Using a Language Model's Perplexity for Evaluating a Trajectory's Outlierness

Zhihao Lyu
Khoury College of Computer Sciences
Northeastern University
San Jose, CA, United States
lv.zh@northeastern.edu

LeAnn Marie Mendoza
Khoury College of Computer Sciences
Northeastern University
San Jose, CA, United States
mendoza.l@northeastern.edu

Ying Shen
Khoury College of Computer Sciences
Northeastern University
San Jose, CA, United States
shen.ying@northeastern.edu

Jie Zhang
Khoury College of Computer Sciences
Northeastern University
San Jose, CA, United States
zhang.jie4@northeastern.edu

Mario Nascimento
Khoury College of Computer Sciences
Northeastern University
Vancouver, BC, Canada
m.nascimento@northeastern.edu

Michal Aibin
Khoury College of Computer Sciences
Northeastern University
Vancouver, BC, Canada
m.aibin@northeastern.edu

***Abstract*—**

***Index Terms*—Trajectory data, Outlier Detection, Perplexity, NLG, Machine Learning, and Location.**

I. Introduction

In the realm of spatio-temporal (ST) data, trajectories refer to a sequence of spatial and temporal locations that describe the path or movement of an object or entity over time. This data can be generated by a wide range of sources, including GPS devices, mobile phones, sensor networks, and surveillance cameras and represent the movement of various entities, such as vehicles, pedestrians, wildlife, ships, and even disease diagnosis [38]. The data typically includes time-stamped coordinates (latitude and longitude), and additional attributes like speed, direction, and altitude. The applications of trajectory data are far-reaching, ranging from enabling location-based services [15] such as navigation apps and targeted advertising to aiding businesses in engaging customers based on their real-time location and preferences.

There are a multitude of applications of trajectory data, including prediction, classification, and outlier detection [40]. Outlier detection holds substantial significance in the domain of spatio-temporal (ST) applications, particularly for its crucial function in identifying anomalous occurrences. This detection technique finds applicability across a range of pragmatic contexts, and a plethora of researchers have historically engaged with this subject [38]–[54]. Nevertheless, the perpetual evolution of new technologies, such as BERT [56], continually affords innovative avenues for advancing research in this area.

In this research, we tackled the task of ST Trajectory outlier detection as a natural language processing challenge. Our approach involved a two-fold process. We first developed a model to map trajectories from their latitude longitude space to "sentences" with unique tokens that correspond to ST regions on a map, then implemented a strategy that employed BERT [56], a seq2seq context-aware language model, to identify outlier trajectories as measured by perplexity scores.

In navigating through the multifaceted landscape of spatio-temporal (ST) trajectory outlier detection, this research embarks on a comprehensive exploration, unfolding in several critical stages. In the following literature review (II), we'll apply define trajectory outliers, exploring various definitions and labels provided by previous researchers in section II-A. Then, in section II-B, we'll shift our focus to the traditional ways of detecting outliers, specifically diving into methods that do not leverage language learning models. In section II-C, we'll delve into the world of Language Learning Models, like BERT, exploring how these models identify outlier words in sentences and what makes them effective, or not, in doing so. Lastly, in section II-D, we'll bring it all together, connecting the dots between language learning models and trajectory outliers detection, with a close examination of a relevant data set. This discussion informs our research gap in Section II-E, as well as how we devise to solve this gap using a combination of previous work coupled with Large Language Models(LLM).

II. Motivation

This research aligns with scientists' enduring efforts to capture trajectory outliers using cutting-edge technology, with a focus on enhancing both accuracy and efficiency in detection. It has come to our attention that contemporary methodologies, grounded in the deployment of large language models, involve a process in which trajectory data, such as GPS coordinates, timestamps, and velocity, is transmuted into high-dimensional features. Subsequently, these features undergo fine-tuning via large language models. Nonetheless, the crux of the matter lies in the fact that modern-day advanced language models have been predominantly trained on human language. As a consequence, deploying numerical data, such as GPS coordinates, as input may engender an adverse impact on the

Identify applicable funding agency here. If none, delete this.

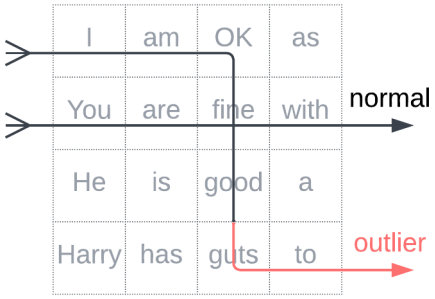

Fig. 1. Example of trajectory mapping and outlier detection

outlier detection. Furthermore, our investigations have revealed that human language exhibits a heightened sensitivity and tolerance towards anomalies. An example being the human capacity to comprehend sentences with inverted word orders and to discern phrasing that is characterized by a disconcerting tonality.

Within the context of this paper, our primary objective is to introduce a novel algorithm and model that is equipped to transform trajectory data, into textual statements (Fig. 1). Subsequently, we employ large language models to undertake modeling of the post-transformation statements. This approach enables us to maximize the computational and predictive potential afforded by, state-of-the-art language models. For instance, subsequent to substantial exposure to a corpus of trajectory data, the language model might detect a semantic transition within a trajectory segment, where the term ”good” transitions to ”fine,” and subsequently undergoes transformation into ”clothes” (Fig. 1). This process continues until the model’s predictive perplexity for the subsequent term surpasses a intolerable threshold. At this juncture, the occurrence of an outlier can be inferred. Such predictive anomaly trajectory mechanisms offer a novel perspective for the current trajectory modeling research, facilitating a more effective identification of anomalous trajectories within smart city systems.

III. Literature Review

A. Trajectory Outlier Definition

Many research studies have delved into trajectory outlier detection, there is no widely accepted definition for trajectory outliers. These definitions vary based on different criteria, considering various types of motion and distinct research objectives. The most commonly used definition, as proposed by Hawkins [27], describes an outlier as an observation that significantly deviates from other observations, raising suspicions that it was generated by a different mechanism.

Some researchers employ key parameters, such as direction, density, speed, and acceleration, to identify similarities and anomalies in trajectories. For instance, Dodge et al. [28] use a time-ordered sequence of coordinates to measure movement similarity, enabling them to detect outliers. Ibrahim et al. [29] introduce speed to study the changing patterns of tropical cyclones. Johnson et al. [29] utilize acceleration sensors in smartphones to distinguish between aggressive and non-aggressive driving behaviors.

Others emphasize the role of spatial context in determining trajectory outliers. Buchin et al. [31] and Elsner et al. [32]have found that movement outliers may be influenced by underlying land/sea structures, latitude, surface temperature, and pressure. This concept also extends to vehicle trajectories, where environmental factors such as road categories and traffic conditions can influence the trajectories of vehicles.

Time is a crucial dimension used by some researchers to define trajectory outliers. For example, Suzuki et al. [33] employ Hidden Markov Models to model time-series features and identify trajectory outliers. Pang et al. [34] adapt likelihood ratio test statistics to identify trajectory outliers that do not conform to periodic variations on a time unit basis.

Additionally, a trajectory outlier is defined as a deviation on high-dimensional features in the field of machine learning. Huang [35] leverages neural networks to represent trajectories as words and employs clustering methods for outlier detection. Ahmed et al. [36] and Belhadi et al. [37] utilize convolutional neural networks to transform trajectories into graphs to detect outliers. Su et al. [21] introduces meta-learning to incorporate high-dimensional features, addressing the issue of insufficient data for outlier detection.

B. Trajectory Outlier Detection Methods

In the previous section, we define spatio-temporal (ST) outliers as objects or instances the perform irregular behavior concerning their spatial neighbors [38]. ST data are comprised of two type of attributes, behavioral and contextual, according to Eldawy et al. [39]. Behavioral attributes spotlight the changeable aspects of an observation while contextual attributes provide the necessary backdrop to fully comprehend, evaluate, and utilize those observations in a meaningful way [39]. Spatio-temporal data has many types, including trajectories, a sequence of reading points of a moving object time-stamped periodically [40]–[42]. In turn, ST trajectory outlier detection becomes a task of finding trajectories that do no behave like the usual behavior of other trajectories in a dataset [43].

Eldawy et al. [39] proposed that the current state of ST trajectory outlier detection methods can be categorized in four types: distance and density-based outlier detection [44], [45], pattern outlier detection, supervised and semi-supervised learning, and statistical and probabilistic techniques. We will be exploring these categories in the following subsections.

1) Distance and Density-based Outlier Detection Methods: A distance-based outlier is a distant object from its neighbors [45], while a density-based outliers an object that does not meet a threshold amount of neighbors [44]. An object is then classified as an outlier when it meets either or both distance-based and density-based criteria. Although, with real life data, the distribution of points within a sample is not always uniform

[39] leading current methods to utilize a blend of the two approaches [46].

In 2006, the DBSCAN algorithm was developed by Kut et al. for computing the ST-outlier using a regional query of two values of epsilons; eps1 and eps2, to recover spatial neighbors, and non-spatial neighbors, respectively [47]. The intersection of eps1 and eps2 then revealed the ST neighbors of an object, and thus the outliers [47]. Distance and density-based approaches are applied well into clustering methodology [48] in which objects not belonging to a cluster after various clustering techniques are classified as an outlier [41], [48], [49].

Detection of sub-trajectory outliers is also a subsection of ST Trajectory outlier detection. First explored in 2008, Lee et al. [46] proposed a partition-and-detect framework coined TRAOD, to detect outliers within a trajectory by partitioning trajectories into line segments and detecting anomalous line segments. Although this method suffered from computational time overhead and complexity of O(n2) [46].

Distance and density-based methods for identifying spatio-temporal outliers utilize straightforward calculations and theory. However, each approach encounters computational overhead, particularly when managing extensive data related to moving objects [39], [46].

2) Pattern Outlier Detection Methods: For vehicle, flight, maritime, and point-to-point trajectories, the process of detecting a ST trajectory outliers' pattern depends on sequencing [39]. Utilizing this method, Zhang et. al [50] proposed an Isolation-Based Anomalous Trajectory (iBAT) detection method that achieved a performance of AUC ¿ 0.99. The iBAT system exploits the property that anomalous trajectories are "few and different" from the majority of normal trajectories, which are "many and similar" [50]. Given a large collection of GPS traces and a city map split into grid-cells of equal size, a taxi's trajectory is mapped as a sequence of the traversed cells[13]. Through this process, historical data can unveil anomalous patterns between source to end destinations of a taxi ride [50].

One of the main challenges of this method is the low-sampling-rate problem, where a taxi trajectory often consists of a series of GPS points that are not adjacent to each other, leading to different representations of the same taxi trajectory, which can affect the performance of anomaly detection methods [50]. Another challenge is the sparsity problem, where there may not be sufficient taxi trajectories between certain source-destination pairs to form "normal" trajectory clusters, thus making it difficult to distinguish anomalous trajectories from normal ones [50].

3) Supervised and Semi-supervised Learning Methods: With labeled ST trajectory data, a supervised or semi-supervised learning model can be applied to recognize the patterns and detect outliers [51] . ROAM (Rule- and Motif-Based Anomaly Detection in Moving Objects) [51] presents a methodology where the trajectories of objects are articulated through discrete pattern segments, known as motifs. The associated features, such as time and location, are extracted and organized into a hierarchical feature space, providing a multi-resolution perspective of the data [51] .

However, one downfall of ROAM, and other supervised and semi-supervised learning models, is that they require a large amount of labeled training data and can be computationally expensive to train and apply, especially for images and/or videos [52] . This can be difficult to obtain in many real-world applications, where it may be expensive or time-consuming to collect and label data. Additionally, ROAM can be sensitive to the quality of the training data. If the training data contains outliers, ROAM may learn to classify those outliers as normal, which can lead to false negatives.

4) Statistic or Probabilistic Methods: This next subcategory of ST trajectory outlier detection utilizes statistics and probability to detect anomalies. Rogers et al. [53] proposed a statistical method for detecting spatio-temporal outliers using the Stroud algorithm. This algorithm computes a strangeness factor for each object in the dataset, which is the sum of the weighted distances to its nearest neighbors in geographic, temporal, and behavioral space [53]. The strangeness of each object is then compared to that of reference objects using a statistical test [**?**]. Objects with a significantly higher strangeness than the reference objects are identified as spatio-temporal outliers [**?**]. However, this approach requires identifying the standard objects in the dataset, which can be difficult in many real-world applications.

To mitigate this, Albanese et al. [54] proposed a Rough Outlier Set Extraction (ROSE) approach for exploring the top outliers in an unlabeled spatio-temporal dataset. ROSE works by first constructing a Kernel set, which is a small subset of the original dataset that is representative of the entire dataset. The Kernel set is constructed using a rough set theory-based approach. Once the Kernel set has been constructed, ROSE computes a strangeness factor for each object in the original dataset. The strangeness factor is a measure of how different an object is from the Kernel set. Objects with high strangeness factors are more likely to be outliers. Finally, ROSE identifies the top outliers in the dataset as the objects with the highest strangeness factors [54].

The introduction of the Kernel set is a key innovation of the ROSE approach. The Kernel set allows ROSE to identify outliers without the need for labeled data. Additionally, the Kernel set can be used to improve the efficiency of the outlier detection process [39].

Overall, ROSE is a promising new approach for outlier detection in spatio-temporal data. It is particularly well-suited for applications where labeled data is limited or unavailable.

C. Outlier Detection using Language Models (LMs)

Sentences that fall into the category of outliers are often characterized by their peculiarity, unexpectedness, or substantial deviation from the usual and coherent patterns found in natural language. This often involves the identification of words exhibiting unusual characteristics within the context of the sentence [17].

The utilization of Large Language Models (LLMs) for outlier detection in natural language processing plays a vital role in uncovering defects, purging data impurities [17], and mitigating the risk of system failures caused by anomalies [19].

Researchers have delved into the realm of LLMs to detect outliers and anomalies. For instance, Elhafsi et al. harnessed the power of the LLM, specifically the text-davinci-003 model, to scrutinize and pinpoint potential semantic anomalies (outliers) in their study [19]. Their work involved the integration of LLMs into a monitoring framework designed to detect semantic anomalies in real time as a robot perceives and interacts with its environment. A key aspect of their approach involved the transformation of visual data into natural language descriptions using an open vocabulary object detector.

In a different vein, Jansen et al. [23] leveraged pre-trained LLMs to tackle the issue of harmful internet content detection by employing sentence perplexity calculations. The concept here is that when sentence perplexity surpasses a predetermined threshold, the content is classified as potentially harmful. What sets their approach apart is the focus on training the model with harmful content, rather than positive content, leading to more robust results compared to traditional methods.

Furthermore, Todd et al. [24] present a novel method for unsupervised anomaly detection, a particularly challenging task due to the scarcity of labeled outlier data. Their approach involves the use of multiple models to calculate perplexity across various dimensions and subsequently summarizes these perplexity scores to provide valuable guidance in identifying the most likely error types.

D. Research Gap

The contemporary trajectory analysis landscape confronts a significant challenge: the intricate processing and semantic representation of latitude and longitude pairings that define trajectories as addressed through NLP systems. A lacuna exists in developing a method that balances both time complexity, granularity and structure when analysing various ST trajectories for outliers. Drawing from the iBAT grid representation, we seek to use NLP to detect outliers using perplexity scores.

Our proposed solution, a structured grid-based representation, addresses this challenge head-on. We commenced by instituting an equivalency-based grid structure spanning the focal geographic area. Uniquely, this grid was tailored to uphold spatial uniformity, ensuring each cell held an equivalent degree division of the encompassing area. This foundational step ensured that our trajectory analysis maintained consistent spatial resolution. However, merely defining grids does not capture the nuances inherent in real-world trajectory data, which can often showcase dense or clustered patterns. To bring this detail to the forefront, a density-based approach was also invoked and compared. By evaluating trajectory point concentrations within grid cells, we achieved a finer grasp of underlying trajectory behaviors.

For each cluster as defined by the mapping representation, we strategically assigned unique identifiers to areas, differentiating between dense and less dense regions using a standard deviation-based threshold. This distinction was paramount in ensuring that the representation was both comprehensive and sensitive to data nuances. The density-based approach ensured each identifier corresponded with a roughly equivalent number of trajectory points.

Utilizing NLP to model our data, we realized that outlier detection could benefit immensely from contextual awareness. Thus, we ventured to employ a sequence-to-sequence model. By training this model on our sentence-structured trajectory data, we converted the challenge of outlier detection into an NLP task. This transformation harnessed the advanced capabilities of modern seq2seq models, presenting an avant-garde perspective for trajectory analysis. Our method is poised to fill the existing gap, providing a structured yet nuanced approach to geospatial data representation and analysis.

IV. Problems Statement

2.1. Problem Statement

In the realm of spatio-temporal (ST) data, trajectories, which represent the sequential spatial and temporal locations of objects or entities, hold paramount significance across a myriad of domains. These trajectories, enriched with time-stamped coordinates and supplementary attributes, such as speed and direction, find applications in diverse fields, ranging from location-based services to real-time customer engagement. Notably, the detection of trajectory outliers plays a pivotal role in the ST domain, as it is instrumental in identifying anomalous occurrences within trajectory data. This capability holds promise for practical applications across numerous domains, from transportation and surveillance to epidemiology.

Despite the enduring interest in trajectory outlier detection, spurred by the continual evolution of technology, such as BERT and other large language models (LLMs), this research area presents ongoing opportunities for innovation and improvement. Traditional methods for identifying trajectory outliers, while effective, may not fully harness the potential of language models, which excel in discerning anomalies in natural language data.

This research embarks on a journey to explore the multifaceted landscape of spatio-temporal (ST) trajectory outlier detection. Our focus is to leverage recent advancements in language learning models to enhance the accuracy and interpretability of outlier detection in trajectory data. Our research approach involves mapping trajectory data into a language-like representation and then employing as a context-aware language model to detect trajectory outliers.

We will first implement a grid-mapping system, similar to iBat, where each point will be assigned to a grid index represented by a unique string of characters, resulting in a ”sentence” representation of a trajectory. The sentences will be used as training data to fine-tune a LLM to recognize the patterns associated with each path, outputting a perplexity score for each sentence. The perplexity scores will be evaluated and sentences with a high perplexity (to be analyzed) will be considered outliers.

A. Evaluation Metrics

To assess the effectiveness of our approach, we will employ the following evaluation metrics:

Perplexity Score: We will use the perplexity score as a primary metric to evaluate the performance of our trajectory outlier detection model. Lower perplexity scores indicate better model performance in identifying outliers within the trajectory data.

Perplexity [57] is defined as the exponentiated average negative log-likelihood of a sequence. If we have a tokenized sequence, it can be mathematically represented as:

$$P(T) = 2^{-\frac{1}{N}\sum_{i=1}^{N}\log_2 p(w_i|w_{i-1})}$$

where:

- $P(T)$ is the perplexity of the text T,
- N is the total number of words in the text,
- $p(w_i|w_{i-1})$ is the probability of word w_i given the preceding word w_{i-1}.

Precision, Recall, and F1-Score: We will calculate precision, recall, and F1-score to measure the accuracy and effectiveness of our model in detecting trajectory outliers. Precision represents the proportion of correctly detected outliers, recall measures the model's ability to capture all actual outliers, and the F1-score balances the trade-off between precision and recall.

AUC-ROC: The Area Under the Receiver Operating Characteristic (AUC-ROC) curve will help us assess the model's ability to distinguish between normal and abnormal trajectories. A higher AUC-ROC score indicates superior performance.

Interpretability: We will also evaluate the interpretability of our approach by analyzing the extent to which the detected outliers are explained by the language-like representations and how easily domain experts can understand these explanations.

By employing these evaluation metrics, we aim to demonstrate the effectiveness of our approach in enhancing the accuracy and interpretability of spatio-temporal (ST) trajectory outlier detection, paving the way for more robust anomaly detection in trajectory data using advanced language models.

V. Algorithms

A. Taxi GPS Trajectory Data

For our analytical process, we leveraged labeled data originating from "MiPo: How to Detect Trajectory Outliers with Tabular Outlier Detectors." [58] Their experiments with taxi datasets showed that MiPo outperformed all baseline methods with 0.99 AUC on average. [58] Their experiments were carried out utilizing a taxi GPS trajectory dataset, with added labels to differentiate outliers from typical data points. This dataset was acquired by Yang et. al from the uci.edu website, accessed on 21 June 2022. It encompasses data from 442 taxis operating in Porto, Portugal, collected between 7 January 2013 and 30 June 2014. The dataset, featuring a GPS sampling interval averaging 15 seconds, contains 1,710,670 individual trajectories.

This dataset served as a foundational element, enabling us to conduct a comparative analysis to gauge the effectiveness and precision of our approach relative to established findings.

B. Employing Long Short-Term Memory Networks

The MiPo dataset [58] contained labeled non-outlier and outlier trajectories in the form of (x,y) points. To convert the trajectory dataset from a trace of points to sentences, we employed a mapping algorithm to assign membership of points within a grid cell containing m x n grid cells. This was done by computing the maximum and minimum x and y, respectively, serving as the range of the grid structure. By defining a 10x10 grid, we compute values that would serve as borders for each grid cell, defining 100 cells total. For each of these cells, we assign a random 3 letter identifier to serve as our "token" or word.

For each trajectory within the non-outlier and outlier dataset, we convert the trace of points into sentences. Each point is mapped to a grid cell and its resulting token is used to construct a sentence. We do this for each point while preserving order.

Once the sentences are constructed, the non-outlier sentences are used as input to a Long Short-Term Memory (LSTM) network [59], a type of recurrent neural network (RNN) architecture. LSTMs are capable of learning long-term dependencies in sequence data, making them suitable for natural language processing tasks [59]. We utilize PyTorch to instantiate the LSTM model. The LSTM learns the underlying patterns, structures, and word orderings typical of the "non-outlier" sentences.

We tested the model using a dataset of new non-outlier and outlier sentences and calculated the model's sentence-wide perplexity. In our analysis, we applied the logarithmic transformation to the perplexity scores and then normalized the data such that values corresponding to the non-outlier trajectories are closer to 0, and values corresponding to the outlier trajectories are closer to 1. This makes it easier to set a threshold for classification and visualization.

To determine the threshold value that assigns perplexity scores to either 'non-outlier' or 'outlier', we used the Area Under the Receiver Operating Curve (AUROC) to find the point that gives the best balance between True Positive Rate (TPR) and False Positive Rate (FPR), giving us our optimal threshold. We used that optimal threshold to classify the sentences based on the perplexity score and compared the produced labels with the MiPo labels to determine the accuracy of our process.

C. Using N-gram Language model

We adopted the Taxi GPS trajectory dataset with labels used by Yang et al. [58]. First, data were read from separate files to inner trajectories and outlier trajectories. The structures of the two datasets were described and they contain multiple trajectories, each trajectory is one array that contains lists of coordinates. The trajectories were visualized in Fig. 2, where inner trajectories were shown as blue lines and outlier trajectories were shown as red lines.

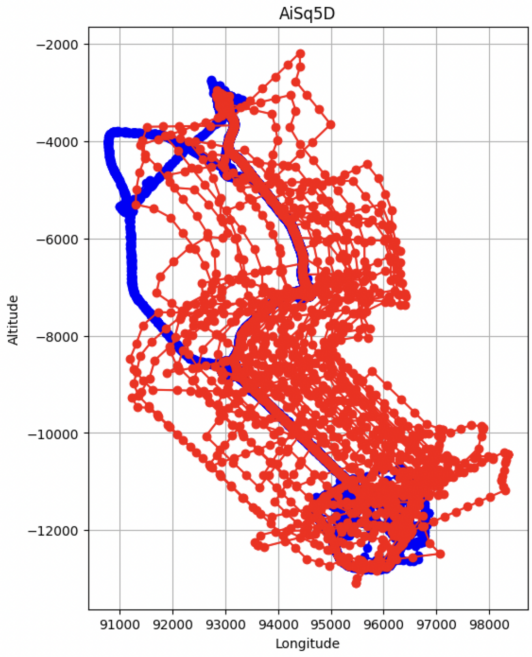

Fig. 2. Trajectories visualization

Then the mapping method mentioned by Mohammed et al. [60] was used to map trajectory data into words. A grid was first created by four steps: calculate the boundaries, define the grid cell size, generate random 3-letter labels for each grid cell, and then add labels to the grid cells. Then according to the position of the trajectory data located in the cell, we can map the trajectory coordinates to certain words in the cell.

Then the N-gram languages model was used to identify sentence outliers. We first tokenize the text data into words or sub-word units for the language model. For the model training part, we split inner data into training and test sets. The training data were used to train the language model, and we used the test set to calculate the perplexity of the model on the set.

For outlier detection, perplexity is commonly used to evaluate language models in natural language processing. It measures how well a language model predicts a sample of text and can be used to identify outliers in language data. To detect outliers using the language model's perplexity, we set a threshold for perplexity. Trajectory descriptions with perplexity scores significantly higher than the mean or median may be considered outliers.

$$\text{Threshold} = \text{Mean}(\text{Perplexity}) + 2 \times \text{Std}(\text{Perplexity})$$

For evaluation, the area under the Receiver Operating Characteristic (ROC) curve (AUC) was employed to evaluate the performance of the detectors. Our data are labeled data, where outliers are labeled as 1, and inners are labeled as 0.

$$\text{AUC} = \text{ROC\_AUC\_Score}(\text{true\_labels}, \text{outlier\_scores})$$

The AUC was a value between 0 and 1, where 1 indicated a perfect detector.

D. Utilizing BERT with semantic mapping augment

The process is designed for detecting trajectory outliers from semi-supervised or unsupervised trajectory data. The task is formally defined as follows: Given a set of trajectory with limited labels, our modeling solution allows clustering all trajectories into two groups. The methodological details are organized into four subsections. The structural steps are the following:

- Trajectory pre-processing, defining the procedure of transforming the original raw trajectory recordings into English sequences, dividing map into different areas and labeling the original raw trajectory into different trajectory segment;
- BERT model training, describing how the masked trajectory are processed by the deep learning model, allowing the system to learn the underlying semantics of hidden patterns;
- Outlier trajectory inference, calculating perplexity of each trajectory segment and using clustering method to divide trajectory into two groups.

1) Trajectory pre-processing: The first methodological step is represented by a process of trajectory mapping, conforming raw trajectory to an adequate input format for the neural network model. First, calculate the map size of the travel area based on the coordinates of the trajectory. Subsequently, divide the map into different coordinate blocks. Each block is labeled with a unique English word, and all coordinates falling within a specific block are associated with that unique English word. The purpose of this process is to minimize the impact of coordinate deviations on the trajectory and establish connections between different coordinates, facilitating the deep learning model's ability to learn the correlations between trajectories. The size of the divided blocks depends on factors such as the precision of the trajectory's coordinates, reliability, and the traffic conditions in the area. Once the coordinates are associated with blocks, they can be converted into English sentences for use by natural language models. Following this, based on the characteristics of the area, the region is further divided into different segments, and a secondary annotation is applied to the trajectories within each segment to indicate the type of area for that trajectory segment like Fig. 3. Dividing the region into segments adds semantic meaning to the trajectories, assisting the model in leveraging available information in cases where annotations are missing. It also equips the model with the capability to predict outliers in trajectory segments, enhancing the accuracy of predictions.

2) BERT model training: To conduct the perplexity calculation, we adopted the MLM (Masked Language Model) training approach using BERT, which is currently considered state-of-the-art for most language processing tasks. While we utilize the same internal architecture and training process, it's essential to note that our model is trained entirely from scratch on the preprocessed trajectories that have been semantically

AA	AB	AC	AD	AE	AF	AG	AH
AI	AJ	AK	AL	AM	AN	AO	AP
AQ	AR	AS	AT	AU	AV	AW	AX
AY	AZ	BA	BB	BC	BD	BE	BF
BG	BH	BI	BJ	BK	BL	BM	BN
BO	BP	BQ	BR	BS	BT	BU	BV
BW	BX	BY	BZ	CA	CB	CC	CD
CE	CF	CG	CH	CI	CJ	CK	CL

Fig. 3. Coordinates mapping and semantic mapping

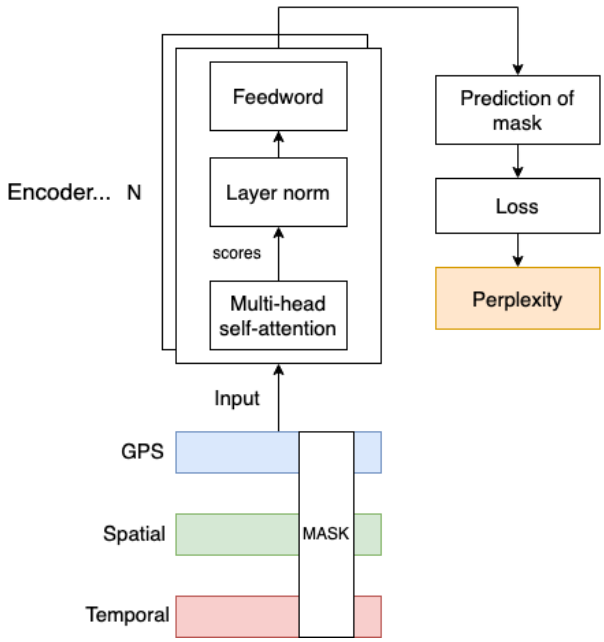

Fig. 4. Masked language modeling of BERT

augmented. An illustrative representation of this process can be found in Fig 4.

The MLM technique involves providing BERT with a partially masked sequence and subsequently optimizing its weights to accurately reveal the masked elements within that sequence as the output. The BERT architecture enables bidirectional learning, allowing the model to understand the context of each element in the sequence by considering the elements that appear both before and after it. As a result, our model utilizes the complete context within the trajectory to predict the masked location, taking into account both the previous and subsequent locations simultaneously.

Similar to the original BERT, which learns linguistic patterns through contextual word occurrences in sentences, our model aims to capture motion patterns by analyzing spatial relationships along trajectories.

3) Outlier trajectory inference: The inference phase encompasses the generation and assessment of results, aimed at gauging the model's ability to generalize after the training and optimization of weights. The fundamental concept involves introducing novel trajectory sequences not encountered during training to explore the perplexity offered by the model. These perplexity scores are then leveraged through a clustering method.

For example, consider an input sequence like [CE CF CG CH, CI CA BS BK, BC AU AM AN AO AG AH]. The objective here is to uncover the perplexity associated with each trajectory segment, with an average perplexity score of around 10. This score indicates a relatively low level of surprise on the part of the model when presented with this combination of English characters.

Contrast this with another sequence, [CE CF CG BY BX BP BH, AZ BA BB, BJ, BK, BC AU AM AF AG AH], where the first four segments exhibit an average perplexity score of 150, significantly higher than the previous example. Following the clustering process, the segment with the higher perplexity score is classified as an outlier trajectory, while the lower-scoring segments are considered inner components. The success of this process hinges on a thorough understanding of the underlying sequential motion patterns across the domain.

E. Using K-Means Clustering and BART model

we present a novel approach for detecting outliers in textual data by synergistically combining k-means clustering with Facebook's Bidirectional and Auto-Regressive Transformers (BART) model. Our method leverages the strengths of BART in generating semantically coherent and contextually appropriate textual representations, and k-means for partitioning the data into distinct clusters.

k-means algorithm with a Large Language Model (LLM) like BERT, GPT, or BART for outlier detection, the math primarily involves representations of data in the high-dimensional space defined by the embeddings from the LLM. **Objective:** To identify and separate outlier sentences from a corpus by leveraging semantic representations obtained from a Large Language Model (LLM).

Notations:

- $S_1, S_2, \ldots, S_n$: Sentences in the corpus.
- $E(S_1), E(S_2), \ldots, E(S_n)$: Embeddings of the sentences, obtained from an LLM (e.g., BART).
- C_1, C_2: Clusters, where C_1 represents the main cluster and C_2 represents the outlier cluster.
- μ_1, μ_2: Centroids of the clusters.
- k: Number of clusters, which is 2 in this case (1 for main data and 1 for outliers).

Steps: label=0.

1) *Embedding:* Convert each sentence S_i into a high-dimensional vector $E(S_i)$ using the LLM.
2) *Initialization:* Choose initial centroids μ_1 and μ_2.
3) *Assignment:* Assign each embedding $E(S_i)$ to the nearest centroid μ_j, based on some distance metric (usually Euclidean or cosine distance):

$$C_j = \{E(S_i) : \|E(S_i) - \mu_j\|^2 \leq \|E(S_i) - \mu_l\|^2 \, \forall l\}$$

4) *Update:* Calculate the new centroids by computing the mean of all the embeddings assigned to each centroid:

$$\mu_j = \frac{1}{|C_j|} \sum_{E(S_i) \in C_j} E(S_i)$$

5) *Repeat:* Repeat the assignment and update steps until the centroids no longer change significantly or until a maximum number of iterations is reached.

Objective Function:
The k-means algorithm aims to minimize the within-cluster sum of squares (WCSS) in the embedding space:

$$J(C_1, C_2; \mu_1, \mu_2) = \sum_{j=1}^{2} \sum_{E(S_i) \in C_j} \|E(S_i) - \mu_j\|^2$$

Remarks:

- Since the embeddings capture semantic information, outliers detected in this space are semantically different from the main cluster.
- The choice of k is crucial. For outlier detection, k is generally set to 2.
- Multiple runs with different initial centroids can provide more robust results.

Pseudocode:

```
1. For each sentence S_i:
   - Compute embedding E(S_i) using LLM
2. Initialize centroids randomly: μ_1, μ_2
3. Repeat until convergence:
    - For each embedding E(S_i):
    - Assign E(S_i) to the cluster C_j with
        the nearest centroid μ_j
    - For each cluster C_j:
    - Update μ_j to be the mean of all embeddings
        E(S_i) in C_j}}
```

BART for Text Embedding BART is pre-trained as a denoising autoencoder by reconstructing the original sentences from their noised versions. The noising functions can include token masking, permutation, and deletion. The model is fine-tuned on the downstream task of text generation, thereby acquiring a rich understanding of the language semantics.

In our approach, we use BART to generate embeddings for the input text data. Given a dataset of textual instances, we forward each instance through the BART model and extract the embeddings from one of the intermediate layers. These embeddings are expected to capture the semantic and syntactic information of the text. K-Means Clustering for Outlier Detection

Once the embeddings are generated, we apply the k-means clustering algorithm to partition the text data into k clusters. K-means minimizes the intra-cluster distances while maximizing the inter-cluster distances, ensuring that semantically similar text instances are grouped together.

Outlier Detection Strategy

Outliers in text data are usually characterized by their semantic dissimilarity from the majority of the instances. To identify these outliers, we compute the distance of each instance from the centroid of its assigned cluster. Text instances that have a distance exceeding a predefined threshold from their respective centroids are flagged as outliers. (convert the dataset to context based data)

Given that BART embeddings capture nuanced semantic features, this approach ensures that the outliers are not just syntactically different but also semantically distant from the common themes present in the data. (draw the transformer layer including embedded layer here)

Evaluation of K-Means for Outlier Detection with LLM

Evaluating the effectiveness of the k-means algorithm for outlier detection using embeddings from a Large Language Model (LLM) like BART can be approached through a variety of metrics and validation techniques.

1. Silhouette Score:

The silhouette score is a measure of how similar an object is to its own cluster (cohesion) compared to other clusters (separation). The silhouette score ranges from -1 to 1, where a higher value indicates that the object is well matched to its own cluster and poorly matched to neighboring clusters.

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \tag{1}$$

where:

- $a(i)$: The average distance from the ith object to the other objects in the same cluster.
- $b(i)$: The minimum average distance from the ith object to objects in a different cluster.

2. Davies-Bouldin Index:

It is a metric that indicates the average similarity ratio of each cluster with its most similar cluster. Lower values of the Davies-Bouldin index indicate better partitioning.

$$DB = \frac{1}{k} \sum_{i=1}^{k} \max_{i \neq j} \left(\frac{S_i + S_j}{d_{ij}} \right) \tag{2}$$

where:

- S_i: The average distance of all elements in cluster i to the centroid of i.
- d_{ij}: The distance between cluster centroids i and j.

3. Intra-Cluster Distance and Inter-Cluster Distance:

For a good clustering, the intra-cluster distance (distance between points within the same cluster) should be low, and the inter-cluster distance (distance between points from different clusters) should be high.

- **Intra-Cluster Distance:**

$$\sum_{i=1}^{k} \sum_{x,y \in C_i} \|x - y\|^2 \tag{3}$$

- **Inter-Cluster Distance:**

$$\sum_{i=1}^{k} \sum_{j=i+1}^{k} \|\mu_i - \mu_j\|^2 \tag{4}$$

4. Visual Inspection:

Visualizing the clusters in a reduced dimensional space (using techniques like t-SNE or PCA) can provide an intuitive understanding of how well the algorithm has separated the outliers from the main data.

5. Comparison with Ground Truth:

If labeled data is available, one can calculate precision, recall, F1-score, etc., to evaluate the accuracy of the outlier detection.

6. Stability and Consistency:

Running the algorithm multiple times with different initializations and checking the consistency of the results can also be an effective evaluation strategy.

ACKNOWLEDGMENT

REFERENCES

[1] F. Meng, G. Yuan, S. Lv, Z. Wang, and S. Xia, "An overview on trajectory outlier detection," Artificial Intelligence Review, vol. 52, no. 4, pp. 2437-2456, 2019/12/01 2019, doi: 10.1007/s10462-018-9619-1.

[2] G. Yuan, S. Xia, L. Zhang, Y. Zhou, and C. Ji, "Trajectory outlier detection algorithm based on structural features," Journal of Computational Information Systems, vol. 7, no. 11, pp. 4137-4144, 2011.'

[3] L. Zhang, Z. Hu, and G. Yang, "Trajectory outlier detection based on multi-factors," IEICE TRANSACTIONS on Information and Systems, vol. 97, no. 8, pp. 2170-2173, 2014.

[4] N. Shahid, I. Naqvi, and S. Qaisar, "Characteristics and classification of outlier detection techniques for wireless sensor networks in harsh environments: A survey," Artificial Intelligence Review, vol. 43, 02/01 2012, doi: 10.1007/s10462-012-9370-y.

[5] S. Kokkula and N. M. Musti, "Classification and outlier detection based on topic-based pattern synthesis," in Machine Learning and Data Mining in Pattern Recognition: 9th International Conference, MLDM 2013, New York, NY, USA, July 19-25, 2013. Proceedings 9, 2013: Springer, pp. 99-114.

[6] R. R. Sillito and R. B. Fisher, "Semi-supervised Learning for Anomalous Trajectory Detection," in BMVC, 2008, vol. 1, pp. 035-1.

[7] Huang, M., Zhu, M., Xiao, Y. and Liu, Y., 2021. Bayonet-corpus: a trajectory prediction method based on bayonet context and bidirectional GRU. Digital Communications and Networks, 7(1), pp.72-81.

[8] Gholipour, M., Aghagolzadeh, A. and Vahidi, J., Behavior Detection by Trajectory Analyzing Using Topic Modeling.

[9] Evermann, J., Rehse, J.R. and Fettke, P., 2017. A deep learning approach for predicting process behaviour at runtime. In Business Process Management Workshops: BPM 2016 International Workshops, Rio de Janeiro, Brazil, September 19, 2016, Revised Papers 14 (pp. 327-338). Springer International Publishing.

[10] Bando, T., Takenaka, K., Nagasaka, S. and Taniguchi, T., 2013, June. Unsupervised drive topic finding from driving behavioral data. In 2013 IEEE Intelligent Vehicles Symposium (IV) (pp. 177-182). IEEE.

[11] Nguyen, D.D., Le Van, C. and Ali, M.I., 2018, June. Vessel trajectory prediction using sequence-to-sequence models over spatial grid. In Proceedings of the 12th ACM International Conference on Distributed and Event-Based Systems (pp. 258-261).

[12] Azzedine Boukerche, Lining Zheng, and Omar Alfandi. 2020. Outlier Detection: Methods, Models, and Clas- sification. ACM Comput. Surv. 53, 3, Article 55 (June 2020), 37 pages. https://doi.org/10.1145/3381028

[13] Pakize Taylan, Fatma Yerlikaya-Özkurt, Burcu Bilgiç Uçak Gerhard-Wilhelm Weber (2021) A new outlier detection method based on convex optimization: application to diagnosis of Parkinson's disease, Journal of Applied Statistics, 48:13-15, 2421-2440, DOI: 10.1080/02664763.2020.1864815

[14] Yu Zheng. 2015. Trajectory data mining: An overview. ACM Trans. Intell. Syst. Technol. 6, 3, Article 29 (May 2015), 41 pages. DOI: http://dx.doi.org/10.1145/2743025

[15] Li, Y., Zhao, X., Zhang, Z. et al. Annotating semantic tags of locations in location-based social networks. Geoinformatica 24, 133–152 (2020). https://doi.org/10.1007/s10707-019-00367-w

[16] Khalifa, M.b., Díaz Redondo, R.P., Vilas, A.F. et al. Identifying urban crowds using geo-located Social media data: a Twitter experiment in New York City. J Intell Inf Syst 48, 287–308 (2017). https://doi.org/10.1007/s10844-016-0411-x

[17] Duraj, Agnieszka, Adam Niewiadomski, and Piotr S. Szczepaniak. "Outlier detection using linguistically quantified statements." International Journal of Intelligent Systems 33.9 (2018): 1858-1868.

[18] Mashaal Musleh, Mohamed F. Mokbel, and Sofiane Abbar. 2022. Let's Speak Trajectories. In The 30th International Conference on Advances in Geographic Information Systems (SIGSPATIAL '22), November 1–4, 2022, Seattle, WA, USA. ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3557915.3560972

[19] Elhafsi, Amine, et al. "Semantic Anomaly Detection with Large Language Models." arXiv preprint arXiv:2305.11307 (2023).

[20] Ahmed, U., Srivastava, G., Djenouri, Y., & Lin, J. C.-W. (2022). Knowledge graph based trajectory outlier detection in sustainable smart cities. *Sustainable Cities and Society*, *78*, 103580. https://doi.org/10.1016/j.scs.2021.103580

[21] Y. Su, D. Yao and J. Bi, "Few-shot learning for trajectory outlier detection with only normal trajectories," 2023 International Joint Conference on Neural Networks (IJCNN), Gold Coast, Australia, 2023, pp. 1-8, doi: 10.1109/IJCNN54540.2023.10191398.

[22] Belhadi, A., Djenouri, Y., Srivastava, G., Djenouri, D., Lin, J. C.-W., & Fortino, G. (2021). Deep Learning for Pedestrian Collective Behavior Analysis in smart cities: A model of group trajectory outlier detection. *Information Fusion*, *65*, 13–20. https://doi.org/10.1016/j.inffus.2020.08.003

[23] Jansen, T., Tong, Y., Zevallos, V., & Suarez, P. O. (2022b, December 20). *Perplexed by quality: A perplexity-based method for adult and harmful content detection in multilingual heterogeneous web data.* arXiv.org. https://arxiv.org/abs/2212.10440

[24] Graham Todd, Catalin Voss, and Jenny Hong. 2020. Unsupervised Anomaly Detection in Parole Hearings using Language Models. In Proceedings of the Fourth Workshop on Natural Language Processing and Computational Social Science, pages 66–71, Online. Association for Computational Linguistics.

[25] Sunghwan Mac Kim, S. W., Wan, S., Paris, C., &; Duenser, A. (2023, April 26). Social media relevance filtering using perplexity-based positive-unlabelled learning. AAAI. https://aaai.org/papers/00370-social-media-relevance-filtering-using-perplexity-based-positive-unlabelled-learning/

[26] Belhadi, A., Djenouri, Y., Srivastava, G., Cano, A. and Lin, J.C.W., 2021. Hybrid group anomaly detection for sequence data: Application to trajectory data analytics. IEEE Transactions on Intelligent Transportation Systems, 23(7), pp.9346-9357./

[27] Hawkins, D.M. Identification of Outliers; Springer: Dordrecht, The Netherlands, 1980; Volume 11.

[28] Dodge, S., Laube, P., Weibel, R., 2012. Movement similarity assessment using symbolic representation of trajectories. International Journal of Geographical Information Science 26 (9), 1563–1588.

[29] Ibrahim, A., et al., 2016. Linking movement and environmental data: The need for representation. Int. J. Appl. Ea

[30] Johnson, D. A. and Trivedi, M. M., 2011. Driving style recognition using a smartphone as a sensor platform. In: 2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC), 1609-1615.

[31] Buchin, M., Dodge, S., Speckmann, B., 2014. Similarity of trajectories taking into account geographic context. Journal of Spatial Information Science 9, 101–124.

[32] Elsner, J. B., Elsner, G. J. B. and Kara, A. B., 1999. Hurricanes of the North Atlantic: Climate and society. Oxford University Press on Demand.

[33] Suzuki, N., et al., 2007. Learning motion patterns and anomaly detection by human trajectory analysis. In: 2007 IEEE International Conference on Systems, Man and Cybernetics, 498-503.

[34] Pang, G., et al., 2021. Deep learning for anomaly detection: A review. ACM Comput. Surv. 54 (2), 1–38.

[35] Huang, S., Feng, Y., Liu, H. (2021). A data-driven method for falsified vehicle trajectory identification by anomaly detection. Transportation Research Part C: Emerging Technologies, 128, 103196. https://doi.org/10.1016/j.trc.2021.103196

[36] Ahmed, U., Srivastava, G., Djenouri, Y., Lin, J. C. (2022). Knowledge graph based trajectory outlier detection in sustainable smart cities. Sustainable Cities and Society, 78, 103580. https://doi.org/10.1016/j.scs.2021.103580

[37] Belhadi, A., Djenouri, Y., Srivastava, G., Djenouri, D., Lin, J. C., Fortino, G. (2021). Deep learning for pedestrian collective behavior analysis in smart cities: A model of group trajectory outlier detection. Information Fusion, 65, 13–20. https://doi.org/10.1016/j.inffus.2020.08.003

[38] C. C. Aggarwal, An introduction to outlier analysis. Springer, 2017.

[39] E. O. Eldawy, E. O. Eldawy, M. Abdalla, A. Hendawi, and H. M. O. Mokhtar, "Spatio-Temporal Outlier Detection."

[40] L. Zhang, Z. Hu, and G. Yang, "Trajectory outlier detection based on multi-factors," IEICE TRANSACTIONS on Information and Systems, vol. 97, no. 8, pp. 2170-2173, 2014.

[41] D. Kumar, J. C. Bezdek, S. Rajasegarar, C. Leckie, and M. Palaniswami, "A visual-numeric approach to clustering and anomaly detection for trajectory data," The Visual Computer, vol. 33, pp. 265-281, 2017.

[42] F. Meng, G. Yuan, S. Lv, Z. Wang, and S. Xia, "An overview on trajectory outlier detection," Artificial Intelligence Review, vol. 52, no. 4, pp. 2437-2456, 2019/12/01 2019, doi: 10.1007/s10462-018-9619-1.

[43] J. D. Mazimpaka and S. Timpf, "Trajectory data mining: A review of methods and applications," Journal of spatial information science, vol. 2016, no. 13, pp. 61-99, 2016.

[44] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: identifying density-based local outliers," in Proceedings of the 2000 ACM SIGMOD international conference on Management of data, 2000, pp. 93-104.

[45] E. M. Knorr, R. T. Ng, and V. Tucakov, "Distance-based outliers: algorithms and applications," The VLDB Journal, vol. 8, no. 3, pp. 237-253, 2000.

[46] J.-G. Lee, J. Han, and X. Li, "Trajectory outlier detection: A partition-and-detect framework," in 2008 IEEE 24th International Conference on Data Engineering, 2008: IEEE, pp. 140-149.

[47] A. Kut and D. Birant, "Spatio-temporal outlier detection in large databases," Journal of computing and information technology, vol. 14, no. 4, pp. 291-297, 2006.

[48] A. Likas, N. Vlassis, and J. J. Verbeek, "The global k-means clustering algorithm," Pattern Recognition, vol. 36, no. 2, pp. 451-461, 2003/02/01/ 2003, doi: https://doi.org/10.1016/S0031-3203(02)00060-2.

[49] T. Cheng and Z. Li, "A multiscale approach for spatio-temporal outlier detection," Transactions in GIS, vol. 10, no. 2, pp. 253-263, 2006.

[50] D. Zhang, N. Li, Z.-H. Zhou, C. Chen, L. Sun, and S. Li, "iBAT: detecting anomalous taxi trajectories from GPS traces," in Proceedings of the 13th international conference on Ubiquitous computing, 2011, pp. 99-108.

[51] X. Li, J. Han, S. Kim, and H. Gonzalez, "Roam: Rule-and motif-based anomaly detection in massive moving object data sets," in Proceedings of the 2007 SIAM International Conference on Data Mining, 2007: SIAM, pp. 273-284.

[52] R. R. Sillito and R. B. Fisher, "Semi-supervised Learning for Anomalous Trajectory Detection," in BMVC, 2008, vol. 1, pp. 035-1.

[53] J. P. Rogers, D. Barbara, and C. Domeniconi, "Detecting spatio-temporal outliers with kernels and statistical testing," in 2009 17th International Conference on Geoinformatics, 2009: IEEE, pp. 1-6.

[54] A. Albanese, S. K. Pal, and A. Petrosino, "A rough set approach to spatio-temporal outlier detection," in Fuzzy Logic and Applications: 9th International Workshop, WILF 2011, Trani, Italy, August 29-31, 2011. Proceedings 9, 2011: Springer, pp. 67-74.

[55] Bhattarai, Bimal, Ole-Christoffer Granmo, and Lei Jiao. "Measuring the novelty of natural language text using the conjunctive clauses of a tsetlin machine text classifier." arXiv preprint arXiv:2011.08755 (2020).

[56] Lingyu Li, Tianyu, Huang Yihao Li, Peng Li.Trajectory-BERT: Pre-training and fine-tuning bidirectional transformers for crowd trajectory. Beijing 100081, China. DOI: 10.1002/cav.2190

[57] Hugging Face. (Year Unknown). Evaluating language models using perplexity. Hugging Face Transformers. Available: https://huggingface.co/docs/transformers/perplexity

[58] Yang, J., Tan, X., Rahardja, S. (2022). MiPo: How to Detect Trajectory Outliers with Tabular Outlier Detectors. Remote Sensing, 14(21), 5394.

[59] Malhotra, P., Vig, L., Shroff, G., Agarwal, P. (2015, April). Long Short Term Memory Networks for Anomaly Detection in Time Series. In Esann (Vol. 2015, p. 89).

[60] Hayat Sultan Mohammed, Mario A. Nascimento, and Denilson Barbosa. 2023. Using Simple Language Models for Trajectory Im- putation. In Proceedings of 18th International Symposium on Spatial and Temporal Data (SSTD 2023). ACM, New York, NY, USA, 4 pages.