

Back to Bayesics: Uncovering Human Mobility Distributions and Anomalies with an Integrated Statistical and Neural Framework

Minxuan Duan
Novateur Research Solutions
Ashburn, Virginia, USA
mduan@novateur.ai

Yinlong Qian
Novateur Research Solutions
Ashburn, Virginia, USA
yqian@novateur.ai

Lingyi Zhao
Novateur Research Solutions
Ashburn, Virginia, USA
lzhao@novateur.ai

Zihao Zhou
University of California, San Diego
La Jolla, California, USA
ziz244@ucsd.edu

Zeeshan Rasheed
Novateur Research Solutions
Ashburn, Virginia, USA
zrasheed@novateur.ai

Rose Yu
University of California, San Diego
La Jolla, California, USA
roseyu@ucsd.edu

Khurram Shafique
Novateur Research Solutions
Ashburn, Virginia, USA
kshafique@novateur.ai

Abstract

Existing methods for anomaly detection often fall short due to their inability to handle the complexity, heterogeneity, and high dimensionality inherent in real-world mobility data. In this paper, we propose DeepBayesic, a novel framework that integrates Bayesian principles with deep neural networks to model the underlying multivariate distributions from sparse and complex datasets. Unlike traditional models, DeepBayesic is designed to manage heterogeneous inputs, accommodating both continuous and categorical data to provide a more comprehensive understanding of mobility patterns. The framework features customized neural density estimators and hybrid architectures, allowing for flexibility in modeling diverse feature distributions and enabling the use of specialized neural networks tailored to different data types. Our approach also leverages agent embeddings for personalized anomaly detection, enhancing its ability to distinguish between normal and anomalous behaviors for individual agents. We evaluate our approach on several mobility datasets, demonstrating significant improvements over state-of-the-art anomaly detection methods. Our results indicate that incorporating personalization and advanced sequence modeling techniques can substantially enhance the ability to detect subtle and complex anomalies in spatiotemporal event sequences.

CCS Concepts

• **Information systems** → **Location based services**; • **Computing methodologies** → *Mixture models*; *Bayesian network models*; **Neural networks**.

Keywords

Spatiotemporal Modeling, Anomaly Detection, Bayesian Principle, Mixture Models

ACM Reference Format:

Minxuan Duan, Yinlong Qian, Lingyi Zhao, Zihao Zhou, Zeeshan Rasheed, Rose Yu, and Khurram Shafique. 2024. Back to Bayesics: Uncovering Human Mobility Distributions and Anomalies with an Integrated Statistical and Neural Framework. In *1st ACM SIGSPATIAL International Workshop on Geospatial Anomaly Detection (GeoAnomalies'24)*, October 29–November 1 2024, Atlanta, GA, USA. ACM, New York, NY, USA, 12 pages. <https://doi.org/10.1145/3681765.3698454>

1 Introduction

How can we gain insights from tracking the movement of human populations? The exploration of human mobility, encompassing its dynamics, causes, motivations, and limitations, has been a focus of study since at least the 19th century, with foundational contributions such as Ravenstein's laws of migration [40]. In contemporary research, this field has advanced significantly through the use of various positioning technologies, such as GPS, cellular networks, and WiFi [15, 17, 56]. These technologies enable the detailed analysis of movement patterns at both individual and societal levels, providing critical information for professionals in urban planning, transportation, and public health monitoring [5, 36].

Anomalies in mobility data — unusual or unexpected patterns of movement — can signify a range of events, from transportation disruptions to shifts in population behavior caused by emergencies, pandemics, or large public gatherings. Identifying these anomalies enables stakeholders to respond effectively, ensuring public health, optimizing infrastructure, and maintaining urban resilience.

Yet, the analysis of mobility data is fraught with challenges that complicate the detection of anomalies. Mobility data is often sparse, with gaps in coverage and missing data points, making it difficult to establish a clear baseline of normal behavior. Additionally, human movement patterns are highly complex and influenced by a myriad of factors, resulting in multivariate dependencies that traditional models struggle to capture. These complexities, coupled with

Publication rights licensed to ACM. ACM acknowledges that this contribution was authored or co-authored by an employee, contractor or affiliate of the United States government. As such, the Government retains a nonexclusive, royalty-free right to publish or reproduce this article, or to allow others to do so, for Government purposes only. Request permissions from owner/author(s).

GeoAnomalies'24, October 29–November 1 2024, Atlanta, GA, USA

© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 979-8-4007-1144-2/24/10

<https://doi.org/10.1145/3681765.3698454>

the dynamic nature of human behavior, make it difficult to distinguish between true anomalies and normal variability in the data. Consequently, traditional statistical methods and machine learning models frequently fail to detect subtle yet significant anomalies.

To effectively detect anomalies in mobility data, it is essential to first understand the underlying distribution of normal behavior. Recovering these underlying distributions is crucial because it provides a baseline against which deviations can be measured. However, this task is inherently challenging due to the intricate nature of human mobility, which is influenced by a wide array of variables such as time, agent identity (e.g., individual or vehicle IDs), points of interest, and social and environmental factors.

For instance, consider the problem of predicting traffic patterns in a large city. The movement of vehicles is not only dependent on time of day and location but also on events such as road closures, public holidays, or weather conditions. Additionally, the identity of the agent (e.g., individual or vehicle performing activities) is crucial in understanding normal versus anomalous behavior. For example, a late-night drive through a typically congested downtown area might be considered anomalous for a typical commuter but perfectly normal for a taxi driver whose work involves frequent trips at all hours. Similarly, what is considered a normal pattern of movement for a door-to-door salesperson would be highly anomalous for someone who typically works from home. These examples underscore the necessity of capturing the full complexity of mobility data, which includes understanding how the variables such as time, location, and agent ID interact to form the overall distribution of normal behavior [20].

Existing methods for recovering these underlying distributions often fall short because they struggle to manage the complexity, heterogeneity, and high dimensionality of mobility data (See Section 2 for a detailed discussion). To address these challenges, many models rely on simplifying assumptions, such as treating variables as independent. However, these assumptions rarely hold true in real-world mobility data, where variables such as time and location are deeply interdependent. As a result, these models frequently produce incomplete or inaccurate representation of the underlying distributions, overlooking critical nuances in the data.

In response to these challenges, we propose DeepBayesian, a novel framework that integrates Bayesian principles with advanced neural density estimation techniques to recover the underlying multivariate distributions from sparse and complex mobility datasets. Our approach is designed to handle the high dimensionality, heterogeneity, and interdependencies inherent in mobility data, providing a more accurate and comprehensive understanding of normal behavior. It combines the complementary strengths of Bayesian theory and neural networks: Bayesian theory offers a probabilistic framework for integrating prior knowledge and managing uncertainty, while neural networks are well-suited for capturing complex, high-dimensional relationships within the data.

DeepBayesian employs a cascade of neural density estimators to model the complex interactions between heterogeneous variables such as time, location, and agent ID. By using a Bayesian framework, the model captures the full multimodal distribution of possible outcomes, rather than relying on single-point predictions. This allows DeepBayesian to identify anomalies that are not only deviations from the expected outcome but also deviations from the entire range of normal behavior.

To personalize the model for individual agents (e.g., specific vehicles, individuals, or groups of individuals with similar socio-economic demographics), DeepBayesian incorporates agent embeddings. These embeddings capture the unique characteristics and behaviors of different agents, allowing the model to better distinguish between normal and anomalous behavior on an individual or group basis. This personalized approach enhances the model's sensitivity to subtle anomalies that might otherwise go unnoticed.

We demonstrate the effectiveness of DeepBayesian through extensive experiments on multiple mobility datasets, showing that it outperforms existing anomaly detection methods.

Our contributions can be summarized as follows:

- We introduce DeepBayesian, a novel framework that combines Bayesian theory with deep neural networks to accurately recover underlying multivariate distributions from sparse and complex mobility data.
- We implement a cascade of neural density estimators that can handle the high dimensionality and complex interdependencies in mobility data, capturing both simple and complex anomalies.
- We develop and incorporate deep agent embeddings to personalize the model, improving its ability to detect anomalies specific to individual agents.
- We demonstrate the effectiveness of DeepBayesian through extensive experiments on multiple mobility datasets, showcasing its superiority over existing baselines in anomaly detection.

The remainder of this paper is organized as follows. In Section 2, we discuss the innovations presented in this work within the context of the state of the art in anomaly detection and neural density estimation, highlighting the key advancements that distinguish our approach. Section 3 formulates the problem, providing a formal definition and setting the stage for the methodological developments that follow. Section 4 details the methodology, describing the structure and components of the proposed DeepBayesian framework, including the integration of Bayesian principles, neural density estimation, and agent embeddings. In Section 5, we present extensive experiments conducted on multiple mobility datasets to evaluate the effectiveness of our approach, comparing its performance with existing baselines. Finally, Section 6 concludes the paper, summarizing the key findings and discussing potential directions for future research.

2 Relationship with State-of-the-Art

In this section, we position our proposed framework within the broader landscape of existing research in the areas of anomaly detection (Section 2.1) and neural density estimation (Section 2.2).

2.1 Anomaly Detection

Anomaly detection in spatiotemporal data has garnered significant attention, particularly for identifying irregular patterns in user behavior. Existing approaches to anomaly detection in mobility data can be broadly categorized into clustering-based, distance-based, reconstruction-based, prediction-based, and density estimation-based methods. *Clustering-based methods* group data into clusters and flag anomalies as outliers [27, 31, 39]. These methods

often assume that normal data forms distinct clusters, an assumption that may not hold true in complex mobility data. *Distance-based methods* detect anomalies by measuring deviations from the norm [4, 21, 34, 43]. While intuitive and straightforward to implement, these approaches heavily rely on the choice of distance metric — a challenge in complex datasets where different features may have varying scales or significance. Moreover, they suffer from the curse of dimensionality, where the effectiveness of distance metrics diminishes as the number of dimensions increases, making it difficult to distinguish between normal and anomalous points. *Reconstruction-based methods* use models like deep autoencoders to reconstruct data and identify anomalies based on reconstruction errors. Although these methods have shown promise [35, 57], they are prone to overfitting, where the model may learn to perfectly reconstruct the training data but fail to generalize to new, unseen data. Additionally, they are sensitive to outliers in the training data, which can lead the model to incorrectly learn to reconstruct these anomalies as normal patterns, thereby diminishing its effectiveness in identifying true anomalies.

Prediction-based methods have been a major focus of anomaly detection research [22, 24, 29, 35, 44, 50]. These methods forecast future data points and identify anomalies as deviations from these predictions. However, existing predictive models [1, 11–14, 16, 19, 30, 32, 33, 42, 51, 52, 54, 55] have severe limitations when forecasting complex mobility patterns. Typically, these models predict only the location of visits [51, 52], which limits their utility in detecting temporal anomalies. While some recent methods have attempted to model both location and time [9, 53], they often make limiting assumptions, such as the independence between location and time, which can affect their predictive accuracy [20].

One of the major drawbacks of prediction-based models is their reliance on single-point predictions, which can overlook the multimodal nature of real-world data distributions. These models typically generate a single expected outcome, assuming that future behavior can be represented by a specific predicted value. However, in reality, the future may involve multiple plausible outcomes, each with its own probability. For example, in urban mobility, travel time might vary widely due to factors like traffic conditions or weather, leading to a multimodal distribution of possible travel times. Single-point predictions can average out these possibilities, resulting in inaccurate forecasts and the potential to miss critical anomalies that arise from these alternative scenarios.

This limitation of prediction-based models underscores the need for approaches that can capture the full distribution of possible outcomes rather than just a single expected value. *Density estimation-based methods* address this by modeling the entire probability distribution of the data and identifying anomalies as data points that fall in regions of low probability. These methods are better suited for detecting a broader range of anomalies, including those arising from complex, multimodal distributions. However, despite their advantages, density-based approaches also face challenges. Traditional methods, such as kernel density estimation (KDE), can struggle with high-dimensional data due to the curse of dimensionality, where the volume of the data space increases exponentially with the number of dimensions, making it difficult to estimate densities accurately. Moreover, density estimation methods often require large amounts of data to model the joint distribution effectively, which can be a

limitation in scenarios with sparse data or missing values. Additionally, these methods may require careful tuning to balance the trade-off between bias and variance, where overly smooth density estimates may miss subtle anomalies, while overly complex models may overfit the data.

2.2 Neural Density Estimation

Neural networks are widely recognized as universal function approximators, making them highly effective tools for modeling complex systems. For mobility data analysis, neural density estimators harness this expressive power to model sequences of multivariate distributions and stochastic processes, effectively capturing the intricate patterns of human movement in high-dimensional spaces. These estimators are especially suited for representing the temporal and spatial dependencies inherent in mobility data, where each observation can be treated as a sample from a time-varying distribution. Neural density estimators for mobility analysis can be categorized based on their approach for handling temporal dynamics and their specific techniques for modeling probability densities.

Continuous-time models offer the advantage of capturing the fluid nature of mobility patterns. *Neural Ordinary Differential Equations* (Neural ODEs) [8] model the continuous evolution of the system state over time, allowing for the hidden state formulation between two events. *Neural Stochastic Differential Equations* (Neural SDEs) [23] build on this approach by incorporating stochastic elements, which more effectively capture the inherent randomness in mobility patterns. Another continuous-time approach, is *Neural Point Processes* [37], which excels at modeling the occurrence rates of sporadic mobility events in continuous time. These models are capable of estimating temporal densities and may offer tractable likelihood computation [58], making them particularly suitable for density estimation tasks in mobility data analysis. In contrast, discrete-time models, such as Autoregressive models [38], Recurrent Neural Networks (RNNs) and Transformer-based architectures [49], operate on fixed time steps. These discrete-time approaches can be effectively combined with various density estimation techniques to provide explicit likelihood computations.

The spatial dimension of mobility data poses unique challenges for density estimation. *Normalizing flows* [28, 41] have shown promise in modeling complex spatial distributions, offering both flexible modeling capabilities and exact likelihood computation. However, they can be computationally intensive, especially in high-dimensional spaces. To address the diverse features often present in mobility data, various approaches have been proposed. *Mixture Density Networks* [7], such as a mixture of Gaussian or Laplace densities [46, 48], can be used as output layers to enable modeling of complex, multimodal distributions at each time step. *Variational Autoencoders* (VAEs) [26] learn latent representations of the data and provide a lower bound on the likelihood. Another approach, *Automatic Integration* [59], learns the density directly as a neural network, offering faster computation than normalizing flows in low dimensional settings, but it does not scale effectively to higher dimensional outputs. Recent advancements in *Diffusion Models* [18] also show promise, as they can be adapted to provide likelihood estimates through variants like Variational Diffusion [25]. The choice of model often requires balancing expressiveness against the computational tractability of likelihood estimation.

The proposed DeepBayesian framework is closely related to autoregressive models and mixture density networks discussed above, but it offers several key advantages over existing approaches. First, it excels in handling heterogeneous inputs by seamlessly integrating both continuous and categorical data. While most autoregressive models primarily focus on modeling distributions of either discrete tokens (e.g., LLMs or grid-based locations [35]) or continuous output [58], they often struggle to handle both types of data simultaneously. This limitation poses significant challenges in real-world applications, such as mobility data analysis, where categorical inputs — such as types of points of interest (POI) — provide crucial context and are essential for accurate modeling. Another critical advantage of DeepBayesian lies in its tailored modeling of diverse feature distributions. In many datasets, different features follow significantly different distributions. For example, arrival times might follow a multimodal normal mixture, while duration could adhere to a log-normal distribution with a strong left skewness. Many autoregressive models treat all features uniformly, which can lead to oversights and inaccuracies by ignoring these important distributional nuances. DeepBayesian addresses this challenge by allowing for the customization of neural density estimators tailored to the specific characteristics of each conditional distribution. Unlike existing autoregressive models, which often rely on a shared architecture across all features, DeepBayesian enables the integration of different types of neural networks, such as recurrent networks for modeling point-of-interest sequences and transformer networks for handling stay durations. This modular approach enables optimization of the architecture for the specific requirements of each data type. By customizing each estimator to capture the unique complexities of the data, DeepBayesian provides a more robust and adaptable framework for mobility data analysis.

3 Problem Statement

Our study focuses on finding anomalies in large-scale urban mobility sequences. A mobility sequence represents a series of recorded movements or activities of an agent, such as an individual or a vehicle, over time. Let P denote the total number of agents, where each agent z , indexed by $z \in \{1, \dots, P\}$, is observed through a mobility sequence denoted by $X = [X_1, \dots, X_n]$. Each observation X_i corresponds to a specific event or activity characterized by a set of attributes such as speed, location, time, and the type of visited place. Formally, each observation X_i is represented as: $X_i = (a_{i,1}, a_{i,2}, a_{i,3}, \dots, a_{i,m})$ where $a_{i,j}$ denotes the j th attribute of X_i . Anomaly detection is achieved by estimating the probability distribution of these observations and identifying instances that significantly deviate from the estimated distribution.

To estimate the probability distribution of the observation X for a given agent z , denoted as $P(X|z)$, we aim to estimate the joint distribution of the attributes that constitute X . By applying Bayesian inference and the chain rule of probability, the probability distribution $P(X|z)$ can be decomposed into a product of conditional probabilities:

$$\begin{aligned} P(X | z) &= P(a_{:,1}, a_{:,2}, \dots, a_{:,j} | z) \\ &= P(a_{:,1} | z) \times P(a_{:,2} | a_{:,1}, z) \times \dots \\ &\quad \times P(a_{:,j} | a_{:,1}, a_{:,2}, \dots, a_{:,j-1}, z) \end{aligned} \quad (1)$$

Each term in this sequence of conditional probabilities can be estimated using a suitable density estimator, enabling us to capture the complex dependencies between attributes effectively.

In this study, we concentrate on three key attributes for each observation that are particularly informative for identifying deviations from typical behavior patterns: arrival time $a_{i,1} := t_i$, stay duration $a_{i,2} := d_i$, and POI type $a_{i,3} := c_i$. Thus, each observation X_i of agent z is defined as:

$$X_i = (c_i, t_i, d_i) \quad (2)$$

For each agent z in our dataset, there is an associated mobility sequence, X_{train} , which predominantly consists of normal activities. These sequences collectively constitute the training dataset used to learn the underlying probability distributions in an unsupervised fashion. Our goal in the anomaly detection task is to assign an anomaly score s to each observation in a separate set of mobility sequences, X_{test} . According to Eq. 1, the probability distribution $P(X | z)$ can be rewritten as:

$$\begin{aligned} P(X | z) &= P(d, c, t | z) \\ &= P(t | z) P(c | t, z) P(d | c, t, z) \end{aligned} \quad (3)$$

Given an estimate of the probability distribution $P(X^{train} | z)$, the anomaly score for each observation in X_{test} can be determined by calculating $1 - P(X^{test} | z)$.

4 Method

Given an agent z , estimating the probability distribution of the associated mobility sequence X_{train} is challenging due to the typically sparse data available for individual agents. To address this challenge, we employ an agent embedding model to extract latent features from the observed activities of all agents, clustering those with similar behaviors closer together in the latent space. This approach enables us to estimate $P(X | h)$, where h is the learned agent embedding, rather than directly estimating $P(X | z)$ from sparse data. Following Eq. 3, given an agent embedding h , we estimate the probability distribution of each observed sequence, X , by sequentially estimating the conditional probability distributions of its attributes, the arrival time ($P(t | h)$), the type of point of interest ($P(c | t, h)$), and the stay duration ($P(d | c, t, h)$).

The subsequent sections will detail the process of obtaining agent representations and the implementation of the three conditional probability distributions.

4.1 Agent Embedding Auto-Encoder

Fig. 1 illustrates the architecture of the agent embedding auto-encoder. We employ a transformer-based auto-encoder, similar to MotionClip [47], to map each staypoint sequence to an agent embedding vector. This model is trained to project the staypoint sequence X of an agent z into a latent vector h (the agent embedding vector) while simultaneously reconstructing the original sequence. The details of each component of this agent embedding auto-encoder are provided in the following subsections.

4.1.1 Sequence Encoder. We begin by encoding each staypoint $X_i = (c_i, t_i, d_i)$ into an encoded space E_i to normalize the data representation. Specifically, we use one-hot encoding for the POI type and use min-max normalization for arrival time and stay duration. A learnable token sequence E_0 , sampled from standard normal

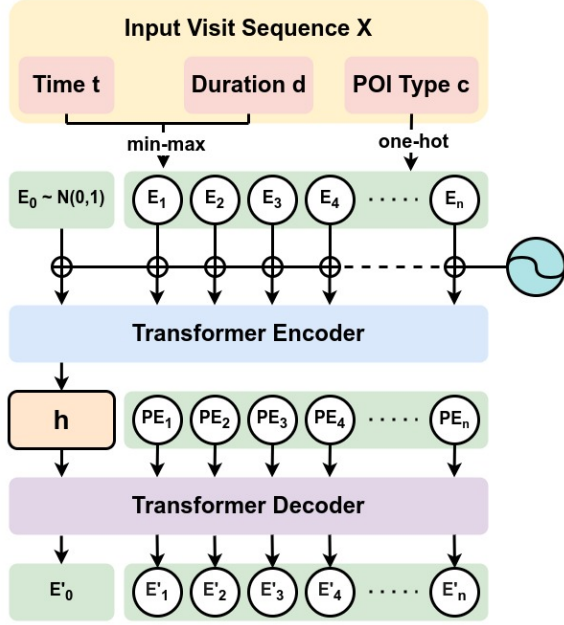


Figure 1: Agent embedding auto-encoder. A Transformer Encoder is trained to project an input feature sequence X , encoded by a Sequence Encoder and prefixed by learnable token E_0 , into its latent representation h . Simultaneously, a Transformer Decoder is trained to reconstruct the encoded sequence from the latent representation h and the standard positional encoding PE .

distribution, is inserted at the beginning of the encoded sequence, following the approach used in [47]. The input to the transformer encoder is denoted as $E = [E_0, E_1, \dots, E_n]$. Further details on the encoding process can be found in Section 5.1.2.

4.1.2 Transformer Encoder. The transformer encoder maps the encoded staypoint sequence E to a latent representation h . First, the encoded sequence is projected into the encoder’s dimension by linear projection. Next, a standard positional embedding is applied to the projected sequence. The latent representation h is then obtained as the first output of the encoder, with the remaining sequence outputs discarded.

4.1.3 Transformer Decoder. The latent representation h is fed to the transformer decoder as key and value, while the positional encoding is inputted to the transformer decoder as query sequence. Subsequently, the transformer decoder predicts an encoded staypoint sequence E' .

4.1.4 Loss Function of the Agent Embedding Auto-Encoder. The agent embedding auto-encoder is trained by minimizing the reconstruction loss, defined as the mean squared error between the input sequence E and the reconstructed output sequence E' . Specifically, the loss function is given by:

$$\mathcal{L}_{ac} = \frac{1}{n * |E|} \sum_{i=1}^n \|E_i - \hat{E}_i\|^2 \quad (4)$$

4.2 Joint Distribution Estimation

In this section, we provide a detailed explanation of the estimation process for the joint probability distribution $P(X | h)$, as outlined in Section 3. The pipeline for this process is illustrated in Fig. 2.

4.2.1 Arrival Time Probability Estimation. To estimate the probability distribution of arrival times, $\hat{P}(t|h)$, for a given agent, we employ a kernel-based Gaussian mixture model (GMM), denoted as $f_t(h)$. This model is trained on the arrival time sequence observed during the training period. The estimated probability is defined as:

$$\hat{P}(t | h) := f_t(h) \quad (5)$$

Since arrival times are inherently non-negative, a clipping function is applied to ensure that all estimated arrival times remain within valid bounds.

4.2.2 POI Type Probability Estimation. Given the agent embedding h and arrival time t , we use a Recurrent Neural Network (RNN), denoted by f_{POI} , to estimate the probability distribution over the POI type c . We chose an RNN architecture because it is particularly effective at handling sequential data and capturing temporal dependencies. After obtaining the hidden state vector $\mathbf{g} = (g_1, \dots, g_n)$ from the RNN, we apply a linear projection followed by a softmax function to \mathbf{g} to predict the probability distribution over the discrete POI types:

$$\hat{P}(c | t, h) = \text{softmax}(\mathbf{W}\mathbf{g} + \mathbf{b}) \quad (6)$$

where \mathbf{W} is the weight matrix and \mathbf{b} is the bias vector of the linear projection layer. The softmax function is defined as:

$$\text{softmax}(y_i) = \frac{\exp(y_i)}{\sum_{j=1}^K \exp(y_j)}$$

where y_i is the i -th element of the output vector, and K is the number of POI types.

4.2.3 Stay Duration Probability Estimation. Given the agent embedding h , arrival time t , and POI type c , we use a transformer-based neural density function, f_d , to model the probability distribution of stay duration d . As discussed in Section 2.2, a neural density function f is a neural network that implicitly parameterizes the variables to be estimated. This approach is especially well-suited for handling high-dimensional datasets with sparse samples, a common characteristic of activity sequence data. When combined with Bayesian inference, it offers a robust framework for incorporating and updating prior knowledge as more data attributes are observed.

To model the distribution of stay duration, $\hat{P}(d | c, t, h)$, we represent it using a Gaussian Mixture Model (GMM), which can be expressed as:

$$\hat{P}(d | c, t, h) := \sum_k m_k p_{\mathcal{N}}(d; \mu_k, \sigma_k) \quad (7)$$

where:

$$p_{\mathcal{N}}(d; \mu_k, \sigma_k) = \frac{1}{\sigma_k \sqrt{2\pi}} \exp\left(-\frac{(d - \mu_k)^2}{2\sigma_k^2}\right) \quad (8)$$

The mixture weights \mathbf{m} are estimated by the neural density function:

$$f_d : c, t, h \rightarrow \mathbf{m} \quad (9)$$

A softmax function is applied to ensure that the mixture weights \mathbf{m} sum to 1. For a given observation of stay duration d , the likelihood

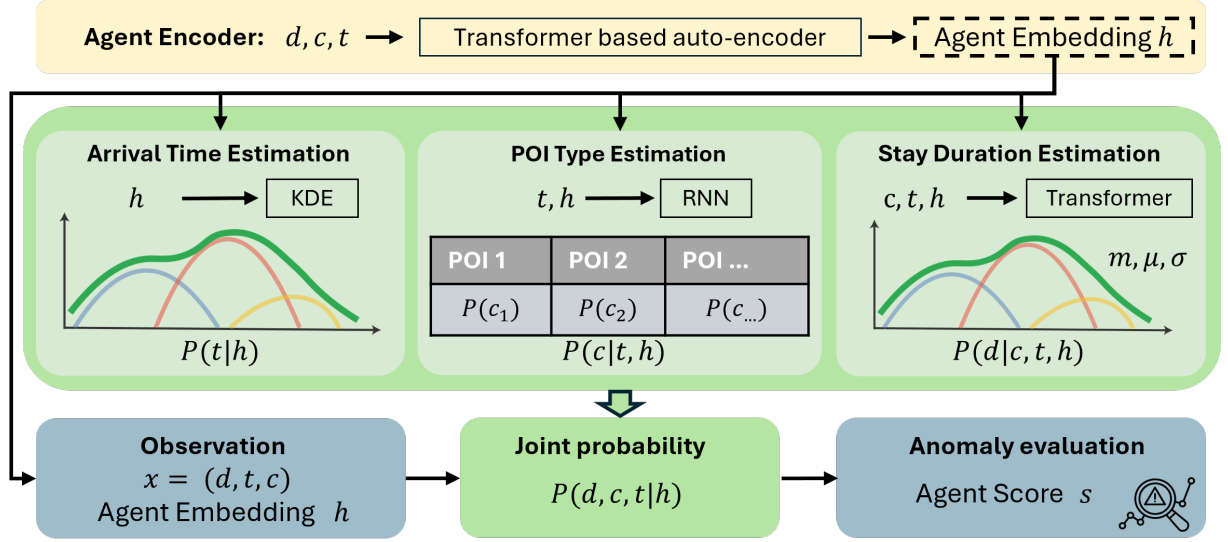


Figure 2: DeepBayesian pipeline. The agent embedding h is fed into three modules: Arrival Time Estimation, POI Type Estimation, and Stay Duration Estimation. Each module uses h along with other relevant inputs to estimate the corresponding conditional probability distributions, which are then integrated into a joint probability model. Finally, the agent embedding h and an observation x are input into the joint probability model to calculate anomaly score s .

under the model p_{GM} is maximized by minimizing its negative log-likelihood, forming the basis for the training loss function for f_d :

$$\mathcal{L}_{f_d}(d) = -\log \left[\sum_k m_k \mathcal{P}_{\mathcal{N}}(d; \mu_k, \sigma_k) \right] \quad (10)$$

The transformer-based neural density function facilitates attention mechanisms across h , t , and c , allowing for an adaptive representation that captures complex dependencies between these variables.

4.2.4 Neural Density Function Loss. Using the chain rule of probability (Eq. 3), the loss function for training the joint distribution estimation function is defined as:

$$\mathcal{L}_f = - \sum_{c,d} [\log \hat{P}(c | t, h) + \log \hat{P}(d | c, t, h)] \quad (11)$$

4.2.5 Total loss. The total loss function is obtained by combining the agent embedding autoencoder loss (Eq. 4) with the neural density function loss (Eq. 11). The total loss is:

$$\begin{aligned} \mathcal{L}_{\text{total}} &= \mathcal{L}_{\text{ae}} + \mathcal{L}_f \\ &= \frac{1}{n * |E|} \sum_{i=1}^n \|E_i - \hat{E}_i\|^2 - \sum_{c,d} [\log \hat{P}(c | t, z) + \log \hat{P}(d | c, t, z)] \end{aligned} \quad (12)$$

4.3 Anomaly Score Assignment

Given the agent embedding h , the inferred staypoint sequence X^{infer} , and probability estimates $\hat{P}(t | h)$, $\hat{P}(c | t, h)$, and $\hat{P}(d | c, t, h)$, we first compute the joint probability $\hat{P}(d^{\text{infer}}, c^{\text{infer}}, t^{\text{infer}} | h)$ as follows:

$$\begin{aligned} \hat{P}(d, c, t | h) &= \text{CLIP}_{\text{arrival_time}}(\hat{P}(t | h)) \times \\ &\quad \text{CLIP}_{\text{POI_type}}(\hat{P}(c | t, h)) \times \\ &\quad \text{CLIP}_{\text{duration}}(\hat{P}(d | c, t, h)) \end{aligned} \quad (13)$$

where CLIP denotes a clipping function applied to ensure valid values. The anomaly score s is then computed as:

$$s = 1 - \hat{P}(d, c, t | h). \quad (14)$$

5 Experiments and Results

In this section, we present the experiments conducted to evaluate the performance of the proposed DeepBayesian framework. Our experiments (Section 5.1) are designed to assess the model's effectiveness in anomaly detection (Section 5.2), its ability to personalize predictions for individual agents (Section 5.3), and the impact of various model components on overall performance (Section 5.4).

5.1 Experimental Setup

5.1.1 Datasets. To evaluate the performance of our proposed approach, we conducted experiments using two synthetic mobility datasets, NUMOSIM-LA [45] and Urban Anomalies [2, 3] that simulate human movement patterns in urban environments.

NUMOSIM-LA¹: NUMOSIM is a synthetic dataset designed to simulate human mobility patterns within urban environments [45]. The initial release, NUMOSIM-LA, focuses on the Los Angeles area and is divided into four weeks for training and four weeks for testing. It contains data for 200,000 agents, of which 381 exhibit various types of anomalous behaviors during the test period. These

¹The NUMOSIM-LA dataset can be accessed at: <https://osf.io/sjyfr/>.

behaviors range from isolated events that disrupt an agent’s typical sequence of activities to recurring patterns that deviate from the norm regularly. At the staypoint level, the dataset comprise a total of 16,667,273 staypoints, with 3,468 labeled as anomalous. The anomaly prevalence rates are 0.19% at the agent level and 0.0208% at the staypoint level. This distribution reflects the significant imbalance typically encountered in real-world anomaly detection problems. Additionally, NUMOSIM-LA provides several benchmarks from state-of-the-art methods, making it a valuable resource for evaluating the effectiveness of the DeepBayesic model against current standards.

Urban Anomalies²: The Urban Anomalies dataset [2, 3] includes synthetic simulations of urban environments in Atlanta and Berlin, each containing mobility sequences for 1000 agents, some of whom are affected by a simulated disease that causes them to eat more frequently, reduce social interactions, and stop going to work. The anomaly prevalence rates are 12% at the agent level and 12.3% at the staypoint level. Each mobility sequence includes one month of normal activities followed by a period of variable length with injected anomalies. The anomalies fall into four categories: i) *Hunger*: Agents visit restaurants more frequently, ii) *Social*: Agents visit random locations instead of meeting friends, iii) *Work*: Agents stop going to work, and iv) *Combined*: A mix of all the above anomalies. The dataset provides comprehensive information, including agent trajectories, staypoints, and social links, offering a rich set of features for evaluating anomaly detection methods.

5.1.2 Preprocessing. For each dataset, we computed the following features for each staypoint:

- Time: Epoch time (day/hour)
- Location: Latitude and longitude
- Duration: Length of stay
- POI type: Type of point of interest

Epoch time was normalized to center around Monday, combining both day and hour values. Urban Anomalies dataset directly provides POI type as an attribute. For NUMOSIM-LA, we determined POI type of each stay point by mapping it to the nearest available POI within a 15-meter radius, or labeling it as ‘unknown’ when no POI was identified within this distance. Each baseline model was adapted to support the NUMOSIM and Urban Anomalies datasets (See [45] for details of these modified pipelines).

5.1.3 Baseline. We evaluate the performance of our anomaly detection pipeline against the following baseline methods:

RioBusData [6]: A convolutional neural network designed to detect outlier trajectories in the bus routes of Rio de Janeiro. The input sequence consists of agent IDs and all the features listed in Section 5.1.2. Location feature was represented by raw geographic latitudes and longitudes. The model’s output was modified to predict anomalies in agent behaviors instead of bus routes.

Spatial-Temporal Outlier Detector (STOD) [10]: A GRU-based neural network that detects anomalies in bus trajectories using GPS points from regular bus routes. This model utilizes all the features in Section 5.1.2. An embedding layer was used to encode latitudes and longitudes into tokens. The H3 resolution used in Geo embedding creation was set to 12.

Gaussian Mixture Variational Sequence AutoEncoder (GM-VSAE) [35]: A VAE-based model designed to detect trajectory anomalies through trajectory generation. The latitudes and longitudes were converted into grid indices, and were used as the only input to the model. This model is limited to detecting anomalies at the agent level, without providing insights into anomalies at the stay-point level.

5.1.4 Metrics. To evaluate the performance of our approach, we use the following metrics:

- Area Under the Precision-Recall Curve (AUPR)
- Area Under the Receiver Operating Characteristic curve (AU-ROC)
- Maximum F1-Score
- Average Precision (AP)

These standard metrics allow us to assess the model’s performance across all possible decision thresholds, providing a comprehensive evaluation of anomaly detection, especially in highly imbalanced datasets where fixed thresholds may not generalize well.

5.2 Anomaly Detection Results

Our evaluation focuses on detecting anomalies at both the staypoint and agent levels across the NUMOSIM-LA, Urban Anomalies-Berlin, and Urban Anomalies-Atlanta datasets. For agent-level anomaly detection, anomaly scores are computed by taking the maximum score among all associated staypoints for each agent. This approach effectively highlights the most anomalous behavior within each agent’s trajectory, enabling more accurate identification of significant deviations.

We compared the performance of our anomaly detection pipeline against all baseline methods. As shown in Table 1 and the ROC curves in Figure 3, our method, DeepBayesic, consistently outperforms the baselines across multiple metrics.

Agent-Level Anomaly Detection: The ROC curves for agent-level detection, shown in Figure 3 (a-c), reveal a clear improvement of our method over the baselines, especially on the NUMOSIM-LA dataset. The curves indicate that our approach effectively identifies a higher number of true positives while maintaining a low rate of false positives. This superior performance can be attributed, in part, to the integration of personalized agent embeddings, which enable our model to more accurately capture individual behavioral patterns, even with sparse data points. We further explore this hypothesis in an ablation study presented in Section (5.4).

Staypoint Level Anomaly Detection: The ROC curves for staypoint-level anomaly detection, depicted in Figure 3 (d-f), demonstrate that while our method still outperforms the baselines, the overall scores are smaller compared to agent-level detection. This is due to the inherent challenges in detecting anomalies at a finer granularity, where the model must identify deviations in behavior at the level of individual staypoints. Additionally, since agent-level anomaly detection aggregates results from staypoint-level detections, it can mitigate the impact of false positives at the staypoint level.

Max F1-Score and Precision: As shown in Table 1 DeepBayesic consistently achieves the highest AUPR and AUROC scores across the datasets. Notably, on the NUMOSIM-LA dataset, the AUPR scores for the RIOBus and STOD baselines near zero, and their

²The Urban Anomalies dataset can be accessed at: <https://osf.io/dg6t3/>.

Table 1: Performance of Our Method and Baseline Methods on NUMOSIM-LA, Urban Anomalies-Berlin, and Urban Anomalies-Atlanta Datasets. The performance metrics (AUPR, AUROC, Precision, and Max F1-Score) are reported for both agent-level and staypoint-level anomaly detection tasks, with the agent-level metrics listed first, followed by the staypoint-level metrics, separated by a ‘/’ within each cell. For GM-VSAE, only agent-level metrics are available.

Method	Dataset	AUPR	AUROC	Precision	Max F1-Score
DeepBayesic	NUMOSIM-LA	1.21% / 0.42%	72.58% / 65.98%	1.91e-03 / 2.11e-04	5.23e-02 / 3.74e-02
	Urban Anomalies-Berlin	16.54% / 17.71%	60.58% / 55.93%	1.20e-01 / 1.28e-01	2.52e-01 / 2.25e-01
	Urban Anomalies-Atlanta	16.12% / 16.70%	60.04% / 54.64%	1.20e-01 / 1.26e-01	2.52e-01 / 2.23e-01
RIO Bus	NUMOSIM-LA	0.16% / 0.02%	50.17% / 50.55%	1.64e-03 / 1.89e-04	3.29e-03 / 3.82e-04
	Urban Anomalies-Berlin	11.91% / 12.36%	49.81% / 50.75%	1.20e-01 / 1.23e-01	2.17e-01 / 2.21e-01
	Urban Anomalies-Atlanta	13.61% / 14.87%	55.36% / 55.31%	1.20e-01 / 1.25e-01	2.34e-01 / 2.22e-01
STOD	NUMOSIM-LA	0.19% / 0.03%	51.81% / 59.44%	1.82e-03 / 2.41e-04	4.08e-03 / 3.83e-03
	Urban Anomalies-Berlin	14.64% / 13.98%	52.11% / 53.29%	1.16e-01 / 1.22e-01	2.24e-01 / 2.32e-01
	Urban Anomalies-Atlanta	13.02% / 13.98%	52.25% / 53.18%	1.28e-01 / 1.37e-01	2.19e-01 / 2.29e-01
GM-VSAE	NUMOSIM-LA	0.19%	50.66%	1.62e-03	4.36e-03
	Urban Anomalies-Berlin	11.28%	47.29%	9.68e-02	2.18e-01
	Urban Anomalies-Atlanta	11.91%	50.14%	1.33e-01	2.20e-01

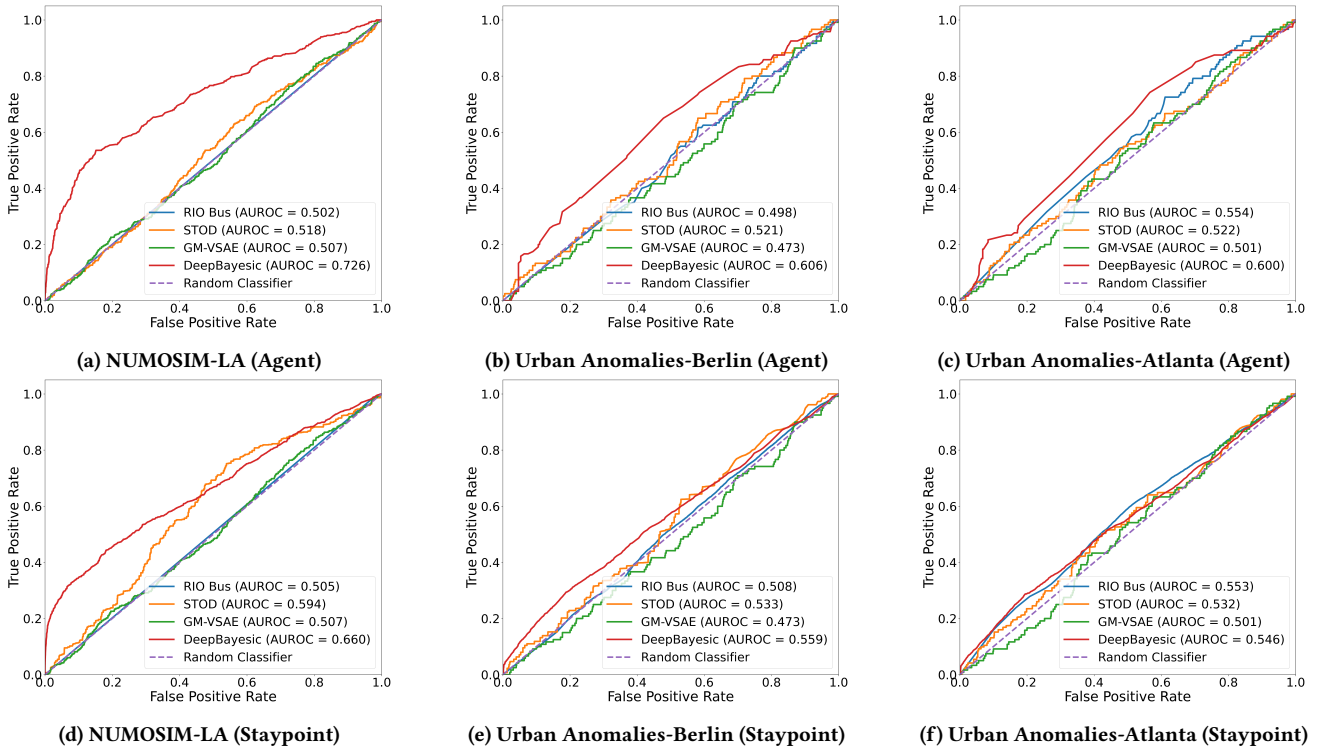


Figure 3: ROC curves for anomaly detection performance across different datasets and levels of granularity. The top row (a-c) shows the ROC curves for agent-level anomaly detection, while the bottom row (d-f) shows the ROC curves for staypoint-level anomaly detection. Our method, DeepBayesic, is represented in red.

AUROC scores hover around 50%, suggesting performance close to that of a random classifier. In contrast, DeepBayesic achieves significantly better results.

Furthermore, DeepBayesic also obtains the highest Max F1-scores at both the agent and staypoint level on the NUMOSIM-LA dataset, while RIOBus and STOD record much lower Max-F1 scores.

Although F1-Score and precision are less informative in highly imbalanced datasets such as NUMOSIM-LA and Outliers-Berlin, our model’s performance in these metrics indicates that it outperforms the baseline models.

Overall, our model demonstrates strong performance, especially at the agent level, where the use of personalized embeddings and

the DeepBayesic framework enables it to detect subtle anomalies effectively. Our method also remains competitive to the baseline at the staypoint level.

To better understand the reasons behind DeepBayesic’s performance, we further analyzed the prediction outcomes across different agents in Section 5.3 and conducted a comprehensive ablation study in Section 5.4 to examine the contributions of each module to the overall performance .

5.3 Impact of Personalization on Prediction

Personalization plays a central role in enhancing the prediction accuracy of our model. By incorporating personalized agent embeddings, the model captures the unique behavioral patterns of individual agents and tailors its predictions accordingly, thereby improving the detection of subtle anomalies that may be overlooked by a generalized model.

To assess the effectiveness of personalization, we visualize the predicted conditional distributions of both arrival times and durations for multiple agents. These visualizations, presented in Figures 4a and 4b, highlight the differences in density patterns among agents.

Figure 4a shows the predicted arrival time distributions for three distinct agents. The personalized embeddings capture unique patterns for each agent: Agent 1 (blue) exhibits a bimodal distribution with peaks around 08:00 and 14:00, suggesting a student with morning and afternoon classes; Agent 2 (orange) displays a strong peak at 09:00 and a smaller peak at 18:00, indicative of a typical work schedule; Agent 3 (yellow) shows a more uniform distribution throughout the day, representing an agent with less structured routine.

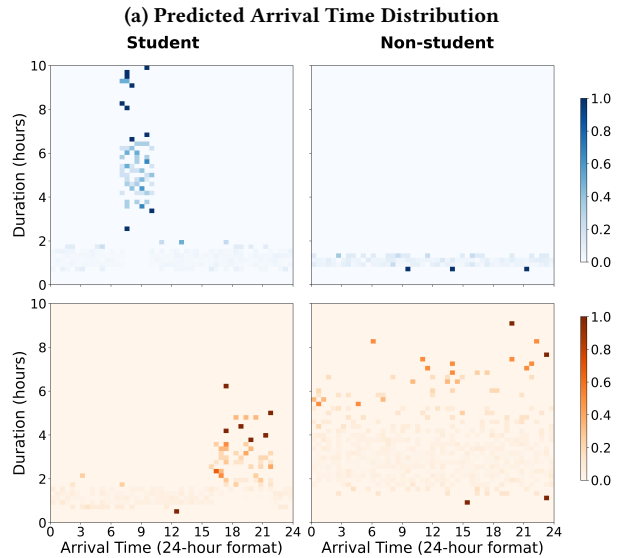
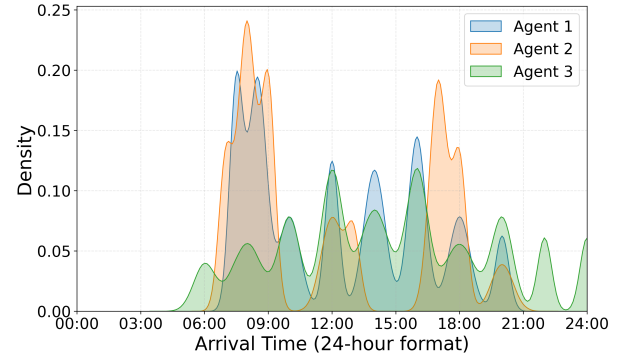
Figure 4b provides deeper insight into the model’s personalization capabilities by focusing on the duration predictions for two example agents: a student and a non-student. The heatmaps reveal distinct patterns for school and recreation POI types: The student agent typically attends school in the morning, spending 6-7 hours on campus, while the non-student agent rarely visits school and spends significantly less time there. For recreation activities, the student agent usually spends shorter periods in the afternoon and evening, whereas the non-student agent’s recreation time is more evenly distributed, indicating a more flexible schedule.

These visualizations confirm that the model effectively differentiates between agents based on their unique patterns, leading to more accurate anomaly detection. For instance, agents who typically follow regular routes and schedules are easily distinguished from those with more erratic behaviors, allowing the model to detect deviations from expected behavior more precisely in both cases.

5.4 Ablation Study: Impact of Model Components

To assess the contribution of each component in our pipeline, we conducted an ablation study by systematically removing one component at a time and evaluating the pipeline’s performance using the metrics introduced in Section 5.1.4.

Table 2 summarizes the results of this ablation study for both NUMOSIM-LA and Urban Anomalies datasets, highlighting the performance degradation observed when each component is excluded from the pipeline. The results show that the most significant drop



(b) Predicted Duration Distribution (Student vs Non-student)

Figure 4: Visualization of (a) the predicted arrival time distribution across multiple agents and (b) the predicted duration distribution conditioned on agent embedding, arrival time, and POI types (blue for school, orange for recreation) for a student agent and a non-student agent.

in performance occurs when the agent embedding is removed: the agent-level AUPR decreases from 1.21% to 0.01% on the NUMOSIM-LA dataset, from 16.54% to 10.47% on the Urban Anomalies-Berlin dataset, and from 16.12% to 6.43% on the Urban Anomalies-Atlanta dataset. Similar declines are observed across other metrics, underscoring the critical role of agent embedding in model performance.

Additionally, removing the arrival time estimation model, the POI type estimation model, or the duration estimation model also leads to a decrease in performance, indicating that each of these components contributes significantly to the overall effectiveness of the model.

5.5 Visit Rate Analysis

The NUMOSIM paper [45] demonstrated that a simple visit rate model — tracking the frequency of visits to various points of interest (POIs) — can outperform many existing baselines in detecting

Table 2: Ablation Study Results for Agent-Level and Staypoint-Level Detection on NUMOSIM-LA and Urban AnomaliesS-BERLIN Datasets. The results for agent-level and staypoint-level are reported together, separated by a '/' within each cell.

Configuration	Dataset	AUPR	AUROC
Full Pipeline	NUMOSIM-LA	1.21% / 0.42%	72.58% / 65.98%
	Urban Anomalies-BERLIN	16.54% / 17.71%	60.58% / 55.93%
	Urban Anomalies-ATLANTA	16.12% / 16.70%	60.04% / 54.64%
Without Arrival Time Prediction	NUMOSIM-LA	0.94% / 0.29%	70.17% / 60.87%
	Urban Anomalies-BERLIN	14.52% / 12.47%	58.16% / 54.13%
	Urban Anomalies-ATLANTA	10.89% / 14.25%	58.12% / 52.91%
Without POI Type Prediction	NUMOSIM-LA	0.85% / 0.31%	66.82% / 55.41%
	Urban Anomalies-BERLIN	13.37% / 11.59%	55.62% / 52.79%
	Urban Anomalies-ATLANTA	0.76% / 13.02%	54.74% / 58.62%
Without Duration Prediction	NUMOSIM-LA	0.76% / 0.28%	65.09% / 51.88%
	Urban Anomalies-BERLIN	12.08% / 10.42%	52.94% / 51.03%
	Urban Anomalies-ATLANTA	8.66% / 12.50%	53.42% / 52.15%
Without Agent Embedding	NUMOSIM-LA	0.01% / 0.01%	50.88% / 52.32%
	Urban Anomalies-BERLIN	10.47% / 8.19%	49.83% / 52.69%
	Urban Anomalies-ATLANTA	6.43% / 10.11%	51.97% / 50.83%

Table 3: Results of DeepBayesic + Visit Rate Incorporation on Agent-Level Detection across NUMOSIM-LA, Urban Anomalies-Berlin, and Urban Anomalies-Atlanta Datasets. The performance metrics (AUPR, AUROC, Precision, and Max F1 Score) are reported for agent-level anomaly detection tasks.

Dataset	Method	AUPR	AUROC	Precision	Max F1 Score
NUMOSIM-LA	Visit Rate Baseline [45]	1.64%	64.6%	-	-
	DeepBayesic + Visit Rate	5.81%	66.17%	4.34e-01	1.63e-01
Urban Anomalies-Berlin	DeepBayesic + Visit Rate	36.37%	71.32%	1.00e+00	3.83e-01
Urban Anomalies-Atlanta	DeepBayesic + Visit Rate	41.38%	77.38%	9.17e-01	4.53e-01

anomalies. We chose not to incorporate the visit rate directly into our main model to ensure a fair comparison with other baseline methods that do not utilize this additional knowledge. This approach allows the evaluation and comparisons in Section 5.2 to focus solely on the core concepts presented in the paper, rather than on specific attributes.

However, for completeness, we integrated the visit rate attribute into our model and compared its performance against the baseline provided in the original NUMOSIM paper. The visit rate model computes anomaly scores by comparing the observed visit rate between the training and testing periods, normalized by the standard deviation observed during training. To integrate this into our model, we normalize the computed anomaly scores between 0 and 1 and multiply them with the final anomaly scores produced by our framework, ensuring that the visit rate contributes proportionally to the overall anomaly score.

The impact of incorporating the visit rate into our framework is summarized in Table 3, where we report the performance metrics for the NUMOSIM-LA, Urban Anomalies-Berlin, and Urban Anomalies-Atlanta datasets. The table includes only AUPR and AUROC for the baseline visit rate model as these are the only metrics provided in the original paper. Notably, our approach significantly surpasses the Visit Rate Baseline on agent level on NUMOSIM-LA dataset. It also surpasses DeepBayesic (without incorporating visit rate) on Urban Anomalies-Berlin and Urban Anomalies-Atlanta datasets. These results highlight its effectiveness in detecting subtle anomalies.

6 Conclusion

In conclusion, DeepBayesic represents a return to foundational principles, demonstrating that by combining the strengths of Bayesian theory with advanced neural density estimation techniques, we can develop powerful, interpretable, and effective solutions for spatiotemporal anomaly detection. Our approach also highlights the importance of personalized modeling in capturing the unique behavioral patterns of individuals in mobility data. By incorporating personalization through learned agent embeddings, the model is able to detect subtle and context-specific anomalies, even in sparse datasets. This integrated approach ensures both robustness and accuracy while providing a solid foundation for future enhancements.

Acknowledgments

Research supported by the Intelligence Advanced Research Projects Activity (IARPA) via the Department of Interior/Interior Business Center (DOI/IBC) contract number 140D0423C0033. 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 or the U.S. Government.

References

- [1] Sofiane Abbar, Rade Stanojevic, and Mohamed Mokbel. 2020. STAD: Spatio-temporal adjustment of traffic-oblivious travel-time estimation. In *2020 21st IEEE International Conference on Mobile Data Management (MDM)*. IEEE, 79–88.
- [2] Hossein Amiri, Will Kohn, Shiyang Ruan, Joon-Seok Kim, Hamdi Kavak, Andrew Crooks, Dieter Pfoser, Carola Wenk, and Andreas Zufle. 2024. The Patterns of Life Human Mobility Simulation. arXiv:2410.00185 [cs.MA] <https://arxiv.org/abs/2410.00185>
- [3] Hossein Amiri, Ruochen Kong, and Andreas Zufle. 2024. Urban Anomalies: A Simulated Human Mobility Dataset with Injected Anomalies. arXiv:2410.01844 [cs.SI] <https://arxiv.org/abs/2410.01844>
- [4] Luis Antonio Souto Arias, Cornelis W Oosterlee, and Pasquale Cirillo. 2023. AIDA: Analytic isolation and distance-based anomaly detection algorithm. *Pattern Recognition* 141 (2023), 109607.
- [5] Hugo Barbosa, Marc Barthelémy, Gourab Ghoshal, Charlotte R James, Maxime Lenormand, Thomas Louail, Ronaldo Menezes, José J Ramasco, Filippo Simini, and Marcello Tomasini. 2018. Human mobility: Models and applications. *Physics Reports* 734 (2018), 1–74.
- [6] Aline Bessa, Fernando de Mesentier Silva, Rodrigo Frassetto Nogueira, Enrico Bertini, and Juliana Freire. 2016. RioBusData: Outlier Detection in Bus Routes of Rio de Janeiro. arXiv:1601.06128 (Jan. 2016). <http://arxiv.org/abs/1601.06128> [cs].
- [7] Christopher M Bishop. 1994. Mixture density networks. (1994).
- [8] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. 2018. Neural ordinary differential equations. *Advances in neural information processing systems* 31 (2018).
- [9] Yile Chen, Cheng Long, Gao Cong, and Chenliang Li. 2020. Context-aware deep model for joint mobility and time prediction. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 106–114.
- [10] Michael Cruz and Luciano Barbosa. 2020. Learning GPS Point Representations to Detect Anomalous Bus Trajectories. *IEEE Access* 8 (2020), 229006–229017. <https://doi.org/10.1109/ACCESS.2020.3046912>
- [11] Soumi Das, Rajath Nandan Kalava, Kolli Kiran Kumar, Akhil Kandregula, Kalpan Suhaas, Sourangshu Bhattacharya, and Niloy Ganguly. 2019. Map enhanced route travel time prediction using deep neural networks. *arXiv preprint arXiv:1911.02623* (2019).
- [12] Alexandre De Brébisson, Étienne Simon, Alex Auvolat, Pascal Vincent, and Yoshua Bengio. 2015. Artificial neural networks applied to taxi destination prediction. In *Proceedings of the 2015th International Conference on ECML PKDD Discovery Challenge-Volume 1526*. 40–51.
- [13] Patrick Ebel, Ibrahim Emre Göl, Christoph Lingensfelder, and Andreas Vogel-sang. 2020. Destination prediction based on partial trajectory data. In *2020 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 1149–1155.
- [14] Tao-yang Fu and Wang-Chien Lee. 2019. DeepIST: Deep image-based spatio-temporal network for travel time estimation. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 69–78.
- [15] Marta C Gonzalez, Cesar A Hidalgo, and Albert-Laszlo Barabasi. 2008. Understanding individual human mobility patterns. *nature* 453, 7196 (2008), 779–782.
- [16] Bharat Gupta, Shivam Awasthi, Rudraksha Gupta, Likhama Ram, Pramod Kumar, Bakshi Rohit Prasad, and Sonali Agarwal. 2018. Taxi travel time prediction using ensemble-based random forest and gradient boosting model. In *Advances in Big Data and Cloud Computing*. Springer, 63–78.
- [17] Samiul Hasan, Christian M Schneider, Satish V Ukkusuri, and Marta C González. 2013. Spatiotemporal patterns of urban human mobility. *Journal of Statistical Physics* 151 (2013), 304–318.
- [18] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems* 33 (2020), 6840–6851.
- [19] Thomas Hoch. 2015. An Ensemble Learning Approach for the Kaggle Taxi Travel Time Prediction Challenge.. In *DC@ PKDD/ECML*.
- [20] Shang-Ling Hsu, Emmanuel Tung, John Krumm, Cyrus Shahabi, and Khurram Shafique. 2014. TrajGPT: Controlled Synthetic Trajectory Generation Using a Multitask Transformer-Based Spatiotemporal Model. *SIGSPATIAL* (2014).
- [21] Wunjun Huo, Wei Wang, and Wen Li. 2019. Anomalydetect: An online distance-based anomaly detection algorithm. In *Web Services–ICWS 2019: 26th International Conference, Held as Part of the Services Conference Federation, SCF 2019, San Diego, CA, USA, June 25–30, 2019, Proceedings 26*. Springer, 63–79.
- [22] John M Irvine, Laura Mariano, and Teal Guidici. 2018. Normalcy modeling using a dictionary of activities learned from motion imagery tracking data. In *2018 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*. IEEE, 1–9.
- [23] Junteng Jia and Austin R Benson. 2019. Neural jump stochastic differential equations. *Advances in Neural Information Processing Systems* 32 (2019).
- [24] Kevin R Keane. 2017. Detecting motion anomalies. In *Proceedings of the 8th ACM SIGSPATIAL Workshop on GeoStreaming*. 21–28.
- [25] Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. 2021. Variational diffusion models. *Advances in neural information processing systems* 34 (2021), 21696–21707.
- [26] Diederik P Kingma. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).
- [27] Istvan Kiss, Béla Genge, Piroska Haller, and Gheorghe Sebestyén. 2014. Data clustering-based anomaly detection in industrial control systems. In *2014 IEEE 10th International Conference on Intelligent Computer Communication and Processing (ICCP)*. IEEE, 275–281.
- [28] Ivan Kobyzev, Simon JD Prince, and Marcus A Brubaker. 2020. Normalizing flows: An introduction and review of current methods. *IEEE transactions on pattern analysis and machine intelligence* 43, 11 (2020), 3964–3979.
- [29] Hoang Thanh Lam. 2016. A concise summary of spatial anomalies and its application in efficient real-time driving behaviour monitoring. In *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. 1–9.
- [30] Wuwei Lan, Yanyan Xu, and Bin Zhao. 2019. Travel time estimation without road networks: an urban morphological layout representation approach. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. 1772–1778.
- [31] Jinbo Li, Hesam Izakian, Witold Pedrycz, and Iqbal Jamal. 2021. Clustering-based anomaly detection in multivariate time series data. *Applied Soft Computing* 100 (2021), 106919.
- [32] Yadong Li, Bailong Liu, Lei Zhang, Susong Yang, Changxing Shao, and Dan Son. 2020. Fast Trajectory Prediction Method With Attention Enhanced SRU. *IEEE Access* 8 (2020), 206614–206621.
- [33] Chengwu Liao, Chao Chen, Chaocan Xiang, Hongyu Huang, Hong Xie, and Songtao Guo. 2021. Taxi-Passenger’s Destination Prediction via GPS Embedding and Attention-Based BiLSTM Model. *IEEE Transactions on Intelligent Transportation Systems* (2021).
- [34] Yueyang Liu, Lance Kennedy, Hossein Amiri, and Andreas Zufle. 2024. Neural Collaborative Filtering to Detect Anomalies in Human Semantic Trajectories. arXiv:2409.18427 [cs.LG] <https://arxiv.org/abs/2409.18427>
- [35] 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)*. IEEE, 949–960.
- [36] Massimiliano Luca, Gianni Barlacchi, Bruno Lepri, and Luca Pappalardo. 2021. A survey on deep learning for human mobility. *ACM Computing Surveys (CSUR)* 55, 1 (2021), 1–44.
- [37] Hongyuan Mei and Jason M Eisner. 2017. The neural Hawkes process: A neurally self-modulating multivariate point process. *Advances in neural information processing systems* 30 (2017).
- [38] George Papamakarios, Theo Pavlakou, and Iain Murray. 2017. Masked autoregressive flow for density estimation. *Advances in neural information processing systems* 30 (2017).
- [39] Guo Pu, Lijuan Wang, Jun Shen, and Fang Dong. 2020. A hybrid unsupervised clustering-based anomaly detection method. *Tsinghua Science and Technology* 26, 2 (2020), 146–153.
- [40] Ernst Georg Ravenstein. 1885. The laws of migration. Royal Statistical Society.
- [41] Danilo Rezende and Shakir Mohamed. 2015. Variational inference with normalizing flows. In *International conference on machine learning*. PMLR, 1530–1538.
- [42] Alberto Rossi, Gianni Barlacchi, Monica Bianchini, and Bruno Lepri. 2019. Modelling taxi drivers’ behaviour for the next destination prediction. *IEEE Transactions on Intelligent Transportation Systems* 21, 7 (2019), 2980–2989.
- [43] Natasa Sarafjanovic-Djukic and Jesse Davis. 2019. Fast distance-based anomaly detection in images using an inception-like autoencoder. In *Discovery Science: 22nd International Conference, DS 2019, Split, Croatia, October 28–30, 2019, Proceedings 22*. Springer, 493–508.
- [44] Li Song, Ruijia Wang, Ding Xiao, Xiaotian Han, Yanan Cai, and Chuan Shi. 2018. Anomalous trajectory detection using recurrent neural network. In *International Conference on Advanced Data Mining and Applications*. Springer, 263–277.
- [45] Chris Stanford, Suman Adari, Xishun Liao, Yueshuai He, Qinhuo Jiang, Chenchen Kuai, Jiaqi Ma, Emmanuel Tung, Yinlong Qian, Lingyi Zhao, Zihao Zhou, Zeeshan Rasheed, and Khurram Shafique. 2024. NUMOSIM: A Synthetic Mobility Dataset with Anomaly Detection Benchmarks. arXiv:2409.03024 [cs.LG] <https://arxiv.org/abs/2409.03024>
- [46] Mark Tenzer, Emmanuel Tung, Zeeshan Rasheed, and Khurram Shafique. 2024. Generating Trajectories from Implicit Neural Models. In *2024 25th IEEE International Conference on Mobile Data Management (MDM)*. IEEE, 129–138.
- [47] Guy Tevet, Brian Gordon, Amir Hertz, Amit H Bermano, and Daniel Cohen-Or. 2022. Motionclip: Exposing human motion generation to clip space. In *European Conference on Computer Vision*. Springer, 358–374.
- [48] Benigno Uribe, Iain Murray, and Hugo Larochelle. 2013. RNADE: The real-valued neural autoregressive density-estimator. *Advances in Neural Information Processing Systems* 26 (2013).
- [49] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).
- [50] Hao Wu, Weiwei Sun, and Baihua Zheng. 2017. A fast trajectory outlier detection approach via driving behavior modeling. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 837–846.

- [51] Jiayi Xie and Zhenzhong Chen. 2023. Hierarchical transformer with spatio-temporal context aggregation for next point-of-interest recommendation. *ACM Transactions on Information Systems* 42, 2 (2023), 1–30.
- [52] Hao Xue, Flora Salim, Yongli Ren, and Nuria Oliver. 2021. MobTCast: Leveraging auxiliary trajectory forecasting for human mobility prediction. *Advances in Neural Information Processing Systems* 34 (2021), 30380–30391.
- [53] Song Yang, Jiamou Liu, and Kaiqi Zhao. 2022. GETNext: trajectory flow map enhanced transformer for next POI recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on research and development in information retrieval*. 1144–1153.
- [54] Lei Zhang, Guoxing Zhang, Zhizheng Liang, and Ekene Frank Ozioko. 2018. Multi-features taxi destination prediction with frequency domain processing. *PloS one* 13, 3 (2018), e0194629.
- [55] Xiaocai Zhang, Zhixun Zhao, Yi Zheng, and Jinyan Li. 2019. Prediction of taxi destinations using a novel data embedding method and ensemble learning. *IEEE Transactions on Intelligent Transportation Systems* 21, 1 (2019), 68–78.
- [56] Jiangchuan Zheng, Siyuan Liu, and Lionel M Ni. 2013. Effective routine behavior pattern discovery from sparse mobile phone data via collaborative filtering. In *2013 IEEE international conference on pervasive computing and communications (PerCom)*. IEEE, 29–37.
- [57] 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*. 665–674.
- [58] Zihao Zhou, Xingyi Yang, Ryan Rossi, Handong Zhao, and Rose Yu. 2022. Neural point process for learning spatiotemporal event dynamics. In *Learning for Dynamics and Control Conference*. PMLR, 777–789.
- [59] Zihao Zhou and Rose Yu. 2024. Automatic integration for spatiotemporal neural point processes. *Advances in Neural Information Processing Systems* 36 (2024).