A Survey of Generative Techniques for Spatial-Temporal Data Mining

---

Abstract

This paper delves into the integration of cutting-edge generative techniques in spatial-temporal data mining, a field of increasing importance in today's data-driven world. With the advent of GPS, IoT, and mobile technologies, spatial-temporal datasets have become more complex and voluminous. Traditional approaches, such as Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs), have been pivotal in analyzing these datasets. However, limitations in scalability, generalization, and interpretability have necessitated the exploration of generative methods, including:

- Large Language Models (LLMs): Powerful in capturing dependencies and offering zero-shot capabilities.
- Diffusion Models: Effective in simulating and forecasting dynamic systems.
- Self-Supervised Learning (SSL): Valuable for leveraging vast amounts of unlabeled data.
- Sequence-to-Sequence (Seq2Seq) Frameworks: Useful in handling trajectory data and forecasting tasks.

This survey presents a taxonomy of these techniques, explores their applications in domains like transportation, healthcare, and climate science, and outlines future research directions to address challenges such as real-time analysis, multi-modal data integration, and interpretability.

Introduction

The rapid advancement of GPS technologies and IoT has generated massive amounts of spatial-temporal data, spanning diverse sectors such as:

1. Urban Planning:
    - Traffic optimization.
    - Infrastructure development.
2. Healthcare:
    - Disease outbreak tracking.
    - Predictive modeling for resource allocation.
3. Climate Science:
    - Weather forecasting.
    - Disaster preparedness.
4. Transportation:

- Route optimization.
- Autonomous vehicle systems.

Limitations of Traditional Techniques

While methods like RNNs and CNNs have been instrumental in processing spatial-temporal data, they exhibit shortcomings such as:

1. Scalability Issues:
    - Difficulty in processing data with high spatial and temporal resolutions.
    - High computational costs for training and inference.
2. Lack of Generalization:
    - Inability to adapt models trained on one dataset to another.
    - Overfitting due to biases in training data.
3. Insufficient Interpretability:
    - Limited insights into underlying patterns and correlations.

Generative techniques address these challenges by providing a unified framework capable of handling diverse datasets and tasks.

Motivation

Why Generative Techniques?

Generative models have shown exceptional performance in domains like computer vision and NLP, where they are used for tasks such as image generation, machine translation, and summarization. Their ability to generalize, adapt, and generate new data makes them ideal for spatial-temporal data mining.

Research Gaps

Despite their potential, existing literature lacks:

- A detailed taxonomy of generative techniques for spatial-temporal data.
- Standardized frameworks for applying these techniques to real-world problems.
- Comprehensive evaluations of their performance across diverse datasets.

3. Data

Spatial-temporal data encompasses various types and is collected using different methods for real-world applications. It provides insights into patterns and relationships across space and time and is widely used in fields such as transportation, urban planning, and environmental monitoring. This section discusses the properties, types, and instances of spatial-temporal data.

3.1 Data Properties

Spatial-temporal data exhibits unique characteristics that pose challenges and opportunities for data mining. Two critical properties include:

1. Correlations:
   - Definition: Interdependencies between spatial and temporal aspects within a dataset.
   - Challenges:
     - Prediction Accuracy: Intricate patterns can hinder accurate predictions for tasks like traffic forecasting and weather prediction.
     - Data Preprocessing and Fusion: Integrating data from heterogeneous sources while preserving spatial-temporal context requires sophisticated preprocessing techniques.
   - Opportunities: Accurately modeling these correlations can lead to better predictive performance and deeper insights.
2. Heterogeneity:
   - Definition: Variability in spatial and temporal patterns across regions and time periods.
   - Challenges:
     - Generalization: Models trained on specific regions or times may not perform well in other contexts due to diverse patterns.
     - Biases and Imbalances: Uneven data distribution can result in skewed analyses and biased conclusions.
   - Opportunities: Developing robust models that generalize well across different contexts can enhance the applicability of spatial-temporal data mining methods.

3.2 Data Types

Spatial-temporal data is categorized into four types, each representing unique phenomena across space and time:

1. Event Data:
   - Definition: Represents discrete events with spatial and temporal attributes, such as crime reports or voting events.
   - Key Features:
     - Includes marked variables, like crime type or political affiliation.
     - Often represented using Euclidean or network distances.
   - Example: Crime events marked with type and location, visualized in Figure 2.
2. Trajectory Data:

- Definition: Tracks the movement of objects over time as sequences of spatial points.
- Key Features:
  - Typically collected via GPS sensors.
  - Higher sampling frequencies improve accuracy.
- Example: Taxi trajectories represented as sequences of GPS points (p1,t1),(p2,t2),...(p_1, t_1), (p_2, t_2),..., shown in Figure 2.

3. Point Data:
   - Definition: Captures continuous spatial-temporal field measurements at mobile reference points.
   - Key Features:
     - Examples include temperature readings at weather stations.
     - Provides valuable observations for reconstructing spatial-temporal fields.
   - Example: Weather data collected at specific points, visualized in Figure 3.
4. Raster Data:
   - Definition: Represents fixed-location observations on a spatial grid.
   - Key Features:
     - Used for satellite imagery, digital elevation models, and climate variables.
     - Captures static geographic phenomena over time.
   - Example: Satellite imagery with fixed pixels, as shown in Figure 4.

3.3 Data Instances

Data instances are structured examples or observations used for analysis and modeling in spatial-temporal data mining. They vary depending on the data type and research objectives.

1. Types of Data Instances:
   - Points: Represent specific locations and timestamps (e.g., event data).
   - Trajectories: Collections of spatial-temporal points forming movement paths.
   - Time Series: Sequential data for spatial or temporal attributes.
   - Spatial Maps: Represent spatial data at a fixed time.
   - Spatial-Temporal Rasters: Represent entire datasets with static grids.
2. Applications:
   - Event data can be represented as point instances.

- Trajectory data can be structured as point collections or time series.
- Raster data can be analyzed as spatial maps or time series, depending on the task.

3. Mapping Relationships:
    - Figure 5 shows the relationships between data types and their corresponding instances, providing flexibility for diverse data mining tasks.

Spatial-temporal data provides a rich source of information but requires specialized methods to address its inherent complexities. Understanding its properties, types, and instances is essential for effective analysis and modeling.

Generative Techniques Overview

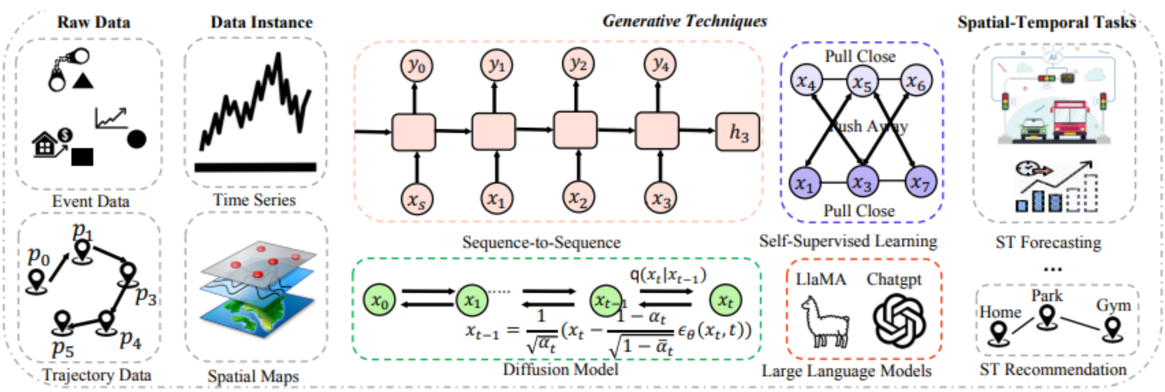

Fig. 6. Framework of Spatial-Temporal Data Mining via *generative techniques.*

4.1 Large Language Models (LLMs)

LLMs are highly capable models trained on extensive text datasets. Their transformer-based architecture employs self-attention mechanisms, enabling them to process and model complex dependencies in sequential data. LLMs are widely applied in computer vision (CV) and natural language processing (NLP), excelling in tasks like text generation, named entity recognition, and text categorization.

Classification of LLMs

LLMs are categorized into three architectural types:

1. Only-Encoder Models:
    - Examples: BERT and its variants.
    - Functionality: Uses the encoder module to understand and encode input sentences into meaningful representations.

- Strengths: Excels in understanding full-context tasks like text categorization and named entity recognition.
- Example Use Cases: Semantic understanding in question-answering systems.

2. Encoder-Decoder Models:
    - Functionality: Combines encoder (to encode input into hidden representations) and decoder (to generate outputs).
    - Strengths: Flexible and suitable for generative tasks, such as summarization and machine translation.
    - Example Use Cases: Neural machine translation systems.
3. Only-Decoder Models:
    - Examples: GPT-3, ChatGPT, and GPT-4.
    - Functionality: Uses the decoder module to generate outputs by predicting the next token in a sequence.
    - Strengths: Capable of few-shot or zero-shot learning, reducing the need for fine-tuning.
    - Recent Advancements: Open-source only-decoder models, such as Alpaca and Vicuna, provide accessible alternatives for researchers.

4.2 Diffusion Models (DMs)

Diffusion Models are probabilistic frameworks that simulate data transformations through a two-step process: the forward process (diffusion) and the reverse process (denoising).

Forward Process

- Describes the gradual addition of Gaussian noise to data over time.
- Mathematically models the evolution of latent variables (x1,x2,...,xTx_1, x_2, ..., x_T) through a Markov chain:

$$q(xt|xt-1) = N\ (xt;\ \mu t(xt-1),\ \beta tI)$$

    - xt : Latent variable at time tt.
    - Mt : Mean function, often parameterized by neural networks.
    - Bt : Variance parameter controlling noise addition.
- The sequence approximates the posterior distribution q(x1:T|x0), where x0 is the original data point.

Reverse Process

- Generates clean samples by reversing the noise added during the forward process.
- Involves iteratively sampling from the conditional distribution: $p(xt|xt+1)=N(xt;\mu\theta(xt+1,t+1),\Sigma\theta(xt+1,t+1))$

- $p(x_0)$: Final clean sample representing the data distribution.
- μ_θ and Σ_θ: Learnable parameters optimized during training.

Training and Optimization

- Training minimizes the difference between generated samples and observed data by optimizing the negative log-likelihood:
  Objective: $E[-\log p_\theta(x_0)] \leq$ Reconstruction Loss
- The model learns to approximate the reverse process, ensuring realistic sample generation.

Applications of Diffusion Models

- Weather Simulation: Predicting dynamic environmental changes, such as rainfall or wind patterns.
- Trajectory Modeling: Generating plausible movement paths in GPS data.
- Data Augmentation: Creating synthetic datasets for under-sampled scenarios.

Summary of Key Insights

- LLMs provide robust capabilities for sequential modeling with varying architectural designs (only-encoder, encoder-decoder, only-decoder), enabling tasks ranging from understanding textual context to generative text outputs.
- Diffusion Models offer powerful probabilistic mechanisms for modeling and generating data, excelling in applications requiring uncertainty handling and realistic sample generation.

These techniques complement each other, addressing diverse challenges in spatial-temporal data mining and beyond.

4.3 Self-Supervised Learning (SSL)

Self-Supervised Learning (SSL) is a framework designed to leverage unlabeled data through surrogate (or pretext) tasks that enable models to learn meaningful data representations. By utilizing these tasks, SSL helps models acquire valuable features and generalize effectively to downstream tasks without explicit supervision.

Applications of SSL

SSL has been widely applied across various domains, including:

- Computer Vision (CV):
  - Tasks like image recognition, object detection, and segmentation.
  - Example: Learning features from unlabeled images for downstream image classification tasks.
- Graph Data:

- Techniques like reconstructing adjacency information for graph representation learning.

Categories of SSL

1. Contrastive Self-Supervised Learning:
   - Focus: Aligning data representations under appropriate transformations (e.g., augmentation).
   - Popular in visual representation learning, leveraging techniques to align local and global representations.
   - Recent advances extend contrastive learning to graph data, capturing structural information by contrasting local and global contexts.
   - Limitation: Lack of invariance to perturbations compared to prior works in image-based contrastive learning.
2. Generative Self-Supervised Learning:
   - Focus: Training models to generate meaningful data representations in an unsupervised way.
   - Uses generative models like Variational Autoencoders (VAEs) and Generative Adversarial Networks (GANs) to model data distributions.
   - Applications:
     - Image generation, data augmentation, and semantic representation.
   - Notable Works:
     - InfoGAN: Adds information-theoretic constraints for disentangled representation learning.
     - β-VAE: Encourages disentanglement with a regularization term.
     - Adversarial Autoencoders (AAEs): Combines VAEs and GANs for better diversity and quality in generative representations.

SSL methods have transformed learning from large-scale unlabeled datasets, enabling efficient feature extraction and representation across diverse domains.

4.4 Sequence-to-Sequence (Seq2Seq)

Sequence-to-Sequence (Seq2Seq) models are widely used for transforming one sequence into another, making them ideal for tasks involving sequential data. These models are built on an encoder-decoder architecture, which captures input context and generates output sequences.

Architecture

1. Encoder:

- Processes the input sequence and encodes it into a hidden representation that captures essential context information.

2. Decoder:
   - Generates the output sequence step-by-step, leveraging the hidden context from the encoder.
   - Employs techniques like autoregressive decoding, where each output token is conditioned on the previously generated token.

Enhancements with Attention Mechanisms

- The attention mechanism dynamically focuses on different parts of the input sequence while generating output tokens.
- This significantly improves the model's ability to handle long-range dependencies, making it a crucial component in Seq2Seq success across various domains.

Applications

1. Machine Translation:
   - Example: The attention-based model by Bahdanau et al. enhances translation quality by focusing on relevant input tokens dynamically.
2. Text Summarization:
   - Example: Nallapati et al. demonstrated Seq2Seq's ability to generate concise and informative text summaries.
3. Speech Recognition:
   - Converts spoken language into written text.
   - Example: Work by Chan et al. showcases Seq2Seq's effectiveness in transcribing speech.

Seq2Seq models, with the integration of attention mechanisms, have achieved state-of-the-art performance in tasks requiring sequential data transformations, demonstrating their versatility and power in real-world applications.

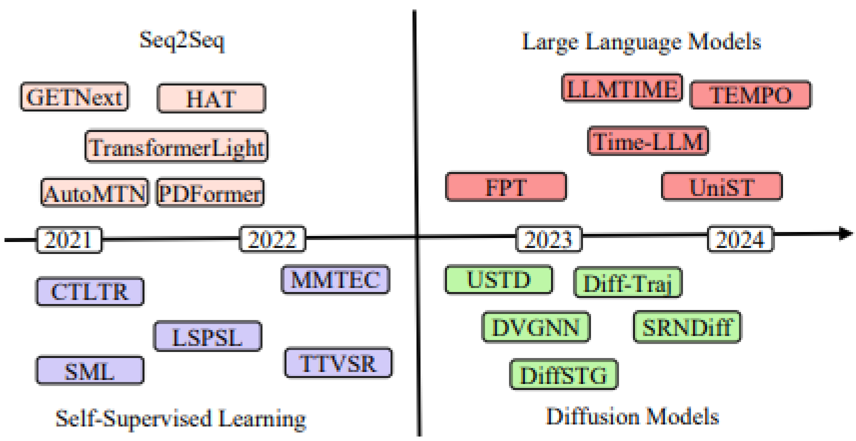

Fig. 1. Examples of existing studies via *generative techniques*.

5. Framework for Spatial-Temporal Data Mining

This section outlines a comprehensive approach to address spatial-temporal data mining challenges using generative techniques. A structured pipeline is introduced, encompassing data preprocessing, data transformation, and the application of generative models to solve various spatial-temporal tasks.

Pipeline Overview

- Data Collection: Raw spatial-temporal data is gathered from various sources, such as:
    - Event data (e.g., timestamps of occurrences).
    - Trajectory data (e.g., vehicle movement paths).
    - Point reference data (e.g., GPS locations).
    - Raster data (e.g., satellite imagery).
- Data Instance Creation: The collected data is structured into formats such as:
    - Points
    - Time series
    - Spatial maps
    - Trajectories
    - Spatial-temporal rasters
- Data Representation: Depending on the generative technique used, the data instances are transformed into specific formats, including:
    - Sequences
    - Matrices
    - Tensors
    - Graphs
- Generative Techniques: Models like LLMs, Diffusion Models, SSL, and Seq2Seq are applied to perform tasks such as:
    - Prediction
    - Classification
    - Representation learning

This framework is designed to extract meaningful insights from spatial-temporal data while addressing its inherent complexities.

5.1 Spatial-Temporal Data Preprocessing

Preprocessing is a crucial step to ensure data quality, consistency, and usability. Key objectives include:

- Data Cleaning: Removing errors, handling missing values, and managing outliers to improve reliability.
- Data Transformation: Converting raw data into usable representations (e.g., trajectories, time series).
- Integration: Merging datasets from multiple sources to provide a unified view for analysis.

Effective preprocessing is essential for preparing spatial-temporal data for subsequent generative modeling tasks.

5.2 The Spatial-Temporal Data Mining Framework

The framework integrates advanced generative techniques to handle diverse spatial-temporal datasets effectively. Key steps include:

1. Data Preparation: Creating data instances (e.g., time series, spatial maps) from raw input.
2. Transformation: Representing instances in formats compatible with generative techniques.
3. Model Application: Leveraging generative models to perform tasks like:
    - Forecasting
    - Clustering
    - Recommendation
    - Representation learning

This structured pipeline streamlines spatial-temporal data mining, enhancing scalability and accuracy.

---

6. Applications of Generative Techniques in Spatial-Temporal Data Mining

Generative techniques find applications in various domains of spatial-temporal analysis. A taxonomy is presented to classify these applications into four categories of generative models: LLMs, SSL, Diffusion Models, and Seq2Seq. Key application areas include:

6.1 Spatial-Temporal Representation Learning

- Focus: Generating embeddings or representations for spatial-temporal data.
- Use Cases:
    - Understanding urban mobility patterns.
    - Identifying trends in climate datasets.

6.2 Spatial-Temporal Forecasting

- Focus: Predicting future trends or states of spatial-temporal phenomena.
- Use Cases:
    - Traffic flow prediction using Seq2Seq models.
    - Weather simulation with diffusion models.

6.3 Spatial-Temporal Recommendation

- Focus: Personalized recommendations based on spatial-temporal behavior.
- Use Cases:
    - Suggesting optimal routes in navigation systems.
    - Recommending popular points of interest based on movement patterns.

6.4 Spatial-Temporal Clustering

- Focus: Grouping spatial-temporal data points into meaningful clusters.
- Use Cases:
    - Identifying high-crime zones.
    - Clustering urban areas based on population movement.

Taxonomy and Datasets

A taxonomy of generative techniques for spatial-temporal tasks highlights specific studies tailored for different applications. Example datasets and tasks include:

- Traffic data for mobility analysis.
- Weather datasets for forecasting.
- Social event logs for clustering and recommendation.

By systematically categorizing these applications, researchers can better understand the methodologies and innovations in spatial-temporal analysis, fostering advancements in the field.

6.1 Spatial-Temporal Representation Learning

Representation learning aims to generate high-quality embeddings for spatial-temporal data, such as trajectories and spatial maps. These representations enhance downstream tasks like prediction, classification, and clustering.

6.1.1 Trajectories

- Non-Generative Methods:
    - Traditional methods use RNNs and LSTMs for trajectory data from GPS devices.

- Example: Yang et al. introduced models combining RNNs and GRUs to capture short- and long-term patterns.
- Temporal Convolutional Networks and residual frameworks like STResNet are used to process crowd traffic patterns.

- Generative Methods:
  - Recent studies use self-supervised learning (SSL) and contrastive learning to enhance trajectory representation.
  - Example: Liu et al. proposed contrastive learning for trajectory-level and point-level distinctions.
  - Seq2Seq models, such as Traj2Vec, encode trajectories using road networks and temporal-spatial properties.
  - Conditional Score-based Diffusion models (e.g., CSDI) are employed for time-series data imputation.

6.1.2 Spatial Maps

- Non-Generative Methods:
  - Techniques like HDGE, ZE-Mob, and MVURE construct graphs from mobility and POI data to capture inter- and intra-region dependencies.
  - Example: CGAL creates region representations by learning from POI and mobility graphs.
- Generative Methods:
  - Generative models, such as AutoST and GraphST, apply contrastive learning to create spatial-temporal graph embeddings for downstream tasks like crime and traffic prediction.
  - Time-Aware Location Embedding (e.g., TALE) uses temporal details for location visitor flow predictions.
  - Diffusion models (e.g., SADI) are applied to handle incomplete spatial-temporal datasets, such as electronic health records.

6.2 Spatial-Temporal Forecasting

Spatial-temporal forecasting involves predicting future states based on input data. Applications include traffic flow prediction, crime forecasting, and climate modeling.

6.2.1 Points

- Non-Generative Methods:
  - Graph-based models, such as DSTAGNN and STGNCDE, use dynamic graph neural networks and controlled differential equations for traffic prediction.
- Generative Methods:

   - STGCL integrates contrastive learning with traffic data analysis.
   - Models like STTN and PDFormer use transformer architectures to capture spatial-temporal dependencies.
   - Diffusion-based approaches (e.g., DiffSTG and DVGNN) improve forecasting accuracy by modeling uncertainty.

6.2.2 Trajectories

- Non-Generative Methods:
   - Temporal Convolutional Networks and residual architectures are employed for trajectory prediction.
   - Example: DCRNN fuses diffusion convolution and RNNs for spatial-temporal dependencies.
- Generative Methods:
   - Contrastive SSL models like SPGCL refine trajectory relations by optimizing graph structures.
   - Diffusion models, such as Diff-Traj and TrajGDM, simulate stochastic human movement and urban flows, addressing data sparsity and privacy concerns.
   - Transformer-based architectures like ST-LLM treat spatial-temporal tokens as input to improve traffic forecasting.

6.2.3 Time Series

- Non-Generative Methods:
   - Methods like GWN and DSANet combine graph convolutions with temporal modeling for traffic prediction.
   - Example: Z-GCNETs apply zigzag topological layers for Ethereum blockchain price predictions.
- Generative Methods:
   - Advanced transformer models, such as GraphERT and DVGNN, combine graph learning with temporal analysis.
   - Large Language Models (LLMs):
      - Techniques like LLMTIME and PromptCast align time-series data with textual formats for zero-shot or few-shot learning.
      - Models like GPT4MTS and TEMPO integrate trend, seasonal, and residual components for effective time-series representation.

Here is the summarized content from the provided text:

6.2.4 Spatial Maps

Spatial maps are represented as matrices and are used for predictive learning tasks such as air quality monitoring and road traffic analysis.

- Non-Generative Methods:
  - ASTGCN: Combines spatial-temporal graph convolution networks with attention mechanisms to improve predictions.
  - Applications: Traffic flow prediction and POI distribution.
- Generative Methods:
  - Diffusion Models:
    - Example: DiffSTG captures a range of possible future outcomes and validates performance on air quality monitoring.
    - USTD: Uses shared spatial-temporal encoders and task-specific attention-based denoising for accurate air quality predictions.

6.2.5 Spatial-Temporal Rasters

Raster data combines spatial and temporal dimensions, such as weather maps and satellite data grids.

- Non-Generative Methods:
  - ASTGCN: Robust for dynamic weather prediction using spatial-temporal attention mechanisms.
  - GraphCast: Utilizes graph neural networks for high-resolution weather forecasts globally.
  - PastNet: Introduces Fourier-domain spectral convolution for modeling physical laws.
- Generative Methods:
  - SVT: Aligns local and global spatial-temporal features for better video-based weather data predictions.
  - W-MAE: Reconstructs spatial correlations and predicts temporal dependencies in meteorological data.
  - Diffusion Models:
    - Example: SRNDiff integrates historical radar data for accurate weather forecasting.
    - Example: Hua et al. use diffusion models for flexible weather prediction with reduced retraining needs.

6.3 Spatial-Temporal Recommendation

Recommendation systems aim to provide personalized location-based suggestions based on spatial and temporal data.

6.3.1 Points

- Non-Generative Methods:
  - Examples:
    - ST-LSTM incorporates spatial-temporal contexts into LSTMs for location recommendations.
    - Content-aware POI embedding models leverage textual content for personalized recommendations.
- Generative Methods:
  - LLMs:
    - Example: Feng et al. develop prompting strategies for user movement predictions.
  - Diffusion Models:
    - Example: Qin et al. use sampling techniques to model user preferences for POI recommendations.

6.3.2 Time Series

- Non-Generative Methods:
  - RNN-based models extract sequential features for recommendations.
- Generative Methods:
  - SSL techniques and transformers enhance feature extraction and prediction capabilities.

6.4 Spatial-Temporal Clustering

Clustering identifies patterns and groups in spatial-temporal data to reveal underlying structures.

6.4.1 Points

- Non-Generative Methods:
  - Example: TRACLUS clusters trajectory segments to uncover shared sub-trajectories.
- Generative Methods:
  - Example: Self-supervised approaches like SDTC refine trajectory clustering using dual optimization layers.

6.4.2 Trajectories

- Non-Generative Methods:
  - Partition-and-group frameworks divide trajectories into clusters.
- Generative Methods:
  - Examples:

        - E2DTC: An end-to-end clustering framework leveraging self-training.
        - Diff-Traj: Incorporates forward and reverse noising processes for trajectory generation.

6.4.3 Time Series

- Non-Generative Methods:
    - Examples:
        - Deep autoencoder networks capture features in ECGs and other time-series data.
- Generative Methods:
    - SSL techniques like SimCLR enable feature extraction with data augmentation.

---

7. Future Research Directions

1. Addressing Dataset Bias:
    - Skewed benchmark datasets impact fairness and accuracy.
    - Future work should focus on mitigating biases in data distributions.
2. Large-Scale Foundation Models:
    - The development of multi-modal datasets can improve generative model performance in spatial-temporal forecasting.
3. Generalization:
    - Improving spatial-temporal methods to adapt across diverse tasks and datasets.
4. Integration with External Knowledge:
    - Incorporating knowledge graphs can enhance the contextual understanding of spatial-temporal models.

---

8. Conclusion

The paper highlights the transformative role of generative techniques in spatial-temporal data mining, offering a detailed review of methods like LLMs, Diffusion Models, and SSL. It provides a framework for applying these techniques, emphasizing prediction, clustering, and recommendation tasks. The study identifies challenges such as dataset biases and scalability while proposing future research directions to unlock the full potential of spatial-temporal analysis. Generative models pave the way for scalable, accurate, and innovative solutions across diverse domains.

REFERENCES

[1] Z. Engin, J. van Dijk, T. Lan, P. A. Longley, P. Treleaven, M. Batty, and A. Penn, “Data-driven urban management: Mapping the landscape,” Journal of Urban Management, vol. 9, no. 2, pp. 140–150, 2020.

[2] K. Button, “City management and urban environmental indicators,” Ecological economics, vol. 40, no. 2, pp. 217–233, 2002.

[3] Z. Liu, J. Li, and K. Wu, “Context-aware taxi dispatching at city-scale using deep reinforcement learning,” IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 3, pp. 1996–2009, 2020.

[4] D.-H. Lee, H. Wang, R. L. Cheu, and S. H. Teo, “Taxi dispatch system based on current demands and real-time traffic conditions,” Transportation Research Record, vol. 1882, no. 1, pp. 193–200, 2004.

[5] M. Kampa and E. Castanas, “Human health effects of air pollution,” Environmental pollution, vol. 151, no. 2, pp. 362–367, 2008.

[6] P. W. Abrahams, “Soils: their implications to human health,” Science of the Total Environment, vol. 291, no. 1-3, pp. 1–32, 2002.

[7] K. Bi, L. Xie, H. Zhang, X. Chen, X. Gu, and Q. Tian, “Panguweather: A 3d high-resolution model for fast and accurate global weather forecast,” arXiv preprint arXiv:2211.02556, 2022.

[8] M. Schuster and K. K. Paliwal, “Bidirectional recurrent neural networks,” IEEE transactions on Signal Processing, vol. 45, no. 11, pp. 2673–2681, 1997.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” Advances in neural information processing systems, vol. 25, 2012.

[10] Y. Li, R. Yu, C. Shahabi, and Y. Liu, “Diffusion convolutional recurrent neural network: Data-driven traffic forecasting,” in ICLR. OpenReview.net, 2018.

[11] B. Yu, H. Yin, and Z. Zhu, “Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting,” in International Joint Conferences on Artificial Intelligence (IJCAI). ijcai.org, 2018, pp. 3634–3640.

[12] Q. Zhang, Z. Wang, C. Long, C. Huang, S.-M. Yiu, Y. Liu, G. Cong, and J. Shi, “Online anomalous subtrajectory detection on road networks with deep reinforcement learning,” in 2023 IEEE 39th International Conference on Data Engineering (ICDE). IEEE, 2023, pp. 246–258.

[13] K. Zhou, J. Yang, C. C. Loy, and Z. Liu, “Learning to prompt for visionlanguage models,” International Journal of Computer Vision, vol. 130, no. 9, pp. 2337–2348, 2022.

[14] B. X. Yu, J. Chang, H. Wang, L. Liu, S. Wang, Z. Wang, J. Lin, L. Xie, H. Li, Z. Lin et al., “Visual tuning,” ACM Computing Surveys, 2023.

[15] J. Duan, H. Zhao, Q. Zhou, M. Qiu, and M. Liu, “A study of pretrained language models in natural language processing,” in 2020 IEEE International Conference on Smart Cloud (SmartCloud). IEEE, 2020, pp. 116–121.