame 2020;220:133–43.
[591] Kalogirou SA. Artificial intelligence for the modeling and control of combustion processes: A review. Prog Energy Combust Sci 2003;29:515–66.
[592] Alizadeh R, Allen JK, Mistree F. Managing computational complexity using surrogate models: A critical review. Res Eng Design 2020;31:275–98.
[593] Adewole BZ, Abidakun OA, Asere AA. Artificial neural network prediction of exhaust emissions and flame temperature in LPG (liquefied petroleum gas) fueled low swirl burner. Energy 2013;61:606–11.
[594] Baklacioglu T. Predicting the fuel flow rate of commercial aircraft via multilayer perceptron, radial basis function and ANFIS artificial neural networks. Aeronaut J 2021;125(1285):453–71.
[595] Hao Z, Qian X, Cen K, Jianren F. Optimizing pulverized coal combustion performance based on ANN and GA. Fuel Process Tech 2004;85(2):113–24.
[596] Bekat T, Erdogan M, Inal F, Genc A. Prediction of the bottom ash formed in a coal-fired power plant using artificial neural networks. Energy 2012;45(1):882–7.
[597] Safdarnejad SM, Tuttle JF, Powell KM. Dynamic modeling and optimization of a coal-fired utility boiler to forecast and minimize NO_x and CO emissions simultaneously. Comput Chem Eng 2019;124:62–79.
[598] Malaczynski GW, Mueller M, Pfeiffer J, Cabush D, Hoyer K. Replacing volumetric efficiency calibration look-up tables with artificial neural network-based algorithm for variable valve actuation. SAE Techn Pap 2010-01-0158 2010.
[599] Martínez-Morales JD, Palacios E, Veláazquez Carrillo GA. Modeling of internal combustion engine emissions by LOLIMOT algorithm. Procedia Technol 2012;3: 251–8.
[600] Mohamed Ismail H, Ng HK, Queck CW, Gan S. Artificial neural networks modelling of engine-out responses for a light-duty diesel engine fuelled with biodiesel blends. Appl Energy 2012;92:769–77.
[601] Ghanbari M, Najafi G, Ghobadian B, Mamat R, Noor MM, Moosavian A. Support vector machine to predict diesel engine performance and emission parameters fueled with nano-particles additive to diesel fuel. IOP Conf Ser-Mater Sci Eng 2015;100(1):012069.
[602] Aghbashlo M, Shamshirband S, Tabatabaei M, Yee PL, Larimi YN. The use of ELM-WT (extreme learning machine with wavelet transform algorithm) to predict exergetic performance of a DI diesel engine running on diesel/biodiesel blends containing polymer waste. Energy 2016;94:443–56.
[603] Niu X, Wang H, Hu S, Yang C, Wang Y. Multi-objective online optimization of a marine diesel engine using NSGA-II coupled with enhancing trained support vector machine. Appl Therm Eng 2018;137:218–27.
[604] Berger B, Rauscher F, Lohmann B. Analysing Gaussian processes for stationary black-box combustion engine modelling. IFAC Proc Vol 2011;44:10633–40.
[605] Wang YY, He Y, Rajagopalan S. Design of engine-out virtual NO_x sensor using neural networks and dynamic system identification. SAE Int J Engines 2011;4(1): 828–36.
[606] Xiao B, Wang S, Prucka RG. A semi-physical artificial neural network for feed forward ignition timing control of multi-fuel SI engines. SAE Techn Pap 2013-01-0324 2013.
[607] Arsie I, Pianese C, Sorrentino M. Development of recurrent neural networks for virtual sensing of NO_x emissions in internal combustion engines. SAE Int J Fuels Lubr 2010;2(2):354–61.
[608] Li H, Butts K, Zaseck K, Liao-McPherson D, Kolmanovsky I. Emissions modeling of a light-duty diesel engine for model-based control design using multi-layer perceptron neural networks. SAE Techn Pap 2017-01-0601 2017.
[609] Huang G-B, Zhu Q-Y, Siew C-K. Extreme learning machine: A new learning scheme of feedforward neural networks. IEEE Int Conf Neural Netw 2004;2: 985–90.
[610] Huang G, Huang G-B, Song S, You K. Trends in extreme learning machines: A review. Neural Netw 2015;61:32–48.
[611] Shamshirband S, Tabatabaei M, Aghbashlo M, Yee PL, Petković D. Support vector machine-based exergetic modelling of a DI diesel engine running on biodiesel-diesel blends containing expanded polystyrene. Appl Therm Eng 2016;94:727–47.
[612] Wong KI, Wong PK, Cheung CS, Vong CM. Modeling and optimization of biodiesel engine performance using advanced machine learning methods. Energy 2013;55: 519–28.
[613] Vaughan A, Bohac SV. Real-time, adaptive machine learning for non-stationary, near chaotic gasoline engine combustion time series. Neural Netw 2015;70: 18–26.
[614] Silitonga AS, Masjuki HH, Ong HC, Sebayang AH, Dharma S, Kusumo F, Siswantoro J, Milano J, Daud K, Mahlia TMI, Chen WH, Sugiyanto B. Evaluation of the engine performance and exhaust emissions of biodiesel-bioethanol-diesel blends using kernel-based extreme learning machine. Energy 2018;159:1075–87.
[615] Wong KI, Wong PK, Cheung CS, Vong CM. Modelling of diesel engine performance using advanced machine learning methods under scarce and exponential data set. Appl Soft Comput 2013;13(11):4428–41.
[616] Moiz AA, Pal P, Probst D, Pei Y, Zhang S, Som Y, Kodavasal J. A machine learning-genetic algorithm (ML-GA) approach for rapid optimization using high-performance computing. SAE Int J Commer Veh 2018;11(5):291–306.
[617] Badra J, Khaled F, Sim J, Pei Y, Viollet Y, Pal P, Futterer C, Brenner M, Som S, Farooq A, Chang J. Combustion system optimization of a light-duty GCI engine using CFD and machine learning. SAE Techn Pap 2020-01-1313 2020.
[618] Petrarolo A, Ruettgers A, Kobald M. Data clustering of hybrid rocket combustion flame. AIAA Pap 2019–4193 2019.
[619] Cao Y, Kaiser E, Borée J, Noack BR, Thomas L, Guilain S. Cluster-based analysis of cycle-to-cycle variations: application to internal combustion engines. Exp Fluids 2014;55:1837.
[620] Xiao L, Pang W, Qin Z, Li J, Fu X, Fan X. Cluster analysis of Al agglomeration in solid propellant combustion. Combust Flame 2019;203:386–96.
[621] Nakaya S, Omi K, Okamoto T, Ikeda Y, Zhao C, Tsue M, Taguchi H. Instability and mode transition analysis of a hydrogen-rich combustion in a model afterburner. Proc Combust Inst 2021;38:5933–42.
[622] Liu Y, Fan Y, Chen J. Flame images for oxygen content prediction of combustion systems using DBN. Energy Fuels 2017;31(8):8776–83.
[623] Wan K, Hartl S, Vervisch L, Domingo P, Barlow RS, Hasse C. Combustion regime identification from machine learning trained by Raman/Rayleigh line measurements. Combust Flame 2020;219:268–74.
[624] Iten R, Metger T, Wilming H, del Rio L, Renner R. Discovering physical concepts with neural networks. Phys Rev Lett 2020;124:010508.
[625] Barwey S, Hassanaly M, Raman V, Steinberg A. Using machine learning to construct velocity fields from OH-PLIF images. arXiv Preprint 2019;1909.13669.
[626] An Q, Steinberg AM. The role of strain rate, local extinction, and hydrodynamic instability on transition between attached and lifted swirl flames. Combust Flame 2019;199:267–78.
[627] Hanuschkin A, Zündorf S, Schmidt M, Welch C, Schorr J, Peters S, Dreizler A, Böhm B. Investigation of cycle-to-cycle variations in a spark-ignition engine based on a machine learning analysis of the early flame kernel. Proc Combust Inst 2021; 38:5751–9.
[628] Kodavasal J, Abdul Moiz A, Ameen M, Som S. Using machine learning to analyze factors determining cycle-to-cycle variation in a spark-ignited gasoline engine. J Energy Resour Technol 2018;140:102204.
[629] Lei Y, Yang B, Jiang X, Jia F, Li N, Nandi AK. Applications of machine learning to machine fault diagnosis: A review and roadmap. Mech Syst Signal Process 2020; 138:106587.
[630] Yadav S, Kalra PK. Automatic fault diagnosis of internal combustion engine based on spectrogram and artificial neural network. Proc. 10th WSEAS Int. Conf. Rob. Control Manuf. Technol.. 2010. p. 101–7.
[631] Wu JD, Huang CK, Chang YW, Shiao YJ. Fault diagnosis for internal combustion engines using intake manifold pressure and artificial neural network. Expert Syst Appl 2010;37(2):949–58.
[632] Xi W, Li Z, Tian Z, Duan Z. A feature extraction and visualization method for fault detection of marine diesel engines. Measurement 2018;116:429–37.
[633] Jafarian K, Mobin M, Jafari-Marandi R, Rabiei E. Misfire and valve clearance faults detection in the combustion engines based on a multi-sensor vibration signal monitoring. Measurement 2018;128:527–36.
[634] Wang YS, Ma QH, Zhu Q, Liu XT, Zhao LH. An intelligent approach for engine fault diagnosis based on Hilbert-Huang transform and support vector machine. Appl Acoust 2014;75:1–9.
[635] Wong PK, Zhong J, Yang Z, Vong CM. Sparse Bayesian extreme learning committee machine for engine simultaneous fault diagnosis. Neurocomputing 2016;174:331–43.
[636] Devasenapati SB, Sugumaran V, Ramachandran KI. Misfire identification in a four-stroke four-cylinder petrol engine using decision tree. Expert Syst Appl 2010; 37(3):2150–60.
[637] Sharma A, Sugumaran V, Babu Devasenapati S. Misfire detection in an IC engine using vibration signal and decision tree algorithms. Measurement 2014;50: 370–80.
[638] Kuzhagaliyeva N, Thabet A, Singh E, Ghanem B, Sarathy SM. Using deep neural networks to diagnose engine pre-ignition. Proc Combust Inst 2021;38:5915–22.

[639] Vincent P, Larochelle H, Lajoie I, Bengio Y, Manzagol P-A. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. J Mach Learn Res 2010;11:3371–408.
[640] Hinton GE, Osindero S, Teh Y-W. A fast learning algorithm for deep belief nets. Neural Comput 2006;18(7):1527–54.
[641] Yan W, Yu L. On accurate and reliable anomaly detection for gas turbine combustors: A deep learning approach. Annu Conf Progn Health Manag Soc 2015; 6:1–8.
[642] Akintayo A, Lore KG, Sarkar S, Sarkar S. Prognostics of combustion instabilities from hi-speed flame video using a deep convolutional selective autoencoder. Int J Progn Health Manag 2016;7:1–14.
[643] Han Z, Hossain MM, Wang Y, Li J, Xu C. Combustion stability monitoring through flame imaging and stacked sparse autoencoder based deep neural network. Appl Energy 2020;259:114159.
[644] Johnson JM, Khoshgoftaar T. Survey on deep learning with class imbalance. J Big Data 2019;6:27.
[645] Zhou Z-H. A brief introduction to weakly supervised learning. Natl Sci Rev 2017; 5:44–53.
[646] Malikopoulos AA, Assanis DN, Papalambros PY. Real-time self-learning optimization of diesel engine calibration. J Eng Gas Turbine Power 2008;131(2): 022803.
[647] Schaefer AM, Schneegass D, Sterzing V, Udluft S. A neural reinforcement learning approach to gas turbine control. Int Jt Conf Neural Netw 2007:1691–6.
[648] Xue J, Gao Q, Ju W. Reinforcement learning for engine idle speed control. Int Conf Meas Tech Mechatron Autom 2010;2:1008–11.
[649] Stephan V, Debes K, Gross H-M, Wintrich F, Wintrich H. A new control scheme for combustion processes using reinforcement learning based on neural networks. Int J Comput Intell Appl 2001;1(2):121–36.
[650] Tsitsiklis J, Van Roy B. An analysis of temporal-difference learning with function approximation. IEEE Trans Automat Contr 1997;42(5):674–90.
[651] Cheng Y, Zou L, Zhuang Z, Sun Z, Zhang W. Deep reinforcement learning combustion optimization system using synchronous neural episodic control. Chin Control Conf 2018:8770–5.
[652] Henry de Frahan MT, Wimer NT, Yellapantula S, Grout RW. Deep reinforcement learning for dynamic control of fuel injection timing in multi-pulse compression ignition engines. Int J Engine Res 2021.In press
[653] Levine S, Kumar A, Tucker G, Fu J. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. arXiv Preprint 2020;2005.01643.
[654] Thomson N. Hazards in Industry. Butterworth-Heinemann; 2001.
[655] Strehlow RA, Baker WE. The characterization and evaluation of accidental explosions. Prog Energy Combust Sci 1976;2:27–60.
[656] Baker WE, Cox PA, Westine PS, Kulesz JJ, Strehlow RA. Explosion Hazards and Evaluation. Elsevier Science; 1983.
[657] Eckhoff R. Dust explosions in the Process Industries. 3rd. Gulf Professional Publishing; 2003.
[658] Brillinger DR. Three environmental probabilistic risk problems. Stat Sci 2003;18 (4):412–21.
[659] Thompson MP, Calkin DE. Uncertainty and risk in wildland fire management: A review. J Environ Manage 2011;92(8):1895–909.
[660] Finney MA, McHugh CW, Grenfell IC, Riley KL, Short KC. A simulation of probabilistic wildfire risk components for the continental United States. Stoch Environ Res Risk Assess 2011;25:973–1000.
[661] Quintiere JG. Fire behavior in building compartments. Proc Combust Inst 2002; 29:181–93.
[662] Dai X, Welch S, Usmani A. A critical review of “travelling fire” scenarios for performance-based structural engineering. Fire Saf J 2017;91:568–78.
[663] Gann R. Advanced technology for fire suppression in aircraft. NIST Special Publication 1069. Gaithersburg, MD: National Institute of Standards and Technology; 2007.
[664] Friedman R. Fire safety in spacecraft. Fire Mater 1996;20:235–43.
[665] Gye H-R, Seo S-K, Bach Q-V, Ha D, Lee C-J. Quantitative risk assessment of an urban hydrogen refueling station. Int J Hydrog Energy 2019;44:1288–98.
[666] Gharari R, Kazeminejad H, Mataji Kojouri N, Hedayat A. A review on hydrogen generation, explosion, and mitigation during severe accidents in light water nuclear reactors. Int J Hydrog Energy 2018;43:1939–65.
[667] Wang Q, Ping P, Zhao X, Chu G, Sun J, Chen C. Thermal runaway caused fire and explosion of lithium ion battery. J Power Sources 2012;208:210–24.
[668] Liu K, Liu Y, Lin D, Pei A, Cui Y. Materials for lithium-ion battery safety. Sci Adv 2018;4(6):eaas9820.
[669] Wang Q, Mao B, Stoliarov SI, Sun J. A review of lithium ion battery failure mechanisms and fire prevention strategies. Prog Energy Combust Sci 2019;73: 95–131.
[670] Sapsis TP. Statistics of extreme events in fluid flows and waves. Annu Rev Fluid Mech 2021;53:85–111.
[671] Taylor SW, Woolford DG, Dean CB, Martell DL. Wildfire prediction to inform fire management: Statistical Science Challenges. Stat Sci 2013;28(4):586–615.
[672] Flannigan MD, Krawchuk MA, de Groot WJ, Wotton BM, Gowman LM. Implications of changing climate for global wildland fire. Int J Wildland Fire 2009;18:483–507.
[673] Sullivan AL. Wildland surface fire spread modelling, 1990–2007. 1: Physical and quasi-physical models. Int J Wildland Fire 2009;18:349–68.
[674] Sullivan AL. Wildland surface fire spread modelling, 1990–2007. 2: Empirical and quasi-empirical models. Int J Wildland Fire 2009;18:369–86.
[675] Sullivan AL. Wildland surface fire spread modelling, 1990–2007. 3: Simulation and mathematical analogue models. Int J Wildland Fire 2009;18:387–403.
[676] Coen J. Some requirements for simulating wildland fire behavior using insight from coupled weather–wildland fire models. Fire 2018;1(6):1–18.
[677] Bakhshaii A, Johnson EA. A review of a new generation of wildfire–atmosphere modeling. Can J For Res 2019;49:565–674.
[678] Xi DDZ, Taylor SW, Woolford DG, Dean CB. Statistical models of key components of wildfire risk. Annu Rev Stat Appl 2019;6:197–222.
[679] Brillinger DR, Preisler HK, Benoit JW. Risk assessment: A forest fire example. In: Goldstein DR, editor. Science and Statistics: A Festschrift for Terry Speed. vol. 40. Institute of Mathematical Statistics; 2003. p. 177–96.
[680] Martell DL, Otukol S, Stocks BJ. A logistic model for predicting daily people-caused forest fire occurrence in Ontario. Can J For Res 1987;17:394–401.
[681] Wotton BM, Martell DL. A lightning fire occurrence model for Ontario. Can J For Res 2005;35:1389–401.
[682] Preisler HK, Brillinger DR, Burgan RE, Benoit JW. Probability based models for estimation of wildfire risk. Int J Wildland Fire 2004;13:133–42.
[683] Vega-Garcia C, Lee BS, Woodard PM, Titus SJ. Applying neural network technology to human-caused wildfire occurrence prediction. AI Appl 1996;10(3): 9–18.
[684] Alonso-Betanzos A, Fontenla-Romero O, Guijarro-Berdiñas B, Hernández-Pereira E, Canda J, Jimenez E, Legido JL, Muñiz S, Paz-Andrade C, Paz-Andrade MI. A neural network approach for forestal fire risk estimation. Proc Europ Conf Artif Intell 2002;15:643–7.
[685] Vasilakos C, Kalabokidis K, Hatzopoulos J, Kallos G, Matsinos Y. Integrating new methods and tools in fire danger rating. Int J Wildland Fire 2007;16:306–16.
[686] Vasilakos C, Kalabokidis K, Hatzopoulos J, Matsinos I. Identifying wildland fire ignition factors through sensitivity analysis of a neural network. Nat Hazards 2009;50:125–43.
[687] Sakr GE, Elhajj IH, Mitri G. Efficient forest fire occurrence prediction for developing countries using two weather parameters. Eng Appl Artif Intell 2011;24 (5):888–94.
[688] Dutta R, Aryal J, Das A, Kirkpatrick JB. Deep cognitive imaging systems enable estimation of continental-scale fire incidence from climate data. Sci Rep 2013;3: 3188.
[689] Sakr GE, Elhajj IH, Mitri G, Wejinya UC. Artificial intelligence for forest fire prediction. IEEE/ASME International Conference on Advanced Intelligent Mechatronics. 2010. p. 1311–6.
[690] Stojanova D, Kobler A, Ogrinc P, Ženko B, Džeroski S. Estimating the risk of fire outbreaks in the natural environment. Data Min Knowl Disc 2012;24:411–42.
[691] Oliveira S, Oehler F, San-Miguel-Ayanz J, Camia A, Pereira JMC. Modeling spatial patterns of fire occurrence in Mediterranean Europe using Multiple Regression and Random Forest. For Ecol Manag 2012;275:117–29.
[692] Woolford DG, Dean CB, Martell DL, Cao J, Wotton BM. Lightning-caused forest fire risk in northwestern ontario, canada, is increasing and associated with anomalies in fire weather. Environmetrics 2014;25(6):406–16.
[693] Linn R, Reisner J, Colman JJ, Winterkamp J. Studying wildfire behavior using FIRETEC. Int J Wildland Fire 2002;11(4):233–46.
[694] Rothermel RC. A mathematical model for predicting fire spread in wildland fuels. Research Paper INT-115. Ogden, UT 84401: US Department of Agriculture, Intermountain Forest and Range Experiment Station; 1972.
[695] Cheney NP, Gould JS, Catchpole WR. Prediction of fire spread in grasslands. Int J Wildland Fire 1998;8(1):1–13.
[696] Finney MA. FARSITE: Fire area simulator–model development and evaluation. Research Paper RMRS-RP-4 Revised. Missoula, MT 59807: US Department of Agriculture, Forest Service, Rocky Mountain Research Station; 2004.
[697] Brun C, Margalef T, Cortés A, Sikora A. Enhancing multi-model forest fire spread prediction by exploiting multi-core parallelism. J Supercomput 2014;70:721–32.
[698] Ntinas VG, Moutafis BE, Trunfio GA, Sirakoulis GC. Parallel fuzzy cellular automata for data-driven simulation of wildfire spreading. J Comput Sci 2017;21: 469–85.
[699] Méndez-Garabetti M, Bianchini G, Caymes-Scutari P, Tardivo ML. Increase in the quality of the prediction of a computational wildfire behavior method through the improvement of the internal metaheuristic. Fire Saf J 2016;82:49–62.
[700] Xue H, Gu F, Hu X. Data assimilation using sequential Monte Carlo methods in wildfire spread simulation. ACM Trans Model Comput Simul 2012;22(4):23.
[701] Abdalhaq B, Cortés A, Margalef T, Luque E. Enhancing wildland fire prediction on cluster systems applying evolutionary optimization techniques. Future Gener Comp Sy 2005;21(1):61–7.
[702] Rodríguez R, Cortés A, Margalef T, Luque E. An adaptive system for forest fire behavior prediction. IEEE Int Conf Comput Sci Eng 2008;11:275–82.
[703] Rodríguez R, Cortés A, Margalef T. Injecting dynamic real-time data into a DDDAS for forest fire behavior prediction. In: Allen G, Nabrzyski J, Seidel E, van Albada GD, Dongarra J, Sloot PMA, editors. Computational Science – ICCS 2009. Springer; 2009. p. 489–99.
[704] Asch M, Bocquet M, Nodet M. Data Assimilation: Methods, Algorithms, and Applications. SIAM; 2016.
[705] Labahn JW, Wu H, Harris SR, Coriton B, Frank JH, Ihme M. Ensemble Kalman filter for assimilating experimental data into large-eddy simulations of turbulent flows. Flow Turbul Combust 2019;104:861–93.
[706] Cencerrado A, Cortés A, Margalef T. Response time assessment in forest fire spread simulation: An integrated methodology for efficient exploitation of available prediction time. Environ Model Softw 2014;54:153–64.
[707] Artés T, Cencerrado A, Cortés A, Margalef T. Core allocation policies on multicore platforms to accelerate forest fire spread predictions. In: Wyrzykowski R, Dongarra J, Karczewski K, Waśniewski J, editors. Parallel Processing and Applied Mathematics. Springer; 2014. p. 151–60.

[708] Artés T, Cencerrado A, Cortés A, Margalef T. Time aware genetic algorithm for forest fire propagation prediction: exploiting multi-core platforms. Concurrency Computat: Pract Exper 2017;29(9):e3837.
[709] Denham M, Wendt K, Bianchini G, Cortés A, Margalef T. Dynamic data-driven genetic algorithm for forest fire spread prediction. J Comput Sci 2012;3:398–404.
[710] Denham M, Laneri K. Using efficient parallelization in graphic processing units to parameterize stochastic fire propagation models. J Comput Sci 2018;25:76–88.
[711] Cencerrado A, Cortés A, Margalef T. Genetic algorithm characterization for the quality assessment of forest fire spread prediction. Procedia Comput Sci 2012;9:312–20.
[712] Carrillo C, Artés T, Cortés A, Margalef T. Error function impact in dynamic data-driven framework applied to forest fire spread prediction. Procedia Comput Sci 2016;80:418–27.
[713] Ascoli D, Vacchiano G, Motta R, Bovio G. Building Rothermel fire behaviour fuel models by genetic algorithm optimisation. Int J Wildland Fire 2015;24(3):317–28.
[714] Lautenberger C, Rein G, Fernandez-Pello C. The application of a genetic algorithm to estimate material properties for fire modeling from bench-scale fire test data. Fire Saf J 2006;41:204–14.
[715] Chetehouna K, Tabach EE, Bouazaoui L, Gascoin N. Predicting the flame characteristics and rate of spread in fires propagating in a bed of *pinus pinaster* using artificial neural networks. Process Saf Environ Prot 2015;98:50–6.
[716] Filippi J-B, Mallet V, Nader B. Representation and evaluation of wildfire propagation simulations. Int J Wildland Fire 2014;23:46–57.
[717] Arca B, Duce P, Laconi M, Pellizzaro G, Salis M, Spano D. Evaluation of FARSITE simulator in Mediterranean maquis. Int J Wildland Fire 2007;16(5):563–72.
[718] Trunfio GA, D'Ambrosio D, Rongo R, Spataro W, Di Gregorio S. A new algorithm for simulating wildfire spread through cellular automata. ACM Trans Model Comput Simul 2011;22(1):6.
[719] Subramanian SG, Crowley M. Learning forest wildfire dynamics from satellite images using reinforcement learning. Multidiscip Conf Reinf Learn and Decis Making 2017;3:1–4.
[720] Subramanian SG, Crowley M. Using spatial reinforcement learning to build forest wildfire dynamics models from satellite images. Front ICT 2018;5:6.
[721] Zheng Z, Huang W, Li S, Zeng Y. Forest fire spread simulating model using cellular automaton with extreme learning machine. Ecol Model 2017;348:33–43.
[722] Kozik VI, Nezhevenko ES, Feoktistov AS. Studying the method of adaptive prediction of forest fire evolution on the basis of recurrent neural networks. Optoelectron Instrum Data Process 2014;50:395–401.
[723] Khakzad N. Modeling wildfire spread in wildland-industrial interfaces using dynamic Bayesian network. Reliab Eng Syst Saf 2019;189:165–76.
[724] Radke D, Hessler A, Ellsworth D. Firecast: leveraging deep learning to predict wildfire spread. Proc Int Jt Conf Artif Intell 2019;28:4575–81.
[725] Huot F, Hu RL, Goyal N, Sankar T, Ihme M, Chen Y-F. Next day wildfire spread: A machine learning data set to predict wildfire spreading from remote-sensing data. arXiv Preprint 2021;2112.02447.
[726] Hodges JL, Lattimer BY. Wildland fire spread modeling using convolutional neural networks. Fire Technol 2019;55:2115–42.
[727] Burge J, Bonanni M, Ihme M, Hu L. Convolutional LSTM neural networks for modeling wildland fire dynamics. arXiv Preprint 2020;2012.06679.
[728] Vianna SSV, Cant RS. Explosion pressure prediction via polynomial mathematical correlation based on advanced CFD modelling. J Loss Prev Process Ind 2012;25:81–9.
[729] Shi J, Chang B, Khan F, Chang Y, Zhu Y, Chen G, Zhang C. Stochastic explosion risk analysis of hydrogen production facilities. Int J Hydrog Energy 2020;45:13535–50.
[730] Kobayashi T, Murayama S, Hachijo T, Gotoda H. Early detection of thermoacoustic combustion instability using a methodology combining complex networks and machine learning. Phys Rev Applied 2019;11:064034.
[731] Hachijo T, Masuda S, Kurosaka T, Gotoda H. Early detection of thermoacoustic combustion oscillations using a methodology combining statistical complexity and machine learning. Chaos 2019;29(10):103123.
[732] Marwan N, Carmen Romano M, Thiel M, Kurths J. Recurrence plots for the analysis of complex systems. Phys Rep 2007;438(5):237–329.
[733] Rosso OA, Larrondo HA, Martin MT, Plastino A, Fuentes MA. Distinguishing noise from chaos. Phys Rev Lett 2007;99:154102.
[734] Scheffer M, Bascompte J, Brock WA, Brovkin V, Carpenter SR, Dakos V, Held H, Van Nes EH, Rietkerk M, Sugihara G. Early-warning signals for critical transitions. Nature 2009;461(7260):53–9.
[735] Gopalakrishnan EA, Sharma Y, John T, Dutta PS, Sujith RI. Early warning signals for critical transitions in a thermoacoustic system. Sci Rep 2016;6:35310.
[736] Grogan KP, Ihme M. Identification of governing physical processes of irregular combustion through machine learning. Shock Waves 2018;28:941–54.
[737] Farmer JD, Sidorowich JJ. Predicting chaotic time series. Phys Rev Lett 1987;59:845–8.
[738] Abarbanel HDI, Brown R, Sidorowich JJ, Tsimring LS. The analysis of observed chaotic data in physical systems. Rev Mod Phys 1993;65:1331–92.
[739] Sapsis TP, Majda AJ. Statistically accurate low-order models for uncertainty quantification in turbulent dynamical systems. Proc Natl Acad Sci USA 2013;110(34):13705–10.
[740] Majda AJ, Lee Y. Conceptual dynamical models for turbulence. Proc Natl Acad Sci USA 2014;110(18):6548–53.
[741] Brunton SL, Brunton BW, Proctor JL, Kaiser E, Kutz JN. Chaos as an intermittently forced linear system. Nat Commun 2017;8(19).
[742] Glaessgen E, Stargel D. The digital twin paradigm for future NASA and U.S. air force vehicles. AIAA Pap 2012-1818 2012.
[743] Baldi P, Sadowski P, Whiteson D. Searching for exotic particles in high-energy physics with deep learning. Nat Commun 2014;5(4308).
[744] Baldi P, Sadowski P, Whiteson D. Enhanced higgs boson to $\tau^+\tau^-$ search with deep learning. Phys Rev Lett 2015;114(11):111801.
[745] Kasieczka G, Nachman B, Shih D, Amram O, Andreassen A, Benkendorfer K, et al. The LHC olympics: A community challenge for anomaly detection in high energy physics. arXiv Preprint 2021;2101.08320.
[746] Adam-Bourdarios C, Cowan G, Germain C, Guyon I, Kégl B, Rousseau D. The Higgs boson machine learning challenge. Proc NIPS 2014 Workshop High-energy Phys and Mach Learn 2015;42:19–55.
[747] Calafiura P, Farrell S, Gray H, Vlimant J-R, Innocente V, Salzburger A, et al. TrackML: A high energy physics particle tracking challenge. IEEE Int Conf e-Sci 2018;14:344.
[748] Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. Adv Neural Inform Process Syst 2012;25:1097–105.
[749] Pickett L.K.. Engine combustion network. 2011. https://ecn.sandia.gov.
[750] Farrell J.T.. Co-optimization of fuels & engines: Fuel properties database. 2021. https://www.nrel.gov/transportation/fuels-properties-database/.
[751] Doshi-Velez F, Kim B. Towards a rigorous science of interpretable machine learning. arXiv Preprint 2017;1702.08608.
[752] Murdoch WJ, Singh C, Kumbier K, Abbasi-Asl R, Yu B. Definitions, methods, and applications in interpretable machine learning. Proc Natl Acad Sci USA 2019;116(44):22071–80.
[753] Carvalho DV, Pereira EM, Cardoso JS. Machine learning interpretability: A survey on methods and metrics. Electron 2019;8(832):1–34.
[754] Samek W, Müller K-R. Towards explainable artificial intelligence. In: Samek W, Montavon G, Vedaldi A, Hansen LK, Müller K-R, editors. Explainable AI: Interpreting, Explaining and Visualizing Deep Learning. Lecture Notes in Computer Science, vol. 11700. Springer; 2019. p. 5–22.
[755] Montavon G, Samek W, Müller K-R. Methods for interpreting and understanding deep neural networks. Digit Signal Process 2018;73:1–15.
[756] Lundberg SM, Lee S-I. A unified approach to interpreting model predictions. Adv Neural Inform Process Syst 2017;30:4768–77.
[757] Shapley LS. A value for n-person games. Contrib Theory Games 1953;2(28):307–17.
[758] Ribeiro MT, Singh S, Guestrin C. "Why should I trust you?": Explaining the predictions of any classifier. Proc ACM SIGKDD Int Conf Knowl Discov Data Min 2016;22:1135–44.
[759] Du M, Liu N, Hu X. Techniques for interpretable machine learning. Commun ACM 2020;63(1):68–77.
[760] Selvaraju RR, Cogswell M, Das A, Vedantam R, Parikh D, Batra D. Grad-CAM: visual explanations from deep networks via gradient-based localization. IEEE Int Conf Comput Vis 2017:618–26.
[761] Rasmussen CE. Gaussian Processes in Machine Learning. In: Bousquet O, von Luxburg U, Rätsch G, editors. Advanced Lectures on Machine Learning: ML Summer Schools 2003, Canberra, Australia, February 2–14, 2003, Tübingen, Germany, August 4–16, 2003, Revised Lectures. Berlin, Heidelberg: Springer; 2004. p. 63–71.
[762] Liu H, Ong YS, Shen X, Cai J. When Gaussian process meets big data: A review of scalable GPs. IEEE Trans Neural Networks Learn Syst 2020;31(11):4405–23.
[763] Chalupka K, Williams CKI, Murray I. A framework for evaluating approximation methods for Gaussian process regression. J Mach Learn Res 2013;14:333–50.
[764] Gneiting T. Compactly supported correlation functions. J Multivar Anal 2002;83:493–508.
[765] Wilson H, Nickisch AG. Kernel interpolation for scalable structured Gaussian processes (KISS-GP). Int. Conf. Mach. Learn.. 2015. p. 1775–84.
[766] Hinton GE, van Camp D. Keeping the neural networks simple by minimizing the description length of the weights. Proc Annu Conf Comput Learn Theory 1993;6:5–13.
[767] Graves A. Practical variational inference for neural networks. Adv Neural Inform Process Syst 2011;24:2348–56.
[768] Gal Y, Ghahramani Z. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. Proc Int Conf Mach Learn 2016;48:1050–9.
[769] Blundell C, Cornebise J, Kavukcuoglu K, Wierstra D. Weight uncertainty in neural networks. Proc Int Conf Mach Learn 2015;37:1613–22.
[770] Lakshminarayanan B, Pritzel A, Blundell C. Simple and scalable predictive uncertainty estimation using deep ensembles. Adv Neural Inform Process Syst 2017;30:6402–13.
[771] Foong AYK, Burt DR, Li Y, Turner RE. On the expressiveness of approximate inference in Bayesian neural networks. Adv Neural Inform Process Syst 2020;33:15897–908.
[772] Koenker R, Hallock KF. Quantile regression. J Econ Perspect 2001;15(4):143–56.
[773] Meinshausen N, Ridgeway G. Quantile regression forests. J Mach Learn Res 2006;7:983–99.
[774] Taylor JW. A quantile regression neural network approach to estimating the conditional density of multiperiod returns. J Forecast 2000;19:299–311.
[775] Nguyen A, Yosinski J, Clune J. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. Proc IEEE Conf Comput Vision Pattern Recognit 2015:427–36.
[776] Amodei D, Olah C, Steinhardt J, Christiano P, Schulman J, Mané D. Concrete problems in AI safety. arXiv Preprint 2016;1606.06565.
[777] Kearns MJ. The Computational Complexity of Machine Learning. MIT Press; 1990.
[778] Valiant LG. A theory of the learnable. Commun ACM 1984;27(11):1134–42.