Visual Analytics for Multivariate Time-Series Data Using Interactive Dimensionality Reduction Methods

Mizuki Emmei* Kobe University
Naoki Okami † Kobe University
Takanori Fujiwara‡ Linköping University
Naohisa Sakamoto§ Kobe University
Jorji Nonaka¶ RIKEN R-CCS

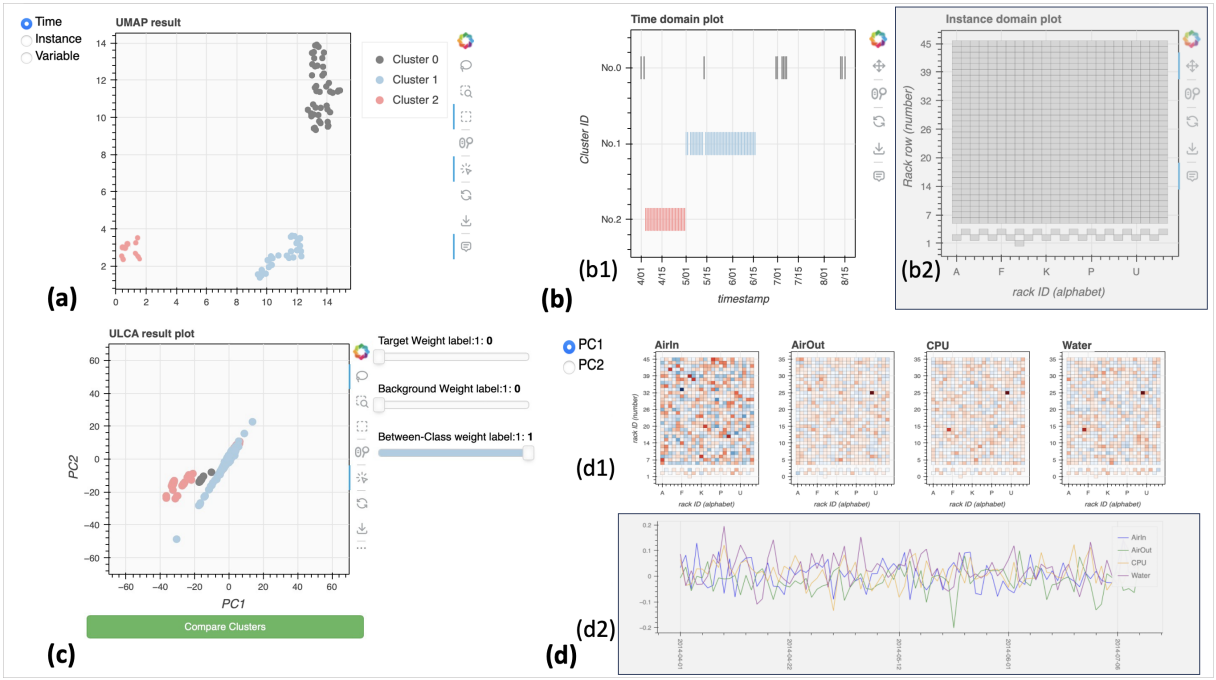

Figure 1: A prototype system for our visual analytics method using interactive dimensionality reduction methods to analyze multivariate time-series data. The system consists of four parts: (a) the UMAP result plot depicting latent patterns extracted from data; (b) the domain plots of (b1) time and (b2) instance domains, which helps understand clusters identified interactively (note: b2 is grayed out as it is not used when the time axis is selected); (c) the ULCA result plot supporting comparative analysis of the clusters; and (d) the contribution plots visualizing information necessary to understand the comparative analysis result.

Abstract

One advancing machine-learning-based analysis approach for multivariate time-series data is representing data as a third-order tensor and then applying dimensionality reduction (DR) methods. In this work, we introduce a visual analytics method that employs multiple interactive DR methods to support both extraction and interpretation of latent patterns of multivariate time-series data. Our method first allows analysts to select an analysis focus from three axes: instance, variable, and time axes. Then, the method applies a multi-step DR method to produce a 2D scatterplot that depicts latent patterns of the selected axis's elements (e.g., time points). Afterward, the analysts interactively investigate data groups that appeared in the plot with a DR method designed for comparative analysis. The method can be further applied iteratively to perform more precise and detailed analyses. We implement a prototype system and demonstrate the effectiveness of our method by analyzing supercomputer log data.

Keywords: Visualization, tensor data, dimensionality reduction, tensor decomposition, interpretation, comparative analysis

1 Introduction

Multivariate time-series data can model various social and scientific phenomena involving temporal changes. With the advancement of sensing, logging, and storage technologies, multivariate time-series data is being collected from a vast amount of sources with high temporal granularity, as seen in the fields of biomedicine [25], meteorology [12], manufacturing [26], and high-performance computing [19, 22, 23]. While such complex multivariate time-series data can be useful to uncover important patterns that lie in targeting phenomena, its complexity at the same time makes performing analyses and gaining insights challenging.

To address this challenge, tensor decomposition [14] and dimensionality reduction (DR) methods [2] have been developed. These machine learning methods represent multivariate time-series data as a third-order tensor with axes of (1) instance, (2) variable, and (3) time and aim to extract a small number of latent features that characterize the data. Since understanding of patterns seen in latent

*e-mail: 248x014x@stu.kobe-u.ac.jp
†e-mail: 237x012x@stu.kobe-u.ac.jp
‡e-mail: takanori.fujiwara@liu.se
§e-mail: naohisa.sakamoto@people.kobe-u.ac.jp
¶e-mail: jorji@riken.jp

features is further needed to gain analytical insights [4], researchers have investigated the conjunction use of interpretable tensor decomposition or DR methods and interactive visualizations [7, 10, 16].

In this work, we contribute to the human-in-the-loop machine learning approach using visual analytics and interpretable DR methods to analyze multivariate time-series data. Our new method enhances an approach used in the MulTiDR framework [10], where three DR methods are used for different purposes: data compression, latent pattern extraction, and latent pattern analysis. Our main enhancement is in the latent pattern analysis. MulTiDR utilizes contrastive learning-based DR method [1] (specifically, ccPCA [9]) to assist analysis of latent patterns. However, this method is to investigate the patterns only from *one limiting* aspect. We instead modify and employ a flexible comparative analysis method [11] to examine the patterns from *multiple* aspects, such as factors that make temporal patterns similar or dissimilar.

Moreover, we improve an analysis workflow suggested in the MulTiDR framework to mitigate analytical drawbacks in the use of DR methods. Applying DR to multivariate time-series data usually causes severe data compression since DR needs to compress many dimensions derived from two tensor axes (e.g., 10 variables and 100 time points produce 1000 dimensions). Our workflow includes a step for an interactive selection of a subpart of the tensor of interest based on initial findings gained from exploratory analysis. Based on this selection, our workflow applies the DR methods again to the selected subtensor to refine the results while reducing the amount of data compression.

Our method is designed for analysts working with high-dimensional time-series data, who have a basic understanding of dimensionality reduction and data visualization. To demonstrate the effectiveness of our method, we analyze supercomputer operational logs collected from the K computer [18] with an operational staff who meets the above requirements as an intended user.

Contributions. The primary contributions of this paper are:

- Improvement of the latent pattern analysis in the MulTiDR framework by integrating an interactive comparative analysis method [11];
- Iterative analysis workflow to reduce the undesired influence from data compression; and
- An analysis of a supercomputer operational log dataset conducted in collaboration with a domain expert.

2 Related Work

We discuss a closely related approach to our work: feature extraction and analysis using (1) tensor decomposition and (2) DR methods. For the broader discussion of visualization of third-order tensors, refer to Bach et al.'s survey [3].

2.1 Feature Extraction Using Tensor Decomposition

Analogous to matrix decomposition, tensor decomposition [14] transforms a tensor into a combination of simpler tensors and matrices. Although tensor decomposition summarizes the original tensor, analyzing the decomposed result itself is a non-trivial task. To address this analysis challenge, visual analytics frameworks have been introduced to support the understanding of tensor decomposition results. These frameworks usually focus on an analysis of spatio-temporal data (i.e., represented as a third-order tensor) and enable exploratory data analysis by visualizing pre- and post-decomposition tensors and matrices. For instance, TPFlow [16] utilizes tensor decomposition to optimally slice a third-order tensor into homogeneous partitions and extract meaningful patterns along instance, variable, and time axes. TPFlow then depicts information on the partitions with visualizations suitable for each axis (e.g., line charts for the time axis). Similarly, Voila [5] uses tensor decomposition to detect anomalies from large-scale spatio-temporal data and visualizes the differences between factor matrices produced by the decomposition to highlight abnormal temporal changes. The existing visual analytics frameworks focus on the extraction of important features or elements from third-order tensors [10]. In our work, we want to not only extract such features and elements but also uncover latent patterns among elements (e.g., groups of similar time points) by using DR methods.

2.2 Multivariate Time-Series Data Analysis with Dimensionality Reduction

DR methods have been utilized to visually analyze multivariate time-series data [7, 10, 13, 20, 24]. DR methods are often applied to a matrix of instances and variables at each time point and then the produced set of DR results is visualized with animation or juxtaposition to show temporal changes. Such DR application examples include Temporal MDS [13], Dynamic t-SNE [20], and Landmark Dynamic t-SNE [24]. However, relying on animation or juxtaposition makes it difficult to find patterns such as outliers or similar time points.

To visualize similarities of all elements of a user-selected axis (i.e., instances, variables, or time points) in one 2D scatterplot, Fujiwara et al. [10] introduced MulTiDR, a visual analysis framework using a two-stage DR process. In MulTiDR, the first stage compresses a third-order tensor into a matrix by reducing the dimensionality of one axis into one with a DR method designed for data compression (e.g., principal component analysis or PCA). From this matrix, the second stage produces a 2D projection. This approach supports the extraction and interpretation of latent patterns of multivariate time-series data. However, this approach has a potential drawback: excessive DR in the first stage may lead to the loss of important information. To address this issue, Fujita et al. [7] designed a multi-step DR method. Their method allows analysts to select elements of interest (e.g., time points) at an intermediate stage of DR processes to reduce the size of an analyzing tensor. Our work introduces a new method by adapting and enhancing MulTiDR and the multi-step DR method to support more flexible and precise analysis of DR results.

3 Method

We design an interactive DR method to extract and review latent patterns of a third-order tensor representing multivariate time-series data. To design an analysis workflow of our method, we utilize the MulTiDR framework [10] while increasing analysis flexibility of each step in the MulTiDR framework. With this enhanced flexibility, our method can more actively involve analysts' knowledge or interest when generating DR results. This involvement of analysts allows a more thorough understanding of extracted latent patterns as well as mitigates the influence of excessive data compression by DR.

Fig. 2 provides an overview of our method. From a third-order tensor, our method first extracts and visualizes latent patterns by utilizing a multi-step DR method [7] (Fig. 2a). In the next step, an analyst investigates clusters seen in the visualized latent patterns (Fig. 2b). For this step, our method utilizes an interactive DR method designed for cluster comparison, specifically, unified linear comparative analysis (ULCA) [11], and provides auxiliary visualizations. Afterward, based on their interest, analysts can apply the extraction of latent patterns to a subpart of the tensor and repeat the same procedure of the latent pattern analysis. This analysis loop allows iterative reduction of the size of the analyzing tensor to refine the DR results. To support this exploratory data analysis, our method provides an interactive visual interface, as shown in Fig. 1.

The main differences from the MulTiDR framework are (1) enhancing flexible latent pattern analysis utilizing ULCA (Fig. 2b), instead of ccPCA [9] used in the MulTiDR framework; (2) integrating an iterative analysis loop to select subtensors and refine DR results (the arrows between Fig. 2a and Fig. 2b); and (3) providing a visual interface that supports our new analysis workflow (Fig. 1).

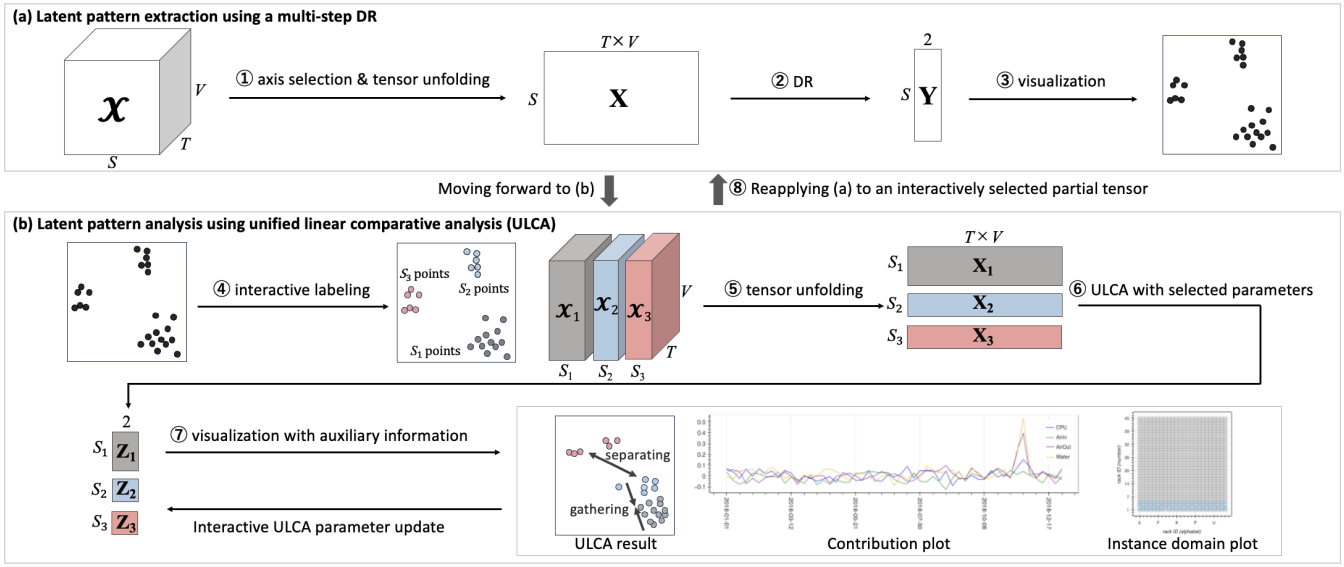

Figure 2: Overview of our method: (a) multivariate time-series data represented as a third-order tensor is projected in a 2D space utilizing a multi-step DR method; (b) patterns in the projected result are investigated through interactive comparative analysis; and these analysis steps are iteratively performed as annotated by ⑧.

3.1 Tensor Representation of Multivariate Time Series

Multivariate time-series data consists of three aspects: (1) instances, (2) variables, and (3) time points. In this study, we assume all instances and variables share the same set of time points. Such data can be modeled as a third-order tensor where three axes correspond to instances, variables, and time points.

Notation. Following the conventions [14], we denote scalars, vectors, matrices, and tensors with lowercase (e.g., x), boldface lowercase (e.g., $\mathbf{x}$), boldface uppercase (e.g., $\mathbf{X}$), and boldface Euler script (e.g., $\mathcal{X}$) letters, respectively. We use indices $s = 1,\ldots,S$, $v = 1,\ldots,V$, and $t = 1,\ldots,T$ for instances, variables, time points, respectively. S, V, and T are axis lengths. In this context, a third-order tensor is described $\mathcal{X} \in \mathbb{R}^{S\times V\times T}$.

3.2 Latent Pattern Extraction

Our method first extracts latent patterns from a third-order tensor. For this extraction, we employ a procedure similar to the multi-step DR used by Fujita et al. [7] as it produces latent patterns that can be visualized as a 2D scatterplot. By visualizing latent patterns in a 2D scatterplot, our method enables analysts to interactively investigate the extracted patterns (e.g., similar instances or time points).

The latent pattern extraction has three steps: tensor unfolding [14] to convert a tensor to a matrix (Fig. 2①), DR on the matrix (Fig. 2②), and visualization of the DR result (Fig. 2③).

Tensor unfolding. Tensor unfolding converts a tensor $\mathcal{X}$ to a matrix $\mathbf{X}$ so that any DR method can be applied. To perform tensor unfolding, analysts first select one axis of a tensor. Fig. 2a① shows an example of tensor unfolding when the instance axis is selected. By slicing $\mathcal{X}$ along one of the non-selected axes, we obtain S matrices of size $T \times V$ in Fig. 2a①. These matrices are then concatenated and generate a matrix with S rows and $(T \times V)$ columns.

Dimensionality reduction. We apply DR to the unfolded matrix $\mathbf{X}$ to extract 2D latent patterns. As a DR method, we use UMAP [17] due to two reasons. First, multivariate time-series data may have complex relationships among instances, variables, and time points; thus, using a DR method that can capture nonlinear patterns such as UMAP is reasonable. Second, multivariate time-series data tends to have a large data size, and UMAP's high computational efficiency is suitable for interactive analysis. Through the DR process, the matrix $\mathbf{X}$ is transformed into a 2D matrix $\mathbf{Y}$ (e.g., $\mathbf{Y} \in \mathbb{R}^{S\times 2}$ in Fig. 2).

Visualization. We visualize the 2D matrix with a scatterplot, as shown in Fig. 1a. Due to the use of UMAP, similar elements are placed closer together in the scatterplot. Consequently, by reviewing the scatterplot, we can find patterns such as clusters and outliers.

3.3 Latent Pattern Analysis

Analysis of latent patterns extracted by DR requires (1) identifying and (2) characterizing clusters [4]. In our method, an analyst visually identifies clusters from the scatterplot and manually labels them (Fig. 2④). Afterward, our method helps the analyst characterize the identified clusters by utilizing a comparative analysis method in conjunction with visualizations (Fig. 2⑤–⑦). We describe how our method helps analysts characterize clusters.

3.3.1 Comparative Analysis of Clusters of Tensors

To analyze the identified clusters, we utilize a method called unified linear comparative analysis (ULCA) [11]. ULCA integrates two DR schemes: discriminant analysis and contrastive learning [1]. Discriminant analysis and contrastive learning help uncover differentiating factors of clusters and more abundant factors in one cluster than others, respectively. By incorporating both schemes, ULCA can flexibly find various essential factors such as the similarity and dissimilarity of clusters. In addition, ULCA is a linear DR method, and the result obtained by ULCA is easy to interpret by referring to a projection matrix (see the description of ULCA below for details).

After labeling clusters, from the original tensor $\mathcal{X}$, our method extracts subtensors $\mathcal{X}_1,\ldots,\mathcal{X}_K$, each of which corresponds to one cluster. Here K denotes the number of selected clusters. For example, in Fig. 2b, three clusters are selected and correspond to $\mathcal{X}_1$, $\mathcal{X}_2$, and $\mathcal{X}_3$. As ULCA can be only applied to a matrix, we apply tensor unfolding to the subtensors (Fig. 2⑤) and generate a submatrix $\mathbf{X}_i$ corresponding to $\mathcal{X}_i$ where $i = \{1,\ldots,K\}$. Afterward, the submatrices are projected into a 2D space with ULCA.

Unified Linear Comparative Analysis (ULCA). We provide brief introduction of ULCA [11]. ULCA enables the interactive adjustment of variance within and distance between clusters by incorporating discriminant analysis and contrastive learning. ULCA

performs the optimization below to derive a projection matrix $\mathbf{M}$. Then, ULCA transforms a submatrix $\mathbf{X}_i$ into a low-dimensional representation $\mathbf{Z}_i$ by computing $\mathbf{Z}_i = \mathbf{X}_i\mathbf{M}$ (Fig. 2⑥). The resultant coordinates of $\mathbf{Z}_i$ are called principal components (PCs).

ULCA's optimization problem can be written as follows:

$$\max_{\mathbf{M}^\top\mathbf{M}=\mathbf{I}_{D'}} = \frac{\mathrm{tr}(\mathbf{M}^\top\mathbf{C}_0\mathbf{M})}{\mathrm{tr}(\mathbf{M}^\top\mathbf{C}_1\mathbf{M})} \tag{1}$$

$$\mathbf{C}_0 = \sum_{i=1}^{K} w_{\mathrm{tg}_i}\mathbf{C}_{\mathrm{wi}_i} + \sum_{i=1}^{K} w_{\mathrm{bw}_i}\mathbf{C}_{\mathrm{bw}_i} + \gamma_0\mathbf{I}_D \tag{2}$$

$$\mathbf{C}_1 = \sum_{i=1}^{K} w_{\mathrm{bg}_i}\mathbf{C}_{\mathrm{wi}_i} + \gamma_1\mathbf{I}_D \tag{3}$$

where D and D' are the numbers of dimensions of $\mathbf{X}_i$ and $\mathbf{Z}_i$, respectively (e.g., $D = T \times V$ and $D' = 2$ in Fig. 2⑥). $\mathbf{I}_D$ and $\mathbf{I}_{D'}$ are identity matrices of size D and D'. $\mathbf{C}_{\mathrm{wi}_i}$ is a within-cluster covariance matrix of the i-th cluster (i.e., $\mathbf{C}_{\mathrm{wi}_i} = \mathbf{X}_i^\top\mathbf{X}_i/N_i$ where N_i is the number of rows of $\mathbf{X}_i$). $\mathbf{C}_{\mathrm{bw}_i}$ is a between-cluster covariance matrix related to the i-th cluster (i.e., $\mathbf{C}_{\mathrm{bw}_i} = (\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^\top/N_i$ where $\boldsymbol{\mu}$ and $\boldsymbol{\mu}_i$ are the centroids of $\mathbf{X}$ and $\mathbf{X}_i$). w_{tg_i}, w_{bg_i}, and w_{bw_i} are weights to control how much the i-th cluster's variance should maintain (w_{tg_i}) or eliminate (w_{bg_i}) as well as how much the i-th cluster should be separated from the other clusters (w_{bw_i}). Lastly, γ_0 and γ_1 are set to zero by default, while ULCA sets $\gamma_j = 1$ when $\mathrm{tr}(\mathbf{M}^\top\mathbf{C}_j\mathbf{M}) = 0$.

ULCA's analysis flexibility stems from the adjustable weights, w_{tg_i}, w_{bg_i}, and w_{bw_i}. Fig. 3 demonstrates how the within-cluster and between-cluster variances change by adjusting the weights. Fig. 3a shows the weight parameters that minimize all within-cluster variances and maximize the between-cluster variances, which yield the same result as linear discriminant analysis (LDA). In Fig. 3b, by decreasing w_{bw_2} and w_{bw_3}, the distance between Clusters 2 and 3 is reduced while maintaining the separation from Cluster 1. Fig. 3c tries to maintain the between-cluster variances and Cluster 3's within-class variance as much as possible while reducing the other clusters' within-class variances. By adjusting the weight parameters, for example, analysts can identify factors that make all clusters different (e.g., Fig. 3a), factors that significantly differentiate one cluster from others (e.g., Fig. 3b), or factors that are salient in one cluster while differentiating all clusters (e.g., Fig. 3c). Note that, in contrast to ULCA, ccPCA employed by MulTiDR can only compare two groups (e.g., Cluster 1 and others) and corresponds to ULCA with a fixed parameter (i.e., $\mathbf{w}_{\mathrm{tg}}$=(1,1), $\mathbf{w}_{\mathrm{bg}}$=(0,1), $\mathbf{w}_{\mathrm{bw}}$=(0,0)), limiting latent pattern analysis flexibility.

As in other linear DR methods, ULCA provides the interpretability of the results. Each column of a projection matrix, $\mathbf{M}$, indicates the original dimensions' contributions to each PC. By referring to these contributions, analysts can grasp which dimensions are highly related to clusters' differences, similarities, etc. Our approach begins by applying a nonlinear DR method (specifically, UMAP) to uncover latent patterns. The rationale behind using nonlinear DR is its ability to effectively identify comparison groups in complex, high-dimensional data. Then, as long as the comparison groups are clearly defined, our subsequent interpretation using ULCA provides consistent interpretability, regardless of the underlying data characteristics or the employed nonlinear DR method.

Automatic PC rotation adjustment. One limitation of ULCA is the arbitrary rotation of PCs caused by its optimization solver (refer to Fujiwara et al.'s work [11] for details). To address this limitation, we introduce automatic adjustment of the rotation of PCs. The adjustment first applies linear regression to the projected elements and obtains the fitted linear line of the direction vector, $\mathbf{v}$. Then, the adjustment rotates PCs by updating $\mathbf{Z}_i$ and $\mathbf{M}$ as follows: $\mathbf{Z}_i \leftarrow \mathbf{Z}_i\mathbf{v}/||\mathbf{v}||$ and $\mathbf{M} \leftarrow \mathbf{M}\mathbf{v}/||\mathbf{v}||$.

This adjustment also provides an analytical benefit. As the adjustment rotates PCs such that the fitted line direction matches with the first PC, elements are also rotated to have the highest scatteredness along the first PC. The direction related to the high scatteredness can be expected to correspond to analytically interesting patterns, such as cluster separations. Consequently, the interpretation process can often focus on the first PC.

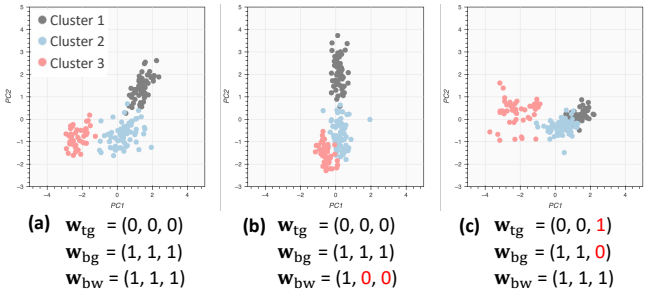

Figure 3: Examples of ULCA results using different weights. We apply ULCA to the Wine dataset [6].

3.3.2 Visualizations for Comparative Analysis

Our method provides a set of visualizations that aid interactive analysis of latent patterns using ULCA. The visualizations include a domain plot depicting the distribution of elements in their domain (Fig. 1b), a scatterplot of the ULCA result (Fig. 1c), and a contribution plot showing the information of the projection matrix (Fig. 1d).

Domain plots. To analyze the identified clusters, we often want to understand what kind of elements are included in each cluster. Domain plots are designed to support this task. We provide two different plots for (1) time points (Fig. 1b1) and (2) instances (Fig. 1b2).

When the time axis is selected during the latent pattern extraction (Sec. 3.2), the time domain plot informs which time points are included in each cluster. The plot uses x- and y-coordinates to represent timestamps and cluster IDs, respectively.

The instance domain plot depicts the information related to instances when the instance axis is selected. Such information can be a spatial distribution of each cluster's instances when the location data is available. For example, when analyzing supercomputer log data, we can show locations of system nodes corresponding to instances. The instance domain plot facilitates intuitive understanding by visualizing the data in its original physical space, allowing analysts to interpret clusters in context of their domain knowledge. Unlike the time domain plot, the instance domain plot should be tailor-made based on a dataset (e.g., a map for geographical data and a compute rack position for supercomputer log data). Tailoring the instance domain plot to a specific dataset is straightforward and can be easily implemented using a plotting library (e.g., Bokeh). MulTiDR provides open-source examples of such tailored plots [10].

ULCA result and contribution plots. We visualize PCs extracted by ULCA as a scatterplot (Fig. 1c). Contribution plots (Fig. 1d) convey the information of the projection matrix, $\mathbf{M}$, which is necessary to understand the PCs. We apply a different design based on the selected axis for the latent pattern extraction.

When the time axis is selected, the original dimensions correspond to the combination of instances and variables (e.g., Fig. 2⑥). For each variable, we color-code the corresponding values in $\mathbf{M}$ with instances' spatial locations, as shown in Fig. 1d1.

In contrast, selecting the instance axis produces $\mathbf{X}$ with dimensions of the combination of time points and variables. For this case, we visualize $\mathbf{M}$ as multiline charts, where each polyline corresponds to one variable, as shown in Fig. 1d2.

3.4 Iterative Update

We enhance the analysis workflow of MulTiDR by adding an iterative update step where our method performs the latent pattern extraction based on an interactively selected subtensor (Fig. 2⑧).

The subtensor can be one of the identified clusters in the previous steps or can be newly selected from the UMAP and ULCA results.

This step is useful to mitigate excessive data compression, which is MulTiDR's main limitation stemming from the high dimensionality of the unfolded tensor (e.g., 10 variables and 100 time points produce 1000 dimensions). For example, the analyst can select a part of instances for further analysis from an initial exploration. In the next iteration, the analyst can investigate similarities of time points of the corresponding subtensor. For instance, when the analyst selects $\mathcal{X}_3$ in Fig. 2, an unfolded tensor for this second iteration can be a matrix with size $T \times (S_3 \times V)$, which has much smaller dimensions than $T \times (S \times V)$ when $S_3 \ll S$. Consequently, DR is performed only on $S_3 \times V$, instead of $S \times V$.

3.5 System Implementation

We have developed a web-based visual interface (Fig. 1), consisting of the aforementioned plots for latent pattern extraction and analysis. Each plot is fully linked and provides necessary interactions to perform an analysis workflow shown in Fig. 2, such as the adjustment of ULCA parameters and selection of points. Both back-end and front-end are implemented with Python with Bokeh.

4 Case Study

We demonstrate the effectiveness of our method by analyzing a supercomputer operational log dataset. This dataset has been analyzed in previous studies [7, 10]. The analysis of the same dataset can facilitate the evaluation of our method's improvements. The case study has been performed with an operational staff working for the supercomputer center.

Dataset. We analyzed the daily operational log dataset collected from a hybrid water/air-cooled supercomputer, named the K computer. The dataset consists of daily average temperature measurements collected at 864 compute racks. The racks received cooled water and air to extract the heat generated by the compute nodes. Four temperature measurements are used for our analysis: intake air temperature (AirIn), exhaust air temperature (AirOut), input cooling water temperature (Water), and the average temperature of CPUs (CPU). While the previous study [7] focused only on one fiscal year, we analyzed three consecutive fiscal years (April 2014 to March 2017). The resulting third-order tensor consists of 864 instances/racks, 4 variables/measurements, and 1,086 time points/days (i.e., 3,753,216 elements in total).

Analysis. We performed an analysis to respond to the supercomputer center's interest in understanding the seasonal water/air cooling behavior during the regular operational period. Their interest stemmed from the fact that the lifespan of water/air cooling facility can be much longer than the supercomputer itself. In fact, their cooling facility for the K computer has been reused for the newly developed supercomputer, Fugaku [21].

Considering the interest in temporal behavior, we selected the time axis when applying the multi-step DR. Fig. 4a shows the UMAP result of 1,086 time points/days. We interactively selected six apparent clusters and labeled them from Clusters 1 to 6. As shown in Fig. 4b, the time domain plot highlighted all clusters mostly are composed of contiguous days.

We first applied ULCA with parameters that generate the same result as LDA to see distinguishing factors among all clusters. This ULCA result (Fig. 5a) highlights that Clusters 1 and 4 are clearly different from the other clusters. By referring to this ULCA result as well as Fig. 4a and Fig. 4b, we decided to focus more on analyzing Clusters 1 and 4 as these clusters have unique properties: they are composed of a small number of time points, located far away from others in the UMAP result, and constituted by contiguous days with each other. Due to the analysis flexibility provided by ULCA, the differentiating factors of these two clusters can be easily investigated by adjusting the weight parameters. We adjusted the

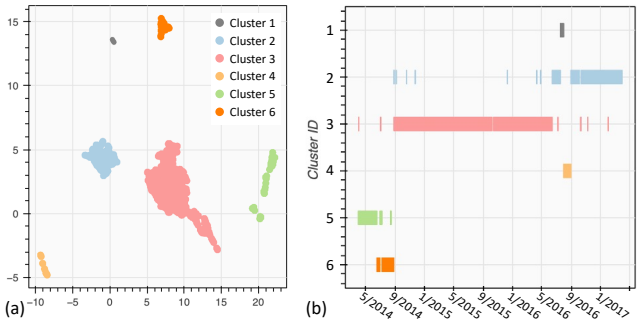

Figure 4: Latent patterns of the supercomputer operational log dataset: (a) the UMAP result and (b) the time domain plot. Each cluster mainly consists of consecutive days.

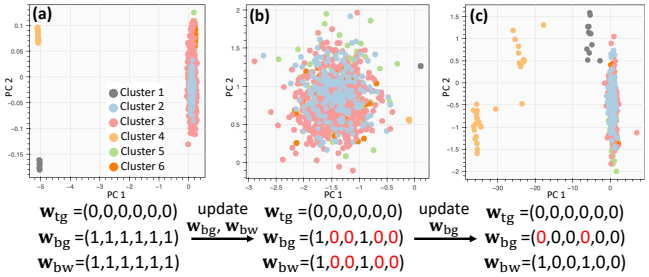

Figure 5: Comparative analysis of the timestamp clusters through adjustments of the weight parameter in ULCA. After the adjustment (c), we see a clear separation between Clusters 1 and 4 along PC 1 as well as subclusters of Cluster 4.

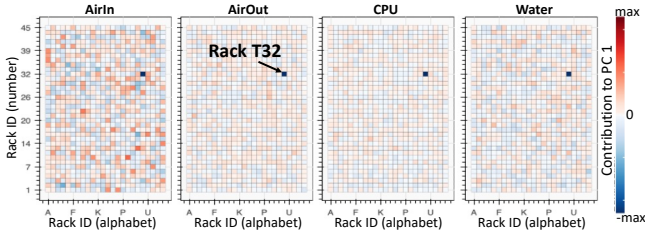

Figure 6: Contribution plots for PC 1 of Fig. 5c. For all four temperature measurements, Rack T31 shows significant negative contributions to PC 1.

weight parameters to emphasize differentiating factors of Clusters 1 and 4, as shown in Fig. 5b. The new parameters mimic a case where LDA is applied to only Clusters 1 and 4. While this ULCA result shows clear separation among Cluster 1, Cluster 4, and others along PC 1, the variances of Clusters 1 and 4 are minimized—limiting the available information related to Clusters 1 and 4. Thus, as shown in Fig. 5c, we further adjusted weights to avoid minimizing the variances of Clusters 1 and 4. This result still shows the clear separations shown in Fig. 5b while revealing two subclusters within Cluster 4. Note that the result shown in Fig. 5c cannot be derived with the other linear DR methods such as PCA, LDA, and ccPCA.

From the above observation, we decided to review the contributions of the dimensions to PC 1, as shown in Fig. 6. We confirmed that the significant negative contributions correspond to four temperature measurements for Rack T32. This result indicates that Rack T32 has high associations with the difference among Cluster 1, Cluster 4, and the others as well as the subclusters of Cluster 4.

To further investigate patterns only related to Clusters 1 and 4,

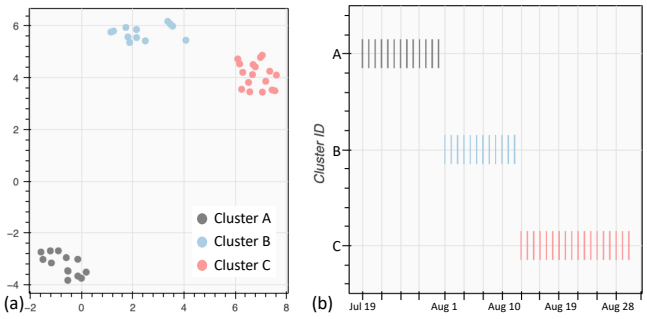

Figure 7: The second iteration of latent pattern extraction after only selecting Clusters 1 and 4: (a) the UMAP result and (b) the time domain plot. Each of the newly found clusters, Clusters A–C, only consists of continuous days.

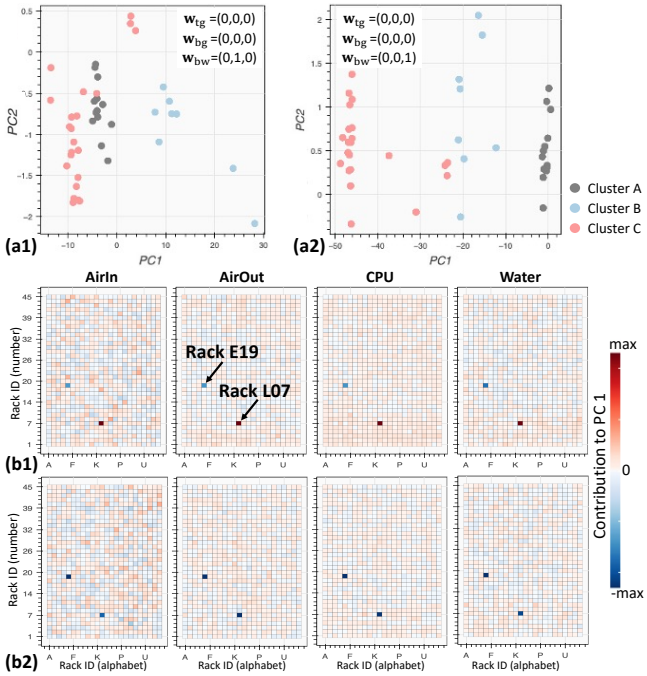

Figure 8: The second iteration of comparative analysis: (a1, b1) the ULCA result and contribution plots when Cluster B is made distant from others; (a2, b2) the same set of plots when Cluster C is made distant from others. The results indicate that Rack E19 and Rack L07 have significant influences on the differences of each cluster.

we utilized the interactive update our method supports (Sec. 3.4) and extracted a subtensor corresponding to Clusters 1 and 4. This subtensor consists of only 43 time points, which is significantly sized down from 1,086 time points. Similar to the initial analysis stage, we selected the time axis, producing the UMAP result and time domain plots shown in Fig. 7. As we observed three clusters in the UMAP result (Fig. 7a), we labeled them as Clusters A, B, and C. From the time domain plot (Fig. 7b), we also noticed that these three clusters only consist of continuous days.

We examined the differentiating factors of each cluster by using various weight parameters of ULCA. Fig. 8a1 shows the ULCA result when the parameters are set to separate Cluster B from others. The corresponding contribution plots (Fig. 8b1) show that Racks E19 and L07's significantly positive and negative contributions. In contrast, Fig. 8a2 and b2 show the ULCA result and contribution plots when separating Cluster C from others. In this case, both Racks E19 and L07 have significantly negative contributions.

By reviewing the temperature data for the three racks identified through the two iterative analyses (i.e., Racks T32, E19, L07), we confirmed that these racks had issues that prevented correct temperature measurements from July to August 2016. These results are not found in the existing study [7] due to its limited capability of latent pattern analysis such as cluster comparison. The analysis processes and results demonstrate the effectiveness of our method in reviewing an extremely large dataset (3,753,216 elements).

5 Discussion

Through the case study, we demonstrated the usefulness of our method's analysis capabilities for a deeper understanding of large-scale multivariate time-series data. Our method's strengths stem from the newly integrated comparative analysis and the iterative analysis workflow designed to perform more precise and detailed analysis. In fact, while performing the case study, the operational staff was convinced regarding the effectiveness of the visual analytics approach taken by our method.

Limitation. Our method heavily involves humans for multivariate time-series analysis. While this approach can incorporate domain knowledge, this involvement can be laborious for analysts. For instance, our method requires manual selection of clusters, adjustment of ULCA parameters, and iterative updates. Adjusting and interpreting ULCA results also requires basic knowledge of ULCA. In our case, we provided such knowledge to the domain expert through a brief demonstration of our system, along with examples of ULCA results (e.g., Fig. 5). In future work, we would like to reduce analysis workload by developing recommendation systems. For example, such systems could suggest parameter sets such that each set shows significantly different patterns, similar to existing works [8, 15].

6 Conclusion

We introduced a method that conjointly uses interactive visualization and machine learning to analyze multivariate time-series data. As machine learning methods, we specifically employed multiple interactive DR methods and applied them to a third-order tensor that represents multivariate time-series data. Our method's functionality of interactive cluster identification and comparative analysis of clusters enables flexible analysis of latent patterns. In addition, the workflow incorporating iterative updates allows more precise and detailed analysis of a specific domain of interest. We performed a case study using supercomputer operational log data, highlighting the effectiveness of our method. Our work contributes to demonstrating how machine learning can help analysts review complex data (e.g., interactive DR-based latent pattern analysis) and how analysts can supplement the limitations of machine learning (e.g., data reduction involving domain knowledge).

Acknowledgments

This work has been supported in part by the Knut and Alice Wallenberg Foundation through Grant KAW 2019.0024.

References

[1] A. Abid, M. J. Zhang, V. K. Bagaria, and J. Zou. Exploring patterns enriched in a dataset with contrastive principal component analysis. *Nat Commun*, 9(1):2134, 2018. doi: 10.1038/s41467-018-04608-8

[2] M. Ashraf, F. Anowar, J. H. Setu, A. I. Chowdhury, E. Ahmed, et al. A survey on dimensionality reduction techniques for time-series data. *IEEE Access*, 11:42909–42923, 2023. doi: 10.1109/ACCESS.2023.3269693

[3] B. Bach, P. Dragicevic, D. Archambault, C. Hurter, and S. Carpendale. A descriptive framework for temporal data visualizations based on generalized space-time cubes. *Comput Graph Forum*, 36(6):36–61, 2017. doi: 10.1111/cgf.12804

[4] M. Brehmer, M. Sedlmair, S. Ingram, and T. Munzner. Visualizing dimensionally-reduced data: Interviews with analysts and a characterization of task sequences. In *Proc BELIV*, pp. 1–8, 2014. doi: 10.1145/2669557.2669559

[5] N. Cao, C. Lin, Q. Zhu, Y.-R. Lin, X. Teng, and X. Wen. Voila: Visual anomaly detection and monitoring with streaming spatiotemporal data. *IEEE Trans Vis Comput Graph*, 24(1):23–33, 2017. doi: 10.1109/TVCG.2017.2744419

[6] D. Dua and C. Graff. UCI machine learning repository. http://archive.ics.uci.edu/ml, 2019.

[7] K. Fujita, N. Sakamoto, T. Fujiwara, T. Tsukamoto, and J. Nonaka. A visual analytics method for time-series log data using multiple dimensionality reduction. *J Adv Simul Sci Eng*, 9(2):206–219, 2022. doi: 10.1007/978-981-19-6857-0_3

[8] T. Fujiwara, Y.-H. Kuo, A. Ynnerman, and K.-L. Ma. Feature learning for nonlinear dimensionality reduction toward maximal extraction of hidden patterns. In *Proc. PacificVis*, pp. 122–131, 2023. doi: 10.1109/PacificVis56936.2023.00021

[9] T. Fujiwara, O.-H. Kwon, and K.-L. Ma. Supporting analysis of dimensionality reduction results with contrastive learning. *IEEE Trans Vis Comput Graph*, 26(1):45–55, 2020. doi: 10.1109/TVCG.2019.2934251

[10] T. Fujiwara, N. Sakamoto, J. Nonaka, K. Yamamoto, and K.-L. Ma. A visual analytics framework for reviewing multivariate time-series data with dimensionality reduction. *IEEE Trans Vis Comput Graph*, 27(2):1601–1611, 2020. doi: 10.1109/TVCG.2020.3028889

[11] T. Fujiwara, X. Wei, J. Zhao, and K.-L. Ma. Interactive dimensionality reduction for comparative analysis. *IEEE Trans Vis Comput Graph*, 28(1):758–768, 2021. doi: 10.1109/TVCG.2021.3114807

[12] P. Hewage, A. Behera, M. Trovati, E. Pereira, M. Ghahremani, F. Palmieri, and Y. Liu. Temporal convolutional neural (TCN) network for an effective weather forecasting using time-series data from the local weather station. *Soft Comput*, 24:16453–16482, 2020. doi: 10.1007/s00500-020-04954-0

[13] D. Jäckle, F. Fischer, T. Schreck, and D. A. Keim. Temporal MDS plots for analysis of multivariate data. *IEEE Trans Vis Comput Graph*, 22(1):141–150, 2016. doi: 10.1109/TVCG.2015.2467553

[14] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, 2009. doi: 10.1137/07070111X

[15] D. J. Lehmann and H. Theisel. Optimal sets of projections of high-dimensional data. *IEEE Trans Vis Comput Graph*, 22(1):609–618, 2016. doi: 10.1109/TVCG.2015.2467132

[16] D. Liu, P. Xu, and L. Ren. TPFlow: Progressive partition and multidimensional pattern extraction for large-scale spatio-temporal data analysis. *IEEE Trans Vis Comput Graph*, 25(1):1–11, 2018. doi: 10.1109/TVCG.2018.2865018

[17] L. McInnes, J. Healy, and J. Melville. UMAP: Uniform manifold approximation and projection for dimension reduction. *arXiv:1802.03426*, 2018. doi: 10.48550/arXiv.1802.03426

[18] H. Miyazaki, Y. Kusano, N. Shinjou, F. Shoji, M. Yokokawa, and T. Watanabe. Overview of the K computer system. *Fujitsu Scientific & Technical Journal*, 48(3):302–309, 2012. https://www.fujitsu.com/global/documents/about/resources/publications/fstj/archives/vol48-3/paper02.pdf.

[19] B. H. Park, Y. Hui, S. Boehm, R. A. Ashraf, C. Layton, and C. Engelmann. A big data analytics framework for HPC log data: Three case studies using the Titan supercomputer log. In *Proc CLUSTER*, pp. 571–579, 2018. doi: 10.1109/CLUSTER.2018.00073

[20] P. E. Rauber, A. X. Falcão, and A. C. Telea. Visualizing time-dependent data using dynamic t-SNE. *Proc EuroVis*, 2(5):73–77, 2016. doi: 10.2312/eurovisshort.20161164

[21] M. Sato, Y. Ishikawa, H. Tomita, Y. Kodama, T. Odajima, et al. Co-design for A64FX manycore processor and "Fugaku". In *Proc SC*, pp. 1–15, 2020. doi: 10.1109/SC41405.2020.00051

[22] Shilpika, T. Fujiwara, N. Sakamoto, J. Nonaka, and K.-L. Ma. A visual analytics approach for hardware system monitoring with streaming functional data analysis. *IEEE Trans Vis Comput Graph*, 28(6):2338–2349, 2022. doi: 10.1109/TVCG.2022.3165348

[23] F. Shilpika, B. Lusch, M. Emani, V. Vishwanath, M. E. Papka, and K.-L. Ma. MELA: A visual analytics tool for studying multifidelity HPC system logs. In *Proc DAAC*, pp. 13–18, 2019. doi: 10.1109/DAAC49578.2019.00008

[24] E. F. Vernier, J. L. D. Comba, and A. C. Telea. Guided stable dynamic projections. *Comput Graph Forum*, 40(3):87–98, 2021. doi: 10.1111/cgf.14291

[25] N. Wu, B. Green, X. Ben, and S. O'Banion. Deep transformer models for time series forecasting: The influenza prevalence case. *arXiv:2001.08317*, 2020. doi: 10.48550/arXiv.2001.08317

[26] P. Xu, H. Mei, L. Ren, and W. Chen. ViDX: Visual diagnostics of assembly line performance in smart factories. *IEEE Trans Vis Comput Graph*, 23(1):291–300, 2017. doi: 10.1109/TVCG.2016.2598664