Trajectory Anomaly Detection with Language Models

Jonathan Kabala Mbuya
jmbuya@gmu.edu
George Mason University
Fairfax, Virginia, USA

Dieter Pfoser
dpfoser@gmu.edu
George Mason University
Fairfax, Virginia, USA

Antonios Anastasopoulos
antonis@gmu.edu
George Mason University
Fairfax, Virginia, USA

ABSTRACT

This paper presents a novel approach for trajectory anomaly detection using an autoregressive causal-attention model, termed LM-TAD. This method leverages the similarities between language statements and trajectories, both of which consist of ordered elements requiring coherence through external rules and contextual variations. By treating trajectories as sequences of tokens, our model learns the probability distributions over trajectories, enabling the identification of anomalous locations with high precision. We incorporate user-specific tokens to account for individual behavior patterns, enhancing anomaly detection tailored to user context. Our experiments demonstrate the effectiveness of LM-TAD on both synthetic and real-world datasets. In particular, the model outperforms existing methods on the Pattern of Life (PoL) dataset by detecting user-contextual anomalies and achieves competitive results on the Porto taxi dataset, highlighting its adaptability and robustness. Additionally, we introduce the use of perplexity and surprisal rate metrics for detecting outliers and pinpointing specific anomalous locations within trajectories. The LM-TAD framework supports various trajectory representations, including GPS coordinates, staypoints, and activity types, proving its versatility in handling diverse trajectory data. Moreover, our approach is well-suited for online trajectory anomaly detection, significantly reducing computational latency by caching key-value states of the attention mechanism, thereby avoiding repeated computations. The code to reproduce experiments in this paper can be found at the following link: https://anonymous.4open.science/r/LMTAD-31EA/.

CCS CONCEPTS

• **Computing methodologies** → *Modeling and simulation.*

KEYWORDS

Anomalous Trajectories, Anomaly Detection, Trajectory Data, Language Modeling, Self-Supervised Learning

ACM Reference Format:
Jonathan Kabala Mbuya, Dieter Pfoser, and Antonios Anastasopoulos. 2024. Trajectory Anomaly Detection with Language Models. In *The 32nd ACM International Conference on Advances in Geographic Information Systems (SIGSPATIAL '24), October 29-November 1, 2024, Atlanta, GA, USA.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3678717.3691257

SIGSPATIAL '24, October 29-November 1, 2024, Atlanta, GA, USA

ACM ISBN 979-8-4007-1107-7/24/10.
https://doi.org/10.1145/3678717.3691257

1 INTRODUCTION

Figure 1: A conceptual visualization of trajectories as natural language statements. Language statements and trajectories share similarities: both consist of ordered elements from a finite set (words vs. GPS points) and require connections by semantic or spatiotemporal relationships to be coherent. They are governed by external rules (grammar for the language, road networks for trajectories) and vary by user or context (writing style vs. movement behavior).

Effective techniques for gathering and analyzing movement data, including the contribution of this work on anomaly detection are becoming increasingly important with the growth in terms of data and different types of applications. Specifically, trajectory anomaly detection has several interesting and practical use cases across various fields, such as Transportation and Traffic Analysis (accident detection, road safety analysis), Maritime Navigation and Safety (shipping lane monitoring, piracy detection), Air Traffic Control (airspace safety), Wildlife Monitoring (behavior change), Sports Analysis (injury prevention, game strategies), Healthcare and Elderly Care (behavior change and detecting health issues or emergencies), Disaster Response and Management (disaster response and crowd monitoring) and Urban Planning and Smart Cities (mobility analysis, public transit optimization, pedestrian safety). This work focuses on detecting trajectory anomalies that deviate from patterns observed in collections of historical datasets.

Extensive research has been conducted on trajectory anomaly detection for unlabeled data [5, 11, 16, 27, 34, 36]. However, this body of prior work has several limitations.

Firstly, it is difficult to pinpoint specific locations within the trajectory where the anomaly occurs, as the anomaly score is attributed to the entire trajectory or sub-trajectory. Secondly, these

methods do not adequately consider anomalies in relation to the individual user's context. This is significant because different users exhibit distinct behavior patterns, and what constitutes a normal pattern for one user might be deemed anomalous for another. Finally, anomaly detection has primarily focused on spatiotemporal trajectory data (i.e., GPS coordinates), but the concept of a trajectory can be more abstract. A trajectory can be a chronological sequence of qualitative staypoints (i.e., work → restaurant → apartment) for a particular user. Additionally, anomalies may not lie solely in the spatial properties of the data; they can also involve the types of places a user visits, such as a restaurant or a shopping mall on specific days of the week or the duration spent at a particular location.

To address these issues, we propose a **L**anguage **M**odel for **T**rajectory **A**nomaly **D**etection, (**LM-TAD**). The motivation for using a language modeling approach stems from the idea of modeling trajectories as statements [19], as illustrated in Figure 1. Language statements and trajectories share several similarities: 1) both consist of ordered elements from a finite set (words vs. GPS points and segments); 2) both require coherence, meaning the elements must be connected by semantic (language) or spatial/temporal (trajectories) relationships. For example, in Figure 1, the word *to* cannot be followed by any random word in the vocabulary, just as a GPS point can only be followed by a limited set of other GPS coordinates. 3) Additionally, both are governed by external rules—grammar for language and road networks or physical constraints for trajectories. Grammar dictates sentence construction, while road networks dictate possible routes from point A to point B. In Figure 1, the paths from the source (S) to the destination (D) are constrained by the road network. 4) Finally, the specific combination of elements (words vs. temporal sequence of locations) varies with the user or context. Similar to writing styles, users have different movement behaviors, determining the use of certain words or speed/mobility patterns, respectively. As illustrated in Figure 1, just as one can choose different words to convey the idea of going to the beach with friends, one can also select different trajectories to travel between the source (S) and destination (D). Based on these similarities, just as a language model can be trained to score the likelihood of sentences, we aim to train a model to score the probability of given trajectories and, hence, detect anomalous trajectories.

Our specific approach is using an autoregressive causal-attention model to learn the distributions over trajectories. We train the model by learning to predict the next location in a trajectory given a historical context. Having learned the model, we can compute the probability of generating a location (i.e., discretized GPS coordinate, staypoints, etc.) in a trajectory given its historical context, e.g., in its simplest case, a location history. Anomalies are detected by identifying low-probability locations. To learn normal behavior for specific users, we can further condition the trajectory generation with a unique user token (i.e., USER_ID) and flag anomalies on a user basis accordingly. The language model uses discrete tokens and can handle different abstractions of a trajectory, such as discretized GPS coordinates using spatial partitions or qualitative staypoint information (i.e., "home, work, restaurant, and so forth").

To distinguish between normal and anomalous trajectories, we use *perplexity*, a well-established metric in natural language processing. Intuitively, perplexity can be viewed as a measure of uncertainty when predicting the next token (location) in a trajectory. We also use the *surprisal rate* of each location to identify anomalous locations to identify where the anomaly is in a trajectory.

Our contributions can be summarized as follows:

- We propose a new way to detect anomalies in trajectory data by using an autoregressive causal-attention model. With this approach, we can 1) identify the location in the trajectory where the anomaly occurs, 2) find anomalies with respect to a user, and 3) handle various definitions of a trajectory (GPS coordinates, staypoints, etc.)
- We show the application of perplexity as a metric for identifying outlier trajectories, both in the context of the entire dataset and with respect to the trajectories of a specific user. Additionally, we illustrate using the surprisal rate to identify potential anomalous locations within a trajectory.
- Our findings indicate that our method performs exceptionally well on the Pattern of Life dataset (PoL) [43], effectively identifying anomalous trajectories in the context of a user while training on all data, including anomalies. We also show that our approach is on par with state-of-the-art methods for trajectory anomaly detection when tested on the Porto dataset [32, 33], using solely GPS coordinates. Furthermore, our approach is suitable for online anomaly detection as the trajectory is being generated. Unlike autoencoder methods that require the computation of the anomaly score for the entire sub-trajectory each time a new GPS coordinate is sampled, our method benefits from low latency by caching key-value (KV cache) states [15, 22] of the attention mechanism for previously generated tokens (i.e., GPS coordinates), thereby avoiding repeated computations.

The remainder of this paper is organized as follows. Section 2 discusses related work. Section 3 gives the basic formulation of the problem. In Section 4, we present our autoregressive generative approach to detect anomalies in trajectories. Section 5 provides an experimental evaluation that highlights the benefits of our method compared to existing approaches. Finally, Section 7 concludes and provides directions for future work.

2 RELATED WORK

2.1 Trajectory Anomaly Detection

Existing work for anomaly detection in trajectories can be grouped into two broad categories: heuristic-based methods [5, 13, 17, 34, 41] and learning-based methods [26, 27].

Heuristic-based methods primarily rely on hand-crafted features to represent normal routes and employ distance or density metrics to compare a target route to normal routes. The study in [13] suggests a partition-and-detect framework for trajectory outlier detection, effectively identifying outlying sub-trajectories by combining two-level trajectory partitioning with a hybrid distance-based and density-based detection approach. Studies by [34] and [5] introduce related methods that systematically extract, group, and analyze trajectories based on the source and destination. These methods identify anomalies by how rare they are and how much

they deviate from usual patterns, using the principle of isolation to ensure effective and reliable anomaly detection. Another research effort by [41] presents a time-dependent approach for detecting trajectory anomalies, employing edit distance metrics to ascertain whether a given target trajectory deviates significantly from historical normal trajectories. Similarly, [17] uses edit distance coupled with a density-based clustering algorithm to identify anomalous trajectories. Heuristic-based methods exhibit certain limitations. Primarily, the characterization of a trajectory is dependent on manually curated features encompassing various parameters, such as frequency, distance, or density thresholds, to flag a trajectory as anomalous. Furthermore, the construction of these features tends to be domain-specific, necessitating specialized expertise in the respective field. Additionally, the applicability of these methods across different regions is constrained owing to the inherent dissimilarities in trajectories across diverse geographical locations.

Learning-based methods rely on machine learning techniques. The work by [27] employs trajectory embedding learned by recurrent neural networks (RNN) to capture the sequential information and distinctive characteristics of trajectories to detect anomalies. However, the model requires labeled data, usually unavailable in real applications due to the cost of labeling a dataset. Several studies on unlabelled data have been proposed to overcome the limitations of using labeled data. [26] suggest a method combining Infinite Gaussian Mixture Models with bi-directional Generative Adversarial Networks to detect anomalies in trajectory data using a combination of the reconstruction loss and discriminator-based loss. Deep learning methods based on autoencoders have recently been applied to various anomaly detection tasks [7, 18, 39, 42]. These methods work by learning to compress and reconstruct the input. They use the premise that anomalous input will produce a significant reconstruction error, as they differ from the learned normal patterns. A recent study by [16] proposes an autoencoder method for online anomalous trajectory detection with multiple Gaussian components in the latent space to discover various types of normal routes. Outside of autoencoder methods, another recent study by [37] suggests a reinforcement learning-based solution for detecting anomalous trajectories and sub-trajectories. However, the methods have two main limitations. Firstly, these methods are limited in pinpointing the specific location of an anomaly within the trajectory, as they rely on an aggregate anomaly metric, typically the reconstruction error. Secondly, there is a notable lack of generalizability in these approaches to scenarios requiring user-specific anomaly identification, as what constitutes an anomaly for one user might be deemed normal for another.

2.2 Language Modeling on Trajectory Data

The field of language modeling has received much attention recently since the introduction of the transformer model [30]. Language models like BERT [8], GPT-2 [24], and LLaMA [28] have been shown to achieve great performance on a variety of natural language tasks, including question-answering, sentiment analysis, and text generation. Language modeling techniques have been extended to other applications, including image classification [23, 40] and speech processing [2, 3]. Recent studies have also applied language modeling techniques to a wide range of applications on mobility

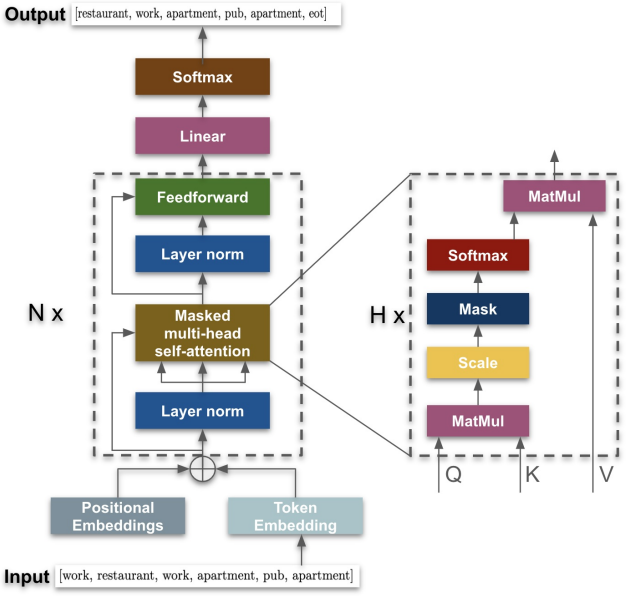


Figure 2: Architecture of LM-TAD, our trajectory model.

data. For example, the work in [31] leverages language modeling techniques for human mobility forecasting tasks, while the work in [12] uses similar techniques to predict the next visited location in a trajectory. The work in [19] proposes a conceptualization of a BERT-inspired system tailored for trajectory analysis. However, none of the previous work used a generative approach for anomaly detection in trajectory data.

3 PROBLEM FORMULATION

A trajectory is a finite chronological sequence of *visited locations* and can be modeled as a list of space-time points modeled as location and time stamp pairs $T = p_0, \ldots, p_n$ with $p_i = \langle l_i, t_i \rangle$ and $l_i \in R^2, t_i \in R^+$ for $i = 0, 1, \ldots, n$ and $t_0 < t_1 < t_2 < \ldots < t_n$.

In the simplest case (shown above), a location l_i is represented as a geographic coordinate in two-dimensional space. Other representations could be to map locations to cells of a discretized space such as a regular spatial or a hexagonal grid [29].

Alternatively, l_i can capture qualitative staypoints (visited points of interest such as "home," "work," or "restaurant") or spatial partitions that capture functional areas of a city, e.g., “commercial”, “business”, or “residential” areas. Therefore, l_i can include both a staypoint and functional area, e.g., $l_i = [\text{apartment}, \text{downtown}]$).

A collection of related trajectories T_i constitutes a dataset $\mathcal{D}$. The dataset $\mathcal{D}$ may contain both normal and anomalous trajectories. In general, an anomalous trajectory refers to one that does not show the *normal* mobility pattern and deviates from the majority of the trajectories in $\mathcal{D}$ [6, 36]. Given a dataset $\mathcal{D}$ with n trajectories, our goal is to train a model that distinguishes between normal and anomalous trajectories without having explicit labels.

(a) Discretized GPS (b) Staypoints

(c) Staypoints + duration (d) Activities

Figure 3: Example location configurations. Locations can be (a) discretized GPS coordinates, (b) staypoints, (c) staypoints enhanced with dwell time, or (d) activities.

4 METHOD

Our approach is to train a model that learns probability distributions over trajectories. An autoregressive generative model will allow us to infer the probability of a trajectory given the historical context

$$P(T) = p(l_1)p(l_2|l_1)p(l_3|l_1l_2)\dots p(l_i|l_{<i})\dots p(l_n|l_{<n}) \quad (1)$$

where the probability of each location $p(l_i|l_{<i})$ is conditioned on all previous location history. We note that there is no time bound between locations; however, such information we could also use that as part of the input. With this approach, we can find anomalous trajectories and identify exactly which locations in the trajectory are anomalous.

4.1 Model and Architecture

Given a dataset of trajectories $\mathcal{D} = \{T_1, \dots, T_m\}$, **LM-TAD**'s goal is to maximize the likelihood of all the trajectories in the dataset:

$$\mathcal{L}(\mathcal{D}) = \sum_{i=1}^{|\mathcal{D}|} \sum_{j=1}^{|T_i|} \log P(l_i|l_{<i}; \theta) \quad (2)$$

where $P(\cdot;\theta)$ is the conditional probability modeled by a neural network parameterized by θ. To learn the parameters θ, we opt for a transformer-based network architecture [30]. This architecture choice is motivated by its proven efficacy in natural language generation tasks, suggesting its potential applicability and effectiveness in modeling trajectories as statements.

Figure 2 shows the overall architecture of our method, **LM-TAD**, which consists of positional and token embeddings, N transformer blocks followed by a linear transformation, and a softmax layer.

To capture input semantics, the token embedding layer transforms each token (location) from a categorical type to a finite-dimensional real-valued vector. Positional embeddings play a critical role in the training process, compensating for the absence of inherent sequential ordering within the causal-attention module. Each transformer block comprises a multi-head causal-attention mechanism, which is preceded and succeeded by a layer-normalization layer and a feedforward layer. In the multi-head causal-attention mechanism, a trajectory is transformed into three sets of vectors —keys, values, and queries— and then split into multiple heads for parallel processing. Each head independently computes a scaled dot-product attention to get attention scores that assess the relevance of different locations (tokens) in a trajectory. This allows the model to concurrently learn dependencies between locations, such as temporal or spatial ones. The outputs from all heads are concatenated and linearly transformed to produce the final output. Additionally, the causal-attention mechanism includes a masking operation to prevent the attention function from accessing information from future tokens (locations), given the autoregressive nature of our approach.

Below is the formal description of the muti-head self-attention:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right)\mathbf{V} \quad (3)$$

where d_k is the dimension of the keys. Concatenating the output values results in:

$$\begin{aligned} \text{MultiHead}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) &= \text{Concat}(\text{head}_1, ..., \text{head}_h)\mathbf{W}^o \\ \text{with head}_i &= \text{Attention}(\mathbf{Q}\mathbf{W}_i^Q, \mathbf{K}\mathbf{W}_i^K, \mathbf{V}\mathbf{W}_i^V) \end{aligned} \quad (4)$$

where the $\mathbf{W}_i^Q \in \mathbb{R}^{d_{model} \times d_q}$, $\mathbf{W}_i^K \in \mathbb{R}^{d_{model} \times d_k}$, $\mathbf{W}_i^K \in \mathbb{R}^{d_{model} \times d_v}$ are projection matrices that are learned during training. The projection matrix $\mathbf{W}^o$ linearly combines the outputs from different attention heads, enabling the model to flexibly adjust and fine-tune the aggregated attention, thereby enhancing the model's capacity to learn complex patterns. In **LM-TAD**, $d_q = d_k = d_v = d_{model}/h$ where h is the number of heads.

The feedforward layer consists of two linear transformations linked with a ReLU activation function:

$$\text{FFM}(x) = \max(0, \mathbf{x}\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2 \quad (5)$$

where the weights $\mathbf{W}_1 \in \mathbb{R}^{d_{model} \times d_{ff}}$, $\mathbf{W}_2 \in \mathbb{R}^{d_{ff} \times d_{model}}$ and the biases $\mathbf{b}_1 \in \mathbb{R}_{d_{ff}}$, $\mathbf{b}_2 \in \mathbb{R}_{d_{model}}$. The transformer block uses the layer-normalization layer in addition to residual connections to stabilize learning and improve training efficiency.

The output of the transformer block goes through linear and softmax layers to predict the distribution of each token in a trajectory.

.

4.2 Location Configurations

An advantage of using a generative approach to model trajectories is the ability to abstract the locations of a trajectory in different ways. Figure 3 shows a few examples of location configurations. In the simplest case, a trajectory can be represented by a finite chronological sequence of GPS coordinates. These coordinates can be discretized using regular grid cells (Figure 3a) [14] or hexagons [9]. However, various other trajectory configurations are possible. Instead of GPS coordinates, we can also use staypoints ("home", "workplace", "restaurant", etc.) (Figure 3b) or points of interest. We can even model a trajectory as a chronological sequence of a person's activities ("eating", "working", and "playing sports") (Figure 3d), where each location in the trajectory corresponds to a person's activity at a particular time. These trajectories can even be enhanced with additional metadata, such as the dwell time at a location (Figure 3c), method of transportation, or proximity to the

previous location. An advantage of LM-TAD is its ability to work with any of these different trajectory configurations.

4.3 Anomaly Score

We use *perplexity* to determine how anomalous a trajectory is. Perplexity is a well-established measure to evaluate language models [4, 8, 21], and can be viewed as a measure of uncertainty when predicting the next token (location) in a trajectory [20]. Euqation 6 shows how we can calculate the perplexity of a trajectory T with t locations, and Equation 7 shows how we compute the perplexity over a dataset $\mathcal{D}$ with n trajectories T_i.

$$PPL(T) = \exp\left\{-\frac{1}{n}\sum_{i=1}^{t}\log P(l_i|l_{<i})\right\} \tag{6}$$

$$PPL(\mathcal{D}) = -\frac{1}{n}\sum_{i=1}^{n} PPL(T_i) \tag{7}$$

The *lowest possible perplexity is 1*, which implies that the model can correctly predict the next location with absolute certainty. However, the maximum of this measure is unbounded. To determine when a trajectory is anomalous ("high" perplexity), we need to provide a threshold. We note that the choice of a threshold can be application- and dataset-dependent [1]. We can compute the threshold as follows:

$$\text{threshold} = \text{mean}[PPL(\mathcal{D})] + \text{std}[PPL(\mathcal{D})]$$

where mean$[PPL(\mathcal{D})]$ and std$[PPL(\mathcal{D})]$ are the mean and standard deviation of the perplexities of n training trajectories. To identify abnormal trajectories with respect to a specific user, we customize the threshold for each user. Here, the mean and standard deviation will be computed only using the training samples of that user.

5 EXPERIMENTS

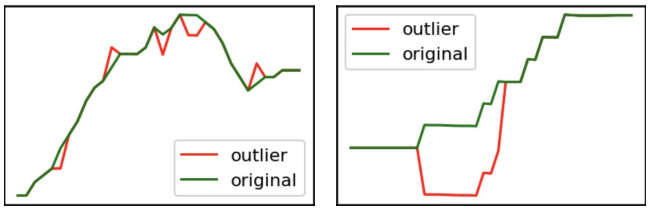
In this section, we describe the experimental setup used to evaluate the effectiveness of our proposed LM-TAD model. We compare our method against several state-of-the-art baselines on two different datasets and utilize different evaluation metrics to measure the accuracy and robustness of anomaly detection.

5.1 Datasets & Preprocessing

In our experiments, we use simulated and real-world datasets. Specifically, we use the Pattern-of-Life (PoL) simulation dataset [43] and the Porto taxi dataset [32, 33].

5.1.1 Pattern-of-Life Dataset (PoL). The PoL dataset was generated through the pattern of life simulation [43]. The PoL simulation consists of virtual agents designed to emulate humans' needs and behavior by performing human-like activities. Activities include going to work, restaurants, and recreational activities with friends. These activities are performed in real locations obtained from OpenStreetMap [10]. While agents engage in these activities, the simulation also records the location, which includes the GPS coordinates and the staypoints (i.e., home, work, restaurant), as well as the timestamps at the location.

Using the raw data from the PoL simulation, we created daily trajectories for each agent, consisting of places they visited on that particular day. The geographic coverage was Atlanta, GA and we simulated the behavior of an agent population consisting of working professionals.



(a) Random shift anomalies **(b) Detour anomalies**

Figure 4: Example of generated anomalies with $\alpha = 0.3$ and $\beta = 3$ for both types of anomalies on the Porto dataset.

The dataset includes an average of 450 daily trajectories for each of the 1000 generated agents, resulting in a total of 444,634 trajectories. Each input to the model represents a virtual agent's daily trajectory. To capture the patterns of each agent, the agent ID is included at the beginning of the trajectory. Based on the hypothesis that individual behavioral patterns exhibit consistency on the same days of the week, we also incorporate weekday information into the feature vector to enhance the model's ability to detect anomalies.

For example, a daily trajectory is represented as: [agent_ID, weekday, work, restaurant, apartment]. In our experiments, we also consider other location confirmations, including discretized GPS coordinates and the duration of stay at a location. We use Uber hexagons [29] for discretized GPS and discretize the stay duration into 1-hour buckets, using a sequence of bucket IDs as input to the model.

The PoL dataset comes with labels to identify anomalous trajectories generated by the simulation. To introduce anomalies, the simulation selects ten virtual agents exhibiting anomalous behaviors. For example, work anomaly is one type supported by this simulation: agents with work anomalies will abstain from going to work when they typically would.

For agents with anomalous trajectories, we have the first 450 days representing normal behavior and the last 14 days that exhibit anomalous behavior.

We trained our model on the entire dataset, including the additional 14 days of anomalous behavior from the ten virtual agents, to ensure it could identify outliers even when they were present in the training data. We then tested our methods against the baselines using the entire dataset

5.1.2 Porto Dataset. The Porto dataset consists of data generated by 442 taxis operating in the city of Porto in Portugal from January 07, 2013, to June 30, 2014. A taxi reports its GPS location at 15s intervals. We employed preprocessing steps similar to [16] and [14]. We discretize the map into $100m \times 100m$ grids and group trajectories with the same source and destination. We discarded trajectories belonging to a "source-destination" group with fewer than 25 trajectories. The input to our model consists of a vector of chronologically ordered and discretized GPS coordinates (grid cells) prepended with SOT (start of trajectory) and appended EOT (end of trajectory) tokens.

Table 1: Outlier detection on RED outlier agents for the Pattern of Life dataset. The best results are bolded. LM-TAD clearly outperforms the baselines, showcasing its modality adaptation capabilities

	baselines						ours	
	SAE		VSAE		GM-VSAE		LM-TAD	
Agent	F1	PR-AUC	F1	PR-AUC	F1	PR-AUC	F1	PR-AUC
57	0.00	0.01	0.00	0.04	0.16	0.03	**0.72**	**0.59**
62	0.00	0.11	0.00	0.02	0.10	0.09	**0.87**	**0.85**
347	0.00	0.02	0.00	0.02	0.25	0.40	**0.78**	**0.78**
546	0.00	0.07	0.00	0.03	0.12	0.12	**0.75**	**0.63**
551	0.00	0.01	0.00	0.02	0.00	0.04	**0.76**	**0.63**
554	0.00	0.02	0.00	0.02	0.00	0.04	**0.61**	**0.46**
644	0.00	0.03	0.00	0.01	0.00	0.04	**0.76**	**0.75**
900	0.00	0.01	0.00	0.06	0.32	0.17	**0.70**	**0.69**
949	0.00	0.01	0.00	0.02	0.00	0.06	**0.77**	**0.79**
992	0.00	0.03	0.00	0.02	0.11	0.13	**0.78**	**0.72**

Since this dataset did not have ground-truth labels of what trajectory is anomalous, we artificially generated anomalies following the work in [16] and [38]. We created two types of anomalies, (i) random shift and (ii) detour anomalies. Figure 5 shows an example of the two types of outliers. For random shift anomalies, we perturb α percentage of locations in a trajectory and move those location β grid cells away. For detour anomalies, we create a detour for the α percentage portion of a trajectory and shift the detour β grid cells away from the original trajectory. Following previous work on this dataset for anomaly detection [14], we did not include the artificially generated anomalous data during training.

5.1.3 Tokenization & Vocabulary. Given the nature of a language model architecture, we created tokens to form our model's vocabulary. In the Porto dataset, a token is considered a discretized GPS coordinate. We also added three special tokens: SOT (start of trajectory), EOT (end of trajectory), and PAD (padding token to help with batch training). In the POL dataset, tokens consist of staypoints (work, apartment, restaurant, etc.), days of the week (Monday, Tuesday, etc.), agent ID, and the EOT and PAD special tokens.

5.2 Baselines

We compared our method to existing unsupervised anomaly detection methods on trajectory data. Given the established better performance of deep learning methods on trajectory anomaly detection [14], we omitted the inclusion of traditional clustering-based algorithms.

- **SAE**: This is a standard autoencoder method trained to optimize the reconstruction loss of a trajectory sequence using a recurrent neural network. Based on the work in [18] and [1], we use the reconstruction error as the anomaly score.
- **VSAE** This method is similar to SAE. However, in addition to optimizing for the reconstruction loss, it also optimizes the KL divergence between the learned distribution over the latent space and a predefined prior [1, 25]. Similar to SAE, we use the reconstruction error as the anomaly score.
- **GM-VSAE** [16]. This method generalizes the VSAE by modeling the latent space with more than one Gaussian component and also uses the reconstruction error as the anomaly score.

5.3 Evaluation Metrics

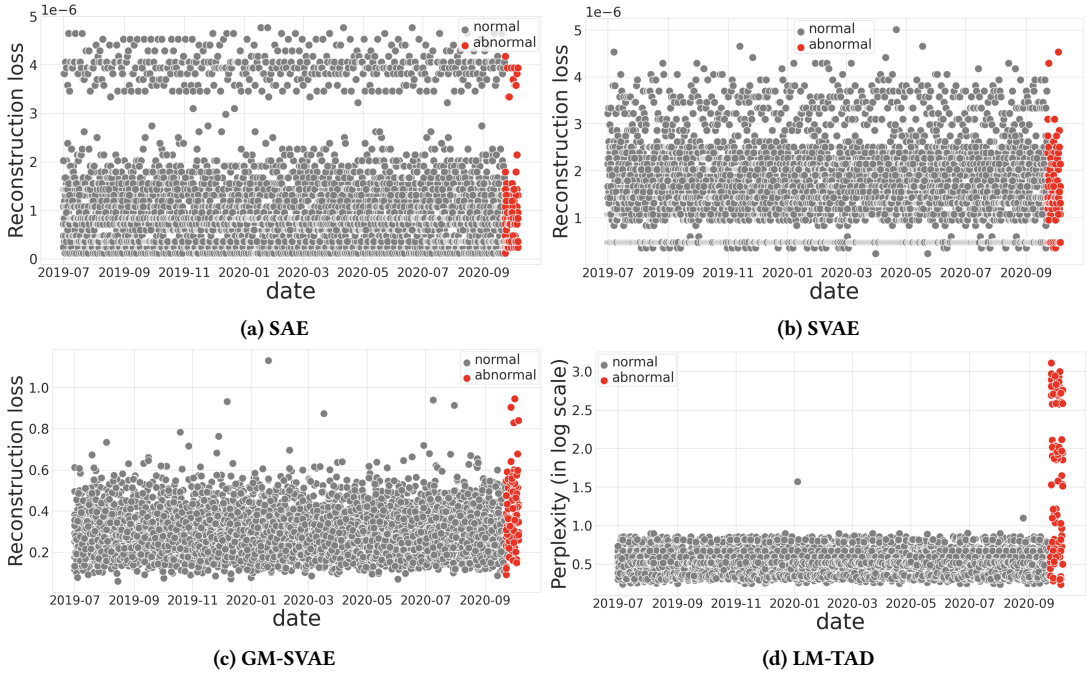
We use Precision-Recall AUC and F1 scores to evaluate the performance of our method and the baseline methods [16, 35]. These metrics are suitable for assessing the performance of anomaly detection methods as the number of anomalies in each dataset is small compared to normal trajectories. For the Porto dataset, these metrics are computed across all trajectories. Conversely, we conduct these evaluations on a per-virtual-agent basis for the Pattern-of-Life (PoL) dataset. Additionally, the surprisal rate metric is used to locate the specific occurrence of an anomaly within a trajectory.

6 RESULTS

The following sections present the anomaly detection results for the PoL and Porto datasets. Anomaly detection for the Porto dataset is a *global challenge*, since taxi movements are customer/ride-driven and the movements captured by individual trajectories are largely independent. This is in contrast to the PoL dataset, which contains sets of trajectories that model the behavior of individual agents, and anomaly detection will be agent-specific.

6.1 Agent-based Outliers - Patterns-of-Life Data

In the PoL dataset, we have user IDs, i.e., sets of trajectories that can be linked to a specific user, and the anomaly detection challenge becomes user-specific. This is justified by the fact that the anomalous behavior of one agent may be the normal behavior of another agent. Therefore, we report results for the ten virtual agents that have each 14 additional days of anomalous behaviors. Table 1 summarizes F1 and Precision-Recall Area-under-the-Curve (PR-AUC) results for these ten anomalous agents. Results of the rest of the agents are not included in table 1 because these agents are not anomalous within their respective contexts. LM-TAD *outperforms all competitor methods* as it is more efficient in finding anomalies with respect to



(a) SAE (b) SVAE

(c) GM-SVAE (d) LM-TAD

Figure 5: Anomaly results for all methods trained on all the pattern of life dataset trajectories. Each dot represents the perplexity of a trajectory for any of the ten agents with normal and anomalous trajectories. Unlike other methods, our method (d) distinguishes between anomalous trajectories and normal trajectories by scoring most anomalous trajectories with high perplexity.

anomalous users. We identify three reasons why the autoencoder approaches fail to identify anomalies for the PoL dataset. 1) First, autoencoder approaches are optimized to reconstruct the input by minimizing the reconstruction error. This training approach would tend to learn a model that overfits training data, which may contain anomalies and hence fail to distinguish normal and anomalous trajectories. 2) Secondly, autoencoder-based methods yield a high reconstruction error for inputs divergent from the overall data patterns within the dataset. Nonetheless, these methods are suboptimal for identifying anomalies on an individual-agent basis. Theoretically, GM-SVAE can model distinct Gaussian distributions that can correspond to each virtual agent, thereby learning trajectory distributions unique to each agent. However, this approach has little control over the distributions learned by each component in the latent dimension. Even if we had control over the distributions in the latent space, this approach would be very expensive to train in practice as the increase in the number of agents would require an increase in Gaussian distributions. 3) The final reason is related to defining what we consider anomalies. Recent literature on trajectory anomaly data [16, 36] define normal trajectory between a source (S) and destination (S) as a trajectory that was traveled by the majority of taxis in the train data. Therefore, in this context, an anomalous trajectory was not or was readily traveled on by taxis. However, the PoL dataset's anomalies differ from the Porto dataset's. As described in section 6.1, anomalies in the PoL dataset refrain from their typical behavior. For example, they refrain from going to work on days of the week when they are supposed to go to work. Autoencoder approaches fail to capture these types of anomalies by not being able to learn the anomalous behavior on a per-agent basis.

LM-TAD offset the limitations of the autoencoder approaches for the following reasons. 1) First, LM-TAD can learn the likelihood that a particular agent will visit a particular location. Therefore, if an agent hardly visited a location, even if it was part of the training data, the model will give a low probability of such a location and distinguish anomalies. 2) LM-TAD uses a special token to provide the context for generating an agent's trajectory, therefore finding anomalies on an individual-agent basis. 3) LM-TAD is highly customized for different types of anomalies. LM-TAD can learn the normal pattern for each agent for each day of the week because LM-TAD can condition the generation of a particular trajectory with an agent and week special tokens. Therefore, this approach is still practically feasible even if the number of agents increases.

Moreover, Figure 5 illustrates as to why competitor methods have low F1 and PR-AUC scores. In this figure, the red dots represent the perplexity or reconstruction error associated with the trajectories of agents. The expectation is that trajectories deemed anomalous (indicated by red dots) would yield higher levels of perplexity or reconstruction error. However, for autoencoder-based methods, we find that the reconstruction error associated with anomalous trajectories is comparatively lower than that of many normal trajectories. In contrast, LM-TAD aligns with the expected model behavior, *attributing higher perplexity scores to anomalous trajectories.*

Table 2: Anomaly detection results on the Porto dataset. The best results for a particular metric in a specific category are bolded. LM-TAD performs largely on par with the best baseline.

	Random shift anomalies						Detour anomalies					
anomalies params:	$\alpha = 3, \beta = 0.1$		$\alpha = 5, \beta = 0.1$		$\alpha = 3, \beta = 0.3$		$\alpha = 3, \beta = 0.1$		$\alpha = 5, \beta = 0.1$		$\alpha = 3, \beta = 0.3$	
Metric	F1	PR-AUC	F1	PR-AUC	F1	PR-AUC	F1	PR-AUC	F1	PR-AUC	F1	PR-AUC
baselines:												
SAE	0.44	0.66	0.53	0.75	0.71	0.90	0.30	0.47	0.36	0.54	0.58	0.78
VSAE	0.40	0.67	0.51	0.76	0.69	0.90	0.28	0.47	0.35	0.56	0.57	0.77
GM-VSAE-10	**0.85**	0.90	**0.86**	**0.91**	**0.91**	**0.99**	**0.73**	**0.72**	**0.79**	**0.77**	**0.90**	**0.96**
ours: LM-TAD	**0.85**	**0.91**	0.85	**0.91**	0.90	**0.99**	0.69	0.65	0.73	0.68	0.89	**0.96**

6.2 Global Outliers - Porto Taxi Data

The results for the Porto taxi dataset are summarized in Table 2. We use different parameter configurations α and β for random-shift and detour anomalies. For random shift, we perturb α percent of locations in a trajectory, moving them β grid cells away. Similarly, for detour anomalies, we create a detour for α percent of a trajectory, shifting the detour β grid cells away from the original trajectory.

Our method outperforms the SAE and VSAE methods. For random shift anomalies, LM-TAD shows performance comparable to that of GM-SVAE. For detour anomalies, GM-SVAE is able to identify more anomalies than our method, especially for cases where the detour is about 10% of the entire trajectory. This behavior of LM-TAD may be explained by the perplexity of being less sensitive to capturing anomalies that consist of changing a small continuous portion of the trajectory (detour). Since most continuous locations in the trajectory are normal, and as such, those probabilities are fairly high, the overall perplexity will also be relatively high. Conversely, in instances of random shift anomalies, our method exhibits comparable and sometimes superior performance to GM-SVAE. This enhanced detection efficacy can be ascribed to the fact that the random shift anomalies break the continuity dependence of one location to its history, resulting in a sequence of locations with lower probabilities, subsequently lowering the perplexity.

We can also observe that the distance (β) of the detour or the randomly shifted location from the original trajectory has less impact on finding anomalous trajectories than the fraction (α) of the trajectory with anomalies. This suggests that metrics that tend to summarize the anomaly of a trajectory by looking at the entire trajectory (i.e., reconstruction error) may find identifying anomalous trajectories with few anomalous locations challenging. We introduce a local *surprisal rate* metric for such cases to offset this limitation.

6.3 Identifying Anomalies using Surprisal Rate

As discussed previously, perplexity alone may not be enough to separate anomalous trajectories from normal ones. Perplexity, being a trajectory-level score, aggregates the scores across all tokens or locations within a trajectory. Consequently, the presence of only a few anomalous tokens may lead to their signal being diluted by the averaging process. As such anomalies might go undetected in such scenarios. Similar limitations apply to autoencoder-based methods, where the reconstruction loss is calculated over all tokens in a trajectory.

A further limitation in using perplexity or the reconstruction error is the inability to *pinpoint the specific location of anomalies within a trajectory*. Here, our work proposes the *surprisal rate* measure that operates at the level of individual locations or tokens within a trajectory.

In our empirical analysis, we explored the application of the surprisal rate for detecting potentially anomalous locations within a trajectory on the PoL data. A high surprisal rate suggests that a particular location in a trajectory may be anomalous. Figure 6 shows the surprisal rate for 30 trajectories of the PoL dataset (ten anomalous and twenty normal ones, randomly chosen from the respective agents). The analysis reveals that certain tokens in anomalous trajectories exhibit significantly higher surprisal rates compared to those in normal trajectories, particularly at the beginning of the trajectories. This pattern aligns with the dataset's structure and the configuration of our input vector, where the initial tokens represent the agent ID, the weekday, and the first location visited by the virtual agent on that day. Given the expected pattern of agents visiting consistent locations on specific weekdays, deviations from this routine, such as visiting an atypical location as the first destination, are flagged as anomalies. Consequently, the inclusion of the weekday token in the trajectory analysis enables the identification of instances where an agent's initial location deviates from the norm, resulting in a larger surprisal rate when an agent visits an unusual place on a given weekday.

6.4 Location Configurations - Ablation Study

We conducted an ablation study to show the versatility of LM-TAD in working with different types of inputs. In this study, we explore the usage of discretized GPS coordinates (Uber hexagons [29]), staypoint labels (i.e., work, restaurant, and so forth), and stay duration (the duration at a particular location) as input for the model to infer the anomaly detection performance of each modality. One of the main advantages of using one modality over another is the type of anomalies we are interested in discovering. Anomalies can lie in the duration at a particular location (stay longer than usual), by visiting an unusual geographical area not visited before, or by going to a different place (i.e., shopping mall) on a given day when supposed to go to another place (i.e., work).

Table 3 summarizes the results of using various location configurations on the PoL dataset. The use of a staypoint label performs the

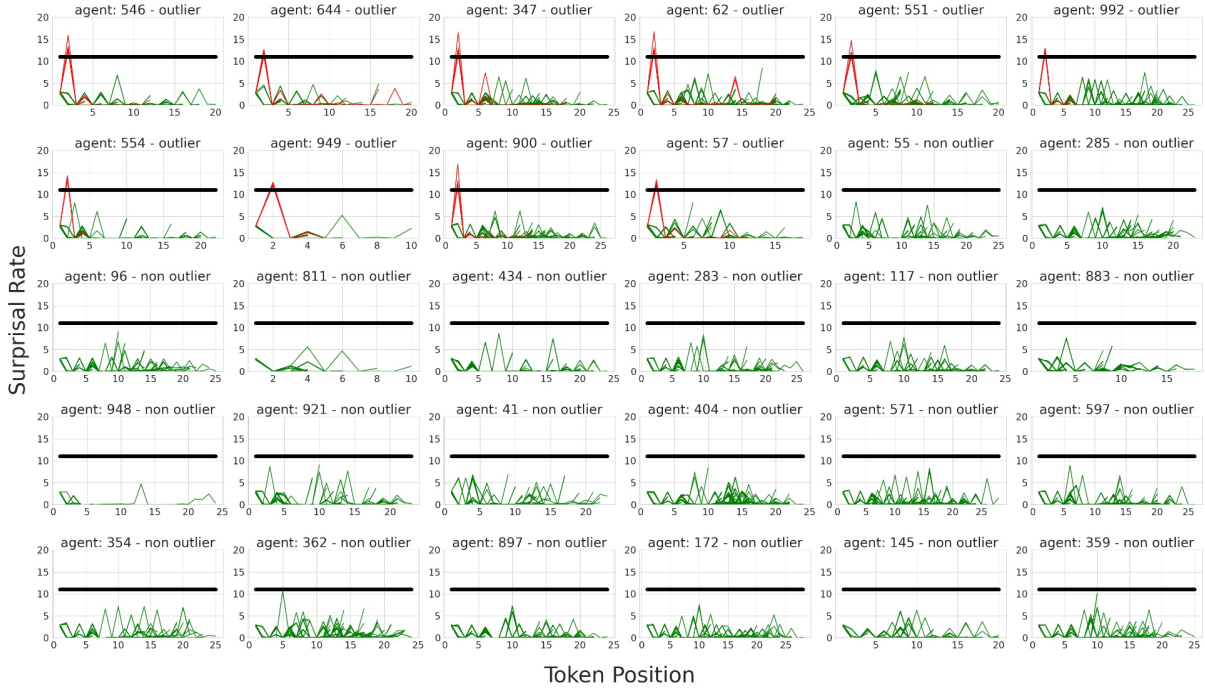


Figure 6: Surprisal rate through trajectories on the Pattern of Life dataset. We plot the trajectories of each agent with some anomalous trajectories (10 first agents), and we randomly selected 20 agents with no anomalous trajectories. The horizontal line shows the (arbitrary) threshold for a surprisal rate high enough to correspond to anomalous trajectories. Anomalous trajectories plotted in red have high surprisal rates for certain locations in the trajectories as opposed to normal trajectories. Hence, LM-TAD can identify anomalous trajectories based on the surprisal rate.

best in identifying anomalies. This is consistent with the anomalies in the PoL dataset as discussed in section 6.1, where agents abstain from going to places they normally go on given days of the week. GPS coordinates can still identify anomalies because the data encapsulates spatial information. The stay duration at locations also proves effective, as disruptions in visited locations also affect the time spent at those locations. These findings underscore the adaptability of our approach to using different feature configurations to find anomalies.

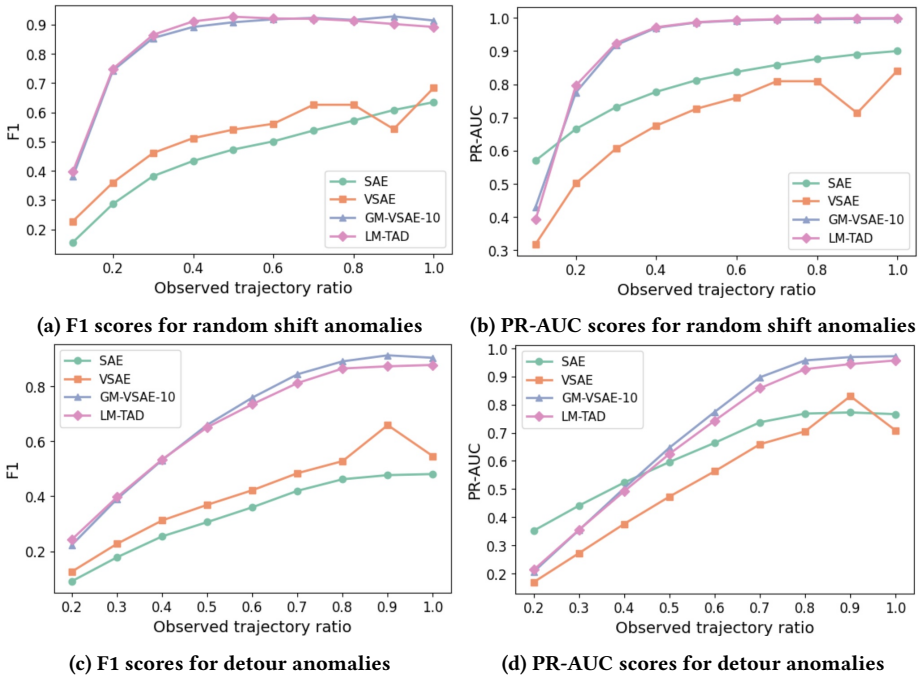
6.5 Online Anomaly Detection

One of the advantages of LM-TAD is the support of online anomalous trajectory detection. LM-TAD does not require the whole trajectory to be generated to compute the anomaly score. In addition, we do not need to know the destination as a priority (although such knowledge would enhance the anomaly detection of sub-trajectories). For instance, as soon as a trip starts, LM-TAD can compute the anomaly score of a partial trajectory each time a new coordinate is sampled. Autoencoder approaches can be used for online anomalous trajectory detection as well. However, they are significantly more expensive to use since they must compute the anomaly score for the entire sub-trajectory each time a new GPS coordinate is sampled. LM-TAD, instead, can cache the key-value (KV cache) states [15, 22] of the attention mechanism for previously generated tokens (i.e., GPS coordinates). This significantly reduces the need for repetitive computations and lowers the latency in computing the anomaly score.

Table 3: Anomaly detection using different location configurations. The best results are bolded. The staypoint label location type performs the best overall, reflecting the types of anomalies present in the PoL dataset

	Staypoint label		Discretized GPS		Stay duration	
Agent	F1	PR-AUC	F1	PR-AUC	F1	PR-AUC
57	**0.62**	**0.69**	0.50	0.54	0.26	0.49
62	0.86	**0.81**	**0.88**	0.80	0.78	0.80
347	**0.78**	**0.75**	0.75	0.68	0.72	0.70
546	**0.75**	**0.76**	0.67	0.65	0.67	0.62
551	0.67	**0.63**	**0.76**	0.63	0.52	0.48
554	**0.62**	0.46	0.53	**0.61**	0.62	0.46
644	**0.80**	**0.78**	0.57	0.70	0.64	0.64
900	**0.71**	**0.66**	0.46	0.60	**0.71**	0.59
949	**0.82**	**0.77**	0.78	0.75	0.82	0.73
992	**0.78**	0.68	0.72	**0.73**	0.67	0.69
average	**0.74**	**0.70**	0.66	0.67	0.64	0.62

Figure 7 shows the accuracy of detecting anomalies on partial trajectories at different observation ratios on the Porto dataset. We evaluate partial trajectories with ratios from 0.2 to 1.0 with 0.1



(a) F1 scores for random shift anomalies

(b) PR-AUC scores for random shift anomalies

(c) F1 scores for detour anomalies

(d) PR-AUC scores for detour anomalies

Figure 7: Online Anomalous Trajectory Detection Results (POL data). The results show the performance of each model for the detour and random shift anomalies as we evaluate different ratios of the trajectory from 0.1 to 1.0. LM-TAD shows competitive results and can detect anomalies of sub-trajectories.

increment. The results suggest that LM-TAD is more than competitive in detecting sub-trajectory anomalies. Especially for random shift anomalies, 40% of the sub-trajectory is enough to detect most anomalies in the dataset compared to the entire trajectory. Detour anomalies may not be easily detected with small ratios of trajectories without knowing the destination because a detour becomes obviously anomalous only *given* the destination or a large portion of the trajectory. In general, LM-TAD performs on par with the best baseline, but as discussed before, comes with the advantage of significantly lower latency.

7 CONCLUSION

In this work, we introduced LM-TAD, an innovative trajectory anomaly detection model leveraging an autoregressive causal-attention mechanism. By conceptualizing trajectories as sequences akin to language statements, our model effectively captures the sequential dependencies and contextual nuances necessary for precise anomaly detection. We demonstrated that incorporating user-specific tokens enhances the model's ability to detect context-specific anomalies, addressing the variability in individual behavior patterns.

Our extensive experiments validated the robustness and adaptability of LM-TAD across various datasets, including the Pattern of Life (PoL) and Porto taxi datasets. The results show that LM-TAD vastly outperforms existing state-of-the-art methods in identifying user-contextual anomalies. At the same, it has competitive performance for detecting outliers in GPS-based trajectory data.

We introduced perplexity and surprisal rate as metrics for outlier detection and localization of anomalies within trajectories, broadening the analytical capabilities of the approach. The model's ability to handle diverse trajectory representations, from GPS coordinates to staypoints and activity types, underscores its versatility and uniqueness.

Importantly, our approach also proves advantageous for online trajectory anomaly detection, reducing computational latency, and gaining a significant performance advantage over existing models. This will enable real-time anomaly detection without the need for repeated expensive computations.

In summary, LM-TAD represents a substantial advance in trajectory anomaly detection, offering a *scalable*, *context-aware*, and *computationally efficient* solution. This work paves the way for future research in user-centric analysis and real-time anomaly detection in trajectory data.

ACKNOWLEDGMENT

This work was supported by the National Science Foundation (Award 2127901) and by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior/ Interior Business Center (DOI/IBC) contract number 140D0419C0050. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes, notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DOI/IBC, or the U.S. Government. Additionally, this work was supported by resources provided by the Office of Research Computing, George Mason University and by the National Science Foundation (Award Numbers 1625039, 2018631).

REFERENCES

[1] Jinwon An and Sungzoon Cho. 2015. Variational Autoencoder based Anomaly Detection using Reconstruction Probability. https://api.semanticscholar.org/CorpusID:36663713

[2] Alexei Baevski and Abdelrahman Mohamed. 2020. Effectiveness of Self-Supervised Pre-Training for ASR. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 7694–7698. https://doi.org/10.1109/ICASSP40776.2020.9054224

[3] Alexei Baevski, Henry Zhou, Abdel rahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations. *ArXiv* abs/2006.11477 (2020). https://proceedings.neurips.cc/paper/2020/file/92d1e1eb1cd6f9fba3227870bb6d7f07-Paper.pdf

[4] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models Are Few-Shot Learners *(NIPS'20)*. Curran Associates Inc., Red Hook, NY, USA, Article 159, 25 pages.

[5] Chao Chen, Daqing Zhang, Pablo Samuel Castro, Nan Li, Lin Sun, Shijian Li, and Zonghui Wang. 2013. iBOAT: Isolation-Based Online Anomalous Trajectory Detection. *IEEE Transactions on Intelligent Transportation Systems* 14, 2 (2013), 806–818. https://doi.org/10.1109/TITS.2013.2238531

[6] Chao Chen, Daqing Zhang, Pablo Samuel Castro, Nan Li, Lin Sun, and Shijian Li. 2012. Real-Time Detection of Anomalous Taxi Trajectories from GPS Traces. In *Mobile and Ubiquitous Systems: Computing, Networking, and Services*, Alessandro Puiatti and Tao Gu (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 63–74.

[7] Jinghui Chen, Saket Sathe, Charu Aggarwal, and Deepak Turaga. [n. d.]. *Outlier Detection with Autoencoder Ensembles*. 90–98. https://doi.org/10.1137/1.9781611974973.11 arXiv:https://epubs.siam.org/doi/pdf/10.1137/1.9781611974973.11

[8] Jacob "Devlin, Ming-Wei Chang, Kenton Lee, and Kristina" Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. https://doi.org/10.18653/v1/N19-1423

[9] Zipei Fan, Xiaojie Yang, Wei Yuan, Renhe Jiang, Quanjun Chen, Xuan Song, and Ryosuke Shibasaki. 2022. Online Trajectory Prediction for Metropolitan Scale Mobility Digital Twin. In *Proceedings of the 30th International Conference on Advances in Geographic Information Systems* (Seattle, Washington) *(SIGSPATIAL '22)*. Association for Computing Machinery, New York, NY, USA, Article 103, 12 pages. https://doi.org/10.1145/3557915.3561040

[10] Hugo Ledoux Filip Biljecki and Peter van Oosterom. 2013. Transportation mode-based segmentation and classification of movement trajectories. *International Journal of Geographical Information Science* 27, 2 (2013), 385–407. https://doi.org/10.1080/13658816.2012.692791 arXiv:https://doi.org/10.1080/13658816.2012.692791

[11] Kathryn Gray, Daniel Smolyak, Sarkhan Badirli, and George O. Mohler. 2018. Coupled IGMM-GANs for deep multimodal anomaly detection in human mobility data. *ArXiv* abs/1809.02728 (2018). https://arxiv.org/abs/1809.02728

[12] Ye Hong, Henry Martin, and Martin Raubal. 2022. How do you go where?: improving next location prediction by learning travel mode information using transformers. *Proceedings of the 30th International Conference on Advances in Geographic Information Systems* (2022). https://arxiv.org/abs/2210.04095

[13] Jae-Gil Lee, Jiawei Han, and Xiaolei Li. 2008. Trajectory Outlier Detection: A Partition-and-Detect Framework. In *2008 IEEE 24th International Conference on Data Engineering*. 140–149. https://doi.org/10.1109/ICDE.2008.4497422

[14] Xiucheng Li, Kaiqi Zhao, Gao Cong, Christian S. Jensen, and Wei Wei. 2018. Deep Representation Learning for Trajectory Similarity Computation. In *2018 IEEE 34th International Conference on Data Engineering (ICDE)*. 617–628. https://doi.org/10.1109/ICDE.2018.00062

[15] Akide Liu, Jing Liu, Zizheng Pan, Yefei He, Gholamreza Haffari, and Bohan Zhuang. 2024. MiniCache: KV Cache Compression in Depth Dimension for Large Language Models. https://arxiv.org/pdf/2405.14366

[16] 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

[17] Zhongjian Lv, Jiajie Xu, Pengpeng Zhao, Guanfeng Liu, Lei Zhao, and Xiaofang Zhou. 2017. Outlier trajectory detection: a trajectory analytics based approach. In *Database Systems for Advanced Applications (Lecture Notes in Computer Science)*, Selçuk Candan, Lei Chen, Torben Bach Pedersen, Lijun Chang, and Wen Hua (Eds.). Springer, Springer Nature, United States, 231–246. https://doi.org/10.1007/978-3-319-55753-3_15 22nd International Conference on Database Systems for Advanced Applications (DASFAA) ; Conference date: 27-03-2017 Through 30-03-2017.

[18] Pankaj Malhotra, Anusha Ramakrishnan, Gaurangi Anand, Lovekesh Vig, Puneet Agarwal, and Gautam M. Shroff. 2016. LSTM-based Encoder-Decoder for Multi-sensor Anomaly Detection. *ArXiv* abs/1607.00148 (2016). https://arxiv.org/pdf/1607.00148.pdf

[19] Mashaal Musleh, Mohamed F. Mokbel, and Sofiane Abbar. 2022. Let's Speak Trajectories. In *Proceedings of the 30th International Conference on Advances in Geographic Information Systems* (Seattle, Washington) *(SIGSPATIAL '22)*. Association for Computing Machinery, New York, NY, USA, Article 37, 4 pages. https://doi.org/10.1145/3557915.3560972

[20] Helen Ngo, Joao M. de Ara'ujo, Jeffrey Hui, and Nick Frosst. 2021. No News is Good News: A Critique of the One Billion Word Benchmark. *ArXiv* abs/2110.12609 (2021). https://api.semanticscholar.org/CorpusID:239769114

[21] Matthew E. "Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke" Zettlemoyer. 2018. Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics, New Orleans, Louisiana, 2227–2237. https://doi.org/10.18653/v1/N18-1202

[22] Reiner Pope, Sholto Douglas, Aakanksha Chowdhery, Jacob Devlin, James Bradbury, Anselm Levskaya, Jonathan Heek, Kefan Xiao, Shivani Agrawal, and Jeff Dean. 2022. Efficiently Scaling Transformer Inference. *ArXiv* abs/2211.05102 (2022). https://arxiv.org/pdf/2211.05102

[23] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning Transferable Visual Models From Natural Language Supervision. In *International Conference on Machine Learning*. https://proceedings.mlr.press/v139/radford21a/radford21a.pdf

[24] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners. https://api.semanticscholar.org/CorpusID:160025533

[25] Noveen Sachdeva, G. Manco, Ettore Ritacco, and Vikram Pudi. 2018. Sequential Variational Autoencoders for Collaborative Filtering. *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining* (2018). https://arxiv.org/pdf/1811.09975.pdf

[26] Daniel Smolyak, Kathryn Gray, Sarkhan Badirli, and George Mohler. 2020. Coupled IGMM-GANs with Applications to Anomaly Detection in Human Mobility Data. *ACM Trans. Spatial Algorithms Syst.* 6, 4, Article 24 (jun 2020), 14 pages. https://doi.org/10.1145/3385809

[27] Li Song, Ruijia Wang, Ding Xiao, Xiaotian Han, Yanan Cai, and Chuan Shi. 2018. Anomalous Trajectory Detection Using Recurrent Neural Network. In *Advanced Data Mining and Applications*, Guojun Gan, Bohan Li, Xue Li, and Shuliang Wang (Eds.). Springer International Publishing, Cham, 263–277.

[28] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models. *ArXiv* abs/2302.13971 (2023). https://arxiv.org/pdf/2302.13971.pdf

[29] Inc Uber. 2023. H3 Hexagonal hierarchical geospatial indexing system. https://h3geo.org/. Accessed: 11/22/23.

[30] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc. https://arxiv.org/pdf/1706.03762.pdf

[31] Hao Xue, Bhanu Prakash Voutharoja, and Flora D. Salim. 2022. Leveraging language foundation models for human mobility forecasting. *Proceedings of the 30th International Conference on Advances in Geographic Information Systems* (2022). https://arxiv.org/pdf/2209.05479.pdf

[32] Jing Yuan, Yu Zheng, Xing Xie, and Guangzhong Sun. 2011. Driving with Knowledge from the Physical World *(KDD '11)*. Association for Computing Machinery, New York, NY, USA, 316–324. https://doi.org/10.1145/2020408.2020462

[33] Jing Yuan, Yu Zheng, Chengyang Zhang, Wenlei Xie, Xing Xie, Guangzhong Sun, and Yan Huang. 2010. T-Drive: Driving Directions Based on Taxi Trajectories *(GIS '10)*. Association for Computing Machinery, New York, NY, USA, 99–108. https://doi.org/10.1145/1869790.1869807

[34] Daqing Zhang, Nan Li, Zhi-Hua Zhou, Chao Chen, Lin Sun, and Shijian Li. 2011. IBAT: Detecting Anomalous Taxi Trajectories from GPS Traces. In *Proceedings of the 13th International Conference on Ubiquitous Computing* (Beijing, China) *(UbiComp '11)*. Association for Computing Machinery, New York, NY, USA, 99–108. https://doi.org/10.1145/2030112.2030127

[35] Longmei Zhang, Wei Lu, Feng Xue, and Yanshuo Chang. 2023. A trajectory outlier detection method based on variational auto-encoder. *Mathematical biosciences and engineering : MBE* 20 8 (2023), 15075–15093. https://api.semanticscholar.org/CorpusID:259926930

[36] Qianru Zhang, Zheng Wang, Cheng Long, Chao Huang, Siu-Ming Yiu, Yiding Liu, Gao Cong, and Jieming Shi. 2023. Online Anomalous Subtrajectory Detection on Road Networks with Deep Reinforcement Learning. In *2023 IEEE*

39th International Conference on Data Engineering (ICDE). 246–258. https://doi.org/10.1109/ICDE55515.2023.00026

[37] Qianru Zhang, Zheng Wang, Cheng Long, Chao Huang, Siu-Ming Yiu, Yiding Liu, Gao Cong, and Jieming Shi. 2023. Online Anomalous Subtrajectory Detection on Road Networks with Deep Reinforcement Learning. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*. 246–258. https://doi.org/10.1109/ICDE55515.2023.00026

[38] Guanjie Zheng, Susan L. Brantley, Thomas Lauvaux, and Zhenhui Li. 2017. Contextual Spatial Outlier Detection with Metric Learning *(KDD '17)*. Association for Computing Machinery, New York, NY, USA, 2161–2170. https://doi.org/10.1145/3097983.3098143

[39] 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

[40] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. 2021. Learning to Prompt for Vision-Language Models. *International Journal of Computer Vision* 130 (2021), 2337 – 2348. https://arxiv.org/pdf/2109.01134.pdf

[41] Jie Zhu, Wei Jiang, An Liu, Guanfeng Liu, and Lei Zhao. 2015. Time-Dependent Popular Routes Based Trajectory Outlier Detection. Springer-Verlag, Berlin, Heidelberg, 16–30. https://doi.org/10.1007/978-3-319-26190-4_2

[42] Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Dae ki Cho, and Haifeng Chen. 2018. Deep Autoencoding Gaussian Mixture Model for Unsupervised Anomaly Detection. In *International Conference on Learning Representations*. https://sites.cs.ucsb.edu/~bzong/doc/iclr18-dagmm.pdf

[43] Andreas Züfle, Carola Wenk, Dieter Pfoser, Andrew Crooks, Joon-Seok Kim, Hamdi Kavak, Umar Manzoor, and Hyunjee Jin. 2021. Urban life: a model of people and places. *Computational and Mathematical Organization Theory* 29 (11 2021), 1–32. https://doi.org/10.1007/s10588-021-09348-7

A MODEL DETAILS

A.1 Architecture

A.2 Implementation Details