ChronoDeck: A Visual Analytics Approach for Hierarchical Time Series Analysis

Lingyu Meng, Shuhan Liu, Keyi Yang, Jiabin Xu, Zikun Deng, Di Weng, Yingcai Wu, *Senior Member, IEEE*

***Abstract*—Hierarchical time series data comprises a collection of time series aggregated at multiple levels based on categorical, geographical, or physical constraints, the analysis of which aids analysts across various domains like retail, finance, and energy, in gaining valuable insights and making informed decisions. However, existing interactive exploratory analysis approaches for hierarchical time series data fall short in analyzing time series across different aggregation levels and supporting more complex analytical tasks beyond common ones like summarize and compare. These limitations motivate us to develop a new visual analytics approach. We first generalize a taxonomy to delineate various tasks in hierarchical time series analysis, derived from literature survey and expert interviews. Based on this taxonomy, we develop ChronoDeck, an interactive system that incorporates a multi-column hierarchical time series visualization for implementing various analytical tasks and distilling insights from the data. ChronoDeck visualizes each aggregation level of hierarchical time series with a combination of coordinated dimensionality reduction and small multiples visualizations, alongside interactions including highlight, align, filter, and select, assisting users in the visualization, comparison, and transformation of hierarchical time series, as well as identifying the entities of interest. The effectiveness of ChronoDeck is demonstrated by case studies on three real-world datasets and expert interviews.**

***Index Terms*—Hierarchical time series visual analysis, time series visualization, multi-level analysis**

I. Introduction

Hierarchical time series analysis refers to the analysis of a set of time series data organized across different levels of aggregation, reflecting various categorical, geographical, or structural dimensions [1]. Such analysis is widely applied in a number of sectors like retail, finance, and energy. For instance, a retail dataset might be structured hierarchically, with data aggregated at the levels of state, region, individual store, and product category. At each hierarchical level, multiple time series capture the changes in sales figures over time.

The significance of analyzing hierarchical time series lies in its ability to provide a nuanced and layered understanding of the data and facilitate informed decision-making. For instance, in the retail sector, a business analyst may examine the sales of several stores in the same region and determine if the sales trends of these store aligns with the regional patterns. Such insights are valuable for retail companies to optimize their sales strategies and supply chain management.

Visual analytics serves as an intuitive approach to facilitate the analysis of hierarchical time series data through interactive interfaces. The prior studies involve such analysis can be categorized into two types: level-by-level and multi-level analysis. The level-by-level analysis [2]–[5] provides the analytical support for analyzing time series in the same aggregation level or leaf nodes, and allows users to subsequently drill down or roll up in the hierarchy to explore time series data in other levels. This method performs well in identifying co-occurrence patterns or anomalies among multiple time series while falls short in analyzing time series in different aggregation levels, which we argue is a crucial aspect of hierarchical time series analysis. The multi-level analysis [6]–[14] supports both within-level analysis and cross-level analysis by visualizing time series in multiple aggregation levels simultaneously. Nevertheless, current visualization mainly focus on limited tasks like comparing multiple time series, which is not sufficient to satisfy diverse goals during hierarchical time series analysis.

Above limitations of existing techniques motivate us to develop a new interactive approach, with which users are able to combine novel hierarchical time series visualization and interactions to perform diverse analytical tasks on both time series in one single level and the ones distributed at different levels. Three challenges occur in developing such a tool:

Delineating diverse analytical tasks of hierarchical time series analysis. The scope of hierarchical time series analysis is still unclear, existing works mainly focus on one specific task of the analysis, such as comparing multiple time series in the same aggregation level to identify anomalous time series. To implement an in-depth analysis of hierarchical time series, a well-structured task summary is critical to steer the whole analytical process.

Visualizing the integration of hierarchical structure and time series data. Analyzing hierarchical time series usually involves multiple time series within or across different hierarchical levels. It is crucial to represent the features of many time series effectively while maintaining awareness of hierarchical contexts such as parent-child or sibling relationships throughout the analytical process. Therefore, a novel and scalable visualization approach is in need to present both aspects of hierarchical time series.

Supporting diverse hierarchy-aware analytical tasks on many time series. Different analytics tasks involves targets with varied scales, ranging from single time series to multiple sub-hierarchies. For example, business analysts would not only

L. Meng, S. Liu, J. Xu, and Y. Wu are with the State Key Lab of CAD&CG, Zhejiang University, Hangzhou, China. E-mail: {kevinmeng, shliu, jiabin_xu, ycwu}@zju.edu.cn.

D. Weng is with the School of Software Technology, Zhejiang University, Ningbo, China. E-mail: dweng@zju.edu.cn. D. Weng is the corresponding author.

K. Yang is with the School of Earth Sciences, Zhejiang University, Hangzhou, China. E-mail: nemoyang@zju.edu.cn.

Z. Deng is with the School of Software Engineering, South China University of Technology, Guangdong, China. E-mail: zkdeng@scut.edu.cn.

Manuscript received xxx. xx, 2024; revised xxx. xx, 2024.

inspect one store's sales trend, but also compare the sales number of multiple regions and their subordinated stores. Moreover, analysts may transform the hierarchy by grouping similar time series, enabling more collective analysis. Therefore, a combination of adaptive visualization and interaction methods is needed to accommodate different scenarios.

We propose ChronoDeck, an interactive visual analytics system to address these challenges. For the first challenge, we conduct literature survey and interviews with domain experts, which results in a taxonomy of analytical tasks for hierarchical time series. Our proposed taxonomy characterizes six analytical tasks including summarize, compare, relate, compute, rearrange, and reshape, which can be performed upon four categories of target entities during the analysis: node, layer, path, and tree. For the second challenge, ChronoDeck leverages a close coordination of dimensionality reduction and small multiples visualizations [15] for users to visualize hierarchical time series in a multi-column layout, presenting both an overview and details of multiple time series. Meanwhile, linking corresponding nodes and time series visualization in adjacent columns make it effortless for users to understand the hierarchical context. Finally, for the third challenge, ChronoDeck incorporates adaptive visualization and various interaction methods including highlight, align, filter, and select to accommodate diverse analytics tasks. Our contributions can be summarized as follows:

- A task taxonomy for hierarchical time series analysis.
- An interactive visual analytics system, ChronoDeck, which combines multi-column visualization and various interactions to enable the integrated analysis of hierarchical time series.
- Case studies on three real-world datasets, complemented by expert interviews, to demonstrate the effectiveness of our method.

II. Related Work

This section presents relevant studies on time series visualization, hierarchy visualization, and visual analytics for hierarchical time series.

A. Time Series Visualization

Several surveys are conducted on the visualization of time series. Müller and Schumann [16] categorize existing visualizations into static and dynamic representations. Fang et al. [17] classify time attribute visualization methods into: spiral diagram, calendar view, theme river view, dynamic visualization, and others. Aigner et al. [18] propose a simplified schema of current visualization techniques based on time, data, and visual representations. We follow the categorization of static and dynamic representations to organize previous work.

Static representation. Most static visualization techniques maintain a fixed representation. The line plot is the most common and most frequently adopted visualization in many visual analytics systems [19]–[23]. There are also variations of the line plot such as spline chart and area chart [24].

Another important technique is calendar-based, which is first proposed by Wijk and Selow [25]. In this work, a combination of calendar visualization and cluster analysis method is introduced to explore time series at different granularities. Xu et al. [26] adopt the similar approach to summarize attributes such as the number of faults that arise over time, and facilitate further exploration. Also, to characterize the seasonality of time series, the spiral diagram [27] is more often used to visualize data in a circular structure. Notably, Tominski and Schumann [28] integrate two-tone pseudo coloring in the spiral diagram to extract cyclic patterns in human infection data. Besides, glyph-based approaches [29], [30] are employed to summarize the characteristics of time series.

In addition, stacked graph or streamgraph are used to visualize multiple time series. Systems such as TIARA [31] utilize this method to analyze the evolution of large textual data. Meanwhile, to meet higher demands on the scalability of the visualization, horizon chart [32], [33] allows users to identify peaks and co-occurrence patterns within a limited space, as seen in systems like CloudDet [34] and FMLens [35].

Dynamic representation. This representation presents time series data through a sequence of visualizations which are considered as frames. For instance, Moere [36] utilizes a collection of boids to visualize time-varying data like live stock market feeds. Another typical example is Gapminder Tools [37], which presents animated scatter plots on a Cartesian coordinate system, allowing users to explore data in each year by dragging a time slider.

Above methods primarily focus on presenting the features of individual or multiple time series, which do not take the structure of hierarchical time series into consideration. To address this limitation, ChronoDeck integrates time series visualization into a hierarchical representation, depicting temporal characteristics across multiple levels of the hierarchy.

B. Hierarchy Visualization

There are a number of studies which propose various representations for hierarchy visualization. Schulz [38] refers to hierarchy visualization as tree visualization and puts forward treevis.net project which gathers a collection of prior tree visualizations. In his study, hierarchy visualization is classified into three categories, namely explicit, implicit and hybrid, based on edge representations.

Explicit representation. Visualizations in this category, often referred to as node-link layouts, have explicit edge representations to encode hierarchical relationships. Among many existing works, Munzner et al. [39] propose TreeJuxtaposer, a system enabling comparison between large trees through a novel focus+context interface combined with guaranteed visibility. Lee et al. [40] develop an interactive visualization named TreePlus, which enables exploration of a large graph by expanding nodes within node-link layouts. To enhance the space efficiency of node-link layouts, Yan and Ma [41] propose an elastic tree layout, which dynamically adjusts the visualization, allowing for the presentation of detailed information and hierarchical context.

Implicit representation. Unlike node-link layout, visualizations such as Icicle Plots [42] and Treemaps [43], [44] use an implicit representations of edges. To address the suboptimal aspect ratios often found in the rectangular layouts and use space more compactly, circular variants such as Sunburst [45], InterRing [46], and Circular Treemaps [47] are introduced. More recently, Zhao and Lu [48] propose Variational Circular Treemap which supports users in navigating different levels of the hierarchical dataset with the assistance of focus+context techniques. Görtler et al. [49] propose Bubble Treemaps, which utilizes a circle-packing layout to display hierarchical data and incorporate uncertainty information into the visualization. Jin et al. [50] propose Radial Icicle Tree which keeps area consistent for nodes of the same size, and enhances visibility with inserted gaps between adjacent nodes.

Hybrid representation. There are visualizations which combine both explicit and implicit representation of the hierarchy. Viégas et al. [51] propose Google+Ripples which integrates node-link diagrams into Circular Treemaps to visualize sharing behaviors in social networks. Li et al. [52] develop a visualization called ClockTree which organizes nodes in a circular layout based on depth-first search, alongside arcs depicting relationships between nodes.

Prior hierarchy visualization techniques effectively encode parent-child relationships but overlook the temporal attributes of nodes within the hierarchy, limiting their ability to cohesively analyze both structural and temporal aspects of hierarchical time series. ChronoDeck uses a multi-column layout to represent multiple aggregation levels of hierarchical time series, enabling the integration of numerous time series while supporting diverse analytical tasks.

C. Visual Analytics for Hierarchical Time Series

Visual analytics is frequently used to analyze hierarchical time series. Existing approaches can be categorized as: level-by-level analysis and multi-level analysis.

Level-by-level analysis. The methods which fall into this category analyze time series data in single aggregation level or leaf nodes, adopting operations like drill-down and roll-up to update current aggregation level and corresponding visualization. Hao et al. [2] propose a method to analyze multiple time series in the same aggregation level within a space-fill layout, where time series are visualized based on importance relations. Meanwhile, tree is the most intuitive representation when it comes to visualize and analyze hierarchical data. Burch et al. [3] propose Timeline trees where timeline visualization is put next to a node-link diagram, users can easily expand or collapse the hierarchy to explore different levels. Also, Fischer et al. [4] propose ClockMap which is based on circular treemap consisting of glyph-based visualization of time series. ClockMap utilizes semantic zooming for users to explore data at diverse granularities. Besides, Janetzko et al. [5] combine treemap and various time series visualization to assist anomaly detection of power consumption data with hierarchical structure. The major drawback of above approach is that only one aggregation level or leaf nodes of hierarchical time series is visualized, which falls short in analyzing time series at different aggregation levels.

Multi-level analysis. This category of methods enable both within-level and cross-level analysis by simultaneously visualizing multiple levels of hierarchical time series. Ziegler et al. [6] use a table-based visualization to analyze multiple sector data and their aggregates in the stock market. Burch and Weiskopf [7] propose TimeEdgeTrees, which overlays timeline visualization onto the edges of an orthogonal tree diagram. With this visualization, users can implement comparison among time series across all levels. However, this method lacks support for necessary analytics goals, such as grouping multiple time series with common features and finding sub-hierarchies with similar behaviors, especially when the scale of the hierarchy grows larger. Another line of research uses stacked graphs [9], [11], [12] or streamgraphs [8], [10], [13], [14] to visualize multiple series within a hierarchical structure. Nevertheless, it may cause confusion in inspecting the trend of single time series and it is hard to manipulate the hierarchy to accommodate diverse analytical tasks.

Existing methods for hierarchical time series analysis can only achieve a limited number of tasks, whereas ChronoDeck leverages the combination of dimension reduction and small multiples visualizations in a multi-column layout, alongside interactions including highlight, align, filter, and select, to support six analytical tasks we have outlined in Sec. III-C.

III. TAXONOMY

The analysis of hierarchical time series requires the implementation of various tasks, so to extract meaningful patterns and derive valuable insights. However, previous studies are confined to simple tasks like summarize and compare, leaving the scope of such analysis unexplored. In order to better guide the whole analytical process, it is essential to establish a systematic collection of tasks involved in hierarchical time series analysis. In this section, we present a task taxonomy for hierarchical time series analysis. We introduce six analytical tasks and categorize the targets these tasks act upon within the hierarchical structure as four entities.

A. Methodology

We adopt a combination of literature survey and interviews with ten experts to develop a task taxonomy for hierarchical time series analysis. The analysis of hierarchical time series is utilized in diverse domains. Therefore, we invite ten experts from representative domains, including energy digitalization ($E_{1,6}$), business analytics ($E_{2,7}$), cloud computing ($E_{3,4}$), urban computing ($E_{5,9}$), machine learning (E_8), and e-commerce (E_{10}). All experts have at least 2 years of experience with analyzing hierarchical time series.

The study process consists of two phases. In the first phase, we produce a draft taxonomy through literature survey and expert interviews. We start by searching for papers in academic databases and digital libraries, including Google Scholar, IEEE Xplore, ACM Digital Library, and ResearchGate, using keywords "hierarchical time series", "tree", "hierarchy", "time series", "visualization", "visual analytics" and their boolean combinations. We expand our selection by leveraging reference lists from initial papers through a snowballing approach [53]. Papers are included if they present visual analytics approaches for hierarchical time series that conform to Hyndman et al.'s definition [1], where lower-level series aggregate to

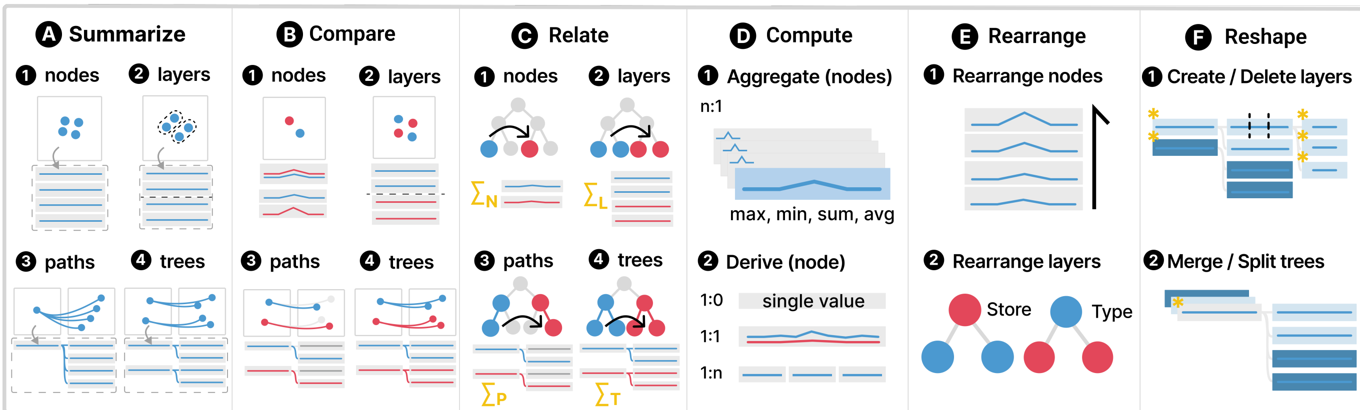


Fig. 1. The taxonomy demonstrates the categorization of targeted entities and general tasks of hierarchical time series analysis. ChronoDeck is implemented based on this taxonomy. (A) ChronoDeck visualizes dimensionality reduction results as scatter plots to summarize features of entities across multiple aggregation levels. (B) ChronoDeck utilizes both dimensionality reduction and time series visualizations to implement the task of compare. (C) Similarity measures are used to relate other similar entities. (D) The tasks of compute is classified into aggregate and derive. (E) ChronoDeck supports rearranging nodes, while does not support rearranging layers. (F) The task of reshape modifies hierarchical time series by creating/deleting layers and merging/splitting trees.

form higher-level series based on categorical, geographical, or physical constraints. After applying this criteria, we finalize 15 relevant papers. Based on gathered papers, we assemble a collective list of recurring and common tasks, which are grouped according to shared analytical goals, operations, and usage contexts. We generalize three initial analytical tasks in this process: *summarize*, *compare*, and *compute*.

Due to the limited number of papers, we conduct informal interviews with ten experts over a two-month period, allowing for open-ended discussion and brainstorming. Each interview is conducted via online meetings with experts from a single domain and lasts approximately 45 minutes. Every expert participates in at least three interview sessions. The interview process consists of three distinct stages. The initial discussions focus on understanding the basic structure and attributes of the datasets. Subsequently, experts demonstrate their general workflow live using their analytical tools. The final stage involves breaking down the workflow into specific tasks. Building upon the initial tasks identified from the literature, we leverage insights from expert interviews to contribute additional scenarios to existing categories, group similar operations within each category, refine task definitions, adjust boundaries, and further generalize three additional analytical tasks: *relate*, *rearrange*, and *reshape*.

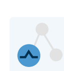
Following the task abstraction model proposed by Munzner [54], we structure our initial taxonomy around two dimensions: *entity* and *task*. The *entity* denotes the targets of analysis, while the *task* represents the analytical actions taken. Based on the tasks generalized earlier, we identify specific targets these tasks act upon within the analytical scenarios mentioned in relevant literature and expert interviews, and categorize them based on their structural characteristics into: *node*, *layer*, *path*, and *tree*.

Afterwards, we proceed to the second phase, where we present the initial taxonomy to all experts for further optimization and validation. We first introduce the definition of each entity and task, alongside specific scenarios to the experts. Then, experts offer suggestions and validation on both general structure of the taxonomy and contents in specific category. Finally, based on their feedback, we revise the taxonomy and produce the final version. The study is approved by the State Key Lab of CAD&CG, Zhejiang University.

B. Entity Categorization

In this section, we first describe the basic structure of hierarchical time series. Building on this understanding, we then demonstrate our categorization of entities.

In hierarchical time series, data is organized in a tree-like structure, with each level of the tree representing a level of aggregation. To illustrate the concept of hierarchical time series, we utilize examples from retail and photovoltaic datasets, which are provided by domain experts. The retail dataset forms a hierarchical structure with different levels of aggregation: state, region, store, and product type. Each node within this hierarchy encapsulates time series data, documenting the progression of daily sales figures. In the retail hierarchy, the data at a parent node represents the aggregate of its child nodes' data. For instance, sales data associated with a particular region is the sum of all its subordinated stores. The photovoltaic dataset exhibits a three-level hierarchical structure comprising transformer, inverter, and string. In a PV field, multiple strings are organized into groups, with each group connected to a single inverter, and multiple inverters are further clustered under a single transformer. Based on this nested nature, a hierarchical time series can be constructed by cumulating current measurements from bottom up. Datasets with similar structures can also be observed in other domains like clouding computing, business analytics, urban computing, and many others. Based on this structure, we categorize the targets of hierarchical time series analysis into four types of entities: *node*, *layer*, *path*, and *tree*.


E1: Node. This entity refers to the individual elements within the hierarchy, providing the most detailed level of information available. Examples include the daily sales data for an individual store, the current measurement for a

single PV string, or the performance monitoring data of one particular server in the data center.

E2: Layer. We define layer as a set of nodes that share the same parent within the hierarchy. It encodes the characteristics of multiple time series in one aggregation level. This could be, for example, sales data of all stores in the same region, current data for several strings connected to the same inverter, or prices of multiple stocks under one sector.

E3: Path. Extending the concept from graph theory [55], in hierarchical time series, a path represents a sequence of linked nodes from one level of the hierarchy to another, highlighting relationships between different aggregation levels. For example, a path might consist of data linked between a store and its corresponding region, reflecting how individual sales trends correlate with regional trends.

E4: Tree. The hierarchy itself or its sub-hierarchies can be represented by trees. It contains multiple time series at different aggregation levels. In the retail dataset, a simple tree can be the sales data of one region and its subordinated stores. For the photovoltaic dataset, the cumulative current measurement at one inverter and current data of each string connected to it can also be represented by a tree.

C. Task Summary

We generalize six analytical tasks: *summarize*, *compare*, *relate*, *compute*, *rearrange*, and *reshape*, which refer to analytical actions taken upon entities mentioned above during the analysis of hierarchical time series. Each analytical task is introduced as follows.

T1: Summarize. This task (Fig. 1A) refers to presenting an overview of a whole set of potentially targeted entities [56]. Burch et al. [3] summarizes the features of multiple nodes by placing thumbnail visualizations next to the leaf nodes of the hierarchy representation. Fischer et al. [4] utilizes multiple circular glyphs to present a general summary. Few prior researches focus on summarizing features across different aggregation levels. Nevertheless, domain experts present us the necessity of summarizing the characteristics of more than one aggregation level. For example, in the regular maintenance of a photovoltaic field, the analysis of strings needs to be combined with the summoned conditions at inverter-level. Therefore, an effective summary of paths contribute to the efficient identification of an anomalous combination of an inverter and a PV string. Meanwhile, in cloud computing, the overall performance of one single data center can not comprehensively uncover the performance of each server inside. Summarizing both data centers and their subordinated servers aids users in locating one specific sub-hierarchy for further analysis. By visualizing a collection of entities based on their characteristics, users can seamlessly proceed with downstream tasks such as *compare*.

T2: Compare. This task involves comparing the characteristics of two or more entities to uncover similarities and differences (Fig. 1B). Most prior studies concentrate on the comparison of nodes and layers. For nodes, Ziegler et al. [6] compare sector data across multiple countries using a table-based layout. And Cuenca et al. [8] enables the comparison between genres and sub-genres by adopting a multi-resolution approach. For layers, Janetzko et al [5] compare multiple collections of time series in a treemap-based visualization. However, insights from domain experts suggest that comparisons among other entities are equally important. For example, the user may want to compare not only the sales data of two regions, but also the sales data of their subordinated stores, which can be abstracted as a comparison between trees. Additionally, comparison between paths is also common. For instance, comparing the correlation between the prices of different stocks and the corresponding sectors can help users identify stocks with abnormal behaviors.

T3: Relate. Once an entity of interest is located, this task seeks out other entities with similar attributes by employing similarity measures and search algorithms tailored for different entities (Fig. 1C). The task is particularly useful when analyzing hierarchical time series with a large scale. The simplest example is when a user identifies a node exhibiting abnormal behavior, the relate action can efficiently find other nodes in the hierarchy with similar behavior. This approach can be also extended to other entities. For instance, photovoltaic experts tend to locate a single high-cumulative-current inverter paired with a dysfunctional string with low current measurement. These anomalous strings are often hindered by the high cumulative current values of their associated inverters. Experts can effectively find such strings by searching for similar paths through the action of relate. In retail data analysis, analysts often search for regions with similar sales conditions to formulate better business strategies. These regions not only exhibit overall similar characteristics but also have stores with comparable feature distributions. To achieve this, analysts often relate relevant trees.

T4: Compute. This task refers to conducting computation on single or multiple time series, and can be divided into two sub-categories: aggregate (Fig. 1D1) and derive (Fig. 1D2). **(1) Aggregate.** This sub-category involves using aggregation functions like *max*, *min*, *average*, and *sum*, etc. to turn multiple time series into single time series. **(2) Derive.** This sub-category focuses on transforming single time series into one single value, a new time series or multiple new time series. For example, deriving one single value may involve calculating the average current measurement of one PV string, or obtaining the ratio of change for one stock over a period of time. Deriving a new time series typically includes transformation operations like smoothing, normalization and anomaly detection. Specifically, in stock market analysis, history data for each stock is usually normalized for trend comparison. In cloud computing, anomaly detection methods are applied to the performance data of each server, generating a new time series where each timestamp is assigned an anomaly score [34]. Deriving multiple time series can be splitting single time series into multiple slices based on seasonality or trend. In the field of urban computing, Deng et al. [57] partition time series into multiple segments based on periodicity or peak identification, and analyze causal relations in each segment.

T5: Rearrange. This task refers to changing the sequence of nodes and layers. The most common one is rearranging nodes in each layer by sorting methods (Fig. 1E1).

The sorting can be based on various metrics: the average value of single time series, the overall anomaly score, etc. For example, stock market data can be sorted by the average difference between adjacent prices which indicates the volatility of one stock or sector. The arrangement can be also based on the meta information of time series. In analyzing the retail dataset, analysts tend to bring together the sales data of stores which are geographically close during the analysis. The rearrangement of layers is also regularly seen in the analysis of the retail dataset, whose hierarchical time series is constructed upon multidimensional data. Users can organize the sequence of dimensions to accommodate various analytical purposes. As an instance, users can analyze the sales of different product types in single store by putting store above product type, or explore the sales of one specific product type in multiple stores by reversing the previous sequence (Fig. 1E2).

T6: Reshape. This task refers to operations which modify the structural semantics of the hierarchy. Two major operations are summarized under this task: **(1) Create/Delete layers.** The operation of creating layers usually takes place when the user analyzes a large number of nodes in the same layer. Additional layer is added upon the current layer, consisting of nodes which is the aggregation of previous nodes with similar characteristics. In the field of photovoltaic, multiple PV strings which merge into one single inverter may demonstrate different features, maintenance personnel often groups similar string data to analyze major features (Fig. 1F1). Another example can be utilizing the result of time series partitioning to create layers underneath the current layer, which is often employed to analyze time series with cyclical patterns (Fig. 1F1). The operation of deleting layers is often adopted when dealing with a large number of aggregation levels. For example, the sales datasets from large corporate possess additional levels, including store type, product type and product version. Analysts usually remove irrelevant layers to simplify the analytical process. **(2) Merge/Split trees.** In stock market analysis, market analysts often group multiple sectors in the same industrial chain, for example, Information Technology and Communication Services, to analyze all stocks below (Fig. 1F2). Conversely, users may also choose to split grouped trees and focus on analyzing a single tree.

Justification. The analytical tasks defined by our task taxonomy primarily aim for analyzing multiple time series in a hierarchical structure with different aggregation levels. We choose not to include the *identify* task mentioned in previous studies [54], [56], as it mainly targets at analyzing the features of a single time series. Also, the functionality of the *identify* task has already been integrated with our existing tasks to a certain extent. The analytical tasks, including *summarize*, *compare*, *relate*, and *rearrange*, can be leveraged to identify specific targets with particular attributes. For example, to identify the store with the lowest sales, users can rearrange stores based on average sales values, effectively locating the target without requiring a separate *identify* task.

IV. ChronoDeck

We present ChronoDeck, an interactive visual analytics system to facilitate the analysis of hierarchical time series. In developing such a tool, we work closely with domain experts (E_{3-8}) and generalize four design goals (see Sec.IV-A) under the guidance of the nested model for visualization design and validation proposed by Munzner [58]. Based on the design goals, we further develop ChronoDeck consisting of three views: data, exploration, and selection views (Fig. 2A-C). With ChronoDeck, users can visualize, transform hierarchical time series, and compare different entities in a multi-column layout through coordinated visualization and diverse interactions, and further identify entities-of-interest in the hierarchy. In this section, we first introduce our design goals, then elaborate the visual design and interactions of ChronoDeck for in-depth analysis of hierarchical time series.

A. Design Goals

G1: Visualize hierarchical time series. Hierarchical time series consists of both structural and temporal features. The design should integrate visual representations of hierarchical structure and time series data, and provide users with a summary of time series across different levels (*summarize*).

G2: Support comparison among different entities. The design should facilitate efficient derivation of similarities and differences among different entities. It should also support detailed comparisons, allowing users to dissect relationships among entities consisting of multiple time series (*compare*).

G3: Enable transformation of hierarchical time series. Given the extensive number of time series within the hierarchy, users should be able to rearrange nodes in one aggregation level based on various criteria (*rearrange*). Also, the design should support operations like creating layers and merging trees (*reshape*). Additionally, various computation methods should be embedded (*compute*) in the system to facilitate the transformation of hierarchical time series.

G4: Facilitate identification of entities-of-interest. The design should enable users to identify entities-of-interest during the exploration process. Moreover, it should allow users to correlate other entities in the hierarchy that share common characteristics based on similarity measures (*relate*).

In the following four subsections, we introduce design features to support the above design goals.

B. Visualization of Hierarchical Time Series

This section illustrates how ChronoDeck employs a multi-column layout to visualize hierarchical time series, providing both a visual summary across different aggregation levels and details of multiple time series. We also introduce the visual design for emphasizing the currently explored hierarchy.

Multi-column layout. Inspired by the column view in macOS®'s Finder, ChronoDeck adopts a multi-column layout to integrate hierarchical structure and time series features. This layout visualizes each aggregation level in one column, with multiple columns aligned to display the hierarchy (Fig. 2B). Each column consists of two components: a scatter plot and small multiples visualizations of time series. To provide a visual summary for each level, we employ t-SNE [59], a robust dimensionality reduction technique, to convert time series at the current level into two-dimensional coordinates. They are

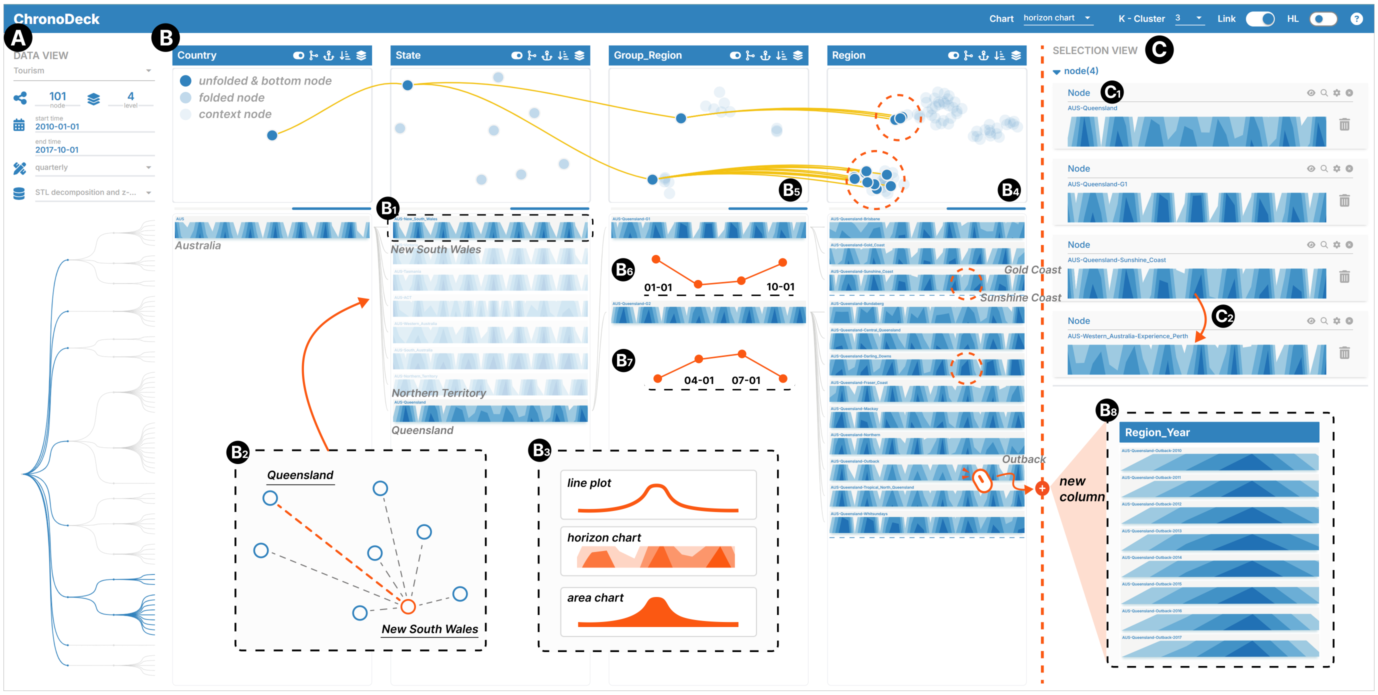


Fig. 2. The user interface of ChronoDeck. (A) The data view presents the basic information of hierarchical time series. (B) The exploration view visualizes hierarchical time series in a multi-column layout, where users are able to compare different entities and transform hierarchical time series through various interactions. (C) The selection view displays selected entities and allows users relate similar entities within the hierarchy.

visualized as scatter plots above each column, encoding key features of time series as positions in the plot and enabling easy identification of clusters (Fig. 2B4). Although the results of t-SNE, such as orientation, may change for different random seeds, the algorithm preserves the local neighborhood structure that is critical for discovering patterns or clusters among multiple time series and ensuring the consistency of repeated analysis. Below each scatter plot, small multiples visualizations detail individual time series. Users can brush specific intervals and update the visualization for further examination.

Progressive exploration. ChronoDeck enables incremental adjustment of the visualization to explore different parts of the hierarchy. Users can unfold or fold corresponding sub-hierarchies by clicking specific time series visualizations. To emphasize users' exploration of hierarchical time series, we utilize opacity to distinguish three categories of nodes within the hierarchy as follows (Fig. 2B): (1) Unfolded & Bottom nodes: Either unfolded with their children displayed in the next column, or the deepest leaf nodes within the currently explored hierarchy. (2) Folded nodes: Within the currently explored hierarchy, but their children are not displayed in deeper levels. (3) Context nodes: Not within the currently explored hierarchy. In scatter plots, three categories of nodes are assigned with opacity of 100%, 30%, and 10%. Moreover, they are placed in top, middle, and bottom layers to emphasize the currently explored hierarchy while maintaining the overall context. In time series visualizations, context nodes are not visualized, while the first two categories are assigned with opacity of 100% and 30%. Two sets of links in yellow and grey are used to connect visualizations in both scatter plots and small multiples with parent-child relationships. Additionally,

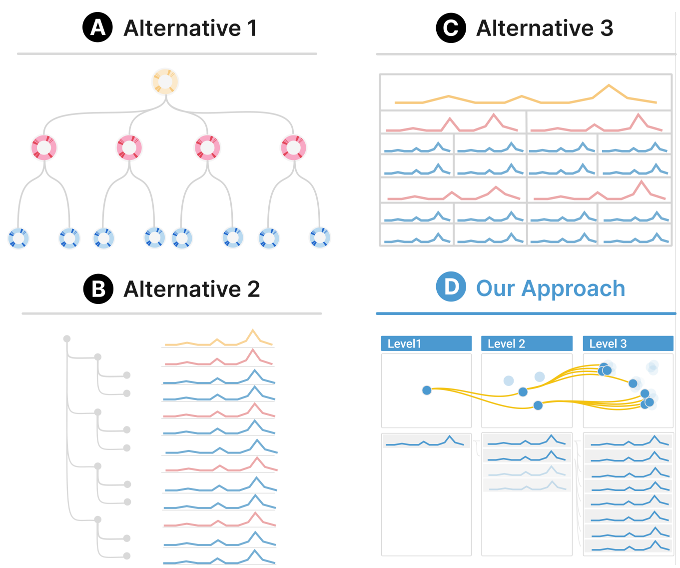


Fig. 3. Design alternatives for hierarchical time series visualization. (A) An explicit node-link layout, integrating circular-glyph-based visualization. (B) A juxtaposed layout which aligns time series visualization with a linearized tree. (C) A treemap-based visualization. (D) Our multi-column visualization.

coordination between scatter plots and time series visualizations allows users to hover over time series visualizations to examine corresponding circles in scatter plots.

Justification. Our design mainly centers around incorporating time series representation into hierarchy visualization. Schulz [38] classifies hierarchy visualization into explicit, implicit and hybrid based on edge representations. We present design alternatives which belong to the first two categories, as hybrid representations are rarely used. We first consider an explicit node-link layout to depict the hierarchical structure, where each node was represented by a circular glyph, illustrating the characteristics of time series data (Fig. 3A). However, this approach has scalability issues, particularly with a large number of time series. Another visualization we consider is similar to Timeline tress [3] which visualizes the hierarchy using an explicit node-link layout and positions time series visualization next to the corresponding node. We further utilized the tree linearization technique [60] to make time series across all aggregation levels be visualized along the same vertical axis (Fig. 3B). Nevertheless, users have to constantly switch between two visualizations, which causes a high cognitive load. We then develop a treemap-based approach which conserves space and allows for detailed inspection of individual time series (Fig. 3C). The drawback of this approach is that time series in different aggregation levels have different proportions, making them hard to compare. With above limitations considered, we finally propose a more scalable and compact visualization with a multi-column layout (Fig. 3D). The parent-children relationship is explicitly encoded using links between adjacent columns.

C. Comparison Among Different Entities

Compare is a critical task in hierarchical time series analysis. Based on the visual comparison action categories proposed by Gleicher [61], we utilize linked scatter plots for identifying and summarizing relationships (similarities and differences) among entities, and facilitates the dissection of relationships by examining details in time series visualization.

The linked scatter plots on the top of columns offer an efficient approach to compare all four entities. Users can identify similarities or differences among multiple nodes or layers based on the distribution of circles in a single scatter plot. Comparison of paths and trees is enabled by examining distances among node-link layouts across different aggregation levels in multiple linked scatter plots. Additionally, ChronoDeck introduces the interaction of highlight. When the folded nodes are hovered over in the time series visualization, corresponding sub-hierarchy is temporarily highlighted in the linked scatter plots and can be compared with the currently explored hierarchy, which is temporarily rendered at lower opacity for visual differentiation (Fig. 4B1). This interaction aids users with a preview for deciding the subsequent operations, such as expanding the hovered node and comparing details in the time series visualization.

The time series visualization supports more detailed comparison. Based on the study conducted by Javed et al. [62], to compare multiple time series within the hierarchy, ChronoDeck employs small multiples visualizations, complemented by shared-space approaches. In each column, entities in the same level (e.g. nodes and layers) are visualized separately and can be configured to different charts including line plot, area chart and horizon chart for juxtapose comparison (Fig. 2B3). Line plots allow users to click the time bar above small multiples for overlapped comparison (Fig. 4A1).

To compare time series across different levels, ChronoDeck introduces the interactions of select and align. Users can select relevant entities in the selection view for further examination by right-clicking the time series visualization. For example, nodes from different levels can be selected for direct comparison by listing vertically in the selection view (Fig. 2C1). The align interaction addresses horizon misalignment between parent-child nodes and enhances entity distinction. Clicking the alignment button above each column aligns subsequent time series relative to the chosen level, facilitating explicit comparison between paths and trees (Fig. 4B2).

Justification. Gleicher et al. [63] categorize visual comparison designs into three fundamental approaches: juxtaposition, superposition, and explicit encoding. For comparing node-link layouts in linked scatter plots, we employ superposition methods, as juxtaposition methods can present scalability issues, and explicit encoding is rarely utilized in this context. Meanwhile, for time series comparison, we primarily utilize small multiples visualization, a juxtapose design, due to its advantages in comparing time series across different levels and interacting with individual time series visualization.

D. Transformation of Hierarchical Time Series

The transformation of hierarchical time series involves two main aspects: rearrange and reshape hierarchical time series.

Rearrange hierarchical time series. ChronoDeck allows users to rearrange hierarchical time series by modifying the sequence of nodes at each aggregation level. This can be achieved in two ways. First, users can sort nodes within a level by clicking buttons at the top of each column, based on criteria such as the average value or the ratio of change. Second, ChronoDeck leverages the result of dimension reduction to rearrange nodes for exploratory analysis. Users can drag a time series to the first position as a reference, and the system would reorder the other nodes, based on their distance from the reference in the scatter plot (Fig. 2B2). Additionally, users can utilize the lasso tool in scatter plots to filter time series.

Reshape hierarchical time series. In ChronoDeck, there are two main operations: creating layers and merging trees.

To reshape the hierarchy by creating layers, users can group time series with similar characteristics by adding a new layer upon the current one. Within each new layer, time series for each node is automatically calculated from its children time series in the grouped layer, adopting aggregation computations like average or sum. To maintain the consistency of the overall hierarchical structure, ChronoDeck groups all time series in the same aggregation level and creates multiple new layers which form additional aggregation level. This operation is implemented by first locating an aggregation level, and assigning a cluster number based on the t-SNE layout, which is optimized for distinguishing clusters. Users then can click on the grouping button on top of the current column, using K-Means to group time series. Subsequently, a new column visualization is added to the left of current column as an additional level. In this newly created column, the coordinate of each circle in the scatter plot is the centroid of its grouped circles (Fig. 2B5), and each time series is visualized using the average trend of its children time series by default.

Besides grouping similar time series, users can also create layers to facilitate analyzing time series with seasonal patterns.

By double-clicking time series visualization, ChronoDeck automatically partitions all time series in the current level and utilizes the partitioning results to create new layers in the subsequent column (Fig. 2B8).

Meanwhile, the operation of merging trees enables users to collectively analyze sub-hierarchies with similar characteristics. To merge two currently explored sub-hierarchies, users can simply click the merge button on top of each column. As a result, the root node of the merged tree will be the average series of the previous two in default, while descendants are inserted under the newly created root node.

Justification. Aghabozorgi et al. [64] generalize three ways to cluster time series: model-based, feature-based, and shape-based. In the process of creating layers, we adopt the feature-based approach due to its close integration with our visual design, with which users first identify clusters in the scatter plots representing time series features, and then apply the K-Means algorithm to group similar time series. We opt for K-Means clustering because it enables users to determine K based on the observed distribution in scatter plots and make iterative adjustments to refine the clustering results. In contrast, automatic approaches such as DBSCAN [65] may yield suboptimal results and offer limited space for adjustments.

E. Identification of Entities-of-Interest

During the analysis process, users can right-click the time series visualization to select relevant entities for further examination in the selection view (Fig. 2C1). Each selected entity is then displayed within a corresponding card in the selection view. Users have the option to refine these entities by editing out less significant features, such as removing less important nodes in a path card.

However, simply selecting entities from the exploration view is not enough for users to identify all potential entities-of-interest. To address this, users can click on the search button to relate other entities in the hierarchy which share common characteristics with the current entity based on similarity measures. After correlating multiple candidates, the corresponding entity cards will be displayed below the current card (Fig. 2C2). We provide similarity measures for each category of entities. We employ the Euclidean distance to measure the similarity between each pair of nodes. For layers, we utilize the Hungarian method [66] to match nodes from two distinct layers using Euclidean distance. Subsequently, the average distance between each matched pair of nodes is calculated to represent the overall similarity between layers. This approach is applied to measure the similarity between paths or trees. We first normalize each time series for a given node pair if there are scale variations at different aggregation levels, then we compute the average distance between each node pair. For trees spanning more than two levels, we adopt the same approach recursively.

Justification. Numerous methods exist for quantifying similarity between time series [67]. We choose Euclidean distance because it is the most common approach to measure time series similarity. Given that matched time series share identical time ranges, Euclidean distance is suitable for assessing similarity for temporal characteristics like trends and seasonality.

V. IMPLEMENTATION

We implement ChronoDeck from two modules: frontend and backend. In the frontend, we use JavaScript, alongside libraries like Vue.js, Vuex, and D^3 [68] to construct the interactive visualization. We build our backend server based on Python and Flask, while using libraries like numpy and sklearn to realize algorithms including dimension reduction, clustering and entity search. Our dataset contains the meta information and hierarchical time series data. The meta information documents the basic attributes of each node and hierarchical time series data is stored inside a nested folder in the format of json file. Finally, we open source our system at https://github.com/ChronoDeck/ChronoDeck.

VI. EVALUATION

In this section, we evaluate the effectiveness of ChronoDeck through case studies on three real-world datasets with domain experts, complemented by expert interviews to collect their feedback. The evaluation demonstrates ChronoDeck's capabilities in implementing diverse analytical tasks on hierarchical time series.

A. Method

Participants and data. Domain experts E_a, E_b, and E_c participate in our case studies. E_a is specialized in the fields of urban computing, and familiar with air quality and travel data analysis. E_b has expertise in energy digitalization, and E_c possesses extensive experience in stock market assessment. All three experts are not involved in the design of the system. We use three datasets in case studies: the tourism dataset, the photovoltaic dataset and the stock market dataset. **(1) The tourism dataset.** This dataset [69] consists of quarterly tourism volumes data in Australia between 1998 to 2017, aggregated at country, state, and region levels. Holiday and visiting purposes are selected because they represent the majority of traveling activity. **(2) The photovoltaic dataset.** The dataset provided by E_b has three aggregation levels: transformer, inverter, and string. Within the hierarchical time series, there are multiple inverters belong to a single transformer, and each inverter is incorporated with multiple grouped strings. The time series documents the current measurements summoned from string level in four months on a daily basis. **(3) The stock market dataset.** This dataset is obtained by Yahoo Finance API, documenting the daily history data of the stock market in March 2023. The hierarchical time series also has three aggregation levels: index, sector, and stock. we choose S&P500 as the index, and 11 sectors including Energy, Information Technology, Utilities, etc. Within each sector, there are stocks for each company like Apple, Microsoft, and Tesla. For each time series, Adj Close is suggested by E_c as our main metric, because it represents the most accurate performance of stocks.

Study protocol. In case studies, each expert spontaneously uses ChronoDeck to analyze the assigned dataset by performing various analytical tasks, while following the think aloud protocol [70]. The study is conducted using the Chrome browser on a 3840×2160 display and we document the whole analytical process. The study is approved by the State Key Lab of CAD&CG, Zhejiang University.

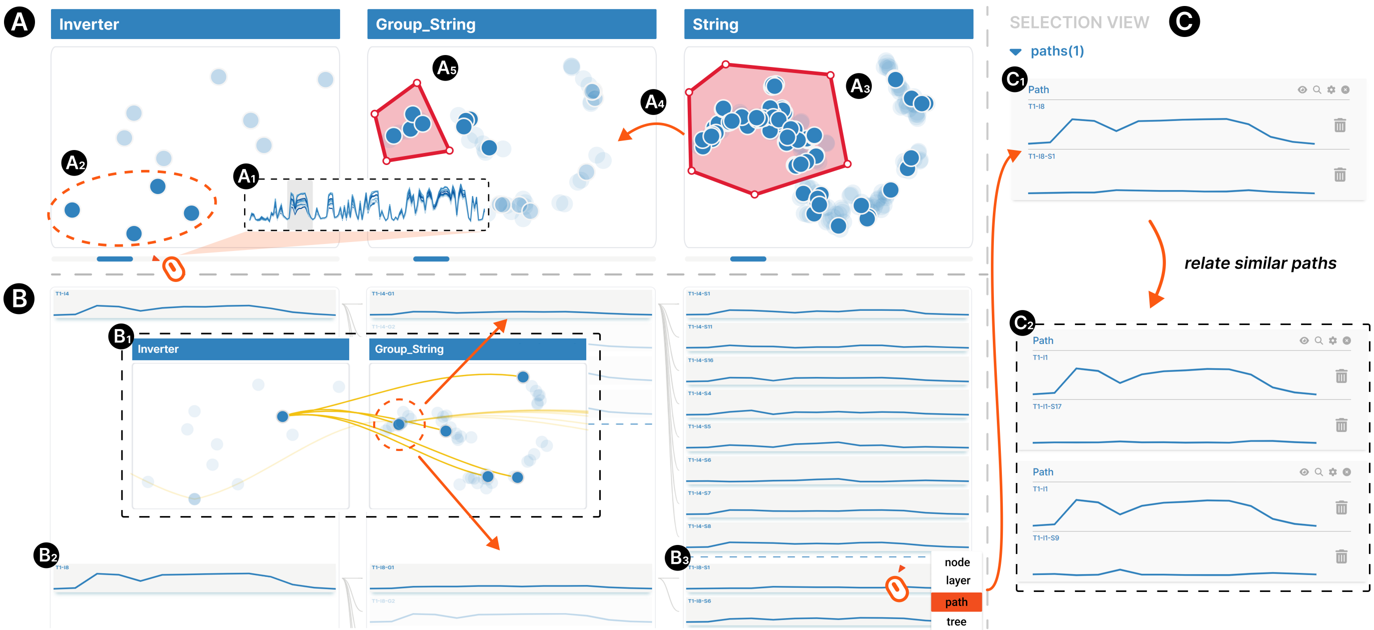


Fig. 4. The process of analyzing the photovoltaic dataset. (A) Distinguish inverters with different cumulative current values and analyze anomalous strings under low-cumulative-current inverters. (B) Analyze anomalous lines under high-cumulative-current inverters. (C) Select an entity-of-interest, and relate paths consisting of time series representing a high-cumulative-current inverter and a low-current PV string.

B. Case 1: Analyzing the Tourism Dataset

E_a intends to analyze the seasonality of tourism volumes in different geographical scales. To highlight seasonal peaks, E_a extracts the seasonal component from all time series using STL decomposition and applies z-normalization to mitigate scale differences (*compute*). Also, he configures the default line plot into horizon chart for enhanced visualization.

Analyze the tourism volumes of states under Australia (G2-3). Focusing on recent data, E_a selects the period since 2010, quickly identifying seasonal peaks in the first quarter across Australia. E_a then expands the hierarchy to the state-level. E_a finds New South Wales state, similar to Australia, reaches its peak in the first quarter (Fig. 2B1). By dragging New South Wales to the top and clicking the anchor button, the rest of states are reordered based on their distances to New South Wales in the scatter plot (Fig. 2B2) (*rearrange*). E_a discovers two states: Northern Territory and Queensland, with peaks in the third quarter, which is different from the pattern found in New South Wales (*compare*).

Analyze the tourism volumes of regions under different states (G1-4). E_a subsequently unfolds regions underneath these two states. The regions under Northern Territory are grouped within one single cluster in the scatter plot (*summarize*), demonstrating peaks in the third quarter. While for regions under Queensland, E_a identifies two distinct clusters in the scatter plot (Fig. 2B4) (*summarize*, *compare*). Therefore, he reshapes the hierarchy by creating new layers in the Group_Region column, grouping time series with similar patterns (Fig. 2B5) (*reshape*, *compute*). E_a aligns the time series visualization to the Group_Region column to facilitate the comparison between two sub-hierarchies (*compare*). Within two groups, one group containing regions such as Gold Coast and Sunshine Coast presents tourism peaks in the first and fourth quarter because of favorable weather conditions and holiday seasons (Fig. 2B6), while the other group mostly peaks in the second and third quarters due to optimal climate for outdoor exploration (Fig. 2B7). To analyze individual region, E_a double-clicks node like Outback, presenting eight partitioned yearly time series in the newly created column (Fig. 2B8) (*reshape*, *compute*).

Afterwards, E_a right-clicks the time series visualization to select nodes from the State and Group_Region columns, comparing time series from different levels on the right selection panel (Fig. 2C1) (*compare*). He also selects Sunshine Coast and clicks the search button to relate similar ones like Experience Perth within the hierarchy (Fig. 2C2) (*relate*).

C. Case 2: Analyzing the Photovoltaic Dataset

The analytical goal for E_b is to identify PV strings with low current values. Since the cumulative current of inverters reflects the general state of their connected strings, and strings under the same inverter typically exhibit similar attributes, E_b wants to leverage the hierarchical structure of the PV system to first identify inverters with distinct features and then drill down into specific strings for effective fault detection and diagnosis.

Distinguish inverters with different cumulative current values (G3). E_b first identifies a period where the variance of cumulative current values among inverters is relatively large by inspecting the overlapped line plots (Fig. 4A1). E_b thus brushes this interval for further analysis. E_b sorts inverters based on the average current values (*rearrange*, *compute*), noting that the four bottom-left circles represent inverters with the minimal current values (Fig. 4A2).

Analyze anomalous strings under low-cumulative-current inverters (G1-3). E_b unfolds these four inverters to analyze PV strings underneath. In the String column, a large number of circles is observed on the left side (*summarize*). Therefore, E_b adopts lasso to select this collection of circles (Fig. 4A3) and groups similar strings by creating new layers in the Group_String column (Fig. 4A4) (*reshape*, *compute*). In the newly created column, two distinct clusters can be observed. Further investigation reveals that four circles on the left are grouped by strings with the lowest current values (*summarize*, *compare*). So E_b adopts lasso again to filter out the right cluster (Fig. 4A5). At this point, E_b has selected four inverters with the lowest cumulative current values, each contains one string group, consisting a number of PV strings with extremely low current values. Next, E_b aligns the visualization to the Group_String column, and discovers that Inverter 4 has eight anomalous strings while the others only own three or four such strings (*compare*). E_b later examines the information of these strings on the construction drawings and finds that the corresponding solar panels have suboptimal installation angles, resulting in little radiation being received, which causes abnormal behaviors of the PV strings.

Analyze anomalous strings under high-cumulative-current inverters (G2, G4). E_b retains Inverter 4 and its corresponding node-link layout for potential comparison. By highlighting sub-hierarchies under inverters with high cumulative current values, E_b finds that one of Inverter 8's children circles in the Group_String column is close to the existing circle, which stands for a string group with low current values (Fig. 4B1) (*compare*). E_b subsequently expands Inverter 8 and discovers an anomalous string group consisting of two strings with very low current values (Fig. 4B2). Unlike the low-value PV strings under low-cumulative-current inverters, anomalous strings like String 1 and String 9 are often hindered by the high cumulative current of inverters, making it hard for the field personnel to detect. Therefore, E_b right-clicks String 1 to select a top-down path (Fig. 4B3). Later, E_b edits the corresponding entity, retaining only Inverter 8 and String 1 (Fig. 4C1). Finally, E_b searches for similar paths in the unexplored parts of the hierarchical time series (*relate*). The result shows that there exist anomalous strings under Inverter 1, which is an inverter with high cumulative current values (Fig. 4C2). In particular, E_b finds that String 9 and String 17 under Inverter 1 not only exhibit low-current values, but also reversing trend compared to the trend of Inverter 1. E_b points out that String 9 and String 17 may have circuit issues and require maintenance.

D. Case 3: Analyzing the Stock Market Dataset

In this study, E_c aims to gather market information and insights for a comprehensive assessment report on stock market, which consists of two parts: identifying growing sectors and stocks and analyzing heterogeneous trends in distinct periods.

Identify growing sectors and stocks (G1-3). First, E_c wants to summarize significant sectors and hot stocks with growing trends. To ensure scale independence, he applies z-normalization to all time series (*compute*). Initially, E_c discovers that the index, after experiencing some fluctuations in the earlier stages, shows a general upward trend in the subsequent period. E_c brushes on the period for further

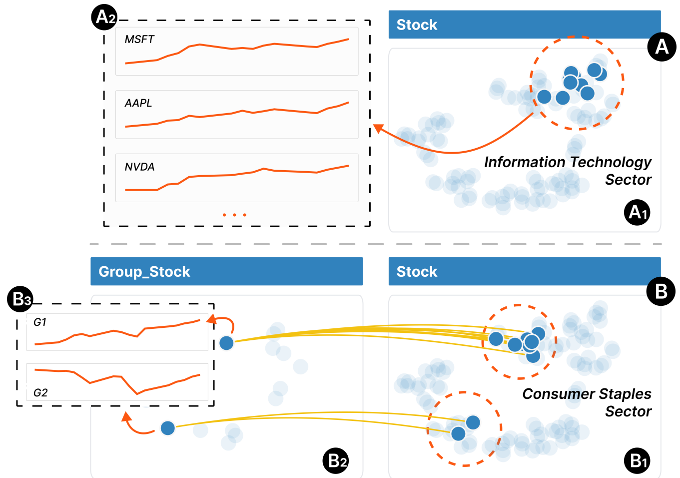


Fig. 5. The process of identifying growing sectors and stocks. (A) Analyze stocks under the Information Technology sector. (B) Analyze stocks under the Consumer Staples sector with the operation of creating layers.

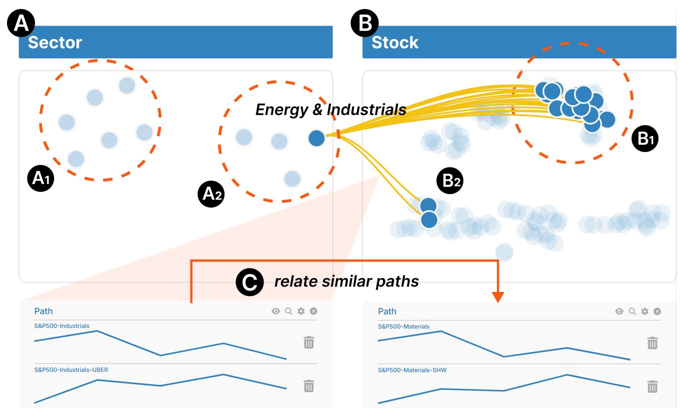


Fig. 6. The process of identifying heterogeneous trends in distinct periods. (A) Analyze sectors with upward and downward trends. (B) Analyze stocks under the Energy and Industrials sectors. (C) After the operation of relate, a similar path is identified under the Materials sector.

analysis. By sorting sectors based on the ratio of change, he finds that sectors related to the technology industry have large increases, while sectors like Energy and Materials show more moderate trends (*rearrange*, *compute*). E_c then expands the top five hottest sectors, and discovers that the circles under the Information Technology sector are clustered in the scatter plot (Fig. 5A1), indicating upward trends (*summarize*). These time series represent tech giants like Microsoft, Apple and Nvidia (Fig. 5A2). Subsequently, in the scatter plot, E_c discovers two distinct clusters of stocks under the Consumer Staples sector (Fig. 5B1) (*summarize*, *compare*). Thus, E_c creates layers to group similar stocks (Fig. 5B2), and utilizes alignment to better differentiate the two categories of time series (*reshape*, *compute*). E_c instantly finds that the first category shows an upward trend, whereas the second category declines in the early and middle stages (Fig. 5B3) (*compare*). Hence, E_c decides not to include the second category in the collection of hot stocks.

Analyze heterogeneous trends in distinct periods (G1-4). Through overlapped line plots, E_c identifies significant variations across sectors in the middle period and proceeds to brush over this segment for detailed analysis. E_c continues to sort these based on the ratio of change (*rearrange*, *compute*). Two distinct clusters are visualized in the scatter plot, where the left cluster represents sectors with upward trends (Fig. 6A1), and the right cluster represents sectors with downward trends (Fig. 6A2) (*summarize*). E_c shifts focus to the Energy sector, which exhibits a large decline. Upon further exploration, E_c finds that the underlying stocks are distributed within a cluster in the scatter plot, trending downwards (*summarize*) (Fig. 6B1). Moreover, E_c investigates stocks under the Industrials sector, part of the same industrial chain, and notes that while most stocks exhibit a distribution similar to that of the Energy sector, there exists two anomalous stocks trending upwards (*summarize*, *compare*). Finding this noteworthy, E_c selects the corresponding path (Fig. 6B2), and relate similar phenomena, revealing that within the declining Materials and Financials sectors, there are stocks experiencing upward trend such as SHW (Fig. 6C) (*relate*). Subsequently, E_c applies the operation of merging trees to amalgamate the Energy and Industrials sectors (Fig. 6A) (*reshape*, *compute*) and uses lasso to filter out outlier stocks, systematically organizing the stocks within the same industrial chain that consistently demonstrate downward trends.

E. Expert Interviews

After the case studies, we continue to conduct interviews with domain experts E_a, E_b and E_c to collect their feedback on ChronoDeck and ask them to compare the analytical process using ChronoDeck with their traditional workflows.

In general, all experts evaluate our proposed methods favorably, acknowledging ChronoDeck's effectiveness across various analytical tasks. E_a commends ChronoDeck's abilities in rearranging and reshaping hierarchical time series with the multi-column layout: "*In ChronoDeck, rearranging nodes and adding new layers to group time series with similar features become much more intuitive and efficient compared to manual adjustment.*" E_b praises ChronoDeck's support for the relate and compute tasks: "*ChronoDeck makes these complex operations more automatic and seamlessly integrates them into the system workflow.*" E_b also appreciates the alignment interaction: "*It brings out a clean look for users to compare multiple sub-hierarchies. The layout in traditional system makes the comparison of trees rather difficult.*" E_c highlights that scatter plots on top of each column are helpful in identifying stocks with various trends: "*Such summary is necessary while lacking in existing workflows. It also facilitates the comparison among different groups of stocks with heterogeneous features.*"

Meanwhile, experts offer some suggestions on future improvement. E_a recommends integrating more time series visualization techniques, such as spiral diagrams or circular glyphs, into ChronoDeck to improve the representation of features like seasonality. E_b suggests that a focus-context method can be applied to each time series card, so that the user can not only inspect the trend in the current time range, but also be aware of the whole context. E_c expects a more refined node-link layout between adjacent columns with less edge crossings.

VII. Discussion

The discussion is centered around four aspects: implications, lessons learned, limitations and future work.

Implications. In this study, we propose a task taxonomy for hierarchical time series analysis, which we think is applicable to various applications involving hierarchy or time series data. We also develop ChronoDeck, a system that adopts a multi-column coordinated visualization for such analysis. We argue ChronoDeck is capable to analyze data in a wide range of domains like retail, energy, and urban computing.

Lessons learned. Two lessons are learned in the process. First, while multiple coordinated views [71]–[74] offer flexibility and scalability in exploring complex datasets, they could introduce increased cognitive load and potential information loss, particularly during the analysis of hierarchical time series. Based on the feedback from domain experts after case studies, the experts prefer integrated design [2], [7] for such analysis, as it enables them to cohesively analyze temporal and structural characteristics. Second, hierarchical context is essential. Initially, we follow the drill-down strategy without additional design features. But we find users uncertain about unfolding without knowing the children nodes. To address this, we utilize linked scatter plots to have a visual summary across levels, and interactions like highlight to preview sub-hierarchies. In this way, users can better analyze data at different levels.

Limitations. We have observed three major limitations in our system. The first limitation can be analyzing hierarchical time series with multiple variables in each aggregation levels. Although we can overlap multiple time series, it may cause visual clutter and also a high cognitive load for users to depict the trend of each variable. Also, ChronoDeck is designed to analyze the hierarchy whose sub-hierarchies are consistent, it may not be suitable for the analysis of unstructured hierarchical time series. For example, hierarchical time series generated by a coal-fired power plant [75], where different sub-systems can possess heterogeneous hierarchical structure. In addition, our method assumes time series in the same level share an identical time range, so it cannot analyze time series with non-uniform cycle intervals through creating layers. Besides, while our taxonomy provides a systematic framework for analyzing hierarchical time series, it can be further optimized for comprehensiveness in two primary directions. On the one hand, the entity categorization, currently features the structure of the hierarchy, can be expanded to include different kinds of time segments involved in the analysis. On the other hand, the task summary can be extended to incorporate more advanced analytical operations, such as explanatory tasks like root cause analysis and query methods based on data properties.

Future work. To improve the capabilities of ChronoDeck and make it applicable to a wider range of analytical scenarios. First, we also want to adjust and expand our multi-column layout, making it suitable for visualizing hierarchical time series with heterogeneous sub-hierarchies. Second, we are intended to incorporate multi-variable visualizations and algorithms into the current system, further extending the coordinated dimensionality reduction and small multiples visualizations to accommodate multi-variable scenarios. Meanwhile, based on current studies on time lags among multiple time series [57], [75]–[77], we expect optimization for ChronoDeck to analyze multiple time series with time lags within hierarchical time series and visualize patterns like cascading effects. Finally, we aim to apply ChronoDeck to real-time data analysis, assisting experts and users in making timely and effective decisions.

VIII. Conclusion

This study proposes ChronoDeck, an interactive visual analytics system which integrates multi-column representation and diverse interactions to facilitate the analysis of hierarchical time series. To tackle three major challenges, namely delineating diverse analytical tasks of hierarchical time series analysis, visualizing the integration of hierarchical structure and time series and supporting diverse hierarchy-aware analytical tasks on many time series. We first generalize a taxonomy for hierarchical time series analysis by conducting literature review and interviews with domain experts. Then we utilize coordinated dimensionality reduction and small multiples visualizations, alongside inter-column links to visualize hierarchical time series. Various interactions are also employed to assist users in the visualization and transformation of hierarchical time series, comparison among different entities, and identifying entities-of-interest. The effectiveness of ChronoDeck is evaluated by case studies on three datasets and expert interviews.

Acknowledgments

We would like to thank the anonymous reviewers for their constructive comments. This work was supported by NSFC (U22A2032, 62402421, 62402184), Zhejiang Provincial Natural Science Foundation of China (LD25F020003), and Ningbo Yongjiang Talent Programme (2024A-399-G).

References

[1] R. J. Hyndman, R. A. Ahmed, G. Athanasopoulos, and H. L. Shang, "Optimal combination forecasts for hierarchical time series," *Computational Statistics & Data Analysis*, vol. 55, no. 9, pp. 2579–2589, 2011.

[2] M. Hao, U. Dayal, D. Keim, and T. Schreck, "Importance-driven visualization layouts for large time series data," in *Proceedings of IEEE Symposium on Information Visualization*, 2005, pp. 203–210.

[3] M. Burch, F. Beck, and S. Diehl, "Timeline trees: visualizing sequences of transactions in information hierarchies," in *Proceedings of the working conference on Advanced Visual Interfaces*, 2008, pp. 75–82.

[4] F. Fischer, J. Fuchs, and F. Mansmann, "ClockMap: Enhancing Circular Treemaps with Temporal Glyphs for Time-Series Data," in *Proceedings of Eurographics Conference on Visualization*, 2012.

[5] H. Janetzko, F. Stoffel, S. Mittelstädt, and D. A. Keim, "Anomaly detection for visual analytics of power consumption data," *Computers & Graphics*, vol. 38, pp. 27–37, 2014.

[6] H. Ziegler, M. Jenny, T. Gruse, and D. A. Keim, "Visual market sector analysis for financial time series data," in *Proceedings of IEEE Symposium on Visual Analytics Science and Technology*, 2010, pp. 83–90.

[7] M. Burch and D. Weiskopf, "Visualizing Dynamic Quantitative Data in Hierarchies - TimeEdgeTrees:Attaching Dynamic Weights to Tree Edges," in *Proceedings of the International Conference on Imaging Theory and Applications and International Conference on Information Visualization Theory and Applications*, 2011, pp. 177–186.

[8] E. Cuenca, A. Sallaberry, F. Y. Wang, and P. Poncelet, "MultiStream: A Multiresolution Streamgraph Approach to Explore Hierarchical Time Series," *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 12, pp. 3160–3173, 2018.

[9] Z. Zhou, Y. Li, Y. Ni, W. Xu, G. Hu, Y. Lai, P. Chen, and W. Su, "VisCI: A visualization framework for anomaly detection and interactive optimization of composite index," *Visual Informatics*, vol. 8, no. 1, pp. 1–12, 2024.

[10] J. Feng, K. Wu, and S. Chen, "Topicbubbler: An interactive visual analytics system for cross-level fine-grained exploration of social media data," *Visual Informatics*, vol. 7, no. 4, pp. 41–56, 2023.

[11] M. Wattenberg and J. Kriss, "Designing for Social Data Analysis," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 4, pp. 549–557, 2006.

[12] D. Baur, B. Lee, and S. Carpendale, "Touchwave: kinetic multi-touch manipulation for hierarchical stacked graphs," in *Interactive Tabletops and Surfaces*, 2012, pp. 255–264.

[13] M. Ghoniem, D. Luo, J. Yang, and W. Ribarsky, "NewsLab: Exploratory Broadcast News Video Analysis," in *Proceedings of IEEE Symposium on Visual Analytics Science and Technology*, 2007, pp. 123–130.

[14] W. Dou, L. Yu, X. Wang, Z. Ma, and W. Ribarsky, "HierarchicalTopics: Visually Exploring Large Text Collections Using Topic Hierarchies," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2002–2011, 2013.

[15] E. R. Tufte, *The Visual Display of Quantitative Information*. Graphics Press, 1983.

[16] W. Müller and H. Schumann, "Visualization methods for time-dependent data - an overview," in *Proceedings of the 2003 Winter Simulation Conference*, vol. 1, 2003, pp. 737–745.

[17] Y. Fang, H. Xu, and J. Jiang, "A Survey of Time Series Data Visualization Research," *IOP Conference Series: Materials Science and Engineering*, vol. 782, no. 2, p. 022013, 2020.

[18] W. Aigner, S. Miksch, H. Schumann, and C. Tominski, *Visualization of Time-Oriented Data*, 2nd ed. Springer, 2023.

[19] C. V. F. de Souza, S. M. Bonnet, D. de Oliveira, M. Cataldi, F. Miranda, and M. Lage, "PROWIS: A Visual Approach for Building, Managing, and Analyzing Weather Simulation Ensembles at Runtime," *IEEE Transactions on Visualization and Computer Graphics*, vol. 30, no. 1, pp. 738–747, 2024.

[20] Q. Lu, Y. Ge, J. Rao, L. Ling, Y. Yu, and Z. Zhang, "WaterExcVA: a system for exploring and visualizing data exception in urban water supply," *Journal of Visualization*, vol. 26, no. 4, pp. 957–976, 2023.

[21] J. Wang, J. Ma, Z. Zhou, X. Xie, H. Zhang, Y. Wu, and H. Qu, "TacPrint: Visualizing the Biomechanical Fingerprint in Table Tennis," *IEEE Transactions on Visualization and Computer Graphics*, vol. 30, no. 6, pp. 2955–2967, 2024.

[22] Y. Yang, X. Yi, Y. Jin, S. Li, K. Ma, S. Liu, D. Deng, D. Weng, and Y. Wu, "PVeSight: Dimensionality reduction-based anomaly detection and visual analysis of photovoltaic strings," *Visual Informatics*, p. 100243, 2025.

[23] L. Zhan, R. Li, R. Cao, J. Lin, and S. Guo, "VisMocap: Interactive visualization and analysis for multi-source motion capture data," *Visual Informatics*, vol. 9, no. 2, p. 100235, 2025.

[24] F. Zhu, X. Zhu, X. Wang, Y. Ma, and J. Zhao, "ATVis: Understanding and diagnosing adversarial training processes through visual analytics," *Visual Informatics*, vol. 8, no. 1, pp. 71–84, 2024.

[25] J. J. van Wijk and E. R. van Selow, "Cluster and Calendar Based Visualization of Time Series Data," in *Proceedings of IEEE Symposium on Information Visualization*, 1999, pp. 4–9.

[26] P. Xu, H. Mei, L. Ren, and W. Chen, "ViDX: Visual Diagnostics of Assembly Line Performance in Smart Factories," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 291–300, 2017.

[27] J. V. Carlis and J. A. Konstan, "Interactive visualization of serial periodic data," in *Proceedings of the 11th Annual ACM Symposium on User Interface Software and Technology*, 1998, pp. 29–38.

[28] C. Tominski and H. Schumann, "Enhanced interactive spiral display," in *Proceedings of the annual SIGRAD conference, special theme: interactivity*, no. 34, 2008, pp. 53–56.

[29] X. Kui, M. Zhang, Q. Liu, Z. Wang, G. Huang, Z. Zheng, J. Xia, and C. Zhang, "TSSeer: a visual analytics approach for exploring the correlation between teachers' multimodal emotions and students' behaviors in massive open online courses," *Journal of Visualization*, vol. 27, no. 4, pp. 749–764, 2024.

[30] N. Cao, C. Shi, S. Lin, J. Lu, Y.-R. Lin, and C.-Y. Lin, "TargetVue: Visual Analysis of Anomalous User Behaviors in Online Communication Systems," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 1, pp. 280–289, 2016.

[31] S. Liu, M. X. Zhou, S. Pan, Y. Song, W. Qian, W. Cai, and X. Lian, "TIARA: Interactive, Topic-Based Visual Text Summarization and Analysis," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 3, no. 2, 2012.

[32] T. Saito, H. Miyamura, M. Yamamoto, H. Saito, Y. Hoshiya, and T. Kaseda, "Two-tone pseudo coloring: compact visualization for one-dimensional data," in *Proceedings of IEEE Symposium on Information Visualization*, 2005, pp. 173–180.
[33] J. Heer, N. Kong, and M. Agrawala, "Sizing the horizon: the effects of chart size and layering on the graphical perception of time series visualizations," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2009, pp. 1303–1312.
[34] K. Xu, Y. Wang, L. Yang, Y. Wang, B. Qiao, S. Qin, Y. Xu, H. Zhang, and H. Qu, "CloudDet: Interactive Visual Analysis of Anomalous Performances in Cloud Computing Systems," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 1, pp. 1107–1117, 2020.
[35] L. Chen, C. Cheng, H. Wang, X. Wang, Y. Tian, X. Yue, W. Kam-Kwai, H. Zhang, S. Hong, and Q. Li, "FMLens: Towards Better Scaffolding the Process of Fund Manager Selection in Fund Investments," *IEEE Transactions on Visualization and Computer Graphics*, vol. 31, no. 7, pp. 4064–4080, 2025.
[36] A. V. Moere, "Time-Varying Data Visualization Using Information Flocking Boids," in *Proceedings of IEEE Symposium on Information Visualization*, 2004, pp. 97–104.
[37] Gapminder Foundation, "Gapminder Tools," 2025. [Online]. Available: https://www.gapminder.org/tools
[38] H.-J. Schulz, "Treevis.net: A Tree Visualization Reference," *IEEE Computer Graphics and Applications*, vol. 31, no. 6, pp. 11–15, 2011.
[39] T. Munzner, F. Guimbretière, S. Tasiran, L. Zhang, and Y. Zhou, "TreeJuxtaposer: scalable tree comparison using Focus+Context with guaranteed visibility," *ACM Transactions on Graphics*, vol. 22, no. 3, pp. 453–462, 2003.
[40] B. Lee, C. S. Parr, C. Plaisant, B. B. Bederson, V. D. Veksler, W. D. Gray, and C. Kotfila, "TreePlus: Interactive Exploration of Networks with Enhanced Tree Layouts," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 6, pp. 1414–1426, 2006.
[41] X. Y. Yan and Y. Ma, "Elastic Tree Layouts for Interactive Exploration of Mentorship," in *Proceedings of Eurographics Conference on Visualization*, 2021, pp. 1–3.
[42] J. B. Kruskal and J. M. Landwehr, "Icicle plots: Better displays for hierarchical clustering," *The American Statistician*, vol. 37, no. 2, pp. 162–168, 1983.
[43] B. Johnson and B. Shneiderman, "Tree maps: A Space-Filling Approach to the Visualization of Hierarchical Information Structures," in *Proceedings of the 2nd IEEE Visualization Conference*, 1991, pp. 284–291.
[44] B. Shneiderman, "Tree Visualization with Tree-Maps: 2-d Space-Filling Approach," *ACM Transactions on Graphics*, vol. 11, no. 1, pp. 92–99, 1992.
[45] J. T. Stasko and E. Zhang, "Focus+Context Display and Navigation Techniques for Enhancing Radial, Space-Filling Hierarchy Visualizations," in *Proceedings of IEEE Symposium on Information Visualization*, 2000, pp. 57–65.
[46] J. Yang, M. O. Ward, and E. A. Rundensteiner, "InterRing: An Interactive Tool for Visually Navigating and Manipulating Hierarchical Structures," in *Proceedings of IEEE Symposium on Information Visualization*, 2002, pp. 77–84.
[47] W. Wang, H. H. Wang, G. Dai, and H. Wang, "Visualization of large hierarchical data by circle packing," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2006, pp. 517–520.
[48] H. Zhao and L. Lu, "Variational circular treemaps for interactive visualization of hierarchical data," in *Proceedings of IEEE Pacific Visualization Symposium*, 2015, pp. 81–85.
[49] J. Görtler, C. Schulz, D. Weiskopf, and O. Deussen, "Bubble Treemaps for Uncertainty Visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 1, pp. 719–728, 2018.
[50] Y. Jin, T. J. A. de Jong, M. Tennekes, and M. Chen, "Radial Icicle Tree (RIT): Node Separation and Area Constancy," *IEEE Transactions on Visualization and Computer Graphics*, vol. 30, no. 1, pp. 251–261, 2024.
[51] F. B. Viégas, M. Wattenberg, J. Hebert, G. Borggaard, A. Cichowlas, J. Feinberg, J. Orwant, and C. R. Wren, "Google+Ripples: a native visualization of information flow," in *Proceedings of the 22nd International World Wide Web Conference*, 2013, pp. 1389–1398.
[52] G. Li, M. Tian, Q. Xu, M. J. McGuffin, and X. Yuan, "GoTree: A Grammar of Tree Visualizations," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2020, pp. 1–13.
[53] S. Jalali and C. Wohlin, "Systematic literature studies: database searches vs. backward snowballing," in *Proceedings of the ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*. ACM, 2012, pp. 29–38.
[54] T. Munzner, *Visualization Analysis and Design*. A K Peters, 2014.
[55] B. Lee, C. Plaisant, C. S. Parr, J.-D. Fekete, and N. Henry, "Task taxonomy for graph visualization," in *Proceedings of the AVI Workshop on Beyond Time and Errors: Novel Evaluation Methods for Information Visualization*, 2006, pp. 1–5.
[56] M. Brehmer and T. Munzner, "A Multi-Level Typology of Abstract Visualization Tasks," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2376–2385, 2013.
[57] Z. Deng, D. Weng, X. Xie, J. Bao, Y. Zheng, M. Xu, W. Chen, and Y. Wu, "Compass: Towards Better Causal Analysis of Urban Time Series," *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 1, pp. 1051–1061, 2022.
[58] T. Munzner, "A Nested Model for Visualization Design and Validation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 6, pp. 921–928, 2009.
[59] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of machine learning research*, vol. 9, no. 11, pp. 2579–2605, 2008.
[60] C. Nobre, M. Streit, and A. Lex, "Juniper: A Tree+Table Approach to Multivariate Graph Visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 1, pp. 544–554, 2019.
[61] M. Gleicher, "Considerations for Visualizing Comparison," *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 1, pp. 413–423, 2018.
[62] W. Javed, B. McDonnel, and N. Elmqvist, "Graphical Perception of Multiple Time Series," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 6, pp. 927–934, 2010.
[63] M. Gleicher, D. Albers, R. Walker, I. Jusufi, C. D. Hansen, and J. C. Roberts, "Visual comparison for information visualization," *Information Visualization*, vol. 10, no. 4, pp. 289–309, 2011.
[64] S. R. Aghabozorgi, A. S. Shirkhorshidi, and Y. W. Teh, "Time-series clustering - A decade review," *Information Systems*, vol. 53, pp. 16–38, 2015.
[65] M. Ester, H. Kriegel, J. Sander, and X. Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, 1996, pp. 226–231.
[66] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.
[67] T. Fu, "A review on time series data mining," *Engineering Applications of Artificial Intelligence*, vol. 24, no. 1, pp. 164–181, 2011.
[68] M. Bostock, V. Ogievetsky, and J. Heer, "D³ Data-Driven Documents," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2301–2309, 2011.
[69] R. Hyndman and G. Athanasopoulos, *Forecasting: Principles and Practice*, 3rd ed. OTexts, 2021.
[70] M. E. Fonteyn, B. Kuipers, and S. J. Grobe, "A description of think aloud method and protocol analysis," *Qualitative health research*, vol. 3, no. 4, pp. 430–441, 1993.
[71] J. C. Roberts, "State of the art: Coordinated & multiple views in exploratory visualization," in *Proceedings of the Fifth International Conference on Coordinated and Multiple Views in Exploratory Visualization*. IEEE, 2007, pp. 61–71.
[72] G. Andrienko and N. Andrienko, "Coordinated multiple views: a critical view," in *Proceedings of the Fifth International Conference on Coordinated and Multiple Views in Exploratory Visualization*, 2007, pp. 72–74.
[73] R. Chen, X. Shu, J. Chen, D. Weng, J. Tang, S. Fu, and Y. Wu, "Nebula: A Coordinating Grammar of Graphics," *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 12, pp. 4127–4140, 2022.
[74] P. Soni, C. de Runz, F. Bouali, and G. Venturini, "A survey on automatic dashboard recommendation systems," *Visual Informatics*, vol. 8, no. 1, pp. 67–79, 2024.
[75] S. Liu, D. Weng, Y. Tian, Z. Deng, H. Xu, X. Zhu, H. Yin, X. Zhan, and Y. Wu, "ECoalVis: Visual Analysis of Control Strategies in Coal-fired Power Plants," *IEEE Transactions on Visualization and Computer Graphics*, vol. 29, no. 1, pp. 1091–1101, 2023.
[76] J. Wang and K. Mueller, "DOMINO: Visual Causal Reasoning With Time-Dependent Phenomena," *IEEE Transactions on Visualization and Computer Graphics*, vol. 29, no. 12, pp. 5342–5356, 2023.
[77] S. Liu, Y. Tian, Z. Deng, W. Cui, H. Zhang, D. Weng, and Y. Wu, "Relation-Driven Query of Multiple Time Series," *IEEE Transactions on Visualization and Computer Graphics*, vol. 31, no. 8, pp. 4210–4225, 2025.

Lingyu Meng received his B.S. degree in geographical information science from Zhejiang University in 2023. He is currently pursuing a doctoral degree with the State Key Lab of CAD&CG, Zhejiang University. His research interests include spatiotemporal data mining and visual analytics.

Shuhan Liu received her B.S. degree in computer science from Zhejiang University in 2021. She is currently pursuing a doctoral degree with the State Key Lab of CAD&CG, Zhejiang University. Her research interests include spatiotemporal data mining, visualization, and industrial data visual analytics.

Keyi Yang received his B.S. degree in geographical information science from Zhejiang University in 2024. His research interests include spatiotemporal data mining and visual analytics.

Jiabin Xu received his B.S. degree in information security from Zhejiang University in 2024. His research interests include visualization and industrial data visual analytics.

Zikun Deng is a tenure-track associate professor at School of Software Engineering, South China University of Technology. He received his Ph.D. degree in Computer Science from Zhejiang University in 2023. His research interests mainly include visual analytics, visualization, data mining, and their application in smart city, industry 4.0, and digital twins. For more information, please visit https://zkdeng.org.

Di Weng is a ZJU100 Young Professor at the School of Software Technology, Zhejiang University. His main research interest lies in data visualization and human-computer interaction, focusing on human-AI collaboration for interactive data wrangling and analysis. He received his Ph.D. degree in Computer Science from the State Key Lab of CAD&CG, Zhejiang University. Before his current position, Dr. Weng was a researcher at Microsoft Research Asia from 2022 to 2023. For more information, please visit https://dwe.ng.

Yingcai Wu is a Professor at the State Key Lab of CAD&CG, Zhejiang University. His main research interests are information visualization and visual analytics, with focuses on sports science and social media analysis. He received his Ph.D. degree in Computer Science from The Hong Kong University of Science and Technology. Prior to his current position, Dr. Wu was a postdoctoral researcher in the University of California, Davis, from 2010 to 2012, and a researcher at Microsoft Research Asia from 2012 to 2015. For more information, please visit http://www.ycwu.org.