Computer Methods and Programs in Biomedicine 242 (2023) 107843

Contents lists available at ScienceDirect

Computer Methods and Programs in Biomedicine

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

Interactive polar diagrams for model comparison

Aleksandar Anžel [a,*], Dominik Heider [a], Georges Hattab [b,c]

[a] Department of Mathematics & Computer Science, University of Marburg, Hans-Meerwein-Straße 6, Marburg, D-35032, Hesse, Germany
[b] Center for Artificial Intelligence in Public Health Research (ZKI-PH), Robert Koch-Institute, Nordufer 20, Berlin, 13353, Berlin, Germany
[c] Department of Mathematics and Computer Science, Freie Universität Berlin, Arnimallee 14, Berlin, 14195, Berlin, Germany

ARTICLE INFO

Keywords:
Bioinformatics
Machine-learning
Visualization
Evaluation
Climate
Comparison
Ai
Data-visualization
Information-visualization
Predictive-analysis
Model-comparison
Climate-model-visualization
Ml-model-evaluation
Taylor-diagram
Mutual-information-diagram
Entropy
Mutual-information
Variation-of-information
Correlation
Medical-data

ABSTRACT

Objective: Evaluating the performance of multiple complex models, such as those found in biology, medicine, climatology, and machine learning, using conventional approaches is often challenging when using various evaluation metrics simultaneously. The traditional approach, which relies on presenting multi-model evaluation scores in the table, presents an obstacle when determining the similarities between the models and the order of performance.

Methods: By combining statistics, information theory, and data visualization, juxtaposed Taylor and Mutual Information Diagrams permit users to track and summarize the performance of one model or a collection of different models. To uncover linear and nonlinear relationships between models, users may visualize one or both charts.

Results: Our library presents the first publicly available implementation of the Mutual Information Diagram and its new interactive capabilities, as well as the first publicly available implementation of an interactive Taylor Diagram. Extensions have been implemented so that both diagrams can display temporality, multimodality, and multivariate data sets, and feature one scalar model property such as uncertainty. Our library, named *polar-diagrams*, supports both continuous and categorical attributes.

Conclusion: The library can be used to quickly and easily assess the performances of complex models, such as those found in machine learning, climate, or biomedical domains.

1. Introduction

One of the last steps of any simulation or predictive analytics experiment is to determine the effectiveness of the used models and find the one that best explains the observed phenomenon. The visual comparison of one or two complex models containing multiple variables (dimensions) becomes impractical and often impossible when the number of dimensions exceeds three [1,2]. However, those models are standard in meteorological, medical, biological, and other similar domains. When considering more than two complex models, determining which model is the best becomes unachievable. Although we provide an in-depth examination of model interpretation in Section 4, it is imperative to note that any n-dimensional numerical vector is considered a model — hence the definition of a model is not restricted to a specific context within this paper.

To address the task of determining the best model, a quantification of the models' quality is required. The related work relies on the observed data by calculating summary statistics or other types of measures (attributes). To present such attributes and statistics, visualization is needed. By visualizing and representing each model in 2-dimensional (2-D) or 3-dimensional (3-D) plots, reducing the dimensionality of the data is an intrinsic part of the process. Commonly used visualization plot solutions typically rely on scatter plots and heatmaps. Both plot types can be seen in Fig. 1. However, these two plot types only allow pairwise comparisons [3]. A possible solution is a scatterplot matrix, which is an arrangement of scatter plots organized in a grid or matrix to visualize bivariate relationships among variable combinations. The matrix includes multiple scatter plots, each of which illustrates the relationship between a pair of variables, enabling the examination of several relationships within a single chart. While scatterplot matrix charts

[*] Corresponding author.
E-mail address: aleksandar.anzel@uni-marburg.de (A. Anžel).

https://doi.org/10.1016/j.cmpb.2023.107843
Received 7 April 2023; Received in revised form 16 September 2023; Accepted 2 October 2023

Available online 6 October 2023

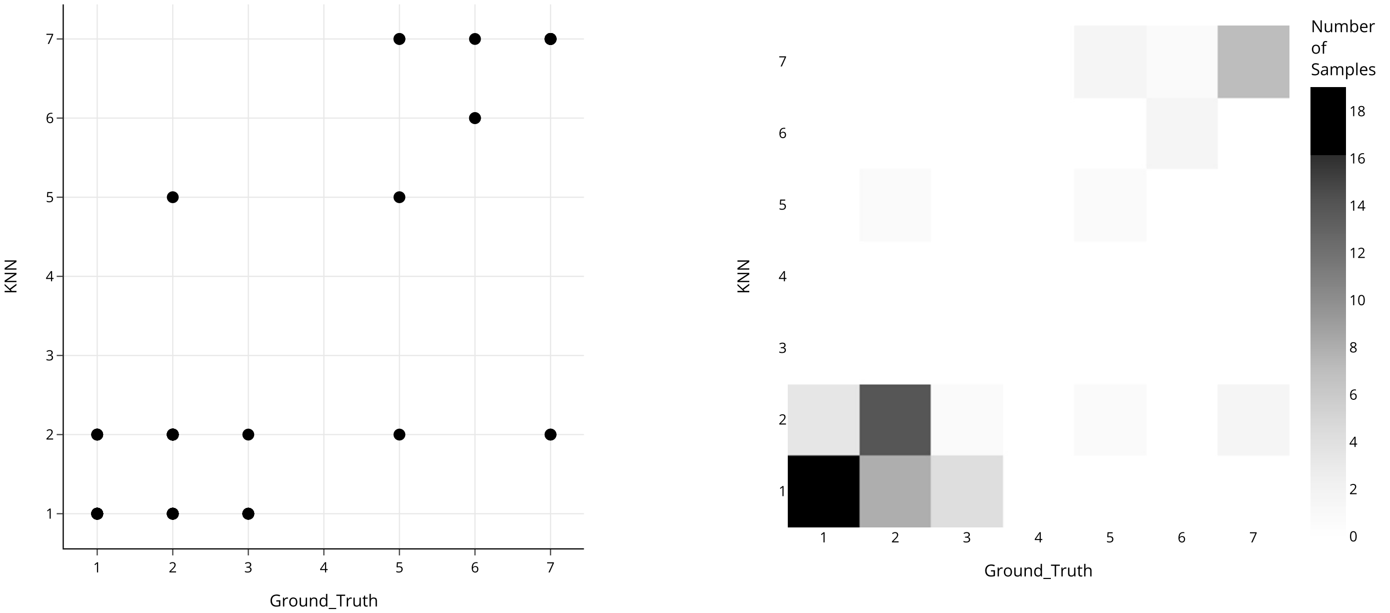


Fig. 1. Traditional visualization approaches for pairwise comparison. Scatter plot (left) and heatmap chart (right) visualizing the relationship and confusion matrix between *Ground_Truth* and *KNN* model trained and evaluated on the *Glass* [10] data set. On one hand, by using the scatter plot, we can only find out whether the *KNN* model was good or bad. Since no diagonal structure is formed from points, we conclude the model was bad for this task. For example, if we take one sample where the *Ground_Truth* value is 7 (lower right point), we can see the predicted value by the model to be, in some cases 7, and, in others, 2. On the other hand, the heatmap chart enables us to assert the quality of model predictions as well by showing us the number of times the model made a mistake. It is important to mention that, in order to maintain consistency, we also consider the *Ground_Truth* as a "model". This is because the *Ground_Truth* serves as a standard reference model against which all other models are evaluated. The same applies to Figs. 2 and 3.

are useful for understanding bivariate relationships between multiple variables, they do have limitations. First, they can get cluttered with a large number of variables, making it difficult to distinguish individual plots and trends. Second, outliers can skew the distribution and make it challenging to visualize correlations accurately. Third, it can be difficult to identify cause-and-effect relationships and additional analysis may be required to understand how variables relate to each other [4,5]. As indicated by Fig. 2, the first drawback becomes evident even with three variables. Alternatively, the parallel coordinates plot is a solution to multivariate analysis where attributes are represented as parallel vertical axes scaled within their data range, as demonstrated in Fig. 3. However, visual cluttering in this plot type can pose a significant problem for the exploration of relationships between the neighboring axes. Ordering of the axes and visual clutter are limiting factors. This problem has been extensively explored in the past [1,6–9]. For these plot types, the task of visually comparing large corpora of models becomes intractable. This defines a bottleneck for high-dimensional models' comparison.

Over the last years, many model-comparison visualization solutions have been developed, yet a majority of them are domain-specific and cannot be translated for use in other fields. One example of a domain-specific visualization tool in the field of Machine Learning (ML) by Zhou *et al.* [11] uses a radial-structure approach, which allows for the comparison of ML models with different numbers of features. While this approach is indeed a viable solution for ML models, it is limited by its domain-specificity and the lack of open-source code, preventing its use in other domains. Another example of a domain-specific visualization tool in the field of ML by Talbot *et al.* [12] is *EnsembleMatrix*, an interactive visualization tool that provides a graphical view of confusion matrices to assess ML classifier models. Unfortunately, this visualization solution is heavily domain-specific and cannot be translated for use in other fields. The same can be said for *Squares* by Ren *et al.* [13], a performance visualization method for multiclass classification problems. The method utilizes standard classification metrics and instance-level distribution information to leverage better explainability of used classifiers. In the field of climatology, besides the Taylor and Mutual Information diagrams, Yatkin *et al.* [14] developed a modified Target Diagram to evaluate the performance of low-cost sensors for air quality monitoring. However, this visualization approach is complicated and requires a level of expertise and extensive training to interpret, making it unsuitable for use in ML or biomedical domains. The limitations of domain-specific visualization methods highlight the need for more generalized techniques that can be applied across different fields.

Previously mentioned limitations were addressed with the publication of the Taylor Diagram [15], which was initially developed for the assessment of climate models. This polar chart efficiently summarizes the model effectiveness according to the observation using three statistical measures: standard deviation, Pearson's correlation coefficient, and centered root mean squared (CRMS) difference or error. However, even though it uses both first- and second-order statistics, the Taylor Diagram cannot capture nonlinear dependencies between models (see Section 2.1.2). Furthermore, if two models are relatively similar but one or both produce outliers, the correlation between them may be low and, in turn, wrongly depict more significant dissimilarity between them.

The Mutual Information Diagram (MID) [16] addresses both issues using information theory. Instead of relying on statistical measures to summarize the models' performance as in the Taylor Diagram, the MID uses entropy, scaled mutual information (SMI), and variation of information (VI). Alternatively, a variant of the diagram incorporates square root of entropy, normalized mutual information (NMI), and the square root of variation of information (RVI). Contrary to the Taylor Diagram, the MID can expose nonlinear dependencies (more in Section 2.1.2), works with both numerical and categorical data, and is far less sensitive to noise (outliers).

Unfortunately, the MID alone does not provide a solution to the original problem because it cannot distinguish between negatively and positively correlated models. Therefore, both diagrams are required in order to get a realistic picture of all model relationships (linear and nonlinear). Moreover, to create the MID, entropy and mutual information have to be calculated for each model. The current implementation re-

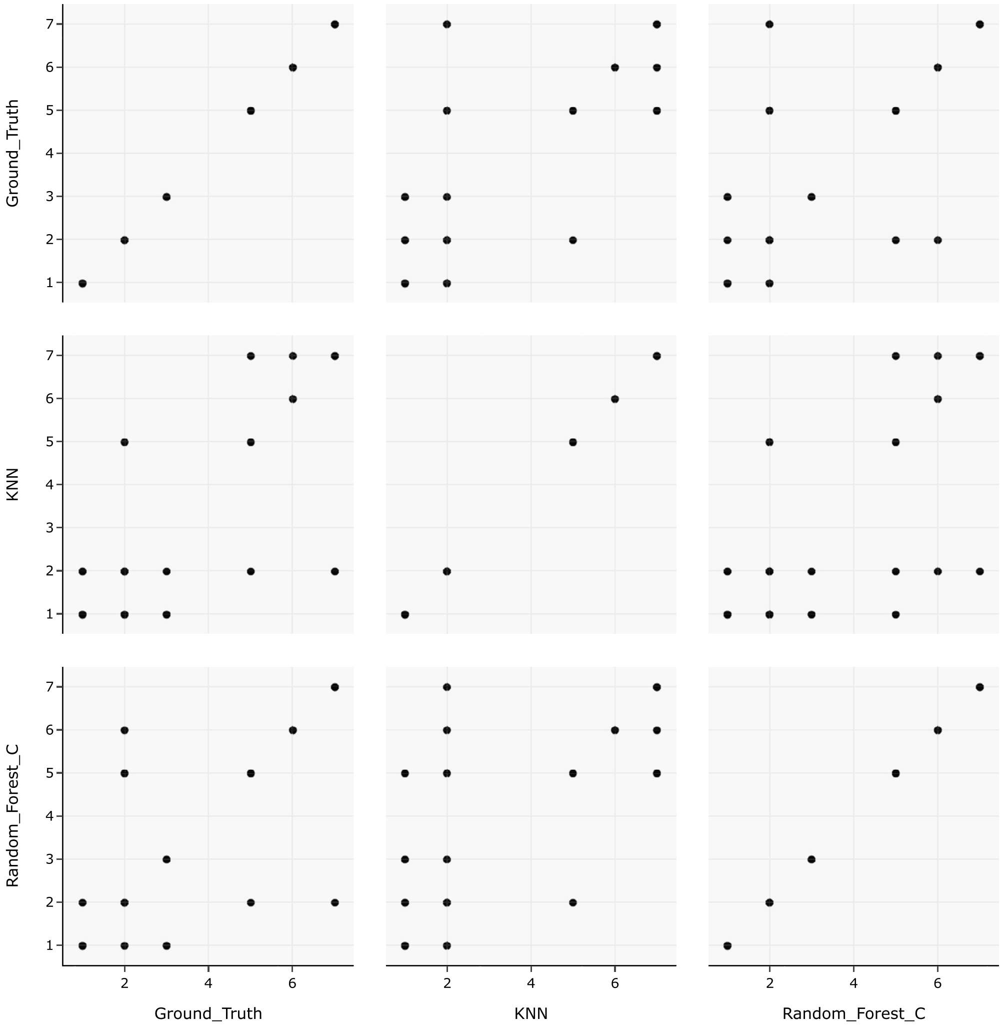


Fig. 2. Scatterplot Matrix. A scatterplot matrix with three variables taken into account — *Ground_Truth*, *KNN*, and *Random_Forest_C*. The *Glass* [10] data set is used to present this plot type.

quires a domain specialist to tune the parameters for each experiment to calculate both the entropy and the mutual information. The choice of these parameters strongly affects resulting diagram [16], thus presenting a significant obstacle to using the MID without any prior knowledge of information theory.

Furthermore, no publicly available open-source library or tool exists to help users create the MID for uncertainty visualization. The authors of the original paper did not provide any source code or data to reproduce the presented results. Consequently, until now, no publicly available implementation of the MID has been available, even though the need for it was shown [17]. On the other hand, the existing libraries for creating the Taylor Diagram (*MATLAB* [18,19], *Python* [20], *R* [21]) do not provide any interactive aspects of the diagram. The resulting visualizations are static images in `PNG` or `JPEG` format. Moreover, these libraries do not support the raw input — the user has to provide pre-calculated standard deviations of all models and correlations of all models with the reference models, severely limiting their adoption.

Even though certain Taylor Diagram libraries and tools allow the visualization of multiple model versions, those implementations rely on adding arrows as visual marks that encode the movements of the models' performances. However, when many models have to be visualized, the diagram quickly becomes overcrowded with visual elements, hence the decreased readability and lower transfer of information from the visualization to the user. Moreover, a set of limiting factors has severely impeded the adoption of polar diagrams until now, including the Taylor Diagram and the Mutual Information Diagram. From static charts to non-scalable graphical formats, to requiring a large set of pre-calculated summary statistics, or even requiring expertise, the adoption and deployment of polar diagrams in the analysis pipeline has suffered greatly.

We show that our library, named *polar-diagrams*, solves completely or partially all of the aforementioned issues. Furthermore, it extends the functionality of both diagrams by allowing users to also visually encode one scalar property of each model or two model versions simultaneously. The resulting diagrams convey more information without overloading the visual space. Moreover, they allow a more granular control by employing multiple interactive techniques such as single- and multi-selection, filter, zoom, and hover. In addition, the back end of *polar-diagrams* employs state-of-the-art methods for calculating mutual information and entropy. As a result, the diagrams become interactive charts that provide accurate information, enable interactivity, and support both discrete and continuous variables.

For the sake of clarity and disambiguation, we adopt the conventional names of the Taylor and the Mutual Information Diagrams for the implemented polar charts, respectively.

2. Methods

This section will cover the mathematical aspects of both diagrams, as well as the technological aspect used to design, implement, and present

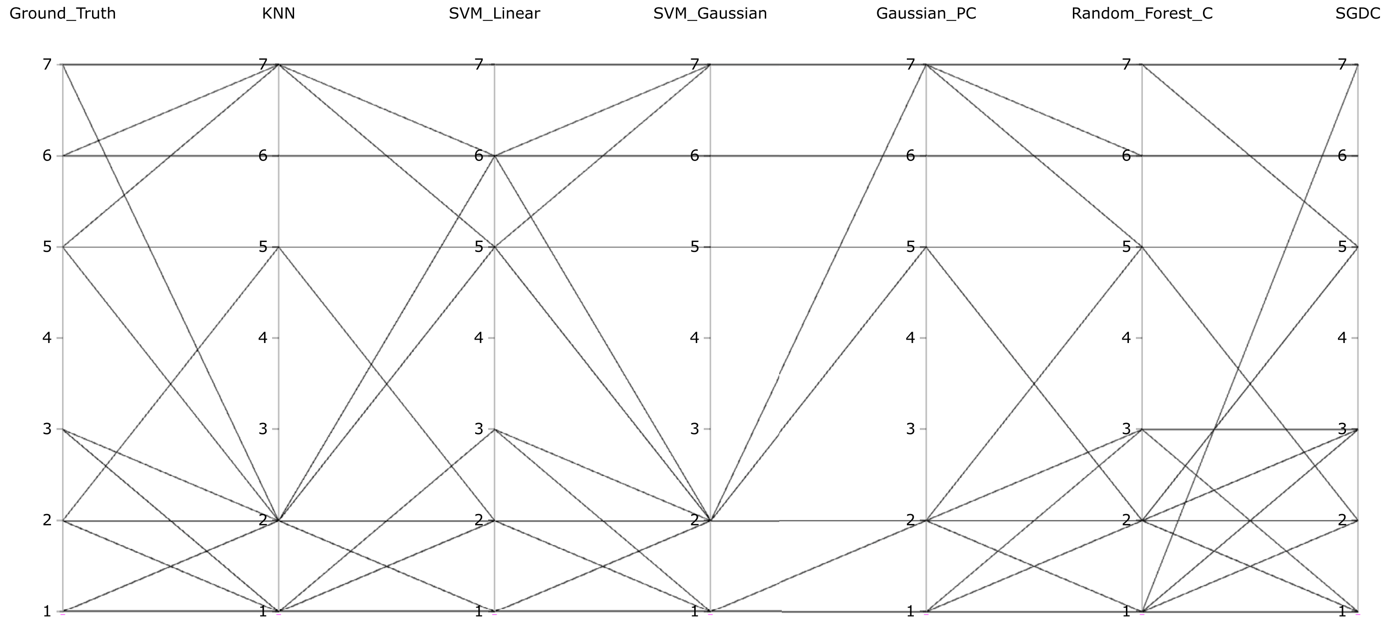


Fig. 3. Parallel Coordinates Plot. A parallel coordinates chart presents a better alternative to the scatter plot matrix and allows a more compact visualization of more than three variables. After performing an evaluation of machine learning models trained on the *Glass* [10] data set, we visualized the performance of six models — *KNN*, *SVM_Linear*, *SVM_Gaussian*, *Gaussian_PC*, *Random_Forest_C*, and *SGDC*, along with the *Ground_Truth*.

the results. The first part is covered by Section 2.1, and the second part by Section 2.2. In this section, the terms *variable* and *model* have the same meaning, and we use them interchangeably.

2.1. Mathematical background

As mentioned earlier, the Taylor diagram relies on first- and second-order statistics to summarize model properties, while the MID relies on the information theory. However, both diagrams exploit the same property of the polar diagrams where the position of each point in a diagram is determined by a distance (radial distance, radial coordinate, or radius) from a reference point (pole) and an angle (polar angle, angular coordinate or azimuth) from a reference direction [22]. Similarities between diagrams essentially end here. We will now present the differences in the construction of both diagrams. In addition, we will denote and explain the mathematical deviations from the original works present in our study.

2.1.1. Taylor diagram

The power of the Taylor Diagram lies in representing each model using three statistical measures: standard deviation, Pearson's correlation coefficient, and centered root mean square error (CRMSE). The "*centered*" aspect of the RMS error definition refers to the subtraction of the respective mean values of both the predicted and observed sets of values before calculating the RMS. This procedure contributes towards rectifying any offset or bias that might have been introduced in the model's predictions, thereby resulting in a more accurate representation of the prediction error [23].

Let us consider a pair of discrete random variables (X, Y), with the cardinality $\mid X \mid = \mid Y \mid = n$, their standard deviations σ_X and σ_Y, and their means μ_X and μ_Y, respectively. We define the mean of a discrete random variable X with the cardinality n as

$$\mu_X = E(X) = \sum_{x\in X} xP(x) = \sum_{i=1}^{n} x_i P(x_i) = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{1}$$

where x represents the values of the random variable X and $P(x)$ represents the corresponding probability. The mean of a discrete random variable X is also known as its expected value and is symbolized as $E(X)$. Furthermore, and for the sake of completeness, we also define the standard deviation of a discrete random variable X with the cardinality n as

$$\begin{aligned}\sigma_X &= \sqrt{\sigma_X^2} = \sqrt{\sum_{x\in X}(x-\mu_X)^2 P(x)} = \sqrt{\sum_{i=1}^{n}(x_i-\mu_X)^2 P(x_i)} \\ &= \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i-\mu_X)^2}\end{aligned} \tag{2}$$

where σ_X^2 is also known as the variance of a discrete random variable X. If we define covariance between X and Y as

$$cov(X,Y) = \frac{1}{n}\sum_{i=1}^{n}(x_i-\mu_X)(y_i-\mu_Y) \tag{3}$$

then Pearson's correlation coefficient is

$$R_{XY} = \frac{cov(X,Y)}{\sigma_X\sigma_Y}. \tag{4}$$

By using the definition of the cosine formula

$$c^2 = a^2 + b^2 - 2ab\cos\theta \tag{5}$$

where a, b, and c are the sides of an arbitrary triangle, and the formula for CRMSE

$$CRMSE(X,Y) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}[(x_i-\mu_X)(y_i-\mu_Y)]^2} \tag{6}$$

we get

$$CRMSE(X,Y)^2 = \sigma_X^2 + \sigma_Y^2 - 2\sigma_X\sigma_Y R_{XY} \tag{7}$$

hence $\theta = \arccos(R_{XY})$. The relation between the CRMSE and the total RMSE can be described by the following expression:

$$\begin{aligned}CRMSE^2 &= RMSE^2 - (\mu_X-\mu_Y)^2 \\ &= \left(\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i-y_i)^2}\right)^2 - (\mu_X-\mu_Y)^2\end{aligned} \tag{8}$$

which also demonstrates how we get Equation (6).

The Taylor Diagram can now be easily constructed using the following procedure:

1. calculate standard deviations for all models,
2. pick one model as a reference model (the variable X in all equations),
3. calculate Pearson's correlation coefficient between the reference model and all other models,
4. calculate the angles using Pearson's correlation coefficient,
5. visualize each model using its standard deviation value as the radius and the calculated angle as the polar angle where the reference direction starts from the pole horizontally to the right, and the polar angle increases to positive angles when traversing the diagram in the counterclockwise direction.

When working with multiple models, it is not uncommon for those models to use different units of measure. That can influence the statistical measures used in the Taylor Diagram. When facing such a situation, CRMSE and standard deviations are normalized ($CRMSE'(X,Y) = CRMSE(X,Y)/\sigma_X$, $\sigma'_Y = \sigma_Y/\sigma_X$), and those "fixed" values are then visualized. As a result, the reference model is now placed on the abscissa with the radius 1.

2.1.2. Mutual information diagram

On the other hand, the Mutual Information Diagram exploits information-theoretic properties of measures such as entropy, mutual information, and variation of information for the construction of the polar diagram. Let us again consider a pair of discrete random variables (X,Y), with the cardinality $\mid X \mid = \mid Y \mid = n$. We define discrete or Shannon entropy as

$$H(X) = -\sum_{i=1}^{n} P(x_i)\log P(x_i) \tag{9}$$

and mutual information (MI) between X and Y as

$$\begin{aligned} I(X;Y) &= \sum_{x\in X}\sum_{y\in Y} P_{(X,Y)}(x,y)\log\frac{P_{(X,Y)}(x,y)}{P_X(x)P_Y(y)} \\ &= H(X) + H(Y) - H(X,Y) \end{aligned} \tag{10}$$

where $H(X,Y)$ is the joint entropy of X and Y defined as

$$H(X,Y) = -\sum_{x\in X}\sum_{y\in Y} P_{(X,Y)}(x,y)\log_2 P_{(X,Y)}(x,y) \tag{11}$$

The term $P_{(X,Y)}(x,y)$ in Equations (10) and (11) denotes the joint probability of values $x \in X$ and $y \in Y$ occurring together, and $P_X(x)$ and $P_Y(y)$ are the marginal probability mass functions of X and Y, respectively. Both equations also demonstrate why MI is a robust measure of dependence as it can identify any connections between random variables that deviate from random chance (*i.e.*, it measures general dependence). When two random variables are independent, the sum of their marginal entropies equals their joint entropy. If the joint entropy is less than the sum of the marginal entropies, it reveals some form of dependency. Unlike correlation, MI is non-parametric, and does not require any specific distributions or mathematical forms of dependence to determine the relationship between random variables. This makes it ideal in detecting both linear and nonlinear correlations [24,25].

The last measure, known as the variation of information (VI), unifies entropy and mutual information and enables us to construct the Mutual Information Diagram. This measure, which is also a metric as shown in [16], is defined as

$$\begin{aligned} VI(X,Y) &= H(X) + H(Y) - 2I(X;Y) \overset{(13)}{=} \\ &= I(X;X) + I(Y;Y) - 2I(X;Y) \end{aligned} \tag{12}$$

where in Equation (12), we used a known property of mutual information where

$$I(X;X) = H(X). \tag{13}$$

If we further notice that Equation (12) can be written as

$$\begin{aligned} \sqrt{VI(X,Y)}^2 = \sqrt{H(X)}^2 + \sqrt{H(Y)}^2 \\ - 2\sqrt{H(X)}\sqrt{H(Y)}\frac{I(X;Y)}{\sqrt{H(X)}\sqrt{H(Y)}} \end{aligned} \tag{14}$$

and apply the cosine formula (Equation (5)) we easily get

$$\theta = \arccos\left(\frac{I(X;Y)}{\sqrt{H(X)}\sqrt{H(Y)}}\right) = \arccos\left(NMI_{XY}\right) \tag{15}$$

where NMI_{XY} denotes the normalized mutual information between X and Y [26]. The authors of the paper [16] named the resulting Mutual Information Diagram, which uses the root entropy value for radius and the NMI for calculating the polar angle of a diagram, as *Normalized Mutual Information Diagram* (NMID).

If we square the left side of Equation (12), we get the following equation

$$\begin{aligned} VI^2(X,Y) &= (H(X) + H(Y) - 2I(X;Y))^2 = \cdots = \\ &= H^2(X) + H^2(Y) - 2H(X)H(Y) * c_{XY} \end{aligned} \tag{16}$$

where

$$c_{XY} = 2I(X;Y)\frac{H(X,Y)}{H(X)H(Y)}. \tag{17}$$

Again, by using the cosine formula, we get $\theta = \arccos c_{XY}$. Since $c_{XY} \in [-1,1]$, the authors of [16] proposed using the unbiased version of mutual information for the diagram creation. The new version is called scaled mutual information and is defined as

$$SMI_{XY} = (c_{XY} + 1)/2 \tag{18}$$

In turn, the resulting diagram is now placed in the range $[0,1] \ni S_{XY}$ and called *Scaled Mutual Information Diagram* (SMID).

Since $NMI_{XY} \in [0,1]$, $SMI_{XY} \in [0,1]$, and $R_{XY} \in [-1,1]$ both positive and negative correlations map to positive mutual information.

Both versions of the Mutual Information Diagram can be constructed similarly to the Taylor Diagram by following the procedure below:

1. calculate

   (SMID) entropies for all models,
   (NMID) square root of entropies for all models,

2. pick one model as a reference model (the variable X in all equations),
3. calculate mutual information between the reference model and all other models,
4. calculate joint entropies between the reference model and all other models using Equation (10),
5. calculate

   (SMID) scaled mutual information using Equation (18)
   (NMID) normalized mutual information using Equation (15)

   between the reference model and all other models,
6. visualize each model using its

   (SMID) entropy value as radius, and the calculated angle from Equation (17)
   (NMID) root entropy value as radius, and the calculated angle from Equation (15)

   as the polar angle where the reference direction starts from the pole horizontally to the right, and the polar angle increases to pos-

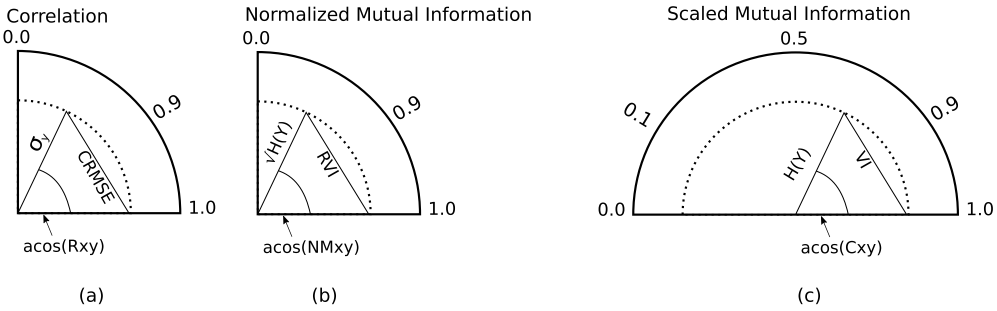


Fig. 4. Taylor Diagram (a), NMID (b), and SMID (c) are presented. As we can see, while the Taylor Diagram and the SMID span the first and the second quadrants, the NMID spans only the first quadrant. The reader should note that the Taylor Diagram presented in this figure is a trimmed version of the full diagram since the negative correlations are not presented. This procedure is usually applied to the SMID as well.

itive angles when traversing the diagram in the counterclockwise direction.

As with the Taylor Diagram, measures are often normalized $(I'(X;Y) = I(X;Y)(H(X)/I(X;X)),\ H'(Y) = H(Y)/H(X))$, and those new values are then visualized. As a result, the reference model is now placed on the abscissa with the radius 1, and the property in Equation (13) is maintained. All three polar diagrams can be seen in Fig. 4.

2.1.3. *Important notes and the deviations from the original work*

In this section, we will cover some aspects of both diagrams we deem important and describe the deviations from the original MID presented in [16].

We started Sections 2.1.1 and 2.1.2 by considering a pair of discrete random variables (X, Y) and then explained the constructions of both diagrams by relying on that initial condition. However, not all data sets will contain only discrete variables. All statistical measures required for creating the Taylor Diagram and presented in Section 2.1.1 are also applicable to continuous random variables. The situation is far more complex for the MID.

One way to calculate entropy for continuous variables is to estimate the underlying probability density function (PDF) of that variable. The authors of the original MID explored and tested multiple different methods for estimating PDF. Their results show that the choice of a method and its parameters significantly influence the resulting MID, even though the locations of the distributions in the MID are more or less preserved. Yet, each of the continuous data set examples they presented in Section *5. RESULTS* uses different methods for estimating PDF. In essence, Example *5.1 Intercomparison Studies* used histograms, and Example *5.2 Analysis of Climate Ensembles* used kernel density estimation with the optimal bandwidth for a bivariate normal distribution and Epanechnikov kernels. The lack of consistency in the methodology for the PDF estimation step in the original paper and the lack of transparency in parameter selection shows that each MID may have been tailored to each data set separately. This presents a great obstacle for any user that is not a domain specialist and wants to use the MID.

To address this problem, we designed *polar-diagrams* by considering the results from the original paper and state-of-the-art methods for calculating continuous entropy and MI.

First, we wanted to give as much flexibility to the user as possible by allowing them to input mixed data sets (*i.e.*, data sets with both continuous and discrete models). To accommodate the possibility of mixed data sets as an input we relied on the results of paper [27]. The authors of the study verified the nearest neighbor method as being far more accurate, less computationally, and less memory expensive than binning-based MI estimators. This is why we decided to use the non-parametric method, known as Kraskov's method [28], to calculate only the MI between variables. Our decision is further corroborated by paper [16], that demonstrates the higher accuracy of this method when estimating MI but larger error when estimating entropy.

Second, our library uses different methods for entropy calculations depending on the data type of the models and the parsed optional arguments. If the model in question is discrete, Equation (9) is used to calculate entropy. If the model is continuous, the entropy is known as *differential* or *continuous* entropy and measures the average information content of a random variable with a continuous probability distribution. Our library selects different methods to calculate entropy based on the given sample size of the (unknown) distribution. If the data set has less than 10 samples, the *Van Es* estimator [29] is used. In case the sample size is between 11 and 1000, the *Ebrahimi* estimator [30] is used. For larger sample sizes, the *Vasicek* estimator [31] is used with the heuristic value for the *window length* parameter proposed in [32]. The selection behavior and implementation details are described and presented in [33]. Even though our library selects the differential entropy estimation method automatically depending on the sample size, the users are able to override this functionality and manually select one of the previously mentioned methods.

Third and last, our proposed methodology gives more accurate results. However, since we are not treating continuous variables as discrete (and *vice versa*) and we are using task-specific methods, one new problem arises. Unlike discrete entropy, differential entropy can be negative. Since MID is only able to show models with positive entropies, this means that models with negative differential entropy will not be present on the resulting MID. We do not consider this a flaw but a limitation of the MID. In that case, the user is encouraged to use the Taylor Diagram to evaluate the results. This motivates and supports the coupling of the polar diagrams presented in this work.

2.2. *Technical background*

We decided to develop *polar-diagrams* using *Python* programming language [34] due to its flexibility and cross-domain popularity. We also relied on multiple well-established libraries for data manipulation, analysis, and visualization. This section will cover all essential libraries on which our library depends and explain the functionalities we use to create the polar diagrams.

The first step in visualizing model results using *polar-diagrams* is to prepare the data set. The user should use *Pandas* [35,36] library to import the raw data into wide-formatted *Pandas* `DataFrame`. The resulting `DataFrame` must have a 1-level index, model names as column names, and model results in rows. Hence, the `DataFrame` has dimensions nxm where n is the number of rows, and m is the number of models (columns). This is the only format of the input data our library accepts. We decided to use *Pandas* because of its ability to parse a plethora of raw data formats (*e.g.,* `XML`, `JSON`, `CSV`, `SQL`, *etc.*), and its wide-spread use across many domains.

Our library also extends the functionalities of both Taylor and MID by enabling users to visualize a scalar property for each model and visualize two different versions of each model on the same diagram simultaneously. However, the user is able to use only one of the extended functionalities at the same time. In case the user wants to visualize one additional scalar property of each model, instead of parsing one `DataFrame` with model results, users should parse a *Python* `list`. The first element of that list should be a `DataFrame` that contains model results, as described in the previous paragraph. The second element should also be a `DataFrame` that has dimensions $1xm$, where the single row contains the scalar value a user wants to visualize. Column names must be the same for both arguments. All scalar values are internally scaled to $[0,1]$ domain in order to prevent the “explosions” of scalar markers on a visual canvas. This means that scalar markers can only be double the size of model markers, thus preventing visual clutter. If the user wants to visualize two different versions of models, a *Python* `list` should be parsed as an argument. The list elements are two nxm-dimensional `DataFrames` representing different versions of m models. As with the previous case, column names must be the same for both arguments (`DataFrames`).

To calculate all statistical measures necessary for the creation of the Taylor Diagram, we used *Pandas*, *NumPy* [37], and *Scikit-learn* [38,39] libraries. To calculate the discrete entropy, we implemented an in-house algorithm according to the original Shannon’s definition presented in [40]. Differential entropy is calculated as described in Section 2.1.3 using the *SciPy* [41] library. The MI is calculated using Kraskov’s method by adopting the implementation provided in the *Scikit-learn* library. Papers [28,27] show the best MI estimation occurs when the number of neighbors for the method is 3. We also use this as a default value, but the user may change it as necessary.

We used *Plotly* [42] library to design and create all diagrams. It also affords all interactive functionalities of both diagrams and the ability to export them in static image formats.

3. Results

Our library presents the first open-source implementation of the interactive Taylor Diagram and the first public implementation of the MID. In addition, it extends the functional aspects of both diagrams by enabling users to visualize one scalar property of each model and two different versions of models on the same diagram. The users can take advantage of the first functionality to visualize any scalar value that is important to the experiment. We used it to encode the training and prediction time of a selection of ML models presented in Section 3.2. The results are presented in Fig. 9. The second functionality can be exploited for visualizing models in two time points or with the changed (hyper-)parameters, thus allowing the user to examine the shift in model performance. We showcased the latter in Fig. 8.

The resulting diagrams can be exported in the following formats: `PNG`, `JPEG`, `WebP`, `PDF`, and `SVG`. Before exporting the diagram in a static image format, the user is able to interact with it and explore it. We now follow the *nested model of visualization* [43] to dissect and present all functionalities of our library’s resulting charts.

First, incorporating interactive elements into our diagrams presents one of the major advantages over traditional Taylor and Mutual Information diagrams, which are static images. Polar coordinate charts often have multiple axes radiating from a central point, which can make it challenging to accurately assess certain properties without proper interactivity. For example, it can be difficult to compare the magnitude of different data points on the chart or to determine the exact coordinates of a particular point. Similarly, interpreting the distances between the axes may be challenging without the ability to zoom in on specific areas of the chart, *i.e.*, to adjust the scaling of the radial axis. Additionally, given the circular nature of polar coordinate charts, it may be difficult to identify trends or patterns in the data without the ability to interactively adjust various display options such as color-coding or labeling. All of these points were tackled either singularly or simultaneously in works by Burch *et al.* [44], Yee *et al.* [45], Qiang *et al.* [46], and Vehlow *et al.* [47]. Overall, interactivity is essential for accurately interpreting and exploring the properties of polar coordinate charts in a way that is intuitive and meaningful to the user.

By relying on previous studies that researched interactivity in polar coordinates, we incorporated multiple interactive idioms to allow users to explore the data and change the charts before they are exported in a static image format. Hovering the mouse over any model in the diagram reveals a tooltip with additional information about the underlying model. The border of each tooltip is colored the same as the model it refers to. This interactive element can be seen in Fig. 7. Users are also allowed to click on the models’ graphical representation in the legend and exclude them from the results. If users double-click on the model in the legend, all models except for the selected one are excluded from the diagram. Besides *Single selection*, *Zoom* is the next and default interactive tool a user can employ to navigate the polar diagrams. This allows users to select specific radial intervals or areas to be visible on the diagram. It is important to note that the *Zoom* tool does not actually zoom into the visualization canvas, rather it rescales the radial axis of the diagrams. The upper-right part of the visualization canvas contains two more tools that allow more granular control on which models to highlight — *Box Select* and *Lasso Select*. The first tool allows the creation of rectangular regions outside which the models will be de-emphasized by decreasing the saturation. The latter provides the same functionality by defining the region using any polygon. They both are elements of the multi-selection interactive aspect. The resulting diagram with some models highlighted could then be easily exported in any of the previously mentioned static image formats, thus better conveying the story of the underlying experiment. All interactive tools are shown in the upper-right corner in Fig. 7.

Second, we used three encoding channels to encode model data. Circles are used as graphical markers that represent models. They represent elements of the shape channel, where each marker has the same size. Hence, when using *polar-diagrams* to visualize model results considering only one version of the models, this channel does not contain any significant information about the models. However, if users invoke the functionality of visualizing two versions of all models at the same time, this channel is used to create a differentiation between model versions. Circles that encode the second-version models have a solid border, while the circles that encode the first-version models are borderless. We also used the shape channel to encode the second extended functionality *polar-diagrams* supports — visualizing a scalar property of each model. When the user wants to visualize the scalar property, the values are encoded using the concentric circle around the model marker (circle) with the same color. The size channel for the concentric circle is used to encode the normalized scalar value. The difference between the models is encoded using the color channel. The reference model is always encoded using the black color, while all other models are encoded using either *Tableau 10* or *Tableau 20* [48] categorical color schemes depending on the number of models. Each color has 60% opacity, thus allowing an easier model distinction when visual markers overlap.

Our library also supports inspecting and exporting intermediary results for diagram creation. Those results are returned as a *Pandas* `DataFrame` object and can be further exported in any tabular format supported by the *Pandas* library.

One of the example arguments presented in [16] for using the MID instead of the Taylor Diagram is Anscombe’s data set [49]. This data set is a set of four data sets, and each of them has the same summary statistics (*i.e.*, mean, standard deviation, and correlation). The authors showed that certain components of Anscombe’s data set fully overlap on the Taylor Diagram while being dispersed on the MID. We also tested this on a “newer” version of Anscombe’s data sets called *The Datasaurus Dozen* data set [50]. Even though this collection of thirteen data sets contains totally different data sets when visualized, they all have the same summary statistics (X/Y mean, X/Y standard deviation, and Pearson’s correlation). Indeed, the MID shows better results for this example as well. Despite the fact that the Taylor Diagram produces results where models are hard to differentiate, by using the interactive

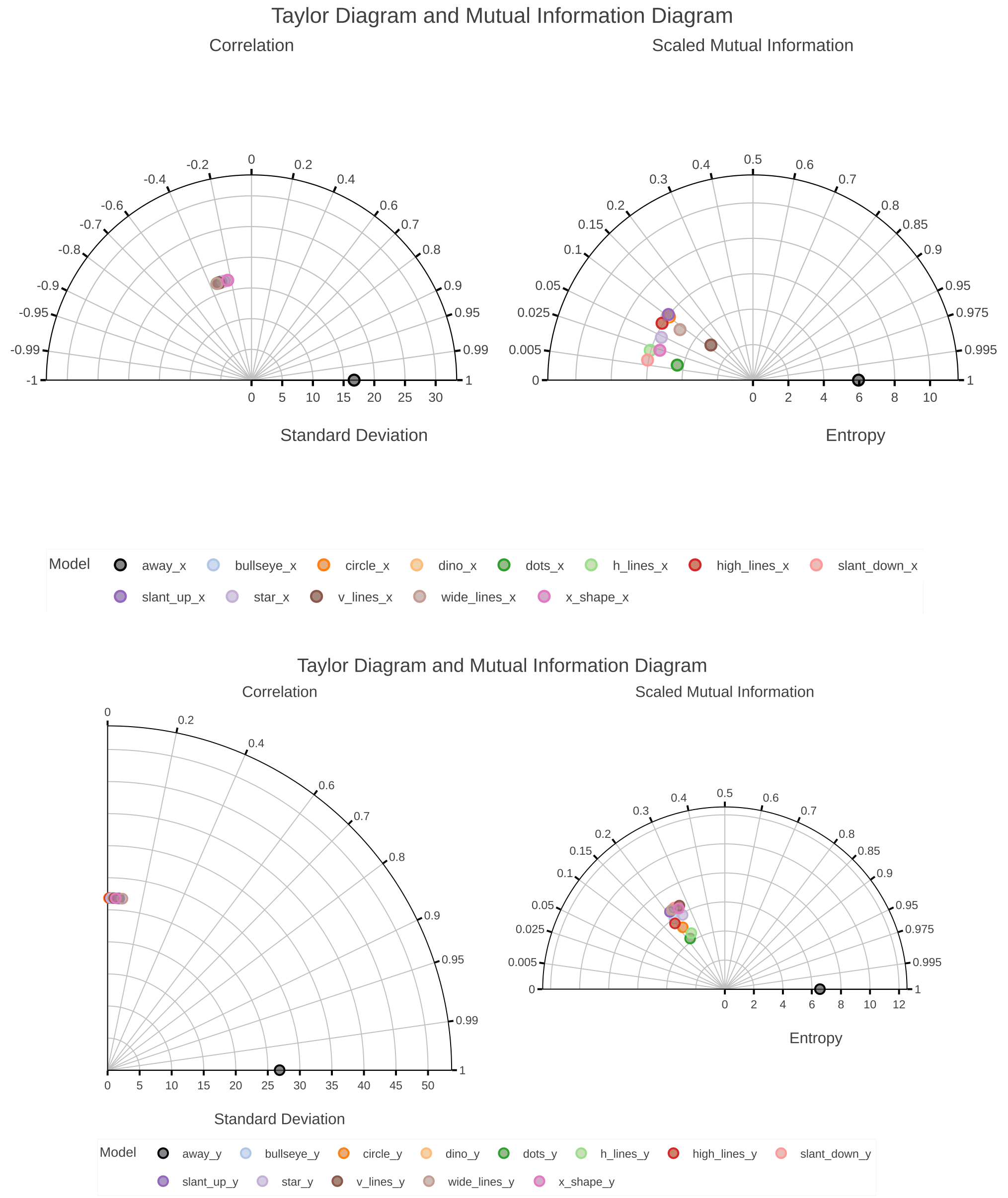


Fig. 5. ***The Datasaurus Dozen* data set**. The top row models present x-axis values, while the bottom row models present y-axis values for all thirteen data sets. The models overlap in all diagrams. However, the Taylor Diagrams (top and bottom left) contain models that fully overlap. The user is notified with a *Python* warning about this phenomenon. The MIDs (top and bottom right) give much better results.

Zoom tool, the user is able to closely inspect if some models are better. We present our results in Fig. 5.

Even though the phenomenon of overlapping model markers occurs less often in the MID, it still can occur under specific circumstances. To solve this problem, we implemented a *Python* `RuntimeWarning`, which notifies the users if any of the diagrams contains overlapping models. Furthermore, the warning reveals the exact models that are overlapping on the diagram, thus providing the user an insight into the data and motivating them to use the interactive functionalities offered by the polar diagrams.

In the following sections, we present our results using the data from three different domains — climate research, machine learning, and biology/medicine.

3.1. Example 1 — climate model evaluation

One of the most important projects of the World Climate Research Programme (WCRP) is the Coupled Model Intercomparison Project (CMIP). The project's objective is to gain insights into past, present, and future climate changes, thus supporting policy-makers and communities worldwide. The understanding of climate phenomena include, among other things, the assessment of various climate models and the quantification of their performance for future projects.

We specifically picked CMIP Phase 3 (CMIP3) data set [51] in an effort to reproduce the results from Section *5.1 Intercomparison Studies* of the original MID paper [16]. However, during this process, we discovered the following pitfalls that prevented us from fully replicating the results:

- the lack of guidelines that specify how to acquire the data from the CMIP3 data repository,
- the lack of details on what data properties (*i.e.,* ensemble runs) were used from CMIP3 data set,
- missing information about the temperature averaging due to the spatial nature of the CMIP3 data set,

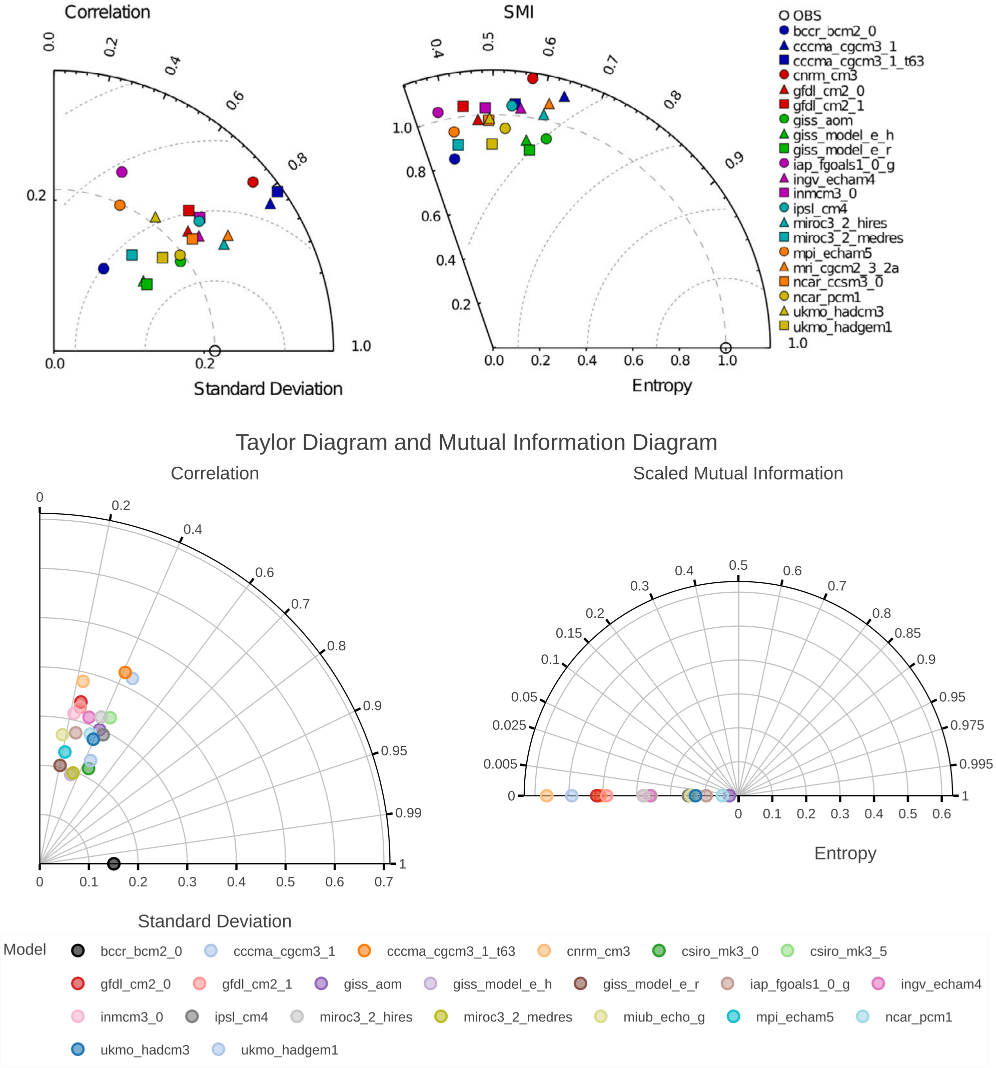


Fig. 6. ***CMIP3* data set**. Taylor Diagram and Mutual Information Diagram of CMIP3 air surface temperature data for the *historical* experiment, of only the first *ensamble* run. The top row shows the diagrams reprinted with permissions from the original study [16], while the bottom row diagrams were created and exported using the library we created — *polar-diagrams*. Model *bccr_bcm2_0* is selected as a reference model. We can see that some models are not visualized on the MID; those models have negative entropies and negative MIs with the reference model. Therefore, by using the Taylor Diagram, we can see the models *miroc3_2_medres* and *csiro_mk3_0* are the most similar to the reference model. This example shows the need to present both diagrams side-by-side. The upper figure is used with permission of Begell House Digital Library, from The mutual information diagram for uncertainty visualization, Correa, C. D., & Lindstrom, P., International Journal for Uncertainty Quantification, 3(3), 2013; permission conveyed through Copyright Clearance Center, Inc.

- missing information about the source of the reference or observation (OBS) data (model),
- the example-specific probability density estimation method.

Nevertheless, we solved each of the problems mentioned above by following our intuition and commonly used approaches when working with climate data. We acquired CMIP3 data by creating the following link query https://esgf-data.dkrz.de/esg-search/wget?download_structure=model&project=CMIP3&experiment=historical&ensemble=run1&variable=ts and downloading the official *wget* script which downloads all model data. As in the original study, the script downloads the data set, which consists of 21 models. The query shows that we selected the data from the CMIP3 project of the *historical* experiment and for the *surface air temperature (ts)* variable. Since the original work is missing the *ensamble* information, we decided to use only *run1* values. Due to the lack of positional information on the data in the original study, we calculated the average of all temperatures across the globe per year and used those values for each model. Fig. 6 shows the original and reproduced results.

As we can see in Fig. 6, both Taylor and MID look very different than those in the original study. The difference is caused by the lack of a step-by-step procedure to replicate the original results and by using a different probability density estimation algorithm. Although the diagrams are not comparable in this way, the overall goal of providing an open-access library is to support the community at large and enable not only static information visualizations but also the creation of interactive data visualizations and the sharing of source code to reproduce the underlying work. Our results reproduce the fundamental principles and analytical steps used in the related work. Although not entirely irreproducible, this also shines a positive light on open tools that facilitate source code adoption, reuse, sharing, and reproducibility. More particularly, positively affecting the advancement of thematic analytics in the field of climate change.

3.2. Example 2 — machine learning model evaluation

The non-parametric and data-type agnostic nature of our library allows us to work with continuous variables without the discretization

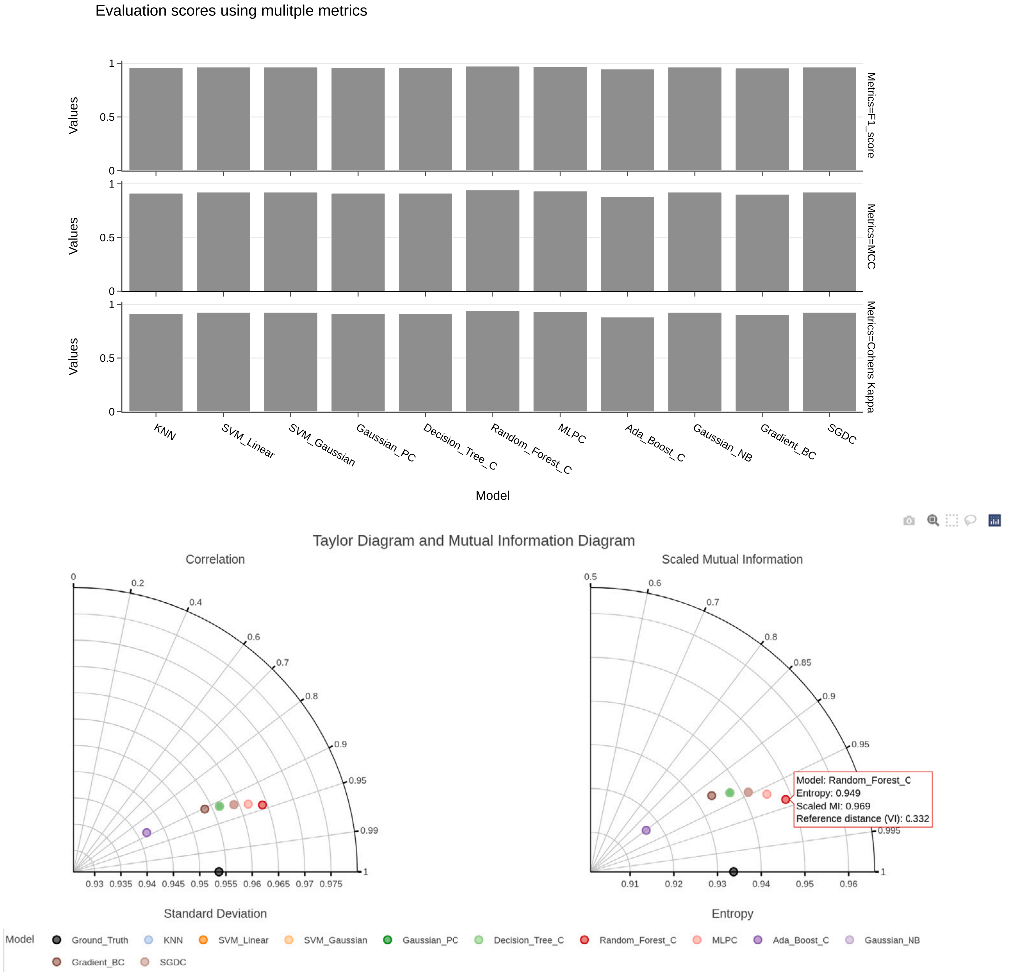


Fig. 7. ***Breast Cancer* data set**. Multiple ML models evaluated on *Breast Cancer* data set. The upper part of the figure contains bar charts of commonly used evaluation scores of classification tasks. We can see all models performed similarly well. However, without further inspection of the bar charts using zooming or a tooltip, it is hard to estimate which model performed the best, which model is the second best, and so on. As an alternative to this approach, the user is able to present the performance of each model by creating a table that holds the final scores. On the other hand, Taylor Diagram and MID facilitate clear distinctions between models without any additional work. We can clearly see the *Random_Forest_C* models being the best, *MLPC* being the second best, and *Ada_Boost_C* being the worst model. We can also notice that some models like *KNN*, *SVM_Linear*, and *SVM_Gaussian* are missing. However, this is not the case. The models are overlapping in both diagrams, and the user is notified with a *Python* warning about this phenomenon.

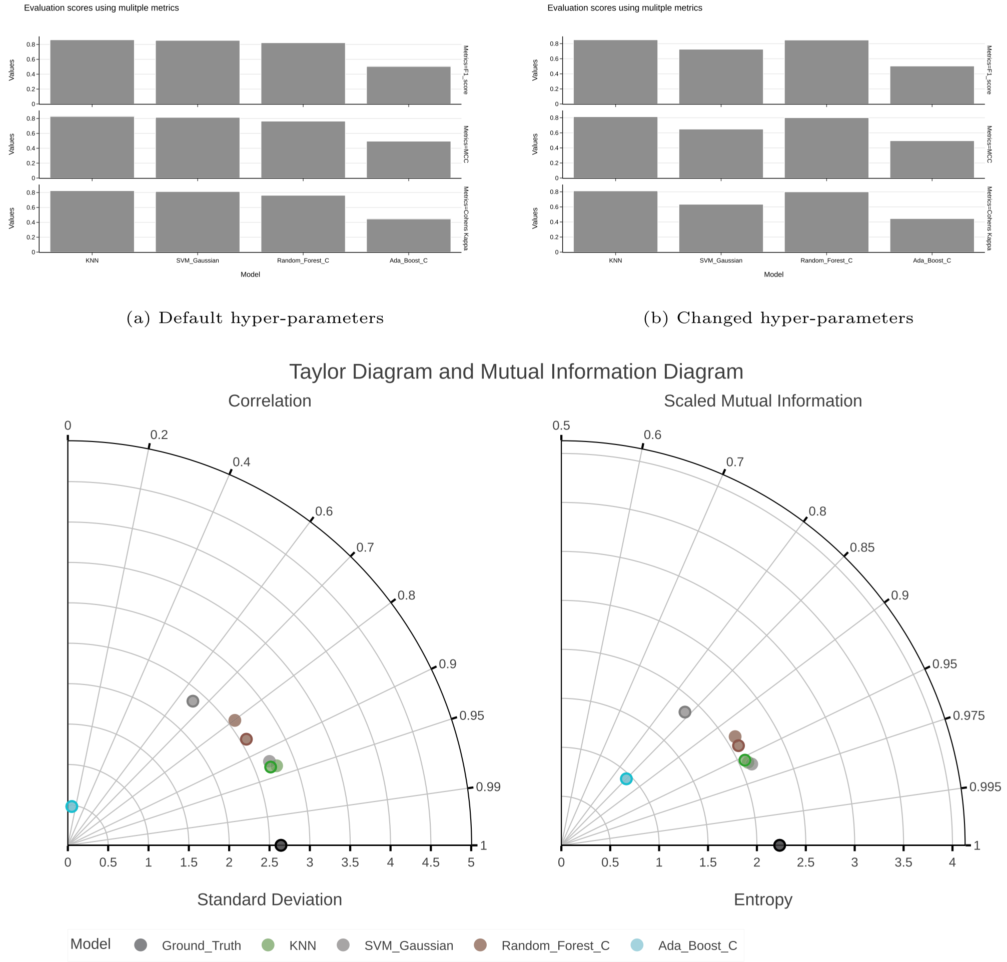
step and the selection of example-specific PDF estimation method, as opposed to the procedure presented in [16]. To showcase the power of *polar-diagrams*, we selected various traditional ML data sets that contain all discrete, all continuous, or some discrete and some continuous features. Furthermore, we chose eleven classification and nine regression models. The main task of the experiment was to assess model performance using the diagrams our library provides, find the best model, and check if our assessment is in line with commonly used performance metrics. We present the experiment in more detail below.

3.2.1. Data sets

We used the following data sets for our ML experiment: *Iris* [52], *Breast Cancer* [53], *Glass* [10], *E. Coli* [54], *Mushroom* [55], *California Housing* [56], and *Ames Housing* [57]. The first five data sets contain a discrete target feature (classification task), while the latter two contain a continuous target feature (regression task). The portion of the target feature used as the test data represents our reference model. We conducted the same preprocessing procedure for all data sets. First, we removed columns that contain identification (ID) numbers. Second, we used the label-encoding method to encode categorical columns of each data set. This step allows the use of the Taylor Diagram for model assessment besides the MID. Third, we removed rows or columns that contain `Null` values. Fourth, and only in the case of the *Mushroom* data set, we sampled the data set using stratification and considered only 40% of all samples. Due to its memory requirements, we had to reduce the size of this specific data set. Fifth, we split the data into the training and test parts with proportions $0.67 : 0.33$. For the classification tasks, this procedure was completed in a stratified fashion. Sixth, we scaled both training and test data using *Scikit-Learn*'s `StandardScaler` that was trained on the training data only. We then proceeded with the ML model training.

3.2.2. Machine learning models

The ML example includes all commonly used ML classification and regression models implemented in the *Scikit-Learn* [38,39] library. We used the following models for both the classification and regression tasks: *k-Nearest Neighbors* [58], *Linear Support Vector (SV) Machine* [59–61], *Kernelized SV Machine* [60,61], *Decision Tree* [62], *Random Forest* [63–65], *Multi-layer Perceptron* [66–68], *Ada Boost* [69–71], *Gradient Boost* [72,73], and *Stochastic Gradient Descent (SGD)* [74–76]. Besides these models, we also used *Gaussian Naive Bayes (NB)* [77,78] for the classification tasks and *Gaussian Process Regressor* [79] for the regression tasks. All models were using the default hyper-parameters, as defined in the *Scikit-Learn* library. For the *Kernelized SV Machine*, we used the *Radial Basis Function (RBF)* kernel with default hyper-parameters.



(a) Default hyper-parameters

(b) Changed hyper-parameters

Fig. 8. ***E. Coli* data set**. A selection of ML model was used with *E. Coli* data set. To showcase the library's ability to visualize two versions of all models, we conducted two classification experiments using four ML models. In the first experiment, we used models with default hyper-parameters, while in the second experiment, we slightly tweaked hyper-parameters, thus causing some models to perform better and some models to perform worse than in the first run. Models from both experiments were evaluated and visualized, as seen in Figs. 8a and 8b. The visible change in these figures is the decrease in performance of the *SVM_Gaussian* model. This can also be seen in both diagrams since the grey dot with a solid border (which encodes the second version of the model) is further from the *Ground_Truth* than the same borderless grey dot. Moreover, diagrams allow us to easily notice the increase in performance of the *Random_Forest_C* model.

To train the models, we used stratified 5-fold cross-validation on the training data while using the following scoring methods to evaluate the performances of the classification models: *accuracy*, *weighted precision*, *weighted recall*, and *weighted F1-score*. To evaluate regression models during training, we used 5-fold cross-validation on the training data with the following scoring methods: *R^2 score* [80], *negative mean absolute error*, *negative mean squared error*, *negative mean squared log error*, and *negative root of mean squared error*. The negative values are used due to the nature of the library, where estimators with higher scores are considered better.

The final evaluation of all models was done with the test data set as defined in Section 3.2.1 and using the scoring methods mentioned earlier. Besides visualizing the model performance using our library, we also visualized different scores acquired during the evaluation. For data sets used in the classification task, we visualized *weighted F1-score*, *ϕ coefficient or Matthews Correlation Coefficient (MCC)* [81,82], and *Cohen's Kappa coefficient* [83]. On the other hand, for data sets used in the regression task, we visualized the *R^2 score*, *mean squared error (MSE)*, and *mean absolute error (MAE)*. It is important to note that when inspecting figures in the paper, higher values are better for all but two metrics: MSE and MAE. For these two metrics, the opposite is true.

Model results for *Breast Cancer*, *E. Coli*, *Ames Housing*, *Glass*, *Iris*, *Mushroom*, and *California Housing* data sets can be seen in Figs. 7, 8, 9, 10, 11, 12, and 13, respectively. Fig. 8 showcases the first extended functionality of *polar-diagrams* that enables users to visualize multiple versions of the same models. Fig. 9 presents the second extended functionality that enables users to visualize one scalar property of each model.

3.3. Example 3 — biomedical similarity assertion

With the rise of electronic medical records and population-level patient profiles, we are getting closer to the widespread use of precision medicine. In order to achieve this goal, it is often required to find similarities between patients, cluster them, and determine the similarity of each new patient to these defined clusters. Besides being comparable to standard medical diagnosis and hence being familiar to physicians, this step also ensures patient privacy and speeds up the decision process [84,85]. On the other hand, comparative studies present an important part of biological research as well. Comparative biology encompasses a plethora of biological sciences (*e.g.*, Ecology, Genomics, Paleontology). It enables users to identify similarities and more specifically the distance of one organism (or other taxa) in relation to another and derive the phylogeny [86,87]. In this section, and for the sake of consistency, we will use the term *model* when considering organisms (or other taxa) and patients.

More than often, the end goal of asserting similarities and finding clusters is the representation of the results in a 2-D space. There-

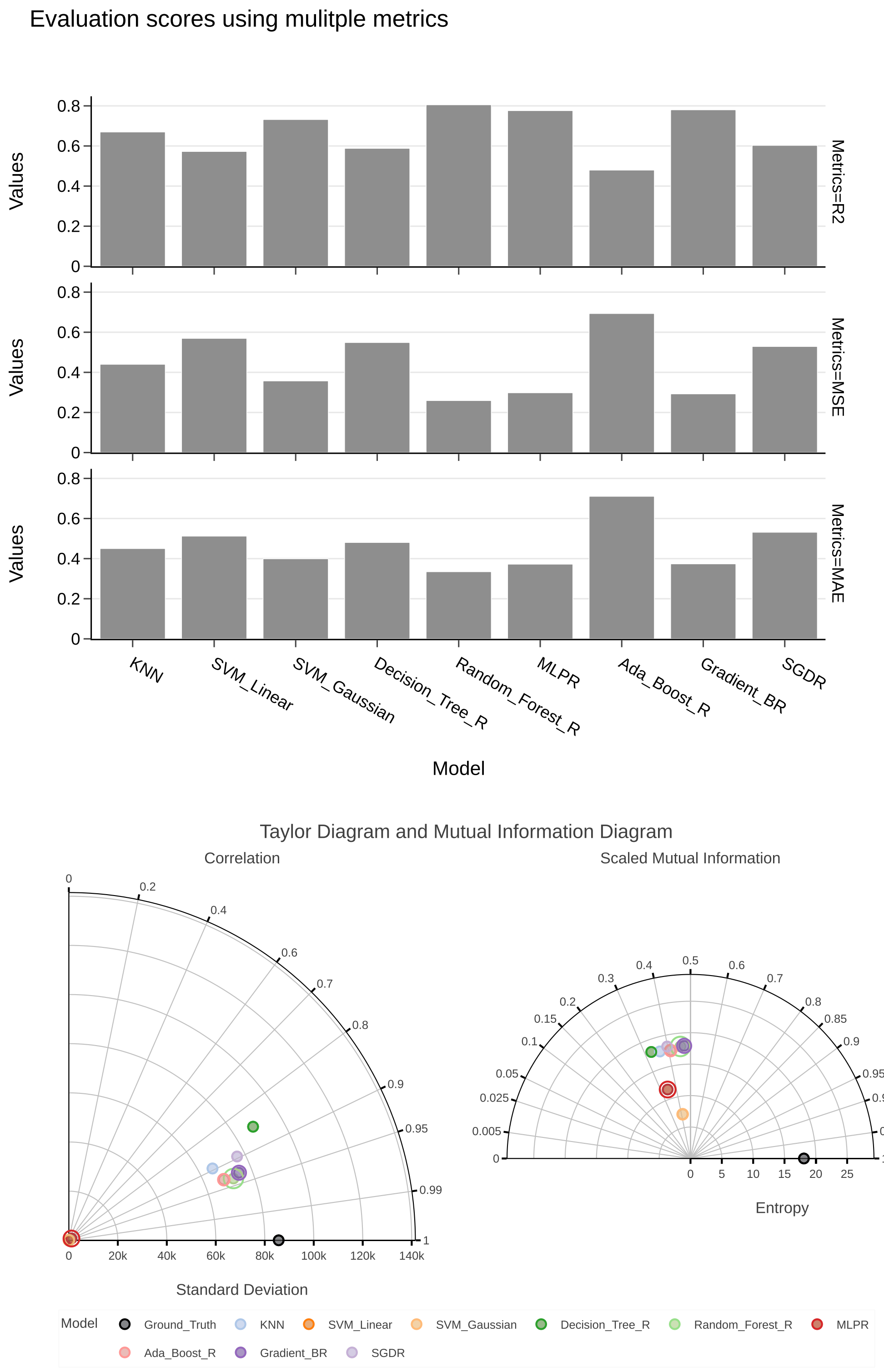


Fig. 9. ***Ames Housing*** **data set**. This example displays the ML models' performances for the regression problem of *Ames Housing* data set. Since the target feature of this data set is continuous, model predictions and ground truth are also continuous. Hence, to visualize all models, the library uses continuous (differential) versions of algorithms for the calculation of entropy and MI. We can see the resulting diagrams are not completely in line, but they agree with both *Random_Forest_R* and *Gradient_BR* being one of the best models for this task. This is completely in line with the commonly used metrics (top row).

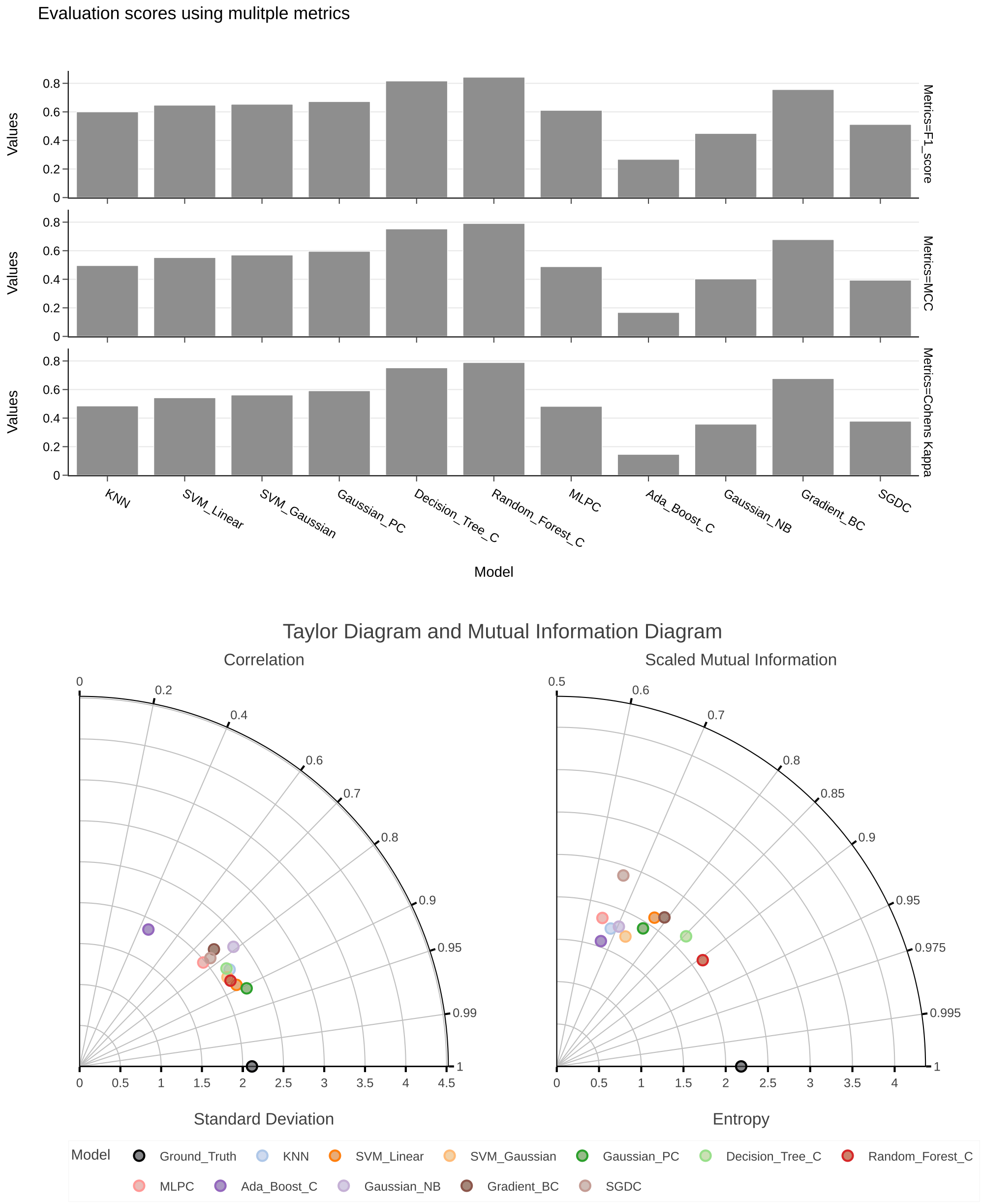


Fig. 10. *Glass* data set. The performance of ML models while solving the classification task of the *Glass* data set differs greatly from all other results. The traditional way of visualizing ML model performance using bar plots (top row) gives us a clear distinction between model performances instead of being hard to read, as in other examples. Therefore, this approach presents a satisfactory way to visualize ML model performances for this data set. The MID created and exported using *polar-diagrams* (bottom row, right) completely agrees with the results of the previously mentioned approach. The best models are *Random_Forest_C*, *Decision_Tree_C*, and *Gaussian_PC* respectively. The lack of power to capture nonlinear relationships between the models hinders the use of the Taylor Diagram for this example.

fore, the traditional approach consists of selecting one of the clustering algorithms (*e.g.*, *K-Means* [88], *OPTICS* [89], *BIRCH* [90]), choosing the distance metric to be used in the algorithm (*e.g.*, Euclidean distance, Manhattan distance), and using some dimensionality reduction technique (*e.g.*, *Principal Component Analysis (PCA)* [*91*], *Multidimensional Scaling (MDS)* [*92*], *T-distributed Stochastic Neighbor Embedding (t-SNE)* [*93*]) to project the data to 2 dimensions and visualize it in a Cartesian plot with colors (or shapes) encoding clusters.

Due to the nature of the Taylor Diagram and the MID, we can skip all these steps and use CRMSE and VI, respectively, to determine similarities between single models, models and clusters, and clusters and clusters (inter-cluster similarity). Multiple studies have shown that VI shows multiple desirable theoretical properties (such as its metric property and its alignment with the lattice of partitions) and, as such, can be used to compare clusters and, by extension, its individual elements [94–96].

To showcase the ability to assert similarities between biomedical models using *polar-diagrams*, we used the *Fertility* [97] (all discrete features) and *Hepatitis* [98] (all continuous features) data sets.

The *Hepatitis* or *HCV* data set consists of patients that are described by demographic properties and laboratory-collected blood values. All patients fall into one of the following categories: *blood donor*, *suspected blood donor*, *hepatitis C patient*, *fibrosis patient*, and *cirrhosis patient*. For the purposes of our study, we included only hepatitis C patients that do not contain `Null` values and without demographic properties. As a result, we were left with twenty patients, each containing ten blood parameters. The results for the *Hepatitis* and the *Fertility* data sets can be seen in Figs. 14 and 15.

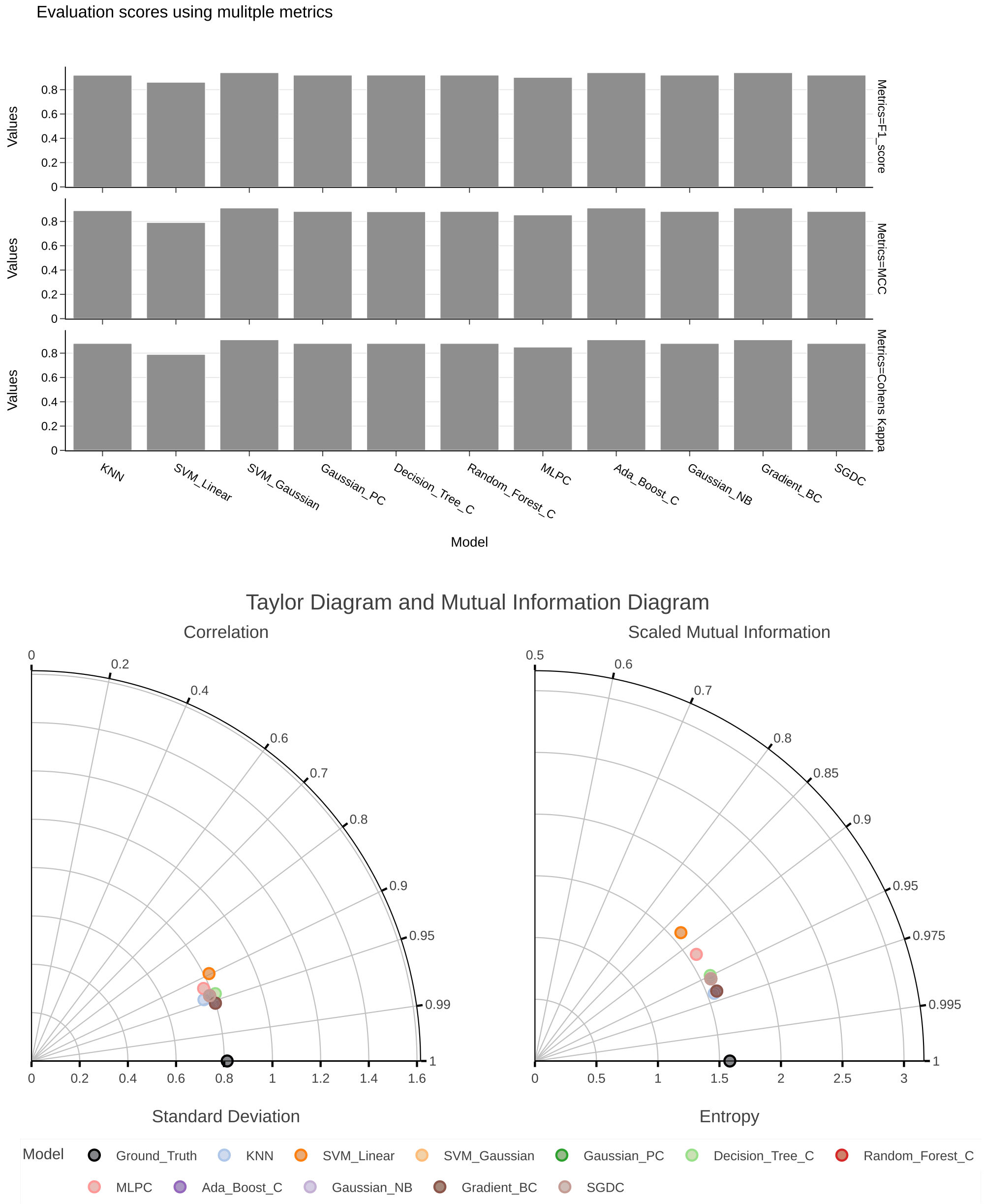


Fig. 11. ***Iris* data set**. Both commonly used ML metrics (top row) and diagrams from our library (bottom row) align with *SVM_Linear* and *MLPC* being the worst models on the *Iris* data set, respectively. The best model performances are hard to read from the top-row visualization, but this problem remains in the case of diagrams as well. However, a quick use of the *Zoom* tool provided by *polar-diagrams* would allow us to zoom into the clusters and determine which model is the best.

Besides visualizing patients using our library, we also visualized them using the traditional approach. First, we used the *Caliński and Harabasz* score [99] to find the best number of clusters for the *K-Means* clustering algorithm. Second, we clustered patients using *K-Means*. Third, we used *t-SNE* [93] to reduce the dimensionality of our data to two dimensions. Fourth and last, we visualized data using a 2-D scatter plot, with clusters color-coded and patients represented by shapes. The results can be found in the upper part of Figs. 14 and 15.

The overview of all results of our study can be seen in Table 1.

4. Discussion

Thanks to our library, we solve multiple hurdles of MID that were mentioned in the original work. However, certain limiting factors remain as “weak” problems, and we take the liberty to discuss them along with other possible improvements.

First, *polar-diagrams* only supports models represented by n-dimensional numerical vectors, which may be perceived as a “hard” constraint. However, representing the data as numerical vectors is a common practice.

Second, the number of models users want to visualize can be perceived as a “soft” constraint. This is caused by the limitation of the human perceptual and cognitive system in its ability to both perceive and retain a large number of categorical colors [43,100–103]. This is especially true for colorblind-safe schemes/palettes which can be used to make designs more accessible to users with visual impairments. Indeed, our library does not prevent users from parsing more than twenty models since the colors are repeated after the twentieth model. Although a serious problem in color usage, this is why this constraint is considered “soft”. Yet, it does not prevent users from exploring the data interactively since they can exclude models that are not of interest by using the interactive legend. When users are faced with more than twenty models, ambiguity is created with repeating colors. Unfortunately, creating

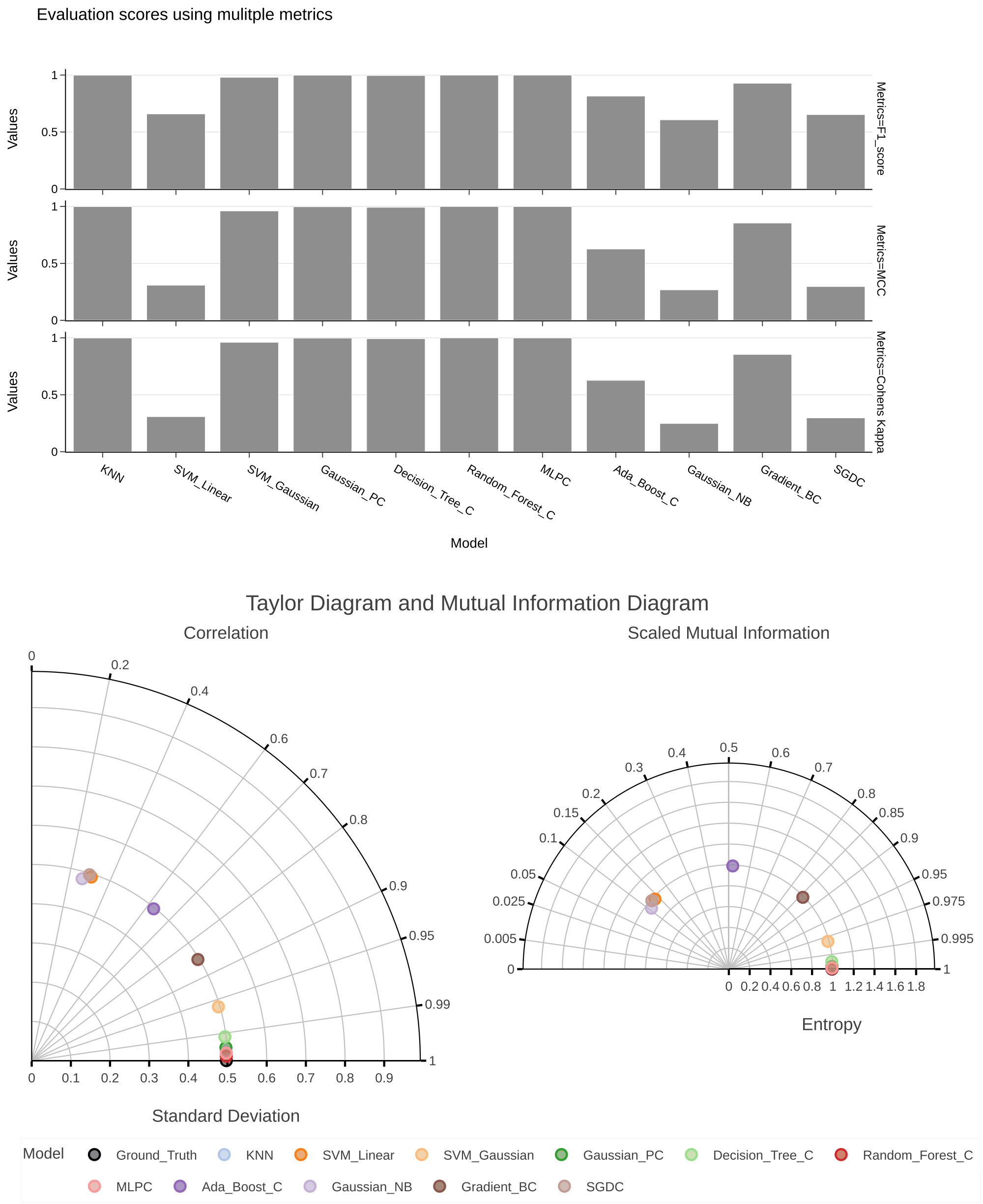


Fig. 12. ***Mushroom* data set**. In the case of the *Mushroom* data set, both diagrams from *polar-diagrams* (bottom row) and commonly used ML metrics (top row) align with each other. As with the *Iris* data set, it is hard to assess the best models using the top-row visualization. This is also the case with the diagrams. The interactive *Zoom* tool would help us to single out the best models for this data set quickly. Indeed, it is clear that *Random_Forest_C* and *MLPC* performed the best, which is in alignment with the results from the original study [55].

a color palette with more than twenty distinguishable colors is unattainable. As an example, *Colorgorical*, a tool by Gramazio *et al.* [104], which creates discriminable color palettes using three color-discriminability scores and a color-preference score, returns a partial palette and an error when more than twenty-one colors are generated due to the exhaustion of the color space. Indeed, more classes require more colors, which are increasingly difficult to distinguish. Depending on the task at hand and the audience, the number of colors varies. For the example of color coding of symbols, Colin Ware suggests using no more than ten colors if reliable identification is required, especially if the symbols are to be used against a variety of backgrounds [105]. Additionally, we explored the possibility of using color harmonies to expand our color palette [106]. However, we have found that this method is insufficient when the number of colors in the palette is more than four. The seven major color schemes are monochromatic, analogous, complementary, split complementary, triadic, square, and rectangle (or tetradic); resulting in a maximum number of four colors. In the first iterations of the library, we tried encoding models using shapes as well in order to increase the number of distinct model encodings. However, that approach yielded diagrams that were cluttered and hard to read. Our results are further corroborated by works [107,108]. We currently give users the freedom and responsibility to decide which models are visualized and control for repeated colors.

Third, another limitation is with models that have a continuous data type. In this case, as we described in Section 2.1.3, our library calculates continuous (differential) entropy and MI for the creation of MID. Since these parameters can be negative, the resulting MID might be empty or without some models, as can be seen in Figs. 13 and 6. However, as mentioned earlier, we do not consider this a true limitation since it presents the nature of continuous entropy.

Fourth, the MID can be further improved by incorporating a normalized variation of information (NVI) as presented in paper [109].

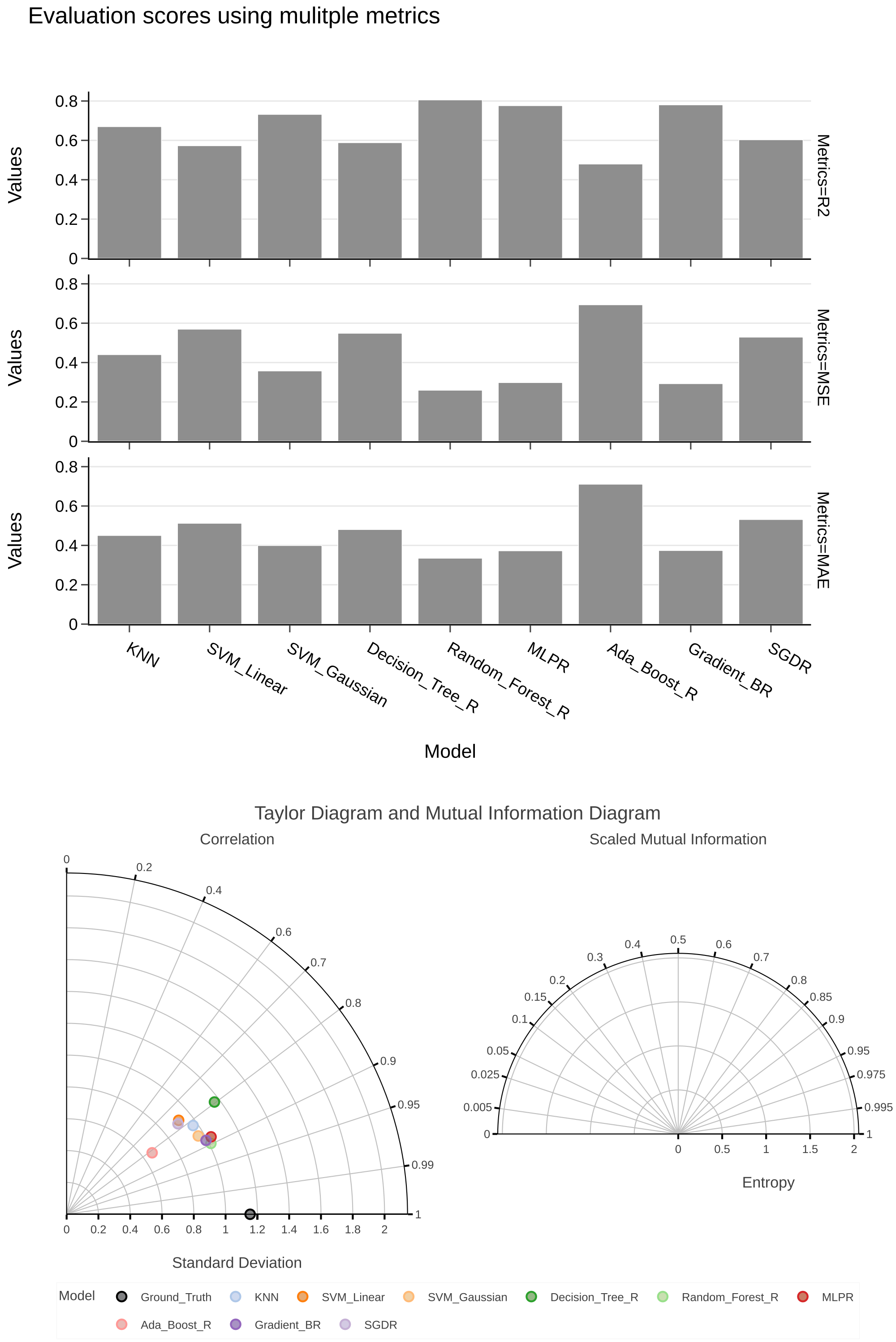


Fig. 13. ***California Housing* data set.** In the case of the *California* data set, due to it being a regression problem, all entropies and MIs are negative, hence an empty MID (bottom row, right). Taylor Diagram (bottom row, left) gives better results since it aligns with all commonly used metrics (top row). Both the diagram and bar charts agree with *Random_Forest_R*, *MLPR*, and *Gradient_BR* being the best models for this task, respectively.

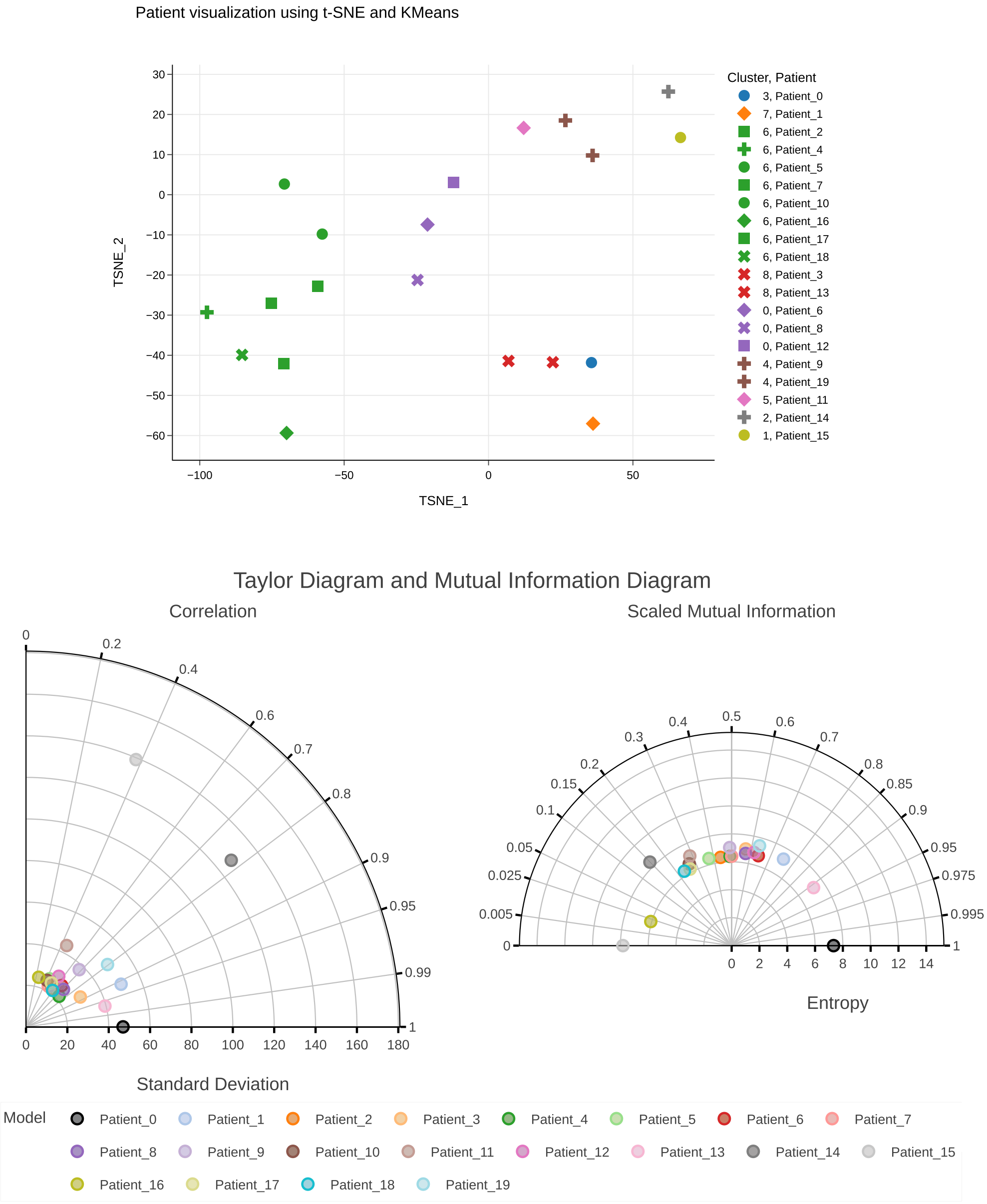


Fig. 14. ***Hepatitis* data set**. Hepatitis C patients are visualized using the traditional approach (top row) and *polar-diagrams* (bottom row). While the creation of the top-row scatter plot required algorithm selection, data processing, and computer science knowledge, the bottom-row polar diagrams do not require any domain-specific knowledge or experience. *Patient_0* is selected as a reference model. We can notice the traditional approach (top row) declaring *Patient_0* as the only element of the cluster. However, if we consider distance in a 2-D plot created by *t-SNE*, we see that the most similar patients with the reference patient are *Patient_13* and *Patient_1*. This agrees with both diagrams. However, both diagrams also tell us that *Patient_14* and *Patient_15* are the most dissimilar to the reference model.

However, this implementation was out of the scope of our paper since it requires further research into its application in the form of a diagram.

Fifth, it is important to note that the differential entropy implemented in *SciKit-Learn* library and used for the creation of MID is not the actual continuous analogue of discrete (Shannon) entropy [110]. All methods mentioned in Section 2.2 present the limiting case of the actual continuous version of discrete entropy called the limiting density of discrete points (LDDP). To the best of our knowledge, the implementation of this measure in *Python* does not currently exist. However, we plan to include this measure as another option for calculating the differential entropy in one of our future versions.

Sixth, we also acknowledge another model comparison chart type — the Target Diagram [111]. This Cartesian plot type extends the functionalities of the Taylor Diagram by including the sign of the bias to the summary information (which is unbiased) and summarizes how they each contribute to the total RMSE. The need for a summary diagram that also encodes and visualizes the statistical bias was also confirmed in [112]. However, our library solves this problem with the previously introduced functionality to visualize one scalar property of each model. Implementing this plot type in *polar-diagrams* is thus unnecessary and out of scope due to it not being of a polar type.

Seventh and last, it is important to mention that our diagrams do not contain isolines indicating CRMSE, VI, and RVI. Instead, we chose to show these values in a tooltip. One of the reasons behind this decision is the lack of support for such functionality in all high-level visualization libraries we reviewed. The other reason is to make diagrams as visually decluttered and readable as possible. The same rationale applies to our decision not to use the traditional approaches for multi-version model visualizations. These approaches use arrows to indicate the flow of in-

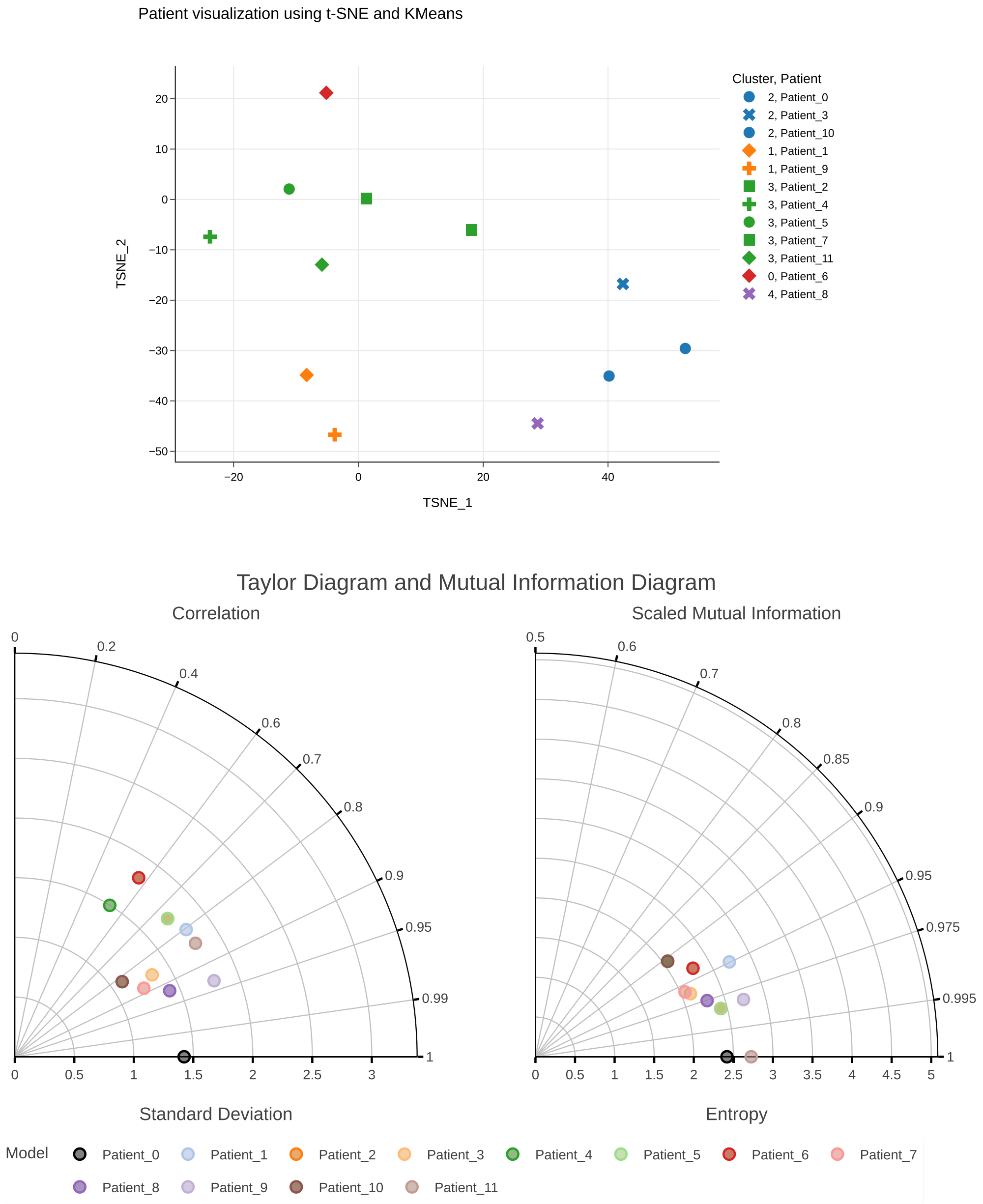


Fig. 15. ***Fertility* data set**. *Fertility* data set proves to be one of the examples where the results from the diagrams do not align and disagree with the traditional approach (top row). As we can see, the Taylor Diagram (bottom row, left) depicts *Patient_9*, *Patient_8*, and *Patient_2* as being the most similar to the reference *Patient_0*. On the other hand, MID (bottom row, right) shows *Patient_11* as being the most similar to the reference model. Due to the equidistant nature of models (all having the same VI), the second most similar models are *Patient_5* and *Patient_2*. This is confirmed by the *Python* warning the library produced. The traditional approach places *Patient_10* and *Patient_3* in the same cluster as the reference patient. Due to such a great disagreement, more analysis is required for this data set.

formation. However, when there are ten or more models, the resulting diagrams have twenty or more model markers with ten arrow lines. Depending on the model shift, those lines can have multiple intersections with each other, thus rendering the resulting visualization incomprehensible. This is the reason we decided to encode the second versions with the same markers as the first versions, but with the added solid border for the purpose of differentiating them.

5. Conclusion

Our library provides the first public implementation of the MID and the first implementation of the interactive Taylor Diagram. It was developed by following all "good" programming conventions (*e.g., PEP 8, PEP 20, PEP 257, PEP 287* [34]) and with state-of-the-art open-source data manipulation, scientific computing, mathematics, machine learning, and high-level visualization libraries. The resulting diagrams can be exported in publication-ready static-image formats.

Furthermore, our library extends the expressiveness of both diagrams by providing two additional functionalities — the ability to visualize multiple versions of all models and the ability to visualize one scalar property of each model.

By providing the interactive aspects to both diagrams, the users are encouraged to explore results in a way not available until now. We expect *polar-diagrams* to be a valuable tool in climate, biomedical, machine learning, and other domains that produce complex models and offer further insights into interdependencies between such models.

Table 1
Overview of the results. The *Agreement* column shows whether the diagrams from our library are in agreement with the traditional evaluation approaches. The *Reproducibility* column shows whether the results are reproducible.

Data set	Agreement	Reproducibility	Result
CMIP3	No[1]	No[1]	Fig. 6
Iris	Yes	Yes	Fig. 11
Breast Cancer	Yes	Yes	Fig. 7
Glass	Yes	Yes	Fig. 10
E. Coli	Yes	Yes	Fig. 8
Mushroom	Yes	Yes	Fig. 12
California Housing	No	Yes	Fig. 13
Ames Housing	Yes	Yes	Fig. 9
Hepatitis	Yes	Yes	Fig. 14
Fertility	No	Yes	Fig. 15

[1] Due to the lack of data in the original study [16].

Code availability

Source code, help, and documentation can be found at https://github.com/AAnzel/Polar-Diagrams-for-Model-Comparison. The repository also contains the source code necessary to reproduce all examples in this work. Our library is also available on the official third-party software repository for *Python* packages called *PyPI* under the following link https://pypi.org/project/polar-diagrams/. It is licensed under the *GNU General Public License, Version 3.0* and can be manipulated, improved, and extended freely by any user.

Availability of data and materials

The data used in this study can be either found or downloaded using the scripts present at https://github.com/AAnzel/Polar-Diagrams-for-Model-Comparison/tree/master/Data. All data sets are also cited in Section 3 and can be downloaded from the originating studies.

CRediT authorship contribution statement

A.A. wrote the manuscript, designed and developed the library, conducted experiments, and evaluated results. D.H. discussed the results and revised the manuscript. G.H. supervised the project, guided the development, proofread, and revised the manuscript. All authors read and approved the final manuscript.

Declaration of competing interest

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

Acknowledgements

We acknowledge the modeling groups, the Program for Climate Model Diagnosis and Intercomparison (PCMDI) and the WCRP's Working Group on Coupled Modelling (WGCM) for their roles in making available the WCRP CMIP3 multi-model data set. Support of this data set is provided by the Office of Science, U.S. Department of Energy.

Funding

This work was financially supported by the LOEWE program of the State of Hesse (Germany) in the MOSLA research cluster.

References

[1] A. Biswas, S. Dutta, H.-W. Shen, J. Woodring, An information-aware framework for exploring multivariate data sets, IEEE Trans. Vis. Comput. Graph. 19 (12) (2013) 2683–2692, https://doi.org/10.1109/TVCG.2013.133.

[2] M. Chen, M. Feixas, I. Viola, A. Bardera, H. Shen, M. Sbert, Information Theory Tools for Visualization, AK Peters Visualization Series, CRC Press, 2016.

[3] L.-E. Pommé, R. Bourqui, R. Giot, D. Auber, Relative confusion matrix: efficient comparison of decision models, in: 2022 26th International Conference Information Visualisation (IV), 2022, pp. 98–103.

[4] C. Ware, Chapter ten - interacting with visualizations, in: C. Ware (Ed.), Information Visualization, fourth edition, Interactive Technologies, Morgan Kaufmann, 2021, pp. 359–392, https://www.sciencedirect.com/science/article/pii/B9780128128756000104.

[5] S.S. Stevens, On the theory of scales of measurement, Science 103 (2684) (1946) 677–680, https://doi.org/10.1126/science.103.2684.677, arXiv: https://www.science.org/doi/pdf/10.1126/science.103.2684.677, https://www.science.org/doi/abs/10.1126/science.103.2684.677.

[6] A. Artero, M. de Oliveira, H. Levkowitz, Enhanced high dimensional data visualization through dimension reduction and attribute arrangement, in: Tenth International Conference on Information Visualisation (IV'06), 2006, pp. 707–712.

[7] C. Hurley, R. Oldford, Pairwise display of high-dimensional information via Eulerian tours and Hamiltonian decompositions, J. Comput. Graph. Stat. 19 (12 2010), https://doi.org/10.1198/jcgs.2010.09136.

[8] L.F. Lu, M.L. Huang, T.-H. Huang, A new axes re-ordering method in parallel coordinates visualization, in: 2012 11th International Conference on Machine Learning and Applications 2, 2012, pp. 252–257.

[9] W. Peng, M. Ward, E. Rundensteiner, Clutter reduction in multi-dimensional data visualization using dimension reordering, in: IEEE Symposium on Information Visualization, 2004, pp. 89–96.

[10] I.W. Evett, E.J. Spiehler, Rule induction in forensic science, in: Knowledge Based Systems, Halsted Press, USA, 1989, pp. 152–160.

[11] J. Zhou, W. Huang, F. Chen, Facilitating machine learning model comparison and explanation through a radial visualisation, Energies 14 (21) (2021), https://doi.org/10.3390/en14217049, https://www.mdpi.com/1996-1073/14/21/7049.

[12] J. Talbot, B. Lee, A. Kapoor, D.S. Tan, Ensemblematrix: interactive visualization to support machine learning with multiple classifiers, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '09, Association for Computing Machinery, New York, NY, USA, 2009, pp. 1283–1292.

[13] D. Ren, S. Amershi, B. Lee, J. Suh, J.D. Williams, Squares: supporting interactive performance analysis for multiclass classifiers, IEEE Trans. Vis. Comput. Graph. 23 (1) (2017) 61–70, https://doi.org/10.1109/TVCG.2016.2598828.

[14] S. Yatkin, M. Gerboles, A. Borowiak, S. Davila, L. Spinelle, A. Bartonova, F. Dauge, P. Schneider, M. Van Poppel, J. Peters, C. Matheeussen, M. Signorini, Modified target diagram to check compliance of low-cost sensors with the data quality objectives of the European air quality directive, Atmos. Environ. 273 (2022) 118967, https://doi.org/10.1016/j.atmosenv.2022.118967, https://www.sciencedirect.com/science/article/pii/S1352231022000322.

[15] K.E. Taylor, Summarizing multiple aspects of model performance in a single diagram, J. Geophys. Res., Atmos. 106 (D7) (2001) 7183–7192, https://doi.org/10.1029/2000JD900719, arXiv: https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2000JD900719, https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2000JD900719.

[16] C. Correa, P. Lindstrom, The mutual information diagram for uncertainty visualization, Int. J. Uncertain. Quantificat. 3 (2013) 187–201, https://doi.org/10.1615/Int.J.UncertaintyQuantification.2012003959.

[17] C.A. Gueymard, A review of validation methodologies and statistical performance indicators for modeled solar radiation data: towards a better bankability of solar projects, Renew. Sustain. Energy Rev. 39 (2014) 1024–1034, https://doi.org/10.1016/j.rser.2014.07.117, https://www.sciencedirect.com/science/article/pii/S1364032114005693.

[18] P.A. Rochford, Skillmetrics: a matlab package for calculating the skill of model predictions against observations, https://github.com/PeterRochford/SkillMetricsToolbox, 2016.

[19] G. Maze, Taylor diagram, https://www.mathworks.com/matlabcentral/fileexchange/20559-taylor-diagram, 2023.

[20] P.A. Rochford, Skillmetrics: a python package for calculating the skill of model predictions against observations, http://github.com/PeterRochford/SkillMetrics, 2016.

[21] L. J, Plotrix: a package in the red light district of r, R News 6 (4) (2006) 8–12.

[22] R. Brown, A. Gleason, M. Brown, Advanced Mathematics: Precalculus with Discrete Mathematics and Data Analysis, McDougal Littell, 1994, https://books.google.de/books?id=0QwOxgEACAAJ.

[23] S. Elvidge, M.J. Angling, B. Nava, On the use of modified Taylor diagrams to compare ionospheric assimilation models, Radio Sci. 49 (9) (2014) 737–745, https://doi.org/10.1002/2014RS005435, arXiv: https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1002/2014RS005435, https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1002/2014RS005435.

[24] R. Smith, A mutual information approach to calculating nonlinearity, Stat 4 (1) (2015) 291–303, https://doi.org/10.1002/sta4.96, arXiv: https://

onlinelibrary.wiley.com/doi/pdf/10.1002/sta4.96, https://onlinelibrary.wiley.com/doi/abs/10.1002/sta4.96.

[25] P. Laarne, M.A. Zaidan, T. Nieminen, ennemi: non-linear correlation detection with mutual information, SoftwareX 14 (2021) 100686, https://doi.org/10.1016/j.softx.2021.100686, https://www.sciencedirect.com/science/article/pii/S2352711021000315.

[26] A. Strehl, J. Ghosh, Cluster ensembles — a knowledge reuse framework for combining multiple partitions, J. Mach. Learn. Res. 3 (null) (2003) 583–617, https://doi.org/10.1162/153244303321897735.

[27] B.C. Ross, Mutual information between discrete and continuous data sets, PLoS ONE 9 (2) (2014) 1–5, https://doi.org/10.1371/journal.pone.0087357.

[28] A. Kraskov, H. Stögbauer, P. Grassberger, Estimating mutual information, Phys. Rev. E 69 (2004) 066138, https://doi.org/10.1103/PhysRevE.69.066138, https://link.aps.org/doi/10.1103/PhysRevE.69.066138.

[29] B. Van Es, Estimating functionals related to a density by a class of statistics based on spacings, Scand. J. Stat. (1992) 61–72.

[30] N. Ebrahimi, K. Pflughoeft, E. Soofi, Two measures of sample entropy, Stat. Probab. Lett. 20 (1994) 225–234, https://doi.org/10.1016/0167-7152(94)90046-9.

[31] O. Vasicek, A test for normality based on sample entropy, J. R. Stat. Soc., Ser. B, Methodol. 38 (1) (1976) 54–59, https://doi.org/10.1111/j.2517-6161.1976.tb01566.x, arXiv: https://rss.onlinelibrary.wiley.com/doi/pdf/10.1111/j.2517-6161.1976.tb01566.x, https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.2517-6161.1976.tb01566.x.

[32] P. Crzcgorzewski, R. Wirczorkowski, Entropy-based goodness-of-fit test for exponentiality, Commun. Stat., Theory Methods 28 (5) (1999) 1183–1202, https://doi.org/10.1080/03610929908832351.

[33] H. Alizadeh Noughabi, Entropy estimation using numerical methods, Ann. Data Sci. 2 (06 2015), https://doi.org/10.1007/s40745-015-0045-9.

[34] G. van Rossum, J. de Boer, Interactively testing remote servers using the python programming language, Quart. - Cent. Wiskd. Inform. 4 (4) (1991) 283–304.

[35] T. pandas development team, Pandas-dev/pandas: pandas, Nov. 2022, https://doi.org/10.5281/zenodo.7344967.

[36] Wes McKinney, Data structures for statistical computing in python, in: Stéfan van der Walt, Jarrod Millman (Eds.), Proceedings of the 9th Python in Science Conference, 2010, pp. 56–61.

[37] C.R. Harris, K.J. Millman, S.J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N.J. Smith, R. Kern, M. Picus, S. Hoyer, M.H. van Kerkwijk, M. Brett, A. Haldane, J.F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, T.E. Oliphant, Array programming with NumPy, Nature 585 (7825) (2020) 357–362, https://doi.org/10.1038/s41586-020-2649-2.

[38] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: machine learning in Python, J. Mach. Learn. Res. 12 (2011) 2825–2830.

[39] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. Vanderplas, A. Joly, B. Holt, G. Varoquaux, Api design for machine learning software: experiences from the scikit-learn project, ArXiv arXiv:1309.0238 [abs], 2013.

[40] C.E. Shannon, A mathematical theory of communication, Bell Syst. Tech. J. 27 (3) (1948) 379–423, https://doi.org/10.1002/j.1538-7305.1948.tb01338.x.

[41] P. Virtanen, R. Gommers, T.E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S.J. van der Walt, M. Brett, J. Wilson, K.J. Millman, N. Mayorov, A.R.J. Nelson, E. Jones, R. Kern, E. Larson, C.J. Carey, İ. Polat, Y. Feng, E.W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E.A. Quintero, C.R. Harris, A.M. Archibald, A.H. Ribeiro, F. Pedregosa, P. van Mulbregt, SciPy 1.0 contributors, SciPy 1.0: fundamental algorithms for scientific computing in Python, Nat. Methods 17 (2020) 261–272, https://doi.org/10.1038/s41592-019-0686-2.

[42] P.T. Inc, Collaborative data science, https://plot.ly, 2015.

[43] T. Munzner, A nested model for visualization design and validation, IEEE Trans. Vis. Comput. Graph. 15 (6) (2009) 921–928, https://doi.org/10.1109/TVCG.2009.111.

[44] M. Burch, D. Weiskopf, On the benefits and drawbacks of radial diagrams, in: On the Benefits and Drawbacks of Radial Diagrams, Springer, New York, New York, NY, 2014, pp. 429–451.

[45] K.-P. Yee, D. Fisher, R. Dhamija, M. Hearst, Animated exploration of dynamic graphs with radial layout, in: IEEE Symposium on Information Visualization, 2001, INFOVIS 2001, 2001, pp. 43–50.

[46] L. Qiang, C. Bingjie, Storycake: a hierarchical plot visualization method for storytelling in polar coordinates, in: 2016 International Conference on Cyberworlds (CW), 2016, pp. 211–218.

[47] C. Vehlow, M. Burch, H. Schmauder, D. Weiskopf, Radial layered matrix visualization of dynamic graphs, in: 2013 17th International Conference on Information Visualisation, 2013, pp. 51–58.

[48] L. Beard, N. Aghassibake, Tableau (version 2020.3), J. Med. Libr. Assoc. 109 (1) (Jan. 2021), https://doi.org/10.5195/jmla.2021.1135.

[49] F.J. Anscombe, Graphs in statistical analysis, Am. Stat. 27 (1) (1973) 17–21, https://doi.org/10.1080/00031305.1973.10478966, arXiv: https://www.tandfonline.com/doi/pdf/10.1080/00031305.1973.10478966, https://www.tandfonline.com/doi/abs/10.1080/00031305.1973.10478966.

[50] J. Matejka, G. Fitzmaurice, Same stats, different graphs: generating datasets with varied appearance and identical statistics through simulated annealing, in: Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, 2017, pp. 1290–1294.

[51] G.A. Meehl, C. Covey, T. Delworth, M. Latif, B. McAvaney, J.F.B. Mitchell, R.J. Stouffer, K.E. Taylor, The wcrp cmip3 multimodel dataset: a new era in climate change research, Bull. Am. Meteorol. Soc. 88 (9) (2007) 1383–1394, https://doi.org/10.1175/BAMS-88-9-1383, https://journals.ametsoc.org/view/journals/bams/88/9/bams-88-9-1383.xml.

[52] R.A. Fisher, The use of multiple measurements in taxonomic problems, Ann. Eugen. 7 (2) (1936) 179–188, https://doi.org/10.1111/j.1469-1809.1936.tb02137.x, arXiv: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1469-1809.1936.tb02137.x, https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1469-1809.1936.tb02137.x.

[53] W. Wolberg, O. Mangasarian, Multisurface method of pattern separation for medical diagnosis applied to breast cytology, Proc. Natl. Acad. Sci. USA 87 (1991) 9193–9196, https://doi.org/10.1073/pnas.87.23.9193.

[54] P. Horton, K. Nakai, A probabilistic classification system for predicting the cellular localization sites of proteins, in: Proceedings of the Fourth International Conference on Intelligent Systems for Molecular Biology, AAAI Press, 1996, pp. 109–115.

[55] D. Wagner, D. Heider, G. Hattab, Mushroom data creation, curation, and simulation to support classification tasks, Sci. Rep. 11 (04 2021), https://doi.org/10.1038/s41598-021-87602-3.

[56] R. Kelley Pace, R. Barry, Sparse spatial autoregressions, Stat. Probab. Lett. 33 (3) (1997) 291–297, https://doi.org/10.1016/S0167-7152(96)00140-X, https://www.sciencedirect.com/science/article/pii/S016771529600140X.

[57] D. Cock, Ames, Iowa: alternative to the Boston housing data as an end of semester regression project, J. Stat. Educ. 19 (11 2011), https://doi.org/10.1080/10691898.2011.11889627.

[58] T. Cover, P. Hart, Nearest neighbor pattern classification, IEEE Trans. Inf. Theory 13 (1) (1967) 21–27, https://doi.org/10.1109/TIT.1967.1053964.

[59] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, C.-J. Lin Liblinear, A library for large linear classification, J. Mach. Learn. Res. 9 (61) (2008) 1871–1874, http://jmlr.org/papers/v9/fan08a.html.

[60] C.-C. Chang, C.-J. Lin Libsvm, A library for support vector machines, ACM Trans. Intell. Syst. Technol. 2 (3) (May 2011), https://doi.org/10.1145/1961189.1961199.

[61] K. Crammer, Y. Singer, On the algorithmic implementation of multiclass kernel-based vector machines, J. Mach. Learn. Res. 2 (2002) 265–292.

[62] A.D. Gordon, Classification and regression trees, Biometrics 40 (3) (1984) 874, http://www.jstor.org/stable/2530946.

[63] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32, https://doi.org/10.1023/A:1010933404324.

[64] L. Breiman, Arcing classifiers, Ann. Stat. 26 (3) (1998) 801–824, http://www.jstor.org/stable/120055.

[65] P. Geurts, D. Ernst, L. Wehenkel, Extremely randomized trees, Mach. Learn. 63 (1) (2006) 3–42, https://doi.org/10.1007/s10994-006-6226-1.

[66] K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: surpassing human-level performance on ImageNet classification, in: 2015 IEEE International Conference on Computer Vision (ICCV), 2015.

[67] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in: Y.W. Teh, M. Titterington (Eds.), Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, in: Proceedings of Machine Learning Research, vol. 9, PMLR, Chia Laguna Resort, Sardinia, Italy, 2010, pp. 249–256, https://proceedings.mlr.press/v9/glorot10a.html.

[68] G.E. Hinton, Connectionist learning procedures, Artif. Intell. 40 (1) (1989) 185–234, https://doi.org/10.1016/0004-3702(89)90049-0, https://www.sciencedirect.com/science/article/pii/0004370289900490.

[69] Y. Freund, R.E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, J. Comput. Syst. Sci. 55 (1) (1997) 119–139, https://doi.org/10.1006/jcss.1997.1504, https://www.sciencedirect.com/science/article/pii/S002200009791504X.

[70] J. Zhu, S. Rosset, H. Zou, T. Hastie, Multi-class adaboost, Stat. Interface 2 (02 2006), https://doi.org/10.4310/SII.2009.v2.n3.a8.

[71] T. Hastie, R. Tibshirani, J. Friedman, Ensemble learning, in: Ensemble Learning, Springer New York, New York, NY, 2009, pp. 605–624.

[72] J.H. Friedman, Greedy function approximation: a gradient boosting machine, Ann. Stat. 29 (5) (2001) 1189–1232, https://doi.org/10.1214/aos/1013203451.

[73] J.H. Friedman, Stochastic gradient boosting, Comput. Stat. Data Anal. 38 (4) (2002) 367–378, https://doi.org/10.1016/S0167-9473(01)00065-2.

[74] L. Bottou, Large-scale machine learning with stochastic gradient descent, in: Y. Lechevallier, G. Saporta (Eds.), Proceedings of COMPSTAT'2010, Physica-Verlag HD, Heidelberg, 2010, pp. 177–186.

[75] S. Shalev-Shwartz, Y. Singer, N. Srebro, Pegasos: primal estimated sub-GrAdient SOlver for SVM, in: ICML '07, Association for Computing Machinery, New York, NY, USA, 2007.

[76] Y. Tsuruoka, J. Tsujii, S. Ananiadou, Stochastic gradient descent training for L1-regularized log-linear models with cumulative penalty, in: ACL '09, Association for Computational Linguistics, USA, 2009.

[77] H. Zhang, The Optimality of Naive Bayes, American Association for Artificial Intelligence, 2004.

[78] T.F. Chan, G.H. Golub, R.J. LeVeque, Updating formulae and a pairwise algorithm for computing sample variances, Tech. Rep., Stanford University, Stanford, CA, USA, 1979.

[79] C.E. Rasmussen, C.K.I. Williams, Gaussian Processes for Machine Learning, The MIT Press, 2005.

[80] D. Chicco, M.J. Warrens, G. Jurman, The coefficient of determination r-squared is more informative than smape, mae, mape, mse and rmse in regression analysis evaluation, PeerJ Comput. Sci. 7 (2021), https://doi.org/10.7717/peerj-cs.623.

[81] B. Matthews, Comparison of the predicted and observed secondary structure of t4 phage lysozyme, Biochim. Biophys. Acta, Protein Struct. 405 (2) (1975) 442–451, https://doi.org/10.1016/0005-2795(75)90109-9, https://www.sciencedirect.com/science/article/pii/0005279575901099.

[82] D. Chicco, G. Jurman, The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation, BMC Genomics 21 (1) (2020) 6, https://doi.org/10.1186/s12864-019-6413-7.

[83] J. Cohen, A coefficient of agreement for nominal scales, Educ. Psychol. Meas. 20 (1) (1960) 37–46, https://doi.org/10.1177/001316446002000104.

[84] E. Parimbelli, S. Marini, L. Sacchi, R. Bellazzi, Patient similarity for precision medicine: a systematic review, J. Biomed. Inform. 83 (2018) 87–96, https://doi.org/10.1016/j.jbi.2018.06.001, https://www.sciencedirect.com/science/article/pii/S1532046418301072.

[85] S. Pai, G.D. Bader, Patient similarity networks for precision medicine, in: Theory and Application of Network Biology Toward Precision Medicine, J. Mol. Biol. 430 (18, Part A) (2018) 2924–2938, https://doi.org/10.1016/j.jmb.2018.05.037, https://www.sciencedirect.com/science/article/pii/S0022283618305321.

[86] L. Wei, Y. Liu, I. Dubchak, J. Shon, J. Park, Comparative genomics approaches to study organism similarities and differences, J. Biomed. Inform. 35 (2) (2002) 142–150, https://doi.org/10.1016/S1532-0464(02)00506-3, https://www.sciencedirect.com/science/article/pii/S1532046402005063.

[87] G. Kaya, C. Ezekannagha, D. Heider, G. Hattab, Context-aware phylogenetic trees for phylogeny-based taxonomy visualization, Front. Genet. 13 (2022), https://doi.org/10.3389/fgene.2022.891240, https://www.frontiersin.org/articles/10.3389/fgene.2022.891240.

[88] D. Arthur, S. Vassilvitskii, K-means++: the advantages of careful seeding, in: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '07, Society for Industrial and Applied Mathematics, USA, 2007, pp. 1027–1035.

[89] M. Ankerst, M.M. Breunig, H.-P. Kriegel, J. Sander, Optics: ordering points to identify the clustering structure, in: Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data, SIGMOD '99, Association for Computing Machinery, New York, NY, USA, 1999, pp. 49–60.

[90] T. Zhang, R. Ramakrishnan, M. Livny Birch, An efficient data clustering method for very large databases, SIGMOD Rec. 25 (2) (1996) 103–114, https://doi.org/10.1145/235968.233324.

[91] M.E. Tipping, C.M. Bishop, Mixtures of probabilistic principal component analyzers, Neural Comput. 11 (2) (1999) 443–482, https://doi.org/10.1162/089976699300016728, arXiv: https://direct.mit.edu/neco/article-pdf/11/2/443/814064/089976699300016728.pdf.

[92] M.A.A. Cox, T.F. Cox, Multidimensional scaling, in: Multidimensional Scaling, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, pp. 315–347.

[93] L. van der Maaten, G. Hinton, Visualizing high-dimensional data using t-sne, J. Mach. Learn. Res. 9 (Nov 2008) 2579–2605, pagination: 27.

[94] M. Meilă, Comparing clusterings—an information based distance, J. Multivar. Anal. 98 (5) (2007) 873–895, https://doi.org/10.1016/j.jmva.2006.11.013, https://www.sciencedirect.com/science/article/pii/S0047259X06002016.

[95] M. Meilă, Comparing clusterings: an axiomatic view, in: Proceedings of the 22nd International Conference on Machine Learning, 2005, pp. 577–584.

[96] M. Meilă, Comparing clusterings by the variation of information, in: Learning Theory and Kernel Machines, 2003, pp. 173–187.

[97] D. Gil, J.L. Girela, J. De Juan, M.J. Gomez-Torres, M. Johnsson, Predicting seminal quality with artificial intelligence methods, Expert Syst. Appl. 39 (16) (2012) 12564–12573, https://doi.org/10.1016/j.eswa.2012.05.028, https://www.sciencedirect.com/science/article/pii/S0957417412007269.

[98] G. Hoffmann, A. Bietenbeck, R. Lichtinghagen, F. Klawonn, Using machine learning techniques to generate laboratory diagnostic pathways—a case study, J. Lab. Precis. Med. 3 (2018) 58, https://doi.org/10.21037/jlpm.2018.06.01.

[99] T. Caliński, J. Harabasz, A dendrite method for cluster analysis, Commun. Stat. 3 (1) (1974) 1–27, https://doi.org/10.1080/03610927408827101, arXiv: https://www.tandfonline.com/doi/pdf/10.1080/03610927408827101, https://www.tandfonline.com/doi/abs/10.1080/03610927408827101.

[100] L. MacDonald, Using color effectively in computer graphics, IEEE Comput. Graph. Appl. 19 (4) (1999) 20–35, https://doi.org/10.1109/38.773961.

[101] S. Silva, B. Sousa Santos, J. Madeira, Using color in visualization: a survey, in: Virtual Reality in Brazil Visual Computing in Biology and Medicine Semantic 3D Media and Content Cultural Heritage, Comput. Graph. 35 (2) (2011) 320–333, https://doi.org/10.1016/j.cag.2010.11.015, https://www.sciencedirect.com/science/article/pii/S0097849310001846.

[102] S. Haroz, D. Whitney, How capacity limits of attention influence information visualization effectiveness, IEEE Trans. Vis. Comput. Graph. 18 (12) (2012) 2402–2410, https://doi.org/10.1109/TVCG.2012.233.

[103] G. Hattab, T.-M. Rhyne, D. Heider, Ten simple rules to colorize biological data visualization, PLoS Comput. Biol. 16 (10) (2020) 1–18, https://doi.org/10.1371/journal.pcbi.1008259.

[104] C.C. Gramazio, D.H. Laidlaw, K.B. Schloss, Colorgorical: creating discriminable and preferable color palettes for information visualization, IEEE Trans. Vis. Comput. Graph. 23 (1) (2017) 521–530, https://doi.org/10.1109/TVCG.2016.2598918.

[105] C. Ware, Chapter four - color, in: C. Ware (Ed.), Information Visualization, third edition, Interactive Technologies, Morgan Kaufmann, Boston, 2013, pp. 95–138, https://www.sciencedirect.com/science/article/pii/B9780123814647000041.

[106] M. Takatsuka, J. Zhou, Automatic transfer function generation using contour tree controlled residue flow model and color harmonics, IEEE Trans. Vis. Comput. Graph. 15 (06) (2009) 1481–1488, https://doi.org/10.1109/TVCG.2009.120.

[107] C. van Onzenoodt, A. Huckauf, T. Ropinski, On the perceptual influence of shape overlap on data-comparison using scatterplots, Comput. Graph. 90 (2020) 169–181, https://doi.org/10.1016/j.cag.2020.05.028, https://www.sciencedirect.com/science/article/pii/S0097849320300881.

[108] D.A. Szafir, Modeling color difference for visualization design, IEEE Trans. Vis. Comput. Graph. 24 (1) (2018) 392–401, https://doi.org/10.1109/TVCG.2017.2744359.

[109] N.X. Vinh, J. Epps, J. Bailey, Information theoretic measures for clusterings comparison: variants, properties, normalization and correction for chance, J. Mach. Learn. Res. 11 (2010) 2837–2854.

[110] E.T. Jaynes, Information theory and statistical mechanics (notes by the lecturer), Stat. Phys. 3 (1963) 181.

[111] J.K. Jolliff, J.C. Kindle, I. Shulman, B. Penta, M.A. Friedrichs, R. Helber, R.A. Arnone, Summary diagrams for coupled hydrodynamic-ecosystem model skill assessment, in: Skill Assessment for Coupled Biological/Physical Models of Marine Systems, J. Mar. Syst. 76 (1) (2009) 64–82, https://doi.org/10.1016/j.jmarsys.2008.05.014, https://www.sciencedirect.com/science/article/pii/S0924796308001140.

[112] J. Chang, S. Hanna, Air quality model performance evaluation, Meteorol. Atmos. Phys. 87 (2004) 167–196, https://doi.org/10.1007/s00703-003-0070-7.