A Radial Visualisation for Model Comparison and Feature Identification

Jianlong Zhou*
Data Science Institute
University of Technology Sydney
Australia

Weidong Huang†
Faculty of Transdisciplinary Innovation
University of Technology Sydney
Australia

Fang Chen‡
Data Science Institute
University of Technology Sydney
Australia

ABSTRACT

Machine Learning (ML) plays a key role in various intelligent systems, and building an effective ML model for a data set is a difficult task involving various steps including data cleaning, feature definition and extraction, ML algorithms development, model training and evaluation as well as others. One of the most important steps in the process is to compare generated substantial amounts of ML models to find the optimal one for the deployment. It is challenging to compare such models with dynamic number of features. This paper proposes a novel visualisation approach based on a radial net to compare ML models trained with a different number of features of a given data set while revealing implicit dependent relations. In the proposed approach, ML models and features are represented by lines and arcs respectively. The dependence of ML models with dynamic number of features is encoded into the structure of visualisation, where ML models and their dependent features are directly revealed from related line connections. ML model performance information is encoded with colour and line width in the innovative visualisation. Together with the structure of visualization, feature importance can be directly discerned to help to understand ML models.

Keywords: Machine learning, performance, comparison, visualisation

1 INTRODUCTION

We have been going through the digital age with the rapid increase of data from various fields such as infrastructure, transport, energy, health, education, telecommunications, and finance. Together with the dramatic advances in Machine Learning (ML), getting insights from these "Big Data" and data analytics-driven solutions are increasingly in demand for different purposes. While these "Big Data" are used by sophisticated machine learning algorithms to train ML models which are then evaluated by various metrics such as accuracy. Building an effective ML model for a data set is a difficult task involving various steps including data cleaning, feature definition and extraction, ML algorithms development, model training and evaluation as well as others [14]. The generated substantial amounts of ML models must be compared by the engineering designers and analysts to find the optimal one for the deployment [14]. Fig. 1 shows a typical pipeline that processes data to find an optimal ML model. Taking a data set with multiple features for ML training as an example, multiple features can be grouped differently as the input for a ML algorithm to train different ML models. For example, if a data set has three features of F1, F2, and F3, these features may have seven different groups: [F1], [F2], [F3], [F1, F2], [F1, F3], [F2, F3], and [F1, F2, F3]. Each feature group can be used as the input for a ML algorithm to train a ML model, thereby obtaining seven different ML models. It is a common thread to find the best/worst model by comparing such models, however it is often challenging when having a large number of features. Furthermore, comparison is more than just finding differences of ML model performance, users are also interested in the relations between features and model performance from comparison, for example, to find which features result in high performance of ML models, and those features are referred as high important features, or vice versa. This is because the identification of the most or least important features are the key steps for feature engineering in effective machine learning [14].

Bar chart, radar chart, line chart as well as others [1] are commonly used visualisation methods in machine learning to compare different variables. However, comparison of ML models with a large number of features is still considered challenging with the aid of these commonly used visualisations: the items for comparison and the relationships between them can be highly complicated. While these commonly used visualisation approaches not only cause information clutters for large number of visual elements (e.g. bars, dots, lines) but also miss relation information between features and models. It is also very difficult for users to differentiate differences of various model performances with these commonly used visualisation approaches. Parallel coordinates are a common way that is often used to visualise and compare multi-attribute data [16]. However, it still has similar challenges as bar chart, radar chart and line chart have as mentioned above when visualising and comparing ML models with different feature combinations. Despite the specific focus on visualising comparison in recent studies [5, 7, 8], little work has been done on the visual comparison of ML models while identifying relations between features and ML models (e.g. the most and least important features). We explore an approach based on the structure of visualisation in addressing challenges of comparison ML models with dynamic number of features: while height information of bars and lines in commonly used visualisation approaches only encode one-dimensional information in a 2-dimensional (2D) space, it is possible to encode ML model information in other dimensions of the space. If both visual elements and structure of visualisation can be used to encode information of ML models, insights about ML models could be automatically generated, users would not have to inspect every model to find optimal one or conduct complex calculations [12] to estimate feature importance.

In this paper, we propose *RadialChart*, a novel visualisation approach to compare ML models with different number of features while revealing implicit dependent relations. In RadialChart, ML models and features are represented by lines and arcs respectively (an arc also represents the model based on the single feature of arc). The challenge of revealing dependence of ML models with dynamic number of features is addressed by encoding such information into the structure of visualisation, where ML models and their dependent features are directly revealed from related line connections. These lines are defined using a recursive function to generate them effectively. ML model performance information is encoded with colour and line width in RadialChart. It simplifies the comparison of different ML models based on these visual encoding. Moreover, together with the structure of visualisation, feature importance can be directly discerned in RadialChart. RadialChart uses a concept of feature path for ML model lines to avoid a large number of line

*e-mail: jianlong.zhou@uts.edu.au
†e-mail: weidong.huang@uts.edu.au
‡e-mail: fang.chen@uts.edu.au

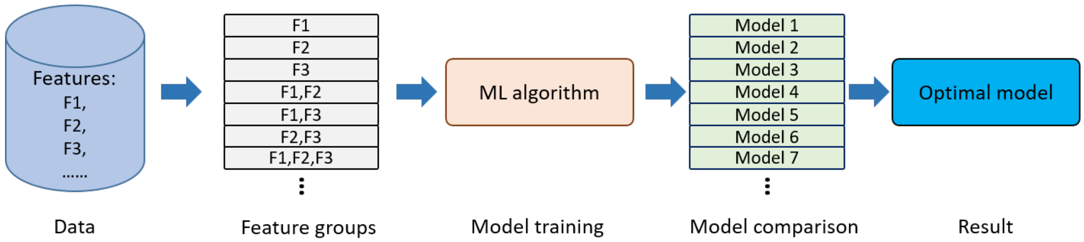

Figure 1: The pipeline of getting an optimal ML model for a data set with multiple features.

entangles. And when visual elements for ML models are crowded, RadialChart allows to interactively change spanning space that RadialChart covers to dynamically control the visual complexities. The results show that RadialChart has advantages in identifying features related to specific models as well as directly revealing importance of features. Furthermore, RadialChart is efficient to help users focus their attention to find visual elements of interest. It is compact to show more information in a limited space.

2 Background and Related Work

In machine learning, given a fixed number of features, it is possible to use different features and their groups to train machine learning algorithms resulting in various machine learning models. Users need to compare these models to find the optimal one for their tasks. Getting the optimal results out of machine learning models requires a truly understanding of all models. However, each data set with large number of features can have hundreds or even thousands of ML models, making it nearly impossible to understand all models based on different feature groups in an intuitive fashion. Visualisation can be used to help unlock nuances and insights in ML models.

2.1 Visualisation of Multi-Attribute Data

The comparison visualisation of machine learning models is related to multi-attribute (or multiple features) data visualisation. The visualisation of multi-attribute data has been frequently investigated for years. One of classical approaches to visualise multi-attribute data points is parallel coordinates [16]. The advantage of this technique is that it can provide an overview of data trend. One of obvious disadvantages of parallel coordinates is that it lacks a tabular view for presenting value details of each coordinates. SimulSort [6] organizes different attributes of data in a tabular and sorts all of the attribute columns simultaneously. However, users still need laborious interactions in SimulSort in order to highlight different points for comparison. Zhou et al. [13] proposed an approach for presenting multi-attribute data by combining advantages of both parallel coordinates and SimulSort, which organizes various attributes in a tabular-like form implicitly. Colours are used to encode data belonging to different groups, instead of highlighting attributes of one point at a time as in SimulSort. Such colour encoding approach provides an overview of points and their associated attribute details to improve the information browsing efficiency. Motivated by such colour encoding, this paper uses colours to encode ML model performance to provide an overview of performance for comparison. However, such visualisation cannot reveal complex relations between machine learning models and their dependent features with dynamic numbers. Draper et al. [3] made a survey on radial visualisation, but there are still challenges when applied to machine learning results comparisons.

2.2 Comparison Visualisation

Supporting comparison is a common challenge in visualisation. Gleicher [5] categorized four considerations that abstract comparison when using visualisation. These four considerations include to identify: the comparative elements, the comparative challenges, a comparative strategy, and a comparative design, which provide a guideline for developing comparison solutions in visualisation. Law et al. [7] presented Duet, a visual analysis system to conduct pairwise comparisons. Duet employs minimal specification in comparison by only recommending similar and different attributes between them when one object group to be compared is specified.

Bar chart is one of commonly used visualisation methods for comparison in machine learning [1]. It works with two variables – one is the length of the bar on one axis and the second is the position of this bar on another axis. The variable is compared by denoting it with the length of the bars when various bars are plotted together. Radar Chart is another commonly used approach to compare multiple quantitative variables. It is useful for seeing which variables have similar values or if there are any outliers amongst the values of each variable. It can also help to find which variables are high or low. Besides, other methods such as line chart and ring chart are also used in comparison. Ondov et al. [8] made evaluations of comparison visualizations of 5 layouts: stacked small multiples, adjacent small multiples, overlaid charts, adjacent small multiples that are mirror symmetric and animated transitions. The data to be compared are encoded with the length of bars in bar charts, slop of lines in line charts, and angle of arcs in donut charts. Advanced visualizations are also developed to compare ML models. For example, Zhang et al. [11] proposed a framework to compare model pairs with local feature importance views.

These previous works provide significant guidelines and advances in comparison visualisation. This paper proposes a new visualisation method for machine learning model comparison with a full consideration of four aspects as categorized in [5]. The new visualisation approach is evaluated by comparing it with other three commonly used visualisation methods (bar chart, line chart, and radar chart) in machine learning model comparisons.

3 RadialChart

This section presents a novel visualisation approach called *RadialChart* to compare machine learning models trained with different feature groups of a data set.

3.1 Design Goals

After having a thorough survey with interviews with experienced researchers and developers in machine learning from the Data Science Institute at University of Technology Sydney on their problems meeting in comparing machine learning models, we phrase following design goals for the RadialChart:

- **Comparison**: To maximise differences among visual elements of models to help users find the optimal target easily. The comparison is the core objective in the ML model visualisation. This is a challenge when substantial amounts of ML models must be compared.
- **Importance**: To easily identify importance of features directly from visualisation. The importance of features plays significant

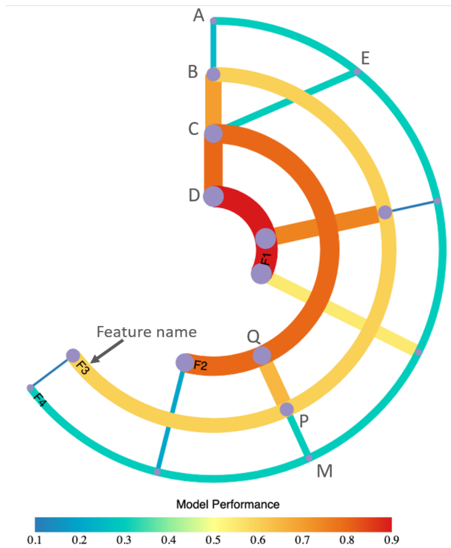

Figure 2: Definition of RadialChart: it is composed of feature points, feature paths, feature arcs, and model lines.

roles in the feature selection in the machine learning pipeline. It is a challenge to identify importance of features directly from visualisation without complex feature importance calculations.

- **Feature identification**: To easily identify relationships between models (and model performance) and their dependent features. This helps users easily link ML models and their dependent features for understanding both features and models, which is usually challenging with commonly used visualisation approaches.
- **Compactness**: To represent complex visualisation in a compact form and reduce the visual clutters because of substantial amounts of information in a limited space.

3.2 Definition of RadialChart

This subsection defines the RadialChart. Fig. 2 shows an example of RadialChart. Based on this example, we firstly give following definitions that are used to set up a RadialChart:

Feature arc Each feature is represented by a concentric arc in RadialChart. The arc is also called feature arc. The name of each feature is displayed at one end of the arc as shown in Fig. 2 (e.g. F1, F2, F3, F4). Each arc also represents the ML model based on that single feature.

Model line RadialChart uses a line segment to represent an ML model based on multiple features. The line is also called model line. For example, in Fig. 2, the line *AB*, *BC*, and *CD* represent different ML models respectively. The features used for the model are defined based on the feature path of the line (see the definition of feature path below).

Feature point A feature point refers to an intersection point of a model line with an arc. It is represented by a dot point on a feature arc as shown in Fig. 2 (e.g. feature points A, B, C).

Feature path A feature path defines features used for a model line. A feature path starts from the feature point of a model line on its outermost arc and ends at the feature point on the innermost arc it can reach through the connected feature point in the RadialChart. For example, in Fig. 2, for the model line *AB*, its feature path starts from the feature point A on the arc F4, passes through B and C, and ends at D on the innermost arc F1. This path can be represented by a list of features corresponding to arcs of each feature point, i.e. the feature path of *AB* is [F4, F3, F2, F1]. Similarly, the feature path of *BC* is [F3, F2, F1], etc.

Furthermore, the model performance is encoded using two methods: the width of the line/arc and the colour of the line/arc. The wider the line/arc is, the higher the model performance. A colour scale is accompanied with the RadialChart to encode model performance and let users easily perceive the difference of performance of different models as shown in Fig. 2. The double encoding with the line width and colour enhances the perception of differences in the model performance comparison. In this study, the rainbow colour map is used. Other colour schemes such as harmonic colours [15] can also be used in the RadialChart. The encoding with the colour scale is set to optional so that users can turn it on/off.

Based on these definitions, the visualisation of lines and arcs are spiraling from the centre to outside and therefore it is called *RadialChart*. The RadialChart has different advantages. For example, given a data set in machine learning, if most of ML models related to one specific feature show high model performance, that feature can be considered as a high important feature, and vice versa if most of ML models related to one specific feature show low model performance, that feature can be considered as a less important feature. The RadialChart can depict importance of features directly through visualisation: if an arc and its connected lines are mostly wider than others and have colours representing high performance values in the colour scale, the feature represented by the arc is an important feature, and vice versa it can also depict less important features. For example, in Fig. 2, the feature F1 is an important feature because the width and colour of the arc as well as its connected lines are mostly wider and red, while the feature F4 is an less important feature. The RadialChart also helps users directly identify features used for a specific model because of the feature path mechanism in RadialChart. Fig. 3 shows the steps used to draw a RadialChart. The definition of different parameters is the key during RadialChart drawing. Firstly, key parameters are defined with user interactions or predefined approaches. Arc parameters and line parameters are then generated based on key parameters. The RadialChart is drawn finally based on generated parameters.

3.3 Key Parameter Initialization

The key parameters include the overall spanning angle of RadialChart, the overall number of models given the number of features, the size of the drawing canvas, as well as others. The overall spanning angle defines the space that the RadialChart covers in degrees. It can be interactively modulated by users to control the compactness of the visualisation in a limited space. If the number of ML models to be visualized is low, a small value can be defined for the spanning angle, and vice versa a large value can be defined for the spanning angle in order to help users easily control and compare ML models in a limited space. Given N features of a data set, F1, F2, ..., FN, a machine learning algorithm uses these features to set up ML models. The ML models can be set up based on one or multiple features of the data set. Typically, the number of models based on various groups of N features can be got from Equation 1:

$$C_N = C_N^1 + C_N^2 + ... + C_N^i + ... + C_N^N = 2^N - 1 \tag{1}$$

where C_N is the number of models based on groups of N features, C_N^i is the group number of selecting i features from N features. It shows that the number of ML models is increased exponentially with the increase of number of features.

3.4 Arc and Line Parameter Generations

The arc is denoted by its start point and end point in polar coordinates. Our algorithm generates arc parameters aiming to make N

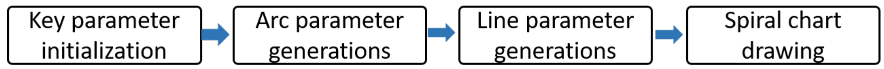

Figure 3: The steps for drawing RadialChart.

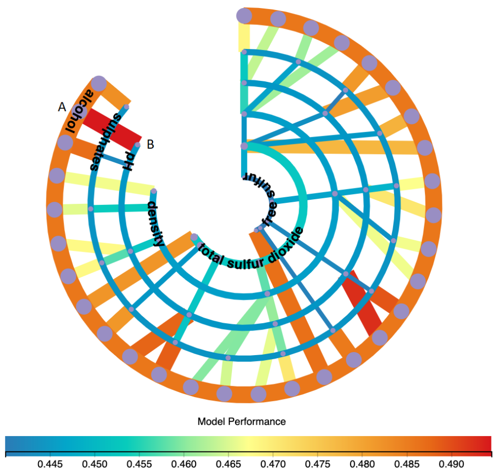

Figure 4: RadialChart of ML models based on a data set with 6 features.

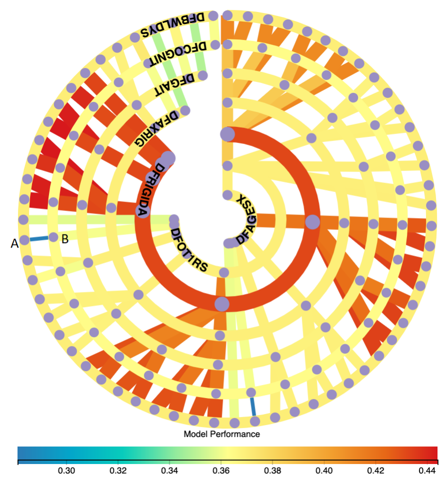

Figure 5: RadialChart of ML models based on a data set with 7 features.

arcs evenly distributed in the drawing canvas space. The algorithm initialises the spanning angle of each arc, and it is dynamically updated to allow arcs in a spiral format. Parameters of arcs are stored in a dictionary and the key of the dictionary is the individual features for the arc. The parameters include radius, spanning angle and width of arcs.

The line is denoted by its start point and end point in polar coordinates. We propose a recursive function for generating model line parameters. In our algorithm, a dictionary is used to store parameters of lines, and the key of the dictionary is the feature list (feature path) used for the line. The line parameters stored in the dictionary include the start and end points of the line in polar coordinates as well as line width of the line. In this algorithm, if the key with the current line features does not exist in the line dictionary, a sub-key with the feature list by removing the last feature in the feature list is created. The algorithm recursively call the function with the current sub-key features to draw lines. The line width is encoded with the model performance based on line features. The colour of the line is also encoded with the model performance using a colour scale.

4 Case Studies

In this section, RadialChart is used to visualise machine learning models based on different data sets and ML algorithms. Two data sets from UCI machine learning data repository [4] and PPMI [9] respectively were analyzed for classification problems (wine quality classification and Parkinson's disease classification respectively), and three machine learning algorithms of K-Nearest Neighbours (KNN), Naïve Bayes (NB) and Random Forest (RF) were deployed in the experiment. Fig. 4 shows the visualisation of different ML models for a data set with 6 features. From this visualisation, we can easily locate the model with the highest performance (the widest red line *AB* as shown in Fig. 4) as well as features (two features of "alcohol" and "pH" on the feature path of the line) used for the model training for classifications. It also helps users easily identify the importance of features, the most important feature "alcohol" is represented by the outermost arc (the arc and its connected lines are mostly redder and wider than others) and the least important feature "free suffur" is represented by the innermost arc (the arc and its connected lines are mostly bluer and narrower than others). Fig. 5 shows the visualisation of different ML models for a data set with 7 features. Compared with Fig. 4, the model number is increased dramatically when the feature number is increased just one. This visualisation also helps users easily locate the model with the lowest performance (the narrowest blue line *AB* as shown in Fig. 5). We can also easily directly identify the most important feature (the third inner arc represented by the widest red arc) and the least important feature (the innermost narrowest yellow arc).

Besides comparison of feature importance of a data in RadialChart, it can also be used to compare performance of different ML algorithms for a given data set. Fig. 6 shows the comparison of three ML algorithms for the same data set with RadialChart visualisation. From this figure, we can easily get that the ML algorithm represented by the left diagram shows the worst performance, compared to algorithms represented by the other two diagrams, because its colour is bluer which is located on the left side of the colour scale. While the algorithm represented by the middle diagram shows the best performance because its colour is redder which is located on the right side of the colour scale. Furthermore, the visualisation shows that the feature represented by the outermost arc (i.e. the feature of "alcohol") is the most important feature because this arc is the widest and its colour is located on the right side of the colour scale in all three visualizations.

In summary, this study proposed a novel visualisation approach to compare variables with different number of dependents. Data information is encoded with colour, line width, and structure of visualisation to reveal insights from data. The results showed that RadialChart has advantages in identifying features related to specific models as well as directly revealing importance of features. Different from conventional feature importance evaluations based on complex computing algorithms [12] (such as by simulating lack of knowledge about the values of the feature(s) [10], or by mean decrease impurity, which is defined as the total decrease in node impurity averaged over all trees of the ensemble in Random For-

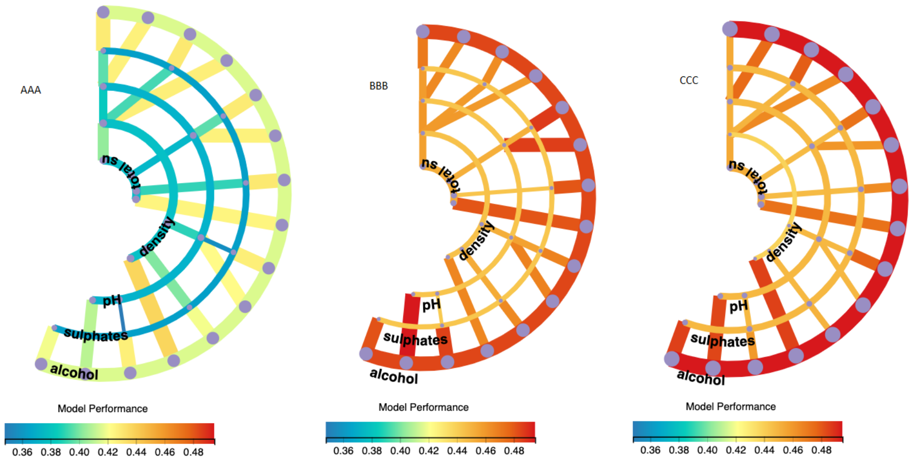

Figure 6: Comparison of three ML algorithms for the same data set with RadialChart.

est [2]), RadialChart allows users to estimate feature importance directly from visualisation by checking lines connected to the feature arc. The consistent large line width of these lines and colours with the colour scale indicate the high importance of the feature to the modelling. RadialChart is compact to show more information in a limited space. And the compactness can be controlled by changing its spanning angle dynamically.

The main limitations of the RadialChart are as the following. RadialChart will be much complex when the number of features is high. This could be compensated with large scale visualisation facilities. The focus+context approach could also be used for this issue. The side-by-side views are used to compare different ML models, which affects the comparison with large number of ML models. A 3D RadialChart with multiple layers could be used to scale the number of ML models flexibly in the comparison.

5 Conclusion and Future Work

This paper presented *RadialChart*, a novel visualisation approach to compare ML models with different number of features while revealing implicit dependent relations. The RadialChart is developed to address the challenges faced in comparing a large amount of ML models with each dependent on a dynamic number of features. It is implemented by representing ML models and features with lines and arcs respectively, which in turn are generated by a recursive function and a feature path concept. We presented our design criteria and described the algorithms for generating the chart. Our case studies showed that the proposed visualisation can help users easily locate target models and important features. Our research provides an effective visualisation approach to represent data with complex relations. It is specifically helpful for users to find optimal machine learning model and discern feature importance visually and directly, but not through complex algorithmic calculations.

For future work, we plan to compare RadialChart with other visualisation approaches used in ML and conduct a comprehensive eye tracking study, focusing on the analysis of eye activities during the task time to learn eye movement patterns in viewing RadialChart. Such studies will help to improve the design of RadialChart for more efficient information browsing. We also hope that the use of RadialChart in ML model comparison presented here can serve as both a template and a motivation for other data and applications.

References

[1] W. Aigner, S. Miksch, H. Schumann, and C. Tominski. *Visualization of Time-Oriented Data*. Human-Computer Interaction Series. Springer, 2011.

[2] L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Wadsworth and Brooks, Monterey, CA, 1984.

[3] G. M. Draper, Y. Livnat, and R. F. Riesenfeld. A survey of radial methods for information visualization. 15(5):759–776, 2009.

[4] D. Dua and E. Karra Taniskidou. UCI machine learning repository, 2017.

[5] M. Gleicher. Considerations for visualizing comparison. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):413–423, 2018.

[6] S. Kim, Z. Dong, H. Xian, B. Upatising, and J. S. Yi. Does an eye tracker tell the truth about visualizations?: Findings while investigating visualizations for decision making. *IEEE Trans. Vis. Comput. Graph.*, 18(12):2421–2430, 2012.

[7] P. Law, R. C. Basole, and Y. Wu. Duet: Helping data analysis novices conduct pairwise comparisons by minimal specification. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):427–437, 2019.

[8] B. D. Ondov, N. Jardine, N. Elmqvist, and S. Franconeri. Face to face: Evaluating visual comparison. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):861–871, 2019.

[9] N. Prakash, C. Caspell-Garcia, C. Coffey, and et al. Feasibility and safety of lumbar puncture in the parkinson's disease research participants: Parkinson's progression marker initiative (PPMI). *Parkinsonism & Related Disorders*, 2019.

[10] M. Robnik-Sikonja, I. Kononenko, and E. Strumbelj. Quality of Classification Explanations with PRBF. *Neurocomput.*, 96:37–46, Nov. 2012.

[11] J. Zhang, Y. Wang, P. Molino, L. Li, and D. S. Ebert. Manifold: A model-agnostic framework for interpretation and diagnosis of machine learning models. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):364–373, 2019.

[12] J. Zhou and F. Chen, eds. *Human and Machine Learning: Visible, Explainable, Trustworthy and Transparent*. Springer, Cham, 2018.

[13] J. Zhou, J. Sun, F. Chen, Y. Wang, R. Taib, A. Khawaji, and Z. Li. Measurable Decision Making with GSR and Pupillary Analysis for Intelligent User Interface. *ACM Transactions on Computer-Human Interaction*, 21(6):33, 2015.

[14] J. Zhou, J. Sun, Y. Wang, and F. Chen. Wrapping practical problems into a machine learning framework: Using water pipe failure prediction as a case study. *International Journal of Intelligent Systems Technologies and Applications*, 16(3):191–207, 2017.

[15] J. Zhou and M. Takatsuka. Automatic transfer function generation using contour tree controlled residue flow model and color harmonics. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):14811488, Nov. 2009.

[16] J. Zhou, C. Xiao, Z. Wang, and M. Takatsuka. A concept of volume rendering guided search process to analyze medical data set. *Computerized Medical Imaging and Graphics*, 32(2):140–149, 2008.