Neural Collaborative Filtering to Detect Anomalies in Human Semantic Trajectories

Yueyang Liu
Emory University, Atlanta, USA
yueyang.liu@emory.edu

Lance Kennedy
Emory University, Atlanta, USA
lance.kennedy@emory.edu

Hossein Amiri
Emory University, Atlanta, USA
hossein.amiri@emory.edu

Andreas Züfle
Emory University, Atlanta, USA
azufle@emory.edu

Abstract

Human trajectory anomaly detection is critical for applications such as security surveillance and public health, yet most existing methods focus on vehicle-level traffic, with limited attention to human-level trajectories. Due to the inherent sparsity of human trajectory data, machine learning approaches are favored for detecting complex patterns. However, concerns about model biases and robustness have highlighted the need for more transparent and explainable solutions. In this paper, we propose a lightweight anomaly detection model specifically designed to detect anomalies in human trajectories. We propose a Neural Collaborative Filtering approach to model and predict normal mobility. Our method is designed to model users' daily patterns of life without requiring prior knowledge, thereby enhancing performance in scenarios where data is sparse or incomplete, such as in cold start situations. Our algorithm consists of two main modules. The first is the collaborative filtering module, which applies collaborative filtering to model normal mobility of individual humans to places of interest. The second is the neural module, responsible for interpreting the complex spatio-temporal relationships inherent in human trajectory data. To validate our approach, we conducted extensive experiments using simulated and real-world datasets comparing to numerous state-of-the-art trajectory anomaly detection approaches.

CCS Concepts

• **Information systems** → **Geographic information systems**; **Location based services**.

Keywords

Recommendation Systems, Collaborative Filtering, Neural Networks
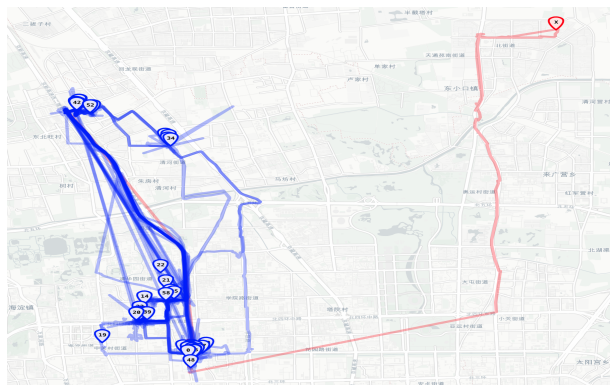
Figure 1: A sample of anomaly trajectory (red) and Normal trajectory (blue).

1 Introduction

A trajectory is a trace generated by a moving object in a geographic space, usually represented by a sequence of chronologically ordered points [61]. A semantic trajectory [38] is a sequence of chronologically ordered points where some points may be enriched with semantic information such as the type of a visited place of interest. Figure 1 shows an example of a semantic trajectory of a user in the GeoLife [62] dataset in Beijing, China. Blue points form lines to visualize the trajectory of the user and location markers highlight staypoints that can be reverse geocoded to obtain semantic information of visited places.

An anomaly is commonly defined as a data point that is significantly different from the remaining data. Anomalies are also referred to as abnormalities, discordants, deviants, or anomalies in the data mining and statistics literature [1, 18, 46]. In this example, the monitored user mainly stays within a certain area and only has a single long-distance trip. Finding anomalous behavior in semantic trajectories has many applications:

- **Infectious Disease Early-Warning:** a large number of people staying at home from work may indicate an outbreak of an infectious disease even before people notice more severe symptoms [37].
- **Elder Monitoring:** A person suffering from dementia wandering the park at 11pm may indicate that the person might be lost and in need of assistance [47, 51].
- **Child Safety:** A child being taken to a far away house when they should be at school may indicate a possible child abduction [16].

Despite these important applications, the implementation of systems to support them is challenging, as it becomes very difficult

to decide: What part of a trajectory is anomalous? What part is normal? For example, Figure 1 shows the six-week trajectory of a user in GeoLife. This trajectory allows us to observe the users patterns of life: including their home location, their work or school location, and other places the user likes to visit. Existing work on trajectory-based user identification has shown that an individual's patterns of life are like a unique fingerprint [15, 45], thus allowing identification of any individual human even among a large crowd of users. But the goal of this work is not to identify users, but to detect anomalous behavior within a user's trajectory. Such anomalies, of having users deviate from their normal patterns of life, may indicate a stolen phone or indicate distress, such as an abducted child or a lost elder with dementia. To illustrate the problem of trajectory anomaly detection, Figure 1 depicts the multi-week trajectory of GeoLife user 15. Let us assume that this trajectory belongs to a nine-year-old child named Maria. Also, assume that the one trip to the East of the City happened during a time when Maria would normally be at school. Do we think that this trip is anomalous? Could it be that Maria was abducted? We don't really have enough information to answer this question without knowing the semantics of the visited place. Maybe this is the location of the National History Museum that high school classes commonly visit and Maria is happily looking at dinosaur fossils? Maybe this is the location of the home of Maria's friend, whom other children like Maria often visit to skip school. None of these cases would be cause for safety concerns. We use this stylized example to show that we need to know more than just time and location to infer anomalous behavior: We need the semantics of places: What other individuals visited this place? Even though Maria never visited this place: Is it normal for others like Maria to visit this place?

Over the past few decades, a considerable amount of research has been dedicated to trajectory anomaly detection across multiple domains, including human mobility [7, 34], maritime [33], and transportation [14]. These works mainly define anomaly trajectories as those that exhibit a large distance (in space and time) from other trajectories. However, the above example shows that space and time alone do not suffice to discriminate between normal-but-not-yet-observed and anomalous movement. To fill this research gap, we propose to consider using the semantics of locations: What is the (latent) semantic or purpose of a visit?

But how can we infer the latent semantics of a trip when trajectory data only includes location information? To answer this question, we propose to use a Collaborative Filtering (CF) approach. CF is traditionally used in recommendation systems and leverages the known preferences of a group of users to make recommendations or predict the unknown preferences of other users [48]. For example, in the Netflix Prize competition, the winning proposal, "Cinematch," demonstrated the effectiveness of CF by surpassing Netflix's own system by more than 6% [8]. The advantage of CF is that it does not require semantic data to infer semantic information: In the movie recommendation example, without needing to know the semantics of movies (genres, actors, etc.) and without knowing the preferences of users (children, action, anime, love stories), CF allows one to estimate the rating that a user will give to a movie by looking at similar users (similar in terms of what movies they liked) and similar movies (similar in terms of liked by similar users).

Figure 2: A stylized example of a User-POI matrix

Figure 3: Expected User-POI visits obtained from the User-POI Matrix of Figure 2. Obtained via Randomized Singular Value Decomposition having three latent components.

While these systems estimate the likelihood that a user will like a movie, our idea is to use collaborative filtering to estimate the likelihood that a person will visit a place. Even though Maria never visited the National History Museum and even though the Museum is far from her home and school, our model may give a high likelihood for Maria to visit the Museum. That's because other children (who visit places similar to Maria, such as elementary schools, playgrounds but not universities and bars) have also visited the Museum in the past. The CF will give us, for each (person, place)-pair a likelihood that the user would (normally) visit this place. For (person, place) pairs for which the model estimates a very low likelihood of a visit, the model tells us: "This person should not normally visit this place". If we observe such a visit in the data, the model will be surprised. We can measure this surprise as the difference between the expected value and the observed value in test and use it report anomalous.

To illustrate this concept, Figure 2 presents a stylized example of a User-POI matrix. This matrix captures the number of visits by each of five users to eight different Points of Interest (POIs). For instance, User 1 visits a house (likely their home) 34 times, an office building (likely their workplace) 20 times, a restaurant (possibly their favorite) 8 times, and a bar three times. User 2 visits entirely different locations, while User 5 lives in the same area as User 1 and frequents similar places. Figure 3 shows the same matrix after factorization using Singular Value Decomposition (SVD) with three latent components and re-expansion of the factor matrices. Thus, Figure 3 is the result of a coder-decoder approach: SVD encodes the matrix into a compressed representation that aims at filtering noise and retaining signal. Then, the re-expansion of the factor matrices decodes this latent representation into a full matrix after dropping the least important principal components. Intuitively, we can interpret Figure 3 as a recommendation matrix: Where does the SVD model think each user should normally go? Figure 4 then measures the reconstruction difference between the original User-POI matrix and the SVD-generated matrix. A high surprise score
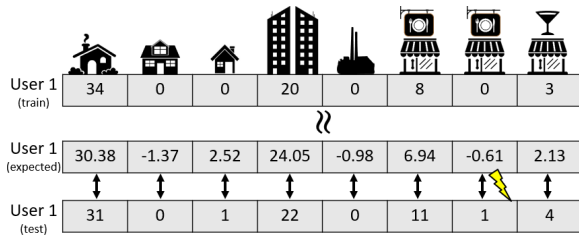
User 1 (train)	34	0	0	20	0	8	0	3
User 1 (expected)	30.38	-1.37	2.52	24.05	-0.98	6.94	-0.61	2.13
User 1 (test)	31	0	1	22	0	11	1	4

Figure 4: Measuring Surprise between the expected User-POI Matrix and observed User-POI visits.

indicates that the user is more likely to visit a POI that they are not typically expected to visit. We see that User 1 visited two places in the test period that were never visited by User 1 in the train period. However, the model judges the visit to the house (third column in Figure 4) as non-surprising. Because others like User 1 visited this house, the model even expected User 1 to visit this house 2.52 times. However, the new restaurant that User 1 visited (second column from the right in Figure 4) yields a high model surprise. The model estimated User 1 to visit this place -0.61 times. This means that the model did not expect User 1 to visit this location as other users like User 1 did not visit this location (or other locations like this one). We can use this surprise to flag this visit as a potential anomaly.

Despite the broad applications for anomaly detection in semantic trajectories, this area remains substantially under-researched. A main reason is a lack of semantic trajectory data with ground truth anomaly labels. Both privacy considerations and a lack of participant initiative result in datasets that are sparsely populated [11, 19, 49]. For example, the commonly used GeoLife dataset [62] includes only 182 users. But over the five years covered by the dataset, more than 60% of the users have fewer than 50 recorded trajectories, and around 70% of POIs are visited by only one or two users. In addition, this dataset has no ground truth on whether any segments of the trajectories can be considered as anomalous.

We also step towards filling this research gap by providing trajectory datasets that have labels for anomalous trajectories. We provide (1) a dataset generated via user-based simulation for which we change the decision-making process of users while they are anomalous, and (2) a dataset generated from GeoLife in which we artificially introduce anomalous behavior by replacing the normal trajectory of users with the trajectory of other users.

To address the sparsity of the data, we leverage a deep learning approach to fill the gaps of missing data. Existing work in trajectory mining focusing primarily on temporal information for sequence embedding or integrating spatial information into Graph Neural Network (GNN) architectures [13, 53]. However, these approaches significantly increase computational costs [35] and require high-quality POI annotation. To address these challenges, we propose a collaborative filtering-based unsupervised anomaly detection approach in the context of human trajectory anomaly detection tasks.

To address those issues, we provide a modified neural collaborative filtering architecture to capture person-level human trajectory anomalous. The main contributions of this work are as follows.

(1) We present a neural collaborative filtering architecture to model the users' daily patterns of life without prior knowledge.
(2) We ascertain that neural collaborative filtering (NCF) can be effectively applied to the spatial-temporal anomaly detection domain. By employing an appropriate surprise function and

training strategy, the model's cold start performance can be significantly enhanced.
(3) We perform extensive experiments on three highly sparse and extremely imbalanced datasets [2, 4, 62] to demonstrate the effectiveness and the promise of our neural collaborative filtering approaches on the user-level spatial-temporal anomaly detection domain.

This paper is organized as follows: In Section II, we explicate the basic concepts of user-level spatial-temporal trajectories and anomalous, as well as various anomaly detection methods and collaborative filtering techniques. For better understanding, Section III introduces the fundamental concepts and definitions of our method. Section IV introduces our methodology and its structure, detailing a Multi-Layer Perceptron (MLP) module and Matrix Factorization (MF) module. It also includes a discussion of the designed surprise functions and other relevant components of our approach. Section V details the experiments conducted on several datasets using our methodology. Finally, Section VI presents our conclusions and future work.

2 Related Works

Here, we summarize widely used trajectory anomaly detection methods, both traditional and machine learning approaches in Section 2.1 followed by a introduction to the collaborative filtering method, covering its algorithm, applications, and recent developments in Section 2.2.

2.1 Trajectory Anomaly Detection

Over the past few decades, a broad spectrum of trajectory anomaly detection methods has been introduced. In this section, we systematically categorize these approaches into statistical, knowledge-driven, and machine learning approaches.

2.1.1 Statistical Approaches: Those methods are commonly used in conventional anomaly detection. These methods rely on distance or probability distributions, such as the Gaussian distribution, to identify data points that significantly deviate from the norm.

Knorr and Ng [25] pioneered the concept of anomaly detection algorithms. In their distance-based approach, a data point is identified as an anomaly if fewer than k other points are within a specified distance (δ) from it in the dataset. Liu et al. [31] introduced the Isolation Forest algorithm(IForest), which detects anomalies using binary trees. This method operates on the principle that anomalous are a minority and have attributes that differ markedly from normal instances. The algorithm isolates these anomalies by recursively partitioning the data into smaller subsets. If a forest of random trees consistently produces shorter path lengths for specific data points, those points are identified as anomalies. Laxhammar et al. [28] conducted a comparative study of the Gaussian Mixture Model (GMM) and Kernel Density Estimator (KDE) for anomaly detection in maritime traffic. Their study found that while KDE excels at capturing the intricacies of normal traffic, especially along sea lanes, both GMM and KDE have their limitations in detecting anomalies. Ristic et al.[42] proposed a method to extract motion patterns from historical AIS data, which are then used to build anomaly detectors based on adaptive kernel density estimation (AKDE). This method supports real-time anomaly detection by sequentially applying the detector to incoming AIS data. Wang et al. [52] proposed a trajectory-based anomaly detection algorithm

UserId	x_i (Latitude)	y_i (Longitude)	tc_i (CheckinTime)	tl_i (LeavingTime)	s_i (VenueType)
153	39.935892	116.453081	2009-06-21T03:03:01	2009-06-21T03:38:57	Workplace
153	39.998524	116.387211	2009-06-21T03:38:59	2009-06-21T04:08:00	Restaurant
153	39.991694	116.389809	2009-06-21T04:27:56	2009-06-21T05:00:27	Recreational
153	39.991424	116.384395	2009-06-21T05:24:01	2009-06-21T05:53:31	Recreational
153	39.995926	116.390297	2009-06-21T06:12:41	2009-06-21T06:52:45	Apartment

Table 1: A template of human spatial-temporal trajectory datasets utilized in this paper.

using multidimensional Hidden Markov Models (HMMs) for model training and likelihood estimation. Their approach allows for real-time detection of anomalous from sensor data, effectively identifying long-term deviations from normal behavior patterns. Piciarell et al. [40] developed a system for detecting anomalous events using trajectory analysis, particularly for video and traffic surveillance. Their system clusters anomalous trajectories with support vector machines (SVMs). Thang & Kim [50] proposed an improved DB-SCAN algorithm, known as DBScan-MP, where each cluster may have different epsilon and minpts values, enhancing its adaptability and performance in identifying clusters and anomalous.

2.1.2 Knowledge-Driven Approaches. Knowledge-driven approaches leverage expert knowledge—such as traffic rules, navigational experience, and established physical models—to assess whether a ship, vehicle, or individual is exhibiting anomalous behavior [12]. These approaches are particularly effective in domains like anomalous ship behavior detection (DBAS) and vehicle-to-vehicle (V2V) obstacle detection.

Avram et al. [5] introduced a method that combines expert knowledge with ship behavior data to develop a ship anomaly detector. This integrated approach optimizes purely data-driven models by refining their structure and making secondary corrections to their outputs. Wu et al. [55] developed the DB-TOD model to detect anomalous in vehicle trajectories. By capturing driving behavior and preferences from historical data, this probabilistic model calculates the latent costs of routing decisions using feature counts and latent biases, effectively identifying anomalous in both full and partial trajectories. Zhu et al. [67] proposed the Time-Dependent Popular Route (TPRO) algorithm, which evaluates trajectory anomalies by considering spatial and temporal characteristics. The TPRO algorithm accounts for the varying popularity of routes between locations, assigning different weights to each when calculating anomaly scores. Yuan et al. [57] developed the Trajectory Outlier Detection based on Structural Similarity (TOD-SS) algorithm. This method segments trajectories based on angles at each point and calculates distances between sub-trajectories using features such as direction, speed, angle, and location. Yu et al. [56] presented the Trajectory Outlier Detection using Common Slice Subsequence (TODCSS) algorithm, which improves accuracy by combining features like direction, position, and continuity to detect anomalies across consecutive anomalous segments. Kinematics extraction, as discussed in studies like [14, 29, 33], Those methods integrates kinematic principles to produce precise and physically realistic trajectory predictions.

2.1.3 Machine Learning Approaches. Recent advancements in machine learning have introduced promising solutions for anomaly detection, utilizing sophisticated neural network architectures to improve the identification of anomalies. These developments offer enhanced capabilities for detecting complex patterns and deviations that traditional methods might miss. Given the challenges of trajectory data sparsity and anomaly characteristics, Graph Neural Networks (GNNs) are increasingly employed in trajectory anomaly detection due to their ability to work with graph-structured data and ability to capture complex spatial patterns [44].

Recently, there has been a significant advancement in GNN: Goodge et al. [21] proposed LUNAR which utilizes information from a node's nearest neighbors to identify anomalies. Zhou et al.[66] introduced AST-GNN, an attention-based spatio-temporal graph neural network that enhances pedestrian trajectory prediction by effectively capturing complex interactions and unique motion patterns. Zheng et al.[60] introduced AddGraph, which combines an extended temporal GCN with an attention-based GRU to detect anomalies in dynamic graphs. It captures node and edge information, updating node embeddings and hidden states at each time step. Anomaly scores are assigned to edges based on the associated nodes.

Another category of deep learning models, aside from GNNs, includes various neural network approaches that treat the trajectory as a series to learn complex patterns and identify anomalies known as series learning models. These methods, including Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks, are particularly effective for analyzing sequential or temporal data.

For example: Niu et al. [36] introduce an adaptive LSTM that utilizes temporal continuity in sequential data. By incorporating a mask gate and maintaining span, this LSTM can update memory adaptively based on changes in the sequence input at each time step. Autoencoder methods have also been effectively applied to trajectory anomaly detection. Liu et al. [32] proposed a semi-supervised approach based on a Variational Autoencoder (VAE) called GM-VSAE, which models the probability distribution of route patterns in the latent space using a Gaussian mixture distribution. This method allows for the identification of anomalous by assessing the likelihood of a trajectory being generated from normal patterns. Similarly, Han et al. [22] introduced DeepTEA, a semi-supervised method designed to detect time-dependent trajectory anomalous. DeepTEA employs a Convolutional Neural Network (CNN) to model road traffic conditions and utilizes a Gaussian Mixture VAE to capture latent trajectory patterns.

2.2 Collaborative Filtering

Collaborative Filtering (CF), first introduced in the 1990s [20], is a widely used technique in recommendation systems. Originally, CF involved users collaborating to filter information by recording their reactions to documents they read, allowing others to access these reactions. As a popular recommendation model, collaborative

filtering utilizes ratings or other forms of user feedback to identify shared preferences among users. It then provides personalized recommendations based on these similarities, without requiring external information about the items or users. Collaborative Filtering employs two main techniques: the neighborhood approach and latent factor models. [27]. The neighborhood approach models a user's preference for an item based on their ratings of similar items, while latent factor models utilize a shared latent factor space between items and users to explain ratings. Nowadays, by leveraging advanced techniques such as matrix factorization [10], singular value decomposition [39] and integration of deep learning [9, 23, 54] have further enhanced the accuracy and scalability of collaborative filtering models.

3 Preliminaries

In this section, we present the key concepts and definitions essential to our study on trajectory anomaly detection used in the following sections.

3.1 Human Semantic Trajectory Representation

A semantic trajectory is represented as a sequence of staypoints (or "check-ins") $T = \{p_1 \rightarrow p_2 \rightarrow \cdots \rightarrow p_n\}$ where n is the number of staypoint observations in the trajectory. Each staypoint observation p_i in this sequence is a six-tuple $(userID, x_i, y_i, tc_i, tl_i, s_i)$, $userID$ is a unique identifier of an individual user, x_i is the longitude, y_i is the latitude, tc_i is the "check-in" timestamp when the trajectory T reaches point $p_i(x_i, y_i)$, tl_i indicates the time the user leaves that point, and s_i provides semantic information about the visited location such as the type of PoI. Table 1 provides an overview of the spatial-temporal trajectory for User 153. The first line of this example shows that this user stayed at a Workplace with the given coordinates from 03:03:01am (GMT) to 03:38:59am (GMT) and went to a restaurant shortly after.

The collection of all users' trajectories can be denoted as $\mathcal{T} = \{T_1, T_2, \ldots, T_m\}$, where m is the number of trajectories or users.

Given the above definitions, in the next subsection, we formally formulate the trajectory anomaly detection problems.

3.2 Human Spatial-temporal Trajectory Anomaly Detection

We assume the availability of (1) a training period during which we assume that users exhibit normal behavior and (2) a test period during which some users may exhibit anomalous behavior. The goal of trajectory anomaly detection is to identify the set of anomalous users during the test period using models of normal behavior learned using the train period. Formally,

Definition 3.1 (Train and Test Period). Let $\mathcal{T} = \{T_1, T_2, \ldots, T_m\}$ be a set of trajectories of m users. We assume that any staypoint that started before a specific staypoint in time t_{split} exhibits normal behavior and anomalous behavior may occur after t_{split}. Then, we can define a training trajectory:

$$\mathcal{T}_i^{Train} = \{(userID, x_i, y_i, tc_i, tl_i, t_i) \in T_i | tc_i <= t_{split}\},$$

and a testing trajectory:

$$\mathcal{T}_i^{Test} = \{(userID, x_i, y_i, tc_i, tl_i, t_i) \in T_i | tc_i > t_{split}\}.$$

This allows us to define a train dataset as

$$\mathcal{T}^{Train} = [T_1, ..., T_m],$$

and a test dataset as

$$\mathcal{T}^{Test} = [T_1, ..., T_m].$$

The goal of trajectory anomaly detection is to detect trajectories $T_i^{Test} \in T^{Test}$ what exhibit changes that deviate so much from their normal behavior that they rouse suspicion that they may have been generated by a different process. Intuitive, such different process may be different individuals (for example due to a stolen cellphone), a sudden change in behavior (such as exposure to an infectious disease), or a gradual change in behavior (such as the progress of a mental illness).

Definition 3.2 (Trajectory Anomaly Detection).
Let $\mathcal{T} = \{T_1, T_2, \ldots, T_m\}$ be a trajectory dataset split into T^{Train} and T^{Test}, and let $P \subseteq T^{Test}$ be a set a anomalous trajectories. The task of Trajectory Anomaly detection is to find P.

Given a training set of users $\mathcal{U}$ and their normal trajectory pairs $\mathcal{T}_u$, the objective is to identify anomaly users $U_{anomaly}$ in the test set who exhibit substantial deviations from the typical trajectory patterns established across the training population. These deviations are quantified using a score function f_{score}, which measures the extent of divergence from the common patterns observed within $\mathcal{U}$.

This goal presents several unique challenges:

(1) **Challenges in sparse data:** Accurately clustering users based on their patterns of life involves addressing complex issues related to both inter-user interactions and the broader context of their trajectories. One challenge as we mentioned before is the reliable formation of user clusters from datasets that are both sparse and consist of only positive pairs. The challenge lies in ensuring that the model consistently generates accurate user pattern clusters from a dataset that is only include positive pairs and highly sparse. Many POIs are visited by only a few users, which exacerbates the difficulty of achieving comprehensive clustering. To address this, the clustering algorithm must effectively identify and interpret meaningful interactions among users despite the limited data coverage. Meanwhile, these methods must be robust enough to ensure prediction accuracy, even when data is unevenly distributed.

(2) **Challenges in quantifying anomalous:** Human behavior is inherently dynamic and can be influenced by various temporal and spatial factors. Traditional anomaly detection methods may be inadequate in this context, as they might misclassify new or infrequent POIs as anomalies. This limitation underscores the need for more sophisticated approaches to accurately quantify anomalous. Sophisticated approaches are required to accurately identify and score anomalous, effectively distinguishing between genuine anomalies and legitimate but less frequent behaviors.

4 Methodology

In this section, we introduce our neural collaborative filtering-based unsupervised anomaly detection approach for human trajectory analysis. While this section provides ideas, concepts, and theory, the interested reader may find implementation details in our GitHub
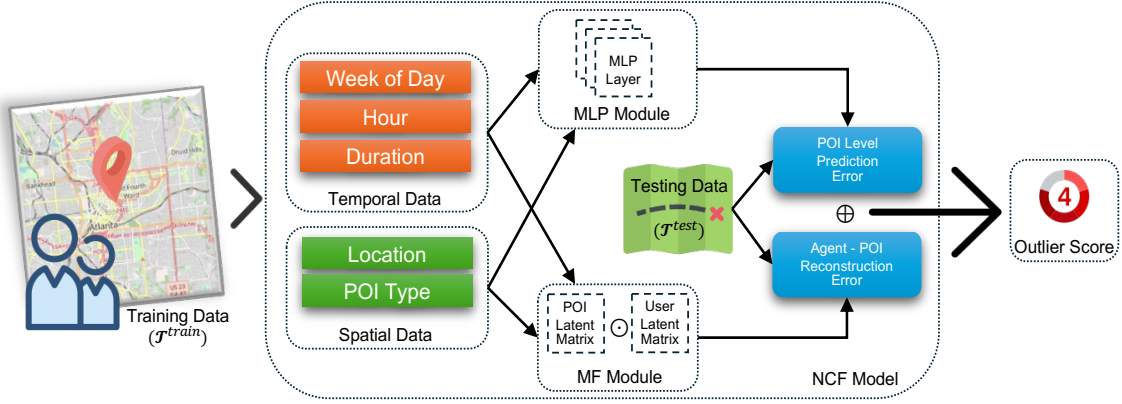
Figure 5: The illustration of our proposed NCF model. (Left) Extraction of spatio-temporal embeddings from input trajectories; (Middle) Neural Collaborative Filtering with two-level of prediction; (Right) Anomaly scores based on weighted prediction.

Repository at https://github.com/alex-cse/NCF_AHSTD. Notably, our method can identify anomalous users without any explicit semantic information of places. Instead, our approach extracts latent semantic information using collaborative filtering (based on users visiting similar places and places visited by similar users). Our framework comprises two modules:

(1) The Collaborative Filtering Module uses collaborative filtering to model the User-POI visit matrix observed in the train period. It identifies anomalous users by measuring surprise, that is by detecting deviations from typical (model) trajectory patterns observed in the test period.

(2) Neural Module: This module take the advantage of neural networks to interpret the complex spatial-temporal relationships within the data. Traditional methods like collaborative filtering often fall short when dealing with those spatial-temporal data because they may not fully capture non-linear patterns or subtle dependencies within the dataset.

To effectively quantify anomalous users, we employ an accumulator based on a continuous-state Markov chain to calculate anomaly scores. This approach is grounded in the principle that each user's trajectory is conceptualized as a sequence where each event is influenced by the preceding one, but making the simplifying (Markov-)assumption that the future is conditionally independent of the past given the present. We assume that when a trajectory exhibits anomalous behavior, this behavior not only affects a single staypoint, but affects a chain of subsequent events. For instance, this might manifest as an increase in travel distance or as visits to a POI during an atypical time frame. To capture these anomalies, the accumulator evaluates each POI for every user based on the trajectory sequence. By doing so, our method can detect not only extreme anomalous but also users with moderate deviations from the expected patterns in their overall trajectory.

4.1 Neural Collaborative Filtering Model

To thoroughly explore and understand the spatio-temporal interactions between users and POIs, we propose a streamlined neural collaborative filtering approach [23, 41]. This model leverages a Multi-Layer Perceptron (MLP) module to effectively capture and learn both the temporal and spatial mobility of users, as well as the spatio-temporal relationships among various locations. By integrating the MLP module with a matrix factorization framework,

we enable a comprehensive analysis of users' behaviors over time and the intricate spatial interactions between different POIs.

Our proposed architecture is illustrated in Figure 5. Specifically, within the trajectory pairs denoted as $\mathcal{T}$, we analyze the interactions between users u and their respective POI p_i, forming trajectory pairs $\mathcal{T}_{ui}$. In the context of $\mathcal{T}_{ui}$, we consider several key parameters.

- $h \in [0, 23]$ is the hour of check-in, providing a temporal resolution on an hourly basis.
- $d \in [0, 6]$ is the day of the week when the check-in occurs.
- The geographical location l specifies the geographical location of the check-in, which is critical for spatial analysis.

Additionally, we preprocess the data within each trajectory pair $\mathcal{T}_{ui}$ make it encompassed a set of movement attributes, including stay duration, and distance traveled. The stay duration indicates the time spent at a particular location, while movement speed and distance provide insights into the mobility patterns and travel behaviors of the user. The final tuples $\mathbf{Z}$ of user i and POI j pair can be represent as:

$$\mathbf{Z_{i,j}} = \begin{bmatrix} u_i \\ h_j \\ d_j \\ l_j \end{bmatrix} \tag{1}$$

In this context, h_j, d_j, and l_j denote the embeddings corresponding to the hour of the user's arrival at the POI, the week of of day the user's visit the POI, and the movement distance from previous check-in POI (calculate base on latitude and longitude) of the POI, respectively. After that we applied a three-layer MLP module to learn the interactions of latent features, the equation can represented as:

$$\phi^{MLP} = a_3 \left(W_3^T \left(a_2 \left(W_2^T \left(a_1 \left(W_1^T \mathbf{Z} + \mathbf{b}_1 \right) \right) + \mathbf{b}_2 \right) \right) + \mathbf{b}_3 \right) \tag{2}$$

Here, $\mathbf{Z}$ is the input latent feature vector, W_i and $\mathbf{b}_i$ are the weight matrices and bias vectors of each layer, and a_i represents the activation functions applied at each layer. For the Generalized Matrix Factorization component, we also utilize embeddings derived from the vectors $\mathbf{Z}$, incorporating additional information about the POI type t. Given that users u_i have features like hour h_i, week of day w_i, and distance d_i, and POI p_j have features like type t_j, the equation for matrix factorization incorporating these auxiliary features

can be written as:

$$\phi_{ij}^{MF} \approx u_i^T p_j + (w_1 \cdot h_i + w_2 \cdot d_i + w_3 \cdot h_i) + (w_4 \cdot t_j) \qquad (3)$$

Where the w_1, w_2, w_3, w_4 are weights for the corresponding auxiliary features. This equation combines the latent factor model with additional terms that account for the auxiliary features, allowing us to capture both latent features (inferred from similar users visiting the same POI) and explicit features (such as known types of POIs like restaurants or schools).

In our Spatial-Temporal model, we extend beyond the conventional method by integrating not only user and POIs latent variables but also additional latent features specifically designed to capture spatial and temporal dynamics. This includes geographic information that captures the spatial context of the interactions, as well as temporal attributes that detail the duration of stays and specific visit times. The geographic information provides a framework for understanding the spatial relationships and locations of the stay-points of interest, while the temporal attributes offer insights into the timing and frequency of visits. After the initial embedding layer, the model utilizes multiple fully connected layers, which function as neural collaborative filtering layers. The overall MLP module was trained on the training dataset $\mathcal{T}^{Train}$, with the objective of capturing the representation of normal patterns. Once trained, any deviations from these established patterns are indicative of anomalies in the test dataset $\mathcal{T}^{Test}$. In the following section, we describe how to measure these deviations and to identify anomalous users.

4.2 Quantification of Anomaly Scores

We propose two measures to quantify the degree to which an individual is anomalous: The first measure directly considers the difference (the surprise) between the expected (by the NMF model) score ϕ_{ij}^{MF} of an individual i visit a place j (described in the previous section) and the observed visits $\mathcal{T}^{Test}$. While some places are visited by a large number of individuals every day, other places may have very few or no daily visits. For such places, it becomes difficult to predict which individual may visit them, and, consequently, a (normal) visit may incur a large surprise to the model. To avoid incurring such false positives, we additionally propose not only to predict (and measure surprise) for individual POIs, but also for the type of POI visited. As an intuitive example, it may surprise the model that individual A visits, during lunch time, a new German restaurant that none have visited during the training period (because the restaurant may not have opened yet). But it is not surprising that A visits a restaurant during lunchtime as they always do. This section describes details for both these measures.

4.2.1 User-POI Matrix Reconstruction Surprise: This surprise function assesses the discrepancy between the User-POI visit status matrix in the test set and its training set Matrix Factorization reconstruction. The User-POI visit status matrix is a binary matrix where each entry signifies whether a specific user has visited a particular POI. A high difference in the User-POI Reconstruction Error indicates that the reconstructed matrix significantly deviates from the actual visit status matrix, suggesting that the user's behavior is inconsistent with the expected patterns and should be considered anomalous. In the training matrix, each observed User-POI pair is given a value of 1 if the corresponding user visited the corresponding POI, and 0 otherwise. The factorized model aims

at reconstructing this matrix using compressed information, thus assigning a visit score to each User-POI pair. We note that these scores may be less than 0 or greater than 1 and thus, should not be interpreted as probabilities.[1]

Applying a Generalized Matrix Factorization provides a matrix ϕ^{MF} with dimensions $m \times n$, where n represents the number of individuals and m denotes the number of POIs. We propose to simply use the absolute difference between this prediction and the observed visits as follows:

$$\text{surprise}_{ij} = \left| \phi_{ij}^{MF} - \mathcal{T}_{ij}^{Test} \right| \qquad (4)$$

This method ensures that user-POI visit will receive a low surprise if the user visits the POI in the test period but the POI is considered a typical location based on the collaborative filtering. Conversely, an anomalous POI, which deviates from the expected pattern, will be assigned a higher surprise. We note that this surprise function also gives a high surprise to a place that is NOT visited in the test period but is expected to be visited by the collaborative filtering model.

If we have additional information about the semantics of the anomalies that were expecting to find, we can adopt the above surprise function accordingly. For example, if we know that anomalous agents visit places they would not normally visit (such as going to random places), we can reflect this apriori knowledge in the surprise by only counting surprise for POIs visited in Test:

$$\text{surprise}^{NewPOI} = \min(0, \mathcal{T}_{ij}^{Test} - \phi_{ij}^{MF}) \qquad (5)$$

Analogously, if we know that anomalous agents stop visiting places they would normally visit (such as skipping going to work), we can adopt the surprise function to only consider POIs that were not visited in Test:

$$\text{surprise}^{MissingPOI} = \min(0, \phi_{ij}^{MF} - \mathcal{T}_{ij}^{Test}) \qquad (6)$$

If we have no apriori knowledge about how anomalies manifest, we can use the surprise function surprise_{ij} in Equation 4.

The surprise functions defined in Equations 4-6 provide us with anomaly scores for individual individual-POI pairs. To obtain anomaly scores at the individual user level, we aggregate these scores by individual. The accumulated error for user i can be computed as:

$$\text{surprise}_i = \sum_{j=1}^{m} \text{surprise}_{ij} \qquad (7)$$

The implicit assumption that we make here is that an anomalous user may have more than a single anomalous visit. Depending on apriori information of what may constitute anomalous behavior of an individual user, we can adapt this surprise function accordingly. For instance, if we know that an anomalous user will visit exactly one anomalous place, then we can adjust this user-level surprise function by computing the maximum surprise over all POIs of the same user:

$$\text{surprise}_i^{singlePOI} = \max_{j=1}^{m} \text{surprise}_{ij} \qquad (8)$$

---

[1]Although we note that using a logistic model that maps scores to probabilities in the interval [0,1] may be an interesting extension of the model.

4.2.2 POI Type Surprise. In addition to measuring surprise at individual POI level, we also propose to measure surprise the the POI type level. The surprise function at the POI type level quantifies the divergence between the model's predicted POI type and the actual POI type. This approach is specifically designed to address the issue of data sparsity: Even though individual POIs may not have enough visits to allow the collaborative filtering to model the latent features of the POI.

Given a set $C = \{c_1, ..., c_{|C|}\}$ of types (classes) of POIs, and letting $c(p_j)$ be the class of POI p_j, then the probability of class $k \in C$ visited by user u_i can be calculated as:

$$p_{ik} = \frac{\sum_{j=1}^{m} \mathcal{T}_{ij}^{train} \cdot \mathbb{I}(c(p_j) = k)}{\sum_{j=1}^{m} \mathcal{T}_{ij}^{train}} \tag{9}$$

where m is the number of POIs and $\mathbb{I}(c(p_j) = k)$ is an indicator function that returns 1 if POI p_j has type k and zero otherwise.

Then, for each user we can define their most likely POI type, as:

$$\text{mostLikelyPOI}_i = \arg\max k \in C(p_{ik})$$

To measure surprise of a user u_i, we count the number of visits observed during the test period that do not match the user's most likely POI:

$$\text{surprise}_i^{\text{POIType}} = \sum_{j=1}^{m} \mathbb{I}(\text{mostLikelyPOI}_i \neq \mathcal{T}_{Test}(i, j)) \cdot \mathcal{T}_{Test}(i, j)$$

Our final anomaly score of each user is then defined as the sum of the two surprise scores:

$$\text{AnomalyScore}_i = \text{surprise}_i + \text{surprise}_i^{\text{POIType}} \tag{10}$$

where surprise_i is the User-POI Matrix Surprise described in Section 4.2.1 and $\text{surprise}_i^{\text{POIType}}$ is the POI Type Surprise described in Section 4.2.2.

5 Experimental Results and Environmental setup

In this section, we conduct a comprehensive evaluation of our neural collaborative filtering model using multiple simulated and real-world datasets. The subsequent parts of this section will provide detailed information about each dataset, including their anomaly characteristics and the specific preprocessing steps applied. Additionally, we will outline the anomalous settings and configurations in the simulated datasets used for our model. resent a thorough evaluation of our model's performance.

5.1 Experimental Datasets

To evaluate the performance of our proposed model, we utilized simulated and real-world dataset. The simulated datasets were generated using the Agent-Based Patterns-of-Life Simulation (POL) [24, 68], while the real-world dataset was based on the GeoLife dataset [58, 59, 64].

5.1.1 Agent-Based Simulation of Patterns of Life. The Patterns of Life simulation is an existing simulation that was designed to emulate human needs and behaviors in an urban environment [3, 26, 68].

Within the simulation, agents engage in activities that mimics real-life actions, such as attending work, forming friendships, and participating in social gatherings. These agents navigate a virtual environment modeled on real-world settings like roads and buildings, sourced from OpenStreetMap.

In our study, we test our model on four distinct map setting: Fairfax County, Virginia (FVA); the French Quarter of New Orleans, Louisiana (NOLA); Atlanta, Georgia (ATL); and Beijing, China (BJNG). These dataset contains 450 days of normal life, followed by an additional 14 days with introduce anomalous behaviors into the regular patterns which serve as the training and test dateset. In the 14-day test dataset, we identified three distinct types of anomalous behaviors: hunger anomalous, social anomalous, and work anomalous. Those anomalous behaviors are defined as follows:

- Hunger anomalous: An agent under this category becomes hungry more quickly. Such agents have to go to restaurants or their homes much more often to satisfy their food needs.
- Social anomalous: This type of agent randomly selects recreational sites to meet and spend leisure time, rather than being guided by their attributes and social network.
- Work anomalous: Agents in this category do not go to work when they normally should be working.

These anomalies were further classified into three intensity levels: red (high intensity), orange (mid intensity), and yellow (low intensity). The detailed intensity levels are defined as follows:

- Yellow Anomalies: The agent displays anomalous behavior at a low intensity. For example, a hunger anomaly with a yellow intensity level experiences hunger more frequently than usual but does not need to visit a restaurant or home as often as a red intensity anomaly. For work and social anomalous, the yellow intensity level indicates that the agents will behave anomalously 20% of the time.
- Orange Anomalies: The agent displays anomalous behavior at a moderate intensity. For example, a hunger anomaly with an orange intensity level feels hungry more frequently than normal and needs to visit a restaurant or home more often than a yellow intensity anomaly, but less often than a red intensity anomaly. For work and social anomalous, the orange intensity level indicates that the agents will behave anomalously 50% of the time.
- Red Anomalies: The agent displays anomalous behavior at the highest intensity. For example, a hunger anomaly with a red intensity level experiences hunger more intensely than other anomalous and needs to visit a restaurant or home more frequently. For work and social anomalous, the red intensity level indicates that the agents will behave anomalously 100% of the time.

5.1.2 Real World Dataset. The real-world dataset for this study was derived from the Microsoft Research Asia's GPS Trajectory dataset [64]. We employed an stay-point extraction algorithm [63] to transform the data into a check-in pattern suitable for our studies. Subsequently, we utilized OpenStreetMap to categorize locations into four groups: apartments, workplaces, recreational, and restaurants. Given the extensive range of categories in OpenStreetMap, we manually classified them into these distinct groups. Following preprocessing, agents with fewer than 50 records were excluded, yielding a final dataset consisting of 69 agents. This refined dataset

| | ATL | | | | NOLA | | | |
Model	Top-10 Hits	Top-100 Hits	Top-150 Hits	AUC score	Top-10 Hits	Top-100 Hits	Top-150 Hits	AUC score
OMPAD	0	4	5	0.4782	1	1	1	0.4509
DSVDD	**10**	<u>28</u>	<u>29</u>	<u>0.6171</u>	**9**	17	<u>26</u>	0.6098
DAE	1	6	7	0.4759	0	3	6	0.4937
ECOD	3	12	15	0.5357	4	<u>18</u>	23	<u>0.6274</u>
IForest	2	6	11	0.5136	0	5	13	0.5247
NCF	<u>9</u>	**35**	**43**	**0.6853**	<u>7</u>	**44**	**58**	**0.7760**
	FVA				BJNG			
OMPAD	1	1	1	0.4796	1	5	6	0.4370
DSVDD	<u>5</u>	<u>19</u>	<u>24</u>	<u>0.6329</u>	**10**	<u>29</u>	<u>33</u>	<u>0.5643</u>
DAE	3	12	17	0.5404	1	7	14	0.5474
ECOD	1	12	14	0.5318	1	17	23	0.5778
IForest	2	7	13	0.5616	0	7	13	0.5274
NCF	**7**	**33**	**41**	**0.6673**	<u>9</u>	**42**	**48**	**0.7209**
	GeoLife							
OMPAD	0	0	1	0.0552				
DSVDD	**4**	**9**	**16**	**0.8714**				
DAE	**4**	4	7	0.4422				
ECOD	**4**	<u>6</u>	<u>11</u>	<u>0.6977</u>				
IForest	<u>3</u>	5	<u>11</u>	0.6531				
NCF	<u>3</u>	<u>6</u>	7	0.6236				

Table 2: Anomaly detection performance. For each dataset, the highest AUC scores are highlighted in bold, and the second-highest scores are underlined. We report Top-5/10/25 Hits for GeoLife dataset due to its size constraint.

includes 14,080 training trajectories and 3,552 test trajectories, spanning a period of over four years. Within this dataset, we introduced a specific type of anomaly known as the "imposter anomaly," where an agent switches trajectories with another agent after a certain point in time. The dataset was divided into two segments: 80% for training, using the stay points, and the remaining 20% for testing, where anomalous were introduced.

5.2 Experimental Settings

In this section, we introduce our competitor methods and evaluation metrics.

5.2.1 Competitor Methods: We compare with several trajectory anomaly detection methods, including non-deep learning methods and learning methods:

OMPAD [6] is an anomaly detection method that analyzes objects' movement patterns by counting the types of locations they visit. It identifies abnormal activities by measuring the deviations in moving trends compared to established normal patterns.

DSVDD [43] is a deep one-class classification based anomaly detection method. We generalize it to handle the task of semantic trajectory anomaly detection in a most intuitive way. We map the weekly trajectories of each user to a high dimensional sphere by a deep neural network encoder. Then the distance of trajectories from the sphere's surface is quantified as an anomaly score.

DAE [17, 65] is a widely-used anomaly detection method that leverages a deep autoencoder. Utilizing an encoder-decoder model architecture, it reconstructs input trajectories, and the resulting reconstruction error is used as an anomaly indicator, signifying deviations from the normal pattern.

ECOD [30] is an unsupervised anomaly detection algorithm that uses empirical cumulative distribution functions to identify anomalous without requiring parameter tuning, offering both efficiency and interpretability.

IForest [31] is an algorithm detects anomalies by recursively partitioning data, isolating anomalous based on shorter path lengths in a forest of random trees.

5.2.2 Evaluation Metrics. To rigorously evaluate the performance of our anomaly detection methodology, we utilize the Top-K hits metric. In our approach, users with the K highest anomaly scores are designated as anomalous. The number of hits serves as an indicator of the method's efficacy in accurately distinguishing anomalous. The Top-K hits of the model refer to the true positives, representing the anomaly users correctly identified within the top K ranked predictions made by the model. For our evaluation, we specifically use Top-10, Top-100, and Top-150 hits. However, for the GeoLife dataset, we report Top-5/10/25 hits instead of Top-100 due to anomaly number constraints. Furthermore, we employ the area under the receiver operating characteristic curve (AUC) as an additional evaluation metric.

5.3 Anomaly Detection Results

Detection Results: Table 2 presents the anomaly detection performance of our proposed method alongside several competitive methods. In the table, the highest AUC scores for each dataset are highlighted in bold, while the second-highest scores are underlined. The results show for the four simulated datasets, our NCF model always achieves the highest AUC scores by a substantial margin compared to other machine learning and non-machine learning

Category	ATL			NOLA			FVA			BJNG		
	Red	Orange	Yellow	Red	Orange	Yellow	Red	Orange	Yellow	Red	Orange	Yellow
Hunger	0.30	0.23	0.30	0.40	0.53	0.33	0.36	0.30	0.30	0.40	0.43	0.43
Work	0.40	0.30	0.10	0.70	0.40	0.30	0.50	0.20	0.10	0.50	0	0.20
Social	0.60	0.40	0	0.50	0.10	0	0.30	0.10	0	0.30	0	0

Table 3: Recall of each category in the top 150 predictions of each dataset. (Recall@150)
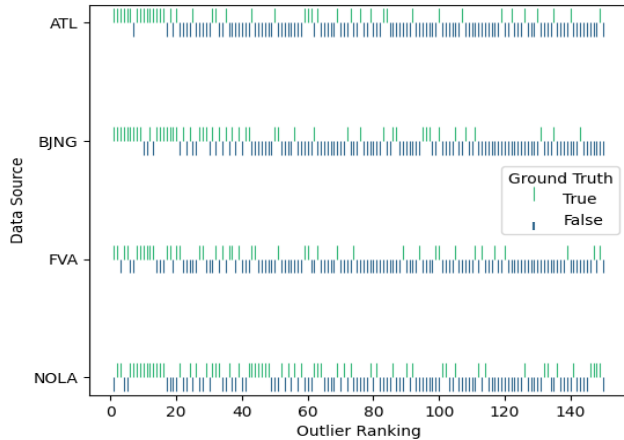

Figure 6: The anomaly ranking among four datasets

methods. However, for the GeoLife real-world dataset, our proposed method is significantly outperformed by state-of-the-art methods. We explain this by the extreme sparsity of the GeoLife dataset. This dataset captures only 182 across all of Beijing, China. Having so few users, there are very few cases where the same place is visited by more than one unique user. Thus, the resulting User-POI matrix only has one non-zero value in most columns. This makes it impossible for the matrix factorization to successfully apply any collaborative filtering. Thus, Table 2 shows promising results in the case where we have more dense data (for all users and all places). But for extremely small datasets like GeoLife, collaborative filtering is not able to infer any common topics and features among the users and places due to a lack of examples where different users visit the same place.

To give more insights on the quality of the anomaly scores returned by NCF, beyond a single AUC-PR score, Figure 6 shows how true positives and true negatives are distributed across the anomaly score ranking. We observe a clear correlation that true positives are frequently among the highest ranked anomaly scores.

Detailed Detection Count for Types of anomalous: To evaluate the detection performance for each anomaly category Table 3 provides the true positive detection count per category in the top 150 predictions of each dataset. This result indicates that, in most cases, our model consistently achieves high detection accuracy for anomalous with the highest intensity, particularly demonstrating heightened sensitivity to categories involving significant spatial changes. This observation indicates that our method is particularly sensitive to variations in spatial patterns but less responsive to temporal patterns.

6 Conclusions and Future Work

In conclusion, this paper presents an unsupervised method for detecting anomalous in human trajectories using a neural collaborative filtering model applied to user-level spatiotemporal datasets.

Our approach effectively addresses challenges such as data sparsity and imbalance by leveraging the strengths of the collaborative filtering module and neural network components, which significantly mitigate these issues. Comprehensive experiments on our five datasets validate the effectiveness and efficiency of the proposed method, which shows substantial improvements in accurately detecting anomalous within complex spatiotemporal datasets. For future research, we aim to explore the impact of geolocation embedding factors on model performance across various types of anomalies. Additionally, we will investigate incorporating more noise data to enhance the robustness of anomaly detection in real-world scenarios.

7 Acknowledgements

References

[1] Charu C Aggarwal and Charu C Aggarwal. 2017. *An introduction to outlier analysis*. Springer.
[2] Hossein Amiri, Will Kohn, et al. 2024. The Patterns of Life Human Mobility Simulation. (2024). arXiv:2410.00185
[3] Hossein Amiri, Ruochen Kong, and Andreas Zufle. 2024. Urban Anomalies: A Simulated Human Mobility Dataset with Injected Anomalies. (2024). arXiv:2410.01844
[4] Hossein Amiri, Shiyang Ruan, et al. 2023. Massive Trajectory Data Based on Patterns of Life. In *SIGSPATIAL'23*. ACM, 1–4.
[5] Vladimir Avram, Uwe Glässer, and Hamed Yaghoubi Shahir. 2012. Anomaly detection in spatiotemporal data in the maritime domain. In *2012 IEEE International Conference on Intelligence and Security Informatics*. IEEE, 147–149.
[6] Arslan Basharat, Alexei Gritai, and Mubarak Shah. 2008. Learning object motion patterns for anomaly detection and improved object detection. In *2008 IEEE conference on computer vision and pattern recognition*. IEEE, 1–8.
[7] Asma Belhadi, Youcef Djenouri, Jerry Chun-Wei Lin, and Alberto Cano. 2020. Trajectory outlier detection: Algorithms, taxonomies, evaluation, and open challenges. *ACM Transactions on Management Information Systems (TMIS)* 11, 3 (2020), 1–29.
[8] James Bennett, Stan Lanning, et al. 2007. The netflix prize. In *Proceedings of KDD cup and workshop*, Vol. 2007. New York, 35.
[9] Jesus Bobadilla, Santiago Alonso, and Antonio Hernando. 2020. Deep learning architecture for collaborative filtering recommender systems. *Applied Sciences* 10, 7 (2020), 2441.
[10] Dheeraj Bokde, Sheetal Girase, and Debajyoti Mukhopadhyay. 2015. Matrix factorization model in collaborative filtering algorithms: A survey. *Procedia Computer Science* 49 (2015), 136–146.
[11] Francesco Bonchi, Laks VS Lakshmanan, and Hui Wang. 2011. Trajectory anonymity in publishing personal mobility data. *ACM Sigkdd Explorations Newsletter* 13, 1 (2011), 30–42.
[12] Nicolas Brax, Eric Andonoff, and Marie-Pierre Gleizes. 2012. A self-adaptive multi-agent system for abnormal behavior detection in maritime surveillance. In *Agent and Multi-Agent Systems. Technologies and Applications: 6th KES International Conference, KES-AMSTA 2012, Dubrovnik, Croatia, June 25-27, 2012. Proceedings 6*. Springer, 174–185.
[13] Khac-Hoai Nam Bui, Jiho Cho, and Hongsuk Yi. 2022. Spatial-temporal graph neural network for traffic forecasting: An overview and open research issues.

Applied Intelligence 52, 3 (2022), 2763–2774.

[14] Henggang Cui, Thi Nguyen, et al. 2020. Deep Kinematic Models for Kinematically Feasible Vehicle Trajectory Predictions. In *IEEE ICRA'20*. 10563–10569. https://doi.org/10.1109/ICRA40945.2020.9197560

[15] Yves-Alexandre De Montjoye et al. 2013. Unique in the crowd: The privacy bounds of human mobility. *Scientific reports* 3, 1 (2013), 1–5.

[16] Leonardo D'Errico, Fabio Franchi, et al. 2017. Design and implementation of a children safety system based on IoT technologies. In *2017 2nd International Multidisciplinary Conference on Computer and Energy Science (SpliTech)*. IEEE, 1–6.

[17] Dario Dotti, Mirela Popa, and Stylianos Asteriadis. 2020. A hierarchical autoencoder learning model for path prediction and abnormality detection. *Pattern Recognition Letters* 130 (2020), 216–224.

[18] Minxuan Duan, Yinlong Qian, et al. 2024. Back to Bayesics: Uncovering Human Mobility Distributions and Anomalies with an Integrated Statistical and Neural Framework. (2024). arXiv:2410.01011 https://arxiv.org/abs/2410.01011

[19] Qiang Gao, Fan Zhou, et al. 2017. Identifying Human Mobility via Trajectory Embeddings.. In *IJCAI*, Vol. 17. 1689–1695.

[20] David Goldberg, David Nichols, et al. 1992. Using collaborative filtering to weave an information tapestry. *Commun. ACM* 35, 12 (1992), 61–70.

[21] Adam Goodge, Bryan Hooi, See-Kiong Ng, and Wee Siong Ng. 2022. Lunar: Unifying local outlier detection methods via graph neural networks. In *AAAI'22*, Vol. 36. 6737–6745.

[22] Xiaolin Han, Reynold Cheng, Chenhao Ma, and Tobias Grubenmann. 2022. DeepTEA: effective and efficient online time-dependent trajectory outlier detection. *Proc. VLDB Endow.* 15, 7 (mar 2022), 1493–1505. https://doi.org/10.14778/3523210.3523225

[23] Xiangnan He, Lizi Liao, et al. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*. 173–182.

[24] Joon-Seok Kim et al. 2020. Location-based social network data generation based on patterns of life. In *2020 21st IEEE International Conference on Mobile Data Management (MDM)*. IEEE, 158–167.

[25] Edwin M Knox and Raymond T Ng. 1998. Algorithms for mining distancebased outliers in large datasets. In *VLDB*. Citeseer, 392–403.

[26] Will Kohn, Hossein Amiri, and Andreas Züfle. 2023. EPIPOL: An Epidemiological Patterns of Life Simulation (Demonstration Paper). In *SIGSPATIAL SpatialEpi'23 Workshop*. ACM, 13–16.

[27] Yehuda Koren, Steffen Rendle, and Robert Bell. 2021. Advances in collaborative filtering. *Recommender systems handbook* (2021), 91–142.

[28] Rikard Laxhammar, Goran Falkman, and Egils Sviestins. 2009. Anomaly detection in sea traffic-a comparison of the gaussian mixture model and the kernel density estimator. In *2009 12th international conference on information fusion*. IEEE, 756–763.

[29] Xiaolei Li, Jiawei Han, and Sangkyum Kim. 2006. Motion-alert: automatic anomaly detection in massive moving objects. In *IEEE ISI 2006*. Springer, 166–177.

[30] Zheng Li, Yue Zhao, Xiyang Hu, Nicola Botta, Cezar Ionescu, and George H. Chen. 2023. ECOD: Unsupervised Outlier Detection Using Empirical Cumulative Distribution Functions. *IEEE Transactions on Knowledge and Data Engineering* 35, 12 (Dec. 2023), 12181–12193. https://doi.org/10.1109/tkde.2022.3159580

[31] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. 2008. Isolation forest. In *2008 eighth ieee international conference on data mining*. IEEE, 413–422.

[32] Yiding Liu, Kaiqi Zhao, Gao Cong, and Zhifeng Bao. 2020. Online Anomalous Trajectory Detection with Deep Generative Sequence Modeling. In *2020 IEEE 36th International Conference on Data Engineering (ICDE)*. 949–960. https://doi.org/10.1109/ICDE48307.2020.00087

[33] Tomás Machado, Rui Maia, et al. 2019. Vessel trajectories outliers. In *Ambient Intelligence–Software and Applications–, 9th International Symposium on Ambient Intelligence*. Springer, 247–255.

[34] Fanrong Meng, Guan Yuan, et al. 2019. An overview on trajectory outlier detection. *Artificial Intelligence Review* 52 (2019), 2437–2456.

[35] Sai Munikoti, Deepesh Agarwal, et al. 2023. Challenges and opportunities in deep reinforcement learning with graph neural networks: A comprehensive review of algorithms and applications. *IEEE transactions on neural networks and learning systems* (2023).

[36] Dejiao Niu, Zheng Xia, et al. 2018. ALSTM: adaptive LSTM for durative sequential data. In *IEEE ICTAI'18*. IEEE, 151–157.

[37] Richard S Ostfeld, Gregory E Glass, and Felicia Keesing. 2005. Spatial epidemiology: an emerging (or re-emerging) discipline. *Trends in ecology & evolution* 20, 6 (2005), 328–336.

[38] Christine Parent et al. 2013. Semantic trajectories modeling and analysis. *ACM Computing Surveys (CSUR)* 45, 4 (2013), 1–32.

[39] Arkadiusz Paterek. 2007. Improving regularized singular value decomposition for collaborative filtering. In *Proceedings of KDD cup and workshop*. 5–8.

[40] Claudio Piciarelli, Christian Micheloni, and Gian Luca Foresti. 2008. Trajectory-based anomalous event detection. *IEEE Transactions on Circuits and Systems for video Technology* 18, 11 (2008), 1544–1554.

[41] Steffen Rendle, Walid Krichene, et al. 2020. Neural collaborative filtering vs. matrix factorization revisited. In *Proceedings of the 14th ACM Conference on*

Recommender Systems. 240–248.

[42] Branko Ristic, Barbara La Scala, Mark Morelande, and Neil Gordon. 2008. Statistical analysis of motion patterns in AIS data: Anomaly detection and motion prediction. In *2008 11th International Conference on Information Fusion*. IEEE, 1–7.

[43] Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. 2018. Deep one-class classification. In *International conference on machine learning*. PMLR, 4393–4402.

[44] Zahraa Al Sahili and Mariette Awad. 2023. Spatio-temporal graph neural networks: A survey. *arXiv preprint arXiv:2301.10569* (2023).

[45] Erik Seglem, Andreas Züfle, et al. 2017. On privacy in spatio-temporal data: User identification using microblog data. In *SSTD'17*. Springer, 43–61.

[46] Chris Stanford, Suman Adari, et al. 2024. NUMOSIM: A Synthetic Mobility Dataset with Anomaly Detection Benchmarks. *arXiv preprint arXiv:2409.03024* (2024).

[47] Thanos G Stavropoulos et al. 2020. IoT wearable sensors and devices in elderly care: A literature review. *Sensors* 20, 10 (2020), 2826.

[48] Xiaoyuan Su. 2009. A Survey of Collaborative Filtering Techniques. (2009).

[49] Kaixin Sui, Youjian Zhao, et al. 2016. Your trajectory privacy can be breached even if you walk in groups. In *IEEE/ACM IWQoS'16*. IEEE, 1–6.

[50] Tran Manh Thang and Juntae Kim. 2011. The anomaly detection by using dbscan clustering with multiple parameters. In *2011 International Conference on Information Science and Applications*. IEEE, 1–5.

[51] Magdalena I Tolea, John C Morris, and James E Galvin. 2016. Trajectory of mobility decline by type of dementia. *Alzheimer Disease & Associated Disorders* 30, 1 (2016), 60–66.

[52] Chen Wang, Hongzhi Lin, and Hongbo Jiang. 2014. Trajectory-based multi-dimensional outlier detection in wireless sensor networks using Hidden Markov Models. *Wireless networks* 20 (2014), 2409–2418.

[53] Xiaoyang Wang, Yao Ma, et al. 2020. Traffic flow prediction via spatial temporal graph neural network. In *Proceedings of the web conference 2020*. 1082–1092.

[54] Jian Wei, Jianhua He, et al. 2017. Collaborative filtering and deep learning based recommendation system for cold start items. *Expert Systems with Applications* 69 (2017), 29–39.

[55] Hao Wu, Weiwei Sun, and Baihua Zheng. 2017. A Fast Trajectory Outlier Detection Approach via Driving Behavior Modeling. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management* (Singapore, Singapore) *(CIKM '17)*. Association for Computing Machinery, New York, NY, USA, 837–846. https://doi.org/10.1145/3132847.3132933

[56] Qingying Yu, Yonglong Luo, Chuanming Chen, and Xiaohan Wang. 2018. Trajectory outlier detection approach based on common slices sub-sequence. *Applied Intelligence* 48 (2018), 2661–2680.

[57] Guan Yuan, Shixiong Xia, Lei Zhang, Yong Zhou, and Cheng Ji. 2011. Trajectory outlier detection algorithm based on structural features. *Journal of Computational Information Systems* 7, 11 (2011), 4137–4144.

[58] Zheng Zhang, Hossein Amiri, et al. 2023. Large Language Models for Spatial Trajectory Patterns Mining. (2023). arXiv:2310.04942

[59] Zheng Zhang, Hossein Amiri, et al. 2024. Transferable Unsupervised Outlier Detection Framework for Human Semantic Trajectories. (2024). arXiv:2410.00054

[60] Li Zheng, Zhenpeng Li, et al. 2019. AddGraph: Anomaly Detection in Dynamic Graph Using Attention-based Temporal GCN.. In *IJCAI*, Vol. 3. 7.

[61] Yu Zheng. 2015. Trajectory data mining: an overview. *ACM Transactions on Intelligent Systems and Technology (TIST)* 6, 3 (2015), 1–41.

[62] Yu Zheng, Quannan Li, Yukun Chen, et al. 2008. Understanding mobility based on GPS data. In *Proceedings of the 10th international conference on Ubiquitous computing*. 312–321.

[63] Yu Zheng, Xing Xie, Quannan Li, and Wei-Ying Ma. 2008. Mining user similarity based on location history. In *SIGSPATIAL'08* (sigspatial'08 ed.).

[64] Yu Zheng, Xing Xie, Wei-Ying Ma, et al. 2010. GeoLife: A collaborative social networking service among user, location and trajectory. *IEEE Data Eng. Bull.* 33, 2 (2010), 32–39.

[65] Chong Zhou and Randy C. Paffenroth. 2017. Anomaly Detection with Robust Deep Autoencoders. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Halifax, NS, Canada) *(KDD '17)*. Association for Computing Machinery, New York, NY, USA, 665–674. https://doi.org/10.1145/3097983.3098052

[66] Hao Zhou, Dongchun Ren, Huaxia Xia, Mingyu Fan, Xu Yang, and Hai Huang. 2021. Ast-gnn: An attention-based spatio-temporal graph neural network for interaction-aware pedestrian trajectory prediction. *Neurocomputing* 445 (2021), 298–308.

[67] Jie Zhu, Wei Jiang, An Liu, Guanfeng Liu, and Lei Zhao. 2015. Time-dependent popular routes based trajectory outlier detection. In *Web Information Systems Engineering–WISE 2015: 16th International Conference, Miami, FL, USA, November 1-3, 2015, Proceedings, Part I 16*. Springer, 16–30.

[68] Andreas Züfle et al. 2023. Urban life: a model of people and places. *Computational and Mathematical Organization Theory* 29, 1 (2023), 20–51.