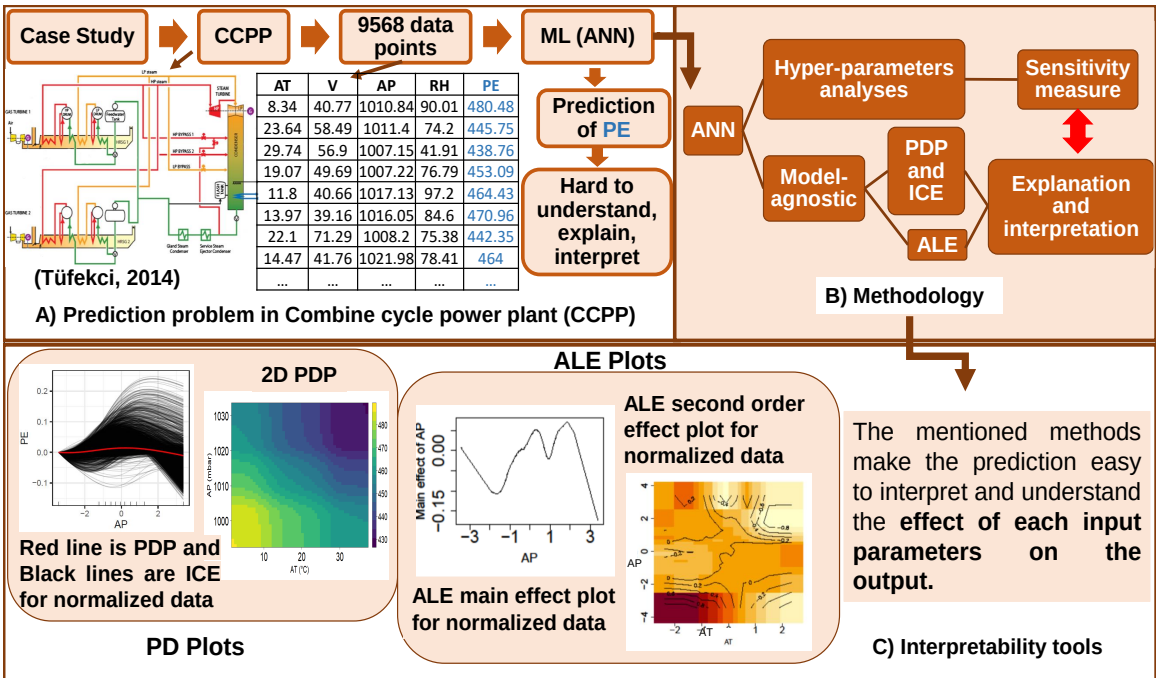
Neural Network Sensitivity and Interpretability Predictions in Power Plant Application

Tina Danesh, Rachid Ouaret, Pascal Floquet, Stéphane Négny

Graphical Abstract

Neural network sensitivity and interpretability predictions in power plant application

Tina Danesh, Rachid Ouaret, Pascal Floquet, Stephane Negny



Highlights

Neural network sensitivity and interpretability predictions in power plant application

Tina Danesh, Rachid Ouaret, Pascal Floquet, Stephane Negny

- Enhancing the interpretability of ANN predictions with the help of SA and model-agnostic techniques.
- Performing the hyper-parameters analyses with the help of sensitivity measure.
- Understanding how variations in inputs affect the predictions of a neural network.
- Identifying the most influential input parameters in the prediction of electrical power (PE) in a combined cycle power plant.

Neural network sensitivity and interpretability predictions in power plant application

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

Machine learning (ML) models such as Deep Neural Networks (DNN) have become increasingly ubiquitous due to their accuracy and flexibility. However, the lack of interpretability and explainability is why they are uncommon in engineering applications. Meanwhile, the research community has identified interpretability as a hot research topic, leading to confusion in various communities. This paper discusses a methodological framework to define and enhance interpretability in the prediction application of Neural Networks. The methods to deal with this problem are *(i)* Sensitivity Analysis (SA) for Neural Network prediction (model-specific interpretation tool) and *(ii)* model-agnostic methods. The latter tools could be used for any ML model prediction. In this study, we enhance the interpretability of the Neural Network predictions with the help of SA and model-agnostic methods. In order to visualize the inputs' impacts on prediction results, Partial Dependence Plots (PDP), Individual Conditional Expectation (ICE), and Accumulated

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)

Local Effects (ALE) are used and compared. The prediction of the electrical power (PE) output of a combined cycle power plant (CCPP) has been chosen to demonstrate the feasibility of these methods under real operating conditions. The results 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.

Keywords: Neural networks, Sensitivity analysis, Model-agnostic, Partial dependence plots, Accumulated local effects, Interpretability, Combined cycle power plant

PACS: 0000, 1111

2000 MSC: 0000, 1111

Nomenclature

Indices

f, h	Number of predictors or features
F	Total number of predictors or features
l	Number of layers
m	Number of sample
n	Number of neurons

U, V Total number of interval

u, v Number of interval

Symbols of multilayer perceptron equations

AF_n^l The activation function of the n^{th} neuron in l^{th} layer

b^l The bias in the l^{th} layer

w_{nj}^l The connection's weight between neurons j^{th} and n^{th} in the $(l-1)^{th}$ layer and the l^{th} layer

y_n^l The weighted sum of the neuron inputs

z_n^l The output of the n^{th} neuron in the l^{th} layer of an MLP

Symbols of PDP and ALE equations

$\hat{g}(x)$ The fitted neural network model

$g(x)$ A black box supervised learning model; here is a neural network

$LB_{0,f}$ 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

Symbols of sensitivity measures equations

S_{in}^{avg}, S_i^{avg} Mean sensitivity for one output neuron and more than one output neuron respectively

S_{in}^{sd}, S_i^{sd} Standard deviation sensitivity for one output neuron and more than one output neuron respectively

S_{in}^{sq}, S_i^{sq} Mean squared sensitivity for one output neuron and more than one output neuron respectively

Variables of real data

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

AP Atmospheric Pressure

AT Ambient Temperature

PE Electrical Power output

RH Relative Humidity

V Vacuum

1. Introduction

Different mathematical models exist in order to describe, analyze and predict different engineering systems' behavior. There are mainly two different visions: Equation-based knowledge models, known as physical-chemical or *"white box"* models, and data-oriented approaches, which are primarily based on Machine Learning (ML) algorithms known as *"black-box"* models.

The Equation-based methods are easy to understand and interpret predictions since the model's assumptions and the relationship 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. In parallel, Machine Learning has been heavily researched and widely used in many areas, such as in engineering design (Sharpe et al., 2019; Balochian and Baloochian, 2019; Romeo et al., 2020), and optimization plants (Mafarja et al., 2019; Tubishat et al., 2020; Tso et al., 2020). The success of ML in many applications is grounded in its powerful capability for prediction purposes. However, they are still hard to understand the relations between predictors and model outcomes (Moradi and Samwald, 2021). Indeed, they suffer from a lack of interpretability and explainability because they function without process knowledge dependency. This is the reason why they are referred to as *black-box models* (lack of transparency and physical significance). Extensive literature attests to the superiority of black-box machine learning algorithms in minimizing predictive error, 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).

From the chemical engineering perspective, the detailed description of the whole system requires complex and often highly parameterized models with numerous assumptions that should be made to analyze and predict accurately, especially thermodynamical analysis. Most of the time, these assumptions do not meet when dealing with real data, bring some uncertainty to the systems and affect the output. However, a thermodynamical analysis of a real application consists of many nonlinear equations; hence it is time-

consuming and takes too much effort. On the other side, chemical engineering problems have become complicated dramatically and consist of more and more data to analyze, especially if the system under study is large and has complex nonlinear behavior. Furthermore, they have some features such as uncertainty, multi-scale, time lag, and large variable space dimensions. Data-oriented and machine learning techniques would be helpful to deal with these barriers (Kesgin and Heperkan, 2005).

Machine learning techniques are applied mainly as alternatives instead of physical approaches, considering the increasing volume of data and information in real-world situations (Chen and Zhang, 2014). These data-oriented models can be applied to take out useful information and supports decision-maker. One of the most popular machine learning algorithms in continuous output predictions is Artificial Neural Network (ANN) thanks to the universal approximation theorem (Hornik, 1991).

Multi-Layer Perceptron (MLP), a subset of ANN architectures, is trained by the back-propagation algorithm and has a wide variety of applications (Rumelhart et al., 1986). The MLP learning is utilized to predict the response of one or more variables given one or many explanatory variables. The significant feature of the MLP is the description of relationships using an arbitrary number of parameters selected via iterative training with the back-propagation algorithm. Conceptually, the MLP as a hyper parameterized nonlinear model could fit a smooth function to any dataset with the minimum residual error representing the relationship between the output and the input variables (Hornik, 1991). 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, making it hard to understand why it predicts successfully and when we can trust it.

A recent advanced research topic in neural networks is to find methods to obtain information and gain the ability to interpret how the input variables affect the output variable to help decision-makers. The interpretability could be handled by using sensitivity analysis 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 overview of the machine learning interpretability procedure. Firstly, the goal is to predict the system's output and help the decision-maker decide and control. For this purpose, a supervised ML model such as ANN is used. In order to give valuable information to the decision-maker, the interpretability methods as model-agnostic methods attempt to address the question of how the inputs impact the model's predictive performance. The interpretability tools are employed after predictions.

In this paper, the prediction *interpretability* of regression problems is defined as the process of extracting relevant knowledge from a model about the learned relationships between features and model outputs. These aspects have been addressed in the sensitivity analysis framework of neural network predictions. Enhancing interpretation, in our context, consists of distinguishing the effect of each input uncertainty on the model output variance.

Some examples of methods for sensitivity analysis that could help to gain helpful information from the neural networks are Neural Interpretation

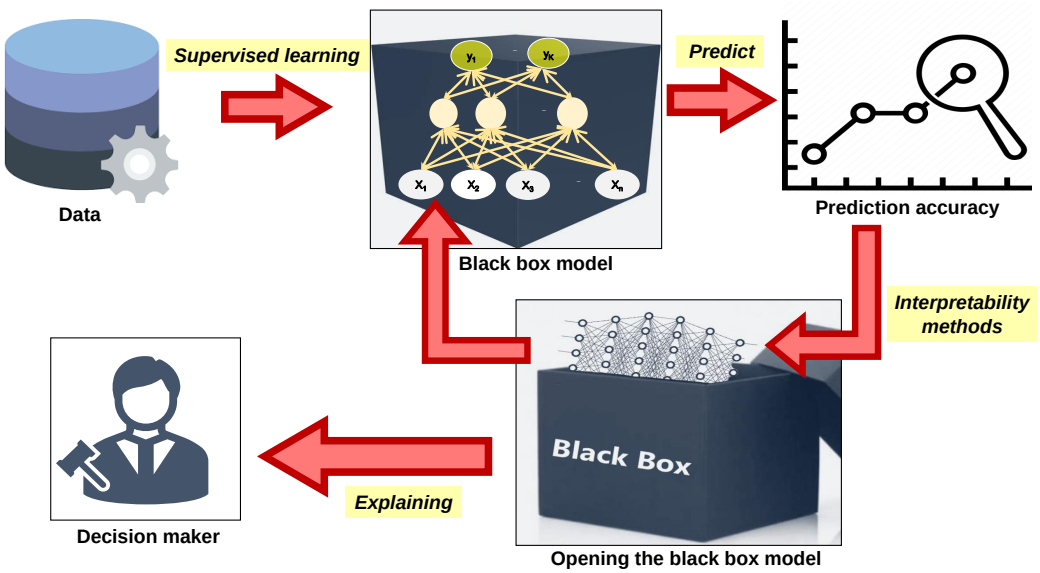


Figure 1: Overview of Machine Learning interpretability.

Diagram (NID) (Özesmi and Özesmi, 1999; Olden and Jackson, 2002), Garson's method for variable importance (Garson, 1991), and partial derivatives method (White and Racine, 2001).

The partial derivatives method calculates the derivative of each output variable according to each input variable evaluated on each data sample of a specified dataset. Each input's effect using all the dataset samples is computed in both magnitude and sign concerning the connection weights, the activation functions, and the values of each input to avoid information loss during learning or calibration steps. Analytically calculating the derivatives gives more robust diagnostic information since it only 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). Its

ability to make sensitivity analysis a beneficial technique for interpreting and improving neural network models is considered its main advantage.

As mentioned before, it is not simple to interpret the machine learning models. The general interpretative framework depends on the models. For example, in linear regression, it is possible and straightforward 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) 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 not easy to extrapolate this process to other models, so one can imagine specific methods or tools to interpret it for any model. That is the reason that 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 express the interpretation of machine learning models, whatever the learning model used. These tools are called *model-agnostic* tools.

Model-agnostic methods could be effective in order to interpret machine learning models by separating the explanations from the machine learning model (Ribeiro et al., 2016). Model-agnostic methods like sensitivity analysis (SA) are distinguished into local and global methods. The PDPs, Individual Conditional Expectation (ICE) plots, and ALE Plots are some model-agnostic techniques (Friedman, 2001; Apley and Zhu, 2020).

We carry out our study on a Combined Cycle Power Plant (CCPP) as

a real-world application. The dataset is taken from Tüfekci's paper and contains 9568 data points 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. It had two primary purposes. The first one was to detect the best subset of the dataset among all other subset configurations. The second one was to realize the most successful machine learning regression method. The Tüfekci's paper does not focus on the interpretability aspects of the applied ML methods, especially from the SA and model-agnostic points of view.

The contributions of our paper include:

- Enhancing the interpretability of ANN predictions with the help of SA and model-agnostic techniques.
- Identifying the most influential input parameters in predicting electrical power in a CCPP.
- Compare the model-specific (partial derivatives) approach and model-agnostic for interpretability purposes.
- Understanding how variations (quantitatively and qualitatively) in inputs affect the predictions of a neural network.

In our study, to calculate the sensitivity measures, the output's partial derivatives are determined concerning the inputs of an MLP model. We used

PDPs, ICE plots, and ALE plots as model-agnostic methods to visualize the input's effect on the output. The MLP theory and model-agnostic methods will be explained concerning its inputs and some sensitivity measures in the following sections.

The rest of this paper is organized as follows. In Section 2, Neural Network Sensitivity and interpretation tools are presented, whereas the experimental and simulation work is given in Section 3. Section 4 is dedicated to analyzing and discussing the findings. Finally, we conclude in Section 5.

2. Neural Network Sensitivity and interpretability

Figure 2 summarizes the methodological scheme of the study. This methodology composes of three main parts: Problem (A), Methodology (B), and result (C). Part A corresponds to the description of the problem or issue we face. In this part, we attempt to answer the following question: how do predictor variables impact the predictions of neural network regression?. Specifically, we have 9568 data points that were preprocessed by (Tüfekci, 2014). The ANN is performed on the data to predict PE, 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 comparison that we gain from each method will present in Part C. This section will explain part B in detail.

2.1. Multilayer perceptron (MLP)

An artificial neural network is a mapping between two Euclidean spaces, nonlinear with respect to its parameter θ that associates to an entry x an output $y = f(x, \theta)$. An MLP is a structure composed of several hidden layers

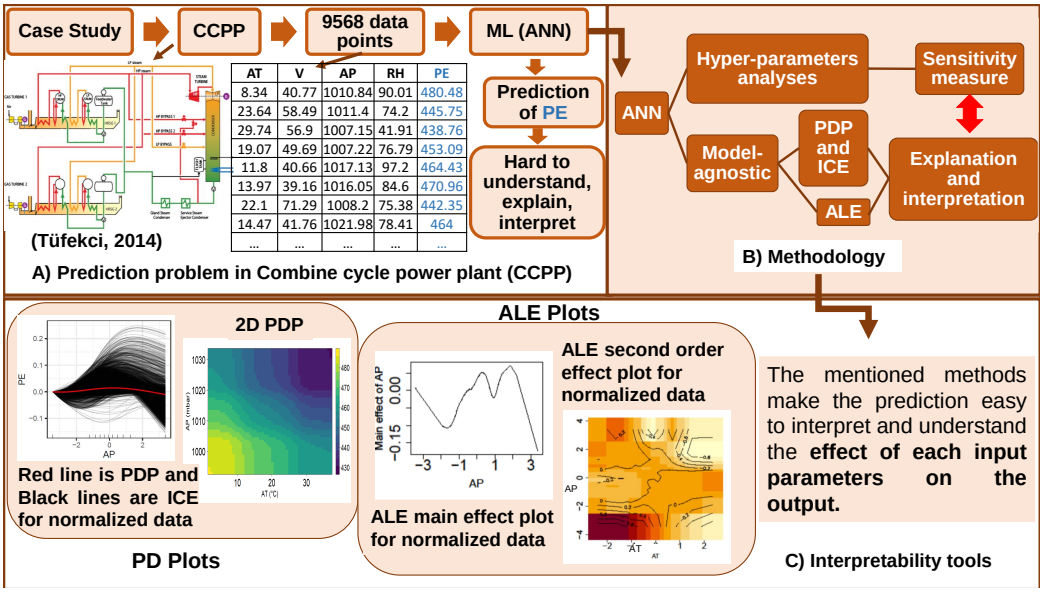


Figure 2: Overview of the methodology used in the study

of neurons (oriented graph), where the output of a neuron of a layer becomes the input of a neuron of the next layer. It contains at least three layers of nodes: an input layer of the predictor variables, a hidden layer composed of several neurons, and an output layer. The output of one section travels along with one connection to another section. It is multiplied by the weight of each connection. After that, the summation of the inputs at each section is added to a constant or bias. Once each section's input terms are calculated, an activation function is utilized to get the result (forward propagation flow). The activation function in a neural network explains how the input's weighted sum is changed into an output from a node or nodes in a network's layer.

For an MLP with L layers, the output z_n^l of the n^{th} neuron in the l^{th} layer

$(1 \leq l \leq L)$ is calculated by the following equation:

$$z_n^l = AF_n^l\left(y_n^l\right) = AF_n^l\left(\sum_{j=1}^{m^{l-1}} w_{nj}^l \,.\, z_j^{l-1} + w_{n0}^l.b^l\right) \tag{1}$$

Where AF_n^l refers to the activation function of the n^{th} neuron in l^{th} layer and y_n^l refers to the weighted sum of the neuron inputs. w_{nj}^l refers to the connection's weight between neurons j^{th} and n^{th} in the $(l-1)^{th}$ layer and the l^{th} layer. m^{l-1} refers to the number of neurons in the $(l-1)^{th}$ layer. b^l refers to the bias in the l^{th} layer. For the initial state of the input layer we have these parameters, $l = 1$, $z_j^{1-1} = x_j$, $w_{nj}^1 = 1$, and $b^1 = 0$.

Weights in the ANN structure specify how the information flows from the first layer to the last layer. The optimal weights that make the minimum prediction error are determined during the neural network training. The backpropagation algorithm is one of the most important parts of training feedforward neural networks (Rumelhart et al., 1985). Partial Derivative (PD) and gradient descent procedures are the backpropagation algorithms' most important parts. Partial derivatives are computed using an expression $\frac{\partial C}{\partial \boldsymbol{w}}$ measuring the cost function C with respect to any weight w (or bias b) in the network. The link between sensitivity analysis and the MLP learning algorithm is remarkable using partial derivatives since it quantifies the importance of each network's weights on the behavior of the output. From this point of view, the sensitivity of an MLP using the partial derivative expression tells us how quickly the cost changes when we change the weights and biases.

2.2. ANN sensitivity through partial derivatives

We could perform sensitivity analysis on the neural networks through 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 as sensitivity and can be calculated by the following equation:

$$s_{in}\,|_{\boldsymbol{x}_m} = \frac{\partial z_n}{\partial x_i}\left(\boldsymbol{x}_m\right) \tag{2}$$

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.

2.3. Sensitivity measures

After calculating the sensitivity for each variable and sample, we could apply different measures to analyze and interpret the results. The related measures are presented in table 1 under two conditions: one output neuron or more than one output neuron. In general case, the following sensitivity

Table 1: The basic sensitivity measures for an MLP using partial derivative. m is the number of samples in the dataset.

Sensitivity measure	One output neuron	More than one output neuron
Mean	$S_{in}^{avg} = \frac{\sum_{j=1}^{m} s_{in}\vert_{\boldsymbol{x}_j}}{m}$	$S_i^{avg} = \frac{\sum_{n=1}^{m^l} S_{in}^{avg}}{m^L}$
Standard deviation	$S_{in}^{sd} = \sigma\left(s_{in}\vert_{\boldsymbol{x}_j}\right); j\epsilon 1, ..., m$	$S_i^{sd} = \sqrt{\frac{\sum_{n=1}^{m^l}\left(\left(S_{in}^{sd}\right)^2 + \left(S_{in}^{avg} - S_i^{avg}\right)^2\right)}{m^L}}$
Mean squared	$S_{in}^{sq} = \sqrt{\frac{\sum_{j=1}^{M}\left(s_{in}\vert_{\boldsymbol{x}_j}\right)^2}{m}}$	$S_i^{sq} = \frac{\sum_{n=1}^{m^l} S_{in}^{sq}}{m^L}$

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 mean effect of the input variable on the output is illustrated by mean sensitivity. The variance of the input variable's effect on the output in the input space is represented by standard deviation sensitivity.

2.4. 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, 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). For example, in terms of MLP learning, all

we get is the importance of the weight. It is relatively simple to know which node connections significantly influence the outcome; it sucks that we do not know in which direction it is affecting. The PDP is an intuitive and easy-to-understand visualization 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 $\hat{g}(x)$. We use upper case X to identify random variables and 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 the machine learning model except 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:

$$\hat{g}_{f,PDP}\left(x_f\right) = E_{X_{\setminus f}}\left[\hat{g}\left(x_f, X_{\setminus f}\right)\right] = \int_{X_{\setminus f}} \hat{g}\left(x_f, X_{\setminus f}\right) dP(X_{\setminus f}) \tag{3}$$

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 the equation 3 is calculated by averaging over the training data that is known as the Monte Carlo method:

$$\hat{g}_f(x_f) = \frac{1}{m}\sum_{a=1}^{m} g(x_f, x_{\setminus f}^{(a)}) \tag{4}$$

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, we assume 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.

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.

2.5. Accumulated Local Effects plots

Accumulated Local Effects (ALE) explain the average impact of features on the prediction of a machine learning 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:

$$\hat{g}_{f,ALE}(x_f) = \int_{LB_{0,f}}^{x_f} E\left[\hat{g}^f\left(X_f, X_{\setminus f}\right) | X_f = LB_f\right] dLB_f - C \qquad (5)$$

Where, $\hat{g}^f\left(X_f, X_{\setminus f}\right) = \frac{\partial \hat{g}(X_1,...,X_F)}{\partial X_f}$. $LB_{0,f}$ 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 $\hat{g}_{f,ALE}\left(x_f\right)$ equal to zero concerning the marginal distribution of X_f or to center the plot vertically.

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:

$$\hat{g}_{\{f,h\},ALE}\left(x_f, x_h\right) = \int_{LB_{0,h}}^{x_h}\int_{LB_{0,f}}^{x_f} E\left[\frac{\partial^2 \hat{g}(X_1,...,X_F)}{\partial X_f \partial X_h}|X_f = LB_f, X_h = LB_h\right] dLB_f dLB_h - f_f(x_f) - f_h(x_h) - C \quad (6)$$

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.

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.
- We average the changes of predictions, not the predictions themselves, and we define the change as the partial derivative.

- The equation has the additional integral over $LB_{0,f}$ that refers to an approximate lower bound of X_f.
- To center the ALE plot, we subtract a constant value from the results; therefore, the average effect over the data is zero.

In order to calculate the estimation of the equations 5 and 6, 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 equations that are proposed by Apley and Zhu (2020) are as follows:

- Estimation of ALE main effect:

$$\hat{\tilde{g}}_{f,ALE}(x_f) = \sum_{u=1}^{u_f(x)} \frac{1}{m_f(u)} \sum_{t:x_{t,f} \in N_f(u)} \left[\hat{g}(LB_{u,f}, x_{t,\setminus f}) - \hat{g}(LB_{u-1,f}, x_{t,\setminus f})\right] - C \tag{7}$$

  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 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 $\frac{1}{m}\sum_{t=1}^{m} \hat{g}_{f,ALE}(x_{t,f}) = 0$.

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

$$\widehat{\widetilde{g}}_{\{f,h\},ALE}(x_f, x_h) = \sum_{u=1}^{u_f(x_f)} \sum_{v=1}^{v_h(x_h)} \frac{1}{m_{\{f,h\}(u,v)}} \sum_{t:x_{t,\{f,h\}} \in M_{\{f,h\}}(u,v)} [\hat{g}(LB_{u,f}, LB_{v,h}, x_{t,\setminus\{f,h\}}) - \hat{g}(LB_{u-1,f}, LB_{v,h}, x_{t,\setminus\{f,h\}})] - \left[\hat{g}(LB_{u,f}, LB_{v-1,h}, x_{t,\setminus\{f,h\}}) - \hat{g}(LB_{u-1,f}, LB_{v-1,h}, x_{t,\setminus\{f,h\}})\right] - C \tag{8}$$

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 3. 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.

3. Case Study and data sets

As a real-world application of a thermodynamic system, we consider a Combine cycle power plant (CCPP). Generally, a CCPP contains gas turbines (GT), steam turbines (ST), and heat recovery steam generators (HRSG). In a CCPP, the gas and steam turbines generate the electricity combined in one cycle, and the electricity is transferred from one turbine to another (Niu and Liu, 2008). The CCPP uses the waste heat to produce extra steam to generate additional electricity.

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

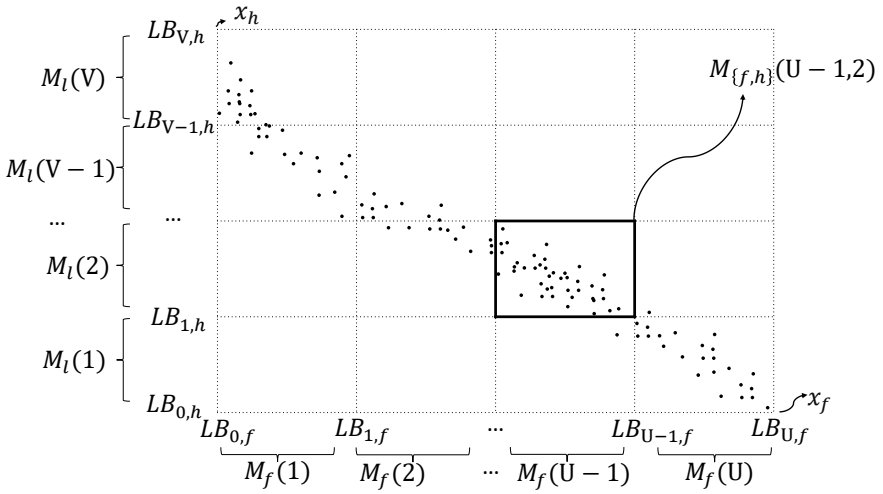


Figure 3: Clarification of the notations utilized in the estimation of ALE second-order effects adopted from (Apley and Zhu, 2020). Each of $\{X_f, X_h\}$ are split up into U and V intervals respectively, and rectangular cells of the grid come from their cross product.

increased, and the 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.

A CCPP not only produces high power outputs efficiently but also releases fairly low exhaust gases. Other types of power plants could generate only 33% electricity and the remaining 67% waste. In comparison, CCPP generates 68% electricity. Due to its advantages, CCPP is used these days increasingly. 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 a base-load power plant's full load electrical power output is essential for a power plant's efficiency. It

is beneficial for maximizing the income from the available megawatt hours (MW h). The reliability and sustainability of a power plant are related significantly to predicting its power generation, especially when there are some high efficiency and contractual liabilities constraints.

Figure 4 illustrates the CCPP and the sensors location of the CCPP installation (Tüfekci, 2014). The CCPP is affected by the 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). We could consider these parameters 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 target variables are average hourly data that are measured by the sensors located in the measurement points in figure 4.

In order to develop a predictive model, Tüfekci tested and compared several machine learning regression methods that could predict the electrical power output of a CCPP. The author worked on the dataset supplied from a power plant over six years designed with a nominal generating capacity of 480 MW, made up of two 160MW ABB 13E2 Gas Turbines, two dual pressure Heat Recovery Steam Generators (HRSG), and one 160MW ABB Steam Turbine. The measured data from the plant, as well as all preprocessing steps, are described in the original paper (Tüfekci, 2014). It consists of 9568 data points collected when the plant worked with a full load over 674 different days. Thus, some data preprocessing operations are needed. In the first step, the dataset is cleaned by removing the incompatible values and the noisy data. We perform our study on the dataset of Tüfekci's paper with

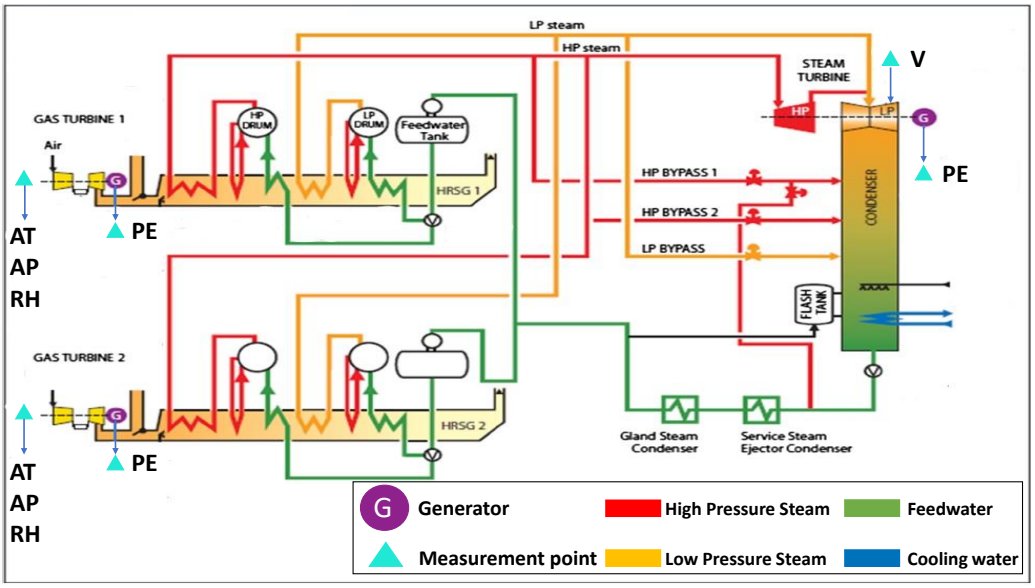


Figure 4: The combined cycle power plant layout presented in (Tüfekci, 2014). It contains two gas turbines, a steam turbine, and heat recovery steam generators. The figure shows the measurement points of the input and output variables.

the same CCPP. The significant difference with the mentioned paper is that we bring a sensitivity analysis 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.

4. Results and discussion

Figure 5 shows the overview of the result section. The first part of our analysis focuses on the optimization of the MLP model's hyper-parameters, namely the number of layers and the number of neurons in each layer (l and m in equation 1). In that respect, the impact of these parameters on the sensitivity measures was evaluated (4.1). After validating the accuracy predictions of MLP, sensitivity measures of the ANN model have been applied to assess the impact of input parameters on the output variability (section 4.1). This step could be considered as a model-specific that gives us a quantitative description. After that, we perform PDP, ICE, and ALE plots as a model-agnostic approach to visualize and describe the predictors' effects with the MLP model as a qualitative description (sections 4.2 and 4.3).

All the results, simulations, and plots are obtained from 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.

4.1. MLP Hyper-parameters analyses

4.1.1. Impact of Hyper-parameters: variation of the number of neurons for one layer

For this simulation, we assume only one hidden layer and vary the number of neurons from 15 to 100. Figure 6 shows the mean sensitivity of the neural

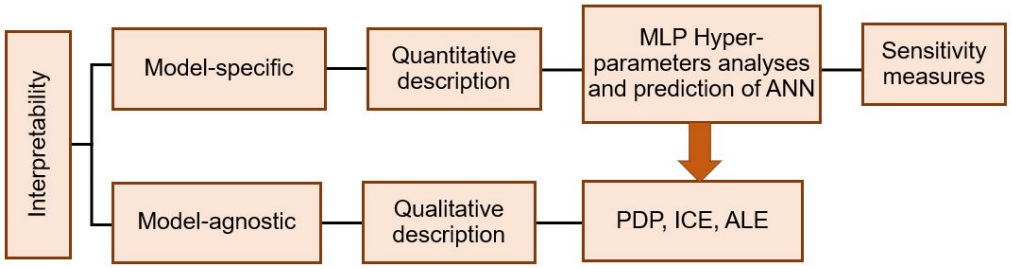


Figure 5: Overview of the result presentation. There are two steps to enhance the interpretability. The first one is named model-specific, which gives us a quantitative description, and the second one is called model-agnostic, which provides us with a qualitative description.

networks as a function of the number of neurons in the hidden layer. It seems that the variability of the neural network's output depends strongly on the temperature variable. However, the sensitivity of the other parameters remains relatively stable.

We can estimate the adequate number of neurons for the lowest sensitivity. About fifty neurons in the hidden layer would correspond to low sensitivity in the output of the neurons; as a result, we choose this number of neurons for the neural network architecture with one layer.

4.1.2. Impact of Hyper-parameters: variation of the number of neurons and layers

For this simulation, we change the number of hidden layers from 2 to 7 and the number of neurons from 2 to 10 for each layer. The same number of neurons for all layers is considered. For example, if we have two layers, the number of neurons changes from 2 to 10 for all two layers. We choose

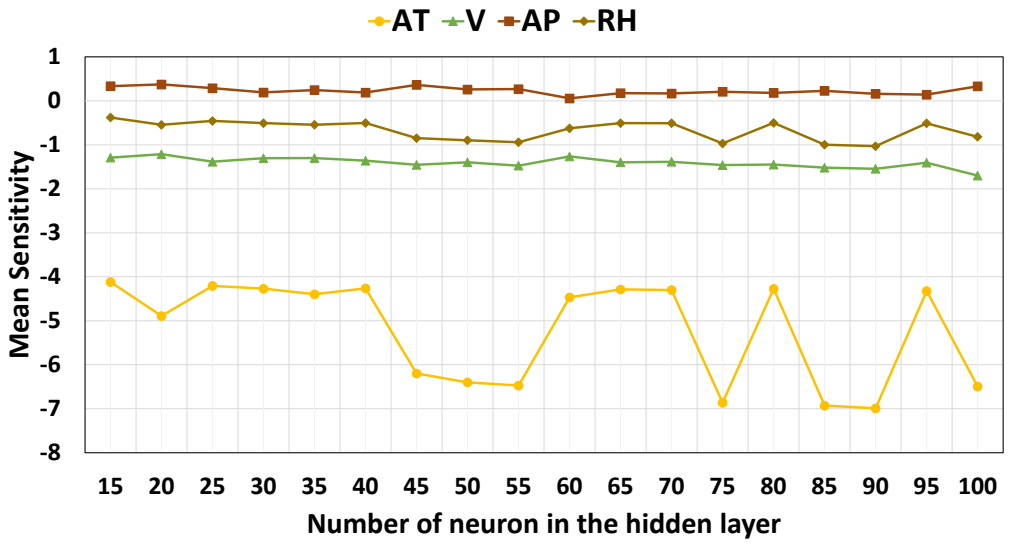


Figure 6: The Mean sensitivities obtained on the output of the MLP neural network as a function of the number of neurons in the hidden layer.

seven layers maximum because the mean sensitivity values do not change remarkably after 6 or 7 layers, and we choose ten neurons maximum because it would be time-consuming for more neurons.

Figure 7 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 while the number increases. We should remember that the lowest sensitivity represents the sufficient number of neurons and layers.

The foremost observation in figure 7 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, and we do not know its reason. From figure 7, it can be concluded that three hidden layers and six neurons would match low sensitivity in the output. Accordingly, these values for the MLP's hyper-parameters are considered.

4.2. PDP and ICE plots

Figures 8 and 9 show the Partial Dependence Plots (PDP) and Individual Conditional Expectation (ICE) simultaneously for different neural network architecture when the data are normalized. Figure 8 presents PDP for an ANN prediction with one layer and fifty neurons, and figure 9 presents PDP for an MLP with three layers and six neurons.

There are some assumptions for PDP that 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 input variables. The partial dependence function can be com-

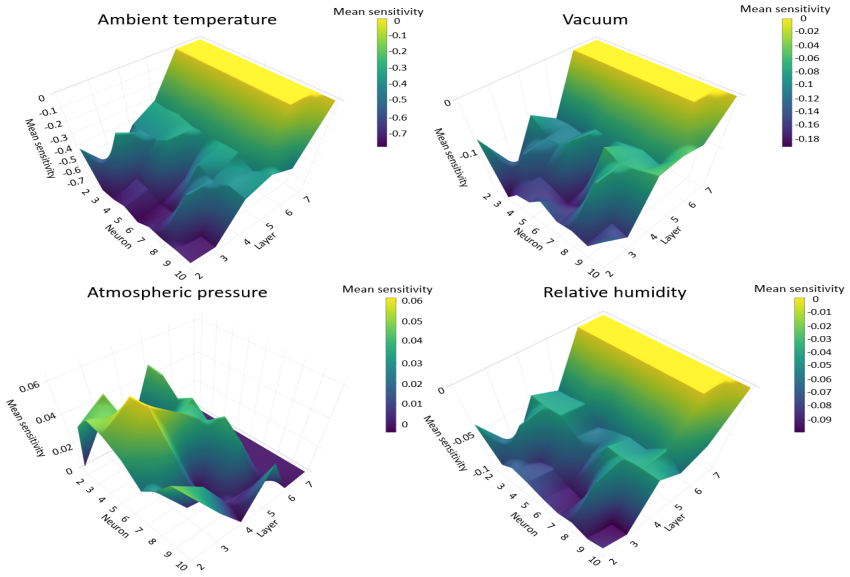


Figure 7: The Mean sensitivities obtained on the output of the MLP neural network as a function of the number of neurons and hidden layers.

puted by averaging predictions with actual feature values of all inputs except the inputs that we want to study their effect, or it calculates the marginal impact of mentioned inputs on the prediction (Molnar, 2019). Individual Conditional Expectation (ICE) plots disaggregate the averages of PDP. ICE plots picture the functional relationship between the predicted response and the input separately for each sample. We can conclude that a PDP averages the individual lines of an ICE plot. The PDP shows the marginal effect one or two features have on the predicted PE of a neural network (MLP). It indicates whether the relationship between the PE and input variables (AT, V, AP, and RH) is nonlinear, monotonic, or more complex.

Figure 8 illustrates that the ambient temperature 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, we have a complex variation. The curve shape of AT remind us the inverse sigmoidal function ($PE = \frac{\alpha}{1+\beta\gamma^{-AT}}$ for $0 < \gamma < 1$). The atmospheric pressure plot is divided into two parts, the first part is almost linear, and in the second part, we get more complex values. Relative humidity and exhaust steam pressure plots are similar and partly linear.

In general, we have smoother results in figure 8 than figure 9 for ICE plots. For example, the sinusoidal turbulence on top of the relative humidity plots can be seen. However, we have approximately the same trend for PDPs. Consequently, PDP is MLP architecture-independent, i.e., changing the hyper-parameters (neither hidden layer nor neurons) impacts the prediction response behavior.

Figures 10 and 11 show the sensitivity of the neural network output based on the variability of two variables for both neural networks with the same architecture as before. We make a uniform color bar for these plots to make it easy to compare them; you can find the plots with the original color bar in the appendix. They are more useful in comparing the effect between every two variables. These figures illustrate which areas PE is more or less high and homogeneous. They also show us that the output does not vary linearly with the simultaneous variability of two variables and how we can reach high PE values.

In subplots of figure 10 with the variability of ambient temperature 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. We could reach the maximum values of PE when the temperature is

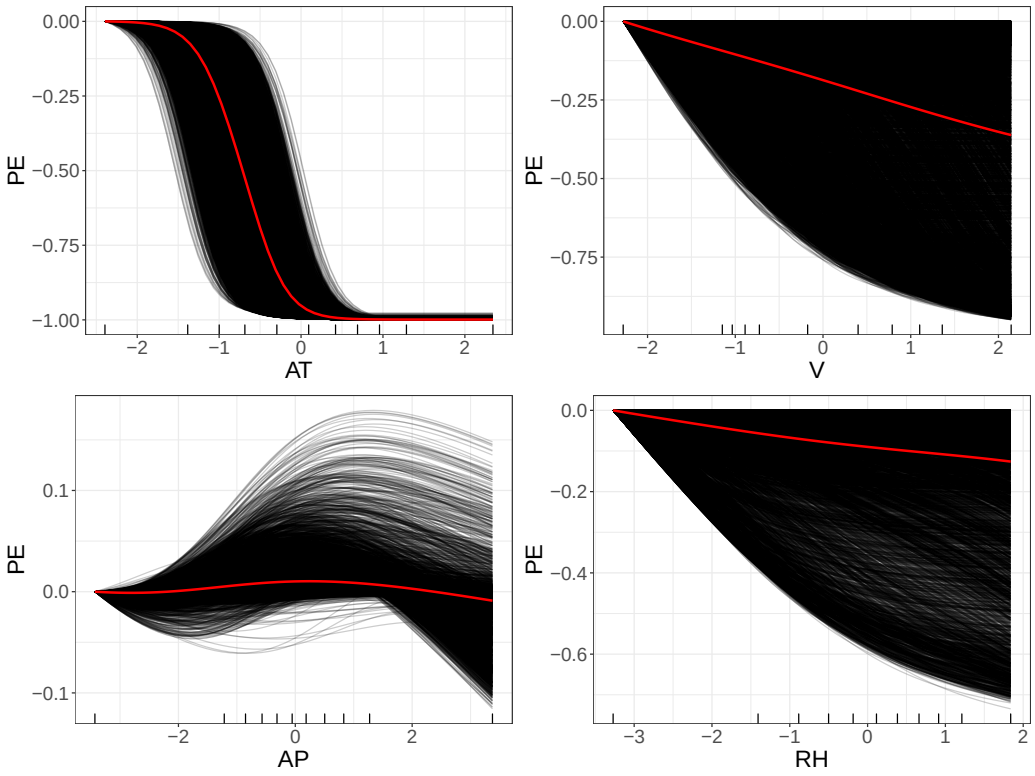


Figure 8: Partial dependence plots (red) and Individual Conditional Expectation plots (black) of a neural network for PE predictions with one layer and fifty neurons. The PDP and ICE are computed after the MLP learning for PE predictions. All variables are standardized during the learning step and kept dimensionless in the PDP computations step.

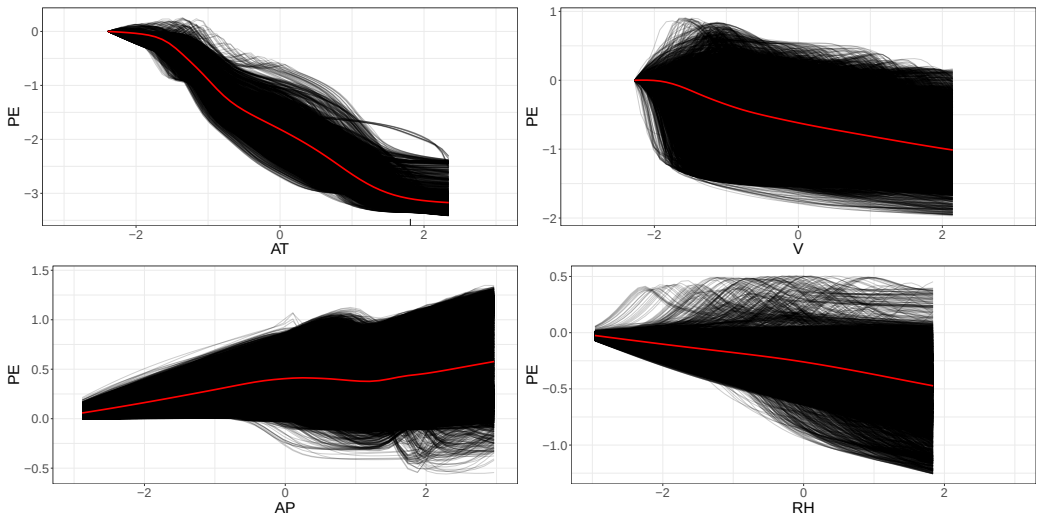


Figure 9: Partial dependence plots (red) and Individual Conditional Expectation plots (black) of an MLP neural network for PE predictions with three layers and six neurons. The PDP and ICE are computed after the MLP learning for PE predictions. All variables are standardized during the learning step and kept dimensionless in the PDP computations step.

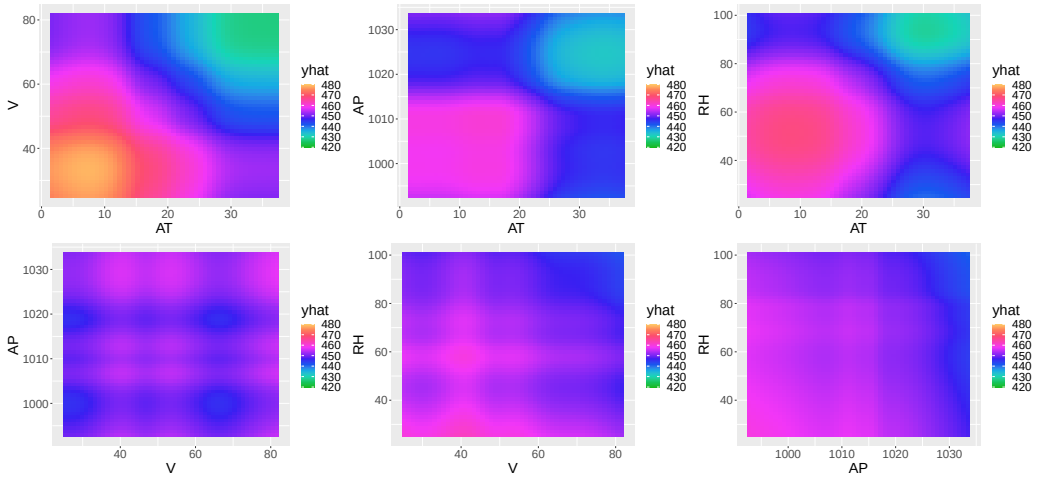


Figure 10: The 2D PDP of variables combination of ANN predictions with 50 neurons (one layer) for the CCPP data values. The gradient legend shows the sensitivity of the neural network output $\widehat{PE}$ (yhat) to the variability of two variables with a uniform color bar.

lower than $15°C$. Moreover, we get approximately the same results for the variability of AT and other input variables (when the variables are low, we get higher PE). We could conclude 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; it is better to see figure 15. It shows the dependence of PE on joint values of AP and V. The vacuum does not affect much, so the influential input variable between them is AP. It illustrates the dependence of PE on joint values of AP and V. We can see an interaction between the two features: for V greater than 35 cm Hg, PE is nearly independent of V.

In the subplot with the vacuum and relative humidity variability, PE

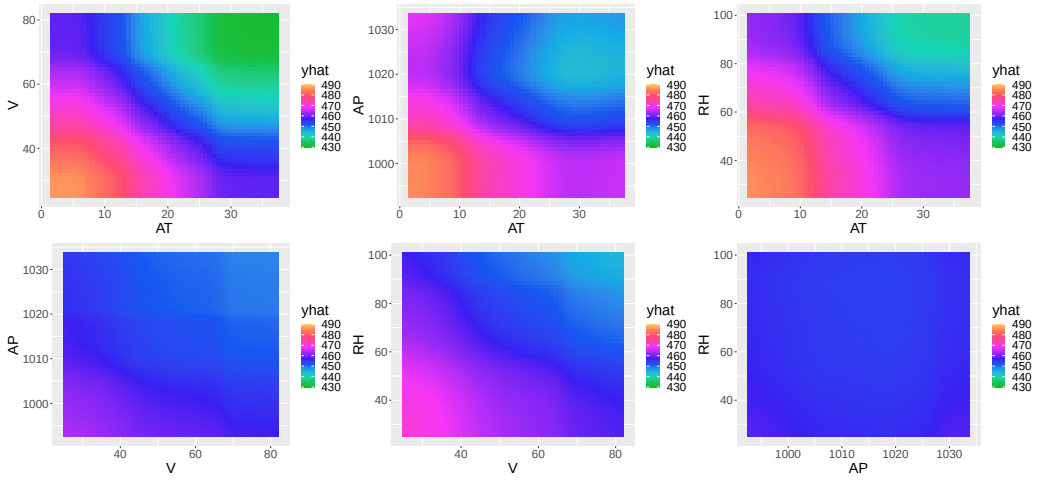


Figure 11: The 2D PDP of variables combination of MLP neural network predictions with 3 layers and 6 neurons for the CCPP data values. The gradient legend shows the sensitivity of the neural network output $\widehat{PE}$ (yhat) to the variability of two variables with a uniform color bar.

could obtain its maximum values when the vacuum is between 35 and 45 cm Hg and 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 lower PE when AP is more than 1025 mbar. PE is nearly independent of RH.

In subplots of figure 11 with the variability of ambient temperature and other input variables (V, AP, and RH), we have almost the same result as figure 10. The relation between PE and the variability of two inputs is almost the same, and we get lower PE when we increase the inputs. We could reach the maximum values of PE 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.

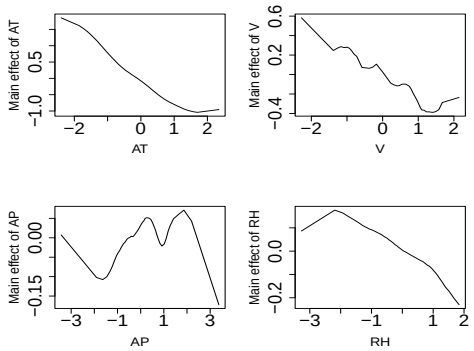
In the subplot with the variability of vacuum and atmospheric pressure, the lowest part is when we have the highest value of input variables, and the highest part is when we have the lowest value of input variables. Variability of vacuum and relative humidity could affect the output almost linearly. 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. We can grasp that PE could obtain its maximum values in the middle of atmospheric pressure and relative humidity plot for AP values between 1005 and 1015 mbar and RH values between 60% and 80% from the figure 16 that is without a uniform color bar.

4.3. ALE plot

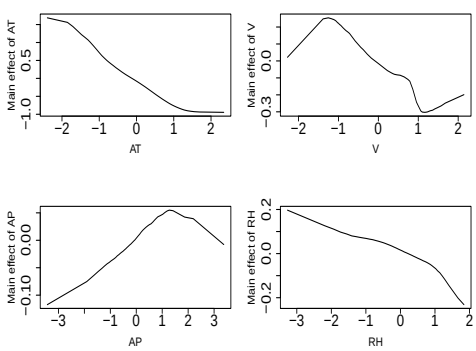
Figure 12 shows the ALE main-effect plot for different neural network architectures. They reveal the main effect of input variables. We could have smoother plots in figure 12b; however, they display approximately the same result. For example, the AT main effect has inverse sigmoidal behavior in both figures. Increasing the AT makes PE decrease. The RH main effect behaves quadratically.

Figures 13 and 14 are the ALE second-order effect plot without the main effect of each input variable. They reveal the interaction between input variables for a neural network with fifty neurons and MLP neural network with three layers and six neurons, respectively. The numbers on the contours show the function values. The darker the chart color, the higher the function value.

Figure 13 reveals notable 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



(a) ALE main-effect plots for neural network with 50 neurons.



(b) ALE main-effect plots for MLP neural network with 3 layers and 6 neurons

Figure 12: ALE main-effect plots for neural networks with different architecture for scaled data

large as the range for the main effect of AT in figure 12a (for scaled data). Figure 13 shows almost moderate interaction in subplots AT-AP and V-AP. It demonstrates negligible interaction between other input variables.

In figure 14, we have lower interaction in general, although the critical and sensitive points remain the same. For example, in both figures 13 and 14, the crucial point in subplot AT-V is when AT is around -1.5 and V is around +1 for normalized data.

We can conclude that AT and V have the most interaction, and AP and RH have the most negligible interaction based on both figures 13 and 14.

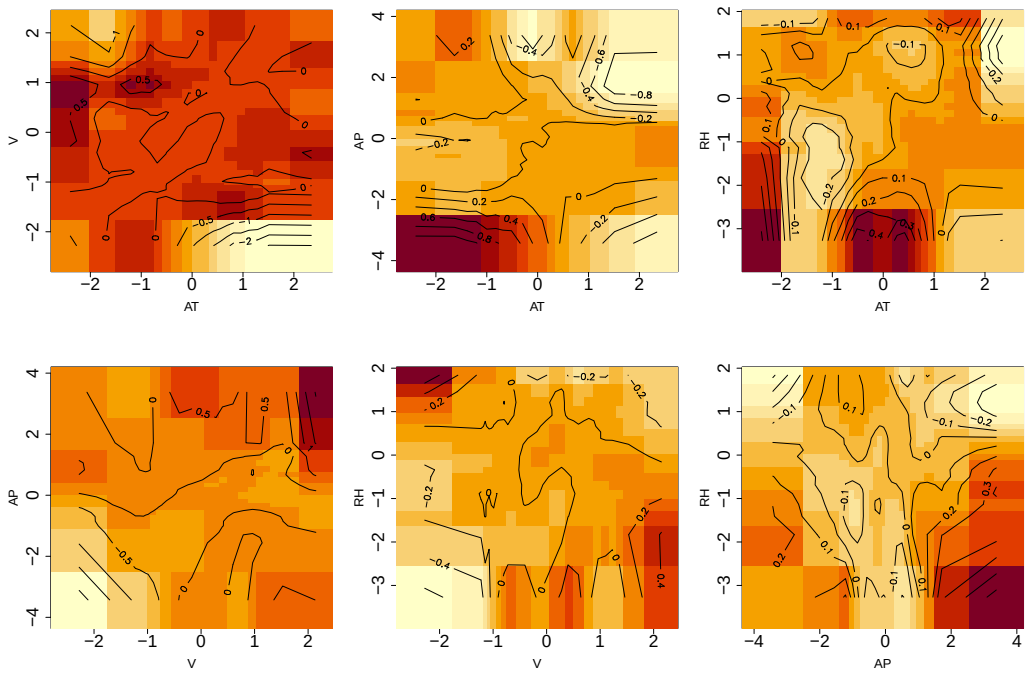


Figure 13: ALE second-order effect plots for neural network with 50 neurons and scaled data. The numbers on the contours represents the function values. The darker the chart color, the higher the function value. All variables are scaled before MLP learning step.

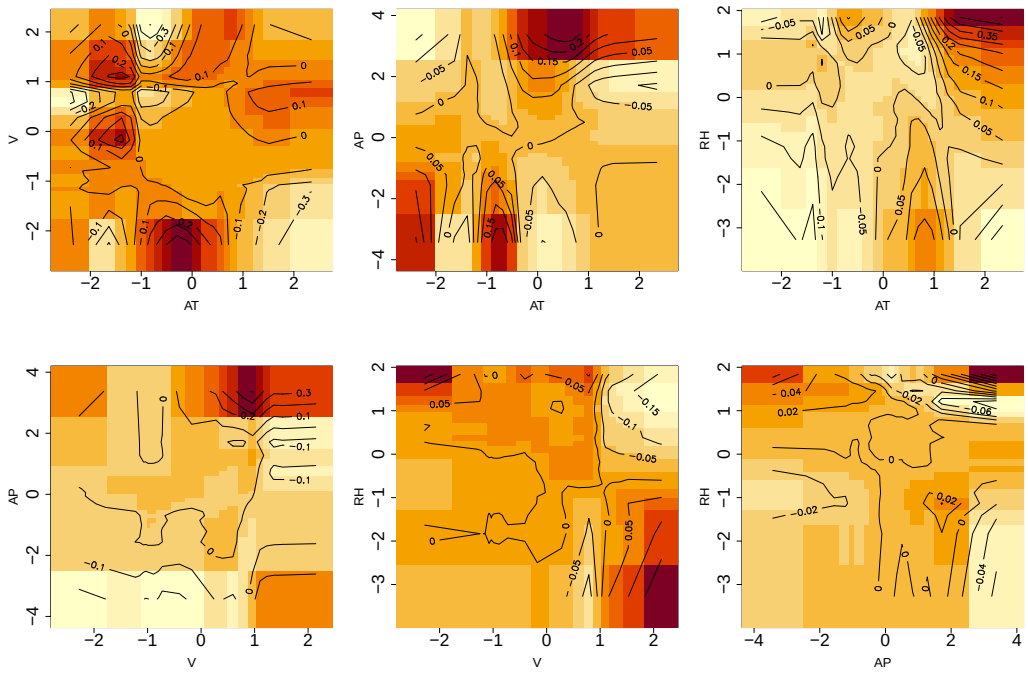


Figure 14: ALE second-order effect plots for MLP neural network with 3 layers and 6 neurons. The numbers on the contours represents the function values. The darker the chart color, the higher the function value. All variables are scaled before MLP learning step.

5. Conclusion

The present study was designed to test the interpretability tools of an MLP prediction model. Two basic approaches have been tested, sensitivity analysis through derivative methods and model-agnostic methods. All of these techniques were applied on full load combined cycle power plant. The main motivation for this study is to analyze the elements of the global context of a real application in a machine learning model. This is a fundamental step towards a hybridization of equation-based models and data driven approaches, which would allow reconciling the prediction accuracy and the interpretability level.

This study was performed on the same data set as Tüfekci's paper (Tüfekci, 2014). while that work has compared different regression methods to predict an electrical power output. This paper focuses on the interpretability point of view of ANN models with different architectures. To this end, first, a clarification on the concept of interpretability of supervised machine learning predictions was provided. Then we perform a sensitivity analysis on the neural network to specify the sufficient number of neurons and layers in our model.

After optimizing hyper-parameters of the MLP, we use model-agnostic methods such as partial dependence plots, individual conditional expectations, and accumulated local effects for visualization and description aspects. The obtained curves would the exhibit the interaction shape between two of the input variables and the output variable, and reveal the most important input variables.

We can conclude through the obtained plots that the most important

parameter in the CCPP is AT. It could affect PE the most, and the combination of AT with other input parameters are more complicated than the other without AT. AT's main or first-order effect on PE has inverse sigmoidal behavior, which means increasing AT decreases PE for the understudy data range. Other input variables could affect PE but for the small range.

The significant advantage of model-agnostic methods is their flexibility, which can be applied to any supervised ML model (regression and classification). The functional relationship provided by these tools is an important model diagnostic technique.

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. The requirement of artificial data instances can impress the estimation of the feature effect. ALE plots are faster to compute than PDPs, but the equivalent of ICE curves presented for the PD plots do not include 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), Long short-term memory (LSTM) used in the field of deep learning for time series. Moreover, we could apply and compare other model-agnostic methods, 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.

6. Appendix

We use the plots with a uniform color bar in subsection 4.2 to make it easy to compare different plots; here you can find the plots with the original color bar.

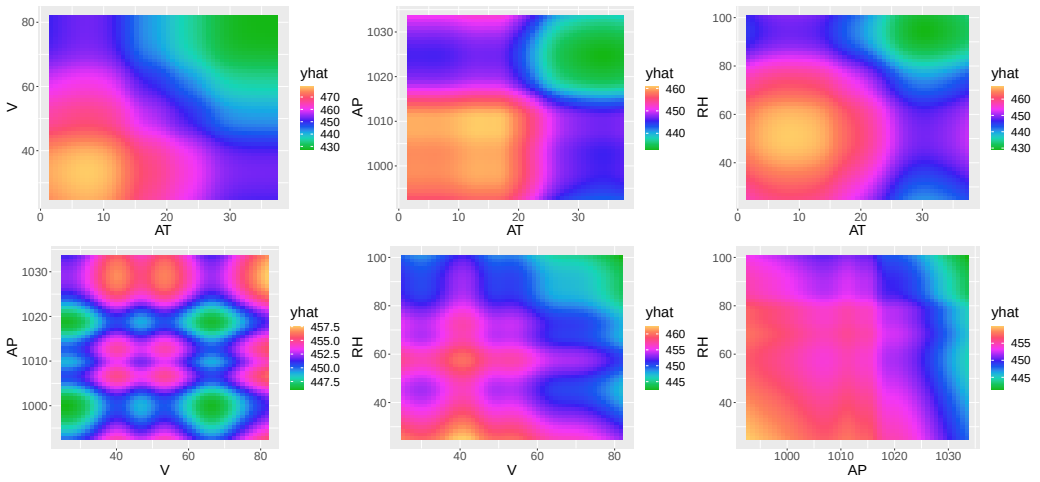


Figure 15: The 2D PDP of variables combination of ANN predictions with 50 neurons (one layer) for the CCPP data values. The gradient legend shows the sensitivity of the neural network output $\widehat{PE}$ (yhat) to the variability of two variables with the original color bar.

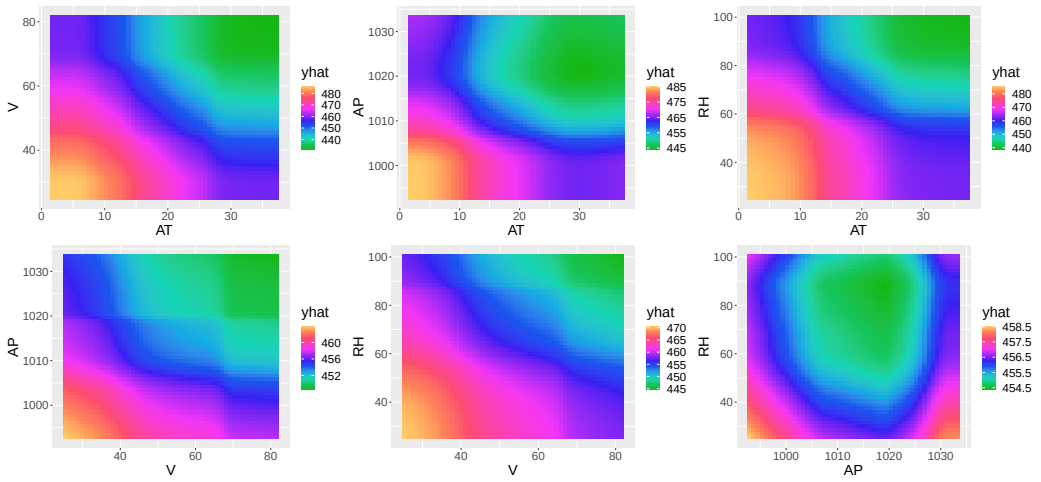


Figure 16: The 2D PDP of variables combination of MLP neural network predictions with 3 layers and 6 neurons for the CCPP data values. The gradient legend shows the sensitivity of the neural network output $\widehat{PE}$ (yhat) to the variability of two variables with the original color bar.

References

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.

Balochian, S., Baloochian, H., 2019. Social mimic optimization algorithm and engineering applications. Expert Systems with Applications 134, 178–191.

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

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

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.

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

Garson, G.D., 1991. Interpreting neural-network connection weights. AI Expert 6, 46–51.

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.

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

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.

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.

Mafarja, M., Aljarah, I., Faris, H., Hammouri, A.I., Ala'M, A.Z., Mirjalili, S., 2019. Binary grasshopper optimisation algorithm approaches for feature selection problems. Expert Systems with Applications 117, 267–286.

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.

Niu, L., Liu, X., 2008. Multivariable generalized predictive scheme for gas turbine control in combined cycle power plant, in: 2008 IEEE Conference on Cybernetics and Intelligent Systems, IEEE. pp. 791–796.

Olden, J.D., Jackson, D.A., 2002. Illuminating the "black box": a randomization approach for understanding variable contributions in artificial neural networks. Ecological modelling 154, 135–150.

Özesmi, S.L., Özesmi, U., 1999. An artificial neural network approach to spatial habitat modelling with interspecific interaction. Ecological modelling 116, 15–31.

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 .

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

Romeo, L., Loncarski, J., Paolanti, M., Bocchini, G., Mancini, A., Frontoni, E., 2020. Machine learning-based design support system for the prediction of heterogeneous machine parameters in industry 4.0. Expert Systems with Applications 140, 112869.

Rumelhart, D.E., Hinton, G.E., Williams, R.J., 1985. Learning internal representations by error propagation. Technical Report. California Univ San Diego La Jolla Inst for Cognitive Science.

Rumelhart, D.E., Hinton, G.E., Williams, R.J., 1986. Learning representations by back-propagating errors. nature 323, 533–536.

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.

Sharpe, C., Wiest, T., Wang, P., Seepersad, C.C., 2019. A comparative evaluation of supervised machine learning classification techniques for engineering design applications. Journal of Mechanical Design 141, 121404.

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

Tso, W.W., Burnak, B., Pistikopoulos, E.N., 2020. Hy-pop: Hyperparameter optimization of machine learning models through parametric programming. Computers & Chemical Engineering 139, 106902.

Tubishat, M., Idris, N., Shuib, L., Abushariah, M.A., Mirjalili, S., 2020. Improved salp swarm algorithm based on opposition based learning and novel local search algorithm for feature selection. Expert Systems with Applications 145, 113122.

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.

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.

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