TSEditor: Interactive Time Series Editing for Privacy Preservation

Zihan Xu
State Key Lab of CAD&CG
Zhejiang University
Hangzhou, Zhejiang, China
zihanxu@zju.edu.cn

Shuhan Liu
State Key Lab of CAD&CG
Zhejiang University
Hangzhou, Zhejiang, China
shliu@zju.edu.cn

Kaicheng Shao
School of Software Technology
Zhejiang University
Ningbo, Zhejiang, China
kaichengshao@zju.edu.cn

Yuanzhe Jin
Department of Engineering Science
University of Oxford
Oxford, United Kingdom
yuanzhe.jin@eng.ox.ac.uk

Xumeng Wang
DISSec
Nankai University
Tianjin, China
wangxumeng@nankai.edu.cn

Zikun Deng
School of Software Engineering
South China University of Technology
Guangzhou, Guangdong, China
zkdeng@scut.edu.cn

Di Weng*
School of Software Technology
Zhejiang University
Ningbo, Zhejiang, China
dweng@zju.edu.cn

Yingcai Wu
State Key Lab of CAD&CG
Zhejiang University
Hangzhou, Zhejiang, China
ycwu@zju.edu.cn

Abstract

Publishing time series datasets raises substantial privacy concerns, as the underlying patterns (e.g., trends, values) can lead to the disclosure of individual identification. Mitigating these concerns remains challenging due to difficulties in pinpointing specific privacy-leaking patterns and protecting them without significantly compromising the analytical utility of the published data. Existing methods remain vulnerable to identity attacks utilizing diverse temporal patterns and may compromise data utility for subsequent analytical tasks. To address these limitations, we collaborated with domain experts to summarize a taxonomy of privacy risks in time series data and developed *TSEditor*, an interactive editing system. *TSEditor* integrates coordinated views for multi-perspective analysis of privacy risks and introduces six editing operations for targeted modifications, providing visual feedback. We demonstrate the effectiveness and usability of *TSEditor* through two case studies, an expert interview, a model evaluation, and a user study.

CCS Concepts

• **Human-centered computing → Interactive systems and tools**.

Keywords

Time series editing, privacy preservation, time series visualization

ACM Reference Format:
Zihan Xu, Shuhan Liu, Kaicheng Shao, Yuanzhe Jin, Xumeng Wang, Zikun Deng, Di Weng, and Yingcai Wu. 2026. TSEditor: Interactive Time Series Editing for Privacy Preservation. In *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems (CHI '26), April 13–17, 2026, Barcelona, Spain.* ACM, New York, NY, USA, 18 pages. https://doi.org/10.1145/3772318.3790885

*Di Weng is the corresponding author.

CHI '26, Barcelona, Spain

ACM ISBN 979-8-4007-2278-3/2026/04
https://doi.org/10.1145/3772318.3790885

1 Introduction

The sharing of time series data offers substantial analytical opportunities across various domains, such as healthcare [16, 42], energy [25], and transportation [85], enabling informed data-driven decision-making and facilitating innovation for temporal data research [31, 39, 100]. However, releasing raw time series datasets poses significant privacy risks, particularly when they involve human-related data, since such datasets may inadvertently reveal confidential temporal patterns [41, 52, 61]. For instance, household electricity consumption data may reveal personal habits, such as daily routines or appliance usage patterns [66]. To mitigate these risks, it is essential to process time series data with privacy-preserving techniques before publication, ensuring both data utility and individual confidentiality.

Traditional privacy-preserving methods for time series datasets involve treating them as time-varying multi-attribute records and applying classic privacy-preserving techniques [29] to remove sensitive information, such as identifiable attributes like names and addresses. However, individuals may still be identifiable through temporal patterns or by combining other information sources, leading to linkage attacks [69]. To address privacy risks from a temporal perspective, anonymization, such as erasing sensitive patterns, can be done manually using spreadsheet software or custom coding. However, these methods are labor-intensive, error-prone, and not scalable. Recent research has explored automatic methods [43, 66], often based on techniques like differential privacy [22]. However, these approaches may introduce excessive noise that obscures important patterns, offer insufficient protection, or lack controllable, interpretable ways of balancing data utility and privacy. Therefore, many time series datasets are withheld entirely to avoid privacy

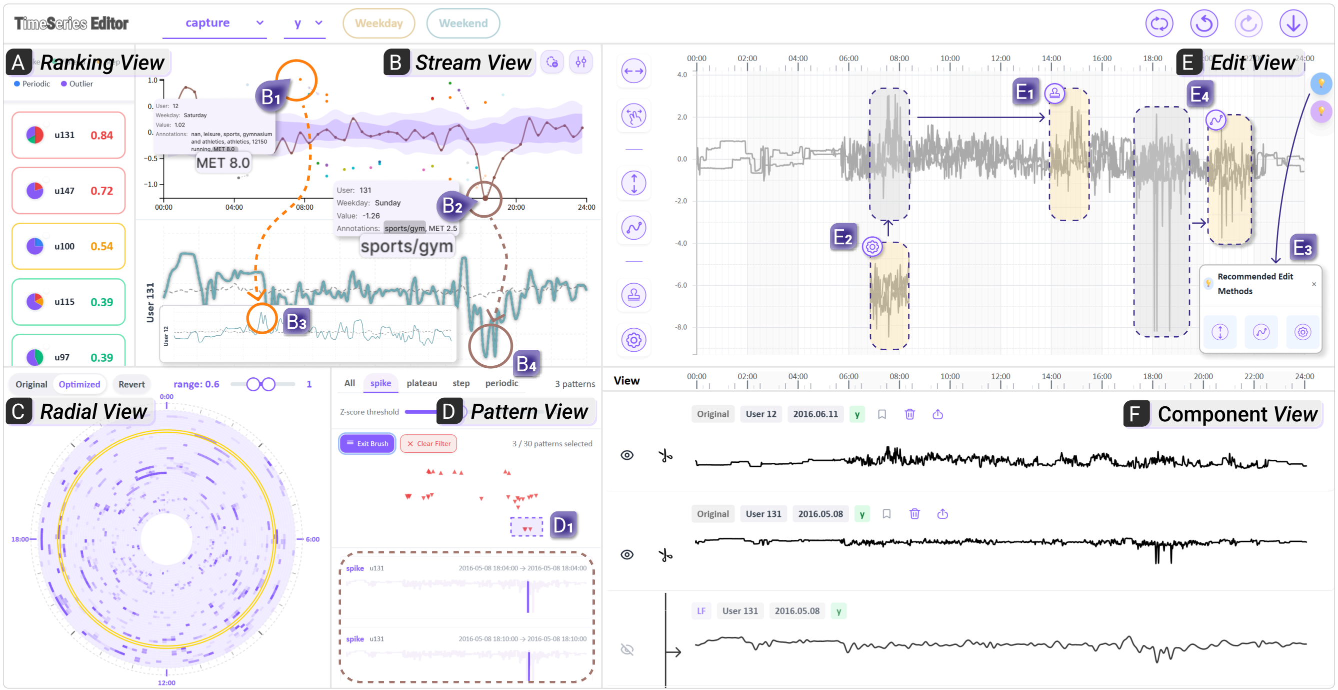

Figure 1: The interface of *TSEditor*. A) The ranking view guides user attention by ranking individuals according to a computed privacy risk score. B) The stream view identifies magnitude-based risks by visualizing outliers against the dataset's overall distribution, represented as a continuous streamgraph. C) The radial view reveals time-based risks by encoding user data into concentric rings, facilitating the discovery of cyclical patterns and synchronous behaviors. D) The pattern view automatically detects and visualizes pattern-based risks using a glyph-based scatterplot linked to a thumbnail list. E) The edit view provides a suite of direct manipulation tools for targeted risk mitigation, categorized into time perturbation, magnitude perturbation, and pattern substitution. F) The component view enables efficient management and decomposition of multiple time series.

risks, severely hindering their potential for collaborative analysis and societal benefit [61].

Motivated by the limitations of existing methods, we collaborate with time series data analysts to propose an **interactive**, **controllable**, and **interpretable** approach for detecting and addressing temporal privacy risks in time series datasets, enabling efficient and effective privacy preservation during the data publication process. Developing such an approach poses the following two challenges:

The Discovery Challenge: How to empower data owners to effectively discover diverse and subtle temporal privacy risks? A fundamental premise of our work is that privacy risks in time series primarily stem from data features that are "relatively unique" within the dataset. These threats manifest as anomalous magnitudes, unique temporal regularities, or distinctive sequential patterns that act as "behavioral fingerprints," enabling user re-identification. The core challenge lies in designing a visual analytics environment that guides human attention and empowers users to comprehensively identify these hidden threats from different, complementary perspectives.

The Mitigation Challenge: How to provide an intuitive and controllable framework for mitigating these risks without destroying data utility? Once a risk is identified, the challenge is to formulate an interactive mitigation framework that replaces rigid, one-size-fits-all algorithms. We need to equip experts with a flexible toolkit that allows for precise, targeted modifications, supports nuanced trade-offs between privacy and utility, and provides immediate feedback to build trust and confidence in the process.

To address these challenges, we propose *TSEditor*, a novel interactive system that helps data owners identify and mitigate privacy risks in time series datasets, thereby achieving more secure dataset publication. First, we built a taxonomy for temporal privacy risks and designed multi-facet visualizations to assist users in identifying these risks. Second, we adopt the paradigm of direct manipulation to develop an interactive time series editing framework that enables privacy risk mitigation through intuitive operations.

In summary, the contributions of this study are as follows:

- We characterized a systematic taxonomy of temporal privacy risks and a corresponding framework of interactive editing operations to mitigate these specific threats.
- We developed *TSEditor*, a novel visual analytics system that assists users in identifying and mitigating temporal privacy risks and demonstrated its effectiveness and usability through a multi-faceted evaluation.

2 Related Work

Our work is informed by two primary areas of research: time series privacy preservation and time series visual analytics. Accordingly, this section first reviews the landscape of privacy preservation, covering both common attack methods and existing defense techniques. We then survey relevant literature in visual analytics, focusing on established approaches for time series visualization and interaction.

2.1 Time Series Privacy Preservation: From Attacks to Defenses

Prior work has established that privacy risks in time series data often arise from their susceptibility to linkage attacks [63]. This class of attacks involves an adversary linking a supposedly anonymous dataset with external, auxiliary information to re-identify individuals or infer sensitive attributes. The core vulnerability lies in the fact that rich temporal data can form unique "*behavioral fingerprints*". Seminal works have shown that distinctive sequential patterns, such as those found in datasets of movie ratings [58], mobility traces [17, 71], smart meter readings [11], and web browsing history [60], are often unique enough to deanonymize individuals when correlated with other available data sources. The demonstrated feasibility of such attacks underscores the critical need for robust privacy-preserving techniques.

In recent years, numerous studies have been conducted on temporal privacy-preserving methods for time series data. These methods can be broadly categorized into three types based on their distinct approaches to processing original data: *value perturbation-*, *temporal perturbation-*, and *synthesis*-based methods.

Value perturbation methods add noise to the original data, which are widely used in the framework of Differential Privacy. To balance the utility of the data with the degree of privacy protection, Kellaris et al. [43] proposed three major privacy levels, namely event-level, user-level, and *w*-event level. At the event level, privacy safeguards are applied exclusively to individual elements within a time series, which reduces the consumption of the privacy budget but may fall short of providing comprehensive privacy protection for the entire dataset (e.g., Ren et al. [64], Jiang et al. [38] and Wang et al. [84]). User-level privacy provides comprehensive protection for all data tied to a user (e.g., Zhang et al. [96] and Gursoy et al. [30], yet maintaining acceptable utility involves consuming a larger privacy budget. A balanced approach, *w*-event level privacy, protects consecutive segments consisting of *w* consecutive timestamps to offer a compromise between the precision of event-level and the broad scope of user-level protection, thereby improving the overall privacy-utility trade-off (e.g., Wang et al. [82] and Zhang et al. [95]). However, these methods lack the fine-grained, interactive control for users to protect specific high-risk segments based on domain expertise and context.

Temporal perturbation methods redistribute elements across timestamps to obfuscate event timings. While value perturbation methods do not work in medical and financial applications where precision is critical, temporal perturbation avoids injecting noise, thereby significantly enhancing the accuracy of key time series statistics and manipulations [90]. Ye et al. [91] first proposed a temporal perturbation method for time series release, but met issues of missing or repetition. Although these problems can be avoided [90], the privacy level needs to be enhanced. Crucially, the process remains algorithmically driven, preventing users from making semantically meaningful modifications.

Synthesis methods generate fake data that exhibits distributional characteristics similar to the original data. In the existing literature, time-series synthesis is primarily dominated by generative adversarial networks (GANs) (e.g., C-RNN-GAN [54], TimeGAN [92], and RCGAN [26]). Given the instability of GANs, some researchers have explored other generative methods. For example, Desai et al. [19] developed TimeVAE, with interpretable temporal structures, showing strong performance in time series synthesis. Yuan et al. [93] propose Diffusion-TS, a diffusion-based framework that generates time series satisfying both interpretability and realness. However, the interpretability of a model's output structure is distinct from the process-level transparency and control needed for interactive work. As a "*black box*", the long and opaque generation process of these models fundamentally lacks real-time feedback and direct control required for an effective, interactive editing workflow.

In essence, while approaching the problem from different technical standpoints, these automated methods share a fundamental limitation: they treat privacy preservation as a monolithic, algorithmic process rather than a nuanced, user-driven task. They consistently lack the interactive control, interpretability, and real-time feedback that would empower domain experts to make granular, context-aware judgments. This failure to place the user in the decision-making loop reveals a critical gap for a new paradigm based on interactive editing.

2.2 Time Series Visual Analytics: From Data Exploration to the Need for Editing

The visual analytics of temporal data plays a pivotal role in diverse application domains, such as smart factory [48, 74], clinical treatment [80, 97], E-transaction analysis [49, 86], and urban planning [50, 94]. In recent years, a wide array of advanced techniques for analyzing such data have been proposed and published, spanning both visualization and interaction methodologies [1–3].

Time series visualization. Line graphs [62] have become the most classic method for visualizing temporal data, and there are also some variations based on this approach, such as time curves [5]. When visualizing multiple time series, existing solutions, as identified by Javed et al. [37], can be classified into two broad categories: *shared-space* (e.g., simple line graph [62, 68] and stacked area chart [99]) and *split-space* (e.g., small multiples [67, 77] and horizon graphs [65]). Systems such as TimeSearcher [13], GeoChron [18], and KD-Box [98] widely adopt shared-space layouts because they outperform split-space layouts in identifying overarching trends and patterns in large-scale time series data. However, the dense superposition of lines in shared-space layouts can sometimes obscure finer details and introduce visual clutter. To address these challenges, dimensionality reduction [78] and clustering [27] techniques have emerged as powerful alternatives for visualizing large-scale time series data [4, 55, 73]. Many advanced visual analytics systems (e.g., ECGLens [88], StreamStory [72], and Van Wijk et al. [79]) have embedded dimensionality reduction and clustering views to assist in data analysis, enabling users to explore patterns, trends, and relationships more effectively. However, they are not specifically designed to support the discovery of privacy leakage patterns through visualization.

Time series interaction. The prevailing interaction techniques applied to time series data are primarily intended to support more effective and intuitive exploration of the underlying information [81]. As summarized by Aigner et al. [3], the basic interaction operations are systematically categorized into four principal categories: *temporal navigation*, *direct manipulation*, *brushing & linking*, and *dynamic queries*. Navigation in time, a fundamental aspect of temporal interaction, is often facilitated by dedicated sliders [76]. DimpVis [44] uses direct manipulation to facilitate navigation to points in time. TimeSplines [59] enables pen-based direct manipulation for expressive timeline authoring. TimeTuner [32] and ViDX [89] empower the selection and analysis of targeted data intervals through brush & linking. Qetch [51] and RelaQ [47] provide sketch-based techniques to support time series queries. We also investigated several works whose interaction objectives align closely with our research. For example, TimeSeriesMaker [6] leverages a tree structure to enable time series generation through a template-based configuration.

Visual data editing. Beyond specific time series techniques, our work is situated within the broader context of visual data editing and its intersection with data wrangling, where editing serves as the core operation for transformation. This field has demonstrated *visualization-based editing* for a diverse range of data types, including tabular data [40, 87], graph data [7, 23, 35], and textual data [36, 53], demonstrating the value of tightly coupled visual and editing interfaces. However, to the best of our knowledge, no prior system integrates these general visual editing principles with privacy preservation specifically for addressing the unique analytical and structural challenges of time series data. It is this specific gap that *TSEditor* aims to fill.

3 Background and Task Abstraction

This section presents a motivating scenario, the taxonomy of temporal privacy risks and corresponding editing solutions in time series, and the summarized user requirements for system design.

We collaborated closely with four experienced domain experts (i.e., EA, EB, EC, and ED) for a year to address the challenge of privacy protection for time series data. EA and EB are researchers with over five years of experience in time series data analysis. EC, a data scientist, has participated in multiple projects focused on privacy protection. ED, a Ph.D. candidate specializing in quality assurance for machine learning applications, frequently engages in dataset publication as part of his research workflow. During the collaboration, we specifically focus on **temporal privacy risks**, i.e., sensitive information, which could possibly be leveraged as quasi-identifiers to link individuals with specific time series, leaking from the temporal perspective rather than the metadata associated with time series. We explore the critical question of how sensitive information can be potentially extracted from time series datasets and propose an interactive approach that safeguards against these temporal privacy risks. Our analysis primarily targets univariate time series, and partially supports multivariate time series via a "multiple univariate" strategy.

3.1 Motivating Scenario

Imagine a healthcare company releases an anonymized dataset of users' daily step counts. An adversary intends to re-identify a specific executive known for their distinctively high activity levels. Using public knowledge, the adversary executes a linkage attack exploiting three temporal vulnerabilities:

First, knowing the executive averages over 20,000 steps daily, the adversary filters the dataset for this distinct **magnitude**. This simple query isolates a small subset of users, including the target, "User X," from thousands of records.

Next, the adversary refines this subset by looking for a unique **pattern**. Public interviews revealed the executive's routine: early morning runs on weekdays and long walks on Saturday afternoons. The adversary matches this exact repeating sequence in User X's data, which serves as a recognizable behavioral fingerprint.

Finally, to confirm the identity, the adversary leverages abnormal **timing**. Knowing the executive participated in a local 10K race last Sunday morning, they verify that User X's data exhibits a specific activity spike aligning perfectly with the race's timeframe. By combining these magnitude, pattern, and time-based risks, the adversary de-anonymizes the individual, highlighting the critical need for tools to identify and mitigate such privacy threats.

3.2 A Taxonomy of Temporal Privacy Risks and Mitigation Strategies

To establish a systematic taxonomy of temporal privacy risks, we conducted a thorough literature review and surveyed public time series datasets. These findings were then refined through iterative discussions with domain experts to ensure practical relevance.

This taxonomy identifies three core types of temporal features that most directly leak sensitive information. These features, and the risks they expose, are defined as follows. For each feature's associated risk, we proposed a corresponding interactive mitigation strategy that allows users to make targeted edits.

- ***Time***: Risks in this category arise from the abnormal timing of events. The threat here is not necessarily the activity itself, but its specific temporal placement, which often represents a marked departure from typical behavior. Such unique temporal placements can act as powerful identifiers, especially when correlated with external information.
- ***Magnitude***: This category covers risks exposed by distinct magnitudes, where the data values themselves are statistical outliers. The threat lies in the presence of anomalous values, such as extreme single-point spikes or sustained plateaus, which are identifiable because of their significant deviation from the norm.
- ***Pattern***: This category involves risks from unique sequences, where the threat lies in the distinctive shape of the sequence. This shape, formed by a complex arrangement of data points over time, acts as a behavioral fingerprint even if the individual values within it are not statistically extreme. This includes unique routines or specific, one-time events recognizable by their characteristic shape or temporal signature.

We investigated solutions to balance privacy risks with analytical utility. Since automated approaches often lack the contextual awareness to balance these objectives, a human-in-the-loop process is essential. We therefore developed an interactive framework with three classes of mitigation operations This framework enables direct manipulation of sensitive temporal features, ensuring the dataset's integrity while addressing privacy concerns.

- To address abnormal timing, the ***time perturbation*** operations modify the time points by shifting timestamps or stretching time segments, ensuring that the positions of sensitive patterns are obscured while retaining the overall trend of the data for analysis.
- To address distinct values, the ***magnitude perturbation*** operations can be applied to vertically shift data segments or remap values through interactive curves, concealing unique fluctuations without fundamentally distorting the overall data distribution.
- To address unique sequences, the ***pattern substitution*** operations can be employed to overwrite sensitive temporal segments with common time segments from other parts of the dataset, effectively obscuring sensitive patterns while preserving the authenticity and analytical value of the data.

3.3 Requirement Analysis

To guide the design of *TSEditor*, we followed the nested model for visualization design and validation [56]. Over a three-month collaborative process, we first characterized the experts' workflows and challenges, which informed our initial low-fidelity prototypes. Through multiple cycles of prototype reviews and feedback, we refined the system's functionalities and consolidated the experts' needs into four key requirements.

R1: Locate privacy-leaking time series and temporal segments from different perspectives. Experts need to find time series and temporal segments with temporal privacy risks. A clear and multi-facet overview of the dataset is requested to understand its overall structure, identify unusual patterns, and detect potential privacy risks. For instance, vulnerabilities in both *time* and *magnitude* features can emerge within a given interval. Additionally, multi-scale comparative analysis and selective filtering are essential. Experts need to find unique behavioral patterns by comparing time series between weekdays and weekends or among varying granularities (e.g., *daily, weekly, annual*).

R2: Edit time series interactively to mitigate temporal privacy risks. After the privacy-leaking time series or segment is identified, experts need to address the risks by editing the temporal features. For each type of temporal feature, the experts require intuitive and flexible interactions to assist them in performing the corresponding operations. For the time features, the proposed system should facilitate adjustments along the time axis, such as delaying critical events or redistributing intervals. For the magnitude features, magnitude perturbation operations, such as modifying peak values or scaling amplitudes, are needed to adjust privacy-sensitive segments. For the pattern features, experts require an approach to obfuscate the original sensitive patterns.

R3: Apply consistent modifications across multiple time series[1] with similar privacy risks. When dealing with large datasets, experts often encounter multiple time series that exhibit similar privacy-sensitive patterns. Moreover, a time series can be decomposed into multiple series for targeted editing, such as separating into low- and high-frequency signals. In such cases, they require intuitive tools to flexibly visualize and compare relevant time series, and the ability to apply consistent modifications across selected groups in a single operation. This ensures temporal coherence and minimizes manual effort while maintaining data integrity.

R4: Assess the impact of user modifications on data privacy and distribution. To ensure informed decision-making, Experts need to evaluate how their editing operations affect the privacy risks (e.g., re-identification potential) and statistical properties (e.g., outlier distribution) of the dataset. The interface should dynamically update these visual feedback mechanisms as modifications are applied, allowing experts to observe trade-offs and unintended consequences (e.g., over-perturbation skewing trends) This enables iterative refinement, where users can evaluate the effectiveness of their changes and make informed adjustments to achieve the desired balance between privacy protection and data utility.

4 TSEditor

To address the requirements outlined in Sec. 3.3, we developed *TSEditor*, a visual analytics system for the visual identification and interactive mitigation of privacy risks in time series data. *TSEditor* incorporates six interconnected views (Fig. 1): (A) the ranking view, (B) the stream view, (C) the radial view, (D) the pattern view, (E) the edit view, and (F) the component view. The workflow on *TSEditor* has three phases: identification, protection, and evaluation (Fig. 2). Every view corresponds to a specific workflow phase. The ranking, stream, radial, and pattern views are dedicated to supporting the identification and evaluation phases, providing detailed visual representations to identify privacy-exposing segments (**R1**) and evaluate the impact of adjustments (**R4**). Accordingly, the edit view and component view are specialized for the protection phase, empowering users to make targeted modifications of multiple time series (**R3**) to address privacy concerns using time perturbation, magnitude perturbation, and pattern substitution approaches (**R2**).

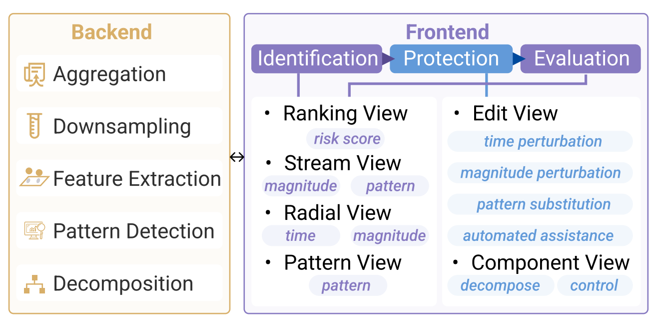

Figure 2: The system overview of *TSEditor*. The architecture consists of a backend for data processing and a frontend with a three-stage workflow: identification, protection, and evaluation. The frontend integrates multiple coordinated views, including ranking, stream, radial, and pattern views for identifying privacy risks and assessing the impact of edits, as well as edit and component views for applying targeted edits and managing multiple time series, thereby supporting users throughout the complete editing pipeline.

[1]Our system is evaluated on hundreds of time series.

4.1 Temporal Privacy Risk Identification

To enable the effective discovery of diverse temporal privacy risks across different feature types (i.e., time, magnitude, and pattern) (**R1**), the identification phase relies on a data processing pipeline. This pipeline processes raw time series using techniques such as data aggregation and pattern detection to compute risk metrics and identify distinctive features. These processed outputs serve as the data foundation for four coordinated views.

4.1.1 Ranking View. To prioritize analyst attention within large-scale datasets, which is a core challenge of risk identification (**R1**), this view serves as an auxiliary analytical starting point, in addition to direct exploration in other views. Since detailed visualizations of the entire population can be overwhelming, analysts require guidance to quickly navigate to the most potentially critical data within a large collection of time series. We implement a risk quantification pipeline. This framework is grounded in established time series analysis principles [9, 70] and refined through expert discussions. Formal mathematical definitions and parameter configurations for all measures are provided in the supplementary material. For each individual, the system first calculates raw severity scores for five types of anomalies: magnitude outliers identified in the stream view, and four temporal motifs (spikes, plateaus, steps, and periodic behaviors) detected in the pattern view. For instance, the severity of magnitude anomalies is quantified by the total count of detected outliers, whereas plateaus are evaluated based on a duration-to-stability ratio. Given the disparate scales of these raw metrics, they are normalized and aggregated using a weighted sum model to produce a unified "risk score." In the current implementation, weights are pre-configured based on expert feedback to prioritize high-impact anomalies (e.g., spikes). This quantification dynamically orders the list, ensuring individuals exhibiting the most distinct behavioral fingerprints are surfaced immediately for inspection.

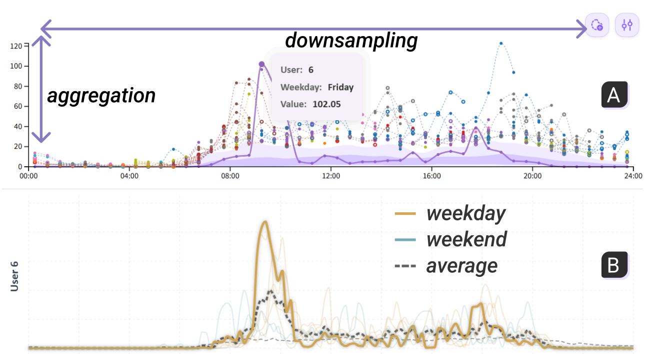

Figure 3: The stream view for magnitude- and pattern-based risk identification. A) The composite overview visualizes the dataset's distribution and outliers. B) The detail view enables granular inspection of a selected individual.

4.1.2 Stream View. Statistical outliers in time series can expose critical privacy vulnerabilities, manifesting as either anomalous *"magnitude"* or distinctive *"patterns"* characterized by abnormal values. To address the challenge of identifying these specific risks across the entire dataset (**R1**), the stream view implements a statistical data processing pipeline designed to highlight deviations against the population norm. This view consists of two components: a composite overview (Fig. 3A), enhancing traditional box plot principles to highlight outliers against a continuous data distribution, and a detail view (Fig. 3B) for individual series inspection.

To effectively visualize the entire dataset in the overview, we first perform a two-stage data preprocessing. First, to capture representative daily behaviors, we aggregate the data by the *day of the week*, calculating the average value for each time point across all weeks for each individual. Second, we apply temporal downsampling by averaging every k consecutive points, which reduces visual clutter and accommodates display limitations. It is important to note that the current default settings are specifically optimized for human-centric datasets. However, the underlying processing pipeline is designed to be scale-agnostic. To accommodate data with different inherent frequencies, users can configure the aggregation granularity (e.g., daily, monthly, or annual) to match the relevant temporal scale. Similarly, the downsampling factor (k) is fully adjustable, allowing analysts to fine-tune the resolution balance.

Using this condensed data, the **overview** visualizes magnitude-based risks through a design inspired by box plots. For each timestamp, we first compute key statistical percentiles. The underlying data distribution is then rendered as a streamgraph with two nested bands: a darker inner band for the Interquartile Range (IQR), and a lighter outer band for the 10th–90th percentiles. Overlaying this stream, data points falling outside a configurable range (default $Q1/Q3 \pm 1.5 \times IQR$) are rendered as outliers, using filled dots for weekdays and outline dots for weekends to differentiate temporal categories. Also, outliers are linked across time if they belong to the same individual and weekday, helping highlight both isolated points and sustained deviations.

Complementing the overview, the **detail view** displays the time series aggregated by weekday for each individual. This allows users to move from identifying an outlier to performing a granular inspection of the specific individual's behavior. The navigation bar offers further flexibility, allowing users to adjust the time granularity and filter (e.g., weekdays vs. weekends) for deeper analysis.

Interaction. The view supports a seamless analytical workflow centered on outlier investigation. Hovering over an outlier simultaneously highlights its corresponding time series in the overview and navigates the detail view to the corresponding individual, providing immediate context for comprehensive analysis.

Justification. While many techniques exist for visualizing multiple time series (Sect. 2.2), conventional methods proved inadequate for our task. We initially explored simple line graphs [62] and small multiples [77]. Given the data's scale, line graphs suffered from significant visual clutter [24], while small multiples lacked enough space for detailed representation. An attempt to cluster simple line graphs hindered traceability back to the original data. In contrast, our composite design retains the statistical power of box plots for robust outlier detection while embedding these critical points within a continuous representation of the data's temporal flow, which is crucial for holistic analysis.

4.1.3 Radial View. The radial view is designed to analyze privacy risks related to temporal characteristics (*"time"*) and quantitative

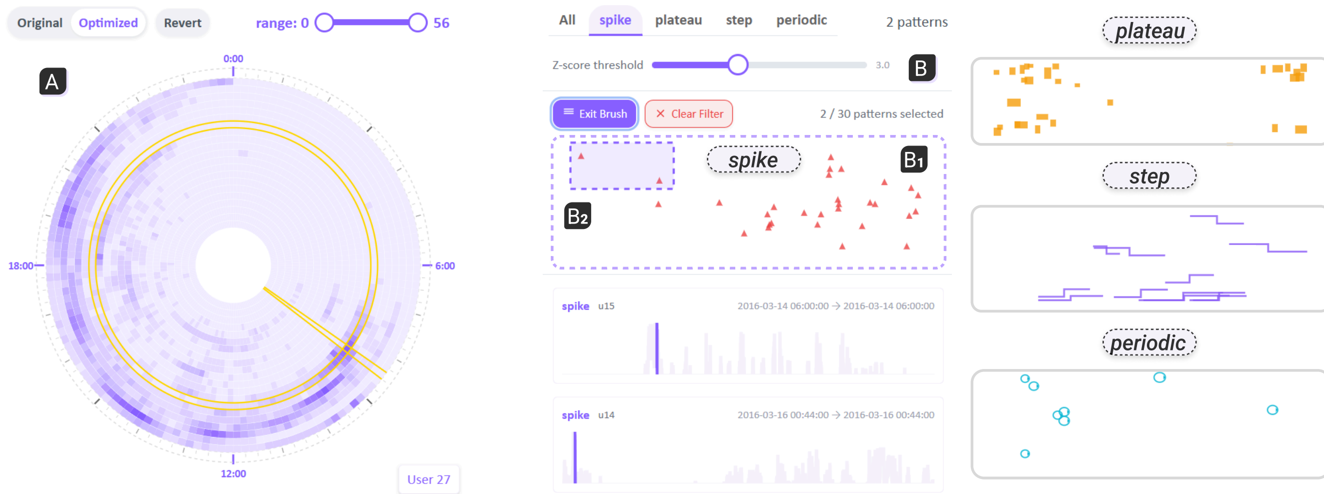

Figure 4: The radial view and pattern view. A) The radial view reveals time- and magnitude-based risks via interactive color mapping and layout reordering. B) The pattern view detects four specific motifs, visualizing pattern distribution and intensity through a glyph-based overview linked to a detailed list.

values (*"magnitude"*) (**R1**). Each individual's aggregated series is represented as a concentric ring in a radial stack (Fig. 4A). The core interaction is a dual-highlighting mechanism enabling multi-perspective analysis: hovering over any time segment highlights both the corresponding individual's entire ring (a longitudinal view of their individual trend) and the same time slice across all individuals (a cross-sectional view of population behavior). For fine-grained individual characterization, a drill-down interaction enables users to double-click a ring, switching into intra-individual mode, where each concentric ring represents a single day's data for selected individual. By mapping time to angular positions, the view facilitates the identification of periodic fluctuations and temporal trends.

Color mapping adjustment. The default color mapping, normalized by global extrema, is often skewed by outliers, masking subtle data variations. We therefore introduce a range slider that allows users to interactively adjust the boundaries of the color scale. Guided by the data distribution in the stream view and domain expertise, users can refine the color mapping to enable clearer identification of meaningful patterns.

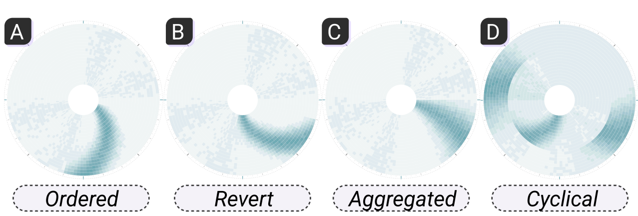

Figure 5: The layout optimization of the radial view. A) The ordered arrangement of individual rings clusters similar patterns together. B) The revert operation allows users to swap the order of the inner and outer rings. C-D) The layout optimization helps reveal significant trends, including aggregated data patterns and cyclical behaviors.

Layout reordering. By default, individual rings are arranged sequentially by ID, which scatters similar patterns and hinders the detection of group behaviors. To address this, we employ a similarity-based sorting strategy adapted from RidgeBuilder [46]. While standard clustering [79] effectively groups similar time series, it does not inherently solve the linear ordering problem. This algorithm, proposed in RidgeBuilder, orders individuals into a continuous visual gradient by explicitly minimizing local discontinuities between adjacent neighbors, thereby revealing visually distinct sectors of synchronized activity (Fig. 5A). Additionally, a revert operation allows users to invert the sorting order to mitigate perceptual bias from varying ring circumferences (Fig. 5B). This interactive reordering is critical for revealing subtle structures: it transforms scattered noise into coherent visual blocks, facilitating the identification of aggregated group behaviors (Fig. 5C), or creating spiral-like wave patterns that expose time-shifted cyclical trends across the population (Fig. 5D).

Justification. Radial charts offer distinct advantages for visualizing temporal patterns and periodic features [14, 21]. Their circular layout inherently mirrors the continuous, cyclical nature of time-series data, enabling intuitive pattern recognition. While calendar- and heatmap-based designs are viable alternatives, the radial view is chosen due to its flexibility in adjusting the granularity of time, allowing users to examine series at various time scales and identify periodic behaviors more effectively. Radial line charts were also considered but dismissed for their space inefficiency and the visual clutter they produce with large datasets.

4.1.4 Pattern View. This view automates the discovery of "*pattern*"-based risks (**R1**) by identifying four temporal motifs: *spikes*, *plateaus*, *steps*, and *periodic* behaviors. The detection relies on a set of statistical rule-based detection algorithms operating on raw time series data. For instance, point anomalies like spikes are identified using Z-score standardization to flag deviations exceeding statistical thresholds, while sustained motifs like plateaus are evaluated based

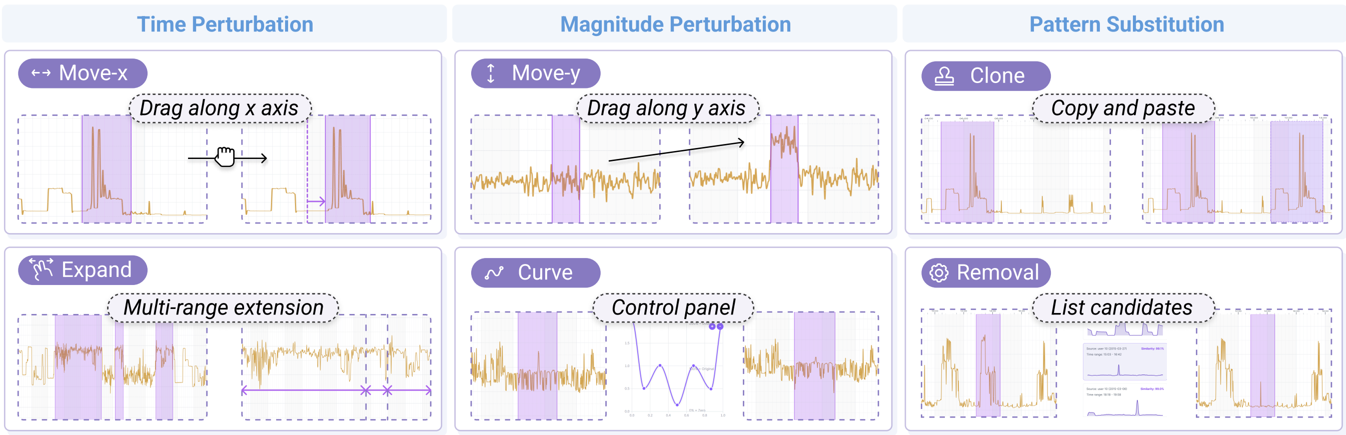

Figure 6: The interaction of *TSEditor*. *TSEditor* supports six interactive editing operations across three privacy-preserving strategies. *Time perturbation* includes Move-x, which allows users to horizontally shift time segments, and Expand, which stretches selected segments to the entire timeline. *Magnitude perturbation* provides Move-y, which vertically shifts value segments, and Curve, which enables value remapping through control-point-based transformation. *Pattern substitution* includes Clone, which copies and pastes segments from the same individual, and Removal, which replaces sensitive segments with candidates retrieved from the dataset.

on a duration-to-stability ratio, where longer segments with lower rolling variance yield higher significance scores. Detected patterns are visualized in two components (Fig. 4B): a glyph-based overview and a detail list. The overview maps each instance to a unique glyph (e.g., a triangle for a spike shown in Fig. 4B_1), with position encoding time and magnitude to reveal distribution clusters. Below, a card list displays a thumbnail line chart , providing visual evidence and context for verifying specific patterns.

Interaction. The glyph overview and the card list are tightly coordinated to support a flexible drill-down analysis with range-based filtering and single-item selection. Brushing a region of interest (Fig. 4B_2) performs a targeted selection (e.g., isolating all high-magnitude spikes in the morning), which instantly filters the card list to that corresponding subset. For direct inspection, clicking a single glyph immediately scrolls the list to and highlights its specific card. Such interaction facilitates both broad exploration of pattern groups and rapid examination of individual, high-risk instances.

4.2 Time Series Editing

4.2.1 Edit View. When users select one or more time series for processing, their workflow centers on the edit view. Corresponding to the privacy risk taxonomy established in Sec. 3.2, we implemented three editing categories (**R2**): 1) **time perturbation** for *"time"*, 2) **magnitude perturbation** for *"magnitude"*, and 3) **pattern substitution** for *"pattern"*. Each of them contains both fundamental and advanced functional operations. Fig. 6 presents the six corresponding operations, along with their interactive mechanisms and application effects.

↔ **Move-x.** This operation provides controllable temporal displacement of selected time series segments along the time axis. Through intuitive drag-and-drop interaction, users can adjust the temporal positioning of data segments while preserving original value distributions.

Expand. This operation allows users to perform temporal scaling on selected segments while maintaining temporal continuity. Users first select multiple temporal segments to preserve and then dynamically expand them to span the whole duration of the original series. This is effective for obscuring the precise length and timing of a sensitive event while retaining its relative shape. Any data outside the selected segments is discarded as unwanted.

↕ **Move-y.** Similar to Move-x, this operation enables vertical displacement of the selected segments along the value axis. Users can drag the series to mitigate magnitude-based risks by shifting an anomalous value range to be more consistent with the norm, while maintaining its temporal sequence.

Curve. This operation enables fine-grained remapping of data values through a control panel. In this panel, a mapping curve defines the transformation, where the x-axis represents the original values and the y-axis represents the new, transformed values. By dragging the curve's control points or using predefined templates (e.g., half-value), users can precisely reshape the value distribution, such as selectively reducing only extreme spikes.

Clone. This operation enables pattern replication from source intervals to target temporal locations, analogous to image editing's copy-paste functionality. By reusing real patterns from the same individual, the modified series preserves plausible characteristics while obscuring sensitive temporal behaviors.

Removal. This data-driven feature replaces a sensitive pattern by retrieving plausible alternatives in the dataset. Its replacement candidates are deliberately identified by matching the start and end-point values of a segment. This ensures numerical continuity while providing a diverse set of alternative patterns from which users can select a substitute that effectively masks the original pattern.

Additionally, a **redo/undo** feature is provided to ensure flexible editing experiences. This functionality maintains an exportable operation history, allowing users to revert any modification step-by-step during the editing process.

Automated Assistance. To further streamline the editing workflow, *TSEditor* provides two semi-automated assistance features. First, when a time series is imported into the edit view, the system analyzes its source and data characteristics and suggests tools for the user (e.g., the curve tool for a series with extreme outliers). Second, after a user manually edits a single component, they can opt to batch-apply that same transformation to all other unedited components. These features serve to guide users and significantly reduce repetitive effort in large-scale editing scenarios.

Interaction. Selection constitutes a fundamental prerequisite for all editing operations. Our implementation incorporates an interactive brushing feature that enables the precise selection of single or multiple segments for targeted modification. The system primarily adopts direct manipulation as its core interaction paradigm, allowing users to modify time series through intuitive operations while receiving real-time feedback. This direct manipulation is consistent with human interaction intuition, thereby improving learnability and reducing cognitive load during operation.

4.2.2 Component View. The component view is designed to address the challenge of applying consistent modifications across multiple time series (R3) and enabling fine-grained control over signal attributes.

Multi-Series Management. At the top level, this view manages the dataset as a collection of independent time series units. It supports batch editing, allowing users to select multiple series simultaneously (e.g., a group of individuals exhibiting similar high-risk patterns). Any editing operation performed in the edit view is consistently applied to all selected units, ensuring coherent sanitization across the dataset and significantly reducing manual effort.

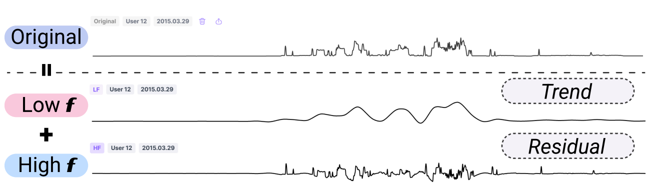

Figure 7: Time series decomposition in the component view. The original time series can be decomposed into low-frequency (trend) and high-frequency (residual) components. This separation enables granular control over signal attributes, allowing users to apply targeted modifications to specific privacy risks without unintentionally affecting other data features.

Time Series Decomposition. To enable granular privacy editing, the view allows users to decompose a time series into distinct signal components, specifically a low-frequency trend and a high-frequency residual (Fig. 7). This feature addresses the limitation of monolithic editing, where modifying a risk in one aspect might inadvertently distort other valuable data features. By separating these components, users can apply specialized strategies. For example, they can smooth a sensitive trend while preserving the realistic texture of the high-frequency residuals, or conversely, remove distinct micro-patterns from the residuals without altering the overall behavioral trend.

4.3 Interaction between Views in TSEditor

All views are tightly linked to support multi-perspective privacy risk analysis and coordinated editing across time series. When users hover over any element in one view, the other views automatically update to display corresponding data and highlight relevant elements in sync [12]. This cross-view interaction allows users to validate insights through multiple visual representations. Users can drag relevant elements from the identification views to directly transfer time series data to the editing suite. The component view offers fine-grained control over data selection and modification for the edit view, enabling customization based on users' analytical needs. Once satisfied with the current stage of editing, users explicitly trigger global updates via a synchronization button to propagate changes to all identification views (**R4**). This on-demand synchronization is designed to prevent distracting layout shifts and performance latency during iterative micro-edits, ensuring a stable context for assessment.

4.4 Implementation

We employed modern frontend (Vue, TailwindCSS, and Pinia) and backend (Flask, Python, and NumPy) libraries to develop *TSEditor*. The visualization and interaction components are implemented using D3.js [8]. The processed time series datasets used in our evaluation and the source code of *TSEditor* are publicly available on GitHub[2]. Additionally, we will deploy *TSEditor* as an online system to support broader adoption and facilitate secure data sharing.

5 Case Study

To comprehensively evaluate the effectiveness and usability of *TSEditor*, we structured our evaluation around three key research questions:

- RQ1 (Identification): Can risks be identified using *TSEditor*?
- RQ2 (Mitigation): Can risks be mitigated using *TSEditor*?
- RQ3 (Assessment): Can mitigation impact be assessed using *TSEditor*?

We addressed these questions through a multi-faceted approach spanning two sections. In this section, we report two case studies (RQ1–RQ3) with domain experts and a quantitative model evaluation (RQ3) to demonstrate the system's depth in handling real-world tasks. In the following section, we present a user study (RQ1, RQ2) to assess the system's accessibility and workflow efficiency for a broader range of users.

Study Protocol. The two experts who conducted the case studies, EC (a data scientist in privacy preservation) and ED (a machine learning researcher), had collaborated closely with us during the design process, helping to define the system's requirements (as detailed in Sec. 3). Due to their deep familiarity with the system's conceptual model, visualizations, and interactions, they did not require additional training. Each expert participated in a 40-minute session for their respective case. The task was exploratory, with a

[2] https://github.com/fickleee/TSEditor

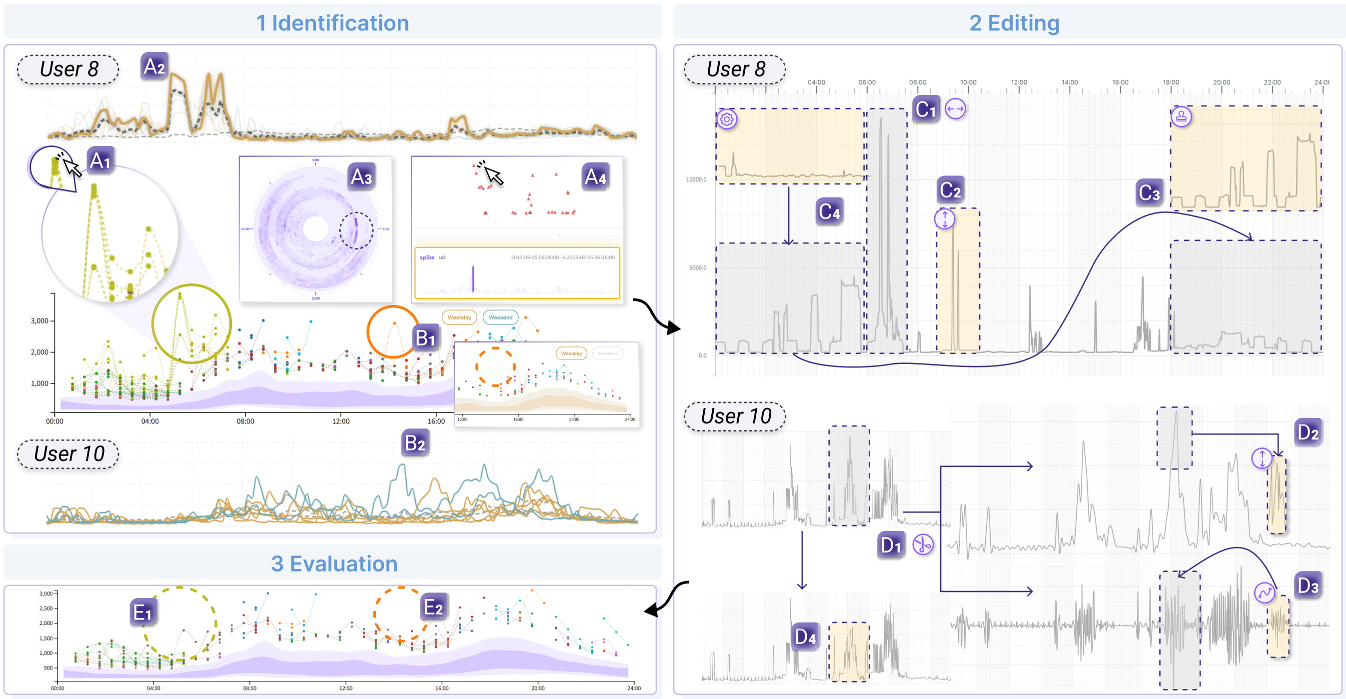

Figure 8: The analysis and editing process of the REFIT Electrical Load Measurements Dataset. A) Identifies abnormal nighttime electricity consumption patterns. B) Detects weekend routine behaviors by comparing weekday and weekend consumption patterns. C) Modifies the nighttime abnormal behavior, obscuring sensitive information while preserving data integrity. D) Refines the weekend routine behavior by time series decomposition without distorting the essential patterns. E) Evaluates the effectiveness of the edits.

high-level objective: "*Use TSEditor to identify and mitigate any patterns you believe could pose a privacy risk, with the goal of preparing the dataset for safe public release.*" During the sessions, the experts were encouraged to follow a think-aloud protocol [28], and the role of the authors was strictly that of passive observers, only intervening to resolve technical issues.

5.1 Case 1: REFIT Electrical Load Measurements Dataset

Case Setting. The REFIT electrical load measurements dataset [57] we used comprises electricity consumption measurements collected from 20 households in Loughborough. Raw data, recorded at 6-8 second intervals, were downsampled to a uniform 1-minute resolution for consistency. The processed data for March 2015 was used, due to discrepancies in time coverage and missing values. The final dataset comprises 864,000 records (22.7 MB) with aggregated active power (Watts) as the primary metric. The objective of implementing privacy-preserving processing on this dataset is to facilitate its secure publication for downstream time series *prediction* tasks.

Anomalous night-time electricity consumption. After loading the data, EC began her analysis in the ranking view, where she was drawn to the user with the highest computed risk score. Selecting this user, her attention was caught by a chain of prominent outliers in the stream view (Fig. 8A_1). Hovering over one of these points revealed they all belonged to User 8, whose detailed consumption curve was automatically displayed below (Fig. 8A_2). EC was surprised to find a highly anomalous **magnitude-based** risk: sharp consumption peaks between 5:00 and 7:30 AM, with virtually no usage during typical evening hours. This stark deviation from normal behavior was corroborated in other identification views: User 8's arc in the radial view stood out with an intense color saturation (Fig. 8A_3), while the glyph that deviates the farthest under the spike tag in the pattern view belongs to User 8 (Fig. 8A_4, RQ1).

EC determined this unique, reversed-schedule pattern constituted a high re-identification risk and dragged User 8's data into the edit view for mitigation. Her primary concern was the time-sensitive nature of the data, but completely removing the morning peak would sacrifice valuable information. She therefore opted for a more nuanced approach. First, she used the *move-x* tool to shift the entire 5:00-7:30 AM segment to a more conventional morning period around 8:00 AM, obscuring the temporal anomaly (Fig. 8C_1). Next, observing that the consumption values were still extreme, she applied the *move-y* tool to vertically scale the peak values down by approximately 50% (Fig. 8C_2). Finally, to make the user's evening pattern less empty, she used the Clone tool to replicate the usage pattern between midnight and 5:00 AM to the 6:00 PM slot (Fig. 8C_3). The original data was then masked using *removal* to prevent potential identification (Fig. 8C_4). After processing, EC synchronized the adjusted data back to the left-side view for verification. Upon inspection, the previously anomalous pattern of User 8 no longer exhibited significant deviation (Fig. 8E_1, RQ2).

Weekend routine behavior. To analyze the differences in electricity consumption patterns between workdays and weekends, EC switched the navigation bar's display mode for comparison. She identified a salient outlier occurring only on weekend afternoons (Fig. 8B_1). Hovering revealed this behavior belonged to User 10, whose detailed view showed consistent, sharp consumption surges exclusively on Saturday afternoons (Fig. 8B_2). EC hypothesized this indicated a unique and regular weekend routine, such as a family gathering, posing a **pattern-based** privacy risk. She added User 10's Saturday data (comprising four time series from a one-month dataset) to the edit view (RQ1).

Among the four candidate time series, EC excluded normal data from March 7, 2015, retaining three anomalous sequences for privacy processing. While most users exhibit increased power consumption during weekend afternoons, User 10's Saturday afternoon load peaks (sustained above 6kW) were anomalously prominent. To mitigate privacy disclosure risks, EC decided to smooth these peaks and reduce fluctuations. To better achieve these two objectives, she decomposed each original time series into low-frequency (trend) and high-frequency (residual) components (Fig. 8D_1). She then applied the *move-y* operation only to the low-frequency trend component, reducing the prominent peak from over 6,000 to a more typical 4,000 Watts (Fig. 8D_2), preserving the overall consumption trend. Subsequently, she used the *curve* tool on the high-frequency residual components, applying a half-value template to attenuate the signal's sharp fluctuations (Fig. 8D_3), preserving essential oscillatory characteristics. After modification, the outlier was gone from the stream view (Fig. 8E_2). While the user's Saturday afternoon electricity consumption still exhibited an increased pattern, it no longer appeared anomalous (Fig. 8D_4, RQ2).

5.2 Case 2: Capture-24 Activity Tracker Dataset

Case Setting. The Capture-24 activity tracker dataset for human activity recognition [15] we used contains Axivity AX3 wrist-worn activity tracker data collected from 151 participants around the Oxfordshire area, nearly 4,000 hours of wearable camera data with labeled activities (e.g., home activity, sports, sleeping). The analysis focuses on 43 users with complete 24-hour acceleration records, enabling a comprehensive assessment of daily activity patterns. The resulting dataset comprises 61,920 triaxial (x, y, z) acceleration samples, occupying 6.54 MB after preprocessing. We edit this dataset to mitigate privacy risks while maintaining its utility for downstream *classification* tasks.

Upon loading the dataset, ED initiated exploratory analysis by examining the three variables (x, y, and z axes) to assess overall data distributions. The initial inspection revealed an obviously higher concentration of outliers along the y-axis (Fig. 1B), prompting a more focused investigation of anomalous y-axis readings.

Motion pattern identification. ED first hovered over an extreme outlier, and then a tooltip displayed its annotation "MET8.0" (Fig. 1B_1), indicating vigorous physical activity. This observation led him to hypothesize that these deviations were caused by vigorous physical activity. He confirmed this by inspecting other outliers, noting that two of the most extreme instances, from User 12 and User 131, were both tagged as high-intensity exercise (Fig. 1B_1, B_2). ED then dragged both users' data into the editing interface for a deeper analysis of the potential privacy risks (RQ1).

For User 12, the issue was a clear, habitual morning run between 7:00 and 8:30 AM (Fig. 1B_3). While the activity itself was not unusual, its consistent and unique timing created a strong, temporally identifiable signature. To mitigate this **time-based** risk without losing the valuable activity data, ED decided to relocate the pattern. He used the *clone* tool to copy the 7:00-8:30 AM exercise sequence to a more common workout period in the afternoon (4:00 PM, Fig. 1E_1) and then masked the original morning segment using the *removal* tool (Fig. 1E_2).

For User 131, the risk was different. The afternoon exercise timeframe was common, but the continuous, extreme acceleration spikes suggested potential critical incidents like falls or collisions, creating a unique **pattern-based** risk (Fig. 1B_4, D_1). Before intervening, ED first inspected the editing operations recommended by the system, indicated by a lightbulb icon (Fig. 1E_3). According to the suggestion, he opted to employ the *curve* tool for a precise, non-linear adjustment, selectively reducing the magnitude of only the most extreme spikes by approximately 25% while leaving other values untouched (Fig. 1E_4). After applying the changes, ED clicked the synchronization button and confirmed in the identification views that the previously prominent outliers for both users had been mitigated (RQ2).

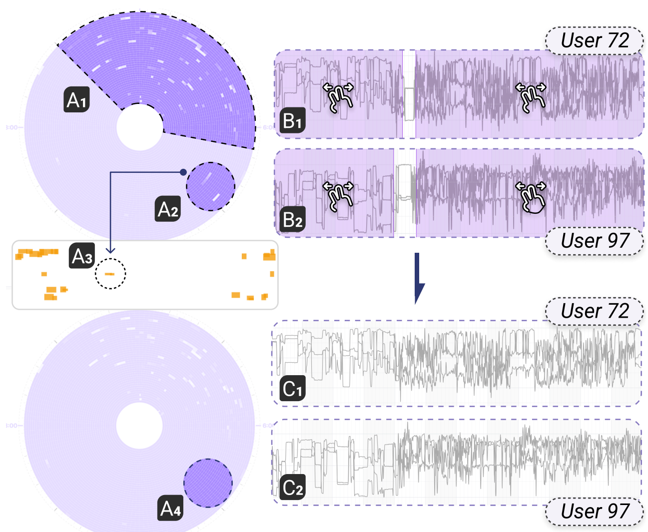

Figure 9: Non-wearing pattern detection in Case 2. A) Detects non-wearing periods using the radial and pattern views, highlighting abnormal gaps in acceleration data patterns. B, C) Show the correction process where the expand operation is applied to remove the non-wearing segments, preserving the overall data integrity while mitigating privacy risks.

Non-wearing pattern detection. Leveraging domain expertise, ED identified deliberate device removal during sensitive periods (resulting in sustained zero-values or flatlined signals) as user privacy-protection behaviors, warranting the exclusion of these segments to prevent potential identification. Returning to the radial view, ED adjusted the color mapping to highlight near-zero values. While sustained zero-values (white bands) are normal during sleep hours (Fig. $9A_1$), two users (72 and 97) exhibited identical **patterns** during other periods (Fig. $9A_2$). To validate this, he turned to the pattern view and selected the **plateau** category. The glyph distribution (Fig. $9A_3$) confirmed his suspicion, showing a distinct cluster of low-magnitude, low-variance plateaus. Brushing this cluster filtered the card list to reveal the specific flat-line segments, which leads to the same conclusion (RQ1).

ED then dragged the three-axis (x, y, z) data of the two users into the editing interface for detailed inspection. He observed constant acceleration across all three axes (Fig. $9B_1$, B_2), and decided to exclude these non-wearing periods. For User 97, he used the *expand* tool, selecting only the segments with valid activity data and stretching them to fill the timeline (Fig. $9C_2$). After that, ED noted that User 72 required the same modification. Instead of repeating the steps manually, he utilized the automated batch-apply feature. With a single click, the system reused the expand operation and applied it to User 97's series (Fig. $9C_1$), improving the workflow efficiency. A final synchronization confirmed in the radial view that the anomalous white bands for both users had been successfully removed (Fig. $9A_4$, RQ2).

5.3 Expert Interview

We conducted semi-structured expert interviews with four external experts, EF, EG, EH, and EI. EF and EG are researchers with over three years of experience in time series analysis. EH is a UI designer with a strong background in user-centered design. EI is a machine learning practitioner who frequently collects and publishes data.

All expert interviews were conducted one-on-one through an online video conferencing platform. The interview procedures were as follows. First, we walked the participants through the basic functionalities of *TSEditor*. Next, the aforementioned two case studies were demonstrated to the participants. The participants could interrupt and raise any questions freely during the demonstration. Finally, we solicited the participants' comments and feedback regarding the effectiveness, usability, and learnability of *TSEditor*. The results are as follows.

Effectiveness. All four experts confirmed the effectiveness and intuitiveness of *TSEditor*. EF, EG, and EH particularly praised the stream view for its utility in analyzing value-sensitive privacy risks: "*After loading the dataset into the system, I could immediately identify problematic patterns in the stream view and further validate them through interactive exploration across other linked views* (RQ1)." EF highlighted the critical importance of implementing both undo/redo functionality and real-time dataset export features: "*These capabilities allow for iterative refinement during editing processes while maintaining multiple versions of the dataset for subsequent selective analysis and comparison* (RQ2, 3)." EI mentioned that when working with public datasets, volunteers usually need to sign consent agreements. "*If there were a system that could help me handle the privacy issues, I would be very willing to use it*," commented EI.

Usability. All four experts agreed that *TSEditor* is useful for privacy-preserving tasks. EH emphasized the practical value of the removal feature, particularly appreciating its automated historical data retrieval capability, which is crucial for addressing a wide range of privacy preservation challenges across different contexts (RQ2). EF and EH both expressed strong approval of the component view, stating, "*The component design is beneficial for managing, comparing, and editing multiple time series simultaneously, providing a clear and organized interface for complex tasks* (RQ2)." EG expressed a strong preference for the radial view, saying, "*The layout maintains clarity even when dealing with numerous individuals, effectively preventing overlap and visual clutter that commonly occurs in other representations, allowing for easier comparison of diverse data patterns across users* (RQ1)."

Learnability. All four experts appreciated that our system is highly user-friendly and requires minimal learning time. They noted that, with only a brief introduction and demonstration, they clearly understood the interaction methods and the implementation effects of each editing operation. EH commented, "*The icons for editing operations are intuitive, and the time series manipulation feels responsive and seamless, providing a smooth user experience*."

Improvement. In addition to the above comments, the experts also provided some valuable advice. EH recommended adding a search functionality, noting that "*when I edit out an outlier, I currently have to manually scroll to relocate its detailed view - a search box would significantly improve this workflow*." EF suggested incorporating additional reference elements in the component view, such as an average time series component across multiple records, to facilitate the editing process. We plan to incorporate them in subsequent iterations of the system.

5.4 Model evaluation

To quantitatively assess whether the editing operations compromise data utility, we adopted the standard evaluation protocol established in representative time series research (e.g., TimeGAN [92], TimeVAE [19]). We evaluated the performance of downstream machine learning models on both the original and edited datasets from our case studies. As argued in prior work, comparable performance in downstream tasks serves as a robust indicator that the processed data preserves the underlying temporal dynamics and conditional distributions essential for analysis. Thus, we aimed to verify that *TSEditor*'s privacy-preserving modifications did not significantly degrade the performance of common analytical tasks. Additionally, to benchmark against fully automated techniques, we implemented a standard automation baseline using the *IBM Differential Privacy Library* [34]. The detailed experimental setup, data processing, model parameters, and results are in the supplementary material.

Prediction (Case 1). For the REFIT consumption dataset, we performed a time series prediction task using TimeMixer [83]. The results indicate that the edits had a negligible impact on forecasting performance. The Mean Absolute Error (MAE) and Mean Squared Error (MSE) between the original and edited datasets showed minimal relative changes of only 1.12% and 0.89%, respectively (RQ3).

Classification (Case 2). For the Capture-24 activity tracker dataset, we evaluated a classification task using the official benchmark model[3]. The model's performance on the edited data remained nearly identical to its performance on the original data across four standard metrics, including Balanced Accuracy and Macro F1 Score

[3]https://github.com/OxWearables/capture24

(RQ3). Comparative results show that the automated baseline led to an observable performance drop. Furthermore, we provide detailed class-wise performance metrics in the supplementary material, showing that recognition rates for specific activities involved in the edits remained at a comparable level (RQ3).

Overall, these evaluations suggest that *TSEditor* can mitigate privacy risks while preserving high data utility for downstream analytical tasks.

6 User Study

While the case studies demonstrated *TSEditor*'s capabilities in expert scenarios, we conducted a task-based user study[4] to verify whether the system enables broader users to perform the core tasks of identification (RQ1) and mitigation (RQ2) efficiently, and to assess the overall usability of the workflow. As no comparable interactive system for time series privacy editing exists (described in Sect. 2.2), this study was designed as a formative evaluation of *TSEditor*'s novel workflow and usability. Consequently, a baseline comparison against manual methods (e.g., scripting) was omitted, as our goal was to assess the specific design contributions of our system, rather than to simply prove the advantages of a dedicated tool.

6.1 Participants

We recruited 12 participants (7 female, 5 male; aged 18-27, M = 23, SD = 2.66), denoted as P1 to P12, via university forums and social media. The participants hailed from a diverse range of academic backgrounds, including computer science (2), electronic engineering (2), optoelectronics (1), mathematics (1), chemistry (1), civil and hydraulic engineering (1), biological sciences (2), and medicine (2). We screened for participants who had a basic understanding of and varying degrees of experience with handling time-series data. Specifically, in a pre-study questionnaire, all participants self-reported their familiarity with time series data on a 5-point Likert scale. We required a score of at least 3 to be included in the study. On average, participants had 2.4 years of experience working with time series data.

6.2 Procedure

The study was conducted both in-person and remotely to accommodate participants' schedules and preferences. In-person sessions took place in a quiet university lab, while remote sessions were conducted via video conferencing software. Regardless of the setting, the entire session was managed through the conferencing software, which was used to record participants' on-screen activities and audio commentary for subsequent analysis. The session lasted for about 75 minutes, after which each participant received 60 CNY in appreciation for their time. The study procedure was structured into three main phases: a training phase, a main task phase, and a post-session feedback phase.

Training Phase (~25min) - Participants watched a 15-minute video tutorial that provided both a brief overview of the study's research background and an introduction to the user interface and core functionalities of *TSEditor*. Following the tutorial, they were given a 10-minute free exploration task with a step-count dataset.[5] This allowed them to familiarize themselves with the system's visual encodings and interactive features before proceeding to the main tasks.

Main Task Phase (~35min) - In this core phase, participants worked with the electricity and capture datasets previously featured in our case studies. To ground the evaluation in a realistic context, we moved beyond blind exploration and designed five tasks that simulate a common privacy threat: a linkage attack using external information. We provided participants with five descriptions of unique individual patterns. For each description, they were instructed to perform a two-step process: 1) explore the dataset to identify the user or time series segment that matched the external information, and 2) apply arbitrary editing operations to the identified sequence to mitigate the privacy risk. After each task, they were prompted to synchronize the data to visually assess if the original pattern's risk was successfully diminished or eliminated. Further details on the external information for the tasks can be found in the supplementary material.

Post-session Feedback Phase (~15min) - After addressing the five tasks, participants were asked to complete two standard questionnaires: the System Usability Scale (SUS) [10] to measure perceived usability and the NASA-TLX [33] to assess cognitive workload. The session concluded with a semi-structured interview to gather qualitative feedback on their overall experience with *TSEditor*.

6.3 Quantitative Results

Task completion time. All participants successfully completed all five tasks, correctly identifying the individual or segment that matched the external information (RQ1) and subsequently editing the data to diminish the pattern's privacy risk (RQ2). Detailed completion times are presented in Fig. 10. On average, participants spent 89 seconds on the risk identification phase and 102 seconds on the editing phase for each task. More specifically, a deeper analysis revealed a significant interaction between the risk type and the time required for each phase. In the identification phase, participants were substantially faster at locating time and value risks compared to the more time-consuming task of identifying complex patterns. Interestingly, this trend inverted in the editing phase: participants spent considerably more time performing careful edits on time and value risks than they did on pattern risks. This suggests that the primary challenge of time and value risks lies in the deliberate, fine-grained modification process. In contrast, for pattern risks, the main effort was concentrated in the initial, more complex identification.

Editing Operation Usage. Participants demonstrated high efficiency, requiring only 2.4 operations on average to resolve each task. This suggests the provided operations are powerful and well-suited to the tasks, sparing users from tedious low-level manipulation. We also observed a broad adoption of the toolset, with most participants (10/12) using four or more distinct operation types. In terms of frequency, clone (52 uses) and curve (39 uses) were the most favored. Our qualitative analysis reveals that this preference is strongly linked to users' editing strategies for data plausibility.

[4]The user study has been approved by State Key Lab of CAD&CG, Zhejiang University.

[5]The step-count series used is part of the Fitbit Fitness Tracker Data, available at: https://www.kaggle.com/datasets/arashnic/fitbit.

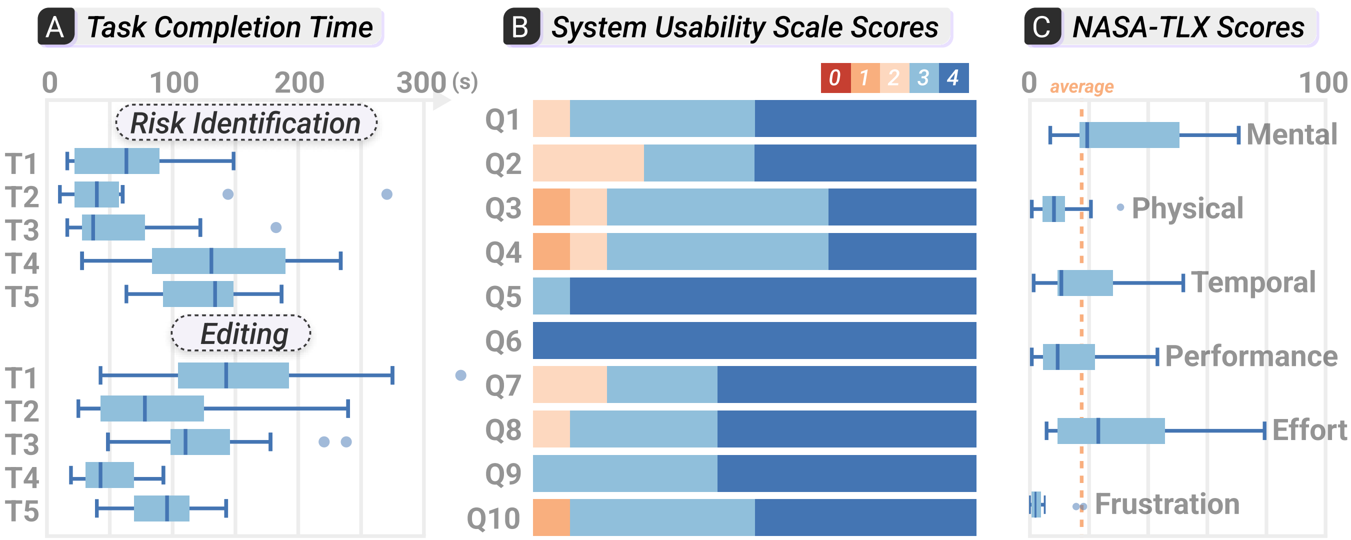

Figure 10: The quantitative results of the user study. (A) Task completion time per task, split by the risk identification and editing phases. (B) Score distribution for the SUS questionnaire. (C) NASA-TLX cognitive workload scores, broken down by its six subscales.

System Usability Scale (SUS). Participants rated the usability of *TSEditor* highly, with the system achieving an average SUS score of 85.5 (see Fig. 10B). This score is well above the industry average and falls into the A^+ grade category (scores above 84.1), which indicates an excellent level of perceived usability [45]. Further analysis of the questionnaire highlights that the system's consistency and integration were its most commendable qualities.

NASA-TLX Workload. To assess the cognitive load, we used the NASA-TLX questionnaire, where lower scores indicate lower workload. The overall workload score for using *TSEditor* was very low (M = 17.97, SD = 13.31), indicating that the system was not cognitively demanding for the analytical tasks. A deeper analysis of the six NASA-TLX subscales provides a more nuanced picture of this low workload. The primary contributors were Mental Demand (M = 30.5) and Effort (M = 30.42), which is expected given the inherent complexity of the data analysis scenarios. Crucially, the Frustration score was exceptionally low (M = 3.92), a key indicator of interaction quality. This pattern suggests that the cognitive load experienced by participants was intrinsic to the analytical challenge itself. As described in Cognitive Load Theory [75], this demonstrates that the system successfully minimized the extraneous cognitive load typically induced by a poorly designed user interface.

6.4 Qualitative Results

Our qualitative analysis, derived from the think-aloud protocol [28] during the tasks and semi-structured interviews, revealed nuanced insights into the user experience of *TSEditor*. Corroborating the quantitative SUS and NASA-TLX scores, participants' feedback described the workflow as highly usable and intuitive. This effective design appeared to minimize extraneous cognitive load, enabling users to concentrate on analytical challenges rather than the mechanics of the tool. Beyond this general feedback, our analysis identified two key themes regarding their analytical strategies:

6.4.1 Navigating the Trade-off between Data Plausibility and Privacy. Interviews revealed the mental models guiding participants' editing behaviors. Central to this was a trade-off between mitigating privacy risks and maintaining data plausibility, which led users to develop sophisticated, in-situ strategies. Many participants were deliberate in their process, aiming to make modifications that would seamlessly blend with the surrounding data, rather than appear as obvious manipulations. As P8 articulated, their thought process was focused on "*how to make the edit seem reasonable*," rather than simply removing the sensitive pattern. A common heuristic was the principle of "*minimal intervention*," where P9 aimed to "*modify the fewest points possible*." To enhance plausibility, participants would creatively combine tools. For example, P6 developed a two-step process of first using *Clone* to replace a pattern and then applying a subtle *Curve* adjustment, explaining: "*the result would not appear as an exact duplicate*." Crucially, the system's visual feedback mechanism was instrumental in building user confidence. The ability to preview edits and compare before-and-after states reassured participants that their changes achieved the intended effect, which P9 cited as key to "*trust the results*."

6.4.2 Divergent Perspectives and Aspirations for Advanced Functionality. Participants' feedback on specific views revealed diverse analytical strategies and preferences. The radial view elicited the most polarized reactions. P2 found it innovative and effective for comparison, with P10 emphasizing its analytical depth through color mapping for more detailed exploration. In contrast, participants accustomed to Cartesian plots reported a learning curve (P9, P11). These results suggest the radial view serves as a specialist complement to conventional visualizations, whose effectiveness appears to depend on user background. Participants also identified opportunities for enhanced functionality to support more efficient and customized workflows. Beyond these, several participants

(P3, P5, P10) desired more intelligent assistance, such as features to learn from their editing habits and apply similar adjustments (P10). They suggested such improvements would reduce repetitive work and better align the system with their individual strategies.

7 Discussion

In this section, we discuss interaction impacts, trade-off between privacy and faithfulness, scalability, limitations, and future work of the proposed system.

7.1 Interaction Impact on Data Characteristics

Regarding the preservation of key time-series characteristics, we identify a distinct hierarchy of invasiveness across our editing tools. At the foundational level, *move-x* and *move-y* are **conservative**: they shift data coordinates without altering the internal relationships between data points, strictly retaining the signature of the original behavior. In contrast, *expand* and *curve* are **distortive**: they perform transformations based on the original data. While they modify attributes like duration or intensity, the edited sequence maintains a relatively high correlation with the original structure. Finally, *clone* and *removal* are **destructive**: they fundamentally replace the original data segment. This represents the most significant alteration to data characteristics, as the unique pattern is substituted with an alternative. Given this variance, the optimal strategy prioritizes the most appropriate intervention tailored to the specific threat. Human expertise is therefore essential to navigate this trade-off, utilizing conservative tools where possible while strategically deploying high-strength, destructive operations when critical privacy risks necessitate complete cover-up.

7.2 Trade-off between Privacy and Faithfulness

A fundamental conflict highlighted by our case studies is the tension between data privacy and data faithfulness. Mitigating a high-risk issue inevitably alters a characteristic time series that contributes to dataset diversity. We argue that this constitutes an inherent paradox of uniqueness: the statistical distinctiveness that renders a time series scientifically interesting is often identical to the feature that makes it vulnerable to re-identification. Unlike automated methods that indiscriminately suppress such deviations, *TSEditor* empowers experts to navigate this conflict by leveraging domain knowledge to decouple sensitive attributes from valuable data semantics. This allows analysts to distinguish between dimensions that pose severe risks, such as precise timing, and those that are benign, such as the morphological wave shape. Consequently, they can apply targeted interventions that mitigate the specific risk while maximizing the preservation of the underlying behavioral distribution. Thus, we hope to redefine "faithfulness" not as the rigid preservation of raw values, but as the retention of plausible data semantics within safe privacy bounds.

7.3 Scalability

Regarding **population scalability**, the system's applicability is constrained not by the total dataset size, but by the number of high-risk individuals. Fundamentally, our approach operates as an anomaly detection workflow. Since the volume of anomalies is typically limited to a manageable range, the need for exhaustive inspection is inherently avoided. Instead, the system remains effective for large-scale datasets by focusing solely on the sparse subset of distinctive cases. However, for highly heterogeneous large-scale datasets, it remains an open challenge requiring future research.

Regarding **temporal scalability**, handling long time series inevitably necessitates data abstraction to accommodate visual resolution limits. To mitigate the potential loss of patterns during this process, *TSEditor* features fully configurable aggregation and downsampling parameters. This design empowers users to interactively tune the visual granularity to match the specific temporal dynamics of the data , effectively offsetting resolution constraints and ensuring that patterns of interest remain discernible.

7.4 Limitations and Future Work

We identify several limitations that require deeper discussion. Reflecting on these challenges guides our roadmap for future work, aiming to address fundamental open problems in interactive time-series editing and trusted data sharing.

7.4.1 Multivariate relationships and constraints. Currently, *TSEditor* supports multivariate data via a "multiple univariate" strategy. Our system treats each variable independently, visualizing and inspecting them as separate time series within the identification views. Simultaneously, we support the mitigation of multivariate through batch editing. However, treating variables independently may neglect inter-variable dependencies like correlations, inviting cross-variable inference attacks where sanitized patterns are reconstructed via associated unedited variables. Furthermore, the current univariate detection strategy overlooks risks arising from joint variable interactions. To address this, future work could integrate relation-based exploration views to visualize the dynamic relationships between variables. This would enable users to identify distinctive attribute combinations, e.g., high physical activity accompanied by a low heart rate, that may uniquely characterize individuals despite appearing normal in isolation. Additionally, to prevent semantic inconsistencies caused by independent modifications, future research could explore constraint-based editing or causal propagation models that automatically synchronize edits across correlated variables.

7.4.2 Interaction fluidity and data plausibility. The isolation of editing in *TSEditor* addresses the incompatibility between risk-centric visual metaphors (e.g., radial layouts) and precise Cartesian manipulation, while ensuring operations on raw data. However, this separation disrupts workflow continuity, necessitating context switching to relocate identified risks within the editing environment. Future research could investigate in-situ editing paradigms via *focus + context* techniques to reconcile these visual and resolution discrepancies, seamlessly unifying exploration and modification. Moreover, regarding editing mechanics, the current hard selection boundaries prioritize control but may introduce unnatural discontinuities. To enhance data plausibility, future work could adopt gradient-based weighting techniques, such as smooth brushing [20], to blend edits seamlessly with adjacent data points, ensuring alterations remain indistinguishable from natural variations.

7.4.3 Evaluation and feedback mechanisms. Regarding quantitative evaluation, while we adopt standard downstream task performance, we acknowledge that such macroscopic metrics may mask local distortions. Crucially, existing benchmarks for time series generation prioritize distributional fidelity over targeted risk mitigation. To address this, the community requires the establishment of privacy-aware and task-specific benchmarks designed to rigorously quantify the trade-off between privacy gain and local feature preservation. Simultaneously, regarding the interaction feedback loop for data quality and utility, *TSEditor* relies on post-hoc verification, as the impacts are inherently task-dependent, making universal real-time metrics computationally non-trivial. Consequently, future work could develop lightweight models to provide instantaneous estimates of utility loss during the editing process.

8 Conclusion

In this work, we present *TSEditor*, an interactive time series editing system that empowers experts to visually analyze privacy risks, interactively edit time series, flexibly manage multiple series, and intuitively assess editing impacts. We first summarize the taxonomy of time series data privacy and propose corresponding editing approaches. We further design coordinated views with privacy-sensitive visualizations and integrate six user-friendly editing operations. The effectiveness and usability of the tool were demonstrated through a multi-faceted evaluation conducted on real-world datasets, including two case studies, an expert interview, a model evaluation, and a user study. In the future, we will extend the tool to address the limitations observed and deploy it online for more secure data-sharing practices.

Acknowledgments

The work was supported by NSFC (U22A2032, 62402421, 62421003, 62402184, 62202244) and Ningbo Yongjiang Talent Programme (2024A-399-G). We extend our sincere gratitude to the anonymous reviewers for their insightful feedback and constructive suggestions, and to all participants in our studies.

References

[1] Wolfgang Aigner, Silvia Miksch, Wolfgang Müller, Heidrun Schumann, and Christian Tominski. 2007. Visualizing time-oriented data — A systematic view. Computers & Graphics 31, 3 (2007), 401–409. doi:10.1016/j.cag.2007.01.030
[2] Wolfgang Aigner, Silvia Miksch, Wolfgang Müller, Heidrun Schumann, and Christian Tominski. 2008. Visual Methods for Analyzing Time-Oriented Data. IEEE Transactions on Visualization and Computer Graphics 14, 1 (2008), 47–60. doi:10.1109/TVCG.2007.70415
[3] Wolfgang Aigner, Silvia Miksch, Heidrun Schumann, and Christian Tominski. 2023. Visualization of Time-Oriented Data. Springer, London. doi:10.1007/978-1-4471-7527-8
[4] Mohammed Ali, Ali Alqahtani, Mark W. Jones, and Xianghua Xie. 2019. Clustering and Classification for Time Series Data in Visual Analytics: A Survey. IEEE Access 7 (2019), 181314–181338. doi:10.1109/ACCESS.2019.2958551
[5] Benjamin Bach, Conglei Shi, Nicolas Heulot, Tara Madhyastha, Tom Grabowski, and Pierre Dragicevic. 2016. Time Curves: Folding Time to Visualize Patterns of Temporal Evolution in Data. IEEE Transactions on Visualization and Computer Graphics 22, 1 (2016), 559–568. doi:10.1109/TVCG.2015.2467851
[6] Franziska Becker and Tanja Blascheck. 2024. TimeSeriesMaker: Interactive Time Series Composition in No Time. In Proceedings of IEEE 17th Pacific Visualization Conference (PacificVis). IEEE, Tokyo, Japan, 301–306. doi:10.1109/PacificVis60374.2024.00042
[7] Philip Berger, Heidrun Schumann, and Christian Tominski. 2022. Integrating Visual Exploration and Direct Editing of Multivariate Graphs. Springer International Publishing, Cham, 459–483. doi:10.1007/978-3-030-93119-3_18
[8] Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. 2011. D^3 data-driven documents. IEEE Transactions on Visualization and Computer Graphics 17, 12 (2011), 2301–2309. doi:10.1109/TVCG.2011.185
[9] George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. 2015. Time series analysis: forecasting and control. John Wiley & Sons.
[10] John Brooke et al. 1996. SUS-A quick and dirty usability scale. Usability evaluation in industry 189, 194 (1996), 4–7. doi:10.1201/9781498710411-35
[11] Erik Buchmann, Klemens Böhm, Thorben Burghardt, and Stephan Kessler. 2013. Re-identification of Smart Meter data. Personal Ubiquitous Comput. 17, 4 (2013), 653–662. doi:10.1007/s00779-012-0513-6
[12] A. Buja, J.A. McDonald, J. Michalak, and W. Stuetzle. 1991. Interactive data visualization using focusing and linking. In Proceeding Visualization '91. IEEE, San Diego, CA, USA, 156–163. doi:10.1109/VISUAL.1991.175794
[13] Paolo Buono, Aleks Aris, Catherine Plaisant, Amir Khella, and Ben Shneiderman. 2005. Visualization and Data Analysis - Interactive pattern search in time series. In Proceedings of SPIE, Vol. 5669. SPIE, San Jose, California, United States, 175–186. doi:10.1117/12.587537
[14] Paolo Buono, Maria Francesca Costabile, and Rosa Lanzilotti. 2014. A circular visualization of people's activities in distributed teams. Journal of Visual Languages & Computing 25, 6 (2014), 903–911. doi:10.1016/j.jvlc.2014.10.025
[15] Shing Chan, Yuan Hang, Catherine Tong, Aidan Acquah, Abram Schonfeldt, Jonathan Gershuny, and Aiden Doherty. 2024. CAPTURE-24: A large dataset of wrist-worn activity tracker data collected in the wild for human activity recognition. Scientific Data 11, 1 (2024), 1135. doi:10.1038/s41597-024-03960-3
[16] Benjamin F. Crabtree, Subhash C. Ray, Priscilla M. Schmidt, Patrick T. O'Connor, and David D. Schmidt. 1990. The individual over time: Time series applications in health care research. Journal of Clinical Epidemiology 43, 3 (1990), 241–260. doi:10.1016/0895-4356(90)90005-A
[17] Yves-Alexandre De Montjoye, César A Hidalgo, Michel Verleysen, and Vincent D Blondel. 2013. Unique in the crowd: The privacy bounds of human mobility. Scientific reports 3, 1 (2013), 1376. doi:10.1038/srep01376
[18] Zikun Deng, Shifu Chen, Tobias Schreck, Dazhen Deng, Tan Tang, Mingliang Xu, Di Weng, and Yingcai Wu. 2024. Visualizing Large-Scale Spatial Time Series with GeoChron. IEEE Transactions on Visualization and Computer Graphics 30, 1 (2024), 1194–1204. doi:10.1109/TVCG.2023.3327162
[19] Abhyuday Desai, Cynthia Freeman, Zuhui Wang, and Ian Beaver. 2021. TimeVAE: A variational auto-encoder for multivariate time series generation. arXiv preprint arXiv:2111.08095 (2021). doi:10.48550/arXiv.2111.08095
[20] Helmut Doleisch and Helwig Hauser. 2002. Smooth Brushing for Focus+Context Visualization of Simulation Data in 3D. Journal of WSCG 10, 1–2 (2002), 147–154. http://hdl.handle.net/11025/5972
[21] Geoffrey M. Draper, Yarden Livnat, and Richard F. Riesenfeld. 2009. A Survey of Radial Methods for Information Visualization. IEEE Transactions on Visualization and Computer Graphics 15, 5 (2009), 759–776. doi:10.1109/TVCG.2009.23
[22] Cynthia Dwork. 2006. Differential privacy. In Proceedings of International colloquium on automata, languages, and programming. Springer, Berlin, Heidelberg, 1–12. doi:10.1007/11787006_1
[23] Christian Eichner, Stefan Gladisch, Heidrun Schumann, and Christian Tominski. 2016. Direct visual editing of node attributes in graphs. In Informatics, Vol. 3. MDPI, 17. doi:10.3390/informatics3040017
[24] Geoffrey Ellis and Alan Dix. 2007. A Taxonomy of Clutter Reduction for Information Visualisation. IEEE Transactions on Visualization and Computer Graphics 13, 6 (2007), 1216–1223. doi:10.1109/TVCG.2007.70535
[25] Meftah Elsaraiti, Gama Ali, Hmeda Musbah, Adel Merabet, and Timothy Little. 2021. Time Series Analysis of Electricity Consumption Forecasting Using ARIMA Model. In 2021 IEEE Green Technologies Conference (GreenTech). IEEE, Denver, CO, USA, 259–262. doi:10.1109/GreenTech48523.2021.00049
[26] Cristóbal Esteban, Stephanie L Hyland, and Gunnar Rätsch. 2017. Real-valued (medical) time series generation with recurrent conditional gans. arXiv preprint arXiv:1706.02633 (2017). doi:10.48550/arXiv.1706.02633
[27] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. 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. AAAI Press, Portland, Oregon, 226–231. https://dl.acm.org/doi/10.5555/3001460.3001507
[28] Marsha E Fonteyn, Benjamin Kuipers, and Susan J Grobe. 1993. A description of think aloud method and protocol analysis. Qualitative health research 3, 4 (1993), 430–441. doi:10.1177/104973239300300403
[29] Benjamin C. M. Fung, Ke Wang, Rui Chen, and Philip S. Yu. 2010. Privacy-preserving data publishing: A survey of recent developments. ACM Comput. Surv. 42, 4, Article 14 (2010), 53 pages. doi:10.1145/1749603.1749605
[30] Mehmet Emre Gursoy, Ling Liu, Stacey Truex, and Lei Yu. 2019. Differentially Private and Utility Preserving Publication of Trajectory Data. IEEE Transactions on Mobile Computing 18, 10 (2019), 2315–2329. doi:10.1109/TMC.2018.2874008
[31] Dareen K Halim and Samuel Hutagalung. 2022. Towards data sharing economy on Internet of Things: a semantic for telemetry data. Journal of Big Data 9, 1 (2022), 1. doi:10.1186/s40537-021-00549-0
[32] Jianing Hao, Qing Shi, Yilin Ye, and Wei Zeng. 2024. TimeTuner: Diagnosing Time Representations for Time-Series Forecasting with Counterfactual Explanations. IEEE Transactions on Visualization and Computer Graphics 30, 1 (2024),

1183–1193. doi:10.1109/TVCG.2023.3327389

[33] Sandra G. Hart and Lowell E. Staveland. 1988. Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. In Human Mental Workload. Advances in Psychology, Vol. 52. North-Holland, 139–183. doi:10.1016/S0166-4115(08)62386-9

[34] Naoise Holohan, Stefano Braghin, Pól Mac Aonghusa, and Killian Levacher. 2019. Diffprivlib: the IBM differential privacy library. arXiv preprint arXiv.1907.02444 (2019). doi:10.48550/arXiv.1907.02444

[35] Tom Horak, Philip Berger, Heidrun Schumann, Raimund Dachselt, and Christian Tominski. 2021. Responsive Matrix Cells: A Focus+Context Approach for Exploring and Editing Multivariate Graphs. IEEE Transactions on Visualization and Computer Graphics 27, 2 (2021), 1644–1654. doi:10.1109/TVCG.2020.3030371

[36] Yanwei Huang, Yan Miao, Di Weng, Adam Perer, and Yingcai Wu. 2025. StructVizor: Interactive Profiling of Semi-Structured Textual Data. In Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI '25). Association for Computing Machinery, New York, NY, USA, Article 393, 18 pages. doi:10.1145/3706598.3713484

[37] Waqas Javed, Bryan McDonnel, and Niklas Elmqvist. 2010. Graphical Perception of Multiple Time Series. IEEE Transactions on Visualization and Computer Graphics 16, 6 (2010), 927–934. doi:10.1109/TVCG.2010.162

[38] Kaifeng Jiang, Dongxu Shao, Stéphane Bressan, Thomas Kister, and Kian-Lee Tan. 2013. Publishing trajectories with differential privacy guarantees. In Proceedings of the 25th International conference on scientific and statistical database management. Association for Computing Machinery, New York, NY, USA, 1–12. doi:10.1145/2484838.2484846

[39] L. Jonkers, O. Cartapanis, M. Langner, N. McKay, S. Mulitza, A. Strack, and M. Kucera. 2020. Integrating palaeoclimate time series with rich metadata for uncertainty modelling: strategy and documentation of the PalMod 130k marine palaeoclimate data synthesis. Earth System Science Data 12, 2 (2020), 1053–1081. doi:10.5194/essd-12-1053-2020

[40] Sean Kandel, Andreas Paepcke, Joseph Hellerstein, and Jeffrey Heer. 2011. Wrangler: interactive visual specification of data transformation scripts. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (Vancouver, BC, Canada) (CHI '11). Association for Computing Machinery, New York, NY, USA, 3363–3372. doi:10.1145/1978942.1979444

[41] Manos Katsomallos, Katerina Tzompanaki, and Dimitris Kotzinos. 2019. Privacy, space and time: A survey on privacy-preserving continuous data publishing. Journal of Spatial Information Science 2019, 19 (2019), 57–103. doi:10.5311/JOSIS.2019.19.493

[42] Shruti Kaushik, Abhinav Choudhury, Pankaj Kumar Sheron, Nataraj Dasgupta, Sayee Natarajan, Larry A Pickett, and Varun Dutt. 2020. AI in healthcare: time-series forecasting using statistical, neural, and ensemble architectures. Frontiers in big data 3 (2020), 4. doi:10.3389/fdata.2020.00004

[43] Georgios Kellaris, Stavros Papadopoulos, Xiaokui Xiao, and Dimitris Papadias. 2014. Differentially Private Event Sequences over Infinite Streams. Proceedings of the VLDB Endowment 7, 12 (2014), 1155–1166. doi:10.14778/2732977.2732989

[44] Brittany Kondo and Christopher Collins. 2014. DimpVis: Exploring Time-varying Information Visualizations by Direct Manipulation. IEEE Transactions on Visualization and Computer Graphics 20, 12 (2014), 2003–2012. doi:10.1109/TVCG.2014.2346250

[45] James R. Lewis and Jeff Sauro. 2009. The Factor Structure of the System Usability Scale. In Human Centered Design. Springer Berlin Heidelberg, Berlin, Heidelberg, 94–103. doi:10.1007/978-3-642-02806-9_12

[46] Shuhan Liu, Yangtian Liu, Junxin Li, Yanwei Huang, Yue Shangguan, Zikun Deng, Di Weng, and Yingcai Wu. 2025. RidgeBuilder: Interactive Authoring of Expressive Ridgeline Plots. In Proceedings of the CHI Conference on Human Factors in Computing Systems. Association for Computing Machinery, New York, NY, USA, 18 pages. doi:10.1145/3706598.3714209

[47] Shuhan Liu, Yuan Tian, Zikun Deng, Weiwei Cui, Haidong Zhang, Di Weng, and Yingcai Wu. 2024. Relation-driven Query of Multiple Time Series. IEEE Transactions on Visualization and Computer Graphics (2024), 1–15. doi:10.1109/TVCG.2024.3397554

[48] Shuhan Liu, Di Weng, Yuan Tian, Zikun Deng, Haoran Xu, Xiangyu Zhu, Honglei Yin, Xianyuan Zhan, and Yingcai Wu. 2023. ECoalVis: Visual Analysis of Control Strategies in Coal-fired Power Plants. IEEE Transactions on Visualization and Computer Graphics 29, 1 (2023), 1091–1101. doi:10.1109/TVCG.2022.3209430

[49] Zhicheng Liu, John Stasko, and Timothy Sullivan. 2009. SellTrend: Inter-Attribute Visual Analysis of Temporal Transaction Data. IEEE Transactions on Visualization and Computer Graphics 15, 6 (2009), 1025–1032. doi:10.1109/TVCG.2009.180

[50] Xinchi Luo, Runfeng Jiang, Bin Yang, Hongxing Qin, and Haibo Hu. 2024. Air quality visualization analysis based on multivariate time series data feature extraction. Journal of Visualization 27, 4 (2024), 567–584. doi:10.1007/s12650-024-00981-3

[51] Miro Mannino and Azza Abouzied. 2018. Expressive Time Series Querying with Hand-Drawn Scale-Free Sketches. In Proceedings of CHI Conference on Human Factors in Computing Systems. Association for Computing Machinery, New York, NY, USA, 1–13. doi:10.1145/3173574.3173962

[52] Yulian Mao, Qingqing Ye, Qi Wang, and Haibo Hu. 2023. Differential privacy for time series: A survey. Data Engineering 47, 2 (2023), 68–93.

[53] Damien Masson, Young-Ho Kim, and Fanny Chevalier. 2025. Textoshop: Interactions Inspired by Drawing Software to Facilitate Text Editing. In Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems. Association for Computing Machinery, New York, NY, USA, 1–14. doi:10.1145/3706598.3713862

[54] Olof Mogren. 2016. C-RNN-GAN: Continuous recurrent neural networks with adversarial training. arXiv preprint arXiv:1611.09904 (2016). doi:10.48550/arXiv.1611.09904

[55] Tanja Munz-Körner and Daniel Weiskopf. 2024. Exploring visual quality of multidimensional time series projections. Visual Informatics 8, 2 (2024), 27–42. doi:10.1016/j.visinf.2024.04.004

[56] Tamara Munzner. 2009. A Nested Model for Visualization Design and Validation. IEEE Transactions on Visualization and Computer Graphics 15, 6 (2009), 921–928. doi:10.1109/TVCG.2009.111

[57] David Murray, Lina Stankovic, and Vladimir Stankovic. 2017. An electrical load measurements dataset of United Kingdom households from a two-year longitudinal study. Scientific data 4, 1 (2017), 1–12. doi:10.1038/sdata.2016.122

[58] Arvind Narayanan and Vitaly Shmatikov. 2008. Robust De-anonymization of Large Sparse Datasets. In 2008 IEEE Symposium on Security and Privacy (sp 2008). IEEE Computer Society, USA, 111–125. doi:10.1109/SP.2008.33

[59] Anna Offenwanger, Matthew Brehmer, Fanny Chevalier, and Theophanis Tsandilas. 2024. TimeSplines: Sketch-Based Authoring of Flexible and Idiosyncratic Timelines. IEEE Transactions on Visualization and Computer Graphics 30, 1 (2024), 34–44. doi:10.1109/TVCG.2023.3326520

[60] Lukasz Olejnik, Claude Castelluccia, and Artur Janc. 2014. On the uniqueness of web browsing history patterns. annals of telecommunications-annales des télécommunications 69, 1 (2014), 63–74. doi:10.1007/s12243-013-0392-5

[61] Spiros Papadimitriou, Feifei Li, George Kollios, and Philip S. Yu. 2007. Time series compressibility and privacy. In Proceedings of the 33rd International Conference on Very Large Data Bases. VLDB Endowment, Vienna, Austria, 459—-470. https://dl.acm.org/doi/10.5555/1325851.1325905

[62] William Playfair. 1801. The commercial and political atlas: representing, by means of stained copper-plate charts, the progress of the commerce, revenues, expenditure and debts of england during the whole of the eighteenth century. T. Burton, London.

[63] Jovan Powar and Alastair R Beresford. 2023. SoK: Managing risks of linkage attacks on data privacy. Proceedings on Privacy Enhancing Technologies 2023, 2 (2023), 97—-116. doi:10.56553/popets-2023-0043

[64] Xuebin Ren, Chia-Mu Yu, Weiren Yu, Shusen Yang, Xinyu Yang, Julie A. McCann, and Philip S. Yu. 2018. LoPub : High-Dimensional Crowdsourced Data Publication With Local Differential Privacy. IEEE Transactions on Information Forensics and Security 13, 9 (2018), 2151–2166. doi:10.1109/TIFS.2018.2812146

[65] Takafumi Saito, Hiroko Nakamura Miyamura, Mitsuyoshi Yamamoto, Hiroki Saito, Yuka Hoshiya, and Takumi Kaseda. 2005. Two-tone pseudo coloring: Compact visualization for one-dimensional data. In Proceedings of IEEE Symposium on Information Visualization (INFOVIS). IEEE Computer Society, USA, 173–180. doi:10.1109/INFVIS.2005.1532144

[66] Sina Shaham, Gabriel Ghinita, Bhaskar Krishnamachari, and Cyrus Shahabi. 2024. Differentially Private Publication of Electricity Time Series Data in Smart Grids. arXiv preprint arXiv:2408.16017 (2024). doi:10.48550/arXiv.2408.16017

[67] Gota Shirato, Natalia Andrienko, and Gennady Andrienko. 2023. Exploring and visualizing temporal relations in multivariate time series. Visual Informatics 7, 4 (2023), 57–72. doi:10.1016/j.visinf.2023.09.001

[68] Gota Shirato, Natalia Andrienko, and Gennady Andrienko. 2023. Identifying, exploring, and interpreting time series shapes in multivariate time intervals. Visual Informatics 7, 1 (2023), 77–91. doi:10.1016/j.visinf.2023.01.001

[69] Lidan Shou, Xuan Shang, Ke Chen, Gang Chen, and Chao Zhang. 2013. Supporting Pattern-Preserving Anonymization for Time-Series Data. IEEE Transactions on Knowledge and Data Engineering 25, 4 (2013), 877–892. doi:10.1109/TKDE.2011.249

[70] Robert H. Shumway and David S. Stoffer. 2025. Time Series Analysis and Its Applications: With R Examples (5th ed.). Springer Cham. doi:10.1007/978-3-031-70584-7

[71] Mudhakar Srivatsa and Mike Hicks. 2012. Deanonymizing mobility traces: using social network as a side-channel. In Proceedings of the 2012 ACM Conference on Computer and Communications Security (Raleigh, North Carolina, USA) (CCS '12). Association for Computing Machinery, New York, NY, USA, 628–637. doi:10.1145/2382196.2382262

[72] Luka Stopar, Primoz Skraba, Marko Grobelnik, and Dunja Mladenic. 2019. StreamStory: Exploring Multivariate Time Series on Multiple Scales. IEEE Transactions on Visualization and Computer Graphics 25, 4 (2019), 1788–1802. doi:10.1109/TVCG.2018.2825424

[73] Emily Chia-Yu Su and Han-Ming Wu. 2024. Dimension reduction and visualization of multiple time series data: a symbolic data analysis approach. Computational Statistics 39, 4 (2024), 1937–1969. doi:10.1007/s00180-023-01440-7

[74] Dong Sun, Renfei Huang, Yuanzhe Chen, Yong Wang, Jia Zeng, Mingxuan Yuan, Ting-Chuen Pong, and Huamin Qu. 2020. PlanningVis: A Visual Analytics Approach to Production Planning in Smart Factories. IEEE Transactions on Visualization and Computer Graphics 26, 1 (2020), 579–589. doi:10.1109/TVCG.2019.2934275

[75] John Sweller, Jeroen JG Van Merrienboer, and Fred GWC Paas. 1998. Cognitive architecture and instructional design. Educational psychology review 10, 3 (1998), 251–296. doi:10.1023/A:1022193728205

[76] Christian Tominski and Heidrun Schumann. 2020. Interactive visual data analysis. AK Peters/CRC Press, New York. doi:10.1201/9781315152707

[77] Edward R Tufte and Peter R Graves-Morris. 1983. The visual display of quantitative information. Graphics Press, USA.

[78] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing Data using t-SNE. Journal of Machine Learning Research 9, 86 (2008), 2579–2605. http://jmlr.org/papers/v9/vandermaaten08a.html

[79] J.J. Van Wijk and E.R. Van Selow. 1999. Cluster and calendar based visualization of time series data. In Proceedings of IEEE Symposium on Information Visualization (InfoVis). IEEE Computer Society, USA, 4–9. doi:10.1109/INFVIS.1999.801851

[80] Markus Wagner, Djordje Slijepcevic, Brian Horsak, Alexander Rind, Matthias Zeppelzauer, and Wolfgang Aigner. 2019. KAVAGait: Knowledge-Assisted Visual Analytics for Clinical Gait Analysis. IEEE Transactions on Visualization and Computer Graphics 25, 3 (2019), 1528–1542. doi:10.1109/TVCG.2017.2785271

[81] James Walker, Rita Borgo, and Mark W. Jones. 2016. TimeNotes: A Study on Effective Chart Visualization and Interaction Techniques for Time-Series Data. IEEE Transactions on Visualization and Computer Graphics 22, 1 (2016), 549–558. doi:10.1109/TVCG.2015.2467751

[82] Qian Wang, Yan Zhang, Xiao Lu, Zhibo Wang, Zhan Qin, and Kui Ren. 2016. RescueDP: Real-time spatio-temporal crowd-sourced data publishing with differential privacy. In Proceedings of IEEE INFOCOM - The 35th Annual IEEE International Conference on Computer Communications. IEEE Press, San Francisco, CA, USA, 1–9. doi:10.1109/INFOCOM.2016.7524458

[83] Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Y Zhang, and JUN ZHOU. 2024. TimeMixer: Decomposable Multiscale Mixing for Time Series Forecasting. In International Conference on Learning Representations (ICLR). doi:10.48550/arXiv.2405.14616

[84] Teng Wang, Xinyu Yang, Xuebin Ren, Wei Yu, and Shusen Yang. 2022. Locally Private High-Dimensional Crowdsourced Data Release Based on Copula Functions. IEEE Transactions on Services Computing 15, 2 (2022), 778–792. doi:10.1109/TSC.2019.2961092

[85] Feng Xia, Jinzhong Wang, Xiangjie Kong, Zhibo Wang, Jianxin Li, and Chengfei Liu. 2018. Exploring Human Mobility Patterns in Urban Scenarios: A Trajectory Data Perspective. Comm. Mag. 56, 3 (March 2018), 142–149. doi:10.1109/MCOM.2018.1700242

[86] Cong Xie, Wei Chen, Xinxin Huang, Yueqi Hu, Scott Barlowe, and Jing Yang. 2014. VAET: A Visual Analytics Approach for E-Transactions Time-Series. IEEE Transactions on Visualization and Computer Graphics 20, 12 (2014), 1743–1752. doi:10.1109/TVCG.2014.2346913

[87] Kai Xiong, Cynthia A Huang, Michael Wybrow, and Yingcai Wu. 2025. TableCanoniser: Interactive Grammar-Powered Transformation of Messy, Non-Relational Tables to Canonical Tables. In Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems (CHI '25). Association for Computing Machinery, New York, NY, USA, Article 849, 20 pages. doi:10.1145/3706598.3714321

[88] Ke Xu, Shunan Guo, Nan Cao, David Gotz, Aiwen Xu, Huamin Qu, Zhenjie Yao, and Yixin Chen. 2018. ECGLens: Interactive Visual Exploration of Large Scale ECG Data for Arrhythmia Detection. In Proceedings of CHI Conference on Human Factors in Computing Systems. Association for Computing Machinery, New York, NY, USA, 1—-12. doi:10.1145/3173574.3174237

[89] Panpan Xu, Honghui Mei, Liu Ren, and Wei Chen. 2017. ViDX: Visual Diagnostics of Assembly Line Performance in Smart Factories. IEEE Transactions on Visualization and Computer Graphics 23, 1 (2017), 291–300. doi:10.1109/TVCG.2016.2598664

[90] Qingqing Ye, Haibo Hu, Kai Huang, Man Ho Au, and Qiao Xue. 2023. Stateful Switch: Optimized Time Series Release with Local Differential Privacy. In Proceedings of IEEE INFOCOM - IEEE Conference on Computer Communications. IEEE, New York City, NY, USA, 1–10. doi:10.1109/INFOCOM53939.2023.10229063

[91] Qingqing Ye, Haibo Hu, Ninghui Li, Xiaofeng Meng, Huadi Zheng, and Haotian Yan. 2021. Beyond Value Perturbation: Local Differential Privacy in the Temporal Setting. In Proceedings of IEEE INFOCOM - IEEE Conference on Computer Communications. IEEE, Vancouver, BC, Canada, 1–10. doi:10.1109/INFOCOM42981.2021.9488899

[92] Jinsung Yoon, Daniel Jarrett, and Mihaela van der Schaar. 2019. Time-series Generative Adversarial Networks. In Proceedings of Advances in Neural Information Processing Systems. Curran Associates Inc., Red Hook, NY, USA, 5509–5519. https://dl.acm.org/doi/10.5555/3454287.3454781

[93] Xinyu Yuan and Yan Qiao. 2024. Diffusion-TS: Interpretable Diffusion for General Time Series Generation. In Proceedings of the 12th International Conference on Learning Representations. doi:10.48550/arXiv.2403.01742

[94] Xiaoqi Yue, Dan Feng, Desheng Sun, Chao Liu, Hongxing Qin, and Haibo Hu. 2024. AirPollutionViz: visual analytics for understanding the spatio-temporal evolution of air pollution. Journal of Visualization 27, 2 (2024), 215–233. doi:10.1007/s12650-024-00958-2

[95] Jiajun Zhang, Xiaohui Liang, Zhikun Zhang, Shibo He, and Zhiguo Shi. 2017. Re-DPoctor: Real-Time Health Data Releasing with W-Day Differential Privacy. In Proceedings of GLOBECOM - IEEE Global Communications Conference. IEEE Press, Singapore, 1–6. doi:10.1109/GLOCOM.2017.8254014

[96] Xueru Zhang, Mohammad Mahdi Khalili, and Mingyan Liu. 2022. Differentially Private Real-Time Release of Sequential Data. ACM Trans. Priv. Secur. 26, 1 (2022), 1–29. doi:10.1145/3544837

[97] Yixuan Zhang, Kartik Chanana, and Cody Dunne. 2019. IDMVis: Temporal Event Sequence Visualization for Type 1 Diabetes Treatment Decision Support. IEEE Transactions on Visualization and Computer Graphics 25, 1 (2019), 512–522. doi:10.1109/TVCG.2018.2865076

[98] Yue Zhao, Yunhai Wang, Jian Zhang, Chi-Wing Fu, Mingliang Xu, and Dominik Moritz. 2022. KD-Box: Line-segment-based KD-tree for Interactive Exploration of Large-scale Time-Series Data. IEEE Transactions on Visualization and Computer Graphics 28, 1 (2022), 890–900. doi:10.1109/TVCG.2021.3114865

[99] Zhiguang Zhou, Yize Li, Yuna Ni, Weiwen Xu, Guoting Hu, Ying Lai, Peixiong Chen, and Weihua Su. 2024. VisCI: A visualization framework for anomaly detection and interactive optimization of composite index. Visual Informatics 8, 2 (2024), 1–12. doi:10.1016/j.visinf.2024.04.001

[100] Tomasz Zieliński, Johnny Hay, and Andrew J. Millar. 2022. Period Estimation and Rhythm Detection in Timeseries Data Using BioDare2, the Free, Online, Community Resource. Springer US, New York, NY, 15–32. doi:10.1007/978-1-0716-1912-4_2