3D-RadViz: Three Dimensional Radial Visualization for Large-Scale Data Visualization

Abdelrahman Elewah, Abeer A.Badawi, Haytham Khalil, Shahryar Rahnamayan, Khalid Elgazzar
Department of Electrical, Computer and Software Engineering, Ontario Tech University
Oshawa, ON, Canada
{abdelrahman.elewah, abeer.badawi, haytham.mohamedabdelaal, Shahryar.Rahnamayan, khalid.elgazzar}@ontariotechu.ca

2021 IEEE Congress on Evolutionary Computation (CEC) | 978-1-7281-8393-0/21/$31.00 ©2021 IEEE | DOI: 10.1109/CEC45853.2021.9504983

***Abstract*—This paper presents 3D-RadViz, a visualization method for high dimensional data using a three dimensional radial visualization technique. The proposed technique extends the capabilities of the classical two dimensional radial visualization (RadViz) in order to reduce overlapping data points. For evaluation purposes, the paper applies the proposed 3D-RadViz alongside with t-SNE and a recently published 3-dimensional radial visualization technique to the same real-world datasets from the UCI machine learning repository. The contribution of the work is an interactive, deterministic, and high-performance library, implemented in Python, that can be utilized to realize better high dimensional data visualization using the proposed 3D-RadViz technique. The paper also suggests some directions for future enhancements to the proposed visualization approach.**

***Index Terms*—Data visualization, two dimensional radial visualization (RadViz), three dimensional radial visualization (3D-RadViz), Magic Ball, RadViz-Plotly, large-scale visualization, high dimensional visualization.**

I. Introduction

Scientists agree that vision is our dominant sense of cognition. Almost 80–85% of the information we perceive, learn, or process is mediated through our vision [1]. Data visualization stands for the representation of information in a pictorial or a graphical arrangement [2]. As a natural trend in the era of big data, most of the recent visualization techniques are targeted towards visualizing high dimensional data [3] [4]. Regarding the visual representation of high dimensional data, the capabilities of the physical display devices and our visual system limit the direct representation of dimensions to three at most [6]. Nevertheless, it is neither practical nor feasible to perform a microscopic analysis for each and every dimension within a high dimensional dataset. That would take a long time to reach a meaningful analysis and likely at a high cost. Essentially, high dimensional data visualization techniques aim to tackle the curse of high dimensionality by devising smart graphical representations to the brain to extract patterns and infer insights from the data through the visual cues.

Although there is no strict classification of high dimensional data visualization techniques [7] [8], the most common approach of classification is based on data transformation. High dimensional visualization techniques can be classified into (a) ***dimension reduction methodologies*** or (b) ***non dimension reduction methodologies*** [9]. The former approach, dimension reduction, selects a principal subset of the original dimensions within a dataset that is easy to visualize and which upon visualizing it, it could retain as much of the information within the original set of dimensions. The dimension reduction approach encompasses a lot of visualization techniques that stick to the same concept. Visualization techniques encircled by the dimension reduction approach include *Principal Component Analysis (PCA)* [10], *Multidimensional Scaling (MDS)* [11], *t-Distributed Stochastic Neighbor Embedding (t-SNE)* [12], and *Uniform Manifold Approximation and Projection (UMAP)* [13]. However, visualization techniques that employ dimension reduction still experience a loss of information within the original dataset, with more loss of information occurs as the number of dimensions increases [14]. While the dimension reduction approach introduces a single embedding of data points in a lower dimensional space that can best describe the structure of the data at the high dimension space, usually referred to as *linear projection*, visualization techniques that don not employ dimension reduction usually provide multiple embeddings based on the clustering of either the attributes or the data points to visualize different aspects of the data, usually referred to as *non-linear projection* [5]. This approach encircles different visualization techniques such as *Parallel Coordinate Plot (PCP)* [15] and *Radial Visualization (RadViz)* [16]. However, these techniques experience drawbacks as the number of attributes within a dataset increases. These drawbacks are often related to records' ambiguities at the resulting visualization, where multiple data points are experiencing the same projection or they are concentrating at a certain region of the visualization space and cannot consequently be identified or distinguished from each other [17]. In order to provide for evaluation metrics, the authors of [18] define metrics to evaluate high dimensional data visualization techniques when displayed on the screen, namely *intrinsic dimension*, *intrinsic record ratio*, and *intrinsic coordinate dimension*.

RadViz is one of the famous techniques for high dimensional data visualization. In RadViz, all the attributes of a dataset are represented by anchor points that are uniformly distributed over the perimeter of a unit circle [19]. Each record is then represented by a point that is linked to each attribute by a spring whose stiffness is proportional to the

978-1-7281-8393-0/21/$31.00 ©2021 IEEE

Authorized licensed use limited to: Ontario Tech University. Downloaded on May 07,2022 at 16:23:18 UTC from IEEE Xplore. Restrictions apply.

value of this attribute for that particular record. The position of the point is defined as the position of equilibrium for all the springs attached to that point. However, the position of dimensional anchors in the unit circle is crucial. Therefore, the best placement of dimensional anchors can be obtained when dimensional anchors that correspond to highly correlated dimensions in the dataset are located closer to each other [20]. Concerning the evaluation metrics of high dimension data visualization techniques mentioned earlier, RadViz is one of the best techniques for high dimensional data visualization regarding the intrinsic dimension and the intrinsic record ratio metrics [18].

In this paper, we present 3D-RadViz, a new technique for high dimensional data visualization that is based on RadViz. The proposed approach reduces the data overlapping of the traditional RadViz by introducing a third dimension projection for the points. We also represent the attributes of the dataset by anchor points that are evenly distributed over the surface of a unit sphere, rather than the perimeter of a unit circle as implemented in RadViz. Further, we developed an algorithm to plot the proposed 3D-RadViz. The source code of the proposed technique is open for public access and is released as interactive python package, called *RadViz-Plotly*, on the Python Package Index (PyPI) [1] for evaluation purposes. To evaluate the proposed visualization scheme, we generate a benchmark of data with predefined statistical properties and apply it to the proposed visualization scheme. To compare the proposed visualization scheme with competitive visualization techniques in the literature, we applied the proposed visualization technique, alongside with comparative techniques in the literature, namely the visualization technique presented in [20] and t-SNE, to five real datasets from UCI Machine Learning Repository datasets to highlight the strength of the proposed 3D-RadViz.

The remainder of the paper is structured as follows. The related work is highlighted in Section II. Section III outlines the methodology and the framework of the proposed visualization scheme. In Section IV, we contrast the proposed visualization scheme with other comparable visualization schemes in the literature and discussed the results in section V. Lastly, we conclude the paper in section VI and propose future directions to furtherly enhance the proposed visualization scheme.

II. Related Work

In order to accommodate the inherent nature of multidimensionality in recent datasets, high dimensional data visualization has emerged as an attractive topic that appeals to a lot of researchers. For example, *Parallel Coordinate Plot (PCP)* is used to visualize high dimensional and multivariate data [15]. It is used to contrast the attributes within the data altogether and visualize the relationship among them using a 2D-plot [15]. However, the arrangement of the axes representing the attributes in PCP affects how a user comprehends the resulting visualization; since the relationships between adjacent attributes are easier to perceive than for non-adjacent attributes [21]. Moreover, as the number of attributes within a dataset increases, the PCP becomes over cluttered and it would then be hard to infer insights from the resulting visualization [22]. Numerous improvements have been devised to alleviate the effect of visual clutter and increase perceptibility for high dimensional data visualization using PCP [22] [23] [24] [25] [25]. Mostly, improvements to PCP are based on clustering or introducing interactivity to the resulting visualization.

Other techniques to visualize high dimensional data are based on dimensional reduction methodologies [31]. Basically, these techniques firstly reduce the number of dimensions in the dataset by obtaining a principal subset of dimensions that preserves as much of the information in the original dataset and then they can be easily visualized [32]. Dimension reduction can be achieved using different approaches, namely, *feature elimination*, *feature selection*, and *feature extraction* [31]. *t-Distributed Stochastic Neighbor Embedding (t-SNE)* is a non-linear technique for dimension reduction that is especially appropriate for visualizing high dimensional datasets [12]. It is broadly applied in image processing, NLP, and speech processing. Nevertheless, t-SNE is a stochastic process that endeavors to extract patterns in the data by identifying classes of points based on their similarities. At this point, the original features in the dataset no longer exist and we cannot make inference based only on the output of t-SNE.

Radial Visualization (RadViz) is another technique to visualize high dimensional data [26] [27] [28]. It generates a 2D plot for high dimensional data using a non-linear mapping of data points onto the area of a unit circle, where the attributes of a dataset are firstly evenly distributed over the perimeter of a unit circle, then a logical spring system model is employed to determine the projection of each data point within the area of the unit circle [16]. However, due to the inherent non-linear mapping of RadViz, some of the information in the original data is lost during visualization [16]. Nevertheless, the loss of information increases as the attributes of a dataset increase. In [29], the authors tried to overcome the limitation of the traditional 2-dimensional RadViz by introducing an enhanced version, denoted by *RadViz++*, which provides interactive exploration for the data visualization. This improvement allowed the authors to effectively visualize a dataset with over a hundred of attributes using conventional 2-dimensional RadViz and reduce the inherent radial layout ambiguities which emerged when visualizing a high dimensional dataset. However, the proposed method still experiences record ambiguity as the dimensions highly increase. In order to decrease the loss effect of the 2-dimensional RadViz, some researchers propose a new method to overcome this issue by using a 3-Dimensional Radial Visualization. Although 2-dimensional RadViz is a mature technique for large-scale data visualization, the concept of 3-dimensional RadViz is relatively new. To the best of our knowledge, there is only one recent publication in [30] that introduces a 3-dimensional RadViz technique to visualize multivariate data. The underlying methodology of the latter which depends on the even distributing the attributes of

[1]RadViz-Plotly package: https://pypi.org/project/RadViz-Plotly/

Authorized licensed use limited to: Ontario Tech University. Downloaded on May 07,2022 at 16:23:18 UTC from IEEE Xplore. Restrictions apply.

the dataset over the surface of a unit sphere is similar to our approach, where the authors use the five platonic solids for the exact uniform distribution of the attributes in the case the number of attributes is equal to specified values. Nevertheless, they used an approximated method like ours for other numbers of attributes, as explained later in the methodology section. However, the methodology for realizing the visualization of the records within the unit sphere is different. The authors of [30] introduced a new term which is called *Maximum-Projection-Ratio*, denoted by *MPR*, to summarize the group information of the dataset in the higher dimensional space and then they use it as a linear projection for the data in the lower dimensional space. The other purpose of this newly introduced term is to also achieve maximum distinctiveness of classes in the lower dimensional space. On the other hand, our approach seems to be an intuitive and natural extension to the mature 2-dimensional RadViz and it can be used to obtain the conventional 2D-RadViz as well, as explained later in the methodology section.

The proposed 3D-RadViz visualization scheme in this paper uses the same concept behind 2-dimensional RadViz which is based on Hooke's law while trying to introduce a third dimension to the representation of the data to realize the 3-dimensional RadViz. The outcome of the work is a reusable library, developed in Python, that can be utilized to get better high dimensional data visualization in a deterministic way. Moreover, the library is interactive and can be employed to furtherly enhance the algorithm. For evaluation purposes of the proposed 3D-Radviz, competitive visualization techniques, namely t-SNE and the 3-dimensional radial visualization proposed in [30], alongside our proposed scheme are all applied to the same sets of data to contrast the differences between them and highlight the key strengths of our proposed scheme, as explained later in section V.

III. Methodology

Fundamentally, in RadViz, the attributes of a dataset are uniformly distributed over the perimeter of a unit circle. In order to project a record of a dataset, a spring is attached to each anchor point whose stiffness is proportional to the value of the attribute. The record is then projected where the point connecting the other side of the springs rest due to equilibrium. To mitigate the effect of record ambiguity in RadViz, we distribute the attributes of a dataset over the surface of a unit sphere rather than the circumference of a unit circle. However, although the problem of evenly distributing the anchor points over the unit circle is easy and straightforward, it is not the case to evenly distributed points, especially with arbitrary numbers, over the surface of a unit sphere.

A. Evenly Distributing Points over the Surface of a Unit Sphere

The difficulty of uniformly distributing points across the surface of the sphere is an important task in both math and computer science fields. The basic point to realize our proposed 3D-RadViz is to be able to uniformly distribute the attributes of a dataset over the surface area of a unit sphere in a deterministic way, such that the visualization obtained for a given dataset is persistent over each run of the algorithm.

It is challenging to precisely distribute points over the surface of a sphere in a uniform way, except for platonic solids. Besides, the answer for this issue is basically reliant on the measures used to pass judgement on the uniformity [36]. One of the elegant solutions to tackle this problem is based upon nodes showing up in nature, for example, the seed conveyance on the top of a sunflower, a phenomenon known as spiral phyllotaxis [36]. These arrangements are shown to be fundamentally related to Fibonacci sequence F_k = 1, 1, 2, 3, 5, 8, 13, ... and the golden Ratio $\phi = (1+\sqrt{5})/2$ [37].

Basically, there are two similar methods in the literature to tackle the problem of uniformly distributed points over the surface of a sphere based on Fibonacci lattice. The first one is well studied in number theory and it is strictly defined for a number of points N equals to one of the terms of the Fibonacci sequence F_k, which is given by:

$$t_i = \left(\frac{i}{F_k}, \frac{iF_{k-1}}{F_k}\right) for \;\; 0 \leq i \leq N-1 \tag{1}$$

On the other hand, the second method to tackle this issue generalizes the number of points N to any arbitrary value. This method is frequently used in computing and it is given by:

$$t_i = \left(\frac{i}{N}, \frac{i}{\phi}\right) for \;\; 0 \leq i \leq N \tag{2}$$

where:

$$\phi = \frac{1+\sqrt{5}}{2} = \lim_{n\to\infty}\left(\frac{F_{n+1}}{F_n}\right) \tag{3}$$

The points set can then be mapped from the unit square to the sphere by employing the cylindrical equal area projection, given by:

$$(x, y) \rightarrow (\theta, \phi) : (cos^{-1}((2x-1) - \frac{\pi}{2}), 2\pi y) \tag{4}$$

$$(\theta, \phi) \rightarrow (x, y, z) : (cos\theta \;\; cos\phi, \;\; cos\theta \;\; sin\phi, \;\; sin\theta) \tag{5}$$

Utilizing the latter method to evenly distribute arbitrary number of points over the surface of sphere, we implemented an algorithm in python to perform this technique, as shown in Fig. 1, which shows a sample run of the python script that uniformly distribute 100 points over a unit sphere in a deterministic way using the golden spiral method.

Using Fibonacci spiral grid concept, we employ an approximate method which generalizes the number of points to be evenly distributed over the unit sphere to an arbitrary number (N) which wouldn't necessarily equal one of the terms of the Fibonacci sequence, F = 1, 1, 2, 3, 5, 8, 13, ... The resulting Fibonacci spiral points are then mapped from the original unit square to the sphere using cylindrical equal area projection method. Although the resulting spherical Fibonacci points are not perfectly evenly distributed, they incorporate excellent

Authorized licensed use limited to: Ontario Tech University. Downloaded on May 07,2022 at 16:23:18 UTC from IEEE Xplore. Restrictions apply.

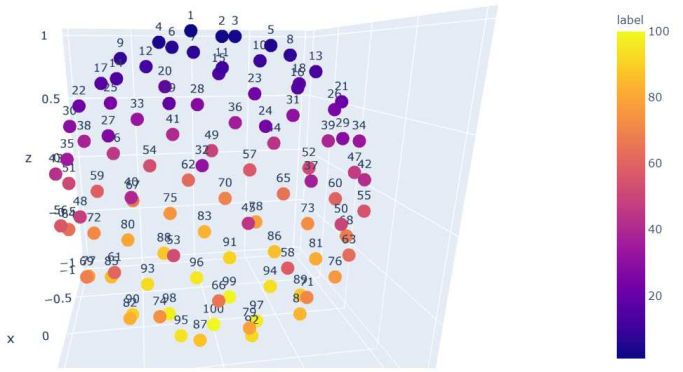


Fig. 1: A sample run of the python script that evenly distributes 100 points over a sphere, showing how the golden spiral method sorts the points.

sampling properties. Moreover, the methodology to realize this approximate method is easy to implement in software, and hence it was implemented by the proposed 3D-RadViz in the paper.

B. The proposed algorithm for 2D and 3D RadViz data visualization

In the current research work , we introduce some formulas that are simpler to use in the 2D or 3D RadViz mapping algorithms. First, let's discuss the 2D RadViz formulas shown in equations 6 and 7 introduced by Sharko [38]. These formulas map the record in the two-dimensional RadViz space.

$$x_i = \frac{\left(\sum_{j=1}^{d} a_{i,j} \cos\theta_j\right)}{\left(\sum_{j=1}^{d} a_{i,j}\right)}, \quad i = 1, 2, 3.....n \tag{6}$$

$$y_i = \frac{\left(\sum_{j=1}^{d} a_{i,j} \sin\theta_j\right)}{\left(\sum_{j=1}^{d} a_{i,j}\right)}, \quad i = 1, 2, 3.....n \tag{7}$$

Where the x_i *and* y_i are the coordinates of record i, in the RadViz space, we assume there are n records. θ_j is the angle between the positive x-axis and the dimension anchor j, we also assume that the number of dimension anchors is d, and $a_{i,j}$ is the value of attribute j for the record i.
These formulas get the resultant vector, which will be the 2D mapped vector by calculating the projection of each value of the anchor in x and y axes, then we divide the result by the sum of attributes' values. A quite simple method to get the projection of a vector is to use the inner (dot) product with the unit vector, for instance, let assume

$$F = 4a_x + 3a_y + 5a_z \tag{8}$$

Then we obtain the projection of F in direction of x

$$F_x = F \cdot a_x \tag{9}$$

From equation 9, F_x is equal to 4, and we assume F in direction of x is equal to 4 in equation 8. Therefore, to get the resultant in the direction of x and y axes as shown in the equations 10 and 11, we have

$$x_i = a_{i,1} \cdot a_{X1_x} + a_{i,2} \cdot a_{X2_x} + ... \; a_{i,d} \cdot a_{Xd_x} \tag{10}$$

$$y_i = a_{i,1} \cdot a_{X1_y} + a_{i,2} \cdot a_{X2_y} + ... \; a_{i,d} \cdot a_{Xd_y} \tag{11}$$

Where $a_{i,1},\ a_{i,2}, ...,\ and\ a_{i,d}$ are the values of the attributes for the record i. $(a_{X1_x},\ a_{X1_y}), (a_{X2_x}, a_{X2_y}), ...\ and\ \ (a_{Xd_x}, a_{Xd_y})$ are the projections pairs of the anchors on x-axis and y-axis. Then, we can represent the equations 6 and 7 by

$$x_i = \frac{\sum_{j=1}^{d} a_{i,j} \cdot a_{X_{j_x}}}{\sum_{j=1}^{d} a_{i,j}}, \quad i = 1, 2, ..., n \tag{12}$$

$$y_i = \frac{\sum_{j=1}^{d} a_{i,j} \cdot a_{X_{j_y}}}{\sum_{j=1}^{d} a_{i,j}}, \quad i = 1, 2, ..., n \tag{13}$$

Another formula to get the coordinates of record in the two-dimensional Radviz space is shown in the next equations. Every dimensional anchor, abbreviated by DA, has a coordinate as shown by the equation 14

$$(x_i, y_i) \tag{14}$$

Equation 15 represents the DAs vector, assuming the dataset contains d attributes.

$$X_i = [X_1,\ X_2, ..., X_d] \tag{15}$$

DAs vector in matrix representation is

$$X_{(d,2)} = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ . & . \\ . & . \\ . & . \\ x_d & y_d \end{bmatrix} \tag{16}$$

Let's refer to the Record (instance) that we want to map in the RadViz space is R_i then the instance vector is shown in the equation 17, and we refer to the mapped result by $\hat{R}_i$ shown in equation 18.

$$R_i = [R_{i1},\ R_{i2}, ...R_{id}] \tag{17}$$

$$\hat{R}_i \ = \frac{R_i \ . \ X_{(d,2)}}{\sum_{j=1}^{d} R_{ij}} \tag{18}$$

Let's refer to the dataset attributes matrix by S and the normalized version by $\overrightarrow{S}$ where the size of S is $(n \times d)$, n is the number of instances and d number of dimensions. Then the mapped version $\hat{S}$ can be represented by:

Authorized licensed use limited to: Ontario Tech University. Downloaded on May 07,2022 at 16:23:18 UTC from IEEE Xplore. Restrictions apply.

$$\hat{S}_{(n,2)} = \frac{\overrightarrow{S}_{(n,d)} \cdot X_{(d,2)}}{sumx_{(n,1)}} \tag{19}$$

Where the $sumx_{(n,1)}$ is the sum vector of the normalized attributes in each instance.

$$sumx_{(n,1)} = \begin{bmatrix} \sum_{i=1}^{d} S_{1,i} \\ \sum_{i=1}^{d} S_{2,i} \\ . \\ . \\ . \\ \sum_{i=1}^{d} S_{n,i} \end{bmatrix} \tag{20}$$

For the 3D RadViz mapping process, all equations are similar to the 2D RadViz mapping process, except that the DAs vector in matrix formula (equation 10), and the mapped S, $\hat{S}$, is (n,3) matrix. DAs vector in matrix formula in 3D Radviz is shown in the equation 21

$$X_{(d,3)} = \begin{bmatrix} x_1 & y_1 & z_1 \\ x_1 & y_2 & z_2 \\ . & . & . \\ . & . & . \\ x_d & y_d & z_d \end{bmatrix} \tag{21}$$

Fig. 2 describes the steps followed to achieve the proposed algorithm of the 3D-RadViz in detail. The first part is data preparation; in this section, we begin with cleaning the data, define our labels and attributes, and finally normalize the characteristics to a new matrix called S. The second part is RadVis mapping; we calculate the anchors coordinates using the golden spiral method to distribute points on a sphere. The third and last step is the visualization, then we prepare the RadViz frame and plot it using the plotly library.

IV. Results and Discussion

In this section, we present the results obtained by implementing our algorithm explained in the previous section. We applied the 2D and 3D RadViz equation to three real-world datasets from UCI Machine Learning Repository datasets, we used the released RadViz Plotly package for this operation. Then, we compare between our method, t-SNE [12], and RadViz3D method [30]. These methods are the state-of-the-art techniques to visualize high dimensional data. The benchmark results and the Jupyter files of the two datasets are hosted on GitHub [1] and linked to Jupyter lab [2].

A. The arrangement of the dimension anchors

In this study, we created a dataset shown in Table 1 to investigate the performance of the proposed visualization method. We use excel sheets to generate this dataset which contains 101 records, 51 and 50, A and B records, respectively. We intentionally created the density of records for labels A and

[1]Benchmark results: https://mybinder.org/v2/gh/elewah/3DRadViz-Results/master

[2]The Jupyter files of the datasets: https://github.com/elewah/RadViz-Plotly-Examples

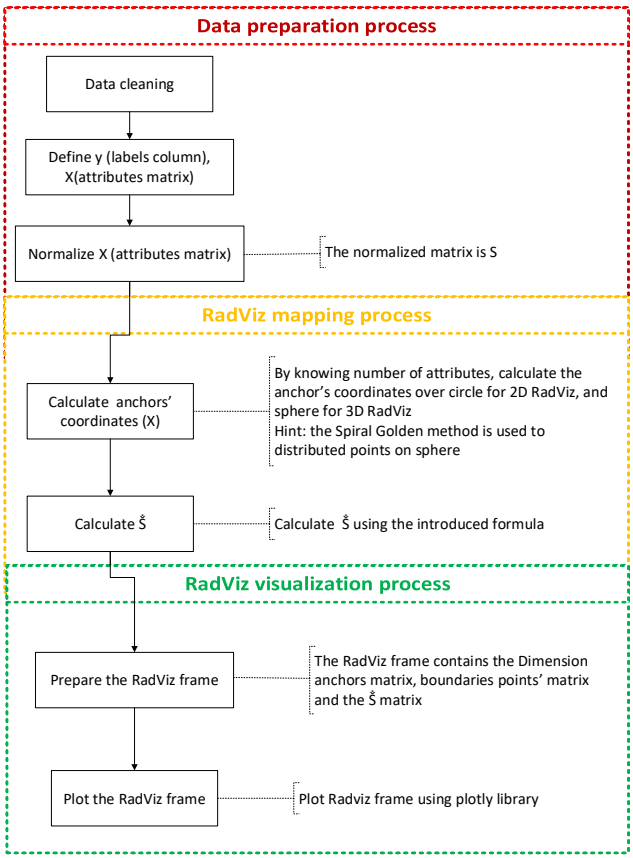


Fig. 2: The flowchart for the proposed 3D-RadViz algorithm.

TABLE I: A&B DATASETS

Dataset	X1	X2	X3	X4
A	random (200,300)	random (50,100)	random (100,250)	random (0,75)
B	random (100,250)	random (0,75)	random (200,300)	random (50,100)

B overlapping in each attribute. A and B clusters are created using an Excel function, RANDBETWEEN(bottom, top), over each attribute.

Fig. 4.a shows that the RadViz method can separate these classes. Fig. 3 shows another way to view these records, A and B, as two clouds, the green and purple cloud, respectively, to explain how the RadViz can separate the different labels in the created A and B dataset. A cloud has greater or equal values to record B on X1 and X2 attributes, and B cloud has greater or equal values to Record A on X3 and X4. The dimension anchors are arranged to secure that the anchors' spring supports the visualization process.

X1 and X2 pull the green cloud, A records, greater than pull the purple cloud B. At the same time, X3 and X4 pull the purple cloud greater than pull the green cloud. However, if we arrange the anchors to make the spring forces district each other, the two clouds will be collected on the origin as

Authorized licensed use limited to: Ontario Tech University. Downloaded on May 07,2022 at 16:23:18 UTC from IEEE Xplore. Restrictions apply.

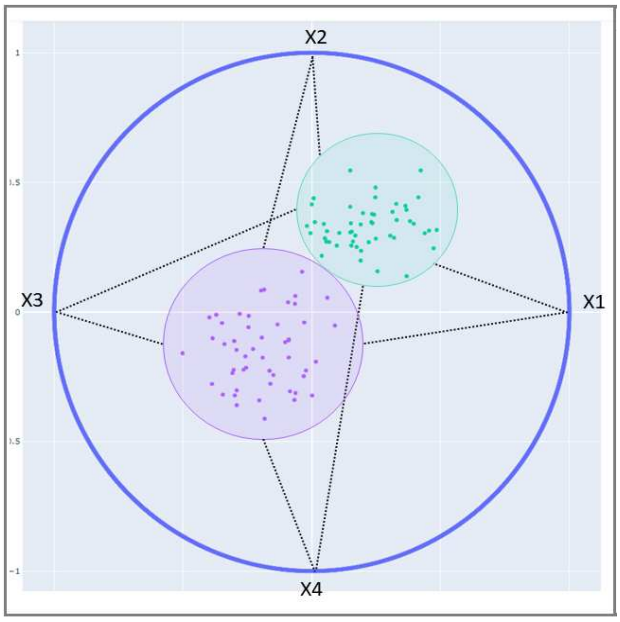


Fig. 3: Illustration of the visualization result.

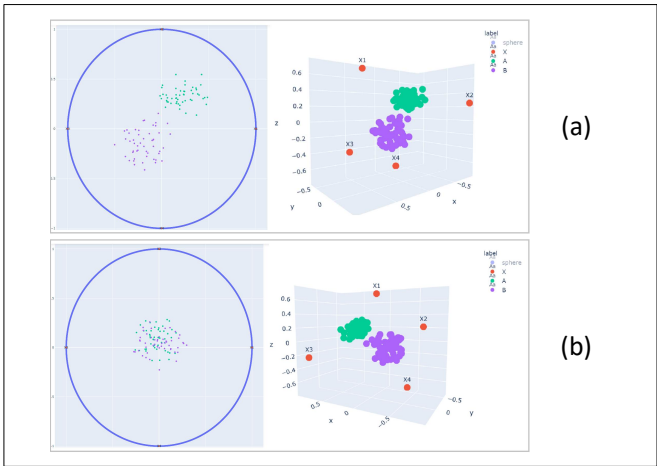


Fig. 4: (a) 2D and 3D Radviz for A&B datasets. (b) RadViz for A&B datasets, enforcing the anchors' force to cancel each other.

shown in Fig. 4.b. Nevertheless, the 3D-RadViz method is still able to separate these clouds as shown since the anchors are not aligned on the same line as in the 2D-RadViz method. In conclusion, the anchors should be arranged according to the correlations between each other as shown in Fig. 5, and the 3D-RadViz can perform better than the 2D-RadViz.

B. Benchmarks

We implemented a python script to generate datasets that simulate the clouds mentioned in the section above. The clouds have the characteristic of the normal distribution curve as shown in Fig. 6. The script's user can decide about the number of labels, attributes that have the same sorting of labels mean, and the distance between the mean of the attributes. The script can generate all possible sorting of labels' mean where the number of possible sortings is equal to the factorial of the

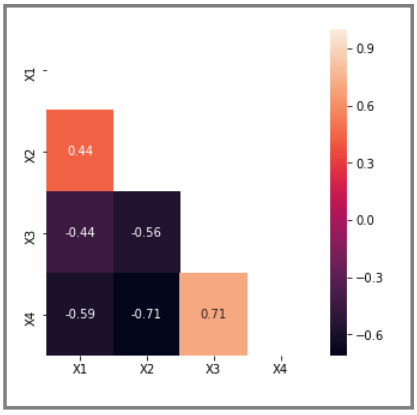


Fig. 5: The correlation matrix of A and B datasets.

number of labels on the dataset. To illustrate the labels' means sorting more clearly, assume a dataset that has three labels, A, B, and C. The number of unique labels' means sorting equal to factorial 3, 6, and the different ways are (A,B,C), (C,B,A), (B,A,C), (C,A,B),(B,C,A), and (A,C,B). That is the number of unique sorting ways equal to the factorial of the number of labels. The explained AB dataset above has two ways of labels' means sorting ((A, B), (B, A)) and two attributes with the same labels' means sorting.

The distributions of labels over each attribute are assumed to follow a normal distribution with equal standard deviation (σ). The distances between the means of the distribution curves are then controlled by the standard deviation of the curves as depicted in Fig. 6. In Fig. 6, the standard deviations of the distribution curves over the attributes are assumed to be equal. The distance between the means of the curves are then assumed to be equal to three, two, one and half times the standard deviations for attributes X1, X2, X3, and X4, respectively. Benchmarks are forty-eight datasets that are used to test the 3D-RadViz.

We published the benchmark results arranged in 4 separate folders, 2-labels, 3-labels,4-labels, and 5-labels [41]. Each folder contains 3D RadViz files for twelve datasets. We divided it into Four subfolders that are named according to the number of the same sorted attribute as we explained previously.

Each label in each attribute has a thousand samples that follow the normal distribution with the standard deviation equal to fifty. Each folder contains three 3D Radviz files for different benchmarks. We choose the file names according to the distance between the label means, 0.5 Sigma, 1 Sigma, and 2 Sigma. The benchmarks explored three parameters that influence the 3D-RadViz's quality in showing clear patterns. The results show that if the dataset has many attributes with the same sorting, the labels would be far from each other. The performance of 3D-RadViz declines against the number of labels. The distance between the groups means over the attributes consider a factor that impacts the 3D-RadViz's results. The smaller distance between the clusters' means has a linear relation with the difficulty to separate these labels.

Authorized licensed use limited to: Ontario Tech University. Downloaded on May 07,2022 at 16:23:18 UTC from IEEE Xplore. Restrictions apply.

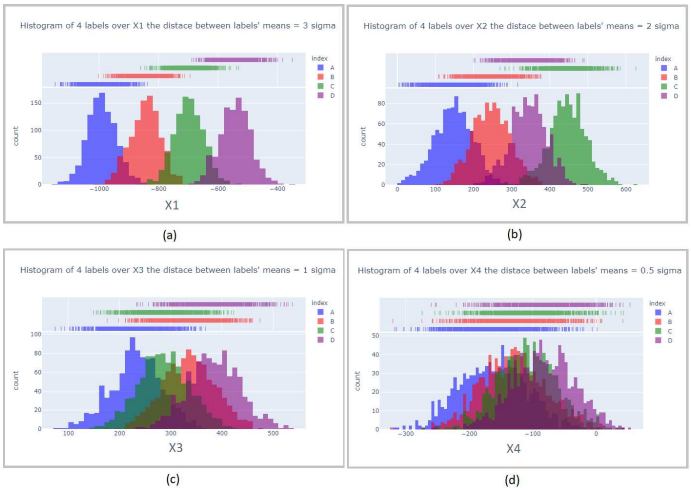


Fig. 6: (a) The distributions of 4 labels over X1 where the distance between the labels' means equals three times sigma. (b) The distributions of 4 labels over X2 where the distance between the labels' means equals two times sigma. (c) The distributions of 4 labels over X3 where the distance between the labels' means equals one sigma. (d) The distributions of 4 labels over X4 where the distance between the labels' means equals 0.5 times sigma.

C. Test cases from UCI ML Repository Datasets

1) ***Iris Data Set****:* The popular Iris flowers dataset [39], consist of 3 classes of 50 instances for each one, each class represents a type of iris plant namely: Iris Setosa, Iris Versicolour, and Iris Virginica. The attributes are sepal length, sepal width, petal length, and petal width. Fig. 7.a shows the 2D RadViz plot using the implemented 2D-Radviz function. Fig. 7.b shows the 3D RadViz plot, we can observe from the 3D plot shown that the Iris Versicolour and Iris Virginica are overlapping, but the Iris Setosa is plotted in different axes. This indicates that the first two types have more common attributes. One of the most powerful advantages of 3D plots, that it clarifies during visualization the difference in classes without using any classification or clustering techniques.

In addition, we plot the iris dataset with two other methods namely; t-SNE and RadViz3D shown in the Fig 7.(c,d) and 7.e, respectively. There are several important differences between our method shown in the Fig. 7.a and the other two methods. Firstly, the key problem with the t-SNE method is shown in the Fig. 7.(c,d). We ran the same code for the iris dataset twice, and we observed a major drawback of that approach. The results were different every time we ran the code. Unlike our method which had consistent outputs regardless of the number of runs of the same code, which makes it more reliable. On the other hand, the RadViz3D method is shown in the Fig. 7.e. This method and our method share a number of key features such as plotting the figure in the 3D plot with the same position of the labels, also the labels and attributes displayed clearly in the figure, unlike the t-SNE method.

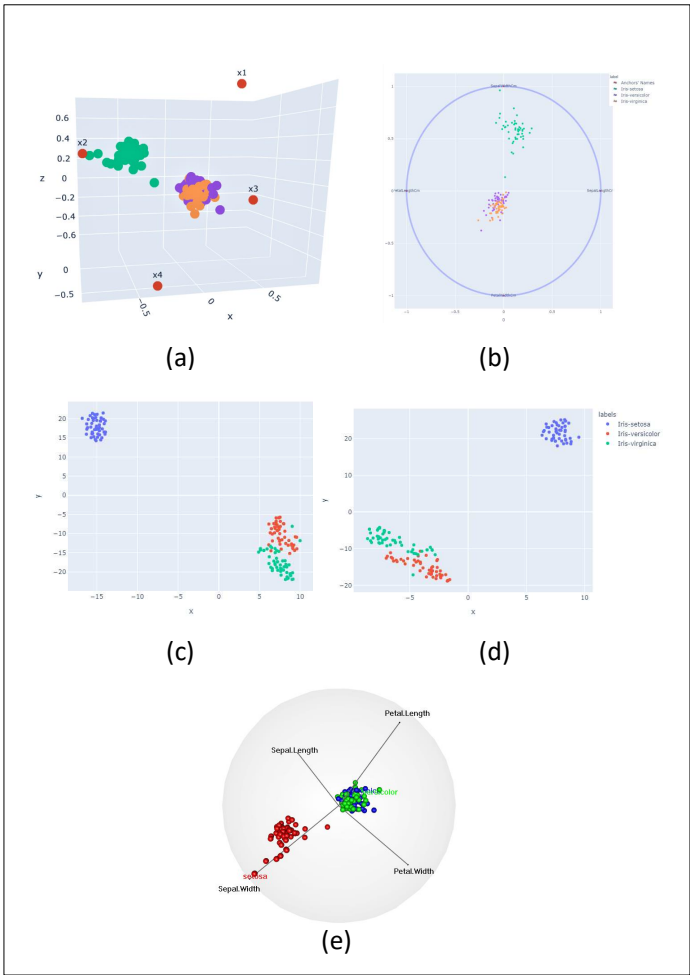


Fig. 7: (a) 2D plot of the Iris dataset showing the three different labels. Iris Setosa in green color, IrisVersicolour in purple color, and Iris Virginica in orange color. (b) 3D plot of the Iris dataset showing the 3 labels and the 4 attributes x1, x2, x3, and x4 for sepal length and width, petal length and width, respectively. (c) First attempt to plot 2D output of the Iris dataset with the t-SNE method. (d) Second attempt to plot 2D output of the Iris dataset with the t-SNE method with the same parameters of the first attempt. (e) Iris dataset plot with the RadViz3D method.

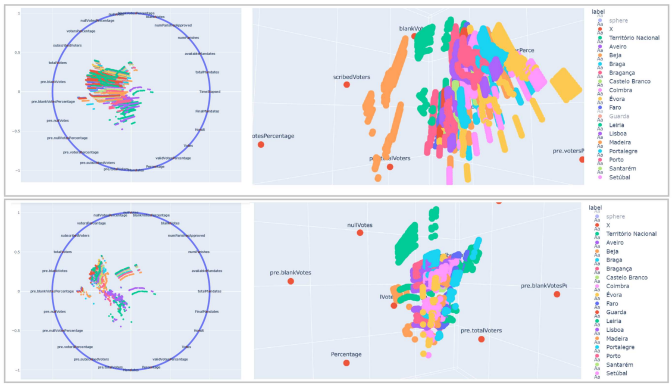


Fig. 8: 2D and 3D RadViz for Election Dataset by parties' classification with and without time elapsed.

2) ***Real-time Election Results: Portugal 2019 Dataset****:* This dataset shows the results of the Portuguese Parliamentary Elections [40] of October 6th, 2019 with 29 attributes. We plot two graphs to represent this dataset, both grouped by parties'

Authorized licensed use limited to: Ontario Tech University. Downloaded on May 07,2022 at 16:23:18 UTC from IEEE Xplore. Restrictions apply.

classification as shown in Fig. 8. The first graph plotted with time elapsed attribute and the second without time elapsed attribute. The Election dataset is created by collecting the parties' situations in Portugal's territories every five minutes. That is, the time elapsed is multiples of five. Results with time elapsed show an interesting pattern where the records of different parties create lines in some territories. The proposed 3D-RadViz managed to minimize the records' ambiguity.

As mentioned earlier, the t-SNE method fails to analyze plots with labels and attributes more than three [12], as shown in Fig. 9 we used the plotly library to explore the t-SNE performance and the results were uninformative and overlapped randomly as expected. For the RadViz3D method, we plot the election dataset by parties' classification with time elapsed as shown in Fig. 10 (a) and without time elapsed as shown in Fig. 10 (b). Although the results look-alike our method, we noticed significant drawbacks of the RadViz3D method. Firstly, the RadViz3D method has modest interactive tools, you can rotate the figure using the mouse only with no additional features.

Secondly, the function showed a noticeable delay in time and response when running a large dataset. Our library using Plotly has more powerful tools such as zoom in, zoom out with mouse roll, or zooming by selecting a specific area. Furthermore, our method has the power of filtering the labels to hide or show specific labels, unlike the RadViz3D method. These tools provided by Plotly helps to visualize the labels as desired by the users within different aspects such as the number of labels, the plot angle of view, and zooming with no time or response delays.

V. Discussion

In this section, we will compare the performance with respect to the execution time as well as the visualization obtained from the three techniques namely; t-SNE, RadVis3D, and our method 3D-RadViz. Table II shows the estimated time for each method to run the function and plot 3D visualization for the iris, and election datasets. Our algorithm performs better than t-SNE and RadViz3D as shown in table II, our proposed algorithm execution was noticeably faster than t-SNE. RadViz3D method and ours show almost the same running time for the iris dataset which is considered as a small dataset. However, for larger datasets such as the election dataset, we noticed that the RadViz3D method consumed longer time to plot the 3D figure. Our method is able to plot the election dataset in 1.99 seconds, but the RadViz3D method plotted the same dataset in 29.17 seconds which considered a notable difference (i.e., 14 times faster).

The t-SNE [12] method is successful at plotting data with high dimensional datasets with fewer number of labels. However, the method did not show any significant results since it has two major drawbacks. Firstly, t-SNE is a stochastic algorithm that produces different outputs at each run. Secondly, t-SNE is affected by the curse of dimensionality since it degrades as the number of dimensions increases.

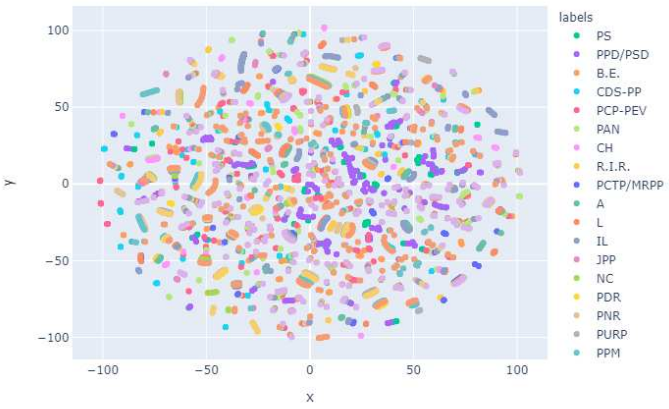


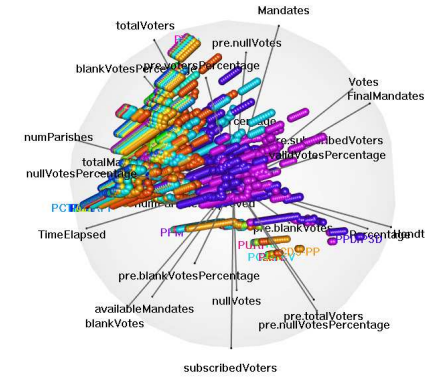
Fig. 9: Election dataset 2D plot with the t-SNE method using plotly.

Furthermore, the RadViz3D method [36] is able to demonstrate the 3D plots for high dimensional datasets. Although this method broadly supports the finding of our method in this area, we noticed some major drawbacks while running the code. The execution time of the algorithm is relatively long as shown in table II and the time increases for larger datasets. Furthermore, the plots are limited by the features provided to examine the figure. This method only provides one feature which is to rotate the figure using the mouse with a notable delay in the response. For dense dimensional datasets with many labels and attributes, this feature will not be efficient to observe the details in the 3D figures.

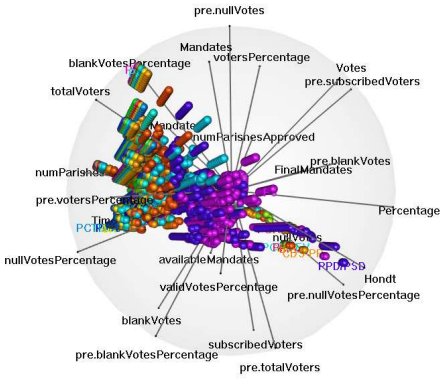
3D-RadViz successfully plots high dimensional data with multiple numbers of labels and attributes with no errors. The execution time to plot the 3D plots is considered short when compared to the other methods even for larger datasets. Furthermore, another obvious finding to emerge from the results that our method has rich features to examine the 2D and 3D plots clearly using Plotly. Plotly figures show an icon with different functions such as download the plot, zoom in and out, add or remove any label from the figure, rotation with no delay in the response, auto-scale and reset axes. These functionalities give a clear view of the 3D plots since they contain multiple labels and attributes that need to be viewed from different angles.

On the other hand, the proposed 3D-RadViz method has some limitations with dense dimensional dataset and record's ambiguity poses. Besides, as compared to RadViz, the dilemma of repositioning of anchor points over the surface of the sphere is a big concern of the proposed visualization scheme. In the future, we plan to enhance the proposed 3D-Radviz by using optimum re-positioning of anchor points based on the relationship among the dimensions within the dataset to improve the insights inferred from the visual. It also helps to alleviate the effect of record's ambiguity with dense dimensional data, since more points end up near the surface of the unit sphere rather than the center of the sphere.

Authorized licensed use limited to: Ontario Tech University. Downloaded on May 07,2022 at 16:23:18 UTC from IEEE Xplore. Restrictions apply.



(a)



(b)

Fig. 10: (a) Election dataset plot by parties' classification with RadViz3D method with time elapsed, (b) Election dataset plot by parties' classification with RadViz3D method without time elapsed.

TABLE II: CODE PROCESSING TIME IN SECONDS TO PLOT 3D VISUALIZATION OF THE TWO DATASETS USING T-SNE, RADVIZ3D, AND 3D-RADVIZ.

Dataset	Method			
	t-SNE 2D	t-SNE 3D	Proposed Method (3D-RadViz)	RadViz3D
iris	1.03	2.04	0.34	0.265698
election	187.33	453.17	1.99	29.18677

VI. Conclusion

In this paper we propose 3D-RadViz, a visualization technique for high dimensional data to extend the famous radial visualization technique (RadViz) while trying to survive its records ambiguity limitation, as the number of attributes within the dataset increases. We introduce a third dimension to the data visualization representation which has been proven through evaluation in this paper to mitigate records' ambiguity in the resulting visualization, allowing better insights inferred from the visual representation of data.

Furthermore, we represent the attributes of a dataset as anchor points which are then evenly distributed over the surface of a unit sphere rather than the perimeter of a unit circle as implemented by RadViz. Therefore, the uniform distribution of anchor points over the surface of sphere is a basic concern to realize the proposed 3D-RadViz. Besides, it was crucial to perform the uniform distribution of anchors in a deterministic way, so that the resulting visualization is perpetual for each run of the proposed scheme in any dataset. It was necessary to evenly distribute an arbitrary number of anchor points over the surface of the sphere since the number of attributes within the datasets is high.

We employed an approximate method to evenly distribute an arbitrary number of points over the surface of a sphere that is based upon Fibonacci lattice and the golden ratio. Although this method is not the best approach to uniformity of points distribution, the resulting points yield excellent sampling properties.To evaluate our proposed visualization scheme, a benchmark data that yields predetermined statistical properties was created and applied to the proposed visualization technique. To contrast the key strength features of the proposed visualization technique, same rReal-life datasets from UCI machine learning repository were applied to the proposed visualization scheme, as well as other competitive visualization techniques in the literature namely; t-SNE and RadViz3D. The results show that the proposed scheme outperforms the other techniques with respect to the time of execution and interactivity (Filtration, Rotation, and Zooming).

References

[1] D. L. Ripley and T. Politzer, "Vision Disturbance after TBI," *NeuroRehabilitation*, vol. 27, no. 3, pp. 215–216, 2010.

[2] C. R. Rao, E. J. Wegman, and J. L. Solka, *Data mining and data visualization*. Amsterdam: Elsevier North Holland, 2005.

[3] K. R. Harrison, A. A. Bidgoli, S. Rahnamayan, K. Deb, Image-Based Benchmarking and Visualization for Large-Scale Global Optimization. arXiv preprint arXiv:2007.12332. 2020.

[4] S. Liu, W. Cui, Y. Wu, and M. Liu, "A survey on information visualization: recent advances and challenges," *The Visual Computer*, vol. 30, no. 12, pp. 1373–1393, 2014.

[5] M. Theus, "High-dimensional data visualization", *Handbook of data visualization*, 2008.

[6] S. Liu, D. Maljovec, B. Wang, P.-T. Bremer and V. Pascucci, "Visualizing high-dimensional data: Advances in the past decade", *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 3, pp. 1249-1268, 2017.

[7] S. Blythe, "Reading in information visualization: using vision to Think [Media Review]," *Technical communication quarterly 9*, no. 3 (July 1, 2000).

[8] E. Bertini, A. Tatu, D. Keim, Quality Metrics in High-Dimensional Data Visualization: An Overview and Systematization, *IEEE Transactions on Visualization and Computer Graphics 17*, no. 12 ,December 2011.

[9] T. Munzner and E. Maguire, Visualization analysis and design. *Boca Raton: CRC Press*, Taylor amp; Francis Group, 2015.

[10] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 2, no. 4, pp. 433–459, 2010.

[11] T. F. Cox and C. M. A. A., Multidimensional scaling. *London: Chapman amp; Hall*, 1994.

[12] L. Maaten and G. Hinton, "Visualizing data using t-SNE", *JMLR*, vol. 9, pp. 2579-2605, 2008.

[13] L. McInnes, J. Healy, N. Saul, and L. Großberger, "UMAP: Uniform Manifold Approximation and Projection," *Journal of Open Source Software*, vol. 3, no. 29, p. 861, 2018.

[14] Y. Ma and L. Zhu, "A Review on Dimension Reduction," *International Statistical Review*, vol. 81, no. 1, pp. 134–150, 2012.

[15] M. Ben-Menachem, "Parallel Coordinates: Visual Multidimensional Geometry and its Applications, *ACM SIGSOFT Software Engineering Notes*, vol. 35, no. 3, pp. 39–39, 2010.

Authorized licensed use limited to: Ontario Tech University. Downloaded on May 07,2022 at 16:23:18 UTC from IEEE Xplore. Restrictions apply.

[16] M. Rubio-Sanchez, L. Raya, F. Diaz, and A. Sanchez, "A comparative study between RadViz and Star Coordinates," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 1, pp. 619–628, 2016.
[17] J. Johansson and C. Forsell, "Evaluation of Parallel Coordinates: Overview, Categorization and Guidelines for Future Research," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 1, pp. 579–588, 2016.
[18] G. Grinstein, M. Trutschl and U. Cvek, "High-dimensional visualizations.", *In Proceedings International Workshop on Visual Data Mining 2001*, pp. 7-19, 2001.
[19] L. de de Carvalho Pagliosa and A. C. Telea, "RadViz: Improvements on Radial-Based Visualizations++," *Informatics*, vol. 6, no. 2, p. 16, 2019.
[20] D. fan, Z. yifan, A. ranjan, "Three-dimensional Radial Visualization of High-dimensional Continuous or Discrete Data," *IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS*, VOL. 17, NO. 8, 2018.
[21] J. Sansen, G. Richer, T. Jourde, F. Lalanne, D. Auber, and R. Bourqui, "Visual Exploration of Large Multidimensional Data Using Parallel Coordinates on Big Data Infrastructure," *Informatics*, vol. 4, no. 3, p. 21, 2017.
[22] G. Ellis and A. Dix, "Enabling Automatic Clutter Reduction in Parallel Coordinate Plots," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 5, pp. 717–724, 2006.
[23] A. O. Artero, M. C. F. de Oliveira, and H. Levkowitz, "Uncovering Clusters in Crowded Parallel Coordinates Visualizations," *IEEE Symposium on Information Visualization*, 2004
[24] H. Hauser, F. Ledermann, and H. Doleisch, "Angular brushing of extended parallel coordinates," *IEEE Symposium on Information Visualization*, 2002.
[25] J. Johansson, P. Ljung, M. Jern, and M. Cooper, "Revealing Structure within Clustered Parallel Coordinates Displays," *Proceedings of the 2005 IEEE Symposium on Information Visualization (INFOVIS'05)*, 2005.
[26] P. Hoffman, G. Grinstein, K. Marx, I. Grosse, and E. Stanley, "DNA visual and analytic data mining," *IEEE Computer Society Press*, 1997.
[27] G. Grinstein, C. Jessee, P. Hoffman, P. O'Neil and A. Gee, "High-Dimensional Visualization Support for Data Mining Gene Expression Data," *DNA Arrays Technologies and Experimental Strategies*, CRC Press, 2002.
[28] K. Daniels, G. Grinstein, A. Russell, and M. Glidden, "Properties of normalized radial visualizations," *Information Visualization*, vol. 11, no. 4, pp. 273–300, 2012.
[29] L. de Carvalho Pagliosa and A. C. Telea, "RadViz: Improvements on Radial-Based Visualizations++," *Informatics*, vol. 6, no. 2, p. 16, 2019.
[30] F. Dai, Y. Zhu, and Ranjan Maitra, "Three-dimensional Radial Visualization of High-dimensional Continuous or Discrete Data," *ArXiv*. abs/1904.06366. 2019.
[31] Y. Ma and L. Zhu, "A Review on Dimension Reduction," *International Statistical Review*, vol. 81, no. 1, pp. 134–150, 2012.
[32] A. N. Gorban, Principal manifolds for data visualization and dimension reduction. Berlin, *Springer*, 2008.
[33] L. G. Nonato and M. Aupetit, "Multidimensional Projection for Visual Analytics: Linking Techniques with Distortions, Tasks, and Layout Enrichment," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 8, pp. 2650–2673, 2019.
[34] A. Inselberg, "Parallel Coordinates: Visual Multidimensional Geometry and its Applications," *ACM SIGSOFT Software Engineering Notes*, vol. 35, no. 3, pp. 39–39, 2010.
[35] R. McGill, J. W. Tukey, and W. A. Larsen, "Variations of Box Plots," *The American Statistician*, vol. 32, no. 1, p. 12, 1978.
[36] H. Vogel, "A better way to construct the sunflower head," *Mathematical Biosciences*, vol. 44, no. 3-4, pp. 179–189, 1979.
[37] E. B. Saff and A. B. Kuijlaars, "Distributing many points on a sphere," *The Mathematical Intelligencer*, vol. 19, no. 1, pp. 5–11, 1997.
[38] J. SHARKO, G. GRINSTEIN AND K. A. MARX , Vectorized Radviz and Its Application to Multiple Cluster Datasets, Transactions on Visualization and Computer Graphics, vol.14, no.6, pp. 1444-1427, 2008.
[39] Fisher and R.A, *Iris Dataset*. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/iris. [Accessed: 15-Apr-2021].
[40] MONIZ AND NUNO, *Real-time Election Results: Portugal 2019 DataSet*, [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Real-time+Election+Results%3A+Portugal+2019. [Accessed: 15-Apr-2021].
[41] A. Elewah, "3DRadViz-Results," Binder. [Online]. Available: https://mybinder.org/v2/gh/elewah/3DRadViz-Results/master. [Accessed: 15-Apr-2021].

Authorized licensed use limited to: Ontario Tech University. Downloaded on May 07,2022 at 16:23:18 UTC from IEEE Xplore. Restrictions apply.