Computing and Informatics, Vol. 42, 2023, 690–715, doi: 10.31577/cai_2023_3_690

MTREEILLUSTRATOR: A MIXED-INITIATIVE FRAMEWORK FOR VISUAL EXPLORATORY ANALYSIS OF MULTIDIMENSIONAL HIERARCHICAL DATA

Guijuan Wang

Information and Technology School, Computer Science and Technology School
Southwest University of Science and Technology
Mianyang 621010, China
e-mail: guijuanwang@swust.edu.cn

Yu Zhao

Institute of Rural Development, Shandong Academy of Social Sciences
Jinan 250002, China
e-mail: yuzhaosdass@foxmail.com

Boyou Tan, Zhong Wang, Jiansong Wang, Hao Guo

Computer Science and Technology School
Southwest University of Science and Technology
Mianyang 621010, China
e-mail: 905109256, 78339239, 1666938053, 2698896107@qq.com

Yadong Wu*

Computer Science and Engineering School
Sichuan University of Science and Engineering
Zigong 645002, China
e-mail: wyd028@163.com

* Corresponding author

Abstract. Multidimensional hierarchical (mTree) data are very common in daily life and scientific research. However, mTree data exploration is a laborious and time-consuming process due to its structural complexity and large dimension combination space. To address this problem, we present mTreeIllustrator, a mixed-initiative framework for exploratory analysis of multidimensional hierarchical data with faceted visualizations. First, we propose a recommendation pipeline for the automatic selection and visual representation of important subspaces of mTree data. Furthermore, we design a visual framework and an interaction schema to couple automatic recommendations with human specifications to facilitate progressive exploratory analysis. Comparative experiments and user studies demonstrate the usability and effectiveness of our framework.

Keywords: Multidimensional hierarchical data, visual exploratory analysis, visualization recommendation, faceted visualization

1 INTRODUCTION

Multidimensional hierarchical data are commonly seen in life and scientific research; examples include census data, enterprise organization data and biological structure data. We call a multidimensional hierarchical structure an mTree for brevity considering that a tree is the most distinctive graphical depiction of a hierarchical structure [1]. Because the widths and depths of different layers and branches vary widely, mTree data feature high structural complexity of structure and an immensely high-dimensional combination space, which makes the exploration of such data a challenging task [2]. Users must go through a tedious and time-consuming process to interactively check and refine the exploration process to search for the combinations that are interesting or useful [3]. Machine learning and visualization can be adopted to accelerate exploration. Machine learning is leveraged to recommend the most important subset to decrease the search space, and visualization is used to present complex data and structures with intuitive graphical representations. Instead of repeated manual iterations, the intelligent visualization recommender can ease the exploration process by suggesting both important data and graphical views for analysts to browse [4].

However, creating intelligent visualization recommender system for mTree data is not easy. It requires a high level of expertise in mTree data visualization. On the one hand, visual mTree data exploration involves both multidimensional information understanding and hierarchical structure perception. To present the knowledge contained in multiple dimensions, techniques that organize the multiple dimensions in one chart are available, such as radar charts, parallel coordinate plots (PCPs) [5] and scatter plot matrices, faceted visualization techniques that organize several simple charts together, where each chart encodes one facet can also be used, such as the small-multiple and multiple coordinate view (MCV) technique. To present hierarchical information, many different visualization methods have been developed. Schulz maintains an online survey treeVis website [6], but determining which is the most suitable method for a given dataset can be a challenge [7]. In practice, mTree data exploration often requires bespoke [8, 2] to combine multidimensional and hierarchical information. For ordinary users without programming skills, a more automatic technique would be more feasible.

On the other hand, some automatic tools have been developed to reduce the technique threshold of visualization, including rule-based recommendation tools [9, 10] ranking mechanic-based tools [11, 10], machine learning-based tools [12, 11] mixed-initiative tools [13]. These tools are mainly designed for tabular data. They do not directly support multidimensional hierarchical data.

To bridge the gap in intelligent visual mTree data exploration, we propose an automatic pipeline and a visual analytic framework mTreeIllustrator. Considering the large combination space of mTree data, the mTreeIllustrator cannot cover all possible combinations and visual representations. Inspired by the mixed-initiative user interface paradigm that enables human to collaborate with the intelligent agents [14]. We integrate the auto-generated faceted visualization into the interactive human exploration, to inspire users to efficiently interpret the mTree data and update their exploration directions. The main contributions are as follows.

1. We propose a novel machine-learning powered pipeline for the workflow of automatic mTree data visualization. With it, the most important subspace of mTree data is automatically selected and encoded as faceted visualizations.
2. We design a mixed-initiative visual analytic framework to couple the intelligent visualization recommendations with user selections to support progressive mTree data exploration. The framework also enables users to refine the recommended visualizations and data subspace, and to visually compare mTree structures.
3. We demonstrate the usability and effectiveness of the proposed method and framework by the comparative performance experiments and user studies.

The rest of this paper is organized as follows: Section 2 discusses the related work. Section 3 describes the task and architecture. The proposed model is presented in Section 4. The visualization design of mTreeIllustrator is presented in Section 5. Section 6 provides a systematic evaluation. We conclude our work in Section 7.

2 RELATED WORKS

This section presents the research topics that are most relevant to our work, namely, mTree data visualization and visualization recommendation.

2.1 Multidimensional and Hierarchical Data Visualization

Compared with tabular data, mTree data are more complex in terms of both their structures and information organization patterns. Visualization plays an important role in exploring complex data [15]. Researchers have introduced various visualization techniques to improve the efficiency of analyzing multidimensional and hierarchical data. TreeVersity [16, 15] explores the cyclical changes in each dimension of mTree data by introducing visualizations such as tables and time trend charts. The McVA system [17] designs multiple coordinate views by combining hierarchical bubble charts, PCP charts, word cloud charts, and radar plots to perform a comparative analysis of different countries and regions. Sakairi et al. [18] conducted a visual comparison analysis on the dosages of different products materials by combining hierarchical data with stacked plots. Li [1] developed a hierarchical data comparison system that supports the interactive exploration and analysis of hierarchical data and allows users to visualize data by selecting different hierarchical visual layout algorithms to understand the characteristics of the data. The MCT method [2] uses a combination of rectangular tree diagrams and PCP charts to assist with the exploration of multidimensional information in hierarchical structures. A rectangular tree diagram is used to encode a hierarchy, and the four edges of the diagram are used as the four axes of parallel coordinates. Limited by the edge count of a rectangle, it can visualize at most four dimensions. Zhou et al. [19] proposed a visualization method to uncover the relationships of multiple attributes. PCP charts and visual interaction techniques are used to assist the analysis process and can help data analyst visually analyse the relationships between multiple attributes and target variables. The relationships are encoded using the sunburst diagram. With this diagram, analysts can determine the overall attribute relationships at a glance. Although the above techniques contribute greatly to mTree data exploration, the techniques themselves are relatively complex and require users to have some visualization knowledge.

2.2 Visualization Recommendation

The goal of visualization recommendation is to automatically recommend suitable charts based on the data characteristics of the given data to lower the technical threshold of visualization and improve the efficiency of data exploration. A number of mechanisms have been proposed to assist with visualization recommendation, mainly including rule-based methods and machine learning-based approaches.

Rule-based visualization recommendation can be traced back to the APT tool [20], developed by Mackinlay in the 1980s; this tool can automatically design effective graphical representations of relational information (e.g., bar charts, scatter diagrams, and connection diagrams). The tool is implemented using synthetic algebra and graph design guidelines. Mackinlay considered graphical representations as sentences of a graphical language. A wide variety of designs can be systematically generated by using the composite algebra that makes up a small set of the original graphical languages. In 1994, Sage [21] extended APT with more properties and enhanced the user-oriented design. In 2007, ShowMe [22] extended automatic representation to charts tables (often called small multiple displays), where VizQL is based on the algebra used in APT, thus improving the algebra and enabling compilation into a database query language. Recently, Voyager [3, 4] aggregated the knowledge derived from previous works using expressiveness and validity criteria to evaluate visual coding options; this method integrates manual selection and rule-based selection and enables users to engage in interactive browsing and refinement based on multiple recommendations. In 2019, Moritz et al. [10] proposed Draco, which develops hard constraints (e.g., the shape encoding channel cannot represent quantity values) and soft constraints (e.g., by default, the temporal field is mapped to the X-axis) based on common visual design guidelines and uses those rules to recommend charts. Nan et al. [23] defined a set of visual language rules based on data transformation, aggregation and visual mapping; summarized seven common visualization tasks; and then recommended visualization charts based on these rules and tasks.

With the expansion of machine learning, many creative works have been proposed for visual chart recommendations based on artificial intelligence. DeepEye [11] trained a recommendation model based on RankSVM. Given a dataset, the model can select valid charts based on the data characteristics and rank them to obtain the top-k options. Dibia et al. [24] proposed Data2Vis, a neural network-based translation model for automatically generating visualizations from a given dataset. In this approach, the visualization generation problem is formulated as a language translation problem, where the data specification is mapped to the visualization specification using the Vega-Lite declarative language [25]. Text-to-Viz [26] supports automatic infographics generation from natural language statements. VizML [12] considers chart recommendation as a prediction problem, where the model predicts the visual encoding of data for the given data column(s).

The above research demonstrates the effectiveness of recommendation-based approaches in data visualization. However, the existing work has mainly focused on tabular data. Compared to tabular data, hierarchical data are more complex and cannot be directly supported. To address this problem, we propose an automatic pipeline and visualization framework for the visual exploration of mTree data.

3 DESIGN REQUIREMENTS AND ARCHITECTURE

3.1 Design Requirements

Based on the research problem, we have identified the following design requirements that form our automatic pipeline and the visual analysis framework.

R1. Automatic dimension combination and selection: After obtaining new data, users typically need to repeatedly select and check different dimension combinations to obtain meaningful results. Such repetitive tasks should be improved by automation procedures.

R2. Automatic chart recommendation: The target users have little or no visualization knowledge, so the system should be able to automatically help the user determine the appropriate visualization for a given dimension or dimension combination.

R3. Support for iterative dimensions and charts refinement: The recommendation provided by a machine learning model may not be optimal. Sometimes the users want to change dimensions or refine the visual coding of a chart to meet their expectations, for example, adding new dimensions or changing the color of a scatter plot.

R4. Support for interactive exploration and comparisons involving hierarchical data: The developed system can support visual explorations and comparisons of hierarchical data with different sizes and granularities, it allows users to select a branch of the input hierarchical data for data dimension exploration, and it supports the comparison of data from different branches.

R5. Support for exploration history tracking: Unlike tabular data, hierarchical data possess a more complex exploration path. Therefore, the design should track and visualize the users' exploration path so that users can clearly know where they are and how they arrived there at any time to lighten their memory burden.

3.2 The Architecture

As shown in Figure 1, the architecture of mTreeIllustrator consists of an automatic recommendation pipeline module (Figure 1, right) and an interactive visualization module (Figure 1, left). After a user uploads data via the graphical user interface, the data are sent to the automatic pipeline. The machine learning-enabled pipeline includes three seamlessly integrated models. First, the subspace importance assessment model evaluates the importance of each dimension of the given mTree data using the random forest (RF) algorithm and outputs the most important subspace to the visualization recommendation model. The recommendation model predicts the chart type for each valid dimension or dimension combination and passes these chart types to the rule-based chart encoding model. Last, the encoding model translates the subspace data and chart types into graphical charts and sends the visualizations back to the user interface. Then, users can interactively explore and prioritize their exploration based on the recommendation results. The UI also provides a set of intuitive visual designs to present the overall mTree structure and the exploration path to simplify the process of exploring complex tree structures.

4 AUTOMATIC PIPELINE

We propose an automatic pipeline to assist users in exploring mTree data. The automatic pipeline consists of a dimension importance evaluation model, a visualization recommendation model and a rule-based chart encoding model. Those models are seamlessly connected, take the user data as inputs, select the most important subspace of the data, and present the subspace visually.

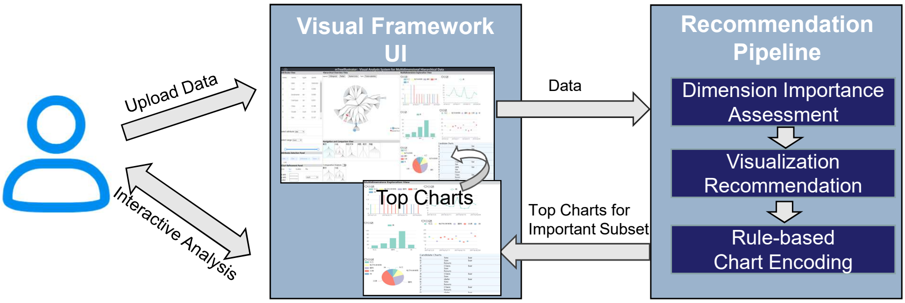

Upload Data
Interactive Analysis
Visual Framework UI
Top Charts
Data
Top Charts for Important Subset
Recommendation Pipeline
Dimension Importance Assessment
Visualization Recommendation
Rule-based Chart Encoding

Figure 1. The architecture of mTreeIllustrator

4.1 Subspace Importance Evaluation Model

To understand mTree data, users need to iteratively check different dimensions and dimension combinations among layers and branches of the tree. With the numerous combinations of dimensions, layers and branches, the search space is large. Although many combinations are not important, users spend considerable time traversing them. To avoid wasting time on low-information dimensions or combinations, we propose a subspace importance assessment algorithm to allow the user to start their exploration from the most important combinations.

Several machine learning models are capable of subspace selection. Considering the interpretability, our model is designed based on the RF algorithm. The RF algorithm is used to select the subspace with the most important dimensions.

An RF comprises multiple tree sets (TSs). The majority of the tree decisions form the final decision. Each tree in a TS is a binary tree. The root node contains all training samples. According to a certain principle, each node selects the dimension that minimizes the "impurity" and uses this dimension as the branching dimension to split the node into two branches, each of which contains the corresponding sub samples. This process is repeated until the stopping condition is satisfied.

The frequently used measurements for "impurity" are the Gini index and out-of-bag (OOB) error. The accuracy of the Gini index is higher than that of the OOB when the signal-to-noise ratio is low, but in practice, it is difficult to obtain data with a low signal-to-noise ratio. The OOB error is more adaptive. Therefore, the OOB error is used to evaluate the importance of dimensions in our model. The OOB-based dimension importance measure is determined as follows: First, an RF is fit by applying the bootstrap aggregation (bagging) technique that repeatedly selects random samples with replacement and fits multiple trees based on these samples [27]. Then, to measure the importance of dimension X_i, in each tree, the OOB prediction error rate O_1 is calculated, the values of the dimension X_i are permuted among the training data, and the OOB error is computed again on the perturbed data set, namely O_2. Finally, the difference between O_1 and O_2 is calculated and normalized.

The difference on all TSs is calculated, and the average value is obtained, this value forms the importance score of X_i, which is denoted as $Vim_i^{(OOB)}$. Dimensions with larger values are ranked as more important than dimensions with smaller values. The $Vim_{ij}^{(OOB)}$ of dimension X_i in tree j can be calculated as follows:

$$Vim_{ij}^{(OOB)} = \frac{\sum_{p=1}^{n_o^j} I(Y_p = Y_p^j)}{n_o^j} - \frac{\sum_{p=1}^{n_o^j} I(Y_p = Y_{p,\pi}^j)}{n_o^j}, \tag{1}$$

where Y_p^j is the observed value of OOB in the j^{th} tree and $I(g)$ is the indicator function, which takes a value of 1 when the two values are equal and 0 when they are not equal. $Y_p \in \{0, 1\}$ is the result of the p^{th} observation, and $Y_{p,\pi}^j \in 0, 1$ is the predicted result of the p^{th} observation in the j^{th} tree after random replacement. When dimension Xi does not appear in the j^{th} tree, its importance is 0.

The importance of dimension X_i in the whole RF algorithm is calculated in (2), where n is the number of trees in RF.

$$Vim_i^{OOB} = \frac{\sum_{j=1}^{n} Vim_{ij}^{OOB}}{n}. \tag{2}$$

To calculate the dimension importance score for mTree dataset, the following steps are used.

Step 1. According to the size of the currently explored mTree data, the multidimensional data of each layer are merged to obtain a multidimensional set (MS).

Step 2. The dimensions in the MS are divided into a user set (US) and an evaluation set (ES). The US includes a user-focused dimension and has a size of one. The ES is the set of dimensions that are not selected by users. The aim of our model is to evaluate the importance of each dimension in the ES relative to the US. The higher the importance score is, the more significant the combination of it and the US is, and the more likely it can help users gain insights.

Step 3. The dimensions are ranked in descending order according to their importance scores. The top 3 important dimensions $\{I_1, I_2, I_3\}$ are returned.

Finally, the user focused dimension U and the top three related dimensions $\{I_1, I_2, I_3\}$ are chosen as the most important dimensions. Accordingly, the subset with dimensions $\{U, I_1, I_2, I_3\}$ of the selected branch or branches is returned as the important subspace.

4.2 LSTM-Based Subspace Visualization Recommendation Model

Selecting an appropriate visualization type for the important subspace is a complex task, and multifaceted information needs to be presented in a limited screen space. We propose an automated model to lower the threshold of this technique. Considering that the target users of our system have little visualization knowledge, we specifically choose a less complex visualization technique: small multiples. It encodes multidimensional information with multiple simple charts, and each chart encodes one facet. We select four chart types, including bar charts, pie charts, line charts, and scatter plots, which are the most commonly used chart types for exploring multidimensional data [28]. These charts can help users complete the most frequent tasks, such as cluster analysis, correlation analysis, and anomaly detection [29].

Based on the design requirement, the recommendation process focuses more on the data exploration width. Therefore, the chart style, such as its color options, is beyond the recommendation scope. The aim of the recommendation model is to determine the suitable chart type for each valid dimension or dimension combination. Therefore, we formulate the recommendation problem as a classification problem: choosing one chart type from the four available types.

For recommendation model selection, two main modeling types are available: the learning-to-rank and the classification models. A learning-to-rank model is trained to judge whether one visual encoding is better than another; examples include the lambdaMART model, and the RankSVM model. A classification model such as Neural Network (NN) model, is used to predict the possible design choice. Based on the state-of-the-art research in visualization recommender systems [12, 10, 11], the NN based classification models have better precision. Furthermore, the long short-term memory (LSTM) model, a recurrent neural network (RNN) model variant, can overcome the vanishing gradient problem of traditional RNNs, and has been widely adopted in visual analysis frameworks in recent years [30, 31]. Therefore, in this work, we choose to adapt the LSTM model to predict chart types. The comparative experiments in the evaluation section (Section 6.1) demonstrate that it has better performance than the baseline NN and RankSVM models in our scenario.

The recommended workflow is shown in Figure 2. It starts from the incoming important subspace and formats it as a 4-dimensional table (1), it computes all valid combinations containing one to three dimensions (2), and for each combination, it extracts features (3) and sends them to the Bi-LSTM model (4) to predict the appropriate chart type (5).

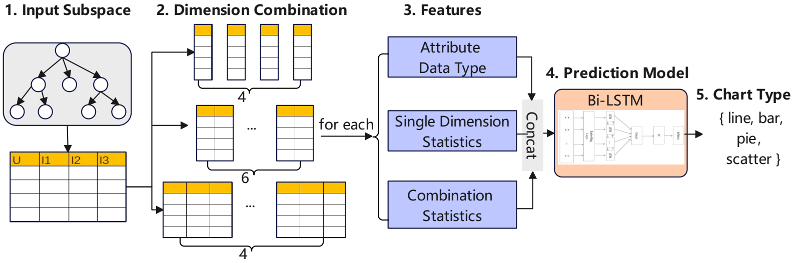


Figure 2. The workflow of visualization recommendation

The input of the recommendation model is a selected subspace with $\{U, I_1, I_2, I_3\}$, where U is the user selected dimension, and I_1, I_2, and I_3 are the top 3 important dimensions. The charts supported by our model can encode $1 - 3$ dimensions. Enumerating all possible cases with $1 - 3$ dimensions from $\{U, I_1, I_2, I_3\}$, we obtain $C_4^1 + C_4^1 + C_4^3 = 14$ combinations, some of which may not be valid. We obtain at most 14 valid dimension combinations. In turn, the visualization recommendation model will predict the most appropriate chart type for each valid combination.

Then, we need a way to convert those different characteristics into a multidimensional vector. Here, we refer to the approach in VizML [12] and the analysis in Table 1 and calculate the embedding vector of dimensions by feature engineering. The embedding vector consists of the type of dimension, the statistical characteristics of each single dimension (the total, mean, max, etc.), and the statistical characteristics of the dimension combinations.

Finally, the output layer uses the Softmax activation function to classify the input sequences and outputs the chart type with the highest probability.

4.3 Rule-Based Chart Encoding Model

The visualization recommendation model is only responsible for determining the chart type. We also need to determine how to map the $\{\text{dimension(s)}, \text{chartType}\}$ pair to a visual chart. For example, suppose that the input data contain two string-type dimensions $Mc1$ and $Mc2$, and that the recommended chart type is a bar chart. Then, a mapping rule is needed to determine which dimension is mapped to the X-axis and which is mapped to the Y-axis, as well as whether operations such as count and min are needed. In this example, the dimension with more categories should be mapped to the X-axis; suppose that this dimension is $Mc1$. Then, each value $Mc1_i$ in $Mc1$ is counted, and the percentage of each value $Mc2_i$ in the other dimension $Mc2$ is used as the color map of the bar chart. In addition, to avoid visual clustering, when the number of categories in $Mc2$ is greater than five, we select the four most frequent categories, and the rest of the categories are categorized as other.

We determine the rules with visualization expert interviews and refine the theme during practice. It would be better if a systematic study could be performed in the future. The mapping rules are developed and depended on the number of dimensions, the types of the dimensions and the chart characteristics. Many combinations of dimensions can be formed, and Table 1 lists only part of the encoding rules. In the table, S refers to the string-data type, N refers to the numeric type, and D refers to the temporal type.

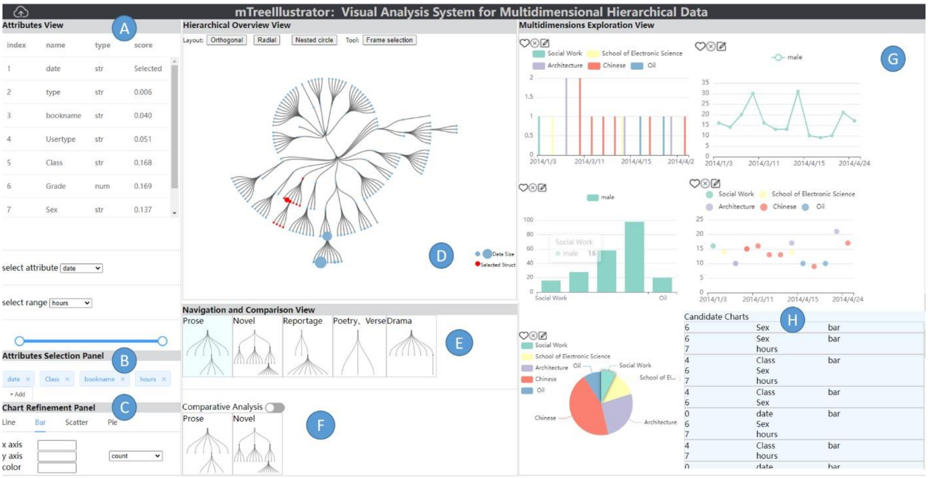
5 VISUAL DESIGN

We design an interactive visualization framework, mTreeIllustrator, to fulfill the design requirements mentioned in Section 3.1. mTreeIllustrator mainly consists of

Dimension(s)	**Recommended Chart**	**Rules**
$\{S\}$	Pie chart	Count and compute the percentage of each category
$\{S, N\}$	Bar chart	Encode S as the X-axis, sum N based on S_i
$\{D, S, N\}$	Line Chart	When D is more than the threshold, map D to the X-axis, map N to the Y-axis, and use S for coloring. When D is less than the threshold, map S to the X-axis, map N to the Y-axis, and use D for coloring.

Table 1. Chart encoding rules

eight components (Figure 3): control panel (Figure 3 A–C), a hierarchical overview (Figure 3 D), navigation and comparison views (Figure 3 E–F), and multidimensional exploration view (Figure 3 G–H). These views coordinate with each other to allow users to conduct deeper exploration and comparison with the inspiration provided by the recommended visualizations.



index	name	type	score
1	date	str	Selected
2	type	str	0.006
3	bookname	str	0.040
4	Usertype	str	0.051
5	Class	str	0.168
6	Grade	num	0.169
7	Sex	str	0.137

select attribute date

select range hours

x axis
y axis count
color

Candidate Charts

6	Sex	bar
6 7	Sex hours	bar
4 6 7	Class Sex hours	bar
4 6	Class Sex	bar
0 6 7	date Sex hours	bar
4 7	Class hours	bar
0	date	bar

Figure 3. The interface of mTreeIllustrator. The left part contains an attribute view (A), an attribute selection panel (B) and a chart refinement panel (C); the middle shows the hierarchical overview (D) and the navigation and comparison views (E, F); the right part contains multidimensional exploration views, including the top 5 recommendation charts (G) and candidate charts (H).

5.1 Hierarchical Overview

The mantra “overview first, zoom and filter, then details on demand” [32] has been widely used in the design of complex data exploration systems. Thus, we follow this mantra and put the hierarchical overview view (Figure 3 D) in the center and surround it with the detailed views.

The hierarchical overview presents the overall structure and distribution of the uploaded hierarchical data (R3). Considering the scale and topological variance of user data, the system provides three layout methods (the top-left buttons) to allow users to switch layouts to better display their data. Each layout method has its own advantages. The orthogonal node-link diagram performs better in presenting structures, but its spatial utilization is low, and it is not suitable for large data. The radial node-link layout has better spatial utilization, but its presentation of the tree depth is limited because its root node is fixed to the center of the circle. Therefore, it is suitable for presenting compact hierarchical data with a small depth. The circular treemap is not as intuitive as the node-link diagram, but it can encode more data elements within the same screen space. It also has advantages in terms of internode comparisons among large hierarchical data [33].

After becoming familiar with the overall information, users generally want to perform deeper exploration based on their analysis interests (R4). To support this requirement, a box selection button (the top-right button) is designed to allow users to select a node or a branch for deeper fine-grained exploration.

5.2 Multiple Dimensions Exploration View

As shown in Figure 3 G, the multiple dimensions exploration view visually presents the recommendation result from the automatic recommendation pipeline (R3). A series of small charts are generated by the visualization recommendation model, and each chart presents a facet of an important subspace. The chart order is sorted according to their importance scores. Based on the recommendation pipeline described in Section 4, we obtain at most 14 graphical charts. These charts are ranked based on whether they encode the user selected dimension (U1), the dimension count, and the dimension importance scores. To promote exploration broadness, we present multiple charts based on the current user selection. However, to avoid overwhelming users with too much information, we need to limit the charts counts. Following the “the seven plus or minus two” rule proposed by psychologist George Miller [34], human short-term memory can store only five to nine pieces of information, five for complex information, and nine for simple information. Therefore, to strike a balance, only the top five charts are shown. The other candidates are listed in a table next to the top charts (Figure 3 H). If users are interested in a candidate chart in the table, they can click on it. The system will display the chart.

5.3 Navigation and Comparison Views

During exploration, another challenge is that the exploration path may be long due to the structural complexity of hierarchical data. To lighten users' memory burdens, the exploration history view (R4-5) is designed to track users' exploration path so that the users can clearly see where they are and how they got there at any time. The branches or nodes that a user has visited during exploration are saved as thumbnails based on the access order. The most recently visited data are inserted from the left. Furthermore, users often need to perform comparisons between different hops of the exploration history, and the comparison view (Figure 3 F) is designed to allow users to select a comparison target to compare (R5). By clicking on the history thumbnail or by directly selecting a branch from the hierarchical overview, a comparison target is selected. With the target, the back end of our system temporarily generates a classification dimension and treats it as a user-selected dimension. Then, the recommendation pipeline automatically generates the most relevant dimensions regarding this target dimension and refreshes the top charts in the multidimensional exploration view. This process can help users efficiently complete the multidimensional substructure comparison.

5.4 Control Panel

The control panels are designed to support users' deeper analyse and free exploration (R3-4), and they mainly consist of the dimension view (Figure 3 A), a dimension selection panel (Figure 3 B) and a chart refinement panel (Figure 3 C). Please note that in the UI design, the term "Attribute" indicates the "Dimension" in the recommendation pipeline. We use this term because it is easier for target users to understand. The attribute view presents the attribute name, attribute category, and importance score in the current exploration. The importance score is calculated by the subspace importance assessment model based on the user-selected attribute. In the attribute selection panel, users can choose attributes, and then the system passes the user selection to the recommendation pipeline. The goal of the chart refinement panel is to allow users to modify and refine the recommended charts (R4). Inspired by the design of Voyager, the panel mainly provides three functions: the chart type selection, data operation selection and visual coding. According to the recommendation pipeline, scatter plots, line charts, bar charts, and pie charts are supported. Tha data operations refer to the max, min, count, sum, and range calculations. For example, if the final chart is a scatter plot, the user can perform data operation on the y-axis. If the max operation is selected, the y-axis encodes the maximum of the data. The visual encoding editor supports the user in changing the element colors and sizes. Users can change the encoding setting according to their preferences. To edit a recommended chart in Figure 3 G, the user clicks on the chart, and then the system automatically loads the configuration options of that chart into the chart editor.

5.5 Interaction Design

Rich interactions are provided in the mTreeIllustrator user interface to facilitate the mixed-initiative data exploration. As shown in Figure 3, users can click to select their branch of interest in the mTree Overview chart (Figure 3 D), or set their desired attribute in the attribute view (Figure 3 A). Accordingly, the recommendation pipeline is automatically triggered to calculate the top attributes and visualizations related to the latest user selection, and then update the attribute table view (Figure 3 A) and the multidimensional exploration view (Figure 3 E). In addition, the system allows users to refine the system recommendation. They can add or delete the recommended attributes (Figure 3 B), change the chart encoding (Figure 3 C), or zoom out the candidate charts (Figure 3 H).

5.6 Scalability Consideration

For scalability, the current mTreeIllustrator design is targeted for moderate-size data that can be rendered in acceptable time and fit into the available screen size, i.e., thousands of data items. In exploration cases with larger data sizes, techniques such as Level-of-Detail (LoD) rendering may be extended from our framework.

6 EVALUATION

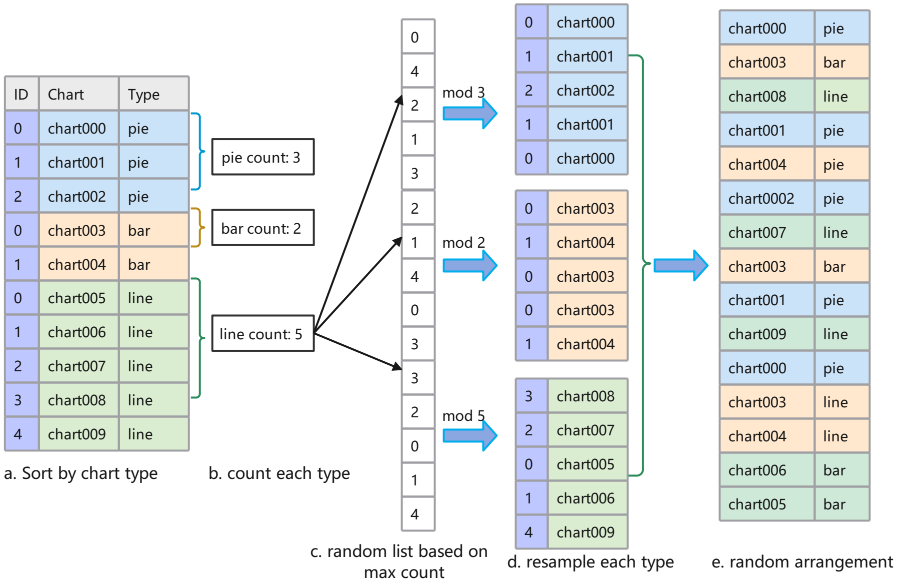
To verify the effectiveness and usability of the visualization framework proposed in this paper, we performed both performance evaluations and user studies.

6.1 Model Performance Evaluations

We conduct a comparative experiment to evaluate the performance of our model.

Data: The experimental dataset is derived from a subset of the VizML corpus which includes data and visual chart mapping pairs published by Plotly community users. After performing data cleaning, the valid dataset consists of 31 829 scatter plots, 12 002 bar charts, 23 702 line charts and 3 144 pie charts. For model training, the dataset is split into training/validation/test sets with a ratio of 60/20/20. The chart type distribution of this dataset is imbalanced, which may cause the prediction to be inclined toward the class with more samples and affect the generalization ability of the model. To prevent this problem, the class reorganization method [35] is used to balance the training dataset. This method needs to be repeated before each training step. The procedure is shown in Figure 4.

First, the original samples are classified and arranged by the chart type. Suppose that chart type M has the maximum number of samples. A random



a. Sort by chart type

b. count each type

c. random list based on max count

d. resample each type

e. random arrangement

Figure 4. Procedure for balancing the training dataset

list L is generated for each class based on the count of M, and the random number in L is used to balance the number of samples in each class to obtain the corresponding index. Then, a random chart list (CLs) are generated by extracting charts from each class according to the index. All CLs are concatenated and randomly placed to obtain the last chart list (LCL). Now, the samples in each class in the LCL are equal. The advantage of this method is that it does not require extra information and can be run automatically.

Environment and Configuration: Our model is implemented with Python version 3.7 and PyTorch framework version 1.7.1 on a Windows desktop (Intel Core@2.30 GHz CPU with 12 GB of memory). The initial learning rate is set to $5 \times 10 - 4$, and the loss is reduced by a factor of 10 if the loss plateau is encountered; otherwise, the reduction is triggered every 5 interactions. The dropout rate is set to 0.5, the batch size is set to 128, and 100 epochs are run to train the model.

Procedure: We select three models which are used in the recent visualization recommendation systems as comparison, namely a support vector machine (SVM), a neural network (NN), and the RankSVM model. RankSVM is the model used by the DeepEyes visualization recommendation, and NN is used in the VizML and LQ2 tools. Among them, the NN has the best performance and is used as the baseline model. The evaluation metrics are as follows: the accuracy (Acc) is calculated in (3); the precision (Pre) is calculated in (4); the recall (Rec) is calculated in (5); and the F1 score is calculated in (6). The metrics are computed based on the confusion matrix that counts the correct and incorrect prediction counts: TPs (true positive), FPs (false positive), TNs (true negative), and FNs (false negative), where $TP + FP + TN + FN =$ the total samples. The details are as follows:

$$Acc = \frac{TP + TN}{TP + TN + FP + FN}, \tag{3}$$

$$Pre = \frac{TP}{TP + FP}, \tag{4}$$

$$Rec = \frac{TP}{TP + FN}. \tag{5}$$

Results: The experimental results are given in Table 2. The evaluation metrics, Acc, Rec, Pre and F1 of our model are above 93.9 %, which is better than those of the other models. The results in the table show that the NN model outperforms the SVM model in terms of accuracy and F1 scores. Among the NN models, the recurrent RNN-based model (Ours) is slightly better than the baseline NN model, which may be because the RNN model better captures the data features during training and requires fewer samples.

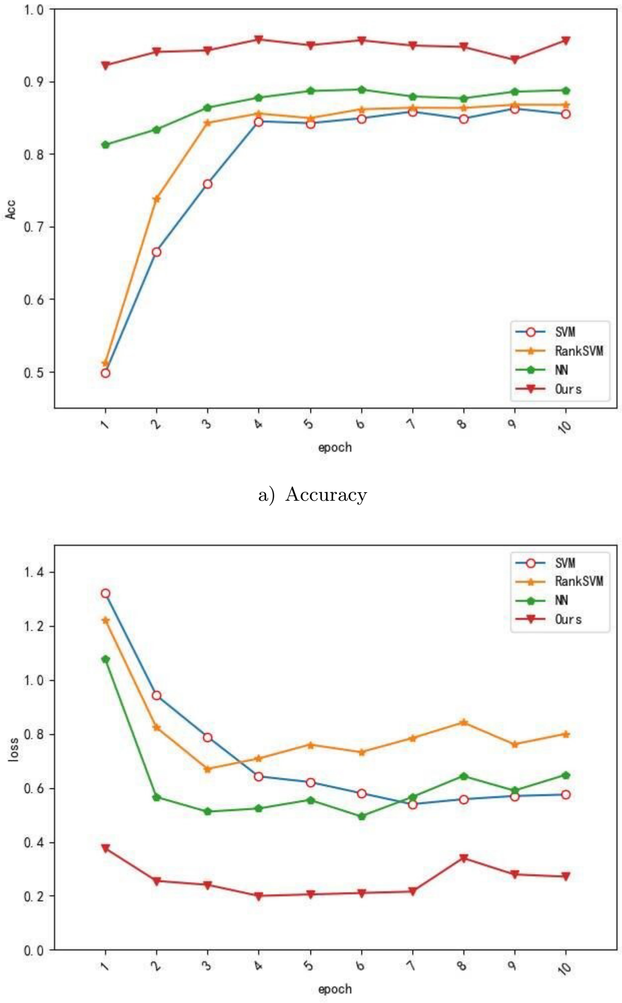
Model	**Acc**	**Rec**	**Pre**	**F1**
SVM	0.851	0.841	0.832	0.836
RankSVM	0.861	0.842	0.835	0.838
NN	0.881	0.874	0.863	0.868
Ours	0.949	0.946	0.939	0.942

Table 2. Performance metric distribution of the four models

Ten epochs are run to compare the training time changes exhibited by the models, the accuracy (Acc) and loss (Loss) values are assessed for each run, and the results are plotted in Figure 5.

Figure 5 a) shows that the accuracies of all models first increase with increasing epochs and then stabilize. In the first 4 epochs, the nonneural models fluctuate considerably. During the stable phase, the accuracies of all models exceed 84 %, and the accuracy values of the two NN models are greater than 90 %. However, our model can reach 90 % accuracy with fewer epochs, so it has a good classification ability in a shorter training time.

As shown in Figure 5 b), the loss rate of our model is in the range of 0.2 % to 0.4 %, which is lower than that of the comparative models, and indicates that the convergence of the proposed model is better. The loss rate fluctuates once, which may be caused by sudden changes in some unknown factors, but it does not affect the overall trend. Overall, our model outperforms the comparative model in terms of convergence speed and accuracy.



a) Accuracy

b) Loss

Figure 5. Prediction distributions of the four models

6.2 User Study

We conducted user studies to evaluate the effectiveness of our visual analysis framework. Procedure and Participants: Three visualization experts and scholars were invited to discuss the evaluation metrics. Each expert had more than 5 years of experience with visualization. After discussion, the practicality, explanation, effectiveness, readability and usability metrics were selected. Based on these five metrics and the analysis objectives of this paper, user studies were designed.

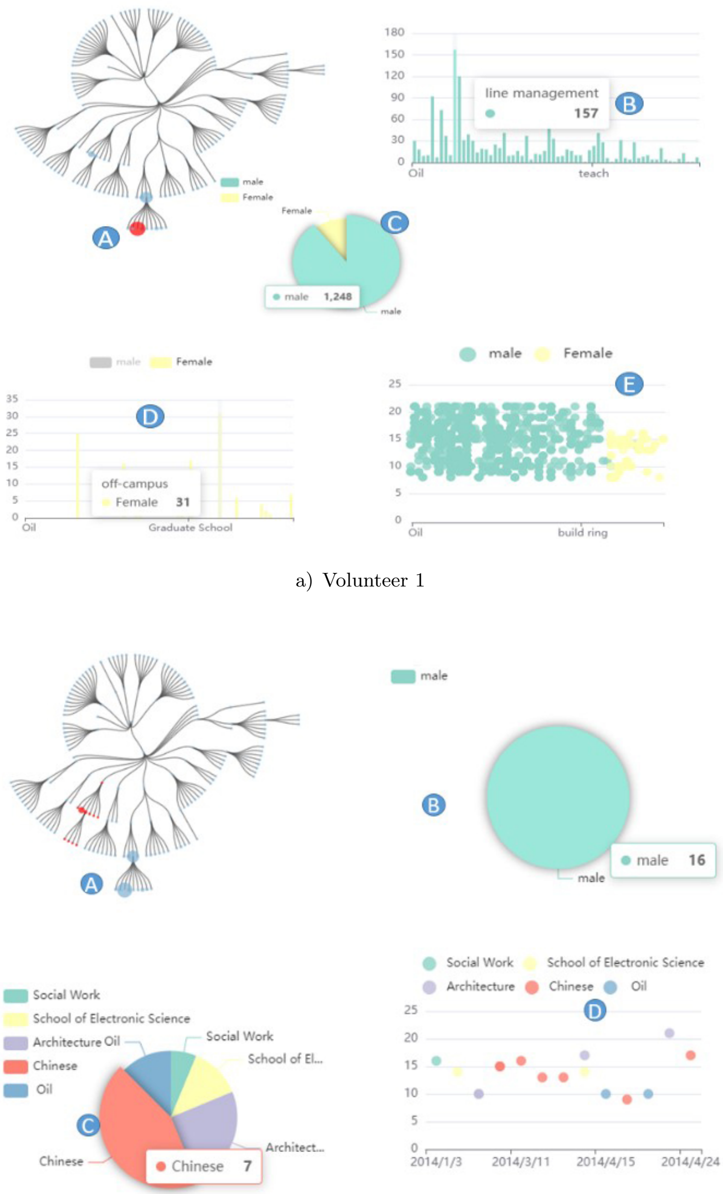
We conducted the study with ten volunteers from our school. The age range was 19 to 25 years, there were 7 males, 3 females, 6 undergraduates, and 4 graduates. Three of them had one year of experience in visualization, and the other had little knowledge of visualization. We selected a dataset that was familiar to the volunteers, namely, the book borrowing records dataset of a university library[1] and a literature books subset. First, we introduced the background, the data and the analysis task and then demonstrated the use of the mTreeIllustrator system. Subsequently, after a Q & A, the volunteers started their exploration. During their explorations, they were asked to record details they found meaningful or interesting.

Result and Analysis: Based on the exploration records, we interviewed the users; two representative use cases are shown in Figure 6 a) and Figure 6 b).

Volunteer 1's attention was first drawn to the hierarchical overview, where he found that the most popular books were romance novels, as shown in Figure 6 a) part A. The volunteer then wanted to know which majors contributed the most. He clicked on the node representing romance novels in the hierarchical overview, and the system automatically updated the overview view with only the romance novel data and generated top visualizations for the important attribute combinations in the multidimensional exploration view, from which the volunteer found the histogram of borrowing statistics for each major (B). Students in the "Administration" major contributed the most, followed by "Storage and Transportation" and "Financial Management". Then, he wanted to know the gender distribution, but the gender attribute was not selected by the recommendation pipeline. Therefore, he manually added that attribute, and the system regenerated top the charts according to the new selection. The volunteer first looked at the recommended pie chart (C) showing the percentages of male and female borrowers and found that the proportion of men was much larger than that of women. This phenomenon was unexpected; he thought that females would be the main readers of romance novels, but the proportion of males was much larger (D–E) in this dataset. The volunteer thought that was an interesting finding. Overall, the volunteer thought that the system could help users efficiently understand the characteristics of borrowing patterns.

Volunteer 2 focused on the prose branch on the hierarchical overview, as shown in Figure 6 b) part A, and added the gender attribute (B) to the currently explored attribute set. From the attribute exploration view, she found that the readers were mainly from "Architecture", "College of Electronic Science", and "Social Work" majors (C), and their borrowing dates were mainly March 2014 (D). The background of the readers shows that readers may be less interested in books in the prose category. Readers from the "Chinese" major contributed the most. This may be related to their course study needs.

[1] `https://github.com/wenbl/LibraryBigData/tree/master/data`



a) Volunteer 1

b) Volunteer 2

Figure 6. Exploration paths

The exploration results from the two volunteers illustrate that the automatic pipeline and the visual analysis system can help users quickly become familiar with mTree data and can also support efficient fine-grained exploration.

Usability Evaluation: After the users finished the experiment, they were asked to complete a questionnaire to evaluate the efficiency of the system. The assessment data are quantified using a five-point Likert scale, as shown in Figure 7.

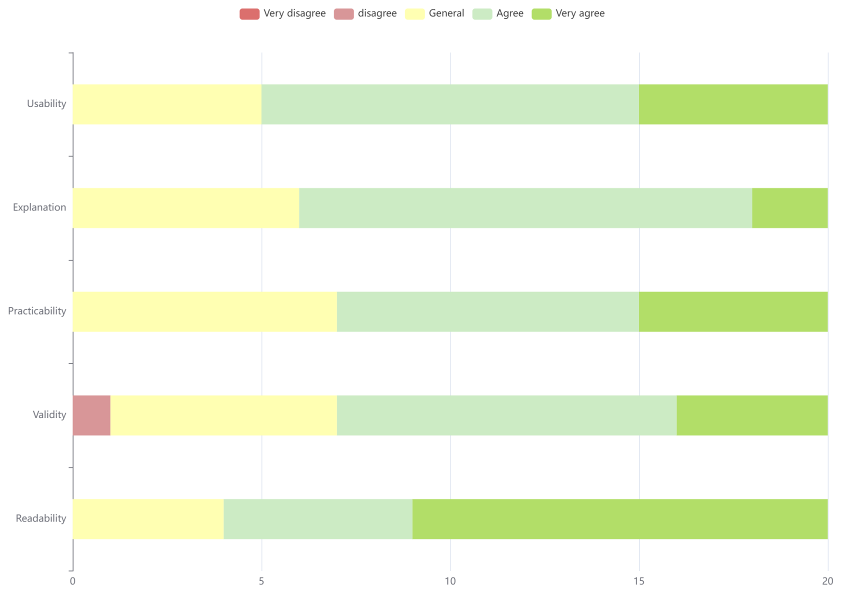


Figure 7. Score distribution of the questionnaire

Most volunteers agreed that mTreeIllustrator is useful, easy to learn, and easy to use. Each volunteer learned to use the tool quickly. When they were asked to compare their experience with that of previous library data exploration tools, they were all more in favor of this visual tool, saying the charts were easier to understand than the abstract data. Volunteer 3 said she especially liked the small charts in the left panel since they provided insight for further exploration. Among all metrics, the validity metric was slightly weaker than those of the other metrics. We interviewed the volunteers and found that the main reason for this score was that the attributes that the user wants to explore were occasionally not included in the recommendation list. For example, volunteer 1 manually added the gender attribute. This is a valuable finding; our model does not consider user differences, but in reality, people from different backgrounds do have different preferences. In the future, personalized learning algorithms would be studied.

7 CONCLUSION

In this paper, we presented mTreeIllustrator, a mixed-initiative framework for visual and interactive mTree data exploration. We proposed a machine learning-powered pipeline, consisting of an RF-based subspace importance evaluation model, a Bi-LSTM based visualization recommendation model and a rule-based chart encoding model, to automatically select the most important subspace from mTree data and encode the subspace into faceted visualizations. Moreover, we designed a visual framework and an interaction schema to couple the autogenerated visualizations with user selections to support progressive mTree data exploration. This approach also allows users to refine both the recommended visualizations and the data subspace, and to visually compare selected mTree structures. Comparative experiments and user studies demonstrated that our framework has good performance and can enable users to perform efficient and insightful mTree data exploration.

In the future, we plan to expand the range of our recommendation models. First, our model is purely data driven, and we plan to also consider the personal preferences and analysis goals to enable more diversified analyse. Another interesting area would involve studying the user interaction patterns exhibited during the mTree data exploration process and to developing models for providing navigation suggestions.

Acknowledgements

This research is jointly supported by the National Natural Science Foundation of China (No. 61872304) and the Talent Project of Sichuan University of Science and Engineering (No. 2020RC20).

REFERENCES

[1] Li, Y.: Hierarchical Data Visualization and Visual Comparison. Master Thesis. Shanghai Jiaotong University, Shanghai, 2016 (in Chinese).

[2] Chen, Y.—Zhen, Y. G.—Hu, H. Y.—Liang, J.—Ma, K. L.: Visualization Technique for Multi-Attribute in Hierarchical Structure. Journal of Software, Vol. 27, 2016, No. 5, pp. 1091–1102, doi: 10.13328/j.cnki.jos.004956 (in Chinese).

[3] Guerra-Gomez, J. A.—Pack, M. L.—Plaisant, C.—Shneiderman, B.: Visualizing Change over Time Using Dynamic Hierarchies: TreeVersity2 and the StemView. IEEE Transactions on Visualization and Computer Graphics, Vol. 19, 2013, No. 12, pp. 2566–2575, doi: 10.1109/TVCG.2013.231.

[4] Wongsuphasawat, K.—Qu, Z.—Moritz, D.—Chang, R.—Ouk, F.—Anand, A.—Mackinlay, J.—Howe, B.—Heer, J.: Voyager 2: Augmenting Visual Analysis with Partial View Specifications. Proceedings of the 2017 CHI Con-

ference on Human Factors in Computing Systems (CHI '17), 2017, pp. 2648–2659, doi: 10.1145/3025453.3025768.

[5] Inselberg, A.: Parallel Coordinates: Visual Multidimensional Geometry and Its Applications. Springer, New York, 2009, doi: 10.1007/978-0-387-68628-8.

[6] Schulz, H. J.: Treevis.net: A Tree Visualization Reference. IEEE Computer Graphics and Applications, Vol. 31, 2011, No. 6, pp. 11–15, doi: 10.1109/MCG.2011.103.

[7] Macquisten, A.—Smith, A. M.—Johansson Fernstad, S.: Evaluation of Hierarchical Visualization for Large and Small Hierarchies. 2020 24th International Conference Information Visualisation (IV), 2020, pp. 166–173, doi: 10.1109/IV51561.2020.00036.

[8] Li, G.—Tian, M.—Xu, Q.—McGuffin, M. J.—Yuan, X.: GoTree: A Grammar of Tree Visualizations. Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20), 2020, pp. 1–13, doi: 10.1145/3313831.3376297.

[9] Wongsuphasawat, K.—Moritz, D.—Anand, A.—Mackinlay, J.—Howe, B.—Heer, J.: Voyager: Exploratory Analysis via Faceted Browsing of Visualization Recommendations. IEEE Transactions on Visualization and Computer Graphics, Vol. 22, 2016, No. 1, pp. 649–658, doi: 10.1109/TVCG.2015.2467191.

[10] Moritz, D.—Wang, C.—Nelson, G. L.—Lin, H.—Smith, A. M.—Howe, B.—Heer, J.: Formalizing Visualization Design Knowledge as Constraints: Actionable and Extensible Models in Draco. IEEE Transactions on Visualization and Computer Graphics, Vol. 25, 2019, No. 1, pp. 438–448, doi: 10.1109/TVCG.2018.2865240.

[11] Qin, X.—Luo, Y.—Tang, N.—Li, G.: DeepEye: An Automatic Big Data Visualization Framework. Big Data Mining and Analytics, Vol. 1, 2018, No. 1, pp. 75–82, doi: 10.26599/BDMA.2018.9020007.

[12] Hu, K.—Bakker, M. A.—Li, S.—Kraska, T.—Hidalgo, C.: VizML: A Machine Learning Approach to Visualization Recommendation. Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19), 2019, doi: 10.1145/3290605.3300358.

[13] Pister, A.—Buono, P.—Fekete, J. D.—Plaisant, C.—Valdivia, P.: Integrating Prior Knowledge in Mixed-Initiative Social Network Clustering. IEEE Transactions on Visualization and Computer Graphics, Vol. 27, 2021, No. 2, pp. 1775–1785, doi: 10.1109/TVCG.2020.3030347.

[14] Horvitz, E.: Principles of Mixed-Initiative User Interfaces. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '99), 1999, pp. 159–166, doi: 10.1145/302979.303030.

[15] Liu, S.—Maljovec, D.—Wang, B.—Bremer, P. T.—Pascucci, V.: Visualizing High-Dimensional Data: Advances in the Past Decade. IEEE Transactions on Visualization and Computer Graphics, Vol. 23, 2017, No. 3, pp. 1249–1268, doi: 10.1109/TVCG.2016.2640960.

[16] Gomez, J. A. G.—Buck-Coleman, A.—Plaisant, C.—Shneiderman, B.: TreeVersity: Comparing Tree Structures by Topology and Node's Attributes Differences. 2011 IEEE Conference on Visual Analytics Science and Technology (VAST), 2011, pp. 275–276, doi: 10.1109/VAST.2011.6102471.

[17] Chen, Y.—Dong, Y.—Sun, Y.—Liang, J.: A Multi-Comparable Visual Analytic Approach for Complex Hierarchical Data. Journal of Visual Languages and Computing, Vol. 47, 2018, pp. 19–30, doi: 10.1016/j.jvlc.2018.02.003.

[18] Sakairi, T.—Ishida, A.—Achilles, H. D.: Visual Analysis Tool for Hierarchical Additive Time-Series Data. 2015 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI), 2015, pp. 18–23, doi: 10.1109/SOLI.2015.7367610.

[19] Zhou, J.—Li, Z.—Zhang, Z.—Liang, B.—Chen, F.: Visual Analytics of Relations of Multi-Attributes in Big Infrastructure Data. 2016 Big Data Visual Analytics (BDVA), 2016, pp. 1–2, doi: 10.1109/BDVA.2016.7787052.

[20] Mackinlay, J.: Automating the Design of Graphical Presentations of Relational Information. ACM Transactions on Graphics, Vol. 5, 1986, No. 2, pp. 110–141, doi: 10.1145/22949.22950.

[21] Roth, S. F.—Kolojejchick, J.—Mattis, J.—Goldstein, J.: Interactive Graphic Design Using Automatic Presentation Knowledge. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '94), 1994, pp. 112–117, doi: 10.1145/191666.191719.

[22] Mackinlay, J.—Hanrahan, P.—Stolte, C.: Show Me: Automatic Presentation for Visual Analysis. IEEE Transactions on Visualization and Computer Graphics, Vol. 13, 2007, No. 6, pp. 1137–1144, doi: 10.1109/TVCG.2007.70594.

[23] Nan, M.—Xiaoru, Y.: Tabular Data Visualization Interactive Construction for Analysis Tasks. Journal of Computer-Aided Design and Computer Graphics, Vol. 32, 2020, No. 10, pp. 1628–1636 (in Chinese).

[24] Dibia, V.—Demiralp, C.: Data2Vis: Automatic Generation of Data Visualizations Using Sequence-to-Sequence Recurrent Neural Networks. IEEE Computer Graphics and Applications, Vol. 39, 2019, No. 5, pp. 33–46, doi: 10.1109/MCG.2019.2924636.

[25] Satyanarayan, A.—Moritz, D.—Wongsuphasawat, K.—Heer, J.: Vega-Lite: A Grammar of Interactive Graphics. IEEE Transactions on Visualization and Computer Graphics, Vol. 23, 2017, No. 1, pp. 341–350, doi: 10.1109/TVCG.2016.2599030.

[26] Cui, W.—Zhang, X.—Wang, Y.—Huang, H.—Chen, B.—Fang, L.—Zhang, H.—Lou, J. G.—Zhang, D.: Text-to-Viz: Automatic Generation of Infographics from Proportion-Related Natural Language Statements. IEEE Transactions on Visualization and Computer Graphics, Vol. 26, 2020, No. 1, pp. 906–916, doi: 10.1109/TVCG.2019.2934785.

[27] Breiman, L.: Random Forests. Machine Learning, Vol. 45, 2001, No. 1, pp. 5–32, doi: 10.1023/A:1010933404324.

[28] Battle, L.—Duan, P.—Miranda, Z.—Mukusheva, D.—Chang, R.—Stonebraker, M.: Beagle: Automated Extraction and Interpretation of Visualizations from the Web. Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18), 2018, doi: 10.1145/3173574.3174168.

[29] Saket, B.—Endert, A.—Demiralp, C.: Task-Based Effectiveness of Basic Visualizations. IEEE Transactions on Visualization and Computer Graphics, Vol. 25, 2019, No. 7, pp. 2505–2512, doi: 10.1109/TVCG.2018.2829750.

[30] Chen, L.—Yang, D.—Zhang, D.—Wang, C.—Li, J.—Nguyen, T. M. T.: Deep Mobile Traffic Forecast and Complementary Base Station Clustering for C-RAN Optimization. Journal of Network and Computer Applications, Vol. 121, 2018, pp. 59–69, doi: 10.1016/j.jnca.2018.07.015.

[31] Lee, C.—Kim, Y.—Jin, S.—Kim, D.—Maciejewski, R.—Ebert, D.—Ko, S.: A Visual Analytics System for Exploring, Monitoring, and Forecasting Road Traffic Congestion. IEEE Transactions on Visualization and Computer Graphics, Vol. 26, 2020, No. 11, pp. 3133–3146, doi: 10.1109/TVCG.2019.2922597.

[32] Shneiderman, B.: The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. Proceedings 1996 IEEE Symposium on Visual Languages, 1996, pp. 336–343, doi: 10.1109/VL.1996.545307.

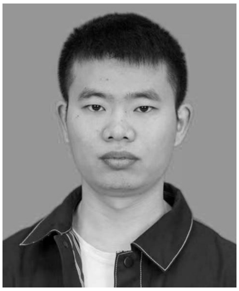
[33] Zhou, M.—Tao, W.—Pengxin, J.—Shi, H.—Dongmei, Z.: Table2Analysis: Modeling and Recommendation of Common Analysis Patterns for Multi-Dimensional Data. Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, 2020, pp. 320–328, doi: 10.1609/aaai.v34i01.5366.

[34] Miller, G. A.: The Magical Number Seven Plus or Minus Two: Some Limits on Our Capacity for Processing Information. Psychological Review, Vol. 101, 1956, No. 2, pp. 343–352, doi: 10.1037/0033-295X.101.2.343.

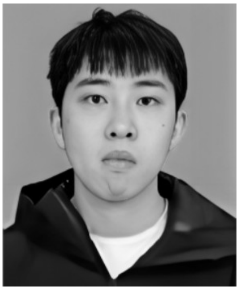
[35] Wei, X.: Analytical Deep Learning: Convolution Neural Network Principles and Visual Practice. Electronic Industry Press, 2018 (in Chinese).

Guijuan Wang received the M.Sc. degree from Beihang University, Beijing, China in 2007. She is currently pursuing the Ph.D. degree in the School of Information and Technology of the Southwest University of Science and Technology. Her research interests include intelligent city visualization and automatic visualization.

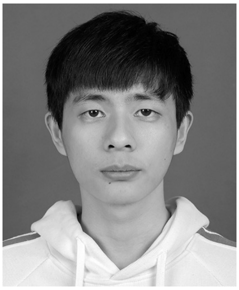
Yu Zhao received his Ph.D. degree from the Brunel University London, UK in 2014. His research interests include cloud computing, data analytics and artificial intelligence. He is currently Assistant Research Scientist at the Shandong Academy of Social Sciences.



Boyou Tan received his M.Sc. degree from the Southwest University of Science and Technology, Mianyang, China, in 2022. His major research interests include information visualization and mobility pattern.



Zhong Wang received his B.Sc. degree from the Southwest University of Science and Technology, Mianyang, China, in 2019. He is currently pursuing his M.Sc. degree in the School of Computer Science and Technology of the same university. His research interests include information visualization and mobility pattern.



Jiansong Wang received his B.Eng. degrees in computer science and technology from the Southeast University, Chengxian College in 2020. He is currently Master student at the Southwest University of Science and Technology. His research interests are data visualization and digital twin.

Hao Guo is M.Sc. student at the Southwest University of science and technology. He is interested in data visualization and natural language processing.

Yadong Wu is Dean at the School of Computer Science and Engineering, Sichuan University of Science and Engineering, Zigong China. He finished his Ph.D. at the University of Electronic Science and Technology of China. His current research interests include visualization, virtual reality and digital twins technology.