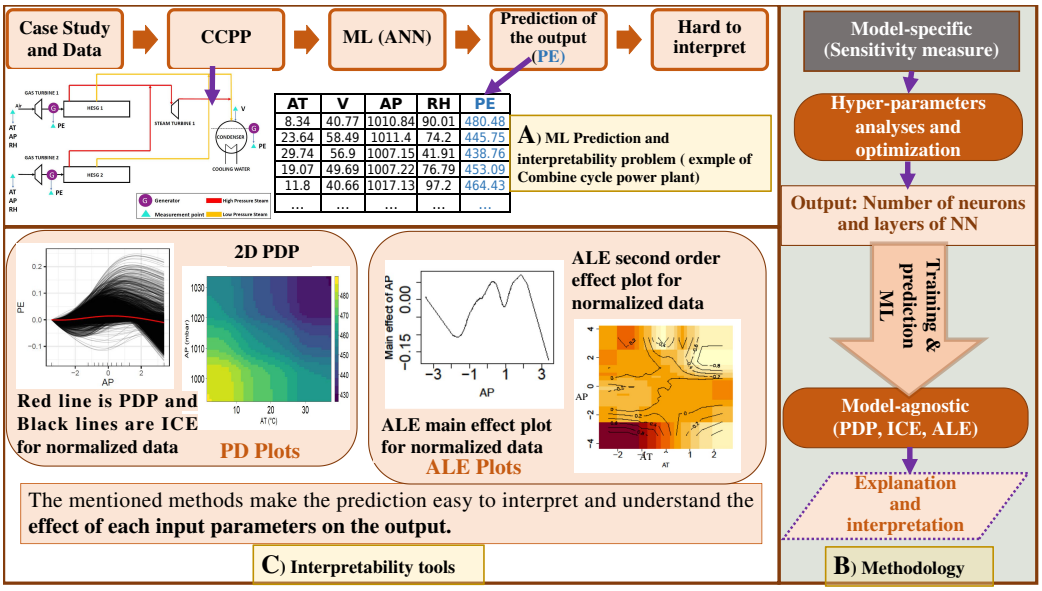
Hybridization of model-specific and model-agnostic methods for interpretability of Neural network predictions: Application to a power plant

Tina Danesh, Rachid Ouaret, Pascal Floquet, Stéphane Négny

Graphical Abstract

Hybridization of model-specific and model-agnostic methods for interpretability of Neural network predictions: Application to a power plant

Tina Danesh, Rachid Ouaret, Pascal Floquet, Stephane Negny



Highlights

Hybridization of model-specific and model-agnostic methods for interpretability of Neural network predictions: Application to a power plant

Tina Danesh, Rachid Ouaret, Pascal Floquet, Stephane Negny

- Hybridizing model-specific and model-agnostic to enhance the interpretability of ANN predictions.
- Reconciling the prediction accuracy and the interpretability for a global approach to making systems more flexible
- Explaining the functionality of the model-specific (partial derivatives) approach and model-agnostic (PDP, ICE, ALE) for interpretability purposes.
- Understanding how variations (quantitatively and qualitatively) in inputs affect the predictions of an ANN for an engineering application.

Hybridization of model-specific and model-agnostic methods for interpretability of Neural network predictions: Application to a power plant

Tina Danesh[a], Rachid Ouaret[a], Pascal Floquet[a], Stephane Negny[a]

[a]*Laboratoire de Genie Chimique, Université de Toulouse, CNRS, INPT, UPS, LGC UMR 5503, 4 allée Emile Monso Toulouse, 31030, France*

Abstract

Advanced computing performance and machine learning (ML) accuracy have pushed engineers and researchers to consider more and more complex mathematical models. Methods such as Deep Neural Networks have become increasingly ubiquitous. However, the problem of the interpretability of machine learning predictions in a decision process has been identified as a hot topic in several engineering fields, leading to confusion in various communities. This paper discusses a methodological framework of hybrid interpretability tools in neural network prediction for an engineering application. These tools analyze a decision's consequences under different circumstances and situations. The aim is to reconcile the ML prediction accuracy and the interpretability for a global approach to making systems more flexible. In this study, the methods used to deal with the interpretability of neural network predictions have been treated from two perspectives: *(i)* model-specific as partial derivatives and *(ii)* model-agnostic methods. The latter tools could be used for any ML model prediction. In order to visualize and explain the inputs' impacts on prediction results, Partial Dependence Plots (PDP), Individual Conditional Expectation (ICE), and Accumulated Local Effects (ALE) are used and compared. The prediction of the electrical power (PE) output of a combined cycle power plant has been chosen to demonstrate the feasibility of these methods under real operating conditions. The results

Email addresses: tina.danesh@toulouse-inp.fr (Tina Danesh), rachid.ouaret@toulouse-inp.fr (Rachid Ouaret), pascal.floquet@toulouse-inp.fr (Pascal Floquet), stephane.negny@toulouse-inp.fr (Stephane Negny)

show that the most influential input parameter among ambient temperature (AT), atmospheric pressure (AP)), vacuum (V), and relative humidity (RH) is AT. The visualization outputs allow us to identify the direction (positive or negative) and the form (linear, nonlinear, random, stepwise) of the relationship between the input variables and the model's output. The results of the interpretation are coherent with the literature studies.

Keywords: Machine learning, Interpretability, Sensitivity analysis, model-specific, Model-agnostic, Partial dependence plots, Individual Conditional Expectation, Accumulated local effects
PACS: 0000, 1111
2000 MSC: 0000, 1111

---

1. Introduction

The discipline of process design and control has made tremendous advances in the last three decades with the advent of computer computation capabilities of complex processes. These advances have made it possible to analyze and predict the behaviors of several complex systems (Ramirez, 1997; Bequette, 2003). There are mainly two different visions: equation-based knowledge models, such as physical-chemical models known as *"white box"* models, and data-oriented approaches, which are primarily based on Machine Learning (ML) algorithms known as *"black-box"* models. The predictions of equation-based methods are easy to understand and interpret since the model's assumptions and the relationships between different variables have physical meanings. In addition, several studies have been performed on them, so they are understandable and well-established by the community.

From the engineering perspective, the detailed description of the whole system requires complex and often highly parameterized models. In addition, numerous assumptions with many nonlinear equations should be made to analyze and predict accurately, particularly for complex dynamical systems. Hence, it is time-consuming and takes effort to analyze a real application. On the other side, engineering problems have become complicated and consist of more data to analyze, especially for a large system with nonlinear behavior. Furthermore, features such as uncertainty, multi-scale, time lag, and large variable space dimensions affect the model prediction and hence any model-based inference for guiding policies in the real world. Data-oriented and ML techniques would be helpful to deal with these barriers (Kesgin and

Heperkan, 2005). In parallel, machine learning has been heavily researched and widely used in many areas, such as process engineering application (Lee et al., 2018; Agarwal et al., 2021), and optimization (Ning and You, 2019; Xia et al., 2022; Qazani et al., 2022). Extensive literature attests to the superiority of black-box ML algorithms in minimizing predictive errors, both from a theoretical (Cybenko, 1989; Hornik, 1991; Park and Sandberg, 1991; Leshno et al., 1993) and an applied perspective (Sahoo et al., 2017; Li et al., 2019). The success of ML in many applications is grounded in its powerful capability for prediction purposes with high accuracy. However, some of them are still hard to interpret regarding the relationship between predictors and model outcomes (Moradi and Samwald, 2021). At the same time, they suffer from a lack of interpretability and explainability because they function without process knowledge dependency.

ML techniques are applied mainly as alternatives to physical approaches, considering the increasing volume of data in real-world situations, while the model development process is laborious and time-consuming (Chen and Zhang, 2014; Venkatasubramanian et al., 2003c,a,b; Bhakte et al., 2022). These data-oriented models can be applied to extract useful information and support decision-makers. One of the most popular ML algorithms for continuous output predictions is the Artificial Neural Network (ANN), thanks to the universal approximation theorem (Hornik, 1991). Among ML techniques, Deep Neural Networks (DNNs) have become popular for predictions and control purposes (Amari, 1967; Schmidhuber, 2015).

The artificial neural network has some advantages that include providing predictive benefits compared to other models, such as detecting complicated nonlinear relationships between dependent and independent variables. Its disadvantages include the complexity of neural networks, which makes it hard to understand why it predicts successfully and when we can trust it.

A recent advanced research topic in ML methods, especially in neural networks, is to find a way to obtain information and gain the ability to interpret and explain how the input variables affect the output variable to help decision-makers that is called interpretability. Therefore, interpretability in the ML community and sensitivity analysis in the engineering community are concerned with practically the same issues and target the same objectives when one examines the literature of two disciplines, although they could be considered as distinct concepts. Thus, we find practically the same vocabulary evolving in two different environments:

- The concept of sensitivity has been widely developed in the field of mathematical applications in engineering (Sobol, 1998; Saltelli, 2002; Saltelli et al., 2008, 2010). It refers to the study of how changes in input variables affect the output or predictions of a model. It involves systematically varying the values of input variables within a specified range and observing the corresponding changes in the output. In addition, it helps rank the most influential input variables on the model's output. By conducting sensitivity analysis, one can gain insights into how the model responds to changes in input variables and identify potential sources of uncertainty or risk in the model's predictions.

- Interest in ML was mainly focused on predictive performance, and it is only recently that the problem of interpretability and explaining ML prediction has been posed as a fundamental issue in the evaluation of black-box models (Simonyan et al., 2013; Shrikumar et al., 2016; Ribeiro et al., 2016a). The interpretability of prediction problems is defined as the process of extracting relevant knowledge from a model about the learned relationships between features and model outputs. It is important in many real-world applications, where decision-makers need to understand the reasons behind a model's predictions to gain trust, make informed decisions, and ensure compliance with regulations. The tools that make the model interpretable have the same goals as the sensitivity analysis; such as analyzing and prioritizing the model's parameters, recognizing the less effective parameters to decrease the dimension and simplify the problem, and minimizing the variation of the most influential parameters to reduce the dispersion of the model output.

In brief, there are advantages to both interpretability tools and sensitivity analysis. Both methods are useful for comprehending and evaluating machine learning models, and the choice of one over the other depends on the analysis's particular objectives, circumstances, and needs. These aspects have been addressed by hybridization model-specific and model-agnostic for neural network predictions.

This study is carried out on a Combined Cycle Power Plant (CCPP) as a real-world application. A CCPP generates electrical power while having relatively little gas emissions. In order to respond to PE demands and enhance the efficiency of the power plant, the decision-maker wants to know

the most influential input variables and the impact of these variables. So a dataset that contains 9568 data points was collected from a CCPP over six years (2006-2011)(Tüfekci, 2014). Tüfekci tested and compared some machine learning regression methods to extend a predictive model for an electrical power output of the CCPP. The paper evaluated the prediction accuracy by Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) for continuous variables. A different ML method for prediction is chosen as Tüfekci, as hyper-parameter analysis is performed for Multi-Layer Perceptron (MLP), resulting in lower RMSEs compared to the best ML method found in (Tüfekci, 2014). The Tüfekci's paper does not focus on the interpretability aspects of the applied ML methods.

The contributions of our paper include:

- Hybridizing model-specific and model-agnostic to enhance the interpretability of ANN predictions. More precisely, it is a question of integrating sensitivity analyses based on the partial derivatives of neural networks and the application of current interpretability methods.
- Explaining the functionality of the model-specific (partial derivatives) approach and model-agnostic (PDP, ICE, ALE) for interpretability purposes.
- Understanding how variations (quantitatively and qualitatively) in inputs affect the predictions of supervised ML predictions for a power plant application. This part is fundamental because it would allow the engineers to act on the most important variables and identify the instability regimes in the studied system. Various ongoing changes in the power system are impacting the need for power production.

The rest of this paper is organized as follows. Section 2 is about the history of the interpretability concept. In Section 3, neural network sensitivity and interpretation tools are presented, whereas the experimental and simulation work is given in Section 4. Section 5 is dedicated to analyzing and discussing the results. Finally, we conclude in Section 6.

2. Related work

Despite the impressive accuracy of ML models, it is not the only factor that matters. A thorough and complete understanding of the model and

the relationships between parameters is crucial for real-world applications. The necessity of interpretability is apparent in sensitive fields like process engineering where a proper understanding of ML is essential and obvious.

The quality of the product or the efficiency of the process could be enhanced by process monitoring, such as fault detection and diagnosis. During the past decades, traditional multivariate statistical methods such as principal component analysis (PCA) widely employed in fault detection and diagnosis (Venkatasubramanian et al., 2003b; Harmouche et al., 2014, 2015; Gajjar and Palazoglu, 2016). When employing these linear approaches, the process' inherent non-linearity presents difficulties, and non-linear procedures can offer more accuracy. To this purpose, advanced ML techniques, such as neural networks, have significantly outperformed older ones and can depict and recognize non-linearity present in the data.

In the recent past, artificial neural networks (ANNs) have drawn a lot of interest in process engineering. The multi-layer perceptron (MLP), which is straightforward and flexible in managing numerous inputs and multiple outputs, is one of the default options for process engineering applications. The ML techniques' effectiveness stems from the complicated mathematical changes they use, but this sacrifices interpretability and explainability. Due to its inherent complexity, it is challenging to understand the reasoning behind a certain trained model's decision or prediction. For example, there are different studies that choose the Tennessee Eastman process as a case study and aim to enhance the interpretability and explainability of their model (Agarwal et al., 2021; Bhakte et al., 2022). Lack of interpretability and explainability problem doesn't only affect chemical engineering applications; it also occurs whenever an ML algorithm's output must be applied by a human. This problem gave rise to the concepts of model interpretability and explainability.

As models might be simple to grasp in specific contexts but not in others, interpretability is tricky to define. While numerous papers have discussed interpretable models in different fields, the ML community lacks a common framework to address what features make a model interpretable and what we aim to gain with interpretable results. Such a framework is proposed by Lipton et al. (2016) within which we may discuss and evaluate models' interpretability. The authors describe both what we want to obtain from interpretable machine learning systems and how interpretability may be attained. Similar to this, Doshi-Velez and Kim (2017) present a methodological framework for considering how interpretability techniques might be assessed.

ML methods can be used in decision-making, although decision-makers want to understand the reasons and explanations behind the predictions since they do not blindly trust the ML models. Therefore, one of the driving factors of explainability is trust. Other driving factors include causality, transferability, informativeness (Lipton, 2018), fair and ethical decision-making (Goodman and Flaxman, 2017), accountability (Freitas, 2014), making adjustments (Selvaraju et al., 2017) and proxy functionality (Doshi-Velez and Kim, 2017).

Many alternative methods have been offered to explain ML predictions (Molnar, 2019). Some attempt to explain the whole model or replace it with an intrinsically intelligible model, such as a decision tree (Freitas, 2014). There are other techniques that attempt to direct the model throughout the learning process to a more interpretable state (Burkart et al., 2019; Schaaf et al., 2019). Other techniques focus on only explaining particular predictions, such as by highlighting important features (Ribeiro et al., 2016a) or comparing different decisions (Wachter et al., 2017).

The interpretability could be handled by using model-specific (sensitivity measures) as a quantitative method and model-agnostic such as Partial Dependence Plots (PDP) (Friedman, 2001), and Accumulated Local Effects (ALE) (Apley and Zhu, 2020) as qualitative methods. Figure 1 shows the machine learning interpretability procedure overview. Firstly, the goal is to predict using a supervised ML model such as ANN. In order to give valuable information to the decision-maker, the interpretability tools attempt to address the question of how the inputs impact the model's predictive performance.

One of the sensitivity analysis methods that could help gain helpful information from the neural networks is the partial derivatives method (White and Racine, 2001). Analytically calculating the derivatives gives more robust diagnostic information since it depends on neural network prediction efficiency. The derivatives will be the same and will not rely on the training conditions and the network structure until the neural network predicts the output variable with high accuracy (Beck, 2018).

As mentioned before, it is not simple to interpret some of the machine learning models. The general interpretative framework depends on the models. For example, it is possible and straightforward in linear regression to understand the *how* and the *why* given the statistical significance of the weights, so the interpretation of the linear regression model can be assessed by its coefficients. The linear regression coefficients (e.g. β_1, β_2, ..., β_p)

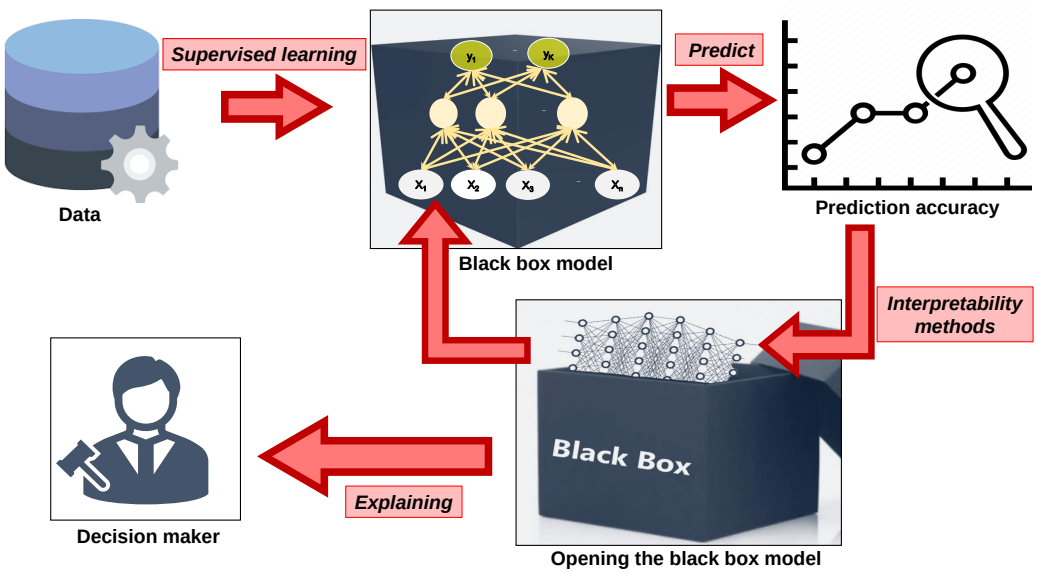


Figure 1: Overview of Machine Learning interpretability.

associated with continuous predictors x_1, x_2, ..., x_p is the difference in the predicted value of the response variable for each one-unit change in the predictor variable, assuming all other predictor variables are held constant. It is difficult to extrapolate this process to non-linear models. That is why they are called *model-specific* interpretations. These approaches have been designed specifically for a given model. Recently, some tools have emerged in ML that are supposed to remove this barrier to expressing the interpretation of machine learning models, whatever the learning model used. These tools are called *model-agnostic* tools.

The goal of model-agnostic is to create methods that can be used with any machine learning model, regardless of its underlying architecture or algorithms. Hence, rather than being limited to a particular model or framework, the approaches created in this subject are intended to be generically usable across a variety of models (Du et al., 2019).

When it is preferable to have the ability to employ a range of models to solve a particular problem and when the choice of model is not predefined or fixed, a model-agnostic approach is very beneficial. Researchers and practitioners can generate more flexible and adaptive solutions to situations that may call for many models or various methods throughout time by establish-

ing procedures that are not dependent on any specific model. This method also makes it easier to compare and assess several models based on how well they function when applied to a particular situation, without the requirement for method-specific expertise (Ribeiro et al., 2016b; Molnar, 2019).

Model-agnostic methods could effectively interpret supervised ML models by separating the explanations from the machine learning model (Ribeiro et al., 2016b). Model-agnostic methods are distinguished into local and global methods. The Partial Dependance Plots (PDPs), Individual Conditional Expectation (ICE) plots, and Accumulated Local Effects (ALE) Plots are some model-agnostic techniques (Friedman, 2001; Apley and Zhu, 2020).

3. Neural Network Sensitivity and interpretability

Figure 2 summarizes the methodological scheme of the study. This methodology consists of three main parts: Prediction problem (A), Methodology (B), and Result (C). Part A corresponds to the description of the prediction problem. In this part, the attempt is made to answer the following question: how do predictor variables impact the predictions of neural network regression? Specifically, there are 9568 data points of four predictor variables and one output variable. The ANN is performed on the data to predict electrical power output, though it lacks explainability and interpretability. Part B corresponds to the methods included in this paper to solve the problem in Part A. The results and comparisons from each method will be presented in Part C. This section will explain Part B in detail. In the first step, the hyper-parameter analysis is performed through partial derivatives and sensitivity measures to choose the appropriate number of layers and neurons in each layer. Then, by applying the visual aspects of model-agnostic tools, the input's impact on the variability of outputs will be investigated.

3.1. ANN sensitivity through partial derivatives

Sensitivity analysis could be performed on the neural networks using the partial derivatives method. This method comprises calculating the derivative of the output according to the inputs of the neural network (Pizarroso et al., 2020). These partial derivatives are considered sensitivity and can be calculated using the following equation:

$$s_{in}\,|_{\boldsymbol{x}_m} = \frac{\partial z_n}{\partial x_i}\,(\boldsymbol{x}_m) \tag{1}$$

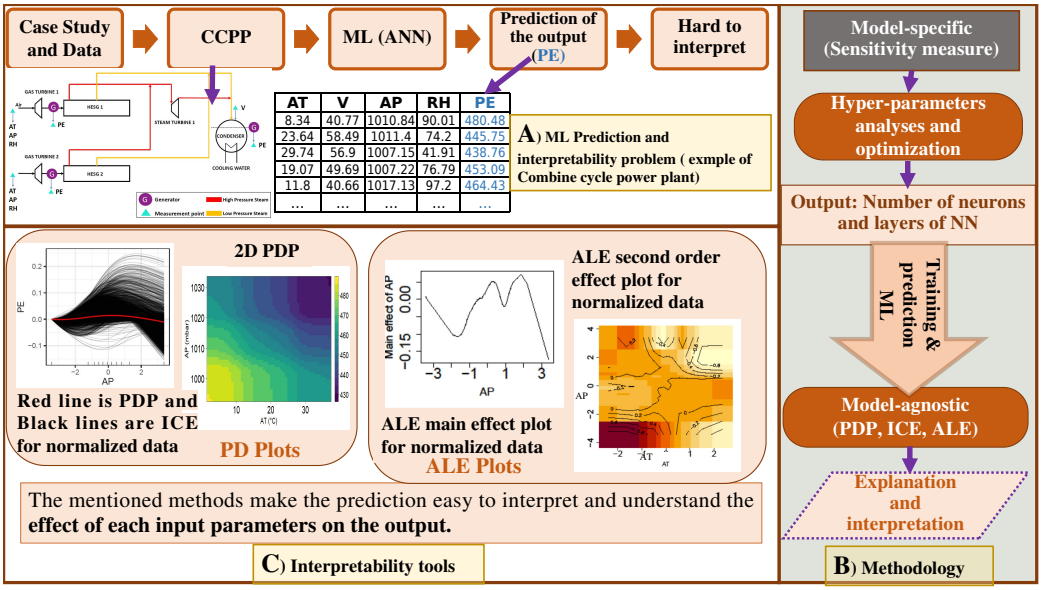


Figure 2: Overview of the methodology used in the study

Where $s_{in}|_{\boldsymbol{x}_m}$ refers to the sensitivity of the n^{th} neuron's output in the output layer according to the i^{th} neuron's input in the input layer that is calculated in $\boldsymbol{x}_m$, and $\boldsymbol{x}_m$ is the m sample of the dataset that the sensitivity analysis is performed on. In order to compute the sensitivity of the inner layers, the chain rule is applied to the partial derivatives. The related equations of the partial derivatives of the inner layers are defined by: *(i)* the derivative of y_n^l regarding z_i^{l-1} is $\frac{\partial y_n^l}{\partial z_i^{l-1}} = w_{ni}^l$ that represents the weight of the connection between the n^{th} neuron in the l^{th} layer and the i^{th} neuron in the $(l-1)^{th}$ layer, and *(ii)* the derivative of z_n^l regarding y_i^l is $\frac{\partial z_n^l}{\partial y_i^l}\Big|_{z_i^l} = \frac{\partial AF_n^l}{\partial y_i^l}\left(y_i^l\right)$ that $\frac{\partial AF_n^l}{\partial y_i^l}$ refers to the partial derivative of the activation function of the n^{th} neuron in the l^{th} layer regarding the n^{th} neuron's input in the l^{th} layer estimated for the input y_i^l of the i^{th} neuron in the l^{th} layer.

3.2. Sensitivity measures

After calculating the sensitivity for each variable and sample, different measures could be applied to analyze and interpret the results. In the general case, the following sensitivity measures are used: *(i)*Mean sensitivity of the n^{th} neuron's output in the output layer regarding the i^{th} input variable, *(ii)*Standard deviation (σ) sensitivity of the n^{th} neuron's output in the output

layer regarding the i^{th} input variable. *(iii)*Mean squared sensitivity of the n^{th} neuron's output in the output layer regarding the i^{th} input variable (Yeh and Cheng, 2010). The related measures are presented under two cases: Single or Multi-target regression.

- Mean sensitivity

  Single target regression:

$$S_{in}^{avg} = \frac{\sum_{j=1}^{m} s_{in}|_{\boldsymbol{x}_j}}{m} \tag{2}$$

  Multi-target regression:

$$S_{i}^{avg} = \frac{\sum_{n=1}^{m^l} S_{in}^{avg}}{m^L} \tag{3}$$

- Standard deviation sensitivity

  Single target regression:

$$S_{in}^{sd} = \sigma\left(s_{in}|_{\boldsymbol{x}_j}\right); j\epsilon 1, ..., m \tag{4}$$

  Multi-target regression:

$$S_{i}^{sd} = \sqrt{\frac{\sum_{n=1}^{m^l}\left(\left(S_{in}^{sd}\right)^2 + \left(S_{in}^{avg} - S_{i}^{avg}\right)^2\right)}{m^L}} \tag{5}$$

- Mean squared sensitivity

  Single target regression:

$$S_{in}^{sq} = \sqrt{\frac{\sum_{j=1}^{M}\left(s_{in}|_{\boldsymbol{x}_j}\right)^2}{m}} \tag{6}$$

  Multi-target regression:

$$S_{i}^{sq} = \frac{\sum_{n=1}^{m^l} S_{in}^{sq}}{m^L} \tag{7}$$

3.3. Partial Dependence Plots and Individual Conditional Expectation

Partial Dependence Plot (PDP) (Friedman, 2001) is an ideal graphical tool to analyze the impact of some input variables on the dependent variable when using nonlinear models such as an ANN, a random forest, or some gradient boosting. This is why they are considered as a model-agnostic tool. The PDP highlights the change in the average predicted value as the specified feature(s) vary over their marginal distribution. For individual data instances, the plots are considered as Individual Conditional Expectation (ICE) (Goldstein et al., 2015). The drawback of ICE plots is that they start with various projections, so it can often be challenging to determine whether the ICE curves differ across individuals. This issue can be overcome by centering the curves at a particular feature point, and the difference in the prediction at this point is all that will be shown. The centered ICE (c-ICE) plot is the name of the resultant plot. In this study, the c-ICE plot is used, which makes heterogeneity more obvious and emphasizes findings that differ from the general pattern. For example, in terms of MLP learning, all that is obtained is the importance of the weight. It is relatively simple to know which node connections significantly influence the outcome; it is not good that the direction of effect is unknown. The PDP and ICE are intuitive and easy-to-understand visualizations of the effect of the inputs on the predicted outcome.

Assume that $g(x)$ is a black-box supervised learning model; here is a neural network in our study. The fitted model is named $\hat{g}(x)$. The upper case X is used to identify random variables and the lower case to identify specific values of the random variables.

The x_f is the feature for which we want to know its effect on the prediction for plotting the partial dependence plots, and $X_{\setminus f}$ are the other features that exist in our model except the x_f, which are considered as random variables. The combination of feature vectors x_f and $x_{\setminus f}$ is the total feature space X.

The partial dependence function is defined as:

$$\hat{g}_{f,PDP}\left(x_f\right) = E_{X_{\setminus f}}\left[\hat{g}\left(x_f, X_{\setminus f}\right)\right] = \int_{X_{\setminus f}} \hat{g}\left(x_f, X_{\setminus f}\right) dP(X_{\setminus f}) \tag{8}$$

Where each subset of predictors f has its own partial dependence function g_f, which gives the average value of g when x_f is fixed and $X_{\setminus f}$ varies over its marginal distribution $dP(X_{\setminus f})$. The $\hat{g}_f$ is the expectation of g over the marginal distribution of all variables other than X_f.

In practice, the estimation of Equation 8 is calculated by averaging over the training data which is known as the Monte Carlo method:

$$\hat{g}_f(x_f) = \frac{1}{m}\sum_{a=1}^{m} g(x_f, x_{\setminus f}^{(a)}) \tag{9}$$

Where $x_{\setminus f}^{(1)}, ..., x_{\setminus f}^{(m)}$ represent the actual feature values that are observed in the training data, and m is the number of instances in the dataset. In PDP, it is assumed that the features in set $\setminus f$ are not correlated with the features in set f; if not, the average calculated for PDP may contain data points that are very unlikely or even impossible. Friedman's partial dependence plot aims to visualize the marginal effect of a given predictor towards the model outcome by plotting out the average model outcome in terms of different values of the predictor (Friedman, 2001).

While PDP provides the average effect of a feature of the predictions over the marginal distribution, ICE plots are a method to disaggregate these averages. ICE plots visualize the functional relationship between the predicted response and the feature separately for each instance. In other words, a PDP averages the individual lines of an ICE plot. In some of our experiments, we used normalized variables.

3.4. Accumulated Local Effects plots

Accumulated Local Effects (ALE) explain the average impact of features on the prediction of an ML model (Apley and Zhu, 2020). They are a faster option than partial dependence plots. ALE methods could work while the features are dependent, although the biggest problem of PDPs is the assumption of feature independence.

As mentioned before, for each $f \in \{1, ..., F\}$, let $X_{\setminus f}$ illustrate the subset of $(F-1)$ predictors excepting X_f. The ALE main effect of predictor x_f is defined as:

$$\hat{g}_{f,ALE}(x_f) = \int_{LB_{0,f}}^{x_f} E\left[\hat{g}^f\left(X_f, X_{\setminus f}\right) | X_f = LB_f\right] dLB_f - C \tag{10}$$

Where, $\hat{g}^f\left(X_f, X_{\setminus f}\right) = \frac{\partial \hat{g}(X_1,...,X_F)}{\partial X_f}$ (local effect of X_f on $\hat{g}^f$) and $LB_{0,f}$ refers to the approximation lower bound of X_f, and it affects the vertical translation of the ALE plot. C is considered as a constant that aims to make the mean of $\hat{g}_{f,ALE}(x_f)$ equal to zero concerning the marginal distribution of X_f or to center the plot vertically.

There are some differences in the ALE formulation compared to the PDP formulation, such as:

- ALE averages the predictions conditional on each grid value of the interested feature, and PDP presumes the marginal distribution at each grid value.
- The changes of predictions, not the predictions themselves, are averaged, and the change is defined as the partial derivative.
- The equation has the additional integral over $LB_{0,f}$ that refers to an approximate lower bound of X_f.
- The changes of predictions, not the predictions themselves, are averaged, and the change is defined as the partial derivative.

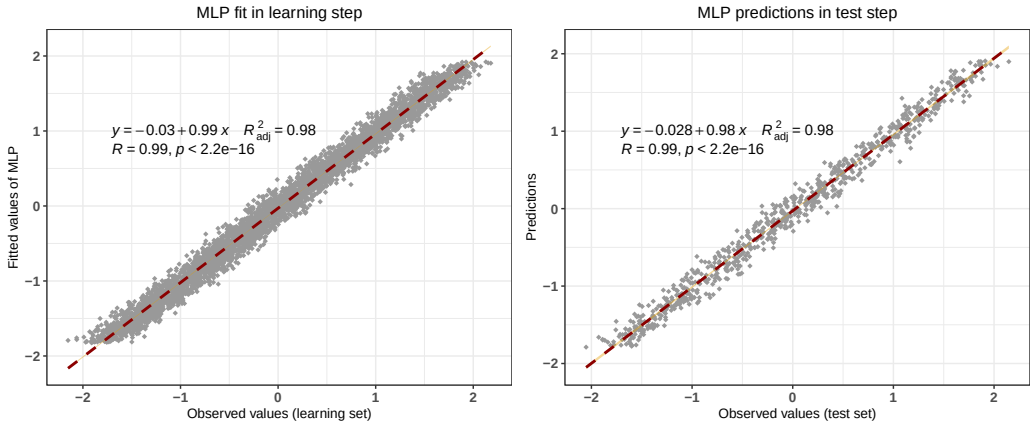
In order to calculate the estimation of Equation 10, first, features are categorized into many intervals, and then the differences in the predictions are calculated. This procedure could approximate the derivatives. This procedure's advantage is that it can work for models with no derivatives. The estimated equation that are proposed by Apley and Zhu (2020) is as follows: Estimation of ALE main effect:

$$\hat{\tilde{g}}_{f,ALE}\left(x_f\right) = \sum_{u=1}^{u_f(x)} \frac{1}{m_f(u)} \sum_{t:x_{t,f}\in N_f(u)} \left[\hat{g}(LB_{u,f}, x_{t,\setminus f}) - \hat{g}(LB_{u-1,f}, x_{t,\setminus f})\right] - C \tag{11}$$

Where for each $u \in \{1, 2, ..., U\}$, $m_f(u)$ refers to the number of training observation that falls into uth interval $M_f(u)$. For each $f \in \{1, 2, ..., F\}$, $\{M_f(u) = (LB_{u-1,f}, LB_{u,f}]; u = 1, 2, ..., U\}$ refers to an enough good partition of the sample range of $\{x_{t,f} : t = 1, 2, ..., m\}$ into U intervals (U is an input argument in the ALEPlot function, and generally is chosen around 100, larger values we often get a better result). $LB_{u,f}$ is assumed as the $\frac{u}{U}$ quantile of the empirical distribution of $\{x_{t,f} : t = 1, 2, ..., m\}$ that $LB_{0,f}$ is considered below the smallest observation, and $LB_{U,f}$ is considered as the largest observation. The constant is chosen in order to have $\frac{1}{m}\sum_{t=1}^{m} \hat{g}_{f,ALE}\left(x_{t,f}\right) = 0$. The second-order ALE equations and estimation are explained in the appendix.

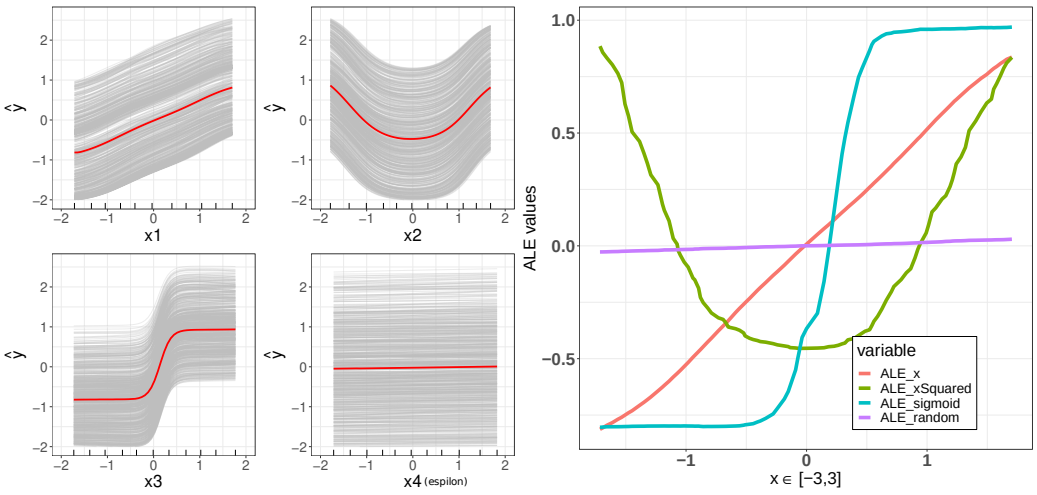
3.5. Example using simulated data

Using a simple simulation, in which the impact of each input on the output of a deterministic model is known a priori. After a fitting step using



(a) Observed and fitted values of the MLP model for the training dataset.

(b) Observed values and predictions of the MLP model for the testing dataset.



(c) The PDP results of the simulations

(d) The ALE main effect of the simulations

Figure 3: Illustration of fitted and predicted values of the ANN model using 3 layers and 6 neurons in each layer for the simulated data.

The estimated PDP and ALE of each input capture the shape of the relationship: linear, quadratic, sigmoid, and random (no impact).

a supervised machine learning model, the goal of PDP and ALE is to be able to capture the non-linear relationships between the inputs and outputs of an estimated model. To do this, we have three independent deterministic

variables (x_1, x_2, and x_3) and one random variable with no impact on y. The global model is described in the following Equation 12:

$$y = 2.5x_1 + 1.4x_2^2 + 15\left(\frac{\exp(1+5x_3)}{\exp(-1+5x_3)}\right) + \varepsilon_i \tag{12}$$

The three deterministic variables are simulated from a uniform distribution on the interval $[-3, 3]$ and the ε_i from a normal distribution, $\varepsilon_i \sim \mathcal{N}(0, 1)$. From the model (12), $n = 5000$ observations are generated, and a neural network (3 layers with 6 neurons in each layer) is trained with scaled data for the training set that is considered 85% of the dataset (4250) and them testing with 15% of the dataset (750). The fitted and predicted values of the ANN are shown in Figure 3-(a) and 3-(b). These plots prove the accuracy of the model. In parallel, we would like to point out that the original and main focus of the paper is on the interpretability point of view. The PDP and ALE are presented in Figures 3-(c) and 3-(d), respectively. Both PDP (with ICE plots) and ALE main effects accurately capture the deterministic (linear for x_1, quadratic for x_2, sigmoid for x_3) and random (for ε_i) effects. Note that the detection of interactions due to correlations between predictor variables is not always efficient with PDP and ICE plots. This is why ALE's main effects are preferred in a real application.

4. Case Study and data sets

As a real-world application of a complex dynamical system, a Combined Cycle Power Plant (CCPP) is considered. Generally, a CCPP contains gas turbines (GT), steam turbines (ST), and heat recovery steam generators (HRSG). The interactions between different parts of the system are complex. Sensitivity analyses, such as those discussed in this article, are therefore necessary for flexible power generation. Flexibility seeks the ability of the system to adapt to variability and uncertainty in demand and generation. Various ongoing changes in the power system are impacting the need for power production.

In the CCPP application, the gas turbine is one of the most efficient devices to convert gas fuels to mechanical and electrical power. Lately, the efficiency of the simple cycle has increased, and natural gas prices have decreased. As a result, gas turbines have been more widely used for base-load power generation, particularly in combined cycle mode, where waste heat is recovered to produce additional electricity.

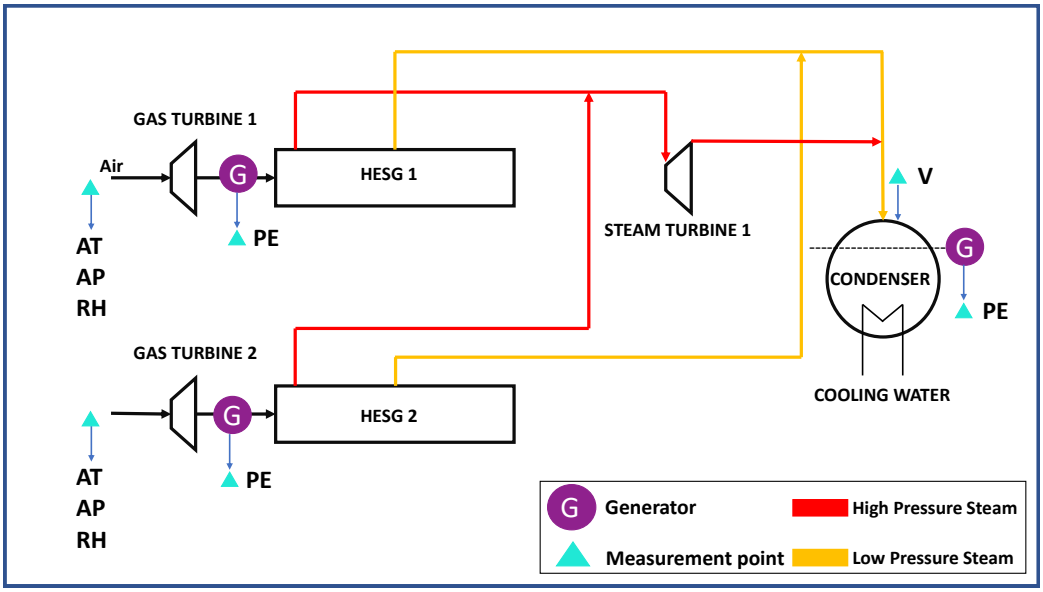


Figure 4: The schematic of combined cycle power plant layout. It contains two gas turbines, a steam turbine, and heat recovery steam generators. The figure shows the measurement points of the input and output variables.

A CCPP produces high power outputs efficiently and releases relatively low exhaust gases. Other types of power plants could generate only 33% electricity and the remaining 67% waste. In comparison, CCPP generates 68% of electricity. Due to its advantages, CCPP is increasingly used to satisfy the electricity demand. In order to adjust production to the variability of electricity demand, an efficient predictive system is desired, along with an understanding of the variables that need to be acted upon to increase flexibility. Prediction can be provided by machine learning, but sensitivity tools must be used to understand how input variables affect production.

Consequently, predicting and interpreting the prediction model of a power plant has been investigated as a crucial real-world problem. Knowing the influential factors to accurately predict an electrical power output is essential for a power plant's efficiency. It is beneficial for maximizing the income from the available megawatt hours (MWh). The reliability and sustainability of a power plant are significantly related to predicting its power generation, especially when there are some high efficiency and contractual liability constraints.

Figure 4 illustrates the CCPP and the sensors' locations within the CCPP

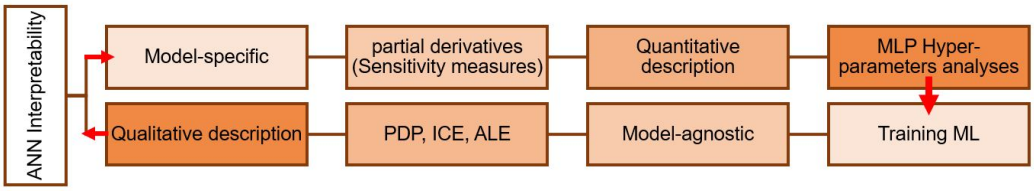


Figure 5: Overview of the result presentation. There are two steps to enhance interpretability. The hybridization is based on two main sides: the first one is based on partial derivatives methods of ANN (model-specific), which gives a quantitative description of sensitivity, and the second one is based on model-agnostic (PDP, ICE, ALE), which provides a qualitative description.

installation. The CCPP is affected by ambient conditions, mostly ambient temperature (AT), atmospheric pressure (AP), and relative humidity (RH). However, the steam turbine is affected by the exhaust steam pressure (or vacuum, V). These parameters could be considered as input variables for the two turbines. The electrical power, generated by both gas and steam turbines, is considered as a target variable. All the input variables and the target variable are average hourly data that are measured by the sensors located in the measurement points in Figure 4. The measured data consists of 9568 data points collected when the plant worked with a full load over 674 different days (Tüfekci, 2014).

The significant difference with Tüfekci's paper is that we hybridize model-specific and model-agnostic methods in the framework of supervised machine learning approaches to understand and visualize the effects of the predictor variables on the predicted response.

5. Results and discussion

Figure 5 shows the overview of the result section. The first part of our analysis focuses on optimizing the ANN model's hyper-parameters, namely the number of layers and neurons in each layer. In that respect, the impact of these parameters on the sensitivity measures was evaluated. After validating the accuracy of ANN's predictions, sensitivity measures of the model have been applied to assess the impact of input parameters on the output variability (Section 5.1). This step could be considered a model-specific step that gives us a quantitative description. After that, we perform PDP, c-ICE, and ALE plots to visualize and describe the predictors' effects with the ANN

model with different architectures as a qualitative description. We use two different architectures to study their impact on our results (Sections 5.2 and 5.3).

In the case study, there are four input parameters (a few numbers), so the study is performed on all of them. But for systems with numerous variables, the permutation feature importance method could be applied first to prioritize the variables. The idea is fairly simple: the increase in the model's prediction error is calculated after permuting a feature to determine how important it is. If changing a feature's values causes the model error to rise and the feature was used by the model to make the prediction, the feature is considered "important." If changing a feature's values causes the model error to remain constant, the feature was disregarded for the prediction, making it "unimportant" (Fisher et al., 2019). After that, other model-agnostic methods could be applied to know how they affect the output and what the relationship between them is (Molnar, 2019).

All the results, simulations, and plots are obtained from the R software (Team et al., 2013). The ANN regressions and PDP results are obtained thanks to `pdp` (Greenwell, 2017) and `RSNNS` (Bergmeir et al., 2012) R packages.

5.1. MLP Hyper-parameters analyses

Several tests of the MLP hyper-parameter have been applied to optimize the network; the number of hidden layers from 2 to 7 and the number of neurons from 2 to 10 for each layer were changed. The same number of neurons is considered for all layers. Seven layers maximum is chosen because the mean sensitivity values do not change remarkably after 6 or 7 layers, and ten neurons maximum is chosen because adding more neurons would be time-consuming.

The conclusions from each sensitivity measure were identical. The mean sensitivity result is shown as an example. Figure 6 depicts the mean sensitivity of the neural networks as a function of the number of neurons and hidden layers. It seems that by increasing the number of layers, the mean sensitivity tends to zero intensely at first and then almost keeps a constant value as the number increases.

The foremost observation in Figure 6 is that three of our input parameters (AT, V, and RH) have pretty similar behavior. However, atmospheric pressure does not give the same result as others. From Figure 6, it can be

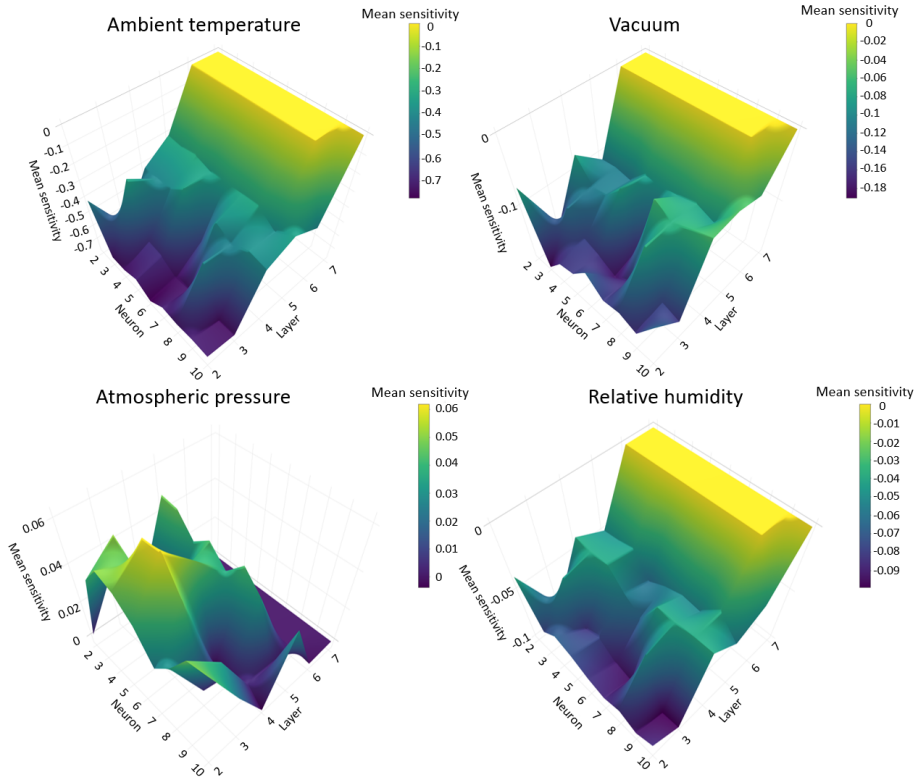


Figure 6: The Mean sensitivities obtained on the MLP neural network's output as a function of the number of neurons and hidden layers. The lowest sensitivity shows the adequate number of neurons and layers.

concluded that three hidden layers and six neurons would match low sensitivity in the output (RMSE = 4.0662). Accordingly, these values for the MLP's hyper-parameters are considered.

5.2. PDP and ICE plots

Figures 7 and 8 show the PDP and c-ICE simultaneously for different neural network architectures when the data are normalized. Figure 7 presents PDP and c-ICE for an ANN prediction with one layer and fifty neurons, and Figure 8 presents PDP and c-ICE for an MLP with three layers and six neurons.

Some assumptions for PDP should be met to have the ability to show the way an input impacts an outcome variable. More accurately, this plot discovers the relationship between the predicted response and the selected

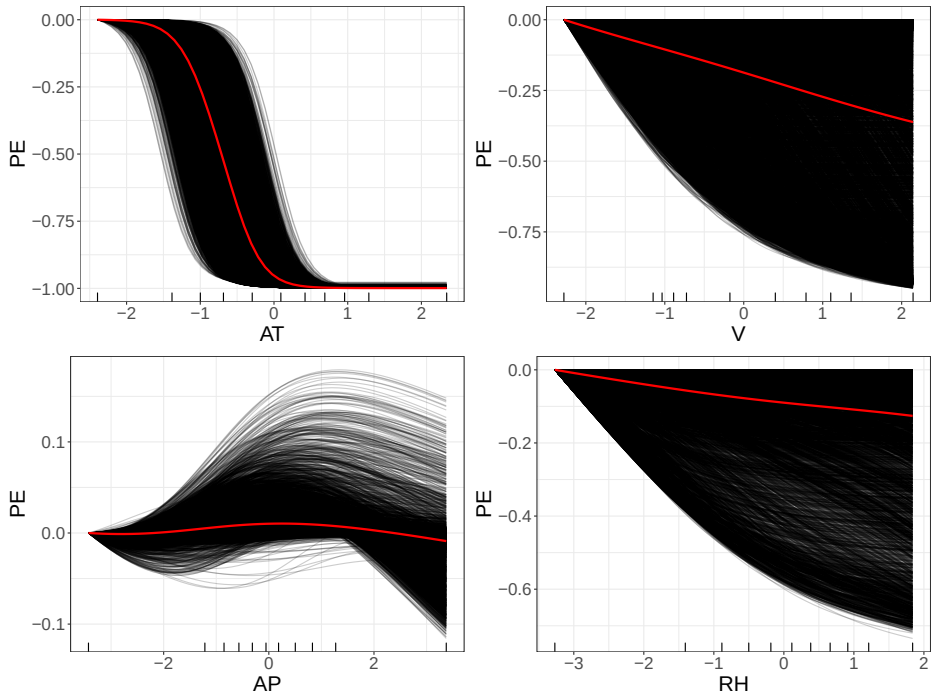


Figure 7: Partial dependence plots (red) and centred individual conditional expectation (c-ICE) plots (black) of a neural network (one layer and fifty neurons) for PE predictions. The PDP and c-ICE are computed after the MLP learning for PE predictions. All variables are standardized during the learning step and kept dimensionless in the PDP computations step.

input variables (Molnar, 2019). The PDP shows the marginal effect of one or two features on the predicted PE. It indicates whether the relationship between the PE and input variables (AT, V, AP, and RH) is nonlinear, monotonic, or more complex.

Figure 7 illustrates that the AT plot is the most complex figure among these four inputs. It is divided into three parts. It is partly linear in the first and third parts. In the middle, there is a complex variation. The curve shape of AT reminds us of the inverse sigmoidal function ($PE = \frac{\alpha}{1+\beta\gamma^{-AT}}$ for $0 < \gamma < 1$) which is consistent with the earlier research (Arrieta and Lora, 2005). The AP plot is divided into two parts: the first part is almost linear, and in the second part, it could be seen more complex values. The RH and V plots are similar and partly linear.

In general, there are smoother results in Figure 7 than in Figure 8 for

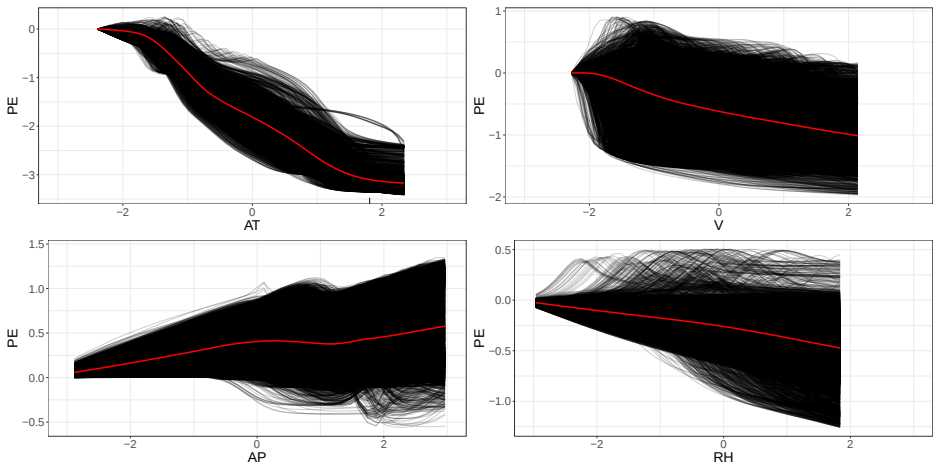


Figure 8: Partial dependence plots (red) and centred individual conditional expectation (c-ICE) plots (black) of an MLP neural network (three layers and six neurons) for PE predictions. The PDP and c-ICE are computed after the MLP learning for PE predictions. All variables are standardized during the learning step and kept dimensionless in the PDP computations step.

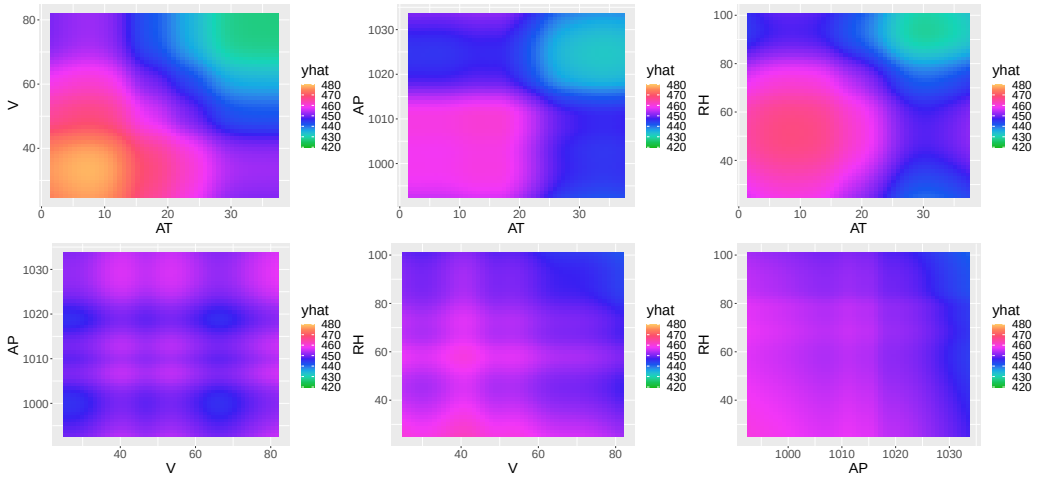


Figure 9: The 2D PDP of variables combination of ANN predictions (50 neurons and one layer) for the CCPP data values. The gradient legend shows the sensitivity of the neural network output $\widehat{PE}$ (yhat) to the variability of two variables, with a uniform color bar.

c-ICE plots. For example, the sinusoidal turbulence on top of RH plots can be seen. However, the same trend for PDPs is observed approximately. Con-

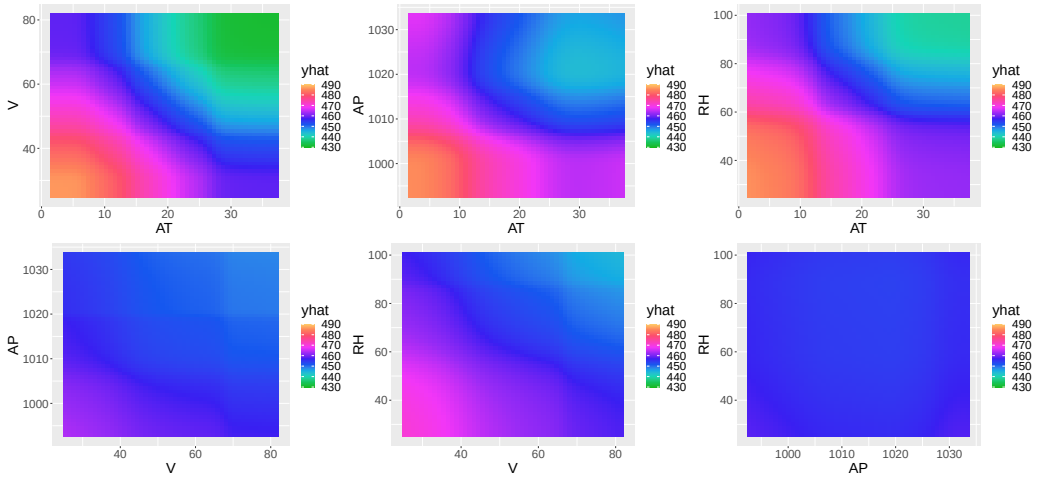


Figure 10: The 2D PDP of variables combination of MLP neural network predictions (3 layers and 6 neurons) for the CCPP data values. The gradient legend shows the sensitivity of the neural network output $\widehat{PE}$ (yhat) to the variability of two variables, with a uniform color bar.

sequently, PDP and c-ICE are MLP architecture-independent, i.e., changing the hyper-parameters (neither hidden layer nor neurons) impacts the prediction response behavior.

Figures 9 and 10 show the sensitivity of the output based on the variability of two input variables for both neural networks with one layer and fifty neurons and three layers and six neurons, respectively. They are more useful in comparing the effects and relations between two variables. These figures illustrate which areas have a more or less high and homogeneous PE. They show us that the output does not vary linearly with the simultaneous variability of two variables and how high PE values can be achieved.

In the subplots of Figure 9 with the variability of AT and other input variables (V, AP, and RH), the relation between PE and the variability of two inputs is almost linear; increasing inputs makes PE decrease. The maximum values of PE could be reached when the temperature is lower than $15°C$. Furthermore, the same results for the variability of AT and other input variables are obtained approximately (when the variables are low, we get a higher PE). It could be concluded that AT is the most effective parameter in our case. In the subplot with the variability of vacuum and atmospheric pressure, PE is high when the atmospheric pressure is high. In the subplot

with the vacuum and relative humidity variability, PE could obtain its maximum values when the vacuum is between 35 and 45 cm Hg and the relative humidity is less than 30% or between 50% and 65%. The relation between PE and the variability of atmospheric pressure and relative humidity is not linear. We get a lower PE when AP is more than 1025 mbar. PE is nearly independent of RH.

In the subplots of Figure 10 with the variability of ambient temperature and other input variables (V, AP, and RH), the same result as in Figure 9 is nearly achieved. The relation between PE and the variability of two inputs is almost the same, and lower PE is obtained when the inputs are increased. The maximum values of PE could be reached when the temperature is lower than $10°C$. In the subplot with ambient temperature and relative humidity variability, for RH less than 50%, PE is nearly independent of RH. In the subplot with the variability of vacuum and atmospheric pressure, the lowest part is observed when the highest value of input variables is present, and the highest part is observed when the lowest value of input variables is present. Variability in vacuum and relative humidity could affect the output almost linearly. The subplot with the variability of relative humidity and atmospheric pressure shows that the variation of these two variables does not have too much effect on PE.

5.3. ALE plot

Figure 11 shows the ALE main-effect plot of MLP architecture. It reveals the main effect of the input variables. The AT's main effect has inverse sigmoidal behavior. Increasing the AT makes PE decrease, which is coherent with the previous studies (Arrieta and Lora, 2005). The RH main effect behaves quadratically.

Figure 12 is the ALE second-order effect plot without the main effect of each input variable. It reveals the interaction between input variables for an MLP neural network with three layers and six neurons (for one layer and 50 neurons, see Figure B.14 in the appendix). The numbers on the contours show the function values. The darker the chart color, the higher the function value.

In Figure 12, there is lower interaction compared to Figure B.14, although the critical and sensitive points remain the same. For example, in both Figures 12 and B.14, the crucial point in the subplot AT-V is when AT is around -1.5 and V is around +1 for normalized data. It can be concluded

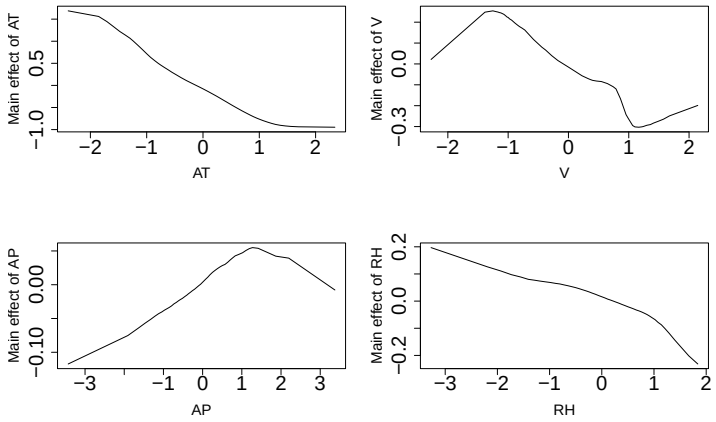


Figure 11: ALE main-effect plots for MLP neural network with 3 layers and 6 neurons for scaled data

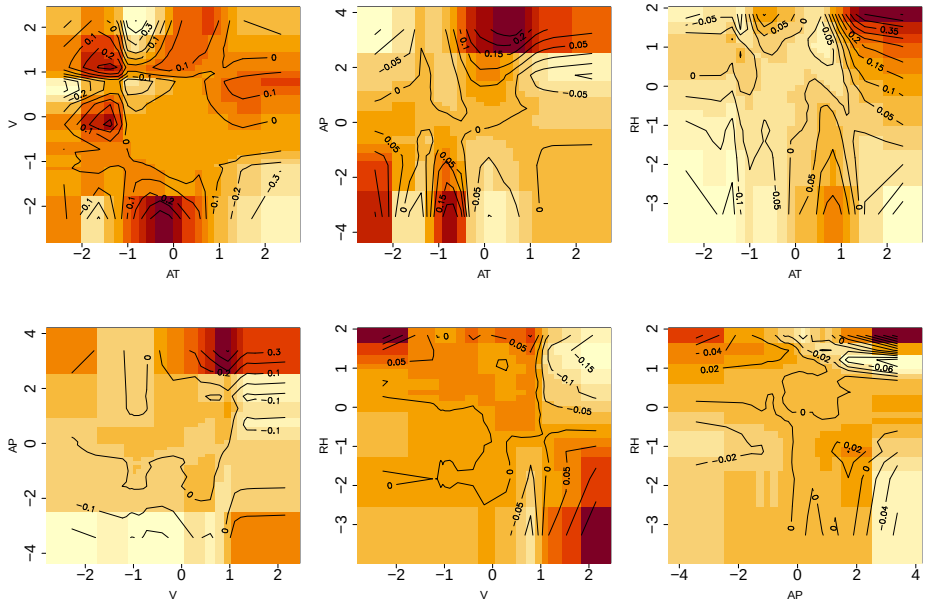


Figure 12: ALE second-order effect plots for MLP neural network with 3 layers and 6 neurons. The numbers on the contours represent the function values. The darker the chart color, the higher the function value. All variables are scaled before the MLP learning step.

that AT and V have the most interaction, and AP and RH have the most negligible interaction, based on both Figures 12 and B.14.

6. Conclusion

The present study was designed to hybridize model-specific and model-agnostic methods, which would allow reconciling the prediction accuracy and the interpretability level and prove that these methods can be applied to any engineering application. Two basic approaches have been tested: sensitivity analysis through partial derivatives of ANN predictions as model-specific to optimize the hyper-parameters of the ANN and PDP, ICE, and ALE as model-agnostic methods for visualization and description aspects. The curves exhibit the interaction shape between input and output variables and reveal the most important input variables. Concerning the overall methodology, one can ask: what if the process is applied with an alternative permutation, i.e. the use of a model-agnostic for the selection of hyperparameters and a model specific for the explanations? An experimental design will be needed to cover all simulations. It will also require substantial computing resources to run these simulations.

These techniques were applied to a full-load combined cycle power plant to make systems more flexible. In the CCPP application, flexibility using ML interpretability tools is designed as the ability to adapt to variability and uncertainty in demand and production.

The functional relationship provided by these tools is an important model for diagnostic techniques. However, there are still some deficiencies in the PDP. It is not trustable in complex systems and data because its computation requires averaging predictions of unrealistic artificial data instances if features of a machine learning model are statistically not independent. ALE plots are faster to compute than PDPs, but the equivalent of ICE curves presented for the PD plots do not appear in ALE plots.

A further research objective will include the comparison of model-agnostic methods for different temporal neural network architectures. For example, Recurrent neural net (RNN) and Long short-term memory (LSTM) are used in the field of deep learning for time series. Moreover, other model-agnostic methods could be applied and compared, such as global surrogate models, local interpretable model-agnostic explanations, and permutation feature importance. Additionally, the interpretability of other machine learning methods, such as random forest and support vector machine, could be examined.

Nomenclature

n	Number of neurons
l	Number of layers
m	Number of sample
f, h	Number of predictors or features
F	Total number of predictors or features
u, v	Number of interval
U, V	Total number of interval
$g(x)$	A black-box supervised learning model; here is a neural network
$\hat{g}(x)$	The fitted neural network model
$LB_{0,f}$	The approximate lower bounds of X_f
$LB_{0,h}$	The approximate lower bounds of X_h
$LB_{U,f}$	The largest observation
X	Random variables
x	Specific values of the random variables
S^{avg}	Mean sensitivity
S^{sd}	Standard deviation sensitivity
S^{sq}	Mean squared sensitivity
AT	Ambient Temperature in degrees Celsius
AP	Atmospheric Pressure in mbar
V	Vacuum in cm Hg
RH	Relative Humidity in percentage
PE	Electrical Power output in megawatts

CCPP Combined Cycle Power Plant

$\widehat{PE}$ Electrical Power output predictions using ANN

PDP Partial Dependence Plots

ICE Individual Conditional Expectation

c-ICE Centered Individual Conditional Expectation

ALE Accumulated Local Effects

Appendix A. ALE second-order formulation

To define the ALE second-order effects, for each pair of indices $\{f, h\} \subseteq \{1, ..., F\}$, let $X_{\setminus f,h}$ illustrate the subset of $(F-2)$ predictors excepting $\{X_f, X_h\}$. The ALE second-order effect of predictors $\{X_f, X_h\}$ is defined by the following equation:

$$\begin{aligned} &\hat{g}_{\{f,h\},ALE}\left(x_f, x_h\right) = \\ &\int_{LB_{0,h}}^{x_h}\int_{LB_{0,f}}^{x_f} E\left[\frac{\partial^2 \hat{g}(X_1, ..., X_F)}{\partial X_f \partial X_h} | X_f = LB_f, X_h = LB_h\right] dLB_f dLB_h \\ &- f_f(x_f) - f_h(x_h) - C \end{aligned} \tag{A.1}$$

Where, $LB_{0,f}$ and $LB_{0,h}$ refer to approximate lower bounds of X_f and X_h, respectively. The functions of single variables X_f and X_h ($f_f(x_f)$ and $f_h(x_h)$) and the constant aims to centralized $\hat{g}_{\{f,h\},ALE}\left(x_f, x_h\right)$ or has the mean of equal to zero concerning the marginal distribution of X_f and X_h.

Estimation of ALE second-order effects for $\{X_f, X_h\}$ at any $(x_f, x_h) \in (LB_{0,f}, LB_{U,f}] \times (LB_{0,h}, LB_{U,h}]$:

$$\begin{aligned} \hat{\tilde{g}}_{\{f,h\},ALE}\left(x_f, x_h\right) = &\sum_{u=1}^{u_f(x_f)} \sum_{v=1}^{v_h(x_h)} \frac{1}{m_{\{f,h\}(u,v)}} \\ &\sum_{t:x_{t,\{f,h\}} \in M_{\{f,h\}}(u,v)} \left[\hat{g}(LB_{u,f}, LB_{v,h}, x_{t,\setminus\{f,h\}}) - \hat{g}(LB_{u-1,f}, LB_{v,h}, x_{t,\setminus\{f,h\}})\right] - \\ &\left[\hat{g}(LB_{u,f}, LB_{v-1,h}, x_{t,\setminus\{f,h\}}) - \hat{g}(LB_{u-1,f}, LB_{v-1,h}, x_{t,\setminus\{f,h\}})\right] - C \end{aligned} \tag{A.2}$$

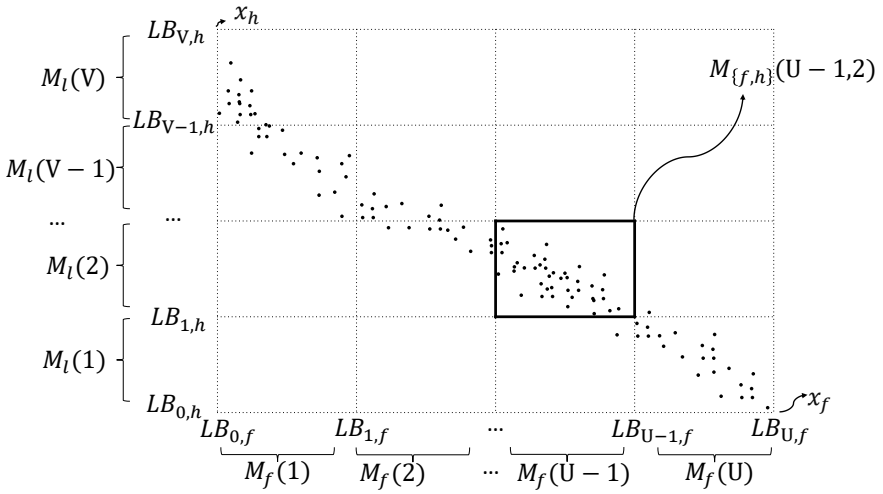


Figure A.13: Clarification of the notations utilized in estimating ALE second-order effects adopted from (Apley and Zhu, 2020). Each of $\{X_f, X_h\}$ are split up into U and V intervals respectively, and rectangular cells of the grid come from their cross product.

Where the $\{X_f, X_h\}$ space is split up into a grid of $U \times V$ rectangular cells $\{M_{f,h}(u,v) = M_f(u) \times M_h(v); u = 1, 2, ..., U; v = 1, 2, ..., V\}$ shown in Figure A.13. For each $u \in \{1, 2, ..., U\}$ and $v \in \{1, 2, ..., V\}$, $m_{f,h}(u,v)$ refers to the number of training observation that falls into cell $M_{f,h}(u,v)$. The constant is chosen in order to center the ALE second-order effects estimation in two directions.

Appendix B. More result

Figure B.14 reveals the interaction between input variables for a neural network with fifty neurons. The interactions between AT and V since the contour values change over a range of 2.5 units (from -2 to +0.5), which is almost as large as the range for the main effect of AT (for scaled data). Figure B.14 shows almost moderate interaction in subplots AT-AP and V-AP. It demonstrates negligible interaction between other input variables.

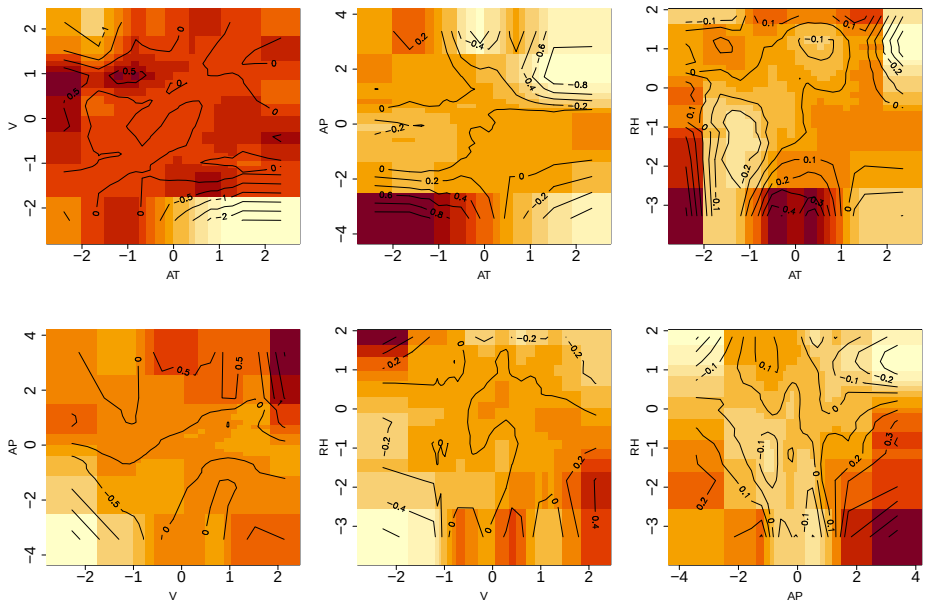


Figure B.14: ALE second-order effect plots for neural network with 50 neurons and scaled data. The numbers on the contours represents the function values. The darker the chart color, the higher the function value. All variables are scaled before MLP learning step.

References

Agarwal, P., Tamer, M., Budman, H., 2021. Explainability: Relevance based dynamic deep learning algorithm for fault detection and diagnosis in chemical processes. Computers & Chemical Engineering 154, 107467.

Amari, S., 1967. A theory of adaptive pattern classifiers. IEEE Transactions on Electronic Computers , 299–307.

Apley, D.W., Zhu, J., 2020. Visualizing the effects of predictor variables in black box supervised learning models. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 82, 1059–1086.

Arrieta, F.R.P., Lora, E.E.S., 2005. Influence of ambient temperature on combined-cycle power-plant performance. Applied energy 80, 261–272.

Beck, M.W., 2018. Neuralnettools: visualization and analysis tools for neural networks. Journal of statistical software 85, 1.

Bequette, B.W., 2003. Process control: modeling, design, and simulation. Prentice Hall Professional.

Bergmeir, C.N., Benítez Sánchez, J.M., et al., 2012. Neural networks in r using the stuttgart neural network simulator: Rsnns, American Statistical Association.

Bhakte, A., Pakkiriswamy, V., Srinivasan, R., 2022. An explainable artificial intelligence based approach for interpretation of fault classification results from deep neural networks. Chemical Engineering Science 250, 117373.

Burkart, N., Huber, M., Faller, P., 2019. Forcing interpretability for deep neural networks through rule-based regularization, in: 2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA), IEEE. pp. 700–705.

Chen, C.P., Zhang, C.Y., 2014. Data-intensive applications, challenges, techniques and technologies: A survey on big data. Information sciences 275, 314–347.

Cybenko, G., 1989. Approximation by superpositions of a sigmoidal function. Mathematics of control, signals and systems 2, 303–314.

Doshi-Velez, F., Kim, B., 2017. Towards a rigorous science of interpretable machine learning. arXiv preprint arXiv:1702.08608 .

Du, M., Liu, N., Hu, X., 2019. Techniques for interpretable machine learning. Communications of the ACM 63, 68–77.

Fisher, A., Rudin, C., Dominici, F., 2019. All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously. J. Mach. Learn. Res. 20, 1–81.

Freitas, A.A., 2014. Comprehensible classification models: a position paper. ACM SIGKDD explorations newsletter 15, 1–10.

Friedman, J.H., 2001. Greedy function approximation: a gradient boosting machine. Annals of statistics , 1189–1232.

Gajjar, S., Palazoglu, A., 2016. A data-driven multidimensional visualization technique for process fault detection and diagnosis. Chemometrics and Intelligent Laboratory Systems 154, 122–136.

Goldstein, A., Kapelner, A., Bleich, J., Pitkin, E., 2015. Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. journal of Computational and Graphical Statistics 24, 44–65.

Goodman, B., Flaxman, S., 2017. European union regulations on algorithmic decision-making and a "right to explanation". AI magazine 38, 50–57.

Greenwell, B.M., 2017. pdp: an r package for constructing partial dependence plots. R J. 9, 421.

Harmouche, J., Delpha, C., Diallo, D., 2014. Incipient fault detection and diagnosis based on kullback–leibler divergence using principal component analysis: Part i. Signal processing 94, 278–287.

Harmouche, J., Delpha, C., Diallo, D., 2015. Incipient fault detection and diagnosis based on kullback–leibler divergence using principal component analysis: Part ii. Signal Processing 109, 334–344.

Hornik, K., 1991. Approximation capabilities of multilayer feedforward networks. Neural networks 4, 251–257.

Kesgin, U., Heperkan, H., 2005. Simulation of thermodynamic systems using soft computing techniques. International journal of energy research 29, 581–611.

Lee, J.H., Shin, J., Realff, M.J., 2018. Machine learning: Overview of the recent progresses and implications for the process systems engineering field. Computers & Chemical Engineering 114, 111–121.

Leshno, M., Lin, V.Y., Pinkus, A., Schocken, S., 1993. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. Neural networks 6, 861–867.

Li, D., Li, X., Zhang, Y., Sun, L., Yuan, S., 2019. Four methods to estimate minimum miscibility pressure of co2-oil based on machine learning. Chinese Journal of Chemistry 37, 1271–1278.

Lipton, Z.C., 2018. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. Queue 16, 31–57.

Lipton, Z.C., Kale, D.C., Wetzel, R., et al., 2016. Modeling missing data in clinical time series with rnns. Machine Learning for Healthcare 56, 253–270.

Molnar, C., 2019. Interpretable Machine Learning.

Moradi, M., Samwald, M., 2021. Post-hoc explanation of black-box classifiers using confident itemsets. Expert Systems with Applications 165, 113941.

Ning, C., You, F., 2019. Optimization under uncertainty in the era of big data and deep learning: When machine learning meets mathematical programming. Computers & Chemical Engineering 125, 434–448.

Park, J., Sandberg, I.W., 1991. Universal approximation using radial-basis-function networks. Neural computation 3, 246–257.

Pizarroso, J., Portela, J., Muñoz, A., 2020. Neuralsens: sensitivity analysis of neural networks. arXiv preprint arXiv:2002.11423 .

Qazani, M.R.C., Pourmostaghimi, V., Moayyedian, M., Pedrammehr, S., 2022. Estimation of tool–chip contact length using optimized machine

learning in orthogonal cutting. Engineering Applications of Artificial Intelligence 114, 105118.

Ramirez, W.F., 1997. Computational methods for process simulation. Butterworth-Heinemann.

Ribeiro, M.T., Singh, S., Guestrin, C., 2016a. " why should i trust you?" explaining the predictions of any classifier, in: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, pp. 1135–1144.

Ribeiro, M.T., Singh, S., Guestrin, C., 2016b. Model-agnostic interpretability of machine learning. arXiv:1606.05386.

Sahoo, S., Russo, T., Elliott, J., Foster, I., 2017. Machine learning algorithms for modeling groundwater level changes in agricultural regions of the us. Water Resources Research 53, 3878–3895.

Saltelli, A., 2002. Sensitivity analysis for importance assessment. Risk analysis 22, 579–590.

Saltelli, A., Annoni, P., Azzini, I., Campolongo, F., Ratto, M., Tarantola, S., 2010. Variance based sensitivity analysis of model output. design and estimator for the total sensitivity index. Computer physics communications 181, 259–270.

Saltelli, A., Ratto, M., Andres, T., Campolongo, F., Cariboni, J., Gatelli, D., Saisana, M., Tarantola, S., 2008. Global sensitivity analysis: the primer. John Wiley & Sons.

Schaaf, N., Huber, M., Maucher, J., 2019. Enhancing decision tree based interpretation of deep neural networks through l1-orthogonal regularization, in: 2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA), IEEE. pp. 42–49.

Schmidhuber, J., 2015. Deep learning in neural networks: An overview. Neural networks 61, 85–117.

Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D., 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization, in: Proceedings of the IEEE international conference on computer vision, pp. 618–626.

Shrikumar, A., Greenside, P., Shcherbina, A., Kundaje, A., 2016. Not just a black box: Learning important features through propagating activation differences. arXiv preprint arXiv:1605.01713 .

Simonyan, K., Vedaldi, A., Zisserman, A., 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. arXiv preprint arXiv:1312.6034 .

Sobol, I.M., 1998. On quasi-monte carlo integrations. Mathematics and computers in simulation 47, 103–112.

Team, R.C., et al., 2013. R: A language and environment for statistical computing .

Tüfekci, P., 2014. Prediction of full load electrical power output of a base load operated combined cycle power plant using machine learning methods. International Journal of Electrical Power & Energy Systems 60, 126–140.

Venkatasubramanian, V., Rengaswamy, R., Kavuri, S.N., 2003a. A review of process fault detection and diagnosis: Part ii: Qualitative models and search strategies. Computers & chemical engineering 27, 313–326.

Venkatasubramanian, V., Rengaswamy, R., Kavuri, S.N., Yin, K., 2003b. A review of process fault detection and diagnosis: Part iii: Process history based methods. Computers & chemical engineering 27, 327–346.

Venkatasubramanian, V., Rengaswamy, R., Yin, K., Kavuri, S.N., 2003c. A review of process fault detection and diagnosis: Part i: Quantitative model-based methods. Computers & chemical engineering 27, 293–311.

Wachter, S., Mittelstadt, B., Russell, C., 2017. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. Harv. JL & Tech. 31, 841.

White, H., Racine, J., 2001. Statistical inference, the bootstrap, and neural-network modeling with application to foreign exchange rates. IEEE Transactions on Neural Networks 12, 657–673.

Xia, Y., Yi, W., Zhang, D., 2022. Coupled extreme learning machine and particle swarm optimization variant for projectile aerodynamic identification. Engineering Applications of Artificial Intelligence 114, 105100.

Yeh, I.C., Cheng, W.L., 2010. First and second order sensitivity analysis of mlp. Neurocomputing 73, 2225–2233.