DatasetWiz: A Visual Analytics Framework for the Comparative Analysis of Dimensionality Reduction and Clustering Quality

Master's Thesis
of

Aditya Handrale
108489

as part of the study program

curriculum
at the

University of Passau
Faculty of Computer Science and Mathematics

Chair of Cognitive Sensor Systems

Advisor:	Prof. Dr. Christoph Heinzl
Assistance:	Anja Heim

Passau, DD.MM.20YY

Contents

Acknowledgments

First and foremost, I would like to express my sincere gratitude to my supervisor, Prof. Dr. Christoph Heinzl, for his invaluable guidance, expertise in visualization research, and unwavering support throughout this thesis journey. His insights into cognitive sensor systems and visual analytics have been instrumental in shaping this work.

I extend my heartfelt appreciation to Anja Heim, my academic assistant, whose patient guidance, constructive feedback, and technical expertise helped me navigate the complexities of visualization design and implementation. Her dedication to helping students succeed is truly commendable.

My deepest gratitude goes to my industry partners at Robert Bosch GmbH: Johannes Mohren and Dr. Sabrina Schmedding. Their real-world perspective on industrial quality assessment challenges provided the essential context that transformed this from a purely academic exercise into a meaningful contribution to industrial practice. The datasets, domain knowledge, and collaborative discussions were invaluable to this research.

I am grateful to the University of Passau for providing the academic environment and resources necessary for this research, and for fostering interdisciplinary collaboration between academia and industry.

Special thanks to my family, my parents and brother, who provided unwavering emotional support and encouragement throughout the challenging periods of this thesis. Their constant reminders to stay focused and not procrastinate (which I may have occasionally ignored) kept me on track during the most demanding phases of this work.

Abstract

In data intensive domains such as manufacturing and medical diagnostics, the success of predictive models is primarily driven by the quality and separability of high-dimensional feature spaces. For practitioners to analyze various properties of these datasets, feature extraction algorithms generate hundreds of features, which are further analyzed using dimensionality reduction (DR) and clustering techniques. An accurate understanding of these high dimensional feature spaces is crucial, especially since data scientists and machine learning engineers make downstream decisions based on the understanding of the feature space. A comparison of various dimensionality reduction techniques, both quantitatively using metrics and qualitatively using visualization is absolutely vital for investigation of the dataset "trainability" and feature space quality assessment. As of now, practitioners rely on "black box" projections and static score tables when analyzing the quality of the high dimensional feature spaces. Visual inspection of lower dimensional projections of these feature spaces is often used to investigate the quality of the dataset and to find various effects, artifacts and outliers. The quality metrics and dimensionality reduction methods must be compared manually, which makes this task time consuming, error prone and cognitively demanding. This thesis aims to support the domain experts in the evaluation of the dataset suitability for downstream classification tasks.

To tackle these challenges, our work introduces DatasetWiz, a comparative visual analytics framework that provides a comprehensive understanding of the feature space quality and projection reliability using summary visualization and two novel visualization techniques. Various dimensionality reduction methods are used to summarize the high dimensional structures and are rendered in a side by side comparison tool called DimCompare (Synchronized dual view scatterplots). Information about why the clusters form is calculated statistically and then visualized using Feature Contribution Glyphs (Cluster annotations that highlight feature level differences). The aggregate performance of the different techniques can be explored in a composite visualisation called BarDar Chart (Summary overview combining Bar chart and Radar chart). The efficacy and usefulness of these visualisations are demonstrated using case studies and a user study with X participants. [The results indicate that DatasetWiz successfully facilitates the identification of structural patterns and artifacts, thereby improving the efficiency of early-stage data diagnostics.]

1 Introduction

In the past few years, high-dimensional data has grown more prevalent in fields like manufacturing, medical diagnostics, environmental monitoring, and quality control. As machine learning has gained popularity across multiple domains, feature extraction techniques, be they statistical, domain-specific, or neural network-oriented, now generate hundreds or even thousands of dimensions/features for a singular data item. These feature vectors often include a lot of information in them, but since they are so complex, it's hard for people to understand, compare, or think about these high-dimensional representations. To address the complexity of these high-dimensional spaces, practitioners make use of Dimensionality Reduction (DR), which is the process of mapping high dimensional data into a lower dimensional representation (typically 2D or 3D), while attempting to preserve the original structural relationships.

1.1 Motivation and problem statement

This problem is especially important in factories and industries, where datasets might not be balanced, might have a lot of noise, or might have been collected under less than optimal settings. Before spending time and money on training a model, machine learning programmers and domain specialists need to perform a trainability assessment to verify whether a dataset is "learnable"or not. This means that the data has enough structure, class separation, or structural signal on its own to make modeling useful. Unfortunately, the technologies we have presently don't assist us in trainability assessment very much. Black-box dimensionality reduction projections, static 2D visualizations like scatterplots, and clustering score tables only show part of the picture. This means that users sometimes have to trust their gut feelings or judgment. Conventional visualization methods do not facilitate the interactive exploration and comparative analysis that is crucial for successful decision-making in industrial settings.

The need for this thesis arose from the disparity between high-dimensional feature representations and human comprehension. A prior research collaboration with Bosch underscored the importance for enhanced feature space diagnostics. The partnership's main goal was to find flaws in industrial hardware parts using images, but the basic problem, i.e. understanding high-dimensional embeddings and their structure, goes much beyond merely images. In fact, the tools that were built in this thesis were tested on conventional, high dimensional CSV-based datasets, such as those that assess air pollution or material strength. This shows that the tools can handle a wide range of data.

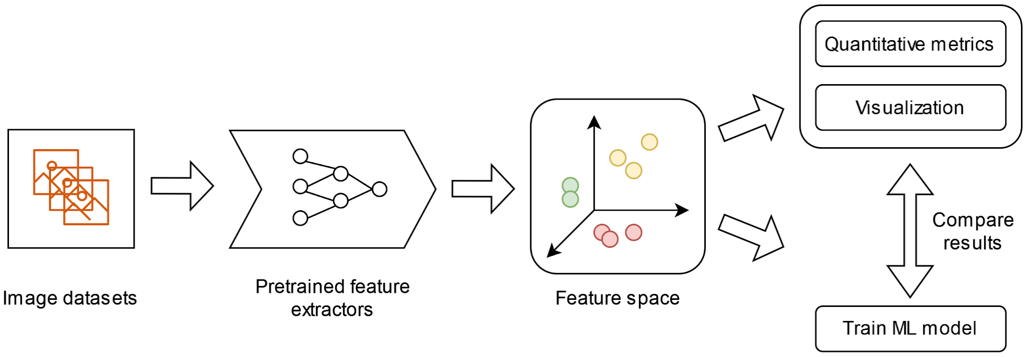

Figure 1: Feature-space analysis pipeline combining dimensionality reduction, visualization, quantitative metrics, and downstream machine learning

More broadly speaking, analyzing and comparing high-dimensional feature spaces is difficult for several reasons:

1. **Projection Instability**: Different Dimensionality Reduction (DR) techniques (PCA, t-SNE, MDS, UMAP) produce inconsistent feature visualizations, each emphasizing different structures. No single method provides a complete picture.
2. **Information loss**: Reducing dimensions often distorts some feature relationships. Clusters may appear well-separated in 2D but actually overlap in the original space, or vice versa.
3. **Lack of visual and quantitative integration**: Clustering metrics like Silhouette Score offer objective numeric values, but interpreting them in context is difficult without visual feedback.

This thesis aims to fill the gap between complex high-dimensional feature spaces and human understanding by introducing novel interactive visualization tools. These tools aim to combine dimensionality reduction, clustering quality metrics, and feature-based visual encodings to support early decision-making in the data science pipeline, for tasks such as evaluating dataset, pointing out anomalies, or identifying the requirement for further preprocessing. Rather than simply illustrating DR results, the goal here is to help the end users, engineers and data scientists, interpret, compare, and assess the quality and separability of high-dimensional data in an intuitive way.

1.2 Research questions

This research is guided by five connected research questions that aim to address both the theoretical and the practical aspects of high-dimensional data visualization in industrial contexts:

Research Question 1 (RQ1): Feature Relationship Understanding

- How can interactive visualization reveal which features drive cluster formation across different dimensionality reduction techniques?

This question seeks to resolve the interpretability challenge that is often posed in high-dimensional feature spaces, where understanding the feature contributions are essential for quality assessment and evaluation, but difficult to achieve with traditional visualization approaches.

Research Question 2 (RQ2): Comparative Evaluation of Dimensionality Reduction

- How can we effectively enable side-by-side comparison of different dimensionality reduction results to support algorithm selection and validation?

This question focuses on the practical needs for practitioners to understand how different dimensionality reduction method choices affect representation of the data and its subsequent analysis.

Research Question 3 (RQ3): Clustering Quality Evaluation and Visualization

- How can composite integrated visualisation effectively facilitate comparative assessment of dimensionality reduction techniques through the aggregation of conflicting quality metrics?

This question addresses the problem of interpreting multiple, and potentially conflicting, quality measures in a visual framework.

Research Question 4 (RQ4): Pattern Identification Across DR Techniques

- How can coordinated multiple views and interactive exploration support the validation of structural patterns and identification of projection induced artifacts across different DR methods?

This question explores the robustness of patterns that are discovered and also the identification of DR technique-specific artifacts.

1.3 Contributions

This thesis makes several novel contributions in the domains of data/information visualization, industrial quality assessment and visual analytics.

1.3.1 Interactive Visualization Techniques

DimCompare Dual-View system: We introduce a novel, dual-view approach for comparing two different dimensionality reduction techniques via synchronized, interactive scatterplots. Compared to traditional single view 2D scatterplots, DimCompare allows for real-time linked interaction between different dimensionality reduction techniques and representations. DimCompare also enables brushing, selection and coordinated exploration.

Dynamic feature contribution glyphs : We developed an innovative cluster annotation "glyphs” that visualize which features contribute most to a cluster formation, dynamically. These cluster annotations provide immediate visual feedback about the feature importance differences between the clusters, which update in real time, as users explore different data subsets.

Bardar Composite Visualization: We created a composite integrated visualiation that utilizes explicit encoding (bar charts) to overcome the well documented perceptual limitations of radar charts (area bias), hence providing an objective ranking of DR method reliability. The integrated design provides users with both detailed metric visualization and also aggregate the performance ranking, enabling more effective metric comparisons across multiple dimensionality reduction techniques.

1.3.2 Technical implementation

Web-based architecture: Development of a complete web-based visualization framework, using Django backend, D3.js frontend, which enables browser-based internet access to visualization capabilities without requiring specialized installation of software.

Scalable Data pipeline: Implementation of efficient algorithms for dimensionality reduction, clustering and metric calculations, that can handle large, industrial scale CSV datasets, while still maintaining, near-real time, interactive performance.

Open source framework: Creation of an open source implementation, that is extensible, and enables reproducible research, while also providing a foundation for future visualization tool development.

1.4 Chapter summary

This chapter introduced the motivation, scope and contributions of this thesis. It establishes the core challenge of assessing the “trainability” of a dataset, or separation of high dimensional feature spaces. This is a critical problem in the Industrial AI and quality assessment, as highlighted by the collaboration with Bosch. The key points discussed are :

- **Core problem:** Identified the black box/uninterpretable nature of feature spaces generated as the main problem. The success of a ML model, further in the pipeline, is dependent on this data, hence leads to uncertainty in the data science pipeline.
- **Identified gaps:** We defined some specific limitations of the current methods, like projection instability (different algorithm = different results), information loss (2D projections lose information) and the disconnect between quantitative metrics and qualitative visual inspection.
- **Research questions:** Synthesised the core research questions (RQs) that guide this work, focusing on the feature level understanding, DR comparison, Metric visualization, pattern identification and industrial validation.
- **Novel Contributions:** Introduced the two primary contributions of this thesis, the DimCompare system for visual comparison exploration and the BarDar chart for quantitative metric comparison.

2 Background and related work

This chapter lays the foundation for this thesis by reviewing some basic ideas and existing research that is important for analyzing high-dimensional data, machine learning, and visual analytics. It covers key areas like extracting features from images, methods for dimensionality reduction, clustering algorithms and their related quality metrics, and different methods to visualize and understand High dimensional data.

2.1 Understanding High dimensional data

High-dimensional data refers to the datasets where each of the sample or row is described by a large number of features or variables, often dozens, hundreds, or even thousands. These features can come from a wide variety of sources, including sensor readings, material strength measurements, air quality parameters, or neural network embeddings for images. In this thesis, high-dimensional data is primarily handled in the form of structured CSV files, where each row represents a data instance/sample and each column corresponds to a feature, usually numeric. Such high dimensional data is common in real-world domains like environmental monitoring, quality control, and manufacturing process analysis. However, as the number of dimensions/features increases, it becomes increasingly challenging to explore, visualize or extract meaningful patterns from the data using traditional visualization techniques. This is commonly known as the curse of dimensionality, and it motivates for the use of dimensionality reduction techniques to uncover structure and gives support in interpretation.

2.1.1 Curse of dimensionality

Analyzing high dimensional data has been recognized as one of the fundamental problems in machine learning and data analysis [1]. As also noted by Jia et. al. [2], the curse of dimensionality significantly increases computational costs and the storage requirements, while also negatively impacting the accuracy and efficiency of the data analysis methods/algorithms. There is an exponential increase in data sparsity and computational demands as dimensionality grows [3] [4].

This phenomenon is especially evident in image applications where CNN(Convolutional Neural Network) based feature extractors (VGG-19, ResNet-50 etc) are used to generate thousands of features from a single image, creating complex feature spaces that are difficult to interpret as well as computationally intensive.

Anowar et. al. [5] gives a complete comparison of dimensionality reduction algorithms. They categorize them into primarily linear vs non linear and supervised vs unsupervised approaches. The empirical analysis performed by them across challenging datasets clearly demonstrates that different dimensionality reduction techniques excel in different contexts, underscoring a need for a comparative analysis tools that can help practitioners select appropriate methods for their specific applications

2.1.2 Image feature extraction in industrial manufacturing

To evaluate properties like structural integrity of industrial hardware, high dimensional representations are generated by using deep neural network based image feature extractors. In current manufacturing workflows, pretrained image feature extraction models such as VGG-19 (Visual Geometry Group) and ResNet-50 (Residual Network) are used to convert visual inspection data into numerical feature vectors, consisting of thousands of dimensions. This work also briefly focuses on analyzing the resulting feature spaces to identify production line induced characteristics such as defects and manufacturing inconsistencies.

2.2 Dimensionality Reduction Techniques

Dimensionality reduction is a key set of methods that are used to change the data from a higher dimensional space into a lower dimensional one, while also trying to keep the important properties and structure present in the original data. [6]. These dimensionality reduction methods are generally split into linear and non linear data, where each has different features and compromises. The fact that there exists a vase selection of dimensionality reduction techniques, each having its own biases and compromises, directly indicates why comparative visualization systems are needed. Building such a comparative visualization system is the main goal of this thesis.

2.2.1 Principal Component Analysis (PCA)

This is the most widely used and commonly known Dimensionality reduction method due to its mathematical simplicity and interpretability [7]. It projects the data in a lower dimensional space. PCA does this by finding orthogonal components (principal components) that capture the maximum variation in the data. The linear nature of the technique provides a significant advantage for preserving global structures and relationships in the data [8].

Boileau et al. [9] extend traditional PCA through sparse contrastive PCA, which works by extracting sparse, stable and interpretable features by leveraging control data. Their work demonstrates how PCA variants can be enhanced to address specific domain requirements , particularly in biological applications where interpretability is crucial.

PCA is fast to compute, predictable, and good at keeping the overall structure of the data [8]. However, the downsides of PCA are that it might not capture complex, non-linear relationships present in the data. Which has led to development of kernel PCA and other non-linear extensions [10]. The technique works best when the underlying data is somewhat linear, but it may also miss important patterns in the data that have complex non-linear relationships, such as those found commonly in industrial image analysis applications.

2.2.2 t-Distributed Stochastic Neighbor Embedding (t-SNE)

This is a powerful non-linear method of dimensionality reduction that is really good at showing the local structures present in the data, often revealing tight groups of data points that are similar. t-SNE was introduced by [11], and it revolutionized the visualization of high dimensional data, by preserving local neighborhood structures, while reducing dimensionality.

t-SNE looks at similarities as probabilities and tries to match these probabilities in both the high and low dimensional spaces. It models similarities between data points using probability distributions and minimizes the Kullback-Leibler divergence (KL) between high-dimensional and low-dimensional representations.

It is great at showing clusters that might be hidden in high dimensional space and at preserving local relationships. Making it really valuable for exploratory data analysis (EDA) and pattern discovery. However t-SNE has many limitations that affect the interpretation, it sometimes distorts global structure [12] and the results obtained are sensitive to hyperparameter settings (like perplexity) [13]. Since t-SNE is stochastic in nature, it produces slightly different results in different runs [14]. t-SNE is also computationally expensive $\mathscr{O}(n\hat{}2)$ and can sometimes twist the overall structure, versions like Barnes-hut t-SNE help mitigate this [14].

The non-deterministic nature of this method makes it challenging to reproduce results, increasing the importance of setting random seeds and understanding the impact of hyperparameters upon the final visualisations.

2.2.3 Multidimensional Scaling (MDS)

Multidimensional Scaling (MDS) is a classical dimensionality reduction technique that aims at preserving pairwise distances between the different data points, when projecting high dimensional data into lower dimensional space. Originally developed in the field of psychometrics, it is now widely used across various domains. MDS works by transforming a dissimilarity matrix into a geometric configuration in fewer dimensions, while still maintaining the relative spatial relationships of the data. [15]

Unlike linear and variance based methods like PCA, MDS is distance-preserving. It attempts to place each data point in a low dimensional space such that the -inter-object distances are preserved as faithfully as possible. MDS works best when input distances are euclidean, and the data conforms to metric assumptions [16]. There also exist variants of MDS such as non-metric MDS, that relax these assumptions by only preserving the rank of order distances, which makes it more robust to non-linear data structures.

This dimensionality reduction method tried to keep the distances between the data points as much as possible in the lower dimensional space. MDS is good for understanding how far apart items are from each other, but it takes a lot of computing power for larger datasets, (with a computational time complexity of $\mathscr{O}(n^3)$ for exact calculation, where N refers to the number of data points). This makes it less practical for datasets without approximation strategies [17]. Because it needs so much computing, it's usually preferred for smaller datasets. Moreover, it does not explicitly model local or global structure tradeoffs the way t-SNE or UMAP do. Therefore MDS can struggle to highlight cluster boundaries and maintain neighborhood relationships in sparse datasets. MDS provides a valuable contrast to techniques like PCA, t-SNE and UMAP. Its role in this thesis is to primarily serve as a comparative benchmark.

2.2.4 Uniform Manifold Approximation (UMAP)

UMAP was developed by [18], and it addressed several limitations of t-SNE while maintaining the ability to preserve local structure. It is based on manifold learning theory and topological data analysis, and provides faster computation than t-SNE while better preserving global structure alongside local neighborhoods.

This method constructs a high dimensional graph representation of the data and then optimises low dimensional graphs to be as structurally similar to the high dimensional graph as possible.

This kind of approach allows UMAP to handle larger datasets more efficiently than t-SNE, while also producing more stable results across multiple different runs.

UMAP has the ability to preserve both local and global structures, which makes it particularly valuable for industrial applications where understanding both the detailed cluster structure and overall data structure is important, like exploratory analyses in industrial datasets [19]. However, the technique introduces its own set of hyperparameters and assumptions that can impact the final visualization, re- emphasizing the need for competitive analysis tools.

2.2.5 Other techniques : Isomap and Autoencoders

Beyond PCA, t-SNE, UMAP and MDS, there exist additional dimensionality reduction techniques which offer alternative perspectives on high dimensional data, like Isomap and Autoencoders. Isomap [20] extends the classic MDS technique by incorporating geodesic distances, instead of euclidean ones, which allows it to preserve the intrinsic geometry of non-linear manifolds. It constructs a neighborhood graph, and computes the shortest path distances between the points. However, Isomap is sensitive to noise and outliers, and its performance degrades if the neighborhood graph is poorly constructed.

Autoencoders, on the other hand, are unsupervised neural network models that learn to compress the data into lower-dimensional latent representations and reconstruct it back to the original space. [21]. This learned embedding captures the non-linear dependencies and is highly flexible due to the representational power of deep neural networks. There also exist variants such as denoising autoencoders or Variational Autoencoders (VAEs) that have further improved robustness and generative capabilities. However, autoencoders typically require very large training data sets and careful tuning, to learn trivial representations and avoid overfitting.

While these techniques were not the focus of the implementation in this thesis, they offer valuable alternatives and are potential candidates for the future extensions of comparative visualization tools, particularly in deep learning focused workflows.

Method	Linearity	Structure Preservation	Time Complexity
PCA	Linear	Global (variance)	$\mathcal{O}(nd^2)$
t-SNE	Nonlinear	Local	$\mathcal{O}(n^2)$
UMAP	Nonlinear	Local + Global	$\mathcal{O}(n \log n)$
MDS	Nonlinear	Distance preserving	$\mathcal{O}(n^3)$

Table 1: Summary of Dimensionality Reduction Techniques

2.3 Comparative analysis challenges

Espadoto et. al [22] provides a quantitative survey of various dimensionality reduction techniques, it evaluates how these DR methods perform across a wide variety of datasets and metrics. The study shows that no single method always outperforms the others, and each one ever gives only a partial or biased view of the high dimensional data, which underscores the need for comparative analysis tools. This is why the core design of DimCompare is justified.

The authors identified several challenges in evaluating dimensionality reduction techniques:

1. Lack of ground truth, which makes it difficult to assess the quality of the projections.
2. Tradeoffs exist between local and global structure preservation.
3. Dataset characteristics, such as noise and dimensionality, have an influence on the dimensionality reduction performance.
4. Computational scalability becomes important for practical use on high-dimensional datasets.

These challenges emphasize the need for visualization tools, like the ones developed in this thesis. These tools help practitioners navigate the trade-offs and make informed choices based on context-specific requirements.

2.4 Clustering algorithms for feature space analysis

Clustering algorithms help us identify groups of similar data points that are present in the feature space and to detect outliers, which is essential for evaluating the structure and separability of high dimensional datasets. Clustering methods are commonly classified into four categories, Partitioning methods (e.g., k-Means, GMM), which assign points into clusters, based on minimizing the within-cluster variance or maximizing likelihood. Density-based methods (e.g., DBSCAN, HDBSCAN), which find clusters by identifying dense regions separated by sparse areas. Graph-based methods (e.g., Spectral clustering), which make use of eigenvectors of similarity matrices to reveal structure. Subspace/high-dimensional methods (e.g., SUBCLU, ENClust) which discover clusters that exist in subsets of dimensions.

2.4.1 k-Means Clustering

k-Means is a centroid-based algorithm that partitions data into k clusters by minimizing within-cluster variance using an iterative refinement method [23]. k denotes the number of clusters, a hyperparameter chosen either based on prior knowledge or quality metrics such as the silhouette score.

$$\min_{C} \sum_{i=1}^{k} \sum_{x \in C_i} \|x - \mu_i\|^2$$

- n = number of data points,
- k = number of clusters,
- d = dimensionality,
- I = number of iterations.
- Time complexity: Typically $\mathcal{O}(I \cdot n \cdot k \cdot d)$

It is efficient, easy to implement and compatible with many internal cluster metrics. However, the drawbacks of k-Means are that it assumes spherical clusters of similar size, and is sensitive to outliers and initialization while also requiring to specify k. Because k-Means assumes spherical clusters of similar size, visual inspection with DimCompare can help assess whether this assumption holds in practice.

2.4.2 DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

A density-based method of clustering that identifies arbitrary shaped clusters based on density connectivity and marks low-density points as noise [24]. DBSCAN defines clusters as areas of high density separated by sparse regions. A point p is considered a core point if at least *minPts* neighbors fall within a radius ε. Clusters are formed by density connectivity.

$$|\{q : \|q - p\| \leq \varepsilon\}| \ \geq \ \text{minPts}$$

- ε = neighborhood radius,
- *minPts* = minimum number of neighbors.
- Time complexity: Typically $\mathcal{O}(n \log n)$

It works well for non-convex clusters and clusters of arbitrary shapes and also detects noise. But choosing appropriate distance thresholds can be challenging, especially in high dimensional spaces where distance metrics start becoming less and less meaningful [25].

2.4.3 HDBSCAN (Hierearchical Density-Based Spatial Clustering of Applications with Noise)

An extension of DBSCAN, HDBSCAN constructs a hierarchical clustering structure and extracts clusters based on their stability, and extracts the stable ones [26].

It eliminates the need to choose epsilon and automatically determines the number of clusters. Time complexity : $\mathcal{O}(n^2)$

It is good at discovering clusters of varying densities and identifying outliers without specifying k. This makes it suitable for practical exploratory data analysis scenarios where cluster counts are often unknown or feature spaces contain a lot of noise. Its weaknesses are that it may over split clusters in noisy high dimensional spaces and it is more computationally intensive.

2.4.4 Spectral Clustering

Spectral clustering is a graph based technique of clustering that uses eigenvectors of the similarity matrix's Laplacian to cluster data in reduced dimensions [27]. It is particularly effective at capturing complex cluster structures, but it scales poorly with large scale datasets, due to the costs related to eigen decomposition.

Spectral clustering uses the eigenvectors of a graph Laplacian L=D minus W derived from a similarity matrix W, to embed data before clustering.

$$L = D - W$$

where W is the similarity (adjacency) matrix and D is the degree matrix with $D_{ii} = \sum_j W_{ij}$.

Parameters: Similarity function (e.g., Gaussian kernel), number of clusters k Time Complexity: $\mathcal{O}(n^3)$

2.4.5 Gaussian Mixture Models (GMM)

This is a probabilistic, soft clustering approach that assumes that data is generated from a mixture of various Gaussian distributions. GMM's provide probabilistic class assignments and can model covariance structures well [28]. However, they often do not perform well when clusters deviate significantly from Gaussian shapes and require very careful hyperparameter tuning.

$$p(x \mid \theta) = \sum_{i=1}^{k} \pi_i \mathcal{N}(x \mid \mu_i, \Sigma_i)$$

2.4.6 Subspace clustering

These types of methods, like SUBCLU, identify clusters present in specific subsets of dimensions [29]. This is critical in high-dimensional data, where clusters sometimes only exist in particular feature subspaces, and some global clustering methods might miss such latent structures. Their drawback is that they have high computational complexity.

While a wide range of clustering algorithms exist, such as DBSCAN, Spectral Clustering , Gaussian Mixture Models, and Subspace Clustering methods, this thesis focuses primarily on k-Means. This method was selected because k-Means offers a simple, efficient, and widely used baseline, allowing for robust evaluation of feature spaces within the visualization framework.

2.5 Projection and Clustering Quality Metrics

Evaluating the quality of clusters and hence the quality of the feature space, is extremely important for downstream tasks like model training and decision making. There exists methods of quantifying the quality of clusters, known clustering validity indices, a.k.a clustering quality metrics. Cluster validity indices can be classified as internal, known as intrinsic and external, known as extrinsic, each offering a different perspective. It is crucial to understand that no single metric is universally optimal, each metric has biases and limitations of its own. It is important to remember that each metric looks at a slightly different part of "cluster quality" and no single measure is perfect for everything, especially in high dimensional scenarios, where results can often be misleading, without normalization of contextual analysis [30]. High dimensional data can also distort metrics, so it is essential to combine quantitative evaluation with visualization. [31]

2.5.1 Internal Validity Indices (Intrinsic Metrics)

These types of metrics check the cluster quality based only on the data itself, without requiring outside ground truth labels. They were found to be very sensitive to data quality problems like blurry images and wrong labels, making them a good fit for automatic quality checks. While internal clustering metrics assess the visual separation of clusters, they often lack the ability to validate the integrity of underlying dimensionality reduction process.

2.5.1.1 Silhouette coefficient

It measures how similar a data point is to its own cluster (how close it is), compared to other clusters , (how far apart it is). It was originally proposed by Rousseeuw [32]. The values go from -1 to +1. Higher values mean better-defined clusters. Values above 0.5 are generally good, values above 0.7 indicate strong clustering, whereas negative values suggest that it is in the wrong cluster, i.e. misassignment. Finally, values around 0 indicate that there is some overlap.

The silhouette score for a point i is calculated as:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

Where $a(i)$ is the average distance between i and all other points in its cluster, and $b(i)$ is the minimum average distance to points in any other cluster.

While it is easy to understand, Silhouette coefficient tends to prefer round, equally sized clusters, and it might not be as useful for clusters that are oddly shaped. One study found it to be a better indicator than Davies-Bouldin and Dunn indices [**?**].

2.5.1.2 Davies Bouldin Index (DBI)

Introduced by [33], it calculates the average similarity ratio of each cluster with its most similar cluster. Similarity is defined as the ratio of how spread out things are within a cluster to how far apart the clusters are. Lower values mean better separation, with 0 being the best possible value, meaning that the clusters are perfectly separate. DBI is defined as

$$R_{i,j} = \frac{S_i + S_j}{M_{i,j}}, \quad DB = \frac{1}{k} \sum_{i=1}^{k} \max_{j \neq i} R_{i,j}$$

where S_i is the intra-cluster dispersion for cluster i, and $M_{i,j}$ is the distance between cluster centroids and k is the number of clusters. Lower DBI values indicate better clustering

DBI is sensitive to outliers and differences in cluster shape and density. [34]

2.5.1.3 Trustworthiness (Dimensionality reduction fidelity)

To fill the “trust” gap that’s inherent to low-dimensional projections we use the Trustworthiness metric. Originally proposed by Venna and Kaski [**?**], this metric quantifies the degree to which local neighborhood structure is preserved when data is being mapped from a high dimensional space to a lower dimensional 2D projection. Specifically, it is used to measure the presence of false neighbors, i.e. data points that appear closer in visualization, but which are actually distant in the original feature space. This metric is bounded between 0 and 1, which makes for an objective comparison of different projection methods, across diverse datasets and ensures a consistent normalization. If a projection has high trustworthiness, (near 1.0) the user can be confident that the clusters that they see on the screen are real structures and not artifacts of the algorithm.

It is defined as

$$T(k) = 1 - \frac{2}{nk(2n-3k-1)} \sum_{i=1}^{n} \sum_{j \in N_i^k} \max(0, (r(i,j) - k))$$

Where n is the number of samples (data points).k is the number of nearest neighbors considered. N_i^k is the set of k nearest neighbors of sample i in the output (embedded) space. $r(i, j)$ is the rank of sample j in the input (original) space, when ranked by distance from sample i (e.g., the closest neighbor has rank 1).

2.5.1.4 Calinski Harabasz Index (CHI)

It is defined as the ratio of how spread out things are between clusters to how spread out things are within the clusters. It was first introduced in 1974 [35]. A higher CHI score means clusters are dense and well separated. This metric has no upper limit and it is fast to compute. It is often used to find the best number of clusters by looking at the peak value as the value of k (no. of clusters) changes.

It is defined as

$$CHI = \frac{BCSS/(k-1)}{WCSS/(n-k)}$$

where $BCSS$ is between-cluster sum of squares, and $WCSS$ is within-cluster sum of squares. Higher values indicate better-defined clusters

However, it is not always linear or the best indicator for feature space quality, no upper limit also makes it challenging to do a clear evaluation.

2.5.1.5 Dunn Index (DI)

The Dunn Index aims to identify dense and well separated clusters.It is defined as the ratio of the minimum inter-cluster distance (the shortest distance between any two points in different clusters) to the maximum intra-cluster distance (the diameter of the largest cluster). It was originally introduced in 1973 by J.C. Dunn [36]. A higher value indicates better separation and compactness.

$$DI = \frac{\min_{i \neq j} \delta(C_i, C_j)}{\max_k \Delta(C_k)}$$

Where $\delta(C_i, C_j)$ is the distance between clusters C_i and C_j, and $\delta(C_k)$ is the diameter of cluster C_k.

Its major drawbacks are that it is sensitive to noise and outliers, which can artificially decrease the inter-cluster distance. It has a high computational complexity on large datasets, as it requires calculating numerous pair wise distances. [37]

2.5.1.6 S_Dbw

S_Dbw is a metric that measures both the compactness and the separation of the clusters. It achieves this by combining two components, an intra-cluster variance term (*Scat*) that measures compactness and an inter-cluster density term ($Dens_{bw}$) that measures separation based on the density of points in the region between clusters.

In research [38], it has been found to be strong across different effects and works well on datasets with noise and complex cluster shapes, it outperforms many traditional indices. Its primary limitation is its implementation complexity compared to simpler metrics.

2.5.1.7 FERM (Feature space Evaluation and Representation Method)

FERM (Feature space Evaluation and Representation Method) is a modern metric designed to evaluate qualities of learned feature spaces for a given classification task. Unlike traditional clustering metrics, which are unsupervised, FERM uses class labels to give a quantitative value to a data representation. It evaluates two properties, class separability (how distinct the representation of the two classes are) and class density (how tightly grouped data points of the same class are). [39]

This makes it particularly suitable for evaluating the feature spaces generated by Deep Neural Networks (DNN), such as ones generated for image analysis in industrial tasks, such as image feature extractors (ex. ResNet, VGG-19 and EfficientNet-B0). Where the goal is to learn a simple representation that eases the task for a downstream classifier.

Metric	**What it Measures**	**Range**	**Better When**
Silhouette Coefficient	Cohesion vs separation of clusters	-1 to +1	Higher
Davies–Bouldin Index	Intra-cluster dispersion relative to inter-cluster separation	0 to ∞	Lower
Calinski–Harabasz Index	Ratio of between vs within cluster variation	$[0,\infty)$	Higher
Dunn Index	Minimum inter-cluster separation over maximum intra-cluster diameter	$[0,\infty)$	Higher
Trustworthiness	Preservation of neighbor ranks in projections	0 to 1	Higher
S_Dbw Index	Compactness and density separation	dataset dependent	Lower*
FERM	Feature space class separability	dataset dependent	Higher*

Table 2: Summary of Internal Clustering and Projection Quality Metrics

2.5.2 External Validity Indices (Extrinsic Metrics)

External Validity indices, evaluate the quality of clustering result by comparing it to a ground truth classification part of the data. This implies that the data should be labelled, making these types of metrics supervised. This type of comparison allows an objective assessment of how well the clustering algorithm has retained the underlying structure of the data labels. While being good tools for benchmarking, the main drawback is the reliance on pre-existing labels, making them unsuitable for many unsupervised discovery tasks where ground truth is often unknown.

2.5.2.1 Adjusted Rand Index (ARI)

This metric measures the similarity between two partitions, i.e. the clustering results and the ground truth labels provided. It does this by considering all pairs of samples and counting pairs that are assigned in the same or different clusters in both partitions. It then corrects this Rand Index (RI) for randomness/chance, ensuring that the expected value from random clusterings is 0. It has a range of [-1, 1] where 1 indicates a perfect match and values near 0 indicate random agreement, this makes ARI highly interpretable. As noted by [40], the adjustment for chance is the key advantage over the original Rand Index, which prevents inflated scores with a large number of clusters.

Given a contingency table n_{ij} for clustering labels U and V, with row sums a_i and column sums b_j, total n:

$$ARI = \frac{\sum_{ij}\binom{n_{ij}}{2} - \frac{1}{\binom{n}{2}}\sum_i\binom{a_i}{2}\sum_j\binom{b_j}{2}}{\frac{1}{2}\left(\sum_i\binom{a_i}{2} + \sum_j\binom{b_j}{2}\right) - \frac{1}{\binom{n}{2}}\sum_i\binom{a_i}{2}\sum_j\binom{b_j}{2}}$$

2.5.2.2 Normalized Mutual Information (NMI)

Originated in information theory, NMI quantifies the statistical information shared between clustering assignment and the ground truth labels. The score is normalized to a [0,1] range, where 1 indicates perfect correlation. A comprehensive analysis by [41] highlights that NMI is a robust and widely used metric, but its value can be influenced by the number of clusters, and different normalization methods yield different results.

Given entropies $H(U)$, $H(V)$ and mutual information $I(U,V)$, it is defined as:

$$NMI(U,V) = \frac{I(U,V)}{\sqrt{H(U)H(V)}}$$

2.5.2.3 V-Measure

This is an entropy based metric that combines two desirable properties, homogeneity and completeness. A clustering is homogeneous if each cluster contains only members of one class. A cluster is considered complete if all the members of a given class are assigned to the same cluster. The V-measure is the harmonic mean of these two values, providing a single, interpretable score between 0 and 1. [42]. This metric was introduced to address the shortcomings in other approaches that might ignore the two properties stated above.

$$V = 2 \cdot \frac{hc}{h+c}$$

Where h and c are normalized information-theoretic scores.

2.5.2.4 Fowlkes-Mallows Index (FMI)

FMI is defined as a geometric mean of precision and recall. FMI computes the similarity between two clusterings by seeing the number of pairs of points that exist in the same cluster in both the partitions. It ranges from [0,1] and can be considered intuitive, however, it can be sometimes misleading. As observed in some studies, FMI assigns high scores even if the clusterings don't align with the ground truth, especially if an algorithm produces a large number of small, pure clusters. It is considered generally to be less discriminative than ARI or NMI due to this. [43] Given true positive (TP), false positive (FP), false negative (FN):

$$FMI = \sqrt{\frac{TP}{TP+FP} \cdot \frac{TP}{TP+FN}}$$

2.5.3 Additional metrics

Other useful metrics for feature space analysis also exist beyond the standard clustering indices.

2.5.3.1 No. of outliers

This refers to the number of data points which deviate significantly from their respective cluster centers, a.k.a. Outliers. The higher the value of this metric, the worse is the quality of the feature space. This simple metric is especially useful for understanding feature space noise and data integrity.

2.5.3.2 Mahalanobis distance

Mahalanobis distance measures the distance of a data point from a data distribution, it is defined by For a sample x, distribution mean μ and covariance Σ:

$$d_M(x) = \sqrt{(x-\mu)^\top \Sigma^{-1} (x-\mu)}$$

This distance accounts for covariance structure and is very effective for doing multivariate outlier detection/ anomaly analysis. [44]

Bhattacharya distance This metric measures the overlap between two probability distributions. Higher Bhattacharya distances imply more dissimilarity. It also serves as a strong metric for evaluating feature separability and also to perform feature selection. [45] [46] Given two distributions p and q:

$$D_B(p,q) = -\ln\left(\sum_x \sqrt{p(x)\,q(x)}\right)$$

Higher distances implies less overlap.

2.5.3.3 Jefferies Matusita

This is a normalised variant of Bhattacharya distance, whose value is bounded between 0 and 2. It is also used often in feature selection due to its intuitive scale and computational efficiency. [47]

$$JM(p,q) = \sqrt{2\left(1 - e^{-D_B(p,q)}\right)}$$

Metric	**Category**	**Range**	**Better When**
Adjusted Rand Index (ARI)	External	$[-1,1]$	Higher
Normalized Mutual Information (NMI)	External	$[0,1]$	Higher
V-Measure	External	$[0,1]$	Higher
Fowlkes-Mallows Index (FMI)	External	$[0,1]$	Higher
Mahalanobis Distance	Distance	$[0,\infty)$	Lower
Bhattacharyya Distance	Distribution distance	$[0,\infty)$	Higher
Jeffries-Matusita	Distribution distance	$[0,2]$	Higher

Table 3: Summary of External and Distance-Based Clustering/Feature Space Metrics

2.5.4 Need for visualization aided interpretation

It is very crucial to note that simply relying exclusively on metrics is not enough, since they simplify complex underlying structures into a single digestible numeric value. Visualization is still essential for understanding nuance on cluster forms and context. This means that we need some kind of interactive visual exploration for gaining a full understanding of the feature space.

The general observation is that internal metrics match well with classification accuracy and supports using these metrics as a way to guess future model performance (e.g. Silhouette scores often align with model performance), suggesting that intrinsic metrics can serve as a proxy for clustering performance when ground truth is not available, as is often unavailable in real world scenarios [**?**]. This thesis acknowledges the biases and limits of individual clustering metrics. Therefore, it uses a smart approach which focuses on checking multiple metrics, and confirming them with visualization, to give a more reliable and detailed view of the feature space quality.

2.6 Visualization of Feature Space and Interpretability

High dimensional datasets and their feature spaces are hard to interpret with numbers alone, there is an underlying visual spatial structure of the feature spaces generated, that metrics might take into calculation, but they ignore the power of visualization techniques and visual interpretation which can enhance the comprehension of such datasets.

Internal clustering metrics compress rich structures into numeric scores, which usually hide tradeoffs like local neighborhood preservation vs. global layout stability. Visualization can complement these internal metrics, by exposing patterns, distortions and outliers that affect the downstream decision making. In this thesis, the goal is to not just show low dimension projects, but to also help end users connect what they visually see in the 2D representations with quantitative, metric-based evidence. Empirical studies also demonstrate that quality metrics can be used to guide visual analyses and filter our uninteresting and cluttered views. [48]

2.4.1 Why visualization is needed alongside metrics Quality metrics offer many valuable insights, each capturing different aspects of the feature space but they may sometimes conflict with each other on the same data. Prior research shows that no single metric can tell the whole story, since “quality” can encompass things such as clutter, overlap, pattern recognition etc. [48] This is why visual analytics is so powerful, metrics guide the whole process, but in the end, people are needed to inspect, compare and interpret the data.

Visualization theory provides some design guidance for doing it well. Munzner’s nested model emphasizes firstly clarifying data and task abstractions, then choosing the appropriate encodings and interactions and finally, validating the result. And for comparison based tasks, Gleicher’s taxonomy highlights three patterns, which are, juxtaposition, superposition and explicit encoding of differences. The methods proposed in the thesis, aim to follow these guiding visualization theory principles. DimCompare does this by juxtaposing projections to compare them by showing them side by side and using annotations for clusters (explicit encoding). BarDar uses explicit encoding to show multiple quality metrics in one easy to read chart. [49] [50]

2.6.1 What static plots miss

Static scatterplots are the way to traditionally visualize high dimensional representations, while still useful, they often hide the crucial artifacts that occur after performing dimensionality reduction, like distortions (false neighbors i.e. data points which appear closer together but are not) and tears (data points closer together in the original data are pulled apart in the plot). A body of research highlights these issues, it is argued that analysts must assess first how reliable the projection is first, before making conclusions. These studies show that different dimensionality reduction techniques often optimize for different criterias, which leads to disagreements in the resulting structure. To address this, they propose use of diagnostic overlays and metrics to reveal errors. [51]

Our system adopts this approach, we make the differences visible, highlight the unique features that drive each cluster formation, and also pair visualizations with the quantitative metrics to help prevent misinterpretation.

Traditional radar charts are difficult to interpret because if the order of the axes is changed, the shapes change for the same data and can sometimes get super noisy visually, meaning the order of the axes matters. Moreover, the shapes are hard to read, human brains aren’t good at judging and comparing complex irregular shapes [52], it is hard to tell which polygon is smaller or bigger just by visual inspection. This makes just static radar plots unsuitable for intuitive comparative analysis.

Our system BarDar solves these issues, we use a hybrid design that combines simple radar-style overview with a separate bar chart, we tried to use an approach based on well known criticisms for radar plots. This barchart shows the aggregated score of each projection, so the user doesn’t have to guess which radar chart is better and do the math in their head.

2.6.2 Visual encodings for feature spaces

Glyphs are a common way to encode multivariate attributes at points or in regions. A recent report discusses when glyphs help and how to design them for readability and detail [15] DimCompare's floating cluster annotations extend this by providing cluster level summaries, which highlight what features deviate from the rest of the data, tying together the visualizations back to the original data space.This design makes use of explicit encodings and the recent literature to enhance projections with context information that is often lost when using standard methods.

2.6.3 How this thesis aims to extend existing tools

There already exist relevant visualization tools like Embedding Projector [53] and Clustrophile 2 [54] which provide good support for explorations of single projections and evaluating clustering, they are not designed to enable and assist with comparative assessment and evaluation of dimensionality reduction methods. This is an important challenge for analysts who have to decide which dimensionality reduction projection best represents their data's structure. Our thesis addresses this by creating a workflow specifically for comparative dimensionality reduction evaluation, with cluster aware explanations and multi-metric summaries, which are optimized to judge dataset "learnability.". Our design addresses the gap noted in recent work by Espadoto et al. that call for techniques like linking projections with high dimensional space, metric quality etc. [22]

2.7 Foundational concepts in Interactive Visualization

Visual exploration plays a key role in understanding high dimensional spaces and the hidden structures present inside them. This thesis makes use of several proven interactivity techniques derived from visual analytics, to make a system tailored to real world, industrial data scenarios. It especially uses ideas from Tamara Munzner's [55] "Visual analysis and design" and how Gleicher et al [50] categorize comparative visualization. This theoretical base makes sure that the design choices that went into the system are not random, but are guided by best practices and aim to solve known problems in the field.

2.7.1 Gleicher et al.'s taxonomy of comparative visualization

This framework identifies three main ways to compare visualization

Juxtaposition

Juxtaposition means putting visualizations side by side for direct comparison. DimCompare view with its two scatterplots which depict two different dimensionality reduction techniques is an example of this strategy. L'Yi et al. [56] revisit comparative layout idioms including juxtaposition, superposition, and explicit encoding. In the BarDar chart, two views help the users compare metrics for different dimensionality reduction methods, one view depicts a radar chart and another shows a bar chart representing the aggregated scores from the radar chart.

Superposition

This means layering different views on top of each other, while not the main method in DimCompare, density contours can be seen as a type of superposition over the scatterplot data points. In the BarDar chart, superposition is used to overlay radar charts of different dimensionality reduction techniques on top of each other.

Explicit encoding

This method involves showing the differences or extra information with special visual elements. Important data is explicitly made known to the user to reduce the cognitive load. The cluster glyphs in DimCompare are an example of explicit encoding; they show high dimensional feature differences. They represent what makes a cluster different in original high dimensional space, bridging the gap between 2D projection and basic feature characteristics. In the BarDar chart, the shapes of radar charts, corresponding to different metrics are explicitly encoded for the user, as a bar chart, since visual calculation of areas of irregular shapes is very challenging.

2.7.2 Munzner's principles

Our system design, both directly and indirectly follows several of Munzner's [55] key ideas for good visualization.

Visual encoding

The design of the system carefully thinks about how visual elements like points, colors, glyphs and bars are used to show data properties and relationships effectively, and following good visualization practices. For example color is always used to show which cluster a point belongs to, transparency is used to manage the density of data, different colour pallets are used for separate visualisations.

Interaction techniques

Munzner stresses how important interaction is for exploring data. Our system includes basic methods like selecting, linking and moving around (panning, zooming etc). In DimCompare, users can use brushing and linking to follow outliers across different projections. Tooltips are also made visible in 2D projection, when the cursor is hovered over a datapoint, which shows original high dimension features.

Overview First, Zoom/Filter, Details on demand

This idea suggests giving a general overview of the data first, letting users zoom or filter to areas they are interested in, and then providing detailed information when asked for. Our visualization framework design follows this, the BarDar chart gives and overall summary with numbers, but users can then decide to zoom in, explore and apply filters for the same data, using the DimCompare view, which also supports brushing, feature selection etc., and finally users can enable cluster annotations to get detailed information about specific clusters on demand.

Scalability

Dealing with clutter in large datasets is very important, the design choices we made, like limiting the number of bars in cluster annotation, having a density aware mode, ability to toggle visual encodings, zoom aware transparency for cluster annotation to avoid occlusion, scrollable list views for feature selection for really high dimensional datasets etc. While the current system is made for small to medium size datasets, considerations like these set the stage for future scaling to large datasets.

2.7.3 Schneiderman's Mantra

Schneiderman's [57] information seeking mantra, "overview first, zoom and filter, then details on demand, gives foundational guidance for the design of the interaction system. This is implemented through:

Overview: The Cluster feature contribution glyphs and metric summaries provide immediate understanding of overall data quality and structure

Zoom and Filter: Interactive exploration in DimCompare projections allows users to focus on their regions of interest. Users can also filter the features which will be shown in the Cluster annotations.

Details on Demand: Tooltips, visibility toggles for various visual encodings and detailed statistics provide information when needed, without cluttering the display.

2.7.4 Further Concepts

Comparative Visualization The use of multiple synchronized views is common in information visualization. Gliecher et al. categorizes this as juxtaposition (side by side views), superposition (overlay) and explicit encoding of differences. DimCompare mainly makes use of synchronized juxtaposition to help users compare two projections performed using different dimensionality reduction techniques directly.

Linking and Brushing This is a basic interactivity method in exploring the data, it refers to selecting a data point in one view and highlighting the corresponding point in another view, which enables tracking of clusters and outliers across different dimensionality reduction projections. Users can make a custom cluster selection by brushing over desired points. Unfortunately current tools provide little to no support for users to explore data in this way. [58]

Glyphs and Annotations Using glyphs, which are compact visual summaries over clusters, fills the gap between 2D projections and original feature spaces. They are used to make scatterplots richer by giving more information about individual points or clusters. Glyphs can be used to encode anything, such as data labels of individual data points or even used for clusters to summarize key feature differences driving these clusters.

2.7.5 Human in the loop Dimensionality reduction

A study by Sacha et. al. [59] suggests a model for adding human interaction into the dimensionality reduction process. They found out different situations where users can guide the algorithm, like by choosing the features or tuning parameters etc. Our system fits into this model by letting the users explore dimensionality reduction results by changing features, number of clusters etc and use what they learn to drive future decisions.

2.7.6 Overplotting and density management

A huge problem in high dimensional data scatterplots is overplotting, any high density data visualization is bound to suffer from clutter. Ways to target this include use of transparency, zooming, grouping data and changing to density based views. DimCompare provides the user with the option to show contour plots, for when dealing with high density data. These highlight the high-density areas using a 2D Kernel Density Estimate (KDE) over the points. We also make use of semantic zooming and toggling data point visibility, the cluster annotations automatically start getting more transparent when zooming in to reveal to the viewer, points which were occluded by the annotation.

Combining these information visualization methods, like juxtaposition, brushing, glyphs, and cluttering solutions, is a good way of dealing with challenges that high dimensional data usually presents.This work addresses limitations of previous methods and supports a more interpretable, metric driven exploration of high dimensional feature spaces.

2.8 Existing Visualization Frameworks for Comparing and Interpreting Dimensionality Reduction

Many visualization systems have been built to help users interpret the embeddings, compare dimensionality reduction methods, and evaluate the projection quality. These systems have informed our design choices and also show what gaps remain in real situations.

2.8.1 Embedding interpretation and inspection tools

Tensorflow embedding projector is a widely used tool, which was introduced as a web interface for exploring high dimensional embeddings, using PCA, t-SNE and UMAP, it supported search, selection and neighborhood inspection [53] It was helpful in popularizing interactive projection exploration for practitioners but it provides limited comparative assessment across different dimensionality reduction methods, with no support for metric overlays.

2.8.2 DimReader

DimReader is a visual interaction framework by Cavallo [60] which focuses on explaining projections rather than just displaying them. They introduce forward and backward projection and landmarks which let users see how changes in the features move the point in 2D projection. This work is important because it links low dimension data to high dimension data which is in line with our cluster annotation/glyph idea.

2.8.3 Clustrophile2

Clustrophile2 is also a tool by Cavallo [54], which provides guided workflows for interactive clustering analysis, it integrates algorithm selection, parameter steering and visual diagnostics . This system shows how to mix semi automated suggestions with a human judging when choosing cluster models and parameters. Unfortunately it was made to explore single projections and still not for comparative analysis across different dimensionality reduction techniques.

2.8.4 Projection comparison and quality assessment

Projection inspector is a tool which provides interactive projection space, where users can move between projection methods and interpolate new layouts between methods, while also inspecting quality metrics [61]. It explains projection choice as a trade off and combines layout browsing with metric readings, which was a guide for the comparative ideas presented in the thesis like DimCompare and the metric summary in BarDar.

Distortion focused tools such as CheckViz, ProxiLens and other follow ups by Aupetit and colleagues encode false neighbors (data points that are close in low dimension but far apart in high dimension) and missing neighbors (points that are close in high dimension but far apart in the projection) and local stress directly into the visualization. [62] [51]. This work argues that users should see where the projections are unusual instead of just reading a single numeric value. The DimCompare tries to follow this philosophy, qualitative and quantitative cues and showing structure in two separate dimensionality reduction views with cluster annotations.

Surveys and frameworks have helped generate and refine the ideas that went into this thesis. Nonato and Aupetits survey links dimensionality reduction techniques, distortions in structures, task and enrichment of layouts, which motivated us in adding local quality overlays and explanations to embedding plots [63]. More recently, a review by Behrisch et al., looked at quality metrics for dimensionality reduction, providing us with guidance on which metrics capture which aspects of structure preservation. [48] which back our choices to combine visual encodings with multiple metrics rather than rely on a single score.

2.8.5 Human interaction and reliability

A structured literature analysis by Sacha et al. looks at how users interact with dimensionality reduction and proposes a process model for human in the loop projection analysis. [59] Their findings support the importance of brushing and linking, parameter steering and explanatory views, all of which are incorporated in the design of DimCompare. A newer survey by Jeon et al. argues that reliability remains a central issue and advocates for workflows where users can “see” assumptions, uncertainty and distortions. [64]

Existing systems let users browse dimensionality reduction projections, inspect neighborhoods and overlay quality cues, but these methods fall short when it comes to cluster level explanations and side by side dimensionality reduction methods comparison with unified multi-metric summaries. DimCompare fulfills the first gap by making use of floating cluster glyphs and BarDar fills the second gap by providing compact multi metric summaries that reduce mental calculation when comparing different dimensionality reduction methods.

2.8.6 What this means for our work

We adopt a side by side dimensionality reduction comparison (DimCompare) rather than a heavy superposition to reduce the clutter and while also keeping the context. We add cluster level feature glyphs to bridge the gap that exists between low dimension patterns and high dimensional explanations, which is a gap in many systems. We integrate multiple quality metrics and summarize them visually (BarDar), this along with explicit encodings reduce the cognitive load and over reliance on any single index. We acknowledge distortions and reliability concerns from prior work in this field and address them using brushing, linked selection and optional density encoding, which help reveal distortions like missing and false neighbors. This boosts interpretability and exploration in visual analysis.

2.9 Chapter summary

This chapter establishes the theoretical foundations for the thesis by reviewing the literature across high-dimensional data analysis, clustering, quality metrics and interactive visualization. The key findings of the review are:

- **Review of dimensionality reduction**: Analyzed the fundamental trade-offs between key dimensionality reduction methods (eg. linear PCA vs non linear UMAP), establishing the projection instability problem where different algorithms reveal different, and sometimes even conflicting, data structures.

- **Survey of clustering and quality metrics**: Detailed the various internal metrics like Silhouette, Davies Bouldin, S_Dbw and external metrics like ARI, NMI etc used to quantify the feature space quality, providing the vocabulary/language used by the BarDar and DimCompare tools.

- **Analysis of Visualization Principles**: Grounded the system's design in established frameworks, including the Gleichers taxonomy for comparative visualization, justifying the DimCompares juxtaposed views, and Munzner;s nested model for the interaction design of the tools.

- **Identification of Research Gap**: Surveyed existing visual analytic tools like Embedding projector, Clustrophile 2, concluding that a clear gap exists for systems that, at the same time, supports side by side dimensionality reduction comparison, multi metric dashboards and feature level cluster explanations.

This review confirmed the need for novel tools presented in this thesis and provided the theoretical and research basis for their design, which is detailed in the next few chapters.

3 Methodology

The previous chapters pointed out the need for more effective tools to evaluate the quality of high dimensional feature spaces. The limitations of plots and abstract quantitative metrics, as discussed earlier, need a solution that closes the gap between algorithm output and human interpretability. This chapter goes into the design, architecture and core methodologies of the visual analytics system we developed to tackle these issues. The system, called DatasetWiz, is created to tackle the research questions proposed in this thesis. It is a robust analytical backend and a frontend with novel and interactive visual components. The design process was guided according to established principles in visualization and visual analytics.

3.1 Industrial Context and Collaboration

The idea for this visualization tool came after an industrial collaboration with Robert Bosch GmbH, that provided the real world context, datasets, and domain expertise essential for this research. This partnership highlights the significant value of combining industry expertise with academic inquiry to solve practical challenges, and advance our fundamental understanding of visualization and data analysis.

3.1.1 Bosch Manufacturing Quality Assessment

Robert Bosch GmBH is a leading global supplier of technology and services, with automotive technology representing its largest business sector, the company's focus on quality is reflected by their investments in advanced hardware manufacturing and quality inspection. In modern manufacturing environments, quality assessment must handle intricate objects while also sustaining the high volume and consistent reliability.

The company's quality assessment challenges span across multiple domains, including automotive components which need to have accurate detection for mechanical defects, dimensional variations and material inconsistencies. Consumer products must also meet strict aesthetic and functional standards. All of these requirements create a need for flexible analytical tools that can adapt to different products and defects.

3.1.2 Current Quality Assessment Workflow

Bosch's existing quality assessment workflow combines traditional inspection methods with advanced computer vision systems. Human inspectors handle complex subjective assignments, while automated systems process most of the high volume of normal samples based on many features such as objective measurements of dimensions, type of material, which machine the part was associated with, manufacturing settings etc. These type of datasets are high dimensional; sometimes, the dataset included just the pictures of the parts, which were converted into vectors by using image feature extractors, based on deep neural networks.

These high dimensional datasets of the industrial hardware parts, are put through dimensionality reduction process, and projected onto a 2D space. This dimensionality reduction is an essential part of the quality analysis process as it gives the users simplified visualized information about the parts, as in how many clusters form, which points deviate from their clusters, which usually implies some defects, mislabelling, outliers etc, as these high dimensional datasets are hard to visualize. Cluster shapes also give insight into the effects happening on the real world parts like blurry images.

These datasets' feature spaces are evaluated for quality using a combination of clustering metrics, in addition to manual 2D projection inspection, to judge whether these datasets can be used to train machine learning classifiers downstream in the pipeline.They found out that generally a dataset with good cluster separation correlated to good accuracy for the machine learning classifier. However, their current approach was limiting because the interpretation of these results by non-technical users was still challenging, and even technical users needed to know what exactly a numeric clustering metric meant. Apart from the understanding of each metric, the intricacies of each dimensionality reduction technique should also be known by the user as projections might look wildly different according to the dimensionality reduction technique used.

This was their current approach and they were interested in evaluating the quality of feature spaces more. They wanted to find out additional metrics which can be used to evaluate the quality of these feature spaces and 2D projections, and this was the main focus of the collaboration with Bosch.

3.1.3 Collaborative framework

The collaboration with Bosch was structured to address both the practical needs and longer term research objectives. Johannes Mohren and Dr. Sabrina Schmedding served as the primary industry contacts, providing domain expertise, dataset access and validation of research directions. Regular meetings ensured that the research remained grounded in practical requirements while still pursuing novel visualization approaches.

The partnership provided access to real manufacturing datasets, that would be impossible to replicate in academic settings, while the industry partners benefited from state of the art visualisation research that could improve their quality assessment capabilities. This kind of mutually benefiting model enabled the development of tools that are both academically novel and practically relevant.

3.2 Task Analysis

Through interviews and observations with academic users and industrial real world users (Bosch quality assessment team), these core tasks were identified.

- **Algorithm comparison :** Users need to compare different dimensionality techniques effectiveness for their specific datasets, understand differences and tradeoffs between global and local structure preservation of these dimensionality reduction algorithms.

- **Pattern Discovery :** Identification of meaningful clusters and outliers in high dimensional feature spaces, with emphasis on which exact features are affecting the groupings. Ex: with the industrial users, they wanted to know which features were causing anomalous clusters in the high dimensional space.

- **Quality Assessment :** Evaluation of clustering and feature space quality of the high dimensional dataset, using multiple quantitative metrics. These metrics helped users evaluate the suitability of the dataset for downstream tasks such as training a Machine Learning model on the dataset. The metrics also reduced cognitive load, by converting the complex 2D projection into an easily digestible number.
- **Knowledge Communication:** Sharing these insights gained, across multiple teams and organisational levels, with technical or non-technical backgrounds, using easily interpretable visualisations.

3.2.1 Current Practice

The initial phase of this work was looking at the current industrial workflow and to perform a requirement analysis, keeping in mind the challenges observed in industrial data science workflows. The primary target users were data scientists and ML engineers, who are tasked to assess the dataset quality prior to training. We did an analysis of their existing practices and limitations, and identified several key user requirements. Bosch's old approach was modelled after existing traditional analysis approaches, it primarily relies on

- **Traditional statistical analysis :** Basic statistical summaries and correlation analysis, which provide limited insights into complex structures of these datasets and image feature extractor derived feature spaces.
- **Single method visualization:** Individual dimensionality reduction techniques, typically only t-SNE and PCA, are applied to the dataset without systematic comparison, or validation of the results.
- **Metric based evaluation :** Clustering quality is assessed using individual numeric metrics, without comprehensive comparison or visualization of trade offs between different approaches.
- **Manual interpretation :** Quality engineers must manually interpret the complex algorithmic results, without visualisation support, leading to potential misunderstandings and possibly suboptimal decisions.

3.2.2 Identified limitations

Several critical limitations emerged after analysing, such as

- **Need for Comparative Analysis :** Users need a way to understand the impact of different dimensionality reduction algorithms on their data, since no single method is universally optimal and cannot help understand their relative strengths and limitations for specific datasets.
- **Need for Feature level explanation:** Dimensionality reduction algorithms have a black box nature which was a core challenge. Users needed insight as to why some clusters are formed, what does this protrusion in the cluster mean for their real world data and their implications etc. Static plots have no connection back to their original data, and users had a hard time figuring out which outlier on the plot corresponded to which data point.

- **Need for Integrated Visual-Metric validation :** Visual inspection alone was not enough, and sometimes deemed to be too subjective. Multiple quality metrics were being calculated independently without visualization tools that can reveal relationships and trade offs. Metrics alone provide quantitative evaluation and validation but do not show the whole picture. In the real world data is often mislabeled, therefore even with ground truth, the metrics might fail to represent some outliers, while visual inspection can highlight these outliers.
- **Need for a Unified, Interactive workflow :** A significant pain point for the users was that they had to rely on separate, disconnected tools, e.g. separate python scripts for preprocessing, embedding, dimensionality reduction and visualizations. A primary requirement was a single, unified, seamless web based interface, that integrated the entire analytical pipeline, from data upload to final insights, while still being accessible to non-technical users.
- **Scalability constraints :** Existing tools struggle with size and complexity of industrial datasets, and hence require manual processes that are time consuming and error-prone.

3.2.3 Requirements for improvement

Based on the collaboration discussion and analysis of current limitations, we identified some key requirements that emerged

- **Comparative analysis:** Tools must enable side by side comparison of different dimensionality reduction techniques with some way of emphasizing similarities or differences.
- **Feature interpretability:** Visualization should reveal which features contribute most to the clustering patterns, allowing users to understand various drivers of different groupings.
- **Integrated Metrics:** Multiple clustering quality metrics should be presented to users, in a unified framework, which enables the user to understand trade offs and relationships.
- **Interactive exploration:** Users need the ability to explore various perspectives of/on the data, via panning, zooming, brushing and filtering operations.
- **Scalable performance:** Tools should be able to handle industrial scale datasets with thousands of samples, each with hundreds of features, with acceptable response times.
- **Intuitive user interface:** The interface must be accessible not only to data scientists, but also to people who are not familiar with methods like dimensionality reduction like quality engineers, students, ML engineers etc who have varying levels of technical expertise.

3.3 Technical requirements for visualization tools

3.3.1 Functional requirements

- **Multi-Technique Comparison :** Ability to compare results from different dimensionality reduction methods like PCA, t-SNE, UMAP and MDS simultaneously, enabling users to to understand how their choice of algorithm affects the data representation.
- **Cluster annotation :** Visual annotation of clusters, with some information about their differentiating characteristics, particularly which features differentiate each cluster from the other clusters.

- **Interactive exploration :** Support for abilities like brushing, selection, zooming and filtering operations, that enable the detailed exploration of the dataset and its subsets.
- **Metric Integration :** Comprehendable presentation of multiple clustering quality metrics with visualisation of their relationships and what it implies for technique selection.
- **Export and documentation :** The ability to export 2D projections and other visualisation/-analysis results for including in the reports and presentations, helping users communicate their findings with other stakeholders.

3.3.2 Performance requirements

Industrial applications call for some specific performance constraints

- **Response time**: Interactive operations like brushing zooming and selection must provide immediate visual feedback (<200ms) to support fluidity in exploration. The time required for the actual dimensionality algorithm will vary according to the size of the dataset and the compute hardware used.
- **Dataset Scale :** Tool must handle dataset with 1000-10,000 samples and 10-5000 features with minimal performance degradation.
- **Memory Efficiency :** Browser based tools typically must operate with memory constraints while handling bigger datasets and complex visualizations.
- **Concurrent Use :** The system must support multiple users accessing different datasets simultaneously, without interference.

3.3.3 Usability requirements

The target use base includes quality engineers, data scientists, students and domain experts who all have a varying level of analytical skillset:

- **Intuitive interface:** The interface must be learnable by the users without extensive training in advanced analytical techniques or visualization tools. The UI must make sense without any prior explanation or self-explanatory, and be visually simple, while incorporating modern web design elements, which is a design language that users tend to be familiar with.
- **Details on demand :** Complex functionality must be organized in such a way that supports both casual exploration and detailed analysis when needed, without overwhelming the new users.
- **Contextual help :** Documentation and tooltips must be integrated into the UI and should provide guidance to the user on interpretation of visualizations and metrics.
- **Error Prevention :** The interface must account for common errors and prevent them, and provide clear feedback to the user about the state of the system, communicate when user actions can not be completed, loading states etc.

3.3.4 Integration requirements

Industrial deployment requires integration with existing workflows and systems:

- **Data format compatibility :** Support for standard industrial data formats (such as simple CSV and Excel) without having to perform complex data transformations.
- **Technology Compatibility :** Browser based deployment to minimize installation, maintenance while ensuring the system is compatible with existing corporate IT infrastructure.
- **Benchmarking :** Comparison with existing analysis approaches both academic and industrial, to demonstrate improved effectiveness.

The requirements identified through this industrial collaboration form the basis for the design and implementation of the DimCompare and BarDar visualisation tools described in the following sections. The combination of needs of practical industry with research opportunities in visualization creates an ideal context for development of tools that advance both the academic understanding and industrial practice.

3.4 Iterative Design Process

The development of DatasetWiz followed an iterative and user centered design process. It began with creating initial low-fidelity mockups for the two primary views, i.e. DimCompare and BarDar. These wireframe mockups were then developed into a series of functional prototypes. Each prototype was evaluated informally, by "thinking aloud” with peers from the data science and visualization domain. This iterative feedback loop was crucial in refining the systems interaction design, visualization choices, visual encodings and the overall workflow. For instance, early feedback led to replacement of simple tooltip on clusters with the more informative and expressive “Cluster Annotation” which includes feature contributions.

Phase 1 Domain analysis: Extensive research for user requirements, pain points amongst peers and also an extensive collaboration with Bosch industry experts to understand current workflows, identify limitations and see the types of insights needed for high dimensionality data analysis.

Phase 2 Initial prototyping: Development of basic prototypes, for DimCompare as well as BarDar chart concepts, focusing on technical feasibility and core functionality.

Phase 3 Iterative refinement: Multiple cycles of prototype enhancement, based on user feedback from both the Bosch industrial partners and academic advisors, with the importance of interaction design in mind, while making the system optimized for performance.

Phase 4 Validation and Documentation: Comprehensive testing with real datasets, user feedback collection and systematic documentation of design choices and the reasoning behind them.

FIG . [ADD FIGURE OF MOCKUPS ADOBE XD]

3.5 Theoretical framework

The design of DatasetWiz is based on a user centric, iterative process that is inspired by the practical requirements of industrial data analysis, while keeping the foundations of information visualization in mind. The main objective was to move away from a passive observation of a dimensionality projection and instead actively engage in the discovery process, guided by our system.

The systems design is built on the two key theoretical frameworks discussed in Chapter 2:

1. **Gleicher et. al's Taxonomy of comparative visualization :** To address common challenges faced during dimensionality reduction, such as projection instability and algorithm bias (RQ2 and RQ4), our system relies heavily on Juxtaposition. The DimCompare view places two distinct dimensionality reduction projections side by side, enabling users to do direct visual comparisons. Further, the cluster glyphs are a form of explicit encoding, as they superimpose additional high dimensional information, directly onto the 2d space, to reveal what exactly drives cluster formation (addressing RQ1). Gleicher et. al's Taxonomy of comparative visualization : To address common challenges faced during dimensionality reduction, such as projection instability and algorithm bias (RQ2 and RQ4), our system relies heavily on Juxtaposition. The DimCompare view places two distinct dimensionality reduction projections side by side, enabling users to do direct visual comparisons. Further, the cluster glyphs are a form of explicit encoding, as they superimpose additional high dimensional information, directly onto the 2d space, to reveal what exactly drives cluster formation (addressing RQ1).

2. **Munzner's Nested Model for Visualization design :** The system's entire workflow follows Munzner's principles. The BarDar chart provides the high-level overview of clustering across multiple dimensionality reduction algorithms. .The users can then proceed to the DimCompare view, to zoom and filter on specific clusters, or regions of interest using interactive brushing and panning tools. Finally, the cluster glyphs provide details on demand, revealing high dimensional deviations in the features, for any of the selected clusters.

By consciously applying these principles, our system aims to provide a more structured and effective exploratory analysis experience, as compared to traditional, static and disconnected tools.

3.6 DimCompare

The DimCompare view is the central component of visual data exploration with DatasetWiz. DimCompare is a novel, dual-view system designed specifically to address the challenges of projection instability, information loss and the need for feature level interpretation, all of which were identified in the user requirement analysis.

3.6.1 Design Rationale

The design rationale for DimCompare was to create an environment, where users can directly and interactively compare the outputs of different dimensionality reduction algorithms. As established in Chapter 2, methods like PCA, t-SNE and UMAP have different mathematical basis and biases. This causes them to preserve different aspects of the data's structure. The primary goal of DimCompare is to make these differences transparent to the user and explorable, so that the users can build a more robust and holistic mental model of their dataset, by observing it from multiple perspectives.

3.6.2 Design Goals

DimCompare was designed to address these specific goals

1. **Simultaneous comparison :** enable side by side visualization of different dimensionality reduction technique results on the same dataset, allowing immediate visual comparison of the algorithm outcomes
2. **Feature Interpretability :** Provide clear visualization of which features contribute the most to the formation of a cluster in each dimensional representation, closing the gap between high dimensional analysis and human understanding.
3. **Interactive exploration :** Support fluid and intuitive interaction patterns that allow users to explore specific data subsets and understand relationships between the different views and finally validate algorithm results through parameter manipulation.
4. **Metric Integration :** Combine qualitative visual assessment with quantitative clustering, i.e. quality metrics, providing users with multiple angles on technique performance.
5. **Scalable performance :** Maintaining interactive, almost real-time response times on high dimensional industrial-scale datasets, while also providing analytical capabilities.

3.6.3 Need for comparative analysis

Traditional approaches to dimensionality reduction analysis focus on using a single technique, so users lack a systematic method of comparing different algorithmic approaches. This limitation is problematic in industrial contexts, where the choice of a dimensionality reduction method has significant impact on downstream tasks, such as training a Machine Learning model on it.

The challenge is made worse by the fact that different techniques may reveal different aspects of data structure, ex: PCA preserves global relationships but it may miss out some non-linear patterns. Whereas t-SNE is good at revealing local cluster structures but it can distort global organization [. UMAP attempts to balance both global and local preservation [65]. Without comparative visualization tools, practitioners resort to trial and error approaches or rely on a singular quantitative metric that might not capture full complexity of the dimensionality reduction techniques performance.

3.6.4 DimCompare Features

3.6.4.1 Dual-View architecture

DimCompare uses a side by side, juxtaposed layout to display two different Dimensionality reduction projections of the same dataset, at the same time. The interface employs a horizontal dual-view layout with two primary scatterplot panels positioned side-by-side. This arrangement supports natural left to right comparison patterns [50] while maximizing the available screen space for detailed visualization.

Each panel displays the results of different dimensionality reduction techniques applied to the same high dimensional dataset. The data points are consistent across both views and enable direct comparison of how different algorithms represent the same information, revealing technique specific patterns. Juxtaposition is a simple way to implement a comparative analysis of two different dimensionality reduction techniques, compare their projections and individual clustering tendency.

Figure 5.1: [PLACEHOLDER: Screenshot of DimCompare dual-view interface showing PCA (left) and t-SNE (right) projections of an industrial dataset with cluster annotations]

3.6.4.2 Integrated metric Visualization

Scatterplots and glyphs allow for a more qualitative exploration, complementing that, DimCompare integrates quantitative feedback directly underneath each projection view. For each of the dimensionality reduction methods (eg. PCA in View1 and t-SNE in View 2), a dedicated bar chart summarizes the clustering quality metrics calculated specifically for that projection.

Purpose : These charts provide instant, quantitative validation for the visual structures that the user observed in the scatterplots, directly addressing our research goal. Users can instantly see if the visually different clusters in a t-SNE plot correspond to a high Silhouette score, or if the more overlapping structure in a PCA plot results in a poor Davies Boulding score etc, without having to do the mental visual calculations of two similar looking projections.

Metrics Displayed : The bar charts display the same set of internal validity indices, used later in the BarDar View, which are also the ones discovered to capture different aspects of cluster and feature space quality. These are : Silhouette Coefficient, Calinski Harabasz Index and the recent S_Dbw index.

Calculation and normalization: It's important to note that the metrics are computed based on the clustering assignments from KMeans applied to the 2D projected data of the specific Dimensionality Reduction method shown in the panel. This allows us to directly assess the quality of the visual clustering that's presented to the user. Similar to BarDar view, we also normalize the scores in DimCompare view, to a common range of [0,1], and the metrics where lower values are better (like Davies Bouldin and S_dbw) are inverted, to ensure that taller bars consistently indicate better quality, according to that specific metric, adding to the intuitiveness of the design.

Contextual Information : Hovering on the bars of the chart, tooltips appear that show the Normalized and also the Raw metric value. There are also brief textual descriptions below the metric names, that clarify what each metric measures specifically (eg. "Cluster Separation“ for Silhouette, “Cluster similarity” for Davies-Bouldin etc). This adds further intuitiveness and lowers the technical knowledge bar that's required to use the tool.

This tight knit integration of visual representation and quantitative metrics within each panel of the DimCompare view allows for users to make more informed judgements about the quality and the trustworthiness of each of the Dimensionality Reduction projection.

3.6.5 Coordinated Views Implementation

The dual view architecture is supported by the shared data model, this enables tightly coupled interactions. The two scatterplots are coordinated through several mechanisms

Brushing and linking : The two views are linked with each other with interactive brushing. When a user selects a group of points from one scatterplot using the brush, the corresponding points are immediately highlighted in the other. This interaction is critical for addressing RQ4, since it allows the user to track how specific clusters, outliers and structures are represented across various dimensionality reduction algorithms. For example the user can see if a cohesive singular looking cluster in a UMAP projection is subdivided into multiple smaller groups by t-SNE, or how outliers behave under different projections.

Synchronized color coding : Data points representing the same samples are colored the same across the two views, which enables immediate tracking of individual samples or groups across different dimensional representations.

Figure 5.42: The DimCompare dual-view interface, showing a PCA projection (left) and a t-SNE projection (right) of the same dataset. A brushed selection in the left view is highlighted in the right view, demonstrating the linked interaction.

3.6.5.1 Visual encoding decisions

Numerous critical visual encoding choices enhance the efficacy of the dual view comparison.

Point representation : Individual data points are represented as semi transparent circles, which help to reduce visual clutter. The points also are highlighted with an outline when the user hovers on it to display tooltips, which help the user see which exact data point is being investigated, especially in dense clusters with overlapping points.

Color Coding : A color palette is used to encode the cluster membership, with use of consistent colors across both views, which enables an immediate comparison of patterns. The palette was selected to give priority to perceptual differentiation and accessibility for the colorblind users.

Density representation : Due to the nature of high dimensional datasets, the scatterplots tend to be very visually dense, in such cases, it can be helpful to enable the density view which shows colored contour lines for clusters, instead of points, which helps approximate general shapes and gain additional insights.

3.6.6 Feature contribution glyph

The most novel element of the DimCompare view is the Feature Contribution Glyph aka Cluster Annotation, it's an information dense visual summary of a cluster, that was designed to overcome the inherent information loss that comes with using dimensionality reduction methods. It directly addresses the RQ1 (How can visualization reveal which features drive cluster formation?). When a clustering algorithm is applied, a cluster glyph or cluster annotation is generated and displayed for each identified cluster.

Calculation: To see what makes clusters different, the cluster annotation visualizes the statistical deviations of the points inside the cluster, to the rest of the dataset. For all the user-selected high dimensional features, it calculates two values. (i) The mean of that feature for all data points within the selected cluster and (ii) The mean of that feature for all the data points outside of the cluster (the "global background"). The percentage difference between these two means is calculated.

Cluster–background deviation.

Let $X \in \mathbb{R}^{n \times p}$ be the data matrix, $C \subset \{1,\ldots,n\}$ the index set of the selected cluster, and $B = \{1,\ldots,n\} \setminus C$ the background. For feature $j \in \{1,\ldots,p\}$, define the cluster and background means

$$\mu_{Cj} = \frac{1}{|C|} \sum_{i \in C} x_{ij}, \qquad \mu_{Bj} = \frac{1}{|B|} \sum_{i \in B} x_{ij}.$$

The (signed) percentage deviation visualized by the glyph is

$$\Delta_j = \begin{cases} 100 \times \dfrac{\mu_{Cj} - \mu_{Bj}}{|\mu_{Bj}|} & \text{if } \mu_{Bj} \neq 0, \\ 0 & \text{if } \mu_{Bj} = 0 \text{ and } \mu_{Cj} = 0, \\ 100 & \text{if } \mu_{Bj} = 0 \text{ and } \mu_{Cj} \neq 0. \end{cases}$$

In vector form for a set of user-selected features S, let $S_0 = \{j \in S : \mu_{Bj} = 0\}$ denote features with zero background mean. Then

$$\Delta_j = \begin{cases} 100 \cdot \dfrac{\mu_{Cj} - \mu_{Bj}}{|\mu_{Bj}|} & j \in S \setminus S_0, \\ 100 \cdot \mathbf{1}[\mu_{Cj} \neq 0] & j \in S_0. \end{cases}$$

Visual encoding : The percentage difference is encoded as a compact bar chart within the glyph. A bar extending upwards from the central horizontal axis indicates a feature whose average value is higher within the cluster compared to all the other data points. Whereas a bar extending downwards indicates a feature with a lower average value. The length of the bar is directly proportional to the magnitude of this percentage difference, allowing users to immediately identify the most discrimination features of that specific cluster.

3.6.7 Feature Contribution Glyph Visual design

Each annotation consists of a compact, regular visualization positioned near the centroid of each cluster. A cluster annotation contains the following:

1. **Feature Bar Charts :** Horizontal bars represent the top 3-5 most distinguishing features for that cluster, with the length of the bar used for encoding the magnitude of the difference from the global mean of that feature.

2. **Directional Encoding:** Bars extending up from the horizontal baseline (x-axis) indicate above average feature values, whereas the downward bars indicate below-average values, providing immediate directional information/encoding. For extreme cases, when a bar extends above the cluster annotation box due to high or low values, we add a pointed cap on it to indicate that it goes above/below the bounds for the cluster annotation box.

3. **Percentage Labels:** Actual numeric percentage differences are displayed alongside each bar, which enable for a quantitative assessment of feature contributions.

4. **Color Coding:** An intuitive color scheme, green for above average and red for below average, allows for rapid pattern recognition and reinforces the directional encoding of the bars.

5. **Sample size :** This is a small indicator on top of each cluster annotation which shows the size of the cluster, eg. n=184, where 184 are the number of data points belonging to that cluster. This helps users get a clearer idea of what the cluster contains

[FIG of individual cluster annotation]

3.6.8 Spatial Placement Rationale

The feature contribution glyphs are positioned at the centroid of each of the clusters rather than in a separate list or sidebar. This choice was made since it maintains the spatial context between the 2D projection and the high dimension feature explanation. This selected layout adheres to visualization principles and coordinated multiple views, by allowing the user to inspect the statistical "why”s of the cluster without needing to move away their gaze from the visual structure that they are currently exploring. To reduce potential visual clutter, we implemented zoom level depended transparency, where the annotations fade as user zooms in to inspect the individual data points.

3.6.9 Dynamic Glyph Updates

The visual annotations dynamically update in response to the user’s interactions

Selection Based : When the user selects the subsets of data points from the projection using brushing, temporary cluster annotation appears on the selection, showing the feature characteristics of the selected subset, compared to the rest of the data.

Real Time calculation : Feature statistics are calculated in real time, according to changes in the users selection area, allowing for immediate feedback about the characteristics of different regions/ points of interest.

Contextual Comparison: The cluster annotations display comparison between the selected cluster and the rest of the dataset (global), between two clusters and also between the selection and the remaining data, depending on user interaction.

3.6.10 Interactive features

DimCompare also has a whole suite of interactive tools to help reduce visual clutter, and support a fluid analysis workflow.

Visual Toggles : Users can independently toggle the data point visibility, density contours (visualized using a 2D Kernel Density Estimate) and the cluster annotations themselves. These features allow the users to customize the view for different analytical tasks like finding the high level patterns using the density contours first, then perform detailed inspection of individual points by enabling the data points visibility.

Feature selection : A dedicated panel displays the features in the dataset, users can interactively select which of the high dimensional features are included in the Feature Contribution Glyph. The features are first ranked

Selection based updates : When a user selects a subset of data points through brushing or clicking, temporary annotations appear on that subset of data that show feature characteristics of user selection compared to the rest of the data, supporting investigative exploration.

Shift + Drag Brushing activation : Brushing is activated using shift+drag, this is done to ensure that users dont accidentally initiate brushing with a single click.

Real time calculation : Feature statistics are recalculated in real time whenever users decide to change selection regions etc, which enable immediate feedback about the characteristics of different subsets of data which the user is interested in.

3.6.11 Implementation details

The following JavaScript snippet illustrates the calculation of the percentage deviation Δ_j and the subsequent rendering of the glyph for the top five features:

```
// Simplified glyph calculation and rendering
function updateClusterGlyph(clusterId, features, clusterData,
    globalData) {
    const featureImportance = calculateFeatureImportance(
        clusterData, globalData, features
    );

    const topFeatures = featureImportance
        .slice(0, 5)  // Top 5 most important features
        .map(f => ({
            name: f.name,
            difference: f.clusterMean - f.globalMean,
            percentage: ((f.clusterMean - f.globalMean) / f.
    globalMean) * 100
        }));

    renderGlyph(clusterId, topFeatures);
}
```

Listing 3.1: Glyph calculation and rendering logic.

3.6.12 Zoom and Pan operations

Dimcompare supports multimodal aka multi input navigation, which allows for different exploration patterns inside the 2D projection, which usually are not supported in traditional dimensionality reduction workflows, where images are rendered as images though a visualisation library like matplotlib.

- **Mouse wheel zoom:** Standard scroll wheel zooming enables rapid change of focus with the cursor position used to orient the zoom center.
- **Middle mouse pan :** Middle mouse press and drag provides pan operations without switching of modes
- **Touchpad controls :** In addition to the mouse, touchpad controls are also supported in DimCompare, users can pinch in and out to zoom and a three finger drag to pan around the projection.

- **UI Button controls :** Explicit zoom in/out buttons are also provided for alternate navigation, users who prefer interface based controls and accessibility.
- **Independent View navigation :** Each of the 2D projection views can be navigated independently, which allows for exploration of technique specific patterns and still maintain the ability to compare overall structures.

3.6.12.1 Toggle controls

Multiple display options enable customization based on the user's analysis needs, these toggle options also help in reducing clutter, which is a common problem when dealing with high dimensional CSV datasets, as discussed in Chapter 2.

1. **Cluster glyph toggle :** Users can show or hide cluster glyphs, to focus on overall patterns or detailed cluster characteristics.
2. **Density toggle :** The colored concentric density visualizations can be turned on or off, these density representations help users get an overview of the structure of the clusters, because otherwise, individual points can clutter easily and start to overlap at some specific zoom level.
3. **Point Display Toggle :** Individual points can be hidden to focus on the densities, cluster level patterns, this is particularly useful in the cases where the dataset is large and the point density creates visual clutter

[ADD FIGURE]

3.6.13 Detailed information display

DimCompare provides many mechanisms to provide users access to detailed information, which is in accordance with the Schneidermanns mantra, which says 'Overview first, zoom and filter, then details on demand'

Hover tooltips: Mouse hovering over the individual data points, displays additional information about the sample, including the original feature values for selected features and also the cluster membership of the displayed point.

Bar chart hover : In the metrics bar chart shown below the dimensionality reduction projections, hovering on the individual metric bars shows the real and normalised values allowing users to have a more in-depth look

Feature selection : The feature selection tab, which is arranged according to feature variation, also shows more information when hovered over it like rank, raw value, log scaled value etc, which is helpful when deciding which features to select for visualisation.

3.6.14 Design Validation and user feedback

The final designs for DimCompare (and also BarDar) visualizations were not arrived at in a single simple step. They are the result of an iterative design process, guided by feedback from weekly academic supervision, expert consultation with industry partners at Bosch, and an analysis of the preexisting principles in visualization research.

The DimCompare view evolved from a simple scatterplot at first, to an interactive dual view analytical tool environment. It was shaped by these primary design challenges and feedback

Challenge 1 Explaining cluster formation (the "Why")

Initial problem : A standard dimensionality reduction scatterplot can show that clusters exist (ii.e. The "what), but it cannot and does not explain why they formed. The link to the original high dimensional feature space is also lost. This was also a core requirement of our initial research questions.

Initial design and feedback: Early mockups of the dimcompare have a cluster annotation which is explored using arrow based glyphs to show cluster deviation. Feedback from supervisors indicated that comparing the lengths and and directions of the multiple arrows was confusing, perceptually difficult and looked cluttered. The arrow direction was meant to encode if the feature is more than average feature values for other clusters or less than using up and down pointing arrows.

Solution : We iterated the cluster annotation design, this component is a compact bar chart that is placed at the centroid of each cluster. This bar graph annotation is used to explicitly visualize the statistical deviation (percentage difference) of that cluster's feature means compared to the global average of the entire dataset. This directly addresses the question of 'what makes this cluster different from the other ones'

- Instead of arrows, we use bars that extend above the 'average' horizontal baseline, these bars going above this indicates higher than average value within that cluster
- Bars extending below this baseline indicate lower than average feature values
- The length of the bar is proportional to the magnitude of this deviation, this makes the most important features immediately apparent.

Challenge 2 Visual Clutter and Annotation placement

Problem: with multiple clusters, the new annotations would inevitably overlap each other, obscuring the data points as well, especially when zoomed out a lot. **Feedback :** Placing annotations in a separate list was a suggestion (by research assistant Anja) that would make them easier to compare but would break the spatial link to the plot. Placing them in the centroid (suggested by Prof. Dr. Heinzl) maintained the link but worsened the clutter.

Solution (hybrid approach) : We retained the centroid placement to keep the crucial spatial context and to solve the clutter problem, we implemented three refinements that are based on this feedback:

- Zoom based scaling : The annotations inversely scale with the zoom level. As the user zooms in, the annotations shrink and fade a little, becoming semi transparent to reveal the points and structures underneath it.
- Toggle visibility : A main toggle for 'Show cluster annotation"was added, allowing the user to hide all the annotations and see a clean overview.

- Annotation design : Instead of coloring the entire annotation box background with the corresponding cluster color (which can cause contrast issues), rather the bars are colored according to the cluster color to indicate the annotation belongs to which cluster, giving an additional cue in addition to the position of the annotation, when things get dense.

Challenge 3 Misleading cluster shapes

Problem: Early prototypes used cluster hulls, a solid colored polygon that connects the outermost points of the cluster.

Feedback : Prof. Heinzl noted that these hulls were misleading, since they imply a uniform density and a sharp boundary which does not exist in the real data.

Solution (Density view) : We replaced the static hulls with a "Show densities" toggle. When it is activated, it renders a 2D Kernel Density Estimate (KDE) contour map for each of the clusters. This provides a more accurate representation of cluster shape and their distribution, clearly indicating where data is dense and where it fades out.

Challenge 4 Rigid vs fluid selection

Problem: The initial annotation only worked for pre-coimputed clusters, e.g. from a KMeans algorithm. This did not allow the user to investigate any arbitrary group of points. For example a few interesting outliers, a region where clusters overlap or some unusual protrusion from the cluster.

Feedback : This limitation was identified through discussions with Bosch industry personnel, Thesis supervisor meetings and peer discussions. Which highlighted the need for a more fluid, custom, interactive and user-driven exploration.

Solution (Brushing and selection) : We implemented a Selection mode (Shift) brushing feature, which allows the users to click and drag on any group of points in the scatterplot. The system can then instantly generate a temporary box around the selection to indicate bounds of user selection and a temporary cluster annotation is generated in the center of this new user selected box, complete with its own feature deviation bart chart. This supports for users to rapidly test their hypotheses and makes the tool a truly exploratory one as compared to a descriptive tool.

Challenge 5 Iterative selection of Metrics

Problem : Initial prototypes of DatasetWiz used the Calinski-Harabasz (CH) index as a measure for quantifying the cluster separation. However, expert consultation and iterative testing across diverse datasets and feedback showed that the CH index produced raw values ranging from 10^1 to 10^5. This high degree of scale sensitivity made normalization across different datasets impossible, and often resulting in skewed radar polygons that favoured datasets with higher sample sizes rather than better structural quality.

Solution : The CH index was remove to prevent these misleading performance rankings . While the Davies Bouldin and S-Dbw are also unbounded (0 to infinity), empirical analysis showed that those two metrics have a natural practical range, typically withing the [0,3] range for standard dataset distributions. This allowed us to implement a fixed cap normalization strategy, ensuring that the metrics still look balanced perceptually in the BarDar component. To fill the gap that was left after removing the CH index, we integrated Trustworthiness, a bounded range [0,1] metric, which is used to validate dimensionality reduction fidelity.

Challenge 6 Scatterplot axes removal

Problem: The scatterplots had axes labellings that were not common between two dimensionality reduction methods, this added unnecessary confusion when trying to make sense of the two scatterplots. Axis labeling are somewhat helpful when it comes to DR methods like PCA, but in other non-linear methods like t-SNE, they are effectively meaningless since the scale and numerical values on the X and Y axes do not correspond to any physical unit or original feature value.

Solution : We intentionally removed the numerical labels and gridlines from the scatterplot axes. This design choice was based on theoretical understanding that the absolute coordinate values in non linear DR are uninterpretable. Including these axes can lead to an introduction of trust gap, where users over interpret the meaning of X and Y positions. By removing the labels, we shift the users towards the topological structure of the clusters, which are the primary information preserved by the DR algorithms.

3.7 Chapter Summary

This chapter presented the new visualization method DimCompare, a novel, dual view interactive system that provides qualitative exploration and comparative analysis of the dimensionality reduction techniques. The key contributions are

- **Dual view Juxtaposition :** Enables direct, side by side visual comparison of two different dimensionality reduction projections, ex t-SNE vs PCA, allowing the users to identify the structural differences and algorithmic impact.
- **High dimensional Cluster Annotation :** Introduces a novel cluster annotation that explicitly encodes high dimensional feature deviations (i.e. the "why") directly onto the 2D scatterplot clusters. This fills the gap between the low dimensional projection and the original high dimensional feature space.
- **Brushing and Selection :** Implements feature which allows users to manually select data points to generate a temporary cluster annotation for that specific selection, which allows for a deeper inspection and exploration.
- **Integrated metric validation :** Provides dedicated metric bar charts directly underneath each of the 2D projections, offering immediate quantitative validation that links what the users perceive visually to concrete clustering metric scores.
- **Fluid Exploratory Interaction :** Supports a flexible analytical workflow through toggles for densities, data points, and cluster annotations, as well as a selection mode that allows users to analyze arbitrary, user defined selections on the fly, all while keeping the UI and visualization clutter free.

The next chapter introduces BarDar, a complementary tool that is designed for the high level quantitative comparison of these dimensionality reduction methods, using an aggregated, multi metric score.

3.8 BarDar chart

BarDar is the second major contribution of this thesis, it is a novel composite visualization. It intends to tackle the limitations of traditional radar charts with barcharts to deliver a comprehensive assessment of clustering quality metrics across multiple dimensionality reduction techniques. The name BarDar reflects this integration, combining "Bar” (bar chart) and “Dar” (radar chart) into a unified visualization approach.

3.8.1 Motivation and design goals

3.8.1.1 Limitations of traditional radar charts

Radar charts (sometimes also known as spider charts or star plots) have long been used to display multivariate data in two-dimensions and multidimensional comparison, particularly in the contexts where many attributes are addressed simultaneously. Despite their popularity in games and sports and its intuitive appeal, radar charts suffer from many limitations which are well documented in the literature [66], which makes them problematic for clustering quality assessment in high dimensional data analysis.

Area Bias : The enclosed area of the radar charts can be sometimes misleading as it may not accurately represent overall performance, and can be influenced by high or extreme values in the individual dimensions. Two methods that have very different result and performance profiles may include similar looking areas, or the other way around. The issue is made worse by the fact that the calculations in the circular or polar shape are also sensitive to the order in which the metrics are placed, the same metrics arranged differently will yield different areas. Human brains are not very adept at calculating minute area differences visually [52].

Comparison Difficulty : When two radar charts are observed side by side (juxtaposed), it is difficult to compare the areas or shapes, even when they are placed on top of each other (superimposed) the problem is only solved slightly, due to visual clutter and intersecting patterns, which make it difficult to see which technique is better overall. Overlapping polygons create complex visual patterns that require significant cognitive effort to interpret. If the number of compa red techniques goes above 3, the visualization rapidly starts going into the unreadable territory.

Perceptual Issues : Human perception of angles and areas in polar coordinates is less accurate than in rectangular coordinates, which can give rise to misinterpretation of relative performance. Research in graphical perception has demonstrated that humans are way better at judging position along a common scale, like in bar charts,than judging at angles or areas [52]

Scale sensitive : Different metrics operate on different scales, and with different scales/orientation, meaning sometimes higher is better and sometimes lower is better. This needs careful normalization that can sometimes obscure differences. Without doing the proper normalization, metrics with bigger ranges will take over and dominate the visual appearance, whether or not they are actually relevant in importance/interpretability.

3.8.2 BarDar Design Innovation

BarDar addresses these limitations through a mixed design that combines multidimensional overview capability of radar charts and the precise comparison aspect of bar charts. This composite/integrated approach leverages the strengths of both types of visualization while addressing the weaknesses.

Radar component: This provides a visual overview of the technique performance across multiple metrics immediately and enables outlier identification and pattern recognition. Users can quickly see which metrics favour which of the techniques and spot unusual performances, hence making radar charts valuable in exploratory contexts. [67] [68]

Barchart Component: This gives precise, quantitative comparison of aggregated performance and allows for clear ranking and selection of optimal techniques. By explicitly calculating and visualising the aggregate score, the horizontal barchart removes the cognitive burden of comparing irregular polygonal shapes.

Integrated Interaction : Coordinated Interaction between both the components enables users to explore detailed metrics and overall performance ranking. The selection and the highlighting feature applies to both the views simultaneously. Axes can be reordered in the horizontal bar chart and it reflects in the radar chart order and overlapping.

This design follows Gleicher's taxonomy of comparative visualization , by making use of both, superposition (i.e. the overlaid radar polygons) and explicit encoding (i.e. the bar chart representation of aggregate scores. This addresses the challenge of multi-feature comparison in a fundamental manner.

3.8.2.1 Theoretical justification for composite integrated design

The BarDar chart is designed as a composite integrated visualisation so as to overcome the well documented perceptual limitations of radar charts. While radar charts provide an effective visual signature for each of the algorithms, humans are very poor at accurately comparing the areas of irregular polygons.

According to the Cleveland and McGills [52] study and hierarchy of graphical perception, humans are more accurate at judging position along a common scale (bar charts) than judging ares or angles (radar charts). By explicitly encoding the aggregate performance in an adjacent bar chart, BarDar reduces the cognitive load required to rank techniques while still preserving metrics using the radar polygons.

3.8.3 Design Rationale

While radar charts are frequently criticised in the visual literature, most notably in the study by Feldman [66], where they argue that the polygon area is an unreliable metric, due to it changing based on the ordering of the axes. The implementation in DatasetWiz is designed to reduce these pitfalls through the use of a composite interactive approach. Instead of using the chart as a static ranking too, it is used as a signature, while delegating the quantitative precision to an integrated bar chart. The rationale for this design is based on three considerations:

Mitigating Occlusion using interactivity: A primary weakness identified in an earlier section is the visual clutter caused by superimposing polygons. To address this issue, BarDar makes use of translucent fills combined with high opacity hulls. It employs active z-order management, i.e. by dragging bars in the adjacent component, the user can dynamically bring the polygons that are of interest to them in the front of the rendering stack. In addition to this, users can also toggle polygons of DR methods on or off.

Priortizing the accuracy of perception : By following the hierarchy of graphical perception established by Cleveland and McGill [52], humans are significantly more accurate at judging positions across a common scale as compared to comparing irregular areas. Hence, BarDar avoids this area bias mentioned previously. During the iterative design process (see Challenge 2), we rejected the use of the shoelace formula for area calculation in favor of a mean normalized metric score. This explicit encoding in the bar chart provides a robust, order independent ranking that simplifies the user's cognitive task.

Shape as a quality signature: By decoupling the "ranking" task from the radar chart, the polygons serve their most effective purpose: pattern recognition. Instead of asking the user to calculate "which area is bigger," the radar component allows experts to identify the shape signature of a technique. For instance, a user can quickly spot if a dimensionality reduction method performs exceptionally well on "Silhouette Score" but poorly on "S_dbw" simply by the "spike" or "dip" in the polygon, regardless of the total area.

3.8.4 Design goals

BarDar visualisation was designed with these goals in mind:

- **Comprehensive metric overview:** Display multiple quality metrics all at once in a way that reveals both individual metric values and overall patterns.
- **Comparative ranking :** Enable immediate identification of which dimensionality reduction technique gives the best overall clustering quality for a given dataset.
- **Detail on demand :** Allow access to individual metric values, while also providing an aggregate summary, supporting detailed and also overview analysis.
- **Interactivity :** Support exploration through click and drag, reordering axes, tooltip popups etc.

3.8.5 Radar Chart

3.8.5.1 Metric selection and normalization

The BarDar chart shows four main clustering quality metrics that provide complementary perspectives on cluster structure. These metrics were selected based on their established use in clustering literature and their ability to capture different aspects of cluster quality.

Silhouette Score : It measures how well separated clusters are from each other, by comparing intra-cluster togetherness with inter cluster separation. The values range between -1, which indicates poor clustering, to +1, which indicates excellent clustering, and values above 5 generally indicate a good cluster structure.

Davies Bouldin Index : It evaluates the average similarity of each cluster with its most similar cluster, where lower values indicate better clustering, For visualization purposes, DBI is inverted during normalization to ensure consistent interpretation, of higher always being better. This metrics is also sensitive to cluster overlap.

S_dbw index : This is a more recent validity index which tended to outperform other clustering indices on real and synthetic datasets [CITE original SDBW PAPER]. Lower values indicate better clustering, and also like DBI, this metric is inverted for display. S_dbw is effective at detecting overlapping clusters and assessing density based cluster structures.

Trustworthiness : Unlike the other indices which focus solely on the 2D cluster structure, Trustworthiness evaluates the quality of the dimensionality reduction itself. It measures the extent to which the *k*-nearest neighbors of a point in the high-dimensional space are preserved in the low-dimensional embedding. The values range from 0 to 1, where higher values indicate that the local structure is accurately being maintained and that the projection has not introduced any "false neighbors" (points that appear close in 2D but are distant in high-dimensional space). This metric is essential for comparing DR techniques as it directly quantifies the reliability of the visual representation.

All the metrics are normalized onto a 0-1 scale to enable meaningful comparison on the same visual axes.

3.8.6 Visual Encoding

The radar chart uses several visual encoding strategies and mechanisms to maximize clarity and interpretability

Axis layout : Metrics are positioned at equal intervals of 90° apart, around a circle, forming a squarish arrangement. Text labels indicate the metric names and small text annotations clarify whether higher or lower metric values are better.

Polygon Render : Each dimensionality reduction technique (PCA, Tt-SNE, UMAP, MDS) is represented by a distinct colored translucent polygon, which connects the metric values onto each axis. The vertices here represent normalized metric scores, and the resulting shape of the polygon provides visual performance profile of the technique.

Grid lines : Concentric circles at positions at 0.2, 0.4, 0.6, 0.8 and at 1.0, provide values references, without visual clutter. Lines are made up of dotted, thin, subtle light grey colors, which provides structure, without competing for visual attention with the colored polygons laid on top. The gridlines enable approximate reading of individual metric values directly from the radar chart, without the need for interaction.

Color palette: The color scheme selected ensures a clear distinction between the techniques, while also being accessible to colorblind users. The color assignments follow a consistent convention throughout the system:

- **PCA :** Blue (#3b82f6)
- **t-SNE :** Orange (#f59e0b)

- **UMAP :** Green (#22c55e)
- **MDS :** Red (#ef4444)

These colors were selected using a qualitative method, that maximizes the perceptual distinction and also maintains sufficient contrast for users who have colored vision deficiencies.

Integrated Legend : A legend is positioned to the top right of the radar chart, in a negative space, it provides information regarding technique name and the corresponding associated color of that technique. The legend also supports interaction, where hovering over an item in the legend, temporarily highlights the color and puts it on top of other polygons, allowing for better clarity and reducing visual confusion.

Figure 6.1: [PLACEHOLDER: Screenshot of BarDar radar chart showing overlaid polygons for PCA, t-SNE, UMAP, and MDS with normalized clustering quality metrics. Annotations highlight key features: concentric gridlines, metric axes with labels, colored polygons with semi-transparency, and interactive legend.]

3.8.6.1 Multiple technique/radar overlay

The radar chart can simultaneously display up to 4 different dimensionality reduction techniques, BarDar makes use of several carefully designed, visual strategies that maintain clarity given the potential for visual clutter with these many techniques.

Transparency : Polygons are translucent with opacities ranging between 0.2 and 0.3, to enable visibility of the overlapping regions/radars, while also maintaining the distinction between colors. The specific opacity value was determined through iterative testing to balance visibility of the individual polygons and the ability to notice overlaps.

Stroke: Each polygon has bold (2-3 pixel wide) borders or stroke emphasis, which ensures polygon visibility even through extensive overlaps. The strokes are made up of the same colour as the polygon, but it's at a higher, full opacity. Doing this creates a visual hierarchy where the boundaries of each polygon remain defined, even if the fill region take on a complex shape.

Interactivity : The rendering order of multiple polygons affects which polygons appear on top and which appear on bottom. Bardar uses z-order management to bring forward and highlight the radar chart of the technique that the user hovers over in the legend. The interested polygon is brought forward in the front of the rendering stack. The BarDar also employs interactivity by allowing users to reorder the rendering order of the radar charts of various techniques, by drag and drop reordering of the actual bars in the barchart component of the BarDar chart. These changes are also reflected in the order of the radar charts, where the charts are re-rendered according to the sequence set by the user.

3.8.7 Barchart Integration

The overlay design of the radar charts follows the 'superposition' principle from the Gleichers visualization taxonomy. It also augments it with the interaction and the explicit encoding via the Barchart component (which will be discussed now) to address the inherent limitations of purely superposition based visualization approaches.

3.8.7.1 Score Aggregation :

During the development and the user testing, we also experimented with a polygonal area based approach for calculating which method performed best. The area of the polygon can be calculated easily using the shoelace formula but it was soon apparent that the area does not encode meaningful information, and it is documented to be a limitation of radar charts sensitive to the order of the axes. [66] Our task is also of comparative ranking rather than shape analysis, and since the radar chart already provides a visual representation of the performance profile, we chose mean normalized metric score as our aggregation for its interpretability and robustness.

According to visualization literature [55] we should choose the most effective encoding for your task, which is to rank the dimensionality reduction techniques. Gleicher also states that explicit encoding of comparisons reduces the cognitive load. The barchart is an explicit encoding to simplify the radar charts complexity, so it should use the most interpretable aggregation.

The techniques are automatically ranked by their aggregate performance, with the bars ordered from the highest to the lowest performance by default. The user can rearrange this ordering according to their analytical and exploration needs.

3.8.7.2 BarChart Design

The barchart part of BarDar makes use of familiar design conventions, while seamlessly integrating with the radar chart component, to create a unified experience

Horizontal orientation : Horizontal placement of the bars rather than vertical, allows for easy reading of technique name and performance values. Each bar extends from a common baseline axis (of score = 0.0), to the right side, with the length of the bars for each technique proportional to the aggregated score for the datasets 2D projection on that metric. Horizontal was chosen over vertical because the technique names are easier to read and the general consensus is that horizontal barcharts are more suitable for nominal type of categorization (technique names like PCA, t-SNE etc), whereas vertical barcharts are more suitable when you have ordinal information like (1-5 years, 5-10 years, 10-15 years etc) [55]

Consistent color coding : The bar colors match their corresponding radar chart poly gons, providing immediate visual connection between the components. This color mapping serves as a visual link that helps users track techniques across the two representations. Then the user sees an interesting polygon pattern in the radar chart component, they can immediately locate the corresponding bar without having to read the labels.

Value labels: the precise percentage values are displayed at the end ofg each of the bars to provide some quantitative assessment, These labels just show the aggregated technique scores to 3 decimal places. (eg “0.927”). This enables precise comparison even when two bar lengths are visually similar. The labels are positioned just outside the bar, to avoid occlusion with the colored bar itself.

Bar Spacing : The bars are spaced comfortably, with 40-50% of the bar height used as the vertical padding between bars. This ensures that theres no label overlap and helps creates clear interactive targets, making it easy to click specific bars.

Sorting option : The bars can be sorted by performance values or how the users see fit, using the drag and drop functionality. They can simply put the technique they are interested in at top, and "discard"/disregard irrelevant techniques by putting it at the bottom.

Figure 6.2: [PLACEHOLDER: Screenshot showing the complete BarDar visualization with radar chart (top) and integrated bar chart (bottom). Annotations highlight: matching colors between radar polygons and bars, value labels on bars, sorting toggle, and coordinated highlighting across both components.]

3.8.7.3 Layout and positioning

The spatial arrangement of the radar and the bar chart components was carefully considered to support natural viewing an intuitive user experience.

Horizontal layout : The radar chart is positioned on the left side of the screen while the bar chart is positioned to the right side of the screen. This arrangement follows the natural reading flow of left to right (in most cultures). The detailed multidimensional view is encountered first, on the left side, preceded by the aggregate simplified summary on the right. Initially the layout was supposed to be vertical with radar on top and barchart below, but we found out that users had to scroll down to have a look at the summary, which defeated the purpose of having a quick glance summary, therefore we opted for a design where both of the views are visible at once.

Proportional sizing : The radar chart occupied the 60-65% of the horizontal space, whereas the barchart element occupied around 35-40%. This sizing reflects the relative information density and the importance of the visualization elements. Since the radar chart contains more information (individual metric values, number of techniques and also the visually complex polygons), while the barchart only provides summary information in the form of simple horizontal bars which do not need a large space to reside.

Responsive design : The UI is set up in a way to support responsive design, meaning the margins and the paddings automatically adjust according to the web browser window width. The design looks good even on different types and sizes of displays like laptops, monitors, tablets and even mobile.

3.8.7.4 Drag and Drop reordering

A novel interaction feature enables reordering of techniques in the bar chart dynamically, aimed at supporting the exploratory comparison patterns that the users may find valuable.

Drag handles : Every bar has a subtle three vertical notches, which suggests the bars can be moved, upon hovering the mouse over the bars, it changes to a hand grab icon further indicating that they are draggable, following the principle of discoverable interfaces.

Visual feedback : When user starts a drag operation, many visual changes occur to provide clear feedback The dragged bar detaches from its original position and follows the cursor Bar gets a drop shadow to indicate separation by creating elevation Other bars animate smoothly to their new positions as the new dragged bar moves past them

Reordering in the Radar Chart : When the bars are reordered, the corresponding radar chart polygon layers are also reordered to bring the selected technique to the front. This ensures that when user focuses on a particular technique by manipulating its associated bar, that technique's polygon becomes fully visible on top even if it was earlier partially occluded by other polygons.

Persistence : The reordering persists until the user resets the page or applies a new order, allowing the users to arrange techniques in custom orders that support their specific analytical questions, e.g. grouping linear vs non linear methods together to compare performance etc.

Figure 6.3: [PLACEHOLDER: Sequential screenshots showing drag-and-drop reordering interaction: (1) user hovers over bar, showing drag handle, (2) during drag with bar elevated and insertion line visible, (3) after release with bars in new positions.]

3.8.8 Metric normalization strategy

3.8.8.1 Normalization challenges

Different clustering quality metric properties presented several normalisation challenges that need to be addressed to create effective visualisations

Scale differences: Clustering quality metrics operate on vastly different scales, which complicates direct visual comparison:

- **Silhouette Coefficient:** $[-1,+1]$
- **Trustworthiness:** $[0,1]$
- **Davies-Bouldin Index:** $[0,\infty)$ (typically $[0.3,3.0]$)
- **S_Dbw:** $[0,\infty)$ (typically $[0.1,2.0]$)

Note: The Calinski-Harabasz Index was initially considered but removed due to its unbounded scale $[0,\infty)$ making normalization misleading and causing it to dominate visualizations.

Without appropriate normalization, metrics like the Calinski-Harabasz index would dominate the visualization purely due to their much larger values, independent of their actual analytical importance.

Different optimization directions : Some metrics are optimized by maximisation, (like Silhouette score and Calinski Harabasz index) whereas others are optimized by minimization (Davies Bouldin and S_dbw). So displaying them on the same radar chart would be quite confusing. Should vertices farther from the centre indicate better performance or worse performance? The answer actually depends on the metric you're examining.

Dataset Dependency: Metrics can vary significantly across different datasets, which makes global normalization strategies problematic. A silhouette score of 0.4 might be excellent for a noisy dataset with many overlapping classes, but it will also be a poor score for a well separated score.

Cross dataset comparison : When comparing metrics for different techniques in the same dataset, we want to normalize in such a fashion that maximises the spread of values across 0 to 1 range, however when comparing across the datasets, we also need the normalization that the metric maintains objective interpretations.

3.8.8.2 Fixed-Threshold Normalization Approach

BarDar uses a fixed-threshold normalization strategy designed to provide consistent, interpretable values across datasets while ensuring all metrics follow a “higher is better” convention.

Silhouette Coefficient: Maps the native $[-1,+1]$ range to $[0,1]$:

$$m_{\text{norm}} = \frac{m+1}{2}$$

Davies-Bouldin Index: Lower values indicate better clustering. Inverted and capped at threshold $c_{DB} = 3.0$:

$$m_{\text{norm}} = 1 - \min\left(\frac{m}{c_{DB}}, 1\right)$$

S_Dbw Index: Lower values indicate better clustering. Inverted and capped at threshold $c_S = 2.0$:

$$m_{\text{norm}} = 1 - \min\left(\frac{m}{c_S}, 1\right)$$

Trustworthiness: Already in $[0,1]$ with higher values indicating better local neighborhood preservation. No transformation needed:

$$m_{\text{norm}} = m$$

Fixed thresholds were chosen over adaptive min-max normalization to:

1. Maintain consistent interpretation across different datasets
2. Avoid misleading comparisons where a “perfect” score of 1.0 might represent mediocre absolute performance
3. Ensure the normalization reflects established ranges from clustering literature

The capping thresholds ($c_{DB} = 3.0$, $c_S = 2.0$) were selected based on typical value ranges observed in real-world datasets, ensuring that most practical values fall within the normalized range while extreme outliers are gracefully bounded.

3.8.8.3 Limitations and tradeoffs of normalization

While the normalization strategy effectively addresses many challenges, it introduces some new tradeoffs that users should be aware of.

Loss of absolute interpretation : Because the normalization is dataset specific, same normalised value (ex. Silhouette score of 0.7) may not indicate the same absolute silhouette score across different datasets, meaning that two radar charts from totally different datasets cannot be compared. Although that is not the purpose of this tool, we mitigate this issue by providing both raw and normalised value using details on demand and tooltips.

Loss of information : Normalization treats all the metrics as equally important, by scaling them to [0,1] range. In reality, some metrics may be more reliable or relevant for certain data characteristics. Additionally they also might encode other information which is lost when we scale the metrics.

3.8.9 User guidance and transparency

To support the correct interpretation despite these normalization complexities, we make use of several guidance features in BarDar chart.

Raw Values access : Tooltips show both normalized (used for visualization) and the raw metric outputs (raw values), enabling users to access absolute scores when needed

Normalization Indicator : A small text label indicates that the values are normalized and data specific.

Metric interpretation guide: Text tips also include simplified guidelines for each metric and their interpretation i.e. higher is better etc.

3.8.10 Relationship to research questions

BarDar directly addresses the research question 2 and 3 (RQ2, RQ3)

RQ3 (Clustering quality visualization) : BarDar provides a comprehensive solution for visualizing multiple clustering quality metrics simultaneously. The radar chart component shows individual metric values and patterns, whereas the bar chart component gives the aggregate assessment. Together they both allow for both detailed and also summary evaluation of the dimensionality reduction techniques effectiveness.

RQ2 (Comparative analysis) : The explicit encoding of aggregate performance through the bar chart enables clear comparative ranking of the dimensionality reduction methods. Unlike traditional approaches where the users must mentally compare irregular radar polygons, BarDar makes the comparison process explicit and quantitative.

Additionally it also supports Industrial use by providing a decision making support tool that helps users and practitioners select the appropriate dimensionality reduction method based on multiple quality criteria, rather than relying on singular metric or subjective visualization assessment.

3.8.11 BarDar chart feedback and iteration

The BarDar chart was designed to provide a high level summary to answer the question, 'which dimensionality reduction method produces the best clustering?'. Many challenges were encountered during the design and development of BarDar

Challenge 1 : Perceptual non-accuracy of radar charts

Initial Problem: The initial idea was to use a single standard radar chart, where each of the vertex represents a different quality metric. The best dimensionality reduction method would theoretically have the largest polygon.

Feedback: The design had two critical, and well documented flaws [52]

1. Area Comparison : Humans are notoriously bad at accurately comparing the areas of irregular polygons.
2. Axis order : The perceived shape and area of the polygon can be dramatically altered simply by reordering the metrics around the circle in a different way, making the visualization arbitrary.

Solution : To solve this issue, we augmented the radar chart, with the "quantitative performance" bar chart at the side of the radar chart. This chart explicitly encodes the aggregate performance of each of the DR method polygons into a simple, trivial, one dimensional bar, which gives the user a more clear and unambiguous summary.

Challenge 2 : Area vs Mean

Problem : Originally, the bar chart was to simply show the actual area of the polygon, based on the logic that if metrics are higher, area will be higher, but it was soon apparent that it will suffer from the same problem of axis order dependency. Same metrics can be placed in different order around the circle to get different areas on the same dataset. The area doesn't really encode any meaningful information.

Solution : Instead of an area based approach for the bar chart, we decided instead to make use of an aggregation score instead, which calculates the normalized and scaled means of all the methods, which is a more closer representation of the dataset/reality than a simple radar area.

Challenge 3 : Nuanced comparison

Problem: The aggregate score in the bar chart is a simple normalized mean of all the metrics for that given dimensionality reduction method. The problem is it does not tell the user why a method is good, since it combines all the metrics into a single bar, possibly losing important discriminatory information.

Feedback : This problem was identified as a limitation during design discussions, which referenced the tool LineUp [69] as an example of a more sophisticated/nuanced comparison method. The professor suggested exploring the use of stacked bar charts instead.

Solution (Stacked Bar Chart): This feedback led to the development of an alternative visualization, i.e. A stacked bar chart instead of a normal bar chart. This view, intended for A/B testing, replaces the simple aggregate bar with a stacked bar chart. Each methods bar is composed of colored segments representing the individual contribution of each of the metric. This allows for a more knowledgeable user to see the precise tradeoffs and understand the composition of the final score, and not just the score itself.

Challenge 4 : User flow

Problem : Original design showed the user the DimCompare view first, then the user went and explored the BarDar view. This selection was initially done arbitrarily, without any regard to the flow of the tool and user experience.

Feedback and Solution (Reorder views) : During the discussions, it was suggested by the research assistant that it made more sense to show the user summary view first, offered primarily by the BarDar chart, and the second view should be the DimCompare view, which offers more explanations and a qualitative analysis. Taking this suggestion, we changed the order of the views. First the user is greeted with a BarDar view which offers them a quick overview and the second view lets them dig deeper.

3.9 Chapter Summary

This chapter presented the BarDar, a novel composite visualisation combining radar charts with bar charts to address the challenge of comparing clustering quality across multiple dimensionality reduction techniques, the key contributions of BarDar include :

- **Explicit aggregate encoding:** Transforms of a difficult perceptual task of comparing polygons into a simple one, i.e. comparing 1 dimensional bar heights.
- **Accessible details :** The radar chart preserves the individual metric values, while the accompanying juxtaposed bar chart provides an aggregated summary.
- **Interaction support :** The users can interact with the bar chart, by dragging and dropping the horizontal bars over one another, which also reflects the changes by reordering the radar chart rendering according to the new order set by the user.
- **Adaptive Normalization :** Dataset specific normalization strategy that maximizes the discriminability while handling metrics with different scales and optimization directions.

The next chapter describes the overall system architecture and implementation details that enable BarDar and Dimcompare within an integrated web based platform/app.

4 System Design and Architecture

In this chapter we discuss the technical implementation and system design of the visual analytics framework, i.e. DatasetWiz. The system is engineered to be a robust and scalable solution for interactive exploration and the comparative analysis of high dimensional data. The design is made to address the research questions and the practical challenges in feature space quality, as identified earlier during the industrial collaboration with Bosch.

The chapter first talks about the systems high level design and architecture patterns. It then provides component by component breakdown of the data pipeline, the analytics engine and the novel visualizations, DimCompare and BarDar.

4.1 Architectural style and Patterns

The system makes use of a client-server architecture, to logically and physically separate the tasks of data processing from the interactive visualization. This architectural pattern is motivated by the severe computational demands of dimensionality reduction for high dimensional datasets. Non linear algorithms like t-SNE also scale poorly with data size increase.Performing all the heavy calculations server-side ensures that the frontend client remains lightweight and responsive, for rendering and interactive operations like zooming, panning and selection.

The system is made up of three primary layers:

1. **Presentation layer (Client) :** Browser based interface that is responsible for all the visualization rendering and user interface/interactions. It operates completely in the browser, and doesn't need any specialized installation.
2. **Application Layer (Server) :** Django (a python server framework library) based web server for managing HTTP requests, API calls, session states and routing. This acts as the central coordinator between the client and the data processing pipeline. Includes endpoints for data upload, analysis request and results delivery.
3. **Data processing Layer (Server) :** Python based analytical engine that handles computationally intensive tasks such as data parsing, normalisation, dimensionality reduction, clustering and metric calculation. It makes use of scientifically established computing libraries to ensure correctness while maintaining good performance for handling industrial scale datasets.

The layered architecture enables the computations to be executed server side, where the performance can be optimized, while the interactive operations are client side so that they remain responsive. The architecture also supports future enhancements, such as distributed or parallelized computing for really big datasets.

4.2 Technology Stack

The implementation of DatasetWiz uses modern web technologies and well known scientific computation libraries both chosen for their performance, maturity and most importantly their open source nature.

4.2.1 Frontend Technologies

- **D3 js v7.8.5 :** The is the foundation for all the visualisations. D3 was chosen over other high level plotting libraries like Plotly and matplotlib, for its granular, low level control of the DOM (Document Object Model) which is essential for creating novel visual encodings such as cluster annotations and the composite BarDar chart.
- **Vanilla JavaScript ES 6+ :** Core application logic, state management and event handling. It allows for minimal dependencies, while maintaining performance. Provides features for asynchronicity and API communication like promises and async/await.
- **Tailwind CSS v3.0 :** CSS framework that is utility first, enabling rapid UI development with consistent styling and responsive design utilities. Used for styling the UI, buttons and other HTML elements.

4.2.2 Backend Technologies

- **Django 5.1.2:** Python web framework providing application structure, URL routing, template rendering and Object Relational Management (ORM) for metadata storage. Django also provides security features like CSRF protection, SQL injection protection etc to ensure safe handling of user data.
- **Numpy 2.0.2 :** Provides helper functions for fundamental array operations, linear algebra and numerical processing. Uses Vectorized operations to significantly accelerate the calculations.
- **Pandas 2.2.3 :** Data wrangling, manipulation CSV parsing and feature statistics
- **Scikit-learn 1.5.2 :** Implementation of PCA, t-SNE, MDS, k-Means clustering and standard metrics (Silhouette, Davies Bouldin, Calinski Harabasz, Trustworthiness). The scikit-learn implementation makes use of the Barnes hut approximation for $\mathcal{O}(n \log n)$ complexity.
- **UMAP-learn 0.5.7 :** Provides UMAP dimensionality reduction technique implementation, chosen for its superior performance on high dimensional data and the ability to preserve both local and global structures,
- **S_dbw 0.4.0 :** Uses the S_Dbw clustering validity metric, developed by [38]. We use a python pip library called s-dbw to calculate this score
- **SQlite :** Lightweight relational database used for storing dataset metadata, analysis configurations and cached results.

Layer	Technology	Role and Primary Function
Frontend	D3.js v7.8.5	Low-level DOM manipulation for custom SVG visualizations (DimCompare, BarDar).
	JavaScript (ES6+)	Application state management, event handling, and asynchronous API communication.
	Tailwind CSS v3.0	Utility-first styling for a responsive and consistent user interface.
Backend	Django 5.1.2	Python web framework for URL routing, session management, and API endpoints.
	SQLite	Metadata storage and caching of dimensionality reduction results.
Analytics	Scikit-learn 1.5.2	Core implementation of PCA, t-SNE, MDS, k-Means, and evaluation metrics.
	UMAP-learn 0.5.7	High-performance implementation of the UMAP algorithm for structure preservation.
	Pandas & NumPy	Data wrangling, CSV parsing, and vectorized numerical array operations.
	S_dbw 0.4.0	Specialized library for calculating density-based clustering validity.

Table 4: Technical Implementation Stack of DatasetWiz

4.3 Data Processing Pipeline

The data processing pipeline transforms the uploaded datasets through a series of modular processing and pre-processing steps.

4.3.1 Data upload and validation

When the user uploads a CSV file of a high dimensional dataset, Djangos file handling manages the upload. The system also provides immediate feedback by performing validation.

```
# Actual implementation from views.py
dataset = Dataset(
  name=request.POST[’name’],
  description=request.POST.get(’description’, ’’),
  file=request.FILES[’file’]
)
dataset.save() # Triggers custom save method to:
# 1. Save file, 2. Identify numeric/categorical columns, 3. Save
  metadata

# Read and process numerical data
df = pd.read_csv(dataset.file.path)
numeric_df = df[dataset.columns[’numeric’]]
```

Listing 4.1: Backend implementation of dataset model and initial processing.

The system validated that sufficient data exists (it has minimum samples and features) and also identifies the numeric columns vs categorical columns.

[ADD PICTRE of main UI on frontend landing page/dataset uplaod]

4.3.1.1 Preprocessing and Normalization

Numerical features go through standardization to ensure that the dimensionality reduction algorithms treat all the features equally, regardless of their original scales. Since distance based algorithms (PCA, MDS, Kmeans) are highly sensitive to differences of scale, Features with larger numerical values would make everything disproportionate and dominate the analysis

The system applies Z-score Normalization (StandardScaler):

The system applies Z-score normalization (StandardScaler) to handle missing values and scale features

```python
# Handle missing values - fill with column mean
for column in numeric_df.columns:
    column_mean = numeric_df[column].mean()
    numeric_df[column] = numeric_df[column].fillna(column_mean)

# Standardize using Z-score normalization
scaler = StandardScaler()
scaled_data = scaler.fit_transform(numeric_df)
```

Listing 4.2: Data cleaning and Z-score standardization.

This implements the transformation:

$$x'_j = \frac{x_j - \mu_j}{\sigma_j} \tag{1}$$

where μ_j and σ_j are the mean and standard deviation of feature j, respectively.

Wherea where uj and o–j are the mean and standard deviation of feature j. This results in zero mean and unit variance, making features directly comparable.

4.3.1.2 Dimensionality reduction module

They system includes four key dimensionality reduction algorithms, each chosen to provide methodologically diverse projections

Principal Component Analysis (PCA) : Linear technique preserving global variance, it is computationally efficient and also has a deterministic baseline

t-SNE : A non-linear technique revealing local neighborhood structures and clusters. Scikit-learn's implementation makes use of the Barnes-Hut approximation, resulting in a $\mathcal{O}(n \log n)$ performance.

UMAP : Modern and non-linear alternative which is based on the manifold theory. It balances preservation of both local cluster structure and global data topology with excellent speed.

MDS : Classic non-linear technique which preserves pairwise distances from high dimensional space in low-dimensional projection. It provides methodology/technique completeness but has $\mathcal{O}(n^3)$ complexity, which limits the scalability.

4.3.1.3 Implementation

```
# Actual DR implementation from views.py
if method == 'pca':
    pca = PCA(n_components=2)
    reduced_data = pca.fit_transform(scaled_data)
  elif method == 'tsne':
    tsne = TSNE(n_components=2, random_state=42)
    reduced_data = tsne.fit_transform(scaled_data)
  elif method == 'umap':
    umap_reducer = UMAP(n_components=2, random_state=42)
    reduced_data = umap_reducer.fit_transform(scaled_data)
  elif method == 'mds':
    mds = MDS(n_components=2, random_state=42, n_init=4,
          dissimilarity='euclidean')
    reduced_data = mds.fit_transform(scaled_data)
```

Listing 4.3: Implementation of dimensionality reduction algorithms.

For the parameters, we use recommendations from the literature. PCA and MDS are deterministic; t-SNE uses the default *perplexity*=30. For UMAP, we use the default *n_neighbors* = 15 and set *min_dist* = 0.1.

4.4 Clustering and metrics

4.4.1 High-Dimensional Clustering Strategy

To ensure that the comparison of two DR techniques is fair, the clustering is performed once in the original high dimensional space, not in the 2D projections. Doing this provides a consistent set of cluster labels used to color both the projections, enabling a fair comparison. If the clustering is done in the low-dimensional space, the comparison cannot be accurate since the clustering algorithm results might change due to the presence of dimensionality reduction artifacts, as each of the DR methods produces a distinct low dimensional 2D projection. Applying the clustering algorithm to the high dimensional data allows for a true comparison.

```
# Clustering performed on HIGH-DIMENSIONAL scaled_data
if auto_detect:
    n_clusters = find_optimal_clusters(scaled_data)
else:
    n_clusters = int(data.get('n_clusters', 5))

# Use KMeans++ initialization for better convergence
kmeans = KMeans(n_clusters=n_clusters, random_state=42, init='k-
    means++')
clusters = kmeans.fit_predict(scaled_data)
```

Listing 4.4: K-means clustering performed on high-dimensional data.

If the user selects automatic detection, the system iterates through potential values of *k* to identify the optimal number of clusters based on the highest mean Silhouette Coefficient:

```
def find_optimal_clusters(data, max_clusters=10, min_clusters=2):
    best_score = -1
    best_k = min_clusters

    for k in range(min_clusters, max_clusters + 1):
        kmeans = KMeans(n_clusters=k, random_state=42)
        labels = kmeans.fit_predict(data)
        score = silhouette_score(data, labels)
        if score > best_score:
            best_score = score
            best_k = k
    return best_k
```

Listing 4.5: Automatic cluster number detection logic.

The k-means implementation, used as the clustering algorithm, uses the scikit-learns optimised version with k-means++ initialization, which differs from standard k-means in the sense that the initial points are chosen based on the furthest distance rather than choosing the initial points randomly, which can cause two initial starting points to be next to each other. Users can either specify the value of *k* or use the automatic detection which in turn is based on silhouette analysis to find the optimal number of clusters.

4.4.2 Metric calculation and normalization

While clustering is performed in the high dimensional space, the metrics, however, are calculated on the 2D reduced data to assess how well each of the projections represents the underlying cluster structure.

```
# Metrics calculated on 2D REDUCED data to assess projection
  quality
silhouette = silhouette_score(reduced_data, clusters)
davies = davies_bouldin_score(reduced_data, clusters)
sdbw = calculate_sdbw(reduced_data, clusters)

# Trustworthiness requires both high-D and low-D data
```

```
  trust = trustworthiness(scaled_data, reduced_data,
              n_neighbors=min(15, len(reduced_data) - 1))
```

Listing 4.6: Calculating clustering and projection metrics on 2D data.

This helps to evaluate how well the 2D projection separates the high-dimensional clusters.

Silhouette coefficient : Measures cluster cohesion versus separation, $s(i) = \frac{b(i)-a(i)}{\max(a(i),b(i))}$ where $a(i)$ is mean intra-cluster distance and $b(i)$ is mean nearest-cluster distance. Range: [-1, 1], where higher value is better.

Trustworthiness: Cluster reliability or distances in high dim vs low dim. Higher values indicate better reliability.

Davies Bouldin index : Average similarity between clusters. Lower values are better.

S_Dbw : Combines the scatter and density. Lower values are better.

To enable meaningful visual comparison within the BarDar component, all metrics are normalized to a consistent $[0,1]$ scale where higher values always indicate superior performance:

```
metrics = {
    'silhouette': float((silhouette + 1) / 2.0), # Map [-1,1] to
   [0,1]
    'trustworthiness': float(trust), # Already in [0,1]
    'davies_normalized': float(1 - min(davies / 3.0, 1.0)),#
   Inverted
    'sdbw_normalized': float(1 - min(sdbw / 2.0, 1.0)),# Inverted
  }
```

Listing 4.7: Normalization of metrics for visual consistency in BarDar.

4.5 Frontend visualization implementation

The frontend visualization renders the analytical results as interactive graphics using the d3.js library, while managing the user interactions and also maintaining the consistency across the views.

4.5.0.1 DimCompare Scatterplot rendering

```
  class DimCompare {
    constructor(container, controlsPrefix) {
      this.container = d3.select(container);
      const bbox = this.container.node().getBoundingClientRect();
      this.width = Math.max(bbox.width, 800);
      this.height = Math.max(bbox.height, 600);
      // Initialize linear scales for 2D projection
      this.xScale = d3.scaleLinear()
        .range([this.margin.left, this.width - this.margin.right]);
      this.yScale = d3.scaleLinear()
        .range([this.height - this.margin.bottom, this.margin.top]);

      // Configure zoom behavior (0.5x to 20x)
      this.zoom = d3.zoom()
        .scaleExtent([0.5, 20])
        .on('zoom', (event) => {
          this.g.attr('transform', event.transform);
          this.updateScaledElements(event.transform);
        });
    }
  }
```

Listing 4.8: Responsive initialization and zoom behavior in DimCompare.

Synchronization between the scatterplots ensures coordinated exploration, through the use of linked hover events.

4.5.0.2 Cluster glyph rendering and placement

Cluster annotations are rendered as floating SVG groups positioned at the centroid of the cluster:

$$\text{centroid} = \left(\frac{1}{n} \sum_{i=1}^{n} x_i, \frac{1}{n} \sum_{i=1}^{n} y_i \right)$$

Each of the glyph displays feature deviations from the global mean. The server calculates these deviations on the original high dimensional data.

```
# Feature deviation calculation from views.py
for cluster in range(n_clusters):
  cluster_mask = clusters == cluster
  cluster_data = numeric_df[cluster_mask]
  other_data = numeric_df[~cluster_mask]

  differences = {}
  for feature in selected_features:
    cluster_mean = float(cluster_data[feature].mean())
    other_mean = float(other_data[feature].mean())

    if other_mean != 0:
      diff_percent = ((cluster_mean - other_mean) / abs(other_mean
 )) * 100
    else:
      diff_percent = 0 if cluster_mean == 0 else 100
    differences[feature] = float(diff_percent)
```

Listing 4.9: Backend calculation of cluster-specific feature deviations.

The frontend renders bar charts in the cluster annotations with logarithmic scaling, to handle percentage differences that may span across orders of magnitude (1% to 1000%) :

```
// Logarithmic bar scaling to handle wide dynamic ranges
const logValue = Math.log10(Math.abs(value) + 1);
const maxLogValue = Math.log10(2001); // Cap visualization at
 2000%
const barLength = Math.min((logValue / maxLogValue) * maxBarLength
 , maxBarLength);
```

Listing 4.10: Logarithmic scaling for cluster glyph bar lengths.

4.5.1 BarDar Chart Implementation

The BarDar visualization utilizes D3's radial line generators to construct the performance profiles of each dimensionality reduction technique.

```
drawRadarChart() {
  const angleSlice = (Math.PI * 2) / metricNames.length;
  const rScale = d3.scaleLinear().domain([0, 1]).range([0,
 maxRadius]);

  ['pca', 'tsne', 'umap', 'mds'].forEach(method => {
    const points = metricNames.map((metric, i) => {
      const angle = angleSlice * i - Math.PI / 2;
      return {
        x: Math.cos(angle) * rScale(normalizedValue),
        y: Math.sin(angle) * rScale(normalizedValue)
      };
    });

    const lineGenerator = d3.line()
      .x(d => d.x)
      .y(d => d.y)
      .curve(d3.curveLinearClosed);

    this.radarSvg.append('path')
      .datum(points)
      .attr('d', lineGenerator)
      .attr('fill', this.colors[method])
      .attr('fill-opacity', 0.2);
  });
}
```

Listing 4.11: Radar chart polygon rendering logic.

The bar chart encodes the aggregate score (mean of normalized metrics), providing a simple 1D comparison with a drag-and drop based reordering mechanism.

4.6 Data Flow

The end to end data flow of the tool looks like:

1. **Upload :** User uploads the CSV file via browser to Django backend
2. Preprocessing : Data processing layer parses the CSV , handles missing values if present, and applies the z-score normalization.
3. **Clustering :** Normalized high dimensional data is then clustered using the k-Means, generating the consistent cluster labels.
4. **Dimensionality Reduction :** Normalized data is passed to selected DR algorithms, generating a different 2D projection/coordinates.
5. **Metric calculation :** 2D reduced data and cluster labels are passed to the metrics calculation module
6. **Aggregation :** The backend bundles the results into a JSON response
7. **Rendering :** D3.js creates scatterplots using different coordinates but using the same labels in each of the projection for coloring.

This flow ensures users can visually compare how different DR techniques represent the exact same high dimensional cluster structure.

4.6.1 Technical Specifications (table of libraryversions)

Table 5.1: Technical Specifications

Component	**Technology**	**Version**	**Purpose**
Backend Framework	Django	5.1.2	Application Layer, API routing, security
Data Processing	NumPy, Pandas	2.0.2, 2.2.3	Numerical computing, CSV parsing
DR & Clustering	Scikit-learn	1.5.2	PCA, t-SNE, MDS, k-Means, metrics
Optimized DR	UMAP-learn	0.5.7	High-performance UMAP implementation
Custom Metrics	S_Dbw	Custom	Cluster validity assessment
Frontend Rendering	D3.js	v7.8.5	Custom visualizations (DimCompare, BarDar)
Frontend Logic	JavaScript	ES6+	UI interactivity, state management
Frontend Styling	Tailwind CSS	v3.x	Utility-first, responsive UI design
Database	SQLite	-	Metadata storage

Table 5: System Implementation Stack

4.7 Design Trade-offs

The final system architecture shows the intentional design tradeoffs made:

- **Processing (Server vs Network) :** Server side computation handles the large datasets efficiently but introduces some network latency. The benefit of robust heavy computation outweighs the network latency costs for typical high dimensional dataset sizes.
- **Code (Vanilla JS vs frameworks) :** Using vanilla JavaScript instead of frameworks like React, nextjs or Vue.js minimizes the dependencies and gives the DOM control directly to D3.js. The tradeoff here is that we have to perform state management manually.
- **Usability vs Original representation (BarDar metric inversions) :** Metrics like Davies Bouldin (where lower is better) are inverted so that "taller is always better”. This sacrifices the academic representation for user friendly interpretation.
- **Accuracy (True data vs pretty pictures/aesthetics):** High-dimensional clustering may produce less “clean” 2D views, but this choice was deliberately made to favor rigorous evaluation over visually pleasing, but potentially misleading visualizations.

4.8 Chapter Summary

This chapter details the complete system architecture and the technical implemetnaion of DatasetWiz. The key design decisions include:

- **Client server architecture**: Pytho/Django backend for computation, Javascript/D3.js frontend for responsive viualization.
- **High dimensional data pipeline**: Clustering in te original space ensures that the projection comparison is unbiased.
- **Technology decisions**: D3.js for low level control, scikit-learn and UMAP-learn for reliable DR implementations.
- **Interactive Data Flow**: Asynchronous, API-driven communication enables features like the glyphs on demand etc.
- **Design Trade-Offs**: Balancing the performance, usability, user intuitiveness

The implementation addresses the research questions through intentional and purpose built visualization components. DimCompare for RQ1 (feature interpretation), and RQ2 (side by side comparison), BarDar for RQ3 (multi-metric visualisation) and interactive features for RQ4 (pattern identification)

The next chapter presents the usage scenarios and the evaluation, demonstrating the systems effectiveness in real world scenarios using real world datasets.

5 Discussion and Evaluation

6 Conclusion

6.1 Summary

6.2 Future Work

6.3 Limitations

6.4 Conclusion

Bibliography

[1] Rizgar R. Zebari, Adnan M. Abdulazeez, Diyar Q. Zeebaree, Dilovan A. Zebari, and Jwan N. Saeed. A comprehensive review of dimensionality reduction techniques for feature selection and feature extraction. *Journal of Applied Science and Technology Trends*, 1(1):56–70, 2020.

[2] Weikuan Jia, Meili Sun, Jian Lian, and Sujuan Hou. Feature dimensionality reduction: A review. *Complex & Intelligent Systems*, 8:2663–2693, 2022.

[3] Richard Ernest Bellman. *Dynamic Programming*. Princeton University Press, Princeton, NJ, 1957.

[4] Dehua Peng, Zhipeng Gui, and Huayi Wu. Interpreting the curse of dimensionality from distance concentration and manifold effect. *arXiv preprint arXiv:2401.00422*, 2023.

[5] Farzana Anowar, Samira Sadaoui, and Bassant Selim. Conceptual and empirical comparison of dimensionality reduction algorithms (pca, kpca, lda, mds, svd, lle, isomap, le, ica, t-sne). *Computer Science Review*, 40:100378, 2021.

[6] Carlos Oscar S. Sorzano, Juan Vargas, and Alberto P. Montano. A survey of dimensionality reduction techniques. *arXiv preprint arXiv:1403.2877*, 2014.

[7] Basna Mohammed Salih Hasan and Adnan Mohsin Abdulazeez. A review of principal component analysis algorithm for dimensionality reduction. *Journal of Soft Computing and Data Mining*, 2(1):20–30, Apr. 2021.

[8] I. T. Jolliffe. Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A*, 374(2065), 2016.

[9] Philippe Boileau, Nima S Hejazi, and Sandrine Dudoit. Exploring high-dimensional biological data with sparse contrastive principal component analysis. *Bioinformatics*, 36(11):3422–3430, 2020.

[10] Bernhard Scholkopf, Alexander J. Smola, and Klaus-Robert Muller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.

[11] Laurens J.P. van der Maaten and Geoffrey E. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[12] Dmitry Kobak and Philipp Berens. The art of using t-sne for single-cell transcriptomics. *Nature Communications*, 10(1):5416, 2019.

[13] Martin Wattenberg, Fernanda Viégas, and Ian Johnson. How to use t-sne effectively. Distill, 2016. Online article.

[14] Laurens van der Maaten. Barnes-hut t-sne. *arXiv preprint*, 2013.

[15] Ingwer Borg and Patrick J. F. Groenen. *Modern Multidimensional Scaling: Theory and Applications*. Springer, New York, NY, 2nd edition, 2005.

[16] Joseph B. Kruskal and Myron Wish. *Multidimensional Scaling*. Quantitative Applications in the Social Sciences. SAGE Publications, 1978.

[17] Jarkko Venna, Jaakko Peltonen, Kristian Nybo, Helena Aidos, and Samuel Kaski. Information retrieval perspective to nonlinear dimensionality reduction for data visualization. *Journal of Machine Learning Research*, 11(3):451–490, 2010.

[18] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

[19] Benyamin Ghojogh, Ali Ghodsi, Fakhri Karray, and Mark Crowley. Uniform manifold approximation and projection (umap) and its variants: Tutorial and survey. *arXiv preprint arXiv:2109.02508*, 2021.

[20] Joshua B. Tenenbaum, Vin de Silva, and John C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[21] Geoffrey E. Hinton and Ruslan R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[22] Mateus Espadoto, Rafael M. Martins, Andreas Kerren, Nina S. T. Hirata, and Alexandru C. Telea. Towards a quantitative survey of dimension reduction techniques. *IEEE Transactions on Visualization and Computer Graphics*, 27(3):2153–2173, 2019.

[23] Stuart P. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.

[24] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 226–231, 1996.

[25] Erich Schubert, Jörg Sander, Martin Ester, Hans-Peter Kriegel, and Xiaowei Xu. Dbscan revisited, revisited: Why and how you should (still) use dbscan. *ACM Transactions on Database Systems (TODS)*, 42(19), 2017.

[26] Ricardo J. G. B. Campello, Davoud Moulavi, and Jörg Sander. Density-based clustering based on hierarchical density estimates. In *Proceedings of the 17th Pacific–Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*, pages 160–172, 2013.

[27] Ulrike von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.

[28] Geoffrey J. McLachlan and David Peel. *Finite Mixture Models*. Wiley Series in Probability and Statistics, 2000.

[29] Karin Kailing, Hans-Peter Kriegel, and Peer Kröger. Density-connected subspace clustering for high-dimensional data. In *Proceedings of the 4th SIAM International Conference on Data Mining (SDM)*, pages 246–257, 2004.

[30] Zahid Ansari, M. F. Azeem, Waseem Ahmed, and A. Vinaya Babu. Quantitative evaluation of performance and validity indices for clustering the web navigational sessions. arXiv preprint arXiv:1507.03340, 2015.

[31] M. Gösgens, A. Tikhonov, and L. Prokhorenkova. Systematic analysis of cluster similarity indices. arXiv preprint arXiv:190?, 2019.

[32] Peter J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987.

[33] David L. Davies and Donald W. Bouldin. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(2):224–227, 1979.

[34] N. R. Pal and J. Biswas. Cluster validation using graph theoretic concepts. *Pattern Recognition*, 30(6):847–857, 1997.

[35] Tadeusz Caliński and Jerzy Harabasz. A dendrite method for cluster analysis. *Communications in Statistics*, 3(1):1–27, 1974.

[36] Joseph C. Dunn. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, 3(3):32–57, 1973.

[37] O. Arbelaitz, I. Gurrutxaga, J. Muguerza, J. M. Pérez, and I. Perona. An extensive comparative study of cluster validity indices. *Pattern Recognition*, 46(1):243–256, 2013.

[38] Maria Halkidi and Michalis Vazirgiannis. Clustering validity assessment: Finding the optimal partitioning of a data set. In *Proceedings of the 2001 IEEE International Conference on Data Mining (ICDM'01)*, pages 187–194, San Jose, CA, USA, November 2001. IEEE.

[39] J. P. Morais, M. Adhikari, T. Taha, I. Gemp, and W. G. Macready. Ferm: A feature-space evaluation and representation measure for classification tasks. arXiv preprint arXiv:1909.02699, 2019.

[40] Laurence Hubert and Phipps Arabie. Comparing partitions. *Journal of Classification*, 2(1):193–218, 1985.

[41] Nguyen Xuan Vinh, Julien Epps, and James Bailey. Information theoretic measures for clusterings comparison: Variants, properties, limits and extensions. *Journal of Machine Learning Research*, 11:2837–2854, 2010.

[42] Andrew Rosenberg and Julia Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 410–420, 2007.

[43] E. Amigó, J. Gonzalo, J. Artiles, and F. Verdejo. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information Retrieval*, 12(4):461–486, 2009.

[44] P. C. Mahalanobis. On the generalised distance in statistics. Proceedings of the National Institute of Sciences of India, 1936.

[45] A. Bhattacharya. On a measure of divergence between two statistical populations defined by their probability distributions. *Bulletin of the Calcutta Mathematical Society*, 35:99–109, 1943.

[46] L. Bruzzone and S. B. Serpico. A neural-network approach to the supervised classification of very high spatial resolution remote sensing imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 36(2):558–571, 1998.

[47] Xianmei Zhang, Xiaofeng Lin, Dongjie Fu, Yang Wang, Shaobo Sun, Fei Wang, Cuiping Wang, Zhongyong Xiao, and Yiqiang Shi. Comparison of the applicability of j-m distance feature selection methods for coastal wetland classification. *Remote Sensing*, 2023.

[48] Michael Behrisch, Bruno Bach, Niklas W. Riche, Tobias Schreck, and Jean-Daniel Fekete. Quality metrics for information visualization. *Computer Graphics Forum*, 37(3):625–662, 2018.

[49] Tamara Munzner. A nested model for visualization design and validation. In *IEEE Transactions on Visualization and Computer Graphics*, volume 15, pages 921–928, 2009.

[50] Michael Gleicher, Daniele Albers, Robert Walker, Iftikhar Jusufi, Chris D. Hansen, and James C. Roberts. Visual comparison for information visualization. *Information Visualization*, 10(4):289–309, 2011.

[51] Sylvain Lespinats and Michaël Aupetit. Checkviz: Sanity check and topological clues for linear and non–linear mappings. *Computer Graphics Forum*, 30(1):113–125, 2011.

[52] William S. Cleveland and Robert McGill. Graphical perception: Theory, experimentation, and application to the development of graphical methods. *Journal of the American Statistical Association*, 79(387):531–554, 1984.

[53] Daniel Smilkov, Nikhil Thorat, Charles Nicholson, Emily Reif, Fernanda B. Viégas, and Martin Wattenberg. Embedding projector: Interactive visualization and interpretation of embeddings. arXiv preprint arXiv:1611.05469, 2016.

[54] Marco Cavallo and Çağatay Demiralp. Clustrophile 2: Guided visual clustering analysis. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):267–276, 2019.

[55] Tamara Munzner. *Visualization Analysis and Design.* A K Peters Visualization Series. Taylor & Francis Ltd., Boca Raton, Florida, December 2014.

[56] Sehi L'Yi, Jaemin Jo, and Jinwook Seo. Comparative layouts revisited: design space, guidelines, and future directions. arXiv preprint arXiv:2009.00192, 2020.

[57] Ben Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Proceedings 1996 IEEE Symposium on Visual Languages*, pages 336–343, 1996.

[58] Arjun Srinivasan, Steven M. Drucker, Alex Endert, and John Stasko. Augmenting visualizations with interactive data facts to facilitate interpretation and communication. In *Proceedings of the 2018 IEEE Conference on Visual Analytics Science and Technology (VAST)*, 2018.

[59] Dominik Sacha, Leishi Zhang, Michael Sedlmair, John A. Lee, Jaakko Peltonen, Daniel Weiskopf, Stephen C. North, and Daniel A. Keim. Visual interaction with dimensionality reduction: A structured literature analysis. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):241–250, 2017.

[60] Marco Cavallo and Çağatay Demiralp. Dimreader: Model-agnostic steerable dimensionality reduction for interactive data analysis. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, 2018.

[61] Paulo A. Pagliosa, Fernando V. Paulovich, Rosane Minghim, Haim Levkowitz, and Luis G. Nonato. Projection inspector: Assessment and synthesis of multidimensional projections. *Neurocomputing*, 150:599–610, 2015.

[62] Nicolas Heulot, Michaël Aupetit, and Jean-Daniel Fekete. Proxilens: Interactive exploration of high-dimensional data using projections. In *1st International Workshop on Visual Analytics Using Multidimensional Projections (VAMP) at EuroVis*, pages 11–15, 2013.

[63] Luis G. Nonato and Michaël Aupetit. Multidimensional projection for visual analytics: Linking techniques with distortions, tasks, and layout enrichment. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):2650–2673, 2019.

[64] Hyeon Jeon, Hyunwook Lee, Yun-Hsin Kuo, Taehyun Yang, Daniel Archambault, Sungahn Ko, Takanori Fujiwara, Kwan-Liu Ma, and Jinwook Seo. Unveiling high-dimensional backstage: A survey for reliable visual analytics with dimensionality reduction. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '25)*, 2025.

[65] A. A. Wani. Comprehensive review of dimensionality reduction algorithms. *Unspecified Journal*, 2025. Summarizes that PCA fails to unfold nonlinear manifolds, UMAP preserves both local and global, and t-SNE emphasizes local structure.

[66] Roger Feldman. Filled radar charts should not be used to compare social indicators. *Social Indicators Research*, 111(3):709–712, 2013.

[67] Alexander Klippel, Frank Hardisty, and Chris Weaver. The shape of the star: How visual metaphors support the interpretation of multivariate data. *Information Visualization*, 8(4):227–244, 2009.

[68] Matthew O Ward, Georges Grinstein, and Daniel Keim. *Interactive Data Visualization: Foundations, Techniques, and Applications*. CRC Press, 2nd edition, 2015.

[69] Samuel Gratzl, Alexander Lex, Nils Gehlenborg, Hanspeter Pfister, and Marc Streit. Lineup: Visual analysis of multi-attribute rankings. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2277–2286, 2013.

Appendix

A First Appendix

. . .

Eidesstattliche Erklärung

Ich versichere hiermit wahrheitsgemäß, die Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt, die wörtlich oder inhaltlich übernommenen Stellen als solche kenntlich gemacht und die Satzung der Universität Passau zur Sicherung guter wissenschaftlicher Praxis in der jeweils gültigen Fassung beachtet zu haben. Die Arbeit ist weder von mir noch von einer anderen Person an der Universität Passau oder an einer anderen Hochschule zur Erlangung eines akademischen Grades bereits eingereicht worden.

Passau, den DD.MM.20YY ____________________

Aditya Handrale

Ich versichere hiermit wahrheitsgemäß, dass

- ❑ die Arbeit ohne Zuhilfenahme von ChatGPT oder anderen generativen KI-Werkzeugen erstellt wurde, oder
- ❑ ich in der nachfolgenden Tabelle vollständig dokumentiert habe, wie solche Systeme bei der Entwicklung der Arbeit verwendet wurden.

Passau, den DD.MM.20YY ____________________

Aditya Handrale

Generative KI-Werkzeuge, die in der Arbeit verwendet wurden.

Kapitel	KI-Tool	Version	Prompt	Erklärung/Kommentar
1.2	ChatGPT	3.5	Schreibe einen Absatz über den Digital Markets Act.	Der generierte Output wurde in folgender Weise angepasst ...
2.3	ChatGPT	4.0	...	...
3.1	ChatGPT	3.5	...	...