IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 6, 2021, accepted July 8, 2021, date of publication July 20, 2021, date of current version July 27, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3098621

A Visual Analytics Interface for Formulating Evaluation Metrics of Multi-Dimensional Time-Series Data

REI TAKAMI[1,2], HIROKI SHIBATA[1], (Member, IEEE), AND YASUFUMI TAKAMA[1], (Member, IEEE)

[1]Graduate School of System Design, Tokyo Metropolitan University, Tokyo 191-0065, Japan
[2]Yahoo Japan Corporation, Tokyo 102-8282, Japan

Corresponding authors: Rei Takami (takamiray@gmail.com) and Yasufumi Takama (ytakama@tmu.ac.jp)

This work was supported in part by JSPS KAKENHI under Grant 19K22896.

ABSTRACT A visual analytics (VA) interface for formulating evaluation metrics of multi-dimensional time-series data is proposed. Evaluation metrics such as key performance indicators (KPI) are expected to play an important role in quantitatively evaluating current situations and the quality of target objects. However, it is difficult for even domain experts to formulate metrics, especially for data with complexity related to dimensionality and temporal characteristics. The proposed interface is designed by extending the concept of semantic interaction to consider the temporal characteristics of target data. It represents metrics as a linear combination of data attributes and provides a means for adjusting it through interactive VA. On an animated scatter plot, an analyst can directly manipulate several visualized objects, i.e., a node, a trajectory, and a convex hull, as the group of nodes and trajectories. The result of manipulating the objects is reflected in the linear combination of attributes, which corresponds to an axis of the scatter plot. Using the axes as the output of the analysis, analysts can formulate a metric. The effectiveness of the proposed interface is demonstrated through an example and evaluated by two user experiments on the basis of hypotheses obtained from the example.

INDEX TERMS Data visualization, data analysis, evaluation metrics, graphical user interfaces, human–computer interaction, time-series data, visual analytics.

I. INTRODUCTION

A visual analytics (VA) interface for formulating evaluation metrics for multi-dimensional time-series data is proposed. In many domains, such as medical, sports, and business intelligence, the importance of multi-dimensional time-series data is increasing. To utilize those data, evaluation metrics such as key performance indicators (KPI) [1] play an important role in various tasks involving decision making [2] and hypothesis generation: they are used to quantitatively evaluate current situations, the quality of target objects, etc. This paper defines an evaluation metric as an interpretable, attribute-based representation of user-defined criteria. As an example of this type of metric, sabermetrics, which is the empirical analysis of baseball, provides several metrics for evaluating the performance of batters by combining multiple basic data (attributes) [3].

Evaluation metrics should be formulated to be meaningful in a target domain emphasizing interpretability rather than accuracy. They are therefore generally formulated by domain experts. However, metrics based on subjective knowledge may include confirmation bias: the analysts subconsciously pay particular attention to a phenomenon that confirms a pre-existing hypothesis [4]. On the contrary, metrics defined only on the basis of data may lead to overfitting problems. Visual analytics (VA) [5] is expected to support the process of formulating metrics (“metrics formulation,” hereafter) by considering both the evidence obtained from data and the cognitive ability of a domain expert. When VA is applied to multi-dimensional time-series data, preprocessing such as dimensionality reduction and clustering is required to reduce the dimension of the data. However, domain experts lack knowledge about the algorithms used for that preprocessing, so they have difficulty in selecting an appropriate algorithm for the preprocessing, adjusting those parameters, and confirming the effect of the parameters on the preprocessing results [6]. To overcome this difficulty, a human-in-the-loop approach, namely, semantic interaction, has been proposed [7]. A system based on the concept of semantic interaction interprets the intention of the

The associate editor coordinating the review of this manuscript and approving it for publication was Hualong Yu.

user's direct manipulation of visualized objects, adjusts the parameters of the model in accordance with the estimated intention, and gives feedback to the user by redrawing the visualized objects [8]. However, to the authors' knowledge, explicitly incorporating semantic interaction into multi-dimensional time-series data has not yet been studied. To extend semantic interaction for multi-dimensional time-series data, this paper assumes three requirements: (R1) flexible adjustment of temporal range of parameter based on both characteristics of specific time point and entire time range; (R2) incremental metrics formulation on a single view; and (R3) addressing specific problems when directly manipulating visualized time-series data [9].

As for existing studies on VA that explicitly consider the relationship between different attributes of multi-dimensional data, a method for formulating linear functions and projecting multi-dimensional data into low-dimensional space defined by the user's examples was proposed by Gleicher [10]. Metrics formulation can also be interpreted as the construction of ranking criteria that define relative importance of each data instance. As an example of VA for constructing ranking criteria, an interface called Podium, which re-learns ranking criteria on the basis of how the user sorts the results of tabular visualization of ranking results, was proposed by Wall *et al.* [11]. However, these studies do not support the task of metrics formulation explicitly, and they do not consider characteristics specific to time-series data, such as temporal trends [12] and trade-offs between visualization methods [13].

In this paper, to support metrics formulation of multi-dimensional time-series data by interactive VA, semantic interaction is extended to satisfy the above-mentioned R1-R3. In particular, a VA interface that expresses the metrics as a linear combination of attributes at an arbitrary time point or range is proposed. Via that VA interface, instances of target data are projected onto a 2D space (scatter plot), each axis of which corresponds to a basis for a metric. An instance in a dataset is represented as three types of visualized objects that are manipulatable by users: (1) a node representing an instance at a specific time point; (2) a trajectory representing the time-series of an instance; and (3) a convex hull representing a group of nodes and trajectories. Different types of visualized objects are associated with different temporal and spatial ranges: a node with a specific temporal and/or spatial point, a trajectory with a time range, and a convex hull with a spatial region; this association contributes to the efficient exploration of the target data. Visualized objects of instances can be moved to arbitrary positions on a scatter plot so that the data distribution on the scatter plot reflects the criteria that the user supposes. The positions of objects on the scatter plot are modeled with global and local parameters, which are used to update the axes. By iteratively adjusting these parameters, the analyst can incrementally acquire ideas for metrics formulation.

As an example of applying the proposed interface to real-world data, metrics formulation using statistical data of a baseball game are presented hereafter. On the basis of the hypothesis obtained through the example, the effectiveness of the proposed interface was experimentally evaluated. The participants in the experiment were asked to use the prototype interface to find similar instances of multi-dimensional time-series data and adjust the linear functions corresponding to the axes of the scatter plot. The experimental results show that the prototype interface enables exploration of multi-dimensional time-series data by appropriately combining target visualized objects. Using the proposed interface, the participants could efficiently reflect their intentions as a data distribution on the scatter plot.

The proposed interface is the first attempt at applying VA to metrics formulation of multi-dimensional time-series data. The main contributions of this paper are summarized as follows:

- A VA interface to support metrics formulation of multi-dimensional time-series data based on the concept of semantic interaction is proposed.
- The design principles necessary to formulate evaluation metrics for time-series data are organized as an analytical framework that extends the concept of semantic interaction to time-series data.
- An example of an application to metrics formulation of statistical data of a baseball game is shown to illustrate the effective usage of the proposed interface.
- The results of user experiments show the effectiveness of the proposed interface.

The concept of the analytical framework was proposed in [14], in which the prototype interface and the example are described. To supplement that paper, this paper describes the proposed interface in further detail and presents the results of the user experiments.

II. RELATED WORK

A. VISUAL ANALYTICS

For the purpose of data-driven decision making, it is necessary to understand the characteristics of data, form hypotheses about the analysis strategy, and select the algorithm appropriately for the analysis. Visualization techniques are used to interpret the intrinsic properties of large amounts of data and enable decision making [15]. However, when analyzing large complex data, such as multi-dimensional and time-series data, it is necessary to reduce the amount of data and the number of attributes by preprocessing [16]. It is also said to be difficult to conduct effective data analysis with only a single visualization. Therefore, VA, which provides a means to interact with data-visualization results in a way that assists sensemaking and analytical reasoning on the basis of domain knowledge, has been studied [5].

When executing complex VA tasks, domain experts evaluate the validity of analytical results on the basis of their knowledge [17]. However, if they lack expertise in algorithmic models, such as dimensionality reduction, issues concerning fundamental usability [8] of VA will occur.

For promoting tighter collaboration between humans and computers, mixed-initiative systems have been proposed and applied domains such as machine learning and agent systems. Through a user interface, a domain expert can understand the intermediate results of a computational model through visualization and adjust the model according to that understanding. The computer improves the model on the basis of the expert's purpose inferred from the interaction [18], [19]. However, as for existing mixed-initiative systems, users may control the parameters of the models by using an indirect UI such as a toolbar [20]. To appropriately set the parameters of the model through such UIs, users must know a reasonable range of values. Thus, adjusting the parameters is a difficult task for domain experts when the visualization results generated by preprocessing differ from the experts' intentions [6], [7]. To deal with these issues, recent studies proposed a mixed-initiative VA approach that can adjust the parameters through the user's direct manipulation of the initial output of the VA [21]. For example, differing from the above-mentioned conventional mixed-initiative approach that specifies parameters indirectly, an analytical framework, called ''semantic interaction,'' based on direct manipulation of visualized objects, was proposed by Endert *et al.* [7]. Semantic interaction has been applied to text mining [22] and dimensionality reduction of multi-dimensional data [23].

B. VISUALIZING MULTI-DIMENSIONAL DATA

Due to limitations on screen area and human cognitive ability, it is difficult to display multi-dimensional data in more than three dimensions directly on a screen [16]. Various methods for visualizing multi-dimensional data have therefore been proposed. These visualization methods can be categorized as two main groups: one is to preserve the attributes of the raw data and visualize them in terms such as parallel coordinates and scatterplot matrices; the other is to create a low-dimensional representation by synthesizing multiple attributes of the raw data [24]. As an example of the latter method, dimensionality reduction involves projecting multi-dimensional data into a low-dimensional space, where the coordinates of the target low-dimensional space contain characteristics such as the similarity between data in the original space [25].

The relationship between data and the overall trend of data can be easily grasped from the scatter plot obtained by applying dimensionality reduction [26]. However, dimensionality reduction may produce visualization results that are not consistent with the intention of the analysis. For that reason, a method to adjust a projection algorithm on the basis of domain knowledge through user interactions has been studied [27], [28]. However, many of these algorithms do not take into account the temporal characteristics of data. As for algorithms for visualizing multi-dimensional time-series data, a method of dimensionality reduction for stream data based on incremental principal component analysis (PCA), which calculates the projection of added and updated data instances, was proposed by Fujiwara *et al.* [29].

Parameter adjustments on the basis of concepts such as semantic interaction have been applied to both linear dimensionality reduction [30], [31] and nonlinear dimensionality reduction [32], [33]. InterAxis [31] is a VA interface that allows users to adjust a linear combination of attributes that corresponds to an axis in a manner that allows them to emphasize a target visualized object. Semantic interaction has also been utilized to assist experts in specific domains with their analysis and decision-making tasks, such as security [34] and understanding machine-learning models [35]. However, existing studies are limited to sensemaking on the basis of exploratory analysis, and they do not assume that the adjusted parameters of models are used to formulate evaluation metrics. Furthermore, to the authors' knowledge, semantic interaction has not been applied to multi-dimensional time-series data in any existing studies.

C. VISUALIZING TIME-SERIES DATA

Data whose attribute values change with time, such as regularly collected statistical data, sensor data, and stock-exchange data, are examples of time-series data [12]. The temporal aspect of such data, namely, a timestamp, is stored in a database as one of the attributes. In the case of multi-dimensional data, however, the characteristics of the temporal aspect differ from those of the other attributes, and they need to be taken into account when preprocessing [36] and visualizing the data [37]. In this paper, it is assumed that M-dimensional time-series data $\boldsymbol{D} = \{d_{tnm}\}$ consists of N instances with M attributes recorded at specific time points, each given as t. Many methods for visualizing time-series data have been proposed [13], and they can be categorized as two types: static visualization, by which the temporal changes of data values are represented as visual objects on a static view, and dynamic visualization, by which the temporal changes are represented as the movement of objects by playback of animation [13]. Examples of static visualization include line charts, small multiples [38], and trajectory representation (traces) [39]. Although dynamic visualization has some advantages in terms of presenting and understanding trends in data, it is not suitable for detailed analysis because it causes overlooking problems [40]. On the contrary, although static visualization is effective for exploratory data analysis, it suffers from the problem of visual clutter when large amounts of data are visualized [38]. Therefore, in some studies, an animation is combined with trajectories and line charts to take advantage of both types of visualization methods [9], [41] [42].

If the time-series data has a multi-dimensional nature, it becomes difficult to simultaneously visualize data attributes and their temporal characteristics. Projection methods [43] are often employed to mitigate that difficulty [37]. As for the visualization proposed by Patashnik *et al.* [44], the characteristics of time-series data (such as burst) are emphasized in preprocessing and the result of dimensionality reduction for each time point is superimposed in a single static view. A trajectory is used to visualize a subset of the data that have large changes in the view. It is easy to apply the concept of semantic interaction to such a single view. However, an analysis focusing on a specific time range becomes difficult.

It is also possible to apply projections at each time point and visualize them separately. However, the cost of parameter adjustment is increased inevitably, and it is difficult to confirm the validity of the adjusted parameter for the global trend of a dataset. Therefore, when applying semantic interaction to multi-dimensional time-series data, it is important to select an appropriate visual encoding that can support the understanding of both the overview of the entire time range and detailed characteristics at a specific time point or range [45].

D. FORMULATION OF EVALUATION METRICS

As noted in Sec. I, evaluation metrics are defined as an interpretable and attribute-based representation of user-defined criteria: they are used to quantify the degree of compatibility of data to a human's subjective criteria. Evaluation metrics are utilized in various domains with different forms, such as thresholds for determining positive and negative examples in classification tasks, criteria for determining rankings in multi-criteria decision problems [46], metrics for decision making in epidemiology [47], and sabermetrics [3]. Other examples of evaluation metrics are the KPI [1] and performance metrics for employees in the business field [48].

As one of the guidelines for formulating such metrics, SMART criteria ("specific," "measurable," "achievable," "relevant," and "time-bounded") have been advocated [49]. To formulate metrics that follow such guidelines, metrics should be based on data and formulated through continuous improvement [49]. For data-driven metrics formulation, it is important to assess the metrics' quality and compare alternative metrics [50]. However, when the number of alternative metrics and data attributes is large, it is difficult for domain experts to comprehensively compare and analyze those metrics.

Regarding VA interfaces for metrics formulation, as mentioned in Sec. I, formulations of projection functions [10] and rankings criteria [11] for multi-dimensional data have been proposed. As an example of visually supporting KPI analysis, an interface that combines a node-link diagram and matrix-based visualization — to represent complex relationships among multiple KPIs and their constituent attributes in manufacturing production systems — was proposed by Brundage *et al.* [51]. In that proposal, however, they do not mention how to formulate new metrics based on the analysis results. As for explicitly dealing with the metrics formulation, a formulation process consisting of the following three tasks was proposed by Chen *et al.* [2]: first, analysts collect data with labeled information, which have a large number of attributes that may be used as metrics, and apply attribute-selection algorithms so that they can identify the attributes that strongly contribute to the prediction of labels. Second, on the basis of the analysts' domain knowledge, they qualitatively confirm the accuracy of the label prediction when the selected attributes are used as evaluation metrics. They stop this step when they find a few attributes consisting of evaluation metrics. Third, they formulate candidate metrics by combining the attributes selected in the previous steps and determine the final metrics on the basis of their effectiveness regarding real-world data. However, to the authors' knowledge, a VA interface for metrics formulation that considers the temporal characteristics of target data has not been studied.

III. PROPOSED FRAMEWORK

A. REQUIREMENT

To formulate evaluation metrics for time-series data, a conceptual framework that extends semantic interaction to cover the temporal characteristics of target data is proposed in this section. The framework is assumed to satisfy the following three requirements.

1) CONSIDERATION OF TEMPORAL CHARACTERISTICS (R1)

To formulate evaluation metrics for multi-dimensional time-series data, it is necessary to consider the temporal characteristics of the data. For example, when such data is being monitored, a fixed threshold for detecting anomalies is not enough for monitoring over a long period; that is, different thresholds should be used for a time range having different temporal characteristics. As another example, to clarify the temporal characteristics of multi-dimensional time-series data for such tasks as prediction, smoothing, and normalization, are often applied to remove noisy fluctuation [36]. However, to apply those techniques without losing important temporal characteristics, analysts have to discriminate noise from true temporal variation of data at each time point. If analysts attempt that discrimination by using an existing interface based on semantic interaction, they need to analyze data independently at each time point. As a result, the interaction cost for adjusting and validating the results of the analysis will increase. On the contrary, applying uniform parameters to the entire time range is not reasonable when the temporal trend of data changes in the middle of the time range. For example, in the case of sabermetrics, players tend to have extremely high or low statistics early in the baseball season due to the relatively small number of plate appearances and innings pitched. Therefore, using the same metrics throughout the season is not practical to make a valid evaluation of player's performances [52], [53]. We think that these problems can be solved if users can adjust the parameters for an arbitrary time range. Therefore, in accordance with the analyst's intention, the framework needs to allow the analyst to flexibly change the time range in which the parameters can be adjusted.

2) INCREMENTAL FORMULATION OF METRICS (R2)

To effectively formulate metrics using the VA interface, it must be possible to adjust the parameters incrementally while checking the validity of the VA results by referring to the user's domain knowledge. These processes can be supported by the concept of semantic interaction described in Sec. II-A. As for existing VA interfaces utilizing semantic interaction [22], [31], the parameters of the models used for data visualization are adjusted in a single view without switching views; accordingly, the analyst can incrementally adjust the parameters while checking the feedback from a computer reflected in the visualization. To apply such interaction, multi-dimensional time-series data also needs to be visualized in a single view. However, users have to not only adjust the parameters at each time point but also formulate metrics with reference to the adjusted parameters in the entire time range (or a specific time range) while preventing overfitting to a specific time point. If existing visualization methods, as described in Sec. II-C, are applied, it is difficult to visualize the temporal trends over the entire time range and data distribution at each time point in a single view. When switching playback points by animation, the context of the semantic interactions at other time points is lost because visualized objects and their associated parameters are updated. Although static visualization such as trajectory representation can visualize the temporal change of an instance in a single view, it suffers from the problem of visual clutter [9]. Small multiples can visualize multiple time points at the same time. However, when the number of time points increases, users need to confirm each view side-by-side, and that confirmation increases the interaction cost. Therefore, to support metrics formulation incrementally, it is necessary to visualize multi-dimensional time-series data in a single view in a form that preserves the context of data and suppresses visual clutter so that the analysts can give feedback on that view only.

3) ANALYTICAL PROBLEMS SPECIFIC TO TIME-SERIES DATA (R3)

The following three problems should be solved when applying semantic interaction to time-series data [9]: (1) the collision of interactions on the temporal aspect, which occurs between updating the placement of visualized objects by animation and the displacement of visualized objects with the user's direct manipulation; (2) the collision of interactions between the temporal and spatial aspects, which occurs between two aspects (temporal or spatial) that can be interpreted from the same interaction (e.g., selection operation); and (3) unavailability of a uniquely effective visualization method for understanding the temporal characteristics of multi-dimensional time-series data and suppressing visual clutter [13]. To effectively manipulate visualized objects on the basis of semantic interaction, it is necessary to combine several visualization methods according to the purpose of the analysis and the characteristics of target data.

B. OVERVIEW OF PROPOSED FRAMEWORK

The proposed framework focuses on multi-dimensional time-series data with a fixed number of attributes as its target data. As mentioned in Sec. II-C, the data is represented as $D = \{d_{tnm}\}$, where t, n, and m are respectively indices of time points, instances (o_n), and attributes. Length of the time series is T, number of instances is N, and each instance has M attributes. Similar to the existing studies mentioned in Sec. II-D [2], it is assumed hereafter that the process of metrics formulation consists of the following three steps:

1) Preprocessing: data collection, attribute selection, and normalization
2) Understanding the characteristics of processed attributes through visual analysis on the basis of the results of step (1)
3) Definition of evaluation metrics on the basis of the results of step (2) and the analyst's domain knowledge

The proposed framework covers step (2) and makes it possible to create a starting point for formulating interpretable and data-driven evaluation metrics in step (3). Regarding the target multi-dimensional time-series data, monthly or yearly statistical data is focused on hereafter. It is supposed that the length of time series T should be short enough to grasp the characteristics of data at each t and the entire time range. That is, directly analyzing raw data such as sensor data and purchase history for customers of e-commerce service is out of the scope of the framework.

After being subjected to dimensionality reduction, the target data are visualized as a 2D scatter plot at each time point (t), and temporal characteristics are expressed by animation. In the same manner as existing VA interfaces applying semantic interaction, visualized objects on the scatter plot can be directly manipulated with interactions such as drag and drop [54]. The visualized objects used in the prototype interface are described in Sec. III-E. Each axis of the scatter plot is represented as a linear combination of instances' attributes, which is updated by direct manipulation of the visualized objects by the user. As will be shown in Sec. V, different axes can be used to reflect the different viewpoints of the analysts. While manipulating the data distribution on the scatter plot, users can understand the distribution of the target data and its temporal characteristics. When they obtain the data distribution that reflects their subjective criteria, the linear combination of attributes corresponding to the axis is used as a metric or a starting point for formulating metrics. Although a linear combination of attributes cannot represent a nonlinear correlation between attributes, it has been used as a means of dimensionality reduction for visualizing multi-dimensional data because of its interpretability [10] and short calculation time [25]. Regardless, in this study, we consider widely used metrics based on a linear combination of attributes, such as sabermetrics [3], well-being metrics [55], and objective functions for learning to rank [56], as sufficiently practical for our proposed framework. This is because, as noted in Sec. I, as for evaluation metrics, interpretability is more important than accuracy. Furthermore, as shown in Sec. V, in addition to linear metrics, complex, nonlinear metrics can be formulated on the basis of the obtained axis.

C. ANALYTICAL PARAMETERS

For vector $\boldsymbol{d}_{tn} \in \mathbb{R}^M$, which corresponds to o_n at time point t, its 2D coordinates $\boldsymbol{p}_{tn} = (p_{tn}^{(\mathrm{X})}, p_{tn}^{(\mathrm{Y})})$ are defined by local parameter $\boldsymbol{\alpha}$ and global parameter $\boldsymbol{\omega}$ according to Eq. (1) as follows:

- $\boldsymbol{\alpha} = (\boldsymbol{\alpha}^{(\mathrm{X})}, \boldsymbol{\alpha}^{(\mathrm{Y})}) \in \mathbb{R}^{2\times T\times N}$: displacement of each instance by user's manipulation.
- $\boldsymbol{\omega} = (\boldsymbol{\omega}^{(\mathrm{X})}, \boldsymbol{\omega}^{(\mathrm{Y})}) \in \mathbb{R}^{2\times T\times M}$: coefficient for attributes of instances, which define the contribution of each attribute to the axes on the scatter plot.

$$\boldsymbol{p}_{tn} = \sum_{m=1}^{M} d_{tnm}\boldsymbol{\omega}_{tm} + \boldsymbol{\alpha}_{tn}. \tag{1}$$

Used as building blocks for the metrics, parameter $\boldsymbol{\omega}$ represents a projection from M-dimensional space to 2D space. Its value is common for all instances, so it is referred to as a "global parameter." Since it can be specified for different time points, it includes t as an index. On the contrary, local parameter $\boldsymbol{\alpha}$ can have different values for different instances at different time points. It is supposed to temporally reflect the user's manipulation on each instance on a scatter plot. The values of $\boldsymbol{\alpha}$ can be absorbed into $\boldsymbol{\omega}$ as explained in Sec. III-D2. To emphasize the difference between trends of different time points, the dimensionality reduction can be applied to calculate the initial value of $\boldsymbol{\omega}_t$ independently at each time point (t). It is also possible to use $\boldsymbol{\omega}_t$ obtained at a certain time point (t) as the initial values of $\boldsymbol{\omega}_{t'}$ at different time points ($t' \neq t$).

Utilizing the advantages of different visualization methods, the proposed framework assigns different time ranges to different types of visualized objects in a manner that satisfies requirement R3 (3) described in Sec. III-A. It also enables users to flexibly change the parameters in different temporal and/or spatial ranges by manipulating appropriate visualized objects according to their intentions (Sec. III-D). Therefore, we think the proposed framework can support metrics formulation considering temporal characteristics (R1). Analysts can give their feedback on the scatter plot by, for example, moving visualized objects alongside a specific axis, which is theoretically interpreted as parameter changes.

D. DIRECT MANIPULATION

Parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\omega}$ can be adjusted to reflect the user's intentions such as "move a specific object and similar objects in the opposite direction to dissimilar objects" or "make outliers more prominent on each axis" through the interaction on the scatter-plot view. Such direct manipulation of visualized objects is classified as three types [57]: absolute, orbital, and relative. Absolute alignment represents a global change to the projection, while orbital and relative alignments represent local changes. Among these strategies, the absolute and relative ones are adopted for the proposed framework; the orbital one is not used because it is covered by the relative strategy. Both strategies correspond to specific types of direct manipulation of visualized objects on the scatter-plot view: users can incrementally adjust interpretable parameters in arbitrary time ranges on the single view (R2 in Sec. III-A).

Regarding interaction with the proposed VA interface for adjusting parameters, the two above-mentioned manipulation methods and manual parameter adjustment are implemented in the following manner. Via the proposed interface, direct manipulation is performed by dragging and dropping a visualized object (described in Sec. III-E) or bar chart (Fig. 2(c)). In the following subsections, $\boldsymbol{\omega}'$ corresponds to $\boldsymbol{\omega}$ before the parameter adjustment.

1) ABSOLUTE MANIPULATION

We suppose that users use absolute manipulation with the intention of adjusting the projection globally to emphasize the target object(s). When a visualized object is dragged and dropped onto one of the axes of the scatter plot at time point t (Fig. 2(a1)), $\boldsymbol{\omega}_t$ is adjusted so that the coordinate of the object on the axis is close to the dropped position (Fig. 2(a2)). During absolute manipulation, the degree of salience for each attribute $\Delta\boldsymbol{d}_t$ is calculated by using Eq. (2). In Eq. (2), O^{S} is a set of instances corresponding to the objects dropped on a projection axis and O^{NS} denotes the set of remaining instances. $\Delta\boldsymbol{d}_t$ obtained by using Eq. (2) is used to update $\boldsymbol{\omega}_t^{(i)}$ from $\boldsymbol{\omega}_t'^{(i)}$ to $\hat{\boldsymbol{\omega}}_t^{(i)}$ by using Eq. (3), and the updated $\hat{\boldsymbol{\omega}}_t^{(i)}$ is normalized by using Eq. (4). In Eq. (3), $c_t^{(i)} \in [0, 1]$ represents the degree of emphasis of all objects with respect to the projected axis, and it is calculated by using Eq. (5) with $\mathrm{P}os(i)$, which represents the normalized dropped position on axis $i \in \{\mathrm{X}, \mathrm{Y}\}$. In Eq. (5), $\mathrm{m}ax(\boldsymbol{p}_t^{(i)})/\mathrm{m}in(\boldsymbol{p}_t^{(i)})$ corresponds to the maximum and minimum values of coordinate $\boldsymbol{p}_t^{(i)}$ over all instances. If multiple objects are dropped, their centroids are treated as $\mathrm{P}os(i)$.

$$\Delta\boldsymbol{d}_t = \frac{1}{|O^{\mathrm{S}}|}\sum_{o_n\in O^{\mathrm{S}}}\boldsymbol{d}_{tn} - \frac{1}{|O^{\mathrm{NS}}|}\sum_{o_n\in O^{\mathrm{NS}}}\boldsymbol{d}_{tn}, \tag{2}$$

$$\hat{\boldsymbol{\omega}}_t^{(i)} = \boldsymbol{\omega}_t'^{(i)} + c_t^{(i)}\Delta\boldsymbol{d}_t, \tag{3}$$

$$\boldsymbol{\omega}_t^{(i)} = \frac{1}{\|\hat{\boldsymbol{\omega}}_t^{(i)}\|_2}\hat{\boldsymbol{\omega}}_t^{(i)}, \tag{4}$$

$$c_t^{(i)} = 2\left(\frac{\mathrm{P}os(i) - \mathrm{m}in(\boldsymbol{p}_t^{(i)})}{\mathrm{m}ax(\boldsymbol{p}_t^{(i)}) - \mathrm{m}in(\boldsymbol{p}_t^{(i)})} - \frac{1}{2}\right). \tag{5}$$

2) RELATIVE MANIPULATION

We suppose that users use relative manipulation with the intention of adjusting local relationships between objects. When a visualized object placed at $p_{tn}'^{(i)}$ is dragged and dropped onto arbitrary position $p_{tn}^{(i)}$ on the scatter plot at time point t (Fig. 2(b1)), $\boldsymbol{\alpha}_{tn}$ is modified by using Eq. (6) so that the object is placed at the dropped position (Fig. 2(b2)). $\boldsymbol{\alpha}_t$ can be absorbed by $\boldsymbol{\omega}_t$ whenever required by solving the optimization problem by using a loss function $L(\boldsymbol{\omega}_t^{(i)})$ (Eq. (7)) for each axis ($i \in \{\mathrm{X}, \mathrm{Y}\}$) independently. As for the proposed interface, [$\boldsymbol{\alpha}$ to $\boldsymbol{\omega}$] button is provided in the scatter-plot view (Fig. 1(a)). In Eq. (7), λ is a hyperparameter that controls the

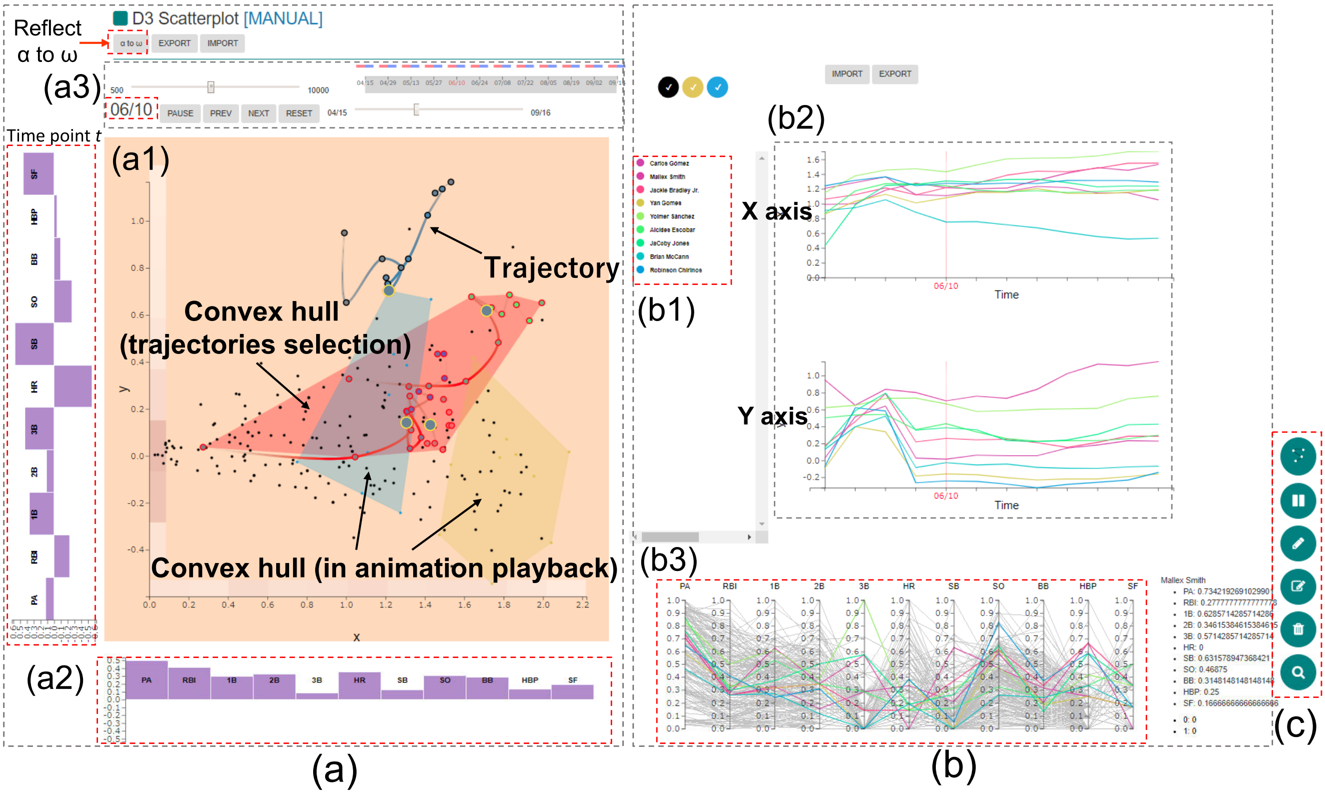


FIGURE 1. Screenshot of prototype interface: (a) scatter-plot view consists of (a1) scatter plot, (a2) bar chart that represents parameter ω, and (a3) animation control UI; (b) detailed view consists of (b1) labels of selected visualized objects, (b2) line charts that represent a temporal change of positions on a scatter plot, and (b3) parallel coordinates of attribute values at the playback point of animation; and (c) analytical functions (e.g., changing manipulation target).

effect of a regularization term. Hereafter, λ is set to 0.5. After solving the optimization problem, $\boldsymbol{\omega}_t$ is used to determine the position of visualized objects, and all $\boldsymbol{\alpha}_{tn}$ values are reset to zero.

$$\alpha_{tn}^{(i)} = p_{tn}^{(i)} - p_{tn}^{\prime(i)}. \tag{6}$$

$$L(\boldsymbol{\omega}_t^{(i)}) = \sqrt{\sum_{n=1}^{N}\left(\sum_{m=1}^{M} d_{tnm}\omega_{tm}^{(i)} - p_{tn}^{\prime(i)}\right)^2} + \lambda\|\boldsymbol{\omega}_t^{\prime(i)} - \boldsymbol{\omega}_t^{(i)}\|_2. \tag{7}$$

3) MANUAL PARAMETER ADJUSTMENT

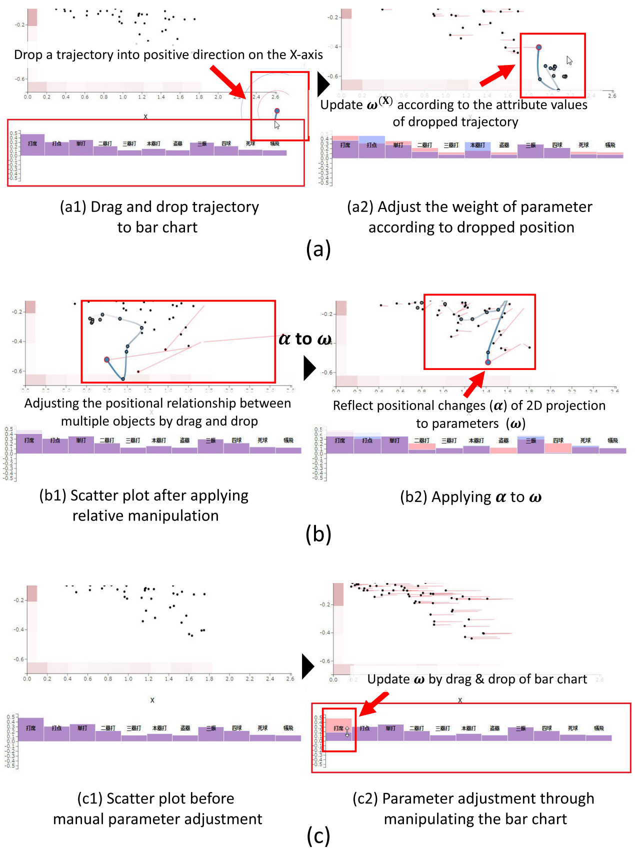
VA based on semantic interaction could lead to confirmation bias or bias due to over-focus on specific objects [4]. To recognize these biases, analysts have to confirm which parameters have been adjusted significantly when the object(s) are moved. To support such a confirmation, the proposed interface visualizes the amount of parameter adjustment at each time point as a bar chart. The values of $\boldsymbol{\omega}$ can be adjusted not only by directly manipulating visualized objects but also by manually manipulating the bar chart (Fig. 2(c)). The combination of semantic interaction and manual control with a UI is expected to be useful for fine-tuning the results of direct manipulation of visualized objects. Also, we suppose that confirming the result of manually changing the value of $\boldsymbol{\omega}_{tm}$ provides a hint to the user about which direction to move the object. When the height of the bar chart on axis i, which corresponds to the degree of emphasis of each attribute d_{tm}, is changed manually, the value of $\omega_{tm}^{(i)}$ is updated on the basis of the modified height, which is immediately reflected in the scatter-plot view.

E. VISUALIZED OBJECTS

On the scatter plot, three types of visualized objects, i.e., nodes, trajectories, and convex hulls, can be dragged and dropped directly as explained in Sec. III-D. The spatial and temporal ranges of the adjusted parameters differ according to the type of visualized objects to be manipulated. As a result, the parameters can be flexibly adjusted in arbitrary spatial and temporal ranges (R1 in Sec. III-A).

1) NODE

A node is located at position $\boldsymbol{p}_{tn}$ on the scatter plot and corresponds to $\boldsymbol{d}_{tn}$. When a node is directly manipulated, only the parameters of the current time point corresponding to the playback point of animation are adjusted. Direct manipulation of nodes is useful for data that contain abnormal values only at specific time points. For such data, the effect of abnormal values on evaluation metrics can be removed. If a drastic change of an object's position that is contrary to the user's mental model is confirmed through playing the


Drop a trajectory into positive direction on the X-axis
Update $\omega^{(X)}$ according to the attribute values of dropped trajectory
(a)
α to ω
Adjusting the positional relationship between multiple objects by drag and drop
Reflect positional changes (α) of 2D projection to parameters (ω)
(b)
Update ω by drag & drop of bar chart
(c)

(a1) Drag and drop trajectory to bar chart

(a2) Adjust the weight of parameter according to dropped position

(b1) Scatter plot after applying relative manipulation

(b2) Applying α to ω

(c1) Scatter plot before manual parameter adjustment

(c2) Parameter adjustment through manipulating the bar chart

FIGURE 2. Example of parameter adjustment via the prototype interface: (a) absolute manipulation, (b) relative manipulation, and (c) manual parameter adjustment.

animation, the user can adjust the parameters for that time point only by dragging and dropping the node.

2) TRAJECTORY

A trajectory shows a change in the position of instance o_n in the playback range of the animation: a line corresponding to a path $(\boldsymbol{p}_{1n}, \boldsymbol{p}_{2n}, \ldots, \boldsymbol{p}_{Tn})$ is drawn with a curve interpolation to represent the change between time points smoothly. The brightness of a trajectory is drawn so that $\boldsymbol{p}_{tn}$ corresponding to the current time point is the darkest. Because a trajectory can represent temporal trends of an instance in a single static view, it is useful to preserve the user's mental model on the consistency of movement of objects [58]. The trajectory is displayed when the mouse cursor is moved over a node. It is also used to change the playback point of animation on the scatter-plot view in the same way as detailed in [59]. When the mouse cursor is moved over a node in the displayed trajectory, the distribution of all nodes at that time point is previewed by semi-transparent superimposition. This preview is expected to be useful because it allows the user to compare the distribution of nodes between different time points.

Differing from direct manipulation of a node, that of a trajectory adjusts parameters $\boldsymbol{\alpha}_{tn}$ and $\boldsymbol{\omega}_{tm}$ for all t in the time range corresponding to the playback range of the animation ($t = 1, 2, \ldots, T$). We suppose that a trajectory is directly manipulated to emphasize an instance for a specific time range. By using a trajectory as the manipulation target, the collision of interactions on the temporal axis (R3 (1)) can be avoided because the position and shape of a trajectory do not change with the animation, even when the trajectory is visualized with a combination of animations [9].

3) CONVEX HULL

A convex hull is a visualized object corresponding to a set of visualized objects on a 2D space. The relationship between multiple instances can be grasped effectively from the shape of the convex hull [60]. It is also used to visualize temporal trends in time-series data on 2D scatter plot [61]. The proposed interface draws convex hulls semi-transparently: users can judge the similarity between multiple clusters by their overlap.

When multiple trajectories are selected, a convex hull of those trajectories (CT) is drawn (Fig. 3(b)). A convex hull is also created by drawing a curve that encloses nodes and trajectories in the same way as lasso selection [62]. The shape of convex hulls can be switched between animation playback mode and pause mode: when animation is played, a convex hull containing the nodes of selected instances at each time point (CN) is drawn (Fig. 3(a)). A convex hull containing the trajectories of selected instances (CT) can also be drawn in pause mode without drawing trajectories (Fig. 3(c)).

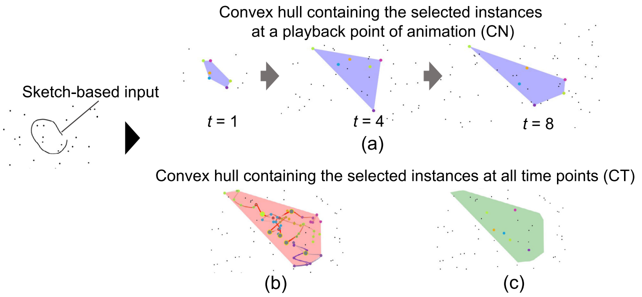

Convex hull containing the selected instances at a playback point of animation (CN)
Sketch-based input
t = 1
t = 4
t = 8
(a)
Convex hull containing the selected instances at all time points (CT)
(b)
(c)

FIGURE 3. Example of a convex hull drawn in (a) animation playback (CN) and (b) and (c) pause mode (CT with or without trajectories).

The combination of dynamic visualization (animation) and static visualization (trajectories and convex hulls) makes it possible to balance the trade-offs between different visualization methods (R3 (3)). A convex hull CT drawn by selecting multiple trajectories (Fig. 3 (b)) is the same as that drawn in pause mode (Fig. 3(c)), from which the temporal trend of multiple objects can be grasped spatially. As a result, the occlusion problem of multiple trajectories can be mitigated. On the contrary, the spatial relationship between objects at specific time point t can be grasped using a CN. Therefore, by utilizing a convex hull, it is possible to understand both the local (t) and global ($t = 1, 2, \ldots, T$) trends described in Sec. II-C.

Direct manipulation of convex hulls makes it possible to adjust the parameters relating to all objects inside the convex hull. For example, O^S in Eq. (6) is selected from the convex hull in the case of absolute manipulation. The target time range of parameter adjustment differs according to the method used for drawing the convex hull, as shown in Fig. 3. CN can be manipulated to emphasize objects only at specific time point t. On the contrary, CT can be manipulated to emphasize objects within the playback range of the animation ($t = 1, 2, \ldots, T$). Different methods for drawing a convex hull make it possible to switch target temporal and/or spatial range in accordance with the user's intention (R3 (2)).

IV. PROTOTYPE INTERFACE

A. IMPLEMENTATION

A prototype interface based on the proposed framework was implemented. The interface includes two views: a scatter-plot view (Fig. 1(a)) and a detailed view (Fig. 1(b)). It consists of a front-end for visualization, a controller for transferring data and parameters, and an API for parameter adjustment. Ruby on Rails 5 is used for the controller, and D3.js v5[1] and jQuery 3.3[2] are used for the front-end. The API was implemented by using Python 3 with Flask.[3] The prototype interface uses principal component analysis (PCA) [63] for dimensionality reduction and determining initial $\boldsymbol{\omega}_t$ for each $t = 1, 2, \ldots, T$, independently. PCA was selected because of its generality [54], interpretability of results, and preservation of global placement [25]. PCA has comparable performance to other methods for dimensionality reduction for multi-dimensional time-series data, such as t-SNE (t-distributed stochastic neighbor embedding) and UMAP (uniform manifold approximation and projection), in terms of distance preservation and temporal consistency [43]. When PCA is applied to each time point independently, the consistency of $\boldsymbol{\omega}_t$ between different time points t is not always preserved. To improve the consistency, the proposed interface performs sign inversion for $\boldsymbol{\omega}_t$ automatically. It loads PCA-applied data in JSON format,[4] and scipy.optimize[5] was used to solve the optimization problem shown in Eq. (7). The value of $\boldsymbol{\omega}$ can be imported or exported in JSON format via the button at the top of the scatter-plot view.

B. SCATTER-PLOT VIEW

The scatter-plot view visualizes the results of dimensionality reduction at an arbitrary time point specified by the user (Fig. 1(a)). Target time point t of the scatter plot can be switched by animation playback and controlled by the UI on the interface (Fig. 1(a3)). This view is used to gain insight into temporal trends of data and adjust $\boldsymbol{\alpha}$ and $\boldsymbol{\omega}$ through direct manipulation of visualized objects (Sec. III-E). A convex hull is drawn with d3-polygon,[6] a library that computes the region of the convex hull using Andrew's monotone chain algorithm. Zooming and panning on objects of interest are possible on the scatter plot. The bar chart visualizing $\boldsymbol{\omega}$ is located along the axis of the scatter plot (Fig. 1(a2)), and the parameters can be adjusted manually (Sec. III-D3).

The interface enables incremental parameter adjustment (R2 in Sec. III-A) in several ways. Firstly, the result of parameter adjustment is immediately reflected on the scatter-plot and detailed views (Sec. IV-C). Secondly, to suppress visual clutter, the display of visualized objects can be turned off on-demand. Thirdly, to prompt the user for confirmation of direct manipulation, manipulated objects are highlighted in border color, and lines are drawn to connect placements of an object before and after it was directly manipulated [23].

To take advantage of both the static and dynamic time-series data-visualization methods, as in [9], the prototype interface maps playback and pause of the animation to temporal and spatial interpretations of the interaction. Playback mode of the animation is used for two purposes: (1) to understand the temporal characteristics of the time series data and (2) to verify the effectiveness of parameter adjustments and the validity of the adjustments across the entire time range. On the contrary, in pause mode, the user forms a hypothesis necessary for metrics formulation through direct manipulation of visualized objects after obtaining the visual insights in playback mode. These interaction modes can resolve the collision of interactions between the temporal and spatial aspects by providing different results for the user's intention (R3 (2) in Sec. III-A). For example, as shown in Sec. III-E2, the shape of the convex hull can represent different information in different modes: visualizing a set of nodes in playback mode and/or a set of trajectories in pause mode. If absolute manipulation (Sec. III-D1) is applied after $\boldsymbol{\alpha}$ is modified, a dialog box is displayed to confirm whether the change of $\boldsymbol{\alpha}$ is absorbed by $\boldsymbol{\omega}$.

C. DETAILED VIEW

When a single object or multiple visualized objects on the scatter plot are selected, detailed information about the object(s) is displayed in this view. This view consists of a list of objects selected from the scatter-plot view (Fig. 1(b1)), a line chart of changes in X and Y coordinates of o_n ($\boldsymbol{p}_{tn}$) (Fig. 1(b2)), and parallel coordinates representing all $d_{tn'm}$ for $n' = 1, 2, \ldots, N, m = 1, 2, \ldots, M$ (Fig. 1(b3)). Raw attribute values d_{tnm} of specific instances (o_n) at the same time point t as the scatter-plot view are also shown (right side of Fig. 1(b3)). The list of instances selected in the scatter-plot view is also presented in this view (Fig. 1(b1)). Each instance has a unique color in the detailed view. This list serves as a legend for each component in the detail view. Objects to be manipulated in the scatter-plot view can also be selected from the detailed view, which is complementary to the scatter-plot view: users can explore the temporal change of instances selected in the scatter-plot view by line charts (Fig. 1(b2)) and the relationship between those attribute values and $\boldsymbol{\omega}$ by parallel coordinates (Fig. 1(b3)). The detailed view is used for generating hypotheses by verifying the visual insights obtained from the scatter-plot view.

[1] https://d3js.org/
[2] https://jquery.com/
[3] http://flask.pocoo.org/
[4] https://www.json.org/json-en.html
[5] https://docs.scipy.org/doc/scipy/reference/optimize.html
[6] https://github.com/d3/d3-polygon

TABLE 1. Correspondence between UI and function on navigation buttons.

Function	Icon	Cursor
Switching playback/pause	⁘	-
Switching target object	▶	-
Labeling visualized object(s)	✎	✎
Searching for objects	🔍	🔍

D. ANALYTICAL FUNCTIONS FOR EXPLORING DATA

The prototype interface has several analytical functions that allow users to explore data. The functions can be invoked via the navigation buttons on the right side of the screen (Fig. 1(c)). The interface implicitly informs the user of the currently activated function by changing the shape of the mouse cursor as shown in Table 1. The four analytical functions are listed below.[7]

- Switching playback and pause of animation (F1): by pressing the button at the top of Fig. 1(c), the animation is switched between playback and pause. The icon on each button corresponds to the currently activated mode of animation (playback or pause). To implicitly indicate the current mode, the background color of the scatter plot is orange for playback and white for pause.
- Switching target object of direct manipulation (F2): by clicking the second button of Fig. 1(c), icons corresponding to three objects (node, trajectory, and convex hull) are shown on the left side of the navigation buttons shown in Fig. 1(c). When one of the buttons is clicked, the manipulation target is switched to the corresponding type of visualized object.
- Labeling visualized object(s) (F3): a set of visualized objects can be grouped explicitly. When a certain button is clicked, a color palette used for labels is displayed. When a color is selected by clicking on the color palette, the color of the mouse cursor changes to the selected color (Fig. 4(a)). When a set of objects is selected after selecting a color, a convex hull is drawn in the selected color, and a label is assigned to the object (Fig. 4(b)). On the basis of the label information, visualized objects can be shown or hidden. The label information can be imported and/or exported in JSON format.
- Searching for objects by metadata (F4): when metadata, such as the name of an instance, is specified, corresponding visualized objects are searched for by partial match. Objects can be found and selected from the scatter-plot view or the detailed view.

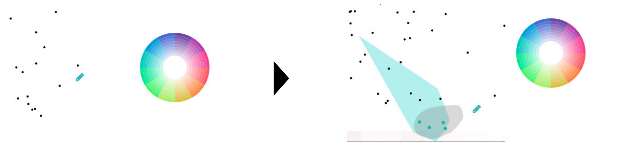

(a) Selecting color from color palette (b) Labeling objects by using selected color

FIGURE 4. Example of labeling functions in the prototype interface.

V. APPLICATION EXAMPLE

To illustrate the effectiveness of the proposed interface, we used it to formulate metrics for statistical data of a baseball game. In this analysis, one of the authors played the role of an analyst. The task was to analyze a dataset about batters in the American League of Major League Baseball (MLB) [64] during the 2018 season. The number of instances (batters) is 198, each of which has eleven attributes, such as the number of home runs and stolen bases. The result of PCA at each time point t is used as the initial value of $\boldsymbol{\omega}_t$. The attributes were sampled every two weeks; thus, the dataset consists of twelve time points. In summary, $N = 198$, $M = 11$, and $T = 12$. Using the axes (a linear combination of attributes) as the output of the analysis, the analyst tried to formulate a metric for simultaneously evaluating the characteristics of the batters and their performance during the season.

The analysis was started by exploring the overall trend of the dataset while playing the animation (Sec. IV-B). Next, batters of a familiar team were searched for by using the object-search function (F4 in Sec. IV-D). Based on his prior knowledge about baseball, the analyst tried to understand the relationship between the batter's properties (shown in the detailed view) and the coordinates on the scatter-plot view. At the same time, the analyst was also interested in the temporal change of these relationships.

After the above-described exploration process, it was noticed that $\boldsymbol{\omega}^{(X)}$ and $\boldsymbol{\omega}^{(Y)}$ roughly correspond to the performance and characteristics (such as contact or power hitters) of the batters, respectively (Fig. 5(a)). Then, he decided to formulate an evaluation metric for classifying batters as contact or power hitters. The obtained hypothesis was used to understand the temporal characteristics in detail by grouping the batters. For example, the object-search function (F4) was used to select the trajectories of the batters considered to be active throughout the season, such as winners of the Silver Slugger Award, and their temporal and spatial tendencies were then analyzed on the scatter plot. The tendencies of the batters corresponding to contact hitters and power hitters were confirmed from the shape of the convex hull created by using the labeling function (F3). Next, the change in the shape of the convex hull (CN in Sec. III-E3) was observed by using animation. It was found that the batters who were active throughout the season were located at the top of the X-axis in the entire time range. It was also found that most of the batters moved to the top of the X-axis as the season progressed. On the contrary, the position of those who lost opportunities to play moved in the opposite direction. However, the animation shows that the temporal trends along each axis were

[7] Descriptions of the functions shown in Fig. 1 that are used in the example and experiments.

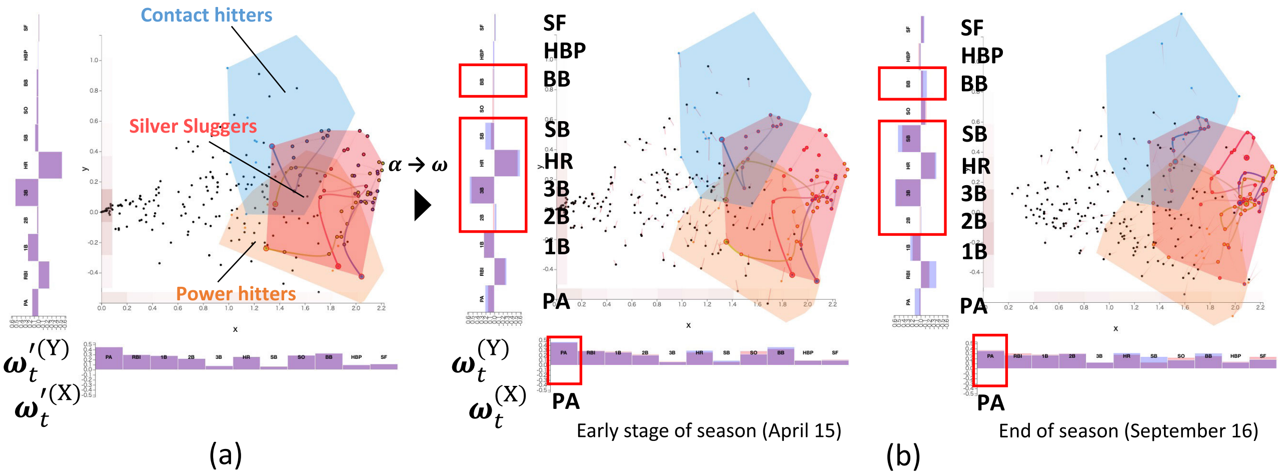


FIGURE 5. Application example of the prototype interface to statistical data of a baseball game during the 2018 season: (a) initial visualization (at April 15) and (b) visualization after parameter adjustment (at April 15 and September 16).

not always apparent through the season. As for the Y-axis, although the position of nodes significantly changes early in the season, no major temporal changes in those positions were found in the second half of the season. As for the distributions of nodes, the impact of attribute m-th on the placement of objects on the Y-axis, $p_{tn}^{(Y)}$, was analyzed by manipulating the bar chart for each m, that is, fine-tuning of ω_{tm} is performed. The result of the analysis revealed that the large changes in $p_{tn}^{(Y)}$ occurred in the early stage of the season because the weight of three-base hits (3B) and home runs (HR) was too strong to discriminate the performance of batters in the early stage of the season.

After each axis was interpreted as described above, $\boldsymbol{\omega}$ was tentatively adjusted by direct manipulation. Two convex hulls were created by selecting power hitters and contact hitters on the basis of the findings of the above-described exploratory analysis. To further emphasize the characteristics on the X-axis, CT was created for the trajectories of batters who maintained high performance throughout the season, and CT was moved to the far right of the X-axis by absolute manipulation.

After the updated scatter plot was interpreted, scatter plots at different time points were compared, and the comparison showed that the characteristics of batters were not clearly separable. Furthermore, to emphasize the characteristics of batters in the scatter-plot view, $\boldsymbol{\alpha}$ was adjusted by moving CT of contact (power) hitters to the upper (lower) ends of the Y-axis by relative manipulation. After $\boldsymbol{\alpha}$ is absorbed by $\boldsymbol{\omega}$, changes in $\boldsymbol{\omega}$ were observed from the bar chart, and the impact of modification of $\boldsymbol{\omega}$ on object placement was analyzed. By observing the updated $\boldsymbol{\omega}$ while playing the animation (Fig. 5(b)), it was noticed that the weight for plate appearance (PA) on the X-axis was higher at the beginning of the season (April 15) than at the end of the season (September 16). Moreover, on the Y-axis, throughout the season, 3B and stolen bases (SB) had large positive weights, while HR and two-base hits (2B) had negative weights. In addition, the weight of BB decreased toward the end of the season.

The analyst tried to formulate an evaluation metric on the basis of the resulting parameter $\boldsymbol{\omega}$, the knowledge acquired through the analysis, and existing metrics for baseball. Finally, a new metric, B_{type} (Eq. (8)), as shown at the bottom of the next page, was formulated with reference to two existing metrics: total bases (TB), which is defined as $1B + 2B \times 2 + 3B \times 3 + HR \times 4$, and on-base percentage (OBP), which is defined as $(1B + 2B + 3B + HR + BB + HBP) \div (AB + BB + HBP + SF)$. In these metrics, 1B, BB, AB, HBP, and SF denote number of single hits, base on balls, at-bat, hit by pitch, and sacrifice flies, respectively. A positive (negative) value of B_{type} corresponds to contact (power) hitters, and its absolute value represents performance. According to the findings acquired from the X-axis, the denominator of the equation for the first half of the season is the minimum required PA instead of the batter's actual PA; it is thus possible to positively evaluate the batters who constantly participate in a game.

As shown in Appendix A, the validity of B_{type} was evaluated using the dataset about batters in the 2019 season. It was confirmed that batters who were active throughout the year (e.g., winners of the Silver Slugger Award) were consistently positioned at the top/bottom in the rankings by B_{type}. On the contrary, batters recognized as contact (power) hitters from their outstanding values of corresponding batting statistics were also in the top (bottom) of the ranking. Therefore, we think that the formulated B_{type} is a valid and consistent metric for evaluating a target batter's performance and classifying their characteristics.

VI. EXPERIMENT

On the basis of the results of the above-mentioned example, the following two hypotheses are postulated:

- H1: multi-dimensional time-series data can be efficiently explored by combining appropriate types of visualized objects according to the target time range and/or number of instances (R3 in Sec. III-A).

- H2: parameters can be efficiently adjusted in an incremental manner to reflect the intentions and knowledge of analysts in the projection by combining absolute manipulation, relative manipulation, and manual parameter adjustment (R1 and R2).

The effectiveness of the proposed interface was quantitatively verified by the following two user experiments, which were designed considering the above-mentioned hypotheses. Eighteen participants (aged 22 to 32 years old) studying or working in the field of computer science, were asked to take part in both experiments. To mitigate the differences in domain knowledge among the participants and the impact of dataset characteristics on the results, artificial datasets were used as the target data. In both experiments, some analytical functions (F3 and F4 in Sec. IV-D) were disabled because they were not directly related to the tasks.

A. EXPERIMENT 1: DATA EXPLORATION

The participants were asked to search for instances with similar temporal characteristics to a given target instance. This experiment aims to verify H1.

1) PROCEDURE

Two baseline interfaces were prepared: EBL1 (exploration-task baseline), which uses a node and a trajectory as visualized objects, and EBL2, which uses a node and a convex hull. The participants performed the data-exploration task by using the proposed interface (exploration-task proposed interface, EPR), EBL1, and EBL2 in random order.

The participants first did tutorial tasks with the prototype interface, and then they performed three tasksets, each of which was to be answered with different interfaces. For each task in a taskset, the participants were asked to find instances similar to a given target instance (the corresponding node was highlighted in color) and to fill in the IDs of the instances they found on an answer sheet. A taskset consists of four tasks (#1, #2, #3, and #4) with different numbers of instances to be found: 1, 3, 5, and 10, respectively. The tasks were presented to the participants in random order to suppress the order effect.

Three datasets were prepared, each of which consisted of 500 instances with 10 attributes and 10 time points, that is, $N = 500$, $M = 10$, and $T = 10$. A different dataset was used for different tasksets to suppress the effect of the characteristics of a dataset on each task. In each dataset, 20 instances were given temporal characteristics, such as monotonic increase, from which the target instance was selected. To emphasize the change in node distribution, the result of the PCA at $t = 1$ was used as $\boldsymbol{\omega}_t$ for the entire time range. After the experiment, the participants answered a questionnaire to rate the interfaces on a five-point Likert scale and gave their reasons, if necessary.

2) RESULTS

The experimental results were evaluated on the basis of execution time of the task, accuracy of finding similar instances, and the results of the post-experiment questionnaire.

To evaluate the similarity between the instances found by the participants and the target instance, it was necessary to define the distance between instances. Since the participants using the prototype interface found similar instances on the basis of the similarity of coordinates on the 2D space, the similarity in projection space rather than in multi-dimensional data space was evaluated. A similarity ranking to the target instance was generated by sorting the remaining instances in ascending order of Euclidean distance to the target instance. Mean reciprocal rank (MRR) was selected as a metric for evaluating the validity of instances stated to be similar. MRR is calculated by using Eq. (9), where K represents the number of found instances, and $r_i \in \{1, 2, \ldots, N-1\}$ corresponds to the rank of the i-th found instance in the similarity ranking.

$$MRR = \frac{1}{K}\sum_{i=1}^{K}\frac{1}{r_i}. \tag{9}$$

Fig. 6 shows task execution time per combination of task and interface. One-way ANOVA (analysis of variance) was used to test for significance. If a significant difference was found, the significance of the difference between the interfaces was evaluated by using Welch's t-test. As a result, only task #1 (number of instances to be found is 1) shows a significant difference ($p < .05$) by ANOVA. On the contrary, the other tasks (#2, #3, and #4) did not show significant differences, possibly due to the fact that the participants used different strategies for different tasks: some participants finished the task when they found a required number of similar instances, while others selected the most similar instances after they found more similar instances than required.

The task for which significant differences between EPR and EBL1/EBL2 were observed is focused on hereafter. The execution time of EBL1 was longer for task #1 than the other tasks ($p < .05$). In EPR and EBL2, multiple instances can be selected as a convex hull. After that, especially when the number of target instances is small, similar instances can be easily found by comparing the selected node' movement by animation playback in the scatter-plot view (Fig. 1(a1)) or temporal changes of those attribute values by using a line chart in the detailed view (Fig. 1(b2)). On the contrary, since

$$B_{type} = \begin{cases} \dfrac{1B \times 2 + SB \times 2 + 3B \times 2 - (2B + HR \times 3)}{\text{Number of team game} \times 3.1} & \text{(1st half)} \\ \dfrac{1B \times 2 + SB \times 3 + 3B \times 3 - (2B + HR \times 4 + BB)}{PA} & \text{(2nd half)} \end{cases} \tag{8}$$

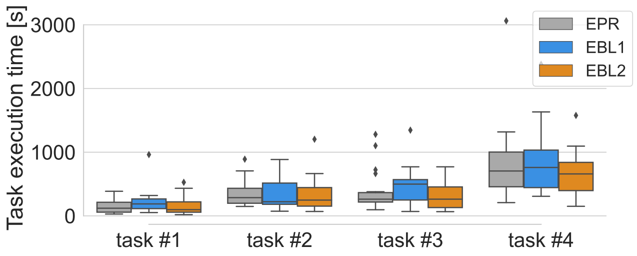


FIGURE 6. Amount of time required to accomplish tasks per interface.

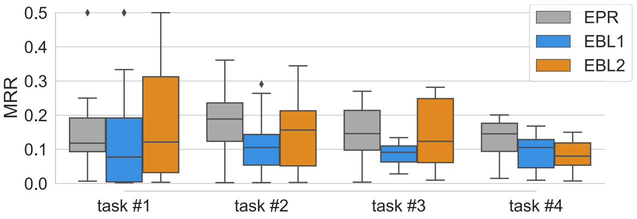


FIGURE 7. MRR (mean reciprocal rank) per interface and task when Euclidean distance is used for the metrics representing similarity between instances.

a convex hull is disabled in EBL1, it is necessary to display the trajectories of candidate instances one by one to compare their similarity. Therefore, execution time of EBL1 is longer than that of EPR and EBL2 for task #1. Similar trends can be confirmed for the other tasks, as shown in Fig. 6.

The mean and standard deviation of MRR for each combination of task and interface are shown in Fig. 7. Significant differences in MRRs for the interfaces were confirmed by one-way ANOVA. The result of Welch's t-test confirmed a significant difference ($p < .05$) for tasks #3 and #4 between EPR and EBL1/EBL2: MRR of EPR was higher than EBL1 for task #3 and both EBL1 and EBL2 for task #4. We think this result is due to the effectiveness of the combination of a trajectory and a convex hull in the case of EPR. When the participants used EBL1, multiple instances could not be selected by a single interaction: since only mouse-over points are compared, they tended to overlook similar instances. This tendency decreased MRR for tasks #3 and #4, in which they had to find more instances than tasks #1 and #2. On the contrary, EBL2 can facilitate the selection of spatially proximate visualized objects by drawing convex hulls. However, when a convex hull enclosing more than 50 nodes is drawn, it is difficult to compare the similarity of all instances in the convex hull due to the overlapping of visualized objects not only in the scatter-plot view but also lines in the parallel coordinates of the detailed view Fig. 1 (b2). Differing from EBL1 and EBL2, EPR supports efficient data exploration by utilizing both a trajectory and a convex hull. As for the questionnaire, 15 out of 18 participants rated the combination of different target visualized objects positively (4 or 5). Many participants commented on the ease of exploratory analysis of time-series data as a reason for that high rating. Two of the participants said that "Checking a trajectory shape after selecting multiple instances (convex hulls) made the data exploration easier." and "A convex hull is useful when observing all the nearby nodes together; a trajectory is useful for finding the target."

These results confirm hypothesis H1 that the combination of different types of visualized objects makes it possible to efficiently explore multi-dimensional time-series data.

B. EXPERIMENT 2: PARAMETER ADJUSTMENT

This experiment aims to verify H2. The participants were asked to perform a task of adjusting parameter $\boldsymbol{\omega}$ to highlight a specified instance on the scatter plot.

1) PROCEDURE

The participants first did tutorial tasks and were then asked to do the following parameter-adjustment tasks. The taskset consists of three tasks of adjusting $\boldsymbol{\omega}$ so the nodes of the target instance (highlighted in color) are positioned at the top of the X- and/or Y- axes for each time point t. Task #1 is for the X-axis, #2 is for the Y-axis, and #3 is for both axes. The tasks were randomly presented to the participants in the experiment to prevent order effects. When the condition for each task was satisfied, a dialog was displayed on the interface, and the participants were asked to finish the task and move to the next one. To prevent an excessive increase in task execution time, a 10-minute time limit was set for each task. As the task is to adjust $\boldsymbol{\omega}_t$ at each time point t, PCA was applied to each time point for obtaining initial values of $\boldsymbol{\omega}$. The following three interfaces were adopted as baseline interfaces. The participants performed three different tasks with the proposed interface (parameter-adjustment-task proposed interface, PPR), PBL1 (parameter-adjustment-task baseline), and one of PBL2 and PBL3. The assignment and the order of interfaces to be used by each participant were based on a round-robin design of experiments.

- PBL1: only manual parameter adjustment
- PBL2: manual parameter adjustment and relative manipulation
- PBL3: manual parameter adjustment and absolute manipulation

A different dataset was used for each taskset. In this experiment, the difficulty of the task is supposed to depend on the number of time points (T). In addition, since the participants do not have to perform a complex data exploration as in Experiment 1, many instances are not needed in this experiment. Therefore, the target dataset consists of 50 instances with 10 attributes and five time points, that is, $N = 50$, $M = 10$, and $T = 5$. This artificial dataset should ensure that the visualized object corresponding to the target instance is placed at the top of the specific axis by adjusting $\boldsymbol{\omega}$. Therefore, target instance o_n^* has a considerably high attribute value d_{tn^*m} for a certain attribute m among all instances in the dataset. To satisfy this condition, for each time point t, d_{tn^*m} is set to be larger than $\frac{1}{N}\sum_{n'} d_{tn'm} + 3SD(\boldsymbol{d}_{tm})$, where $SD(\boldsymbol{d}_{tm})$ denotes the standard deviation of $\boldsymbol{d}_{tm}$ at time point t. After the experiment, the participants answered a questionnaire to rate the interfaces on a 5-point Likert scale and gave their reasons, if necessary.

2) RESULTS

The experimental results were evaluated on the basis of task-achievement rate, amount of parameter adjustment, and results of the post-experiment questionnaire.

Amount of parameter adjustment for each projection axis, $amt(i)$ ($i \in \{X, Y\}$), is calculated by using Eq. (10), where $\boldsymbol{\omega}'$ and $\boldsymbol{\omega}$ correspond to $\boldsymbol{\omega}$ before and after the adjustment task, respectively.

$$amt(i) = \frac{1}{T}\sum_{t=1}^{T} \|\boldsymbol{\omega}_t^{(i)} - \boldsymbol{\omega}_t'^{(i)}\|_1. \tag{10}$$

Figs. 8 and 9 show boxplots of $amt(i)$ for the X- and Y- axes, respectively. As for the axis that is not the target of the parameter adjustment, it is obvious that the amount of parameter adjustment with PBL1 and PBL3 was smaller than that with PPR and PBL2, where only the parameters for the target axis are adjusted. Therefore, the difference in the amounts of parameter adjustment by the interfaces only for the adjusted target axes is focused on hereafter. One-way ANOVA was applied to verify significant differences in the amounts of parameter adjustment per interface. Then, the two-tailed F-test was applied only when a significant difference was confirmed. When a significant difference in variance was confirmed by the F-test, Student's t-test was applied; otherwise, Welch's t-test was applied. As a result of the t-test on PPR, significant differences ($p < .05$) were confirmed only for PBL1 (for task #3) for the Y-axis. The amount of parameter adjustment using PBL1 was less than that using PPR because the task could be accomplished by increasing $\omega_{tm}^{(i)}$, which corresponds to the weight for the attribute that the target instance has a considerably high value, by using the bar chart.

As a tendency common to all tasks, the interfaces tend to be sorted as PBL2 > PPR > PBL3 > PBL1 by the amount of parameter adjustment. In the case of PBL2, in which $\boldsymbol{\alpha}$ is absorbed by $\boldsymbol{\omega}$ through optimization, $\boldsymbol{\omega}$ changed drastically when the number of manipulated instances increases. In the case of PBL3, $\boldsymbol{\omega}$ was adjusted on the basis of the target instance's attribute values. If the visualized object of the target instance was moved to the top of the axis with absolute manipulation, $\boldsymbol{\omega}$ changed drastically only if its attribute values were higher or lower than those of other instances. Therefore, the amount of parameter adjustment in the case of PBL3 was less than in the case of PBL2. In contrast to PBL2, PPR using both relative and absolute manipulations could flexibly reflect the participants' intentions in $\boldsymbol{\omega}$ with a small amount of parameter adjustment.

Fig. 10 shows the task-achievement rate for each interface and task within the time limit. A chi-square test with Fisher's exact probability test was applied for testing for a significant difference in task-achievement rate. Significant differences ($p < .05$) between the baselines (PBL1 or PBL3) and PPR were found in three cases: two cases for PBL1 (tasks #2 and #3) and one case for PBL3 (task #3). In all cases, the task-achievement rate of PPR was higher than that of the baselines. The improvement in the task-achievement rate on the interfaces using relative manipulation (PBL2 and PPR) suggests that participants could adjust parameters efficiently within the time limit. On the contrary, in the case of PBL1, the parameters must be manually adjusted for each time point. In the case of PBL3, it is difficult to fine-tune the placement of objects. As a result, we suppose that the task-achievement rate for those interfaces decreased.

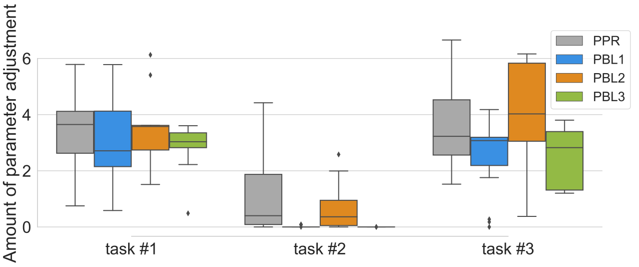


FIGURE 8. Amount of adjusted parameters per interface and task (X-axis).

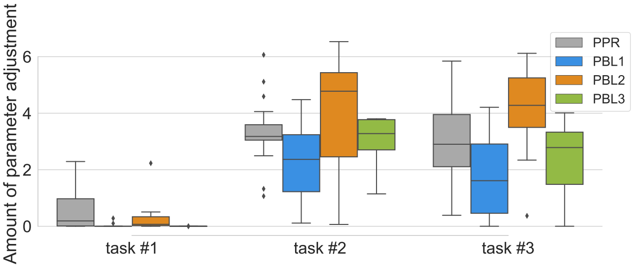


FIGURE 9. Amount of adjusted parameters per interface and task (Y-axis).

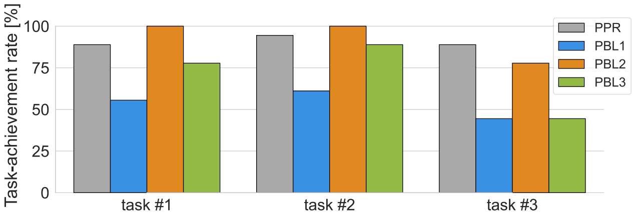


FIGURE 10. Task-achievement rate by interface.

As for the questionnaire, 16 out of 18 participants rated the combination of different parameter-adjustment methods positively (4 or 5). One of the participants commented, "It was easy to roughly adjust the parameters with drag and drop of multiple visualized objects and adjust the details with a bar chart." From the results of the questionnaire and the interaction logs, it was confirmed that some participants adjusted the parameters incrementally with the following strategy:

1) Adjust the global shape of the projection by using absolute manipulations for identifying the trend of each attribute value.
2) Adjust the position of an individual object by using relative manipulations by trial and error for identifying t and m (time point and attribute index) of ω_{tm} that is to be tuned.
3) Based on the knowledge obtained in the above process, fine-tune $\boldsymbol{\omega}_{tm}$ with manual parameter adjustment of the bar chart for each t and m.

As the target data is a synthetic one, participants did not have prior knowledge of the data. As one of the notable analytical behaviors, immediately after starting a task, some participants tried to understand the relationship between attributes (denoted as m here) of instances $\boldsymbol{d}_{tm}$ and $\boldsymbol{\omega}_{tm}$ by observing objects moved after performing manual parameter adjustment (Fig. 2(c)). After that, to find instances having similar attribute values as the target instance, which could be the candidate for applying direct manipulation, they also tried to identify instances that have high or low $\boldsymbol{d}_{tm}$ in parallel coordinates (Fig. 1(b3)).

As for the questionnaire, two participants reported that they could flexibly adjust the parameters by switching the visualized objects to manipulate (F2 in Sec. IV-D): analyzing their interaction logs revealed that they used convex hulls when they wanted to change the positions of multiple visualized objects at once, trajectories when they wanted to emphasize a visualized object that corresponds to specific instance o_n for the entire time range ($t = 1, 2, \ldots, T$), and nodes to emphasize o_n at a specific time point t.

These results confirm hypothesis H2 that the proposed interface makes it possible to flexibly adjust parameters by combining the advantages of absolute manipulation, relative manipulation, and manual parameter adjustment.

VII. LIMITATIONS

A. USABILITY OF FUNCTIONS

Whereas hypotheses H1 and H2 were verified through the above-described experiments, from the results of questionnaires, we found several usability problems regarding the analytical functions. In Experiment 1, two of the participants commented that it is difficult to use the zooming function described in Sec. IV-B, for exploring relationships among individual visualized objects. We think introducing semantic zooming [65] will solve this problem. In Experiment 2, some participants pointed out that they were able to flexibly adjust the parameters by switching different types of visualized objects to manipulate. However, we did not explicitly evaluate the effectiveness of switching visualized objects (F2 in Sec. IV-D) during the parameter adjustment. In future work, it is thus necessary to verify the effectiveness through additional experiments.

When comparing multiple time points, it is necessary to control the animation step-by-step. Therefore, high interaction cost was pointed out by two participants in the questionnaire after Experiment 2. Although it is possible to superimpose node distribution of other time points by a mouseover on the trajectory (explained in Sec. III-E2), it is difficult to compare projections and parameters for more than two time points simultaneously. The validity of the parameters at other time points can be grasped from animation and the shape of convex hulls (Sec. III-E3), but more support for such a validation is needed. We suppose that the introduction of Streamgraph [66] to the detailed view is effective because it allows users to explore the dominant parameters at each time point in a single view.

In the questionnaire given in Experiment 2, three of the participants requested an undo/redo function for parameter adjustment to cancel erroneous manipulations. That function is expected to be effective for data exploration to compare the results of applying several parameters and selecting the appropriate one [67]. Furthermore, the externalization of the interaction history is also useful for sharing the findings through VA. In the experiments, analytical functions F3 and F4 were disabled due to the irrelevance of the task. However, in the questionnaire of Experiment 1, four participants mentioned the necessity of the function for selecting objects with metadata. This function is the same as the search function (F4). We suppose F3 is effective for the classification of multiple sets of instances, as in Sec. V. In future work, these functions also need to be evaluated.

B. SCALABILITY

One of the problems in visualizing a large amount of data using a scatter plot is visual clutter due to the overlap of the objects, which decreases the interpretability of visualization [26]. To solve this problem, decreasing the number of visualized objects by aggregating data instances is promising. Alternatively, it can be effective to sample representative data instances before visualizing them for metrics formulation. As for the bar-chart visualization of multi-dimensional data (Fig. 1(a2)), the cognitive load may increase when the dataset has too many attributes. For scalability regarding number of attributes, attribute selection and reduction can be applied in the preprocessing step of metrics formulation described in Sec. III-B. Although no participant mentioned the response speed of parameter adjustment in the proposed interface, it is necessary to consider the latency when number of objects increases. Applying incremental dimensionality reduction [29] is one of the possible solutions to this problem.

Regarding temporal scalability when the number of time points increases, drawing trajectories and convex hulls on the basis of a temporal range is more effective than reducing time points. As noted in Sec. III-B, this paper focused on monthly or yearly statistical data and supposed that the length of time series should be short enough to grasp the characteristics of data at each t and the entire time range. To analyze continuous data such as sensor data, preprocessing such as SAX (symbolic aggregate approximation) [68] should be integrated as the preliminary step in the entire analytic process. To promote the preprocessing of time-series data such as sampling and smoothing, we also think that it is necessary to combine the proposed analytical framework with existing interactive preprocessing frameworks [36].

C. COMPREHENSIVE SUPPORT FOR METRICS FORMULATION

Although the prototype interface uses an animated scatter plot as one of the main visualization methods, we should consider adopting other visualization methods to support more complex VA tasks. In a preliminary user study, a user wanted to compare data at each time point by juxtaposing multiple scatter plots. To address this issue, the interface can be improved from the aspect of visualization, such as providing functions for juxtaposing multiple scatter-plot views after selecting the individual time points separately. Also, we think it is useful to combine views in three-dimensional space and switch visualization methods, such as animation and small multiple, in a similar manner to immersive analytics [69].

Another direction of extending the proposed interface is to deal with nonlinearity to cover various forms of metrics. The introduction of nonlinear dimensionality reduction such as t-SNE and Kernel PCA, and visualization on the basis of a nonlinear axis (e.g., logarithmic) would be possible approaches to that extension. Rather than directly using parameters of nonlinear dimensionality reduction, we think it is effective to apply such a dimensionality reduction after users interactively adjust the weights of each attribute, as in [70], and to formulate metrics based on the obtained axis. In addition to that, introducing a function to switch multiple dimensionality-reduction algorithms is effective for broadening the applicability of the proposed interface.

VIII. CONCLUSION AND FUTURE WORK

We proposed a VA interface of metrics formulation for multi-dimensional time-series data. An example using statistical data of a baseball game showed how the proposed interface can support the process of metrics formulation. The results of the experiments with participants confirmed that the combination of different types of visualized objects (nodes, trajectories, and convex hulls) makes it possible to explore multi-dimensional time-series data, and the combination of absolute manipulation, relative manipulation, and manual parameter adjustment makes it possible to fine-tune the weights of attributes acquired from PCA. Therefore, we think that the proposed interface can support the task of metrics formulation for multi-dimensional time-series data. The proposed interface is the first attempt at providing VA to support these tasks: we expect the results obtained in this study to promote discussions on the extension of semantic interaction towards time-series data. To extend the proposed interface for covering a wide range of metrics and data types, an integrated framework that includes preprocessing and multiple dimensionality-reduction algorithms should be considered in future works. In addition to performing additional experiments to quantitatively evaluate the usability of the interface, as discussed in Sec. VII-A, we plan to do long-term case studies with domain experts to qualitatively verify the effectiveness of the proposed interface and the advantages against other VA interfaces.

IX. EVALUATION OF THE FORMULATED METRICS

To evaluate the B_{type} metrics formulated in the example (Sec. V), we compared the number of games played, TB, OBP, and on-base plus slugging (OPS) of the top (bottom) 10 players of B_{type} in the 2019 season. OPS is a metric of a batter's ability to score runs which are defined as OBP + TB/AB. The data is obtained in the same way as those used in Sec. V. The number of instances (batters) is 187. Table 2 shows the metrics for batter's performance at the early stage of the season (April 15), and Table 3 shows the metrics at the end of the season (September 16).

TABLE 2. Evaluation of batting performance by various metrics in 2019 season (April 15).

Player	Games Played	B_{type}	TB	OBP	OPS
Top 10					
Dee Strange-Gordon	19	0.677	26	0.316	0.698
Tim Anderson*	14	0.645	36	0.431	1.063
Whit Merrifield‡*	17	0.626	36	0.342	0.828
Elvis Andrus*	16	0.625	42	0.420	1.047
Mallex Smith*	16	0.452	21	0.311	0.639
Leury García*	14	0.423	26	0.312	0.732
DJ LeMahieu†‡	15	0.403	25	0.433	0.905
Jonathan Lucroy	15	0.398	14	0.283	0.558
Adalberto Mondesi*	17	0.394	36	0.286	0.793
Billy Hamilton	14	0.380	9	0.265	0.470
Bottom 10					
Jay Bruce	17	-0.629	37	0.264	0.851
Asdrúbal Cabrera	15	-0.464	32	0.283	0.876
Khris Davis	19	-0.435	51	0.316	1.006
Mitch Moreland	17	-0.430	33	0.300	0.900
Daniel Vogelbach‡	14	-0.430	36	0.434	1.271
Christin Stewart	15	-0.323	27	0.300	0.829
Jorge Soler*	17	-0.304	31	0.271	0.748
Robinson Chirinos	13	-0.304	20	0.370	0.925
Randal Grichuk	18	-0.287	29	0.319	0.780
Gary Sánchez‡	12	-0.255	30	0.306	0.973

TABLE 3. Evaluation of batting performance by various metrics in 2019 season (September 16).

Player	Games Played	B_{type}	TB	OBP	OPS
Top 10					
Dee Strange-Gordon	117	0.494	141	0.302	0.660
Myles Straw	56	0.484	37	0.375	0.718
Adalberto Mondesi*	102	0.449	176	0.289	0.713
Harold Castro	97	0.431	136	0.304	0.688
Víctor Reyes	69	0.421	119	0.336	0.767
Mallex Smith*	134	0.399	171	0.299	0.634
Billy Hamilton	119	0.397	87	0.286	0.561
Delino DeShields	118	0.387	124	0.319	0.666
Luis Arraez	92	0.385	143	0.399	0.838
Elvis Andrus*	147	0.381	236	0.313	0.707
Bottom 10					
Jay Bruce	98	-0.138	162	0.261	0.784
Mitch Garver†	93	-0.106	196	0.365	0.995
Miguel Sanó	105	-0.096	219	0.346	0.923
Gary Sánchez‡	106	-0.070	208	0.316	0.841
Mike Ford	50	-0.061	80	0.350	0.909
Yordan Álvarez*	87	-0.051	205	0.412	1.067
Jorge Soler*	162	-0.046	335	0.352	0.918
Max Kepler	134	-0.030	272	0.336	0.855
Justin Bour	52	-0.024	55	0.259	0.623
Matt Olson†	127	-0.022	263	0.351	0.896

In both tables, winners of the Silver Slugger Award or the Gold Glove Award are annotated with †, and batters who participated in the All-Star Game, who had an outstanding performance in the first half of the season, are annotated with ‡. Batters who got titles such as batting champion in the league are annotated with $*$. In Tables 2 and 3, batters who were active throughout the year, such as those marked with † and $*$ were also found in both the top and bottom of the rankings.

To evaluate the consistency of the rankings generated by B_{type}, we calculated MRRs of B_{type}, TB, OBP, and OPS on

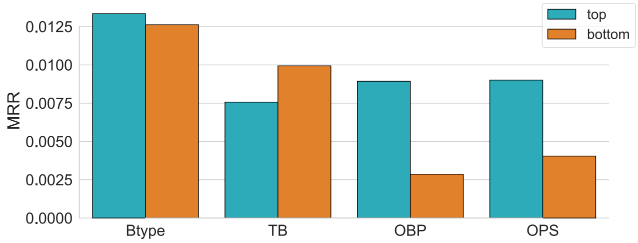


FIGURE 11. MRR (mean reciprocal rank) calculated for the top/bottom 20 batters in ranking by B_{type}, TB, OBP, and OPS on April 15 against a ranking of those metrics on September 16.

April 15 when the rank of these metrics on September 16 was considered as the ground truth. Fig. 11 shows the results of MRRs calculated for the top and bottom 20 batters on April 15. From Fig. 11 it can be confirmed that the MRR of B_{type} is higher than that of the other metrics. In other words, we think that B_{type} has achieved the purpose of the evaluation metrics described in Sec. V, which is to evaluate batters who have high performance even from the beginning of the season.

REFERENCES

[1] M. Badawy, A. A. A. El-Aziz, A. M. Idress, H. Hefny, and S. Hossam, "A survey on exploring key performance indicators," *Future Comput. Informat. J.*, vol. 1, nos. 1–2, pp. 47–52, Dec. 2016.

[2] A. C. Chen and X. Fu, "Data + intuition: A hybrid approach to developing product north star metrics," in *Proc. 26th Int. Conf. World Wide Web Companion (WWW)*, Apr. 2017, pp. 617–625.

[3] J. Albert, "Sabermetrics: The past, the present, and the future," in *Mathematics and Sports*, J. A. Gallian, Ed. Washington, DC, USA: Mathematical Association of America, 2010, pp. 3–14.

[4] E. Wall, L. M. Blaha, L. Franklin, and A. Endert, "Warning, bias may occur: A proposed approach to detecting cognitive bias in interactive visual analytics," in *Proc. IEEE Conf. Vis. Anal. Sci. Technol. (VAST)*, Oct. 2017, pp. 104–115.

[5] D. A. Keim, F. Mansmann, J. Schneidewind, J. Thomas, and H. Ziegler, "Visual analytics: Scope and challenges," in *Visual Data Mining: Theory, Techniques and Tools for Visual Analytics*, S. J. Simoff, M. H. Böhlen, and A. Mazeika, Eds. Berlin, Germany: Springer, 2008, pp. 76–90.

[6] D. Sacha, M. Sedlmair, L. Zhang, J. A. Lee, J. Peltonen, D. Weiskopf, S. C. North, and D. A. Keim, "What you see is what you can change: Human-centered machine learning by interactive visualization," *Neurocomputing*, vol. 268, pp. 164–175, Dec. 2017.

[7] A. Endert, M. S. Hossain, N. Ramakrishnan, C. North, P. Fiaux, and C. Andrews, "The human is the loop: New directions for visual analytics," *J. Intell. Inf. Syst.*, vol. 43, no. 3, pp. 411–435, 2014.

[8] A. Endert, W. Ribarsky, C. Turkay, B. L. W. Wong, I. Nabney, I. D. Blanco, and F. Rossi, "The state of the art in integrating machine learning into visual analytics," *Comput. Graph. Forum*, vol. 36, no. 8, pp. 458–486, Dec. 2017.

[9] R. Takami and Y. Takama, "Proposal and evaluation of visual analytics interface for time-series data based on trajectory representation," *IEICE Trans. Inf. Syst.*, vol. E103.D, no. 1, pp. 142–151, 2020.

[10] M. Gleicher, "Explainers: Expert explorations with crafted projections," *IEEE Trans. Vis. Comput. Graphics*, vol. 19, no. 12, pp. 2042–2051, Dec. 2013.

[11] E. Wall, S. Das, R. Chawla, B. Kalidindi, E. T. Brown, and A. Endert, "Podium: Ranking data using mixed-initiative visual analytics," *IEEE Trans. Vis. Comput. Graphics*, vol. 24, no. 1, pp. 288–297, Jan. 2018.

[12] W. Aigner, S. Miksch, W. Müller, H. Schumann, and C. Tominski, "Visualizing time-oriented data—A systematic view," *Comput. Graph.*, vol. 31, no. 3, pp. 401–409, Jun. 2007.

[13] S. Kriglstein, M. Pohl, and M. Smuc, "Pep up your time machine: Recommendations for the design of information visualizations of time-dependent data," in *Handbook of Human Centric Visualization*, W. Huang, Ed. New York, NY, USA: Springer, 2014, pp. 203–225.

[14] R. Takami, H. Shibata, and Y. Takama, "An analytical framework for formulating metrics for evaluating multi-dimensional time-series data," in *Proc. 25th Int. Conf. Intell. User Interfaces (IUI)*, Mar. 2020, pp. 207–211.

[15] R. Mazza, *Introduction to Information Visualization*, 1st ed. London, U.K.: Springer, 2009.

[16] S. Liu, D. Maljovec, B. Wang, P.-T. Bremer, and V. Pascucci, "Visualizing high-dimensional data: Advances in the past decade," *IEEE Trans. Vis. Comput. Graphics*, vol. 23, no. 3, pp. 1249–1268, Mar. 2017.

[17] D. Sacha, A. Stoffel, F. Stoffel, B. C. Kwon, G. Ellis, and D. A. Keim, "Knowledge generation model for visual analytics," *IEEE Trans. Vis. Comput. Graphics*, vol. 20, no. 12, pp. 1604–1613, Dec. 2014.

[18] S. Amershi, M. Cakmak, W. B. Knox, and T. Kulesza, "Power to the people: The role of humans in interactive machine learning," *AI Mag.*, vol. 35, no. 4, pp. 105–120, 2014.

[19] E. Horvitz, "Principles of mixed-initiative user interfaces," in *Proc. SIGCHI Conf. Human factors Comput. Syst. (CHI)*, 1999, pp. 159–166.

[20] A. Endert, L. Bradel, and C. North, "Beyond control panels: Direct manipulation for visual analytics," *IEEE Comput. Graph. Appl.*, vol. 33, no. 4, pp. 6–13, Jul./Aug. 2013.

[21] S. Makonin, D. McVeigh, W. Stuerzlinger, K. Tran, and F. Popowich, "Mixed-initiative for big data: The intersection of human + visual analytics + prediction," in *Proc. 49th Hawaii Int. Conf. Syst. Sci. (HICSS)*, Jan. 2016, pp. 1427–1436.

[22] A. Endert, P. Fiaux, and C. North, "Semantic interaction for visual text analytics," in *Proc. SIGCHI Conf. Hum. Factors Comput. Syst. (CHI)*, May 2012, pp. 473–482.

[23] J. Z. Self, X. Hu, L. House, S. Leman, and C. North, "Designing usable interactive visual analytics tools for dimension reduction," in *Proc. CHI Workshop Hum.-Centered Mach. Learn. (HCML)*, 2016, pp. 1–7.

[24] S. Ingram, T. Munzner, V. Irvine, M. Tory, S. Bergner, and T. Möller, "DimStiller: Workflows for dimensional analysis and reduction," in *Proc. IEEE Symp. Vis. Anal. Sci. Technol. (VAST)*, Oct. 2010, pp. 3–10.

[25] L. G. Nonato and M. Aupetit, "Multidimensional projection for visual analytics: Linking techniques with distortions, tasks, and layout enrichment," *IEEE Trans. Vis. Comput. Graphics*, vol. 25, no. 8, pp. 2650–2673, Aug. 2019.

[26] A. Sarikaya and M. Gleicher, "Scatterplots: Tasks, data, and designs," *IEEE Trans. Vis. Comput. Graphics*, vol. 24, no. 1, pp. 402–412, Jan. 2018.

[27] P. Joia, F. V. Paulovich, D. Coimbra, J. A. Cuminato, and L. G. Nonato, "Local affine multidimensional projection," *IEEE Trans. Vis. Comput. Graphics*, vol. 17, no. 12, pp. 2563–2571, Dec. 2011.

[28] F. V. Paulovich, C. T. Silva, and L. G. Nonato, "User-centered multidimensional projection techniques," *Comput. Sci. Eng.*, vol. 14, no. 4, pp. 74–81, 2012.

[29] T. Fujiwara, J.-K. Chou, Shilpika, P. Xu, L. Ren, and K.-L. Ma, "An incremental dimensionality reduction method for visualizing streaming multidimensional data," *IEEE Trans. Vis. Comput. Graphics*, vol. 26, no. 1, pp. 418–428, Jan. 2020.

[30] D. H. Jeong, C. Ziemkiewicz, B. Fisher, W. Ribarsky, and R. Chang, "iPCA: An interactive system for PCA-based visual analytics," in *Proc. 11th Eurograph./IEEE VGTC Conf. Vis.*, Jun. 2009, pp. 767–774.

[31] H. Kim, J. Choo, H. Park, and A. Endert, "InterAxis: Steering scatterplot axes via observation-level interaction," *IEEE Trans. Vis. Comput. Graphics*, vol. 22, no. 1, pp. 131–140, Jan. 2016.

[32] E. T. Brown, J. Liu, C. E. Brodley, and R. Chang, "Dis-function: Learning distance functions interactively," in *Proc. IEEE Conf. Vis. Anal. Sci. Technol. (VAST)*, Oct. 2012, pp. 83–92.

[33] B. C. Kwon, H. Kim, E. Wall, J. Choo, H. Park, and A. Endert, "AxiSketcher: Interactive nonlinear axis mapping of visualizations through user drawings," *IEEE Trans. Vis. Comput. Graphics*, vol. 23, no. 1, pp. 221–230, Jan. 2017.

[34] P. A. Legg, "Visualizing the insider threat: Challenges and tools for identifying malicious user activity," in *Proc. IEEE Symp. Vis. Cyber Secur.*, Oct. 2015, pp. 1–7.

[35] M. Cavallo and Ç. Demiralp, "A visual interaction framework for dimensionality reduction based data exploration," in *Proc. CHI Conf. Hum. Factors Comput. Syst. (CHI)*, Apr. 2018, pp. 635:1–635:13.

[36] J. Bernard, M. Hutter, H. Reinemuth, H. Pfeifer, C. Bors, and J. Kohlhammer, "Visual-interactive preprocessing of multivariate time series data," *Comput. Graph. Forum*, vol. 38, no. 3, pp. 401–412, Jun. 2019.

[37] W. Aigner, S. Miksch, W. Müller, H. Schumann, and C. Tominski, "Visual methods for analyzing time-oriented data," *IEEE Trans. Vis. Comput. Graphics*, vol. 14, no. 1, pp. 47–60, Jan./Feb. 2008.

[38] G. Robertson, R. Fernandez, D. Fisher, B. Lee, and J. Stasko, "Effectiveness of animation in trend visualization," *IEEE Trans. Vis. Comput. Graphics*, vol. 14, no. 6, pp. 1325–1332, Nov./Dec. 2008.

[39] C. Perin, T. Wun, R. Pusch, and S. Carpendale, "Assessing the graphical perception of time and speed on 2D+time trajectories," *IEEE Trans. Vis. Comput. Graphics*, vol. 24, no. 1, pp. 698–708, Jan. 2018.

[40] S. Kriglstein, M. Pohl, and C. Stachl, "Animation for time-oriented data: An overview of empirical research," in *Proc. 16th Int. Conf. Inf. Vis.*, Jul. 2012, pp. 30–35.

[41] J. Bernard, N. Wilhelm, M. Scherer, T. May, and T. Schreck, "TimeSeriesPaths: Projection-based explorative analysis of multivariate time series data," *J. WSCG*, vol. 20, no. 2, pp. 97–106, 2012.

[42] A. Rind, W. Aigner, S. Miksch, S. Wiltner, M. Pohl, F. Drexler, B. Neubauer, and N. Suchy, "Visually exploring multivariate trends in patient cohorts using animated scatter plots," in *Proc. Int. Conf. Ergonom. Health Aspects Work Comput.*, 2011, pp. 139–148.

[43] E. F. Vernier, R. Garcia, I. P. D. Silva, J. L. D. Comba, and A. C. Telea, "Quantitative evaluation of time-dependent multidimensional projection techniques," *Comput. Graph. Forum*, vol. 39, no. 3, pp. 241–252, Jun. 2020.

[44] O. Patashnik, M. Lu, A. H. Bermano, and D. Cohen-Or, "Temporal scatterplots," *Comput. Vis. Media*, vol. 6, no. 4, pp. 385–400, Dec. 2020.

[45] A. Endert, W. A. Pike, and K. Cook, "From streaming data to streaming insights: The impact of data velocities on mental models," in *Proc. Workshop Hum. Centered Big Data Res.*, 2014, pp. 24:24–24:26.

[46] R. V. Rao and B. K. Patel, "A subjective and objective integrated multiple attribute decision making method for material selection," *Mater. Des.*, vol. 31, no. 10, pp. 4738–4747, Dec. 2010.

[47] World Health Organization. (Jun. 5, 2020). *Monitoring and Evaluation Framework. COVID-19 Strategic Preparedness and Response*. Accessed: Jun. 5, 2021. [Online]. Available: https://www.who.int/publications/i/item/monitoring-and-evaluation-framework

[48] J. Zhao, M. Karimzadeh, L. S. Snyder, C. Surakitbanharn, Z. C. Qian, and D. S. Ebert, "MetricsVis: A visual analytics system for evaluating employee performance in public safety agencies," *IEEE Trans. Vis. Comput. Graphics*, vol. 26, no. 1, pp. 1193–1203, Jan. 2020.

[49] M. B. Bjerke and R. Renger, "Being smart about writing SMART objectives," *Eval. Program Planning*, vol. 61, pp. 125–127, Apr. 2017.

[50] P. Dmitriev and X. Wu, "Measuring metrics," in *Proc. 25th ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, 2016, pp. 429–437.

[51] M. P. Brundage, W. Z. Bernstein, K. Morris, and J. A. Horst, "Using graph-based visualizations to explore key performance indicator relationships for manufacturing production systems," in *Proc. 24th CIRP Conf. Life Cycle Eng.*, vol. 61, 2017, pp. 451–456.

[52] C. Milhaupt. (2018). *A Smarter Way to Watch Early Season Baseball*. Accessed: Jun. 5, 2021. [Online]. Available: http://bronxpinstripes.com/featured-column/a-smarter-way-to-watch-early-season-baseball/

[53] Z. W. Arth and A. C. Billings, "Batting average and beyond: The framing of statistics within regional major league baseball broadcasts," *Int. J. Sport Commun.*, vol. 14, no. 2, pp. 212–232, 2021.

[54] D. Sacha, L. Zhang, M. Sedlmair, J. A. Lee, J. Peltonen, D. Weiskopf, S. C. North, and D. A. Keim, "Visual interaction with dimensionality reduction: A structured literature analysis," *IEEE Trans. Vis. Comput. Graphics*, vol. 23, no. 1, pp. 241–250, Jan. 2017.

[55] J. F. Helliwell, R. Layard, and J. D. Sachs. (2019). *World Happiness Report*. Accessed: Jun. 5, 2021. [Online]. Available: https://worldhappiness.report

[56] X. Ye, R. Bunescu, and C. Liu, "Learning to rank relevant files for bug reports using domain knowledge," in *Proc. 22nd ACM SIGSOFT Int. Symp. Found. Softw. Eng. (FSE)*, Nov. 2014, pp. 689–699.

[57] J. Bernard, D. Sessler, T. Ruppert, J. Davey, A. Kuijper, and J. Kohlhammer, "User-based visual-interactive similarity definition for mixed data objects—Concept and first implementation," in *Proc. 22nd Int. Conf. Central Eur. Comput. Graph., Vis. Comput. Vis.*, 2014, pp. 329–338.

[58] A. Dasgupta, D. L. Arendt, L. R. Franklin, P. C. Wong, and K. A. Cook, "Human factors in streaming data analysis: Challenges and opportunities for information visualization," *Comput. Graph. Forum*, vol. 37, no. 1, pp. 254–272, Feb. 2018.

[59] B. Kondo and C. Collins, "DimpVis: Exploring time-varying information visualizations by direct manipulation," *IEEE Trans. Vis. Comput. Graphics*, vol. 20, no. 12, pp. 2003–2012, Dec. 2014.

[60] T. Schreck and C. Panse, "A new metaphor for projection-based visual analysis and data exploration," *Proc. SPIE*, vol. 6495, Jan. 2007, Art. no. 64950L.

[61] T. V. Landesberger, S. Bremm, T. Schreck, and D. W. Fellner, "Feature-based automatic identification of interesting data segments in group movement data," *Inf. Vis.*, vol. 13, no. 3, pp. 190–212, 2014.

[62] Adobe. (2020). *Select With the Lasso Tool*. Accessed: Jun. 5, 2021. [Online]. Available: https://helpx.adobe.com/photoshop/using/selecting-lasso-tools.html

[63] I. T. Jolliffe, *Principal Component Analysis*. New York, NY, USA: Springer-Verlag, 1986.

[64] SR LLC. (2020). *Baseball-Reference.com—Major League Statistics and Information*. Accessed: Jun. 5, 2021. [Online]. Available: https://www.baseball-reference.com/

[65] W. Aigner, A. Rind, and S. Hoffmann, "Comparative evaluation of an interactive time-series visualization that combines quantitative data with qualitative abstractions," *Comput. Graph. Forum*, vol. 31, no. 3, pp. 995–1004, Jun. 2012.

[66] L. Byron and M. Wattenberg, "Stacked graphs—Geometry & aesthetics," *IEEE Trans. Vis. Comput. Graphics*, vol. 14, no. 6, pp. 1245–1252, Nov. 2008.

[67] K. Nakakoji and Y. Yamamoto, "What does the representation talk back to you?" *Knowl.-Based Syst.*, vol. 14, no. 8, pp. 449–453, Dec. 2001.

[68] J. Lin, E. Keogh, S. Lonardi, and B. Chiu, "A symbolic representation of time series, with implications for streaming algorithms," in *Proc. 8th ACM SIGMOD Workshop Res. Issues Data Mining Knowl. Discovery*, Jun. 2003, pp. 2–11.

[69] M. Cordeil, A. Cunningham, T. Dwyer, B. H. Thomas, and K. Marriott, "ImAxes: Immersive axes as embodied affordances for interactive multivariate data visualisation," in *Proc. 30th Annu. ACM Symp. User Interface Softw. Technol. (UIST)*, Oct. 2017, pp. 71–83.

[70] W. Jentner, D. Sacha, F. Stoffel, G. Ellis, L. Zhang, and D. A. Keim, "Making machine intelligence less scary for criminal analysts: Reflections on designing a visual comparative case analysis tool," *Vis. Comput.*, vol. 34, no. 9, pp. 1225–1241, Sep. 2018.

REI TAKAMI received the M.S. degree from Tokyo Metropolitan University, in 2020. He is currently working as a Software Engineer at Yahoo Japan Corporation, Tokyo. His main research interests include information visualization and visual analytics of time-series data.

HIROKI SHIBATA (Member, IEEE) received the Dr.Eng. degree from Tokyo Metropolitan University. He has been an Assistant Professor with the Graduate School of System Design, Tokyo Metropolitan University, since 2019. His current research interests include optimization, stochastic model, and information recommendation. He is a member of the ACM and Japanese Society of Artificial Intelligence (JSAI).

YASUFUMI TAKAMA (Member, IEEE) received the Dr.Eng. degree from The University of Tokyo, in 1999. From 1999 to 2002, he was a Research Associate with the Interdisciplinary Graduate School of Science and Engineering, Tokyo Institute of Technology. From 2002 to 2005, he was an Associate Professor with the Department of Electronic Systems and Engineering, Tokyo Metropolitan Institute of Technology. From 2005 to 2013, he was an Associate Professor with the Faculty of System Design, Tokyo Metropolitan University, Tokyo, Japan, where he has been a Professor, since 2014. His current research interests include Web intelligence, information visualization, and information recommendation. He is a member of the ACM.