Anomaly Detection in Trajectory Data with Normalizing Flows

Madson L. D. Dias
Department of Computer Science
Federal University of Ceará
Fortaleza, Brazil
madsonddias@gmail.com

César Lincoln C. Mattos
Department of Computer Science
Federal University of Ceará
Fortaleza, Brazil
cesarlincoln@dc.ufc.br

Ticiana L. C. da Silva
Virtual UFC Institute
Federal University of Ceará
Fortaleza, Brazil
ticianalc@insightlab.ufc.br

José Antônio F. de Macêdo
Department of Computer Science
Federal University of Ceará
Fortaleza, Brazil
jose.macedo@insightlab.ufc.br

Wellington C. P. Silva
National Department of Public Security
Federal District, Brazil
wellington.wcps@gmail.com

***Abstract*—The task of detecting anomalous data patterns is as important in practical applications as challenging. In the context of spatial data, recognition of unexpected trajectories brings additional difficulties, such as high dimensionality and varying pattern lengths. We aim to tackle such a problem from a probability density estimation point of view, since it provides an unsupervised procedure to identify out of distribution samples. More specifically, we pursue an approach based on normalizing flows, a recent framework that enables complex density estimation from data with neural networks. Our proposal computes exact model likelihood values, an important feature of normalizing flows, for each segment of the trajectory. Then, we aggregate the segments' likelihoods into a single coherent trajectory anomaly score. Such a strategy enables handling possibly large sequences with different lengths. We evaluate our methodology, named aggregated anomaly detection with normalizing flows (GRADINGS), using real world trajectory data and compare it with more traditional anomaly detection techniques. The promising results obtained in the performed computational experiments indicate the feasibility of the GRADINGS, specially the variant that considers autoregressive normalizing flows.**

***Index Terms*—trajectory data, anomaly detection, density estimation, normalizing flows**

I. Introduction

The wide availability of spatial data acquisition devices, from specialized remote sensors to standard GPS equipped smartphones, has resulted in the creation of several location-based applications. Although the object to be localized can vary (a vehicle, an animal, a person, etc.), in general, such trajectory data can be understood as a series of ordered points that characterize the object motion [1].

In the context of trajectory data, anomaly detection is a task critical to monitor spatial events and enable recognition of unexpected behaviors [2]. One could define an anomaly, or outlier[1], as a data point which significantly differs from the overall observed data [3]. It is worth emphasizing that, as opposed to single point standard regression, in such a scenario a data example is a full trajectory or at least a segment of it.

Meng *et al.* [2] propose a traditional anomaly detection taxonomy that includes methods based on classification, clustering, distance, density and statistics. We pursue the latter, which consists in a model-based procedure that aims to explain the available data, mostly within a probabilistic density estimation framework. Anomalies are then detected by measuring how much the model fits a given new data point. Such an approach does not require labeled data, as it is a form of unsupervised learning. However, probabilistic density estimation approaches for trajectory anomaly detection usually consider simple distributions, such as a multivariate Gaussian [4]. Even if a more flexible Gaussian mixture model (GMM) is used, such as in [5], [6], it is not straightforward to determine the number of components in the mixture.

In this work we tackle the task of trajectory data analysis by pursuing an approach based on normalizing flows (NFs, [7]), a general framework for estimating complex probabilistic densities. In summary, a NF transforms an initial simple density by a sequence of invertible transformations to better explain the observed data. Following recent works, such as [7], each transformation, i.e. a flow, is parametrized by (possibly deep) neural networks. One of the main advantages of such flow-based approach is the available exact model log-likelihood, which is used as an objective function to jointly optimize all the model parameters. Furthermore, we can compute exact log-likelihood values for new data points, which we will then apply as a coherent anomaly score.

NF-based anomaly detection approaches have been recently proposed [8], [9]. In contrast to those works, we aim to evaluate NF models with trajectory data, which is inherently sequential. Thus, we include in our evaluations the so-called masked autoregressive flow (MAF), an autoregressive flow framework that directly models the conditional distributions of the input variables [10].

[1]In this work we use the terms *outlier* and *anomaly* interchangeably.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

Trajectory data can be high dimensional due to the presence of several measured points within a single trajectory. Besides, distinct data examples can have different lengths, which cannot be straightforwardly compared. We propose a methodology that tackles both issues by considering fixed-size segments of the available trajectories. A NF generative model is than used to estimate the probability density of such segments. It is expected that segments which belong to trajectories considered normal correspond to higher model likelihoods than segments that belong to trajectories considered anomalous. In the test step, a single anomaly score for a new trajectory is computed from its segments using an aggregation function. Moreover, since we choose a trajectory data representation that incorporates timestamps, the time domain is considered in the modeling. We name our approach aggregated anomaly detection with normalizing flows (GRADINGS). We emphasize that, to the best of our knowledge, our work is the first evaluation of NF-based models in the task of anomaly detection in trajectory data.

We evaluate the proposed GRADINGS approach using real-world trajectories available in the Microsoft GeoLife data set [11]–[13]. The obtained experimental results indicate the feasibility of our solution. Our framework, specially the variant that considers the MAF model, achieves better anomaly detection results in comparison to standard techniques, such as the GMM and the local outlier factor (LOF, [14]) method.

II. Problem statement and data representation

The problem of anomaly detection can be vaguely described as the task of finding data patterns that differ from what is expected and is considered normal [15]. In this work, such unexpected patterns are related to trajectories sufficiently different from the previously seen data, which is assumed to be mostly normal. We consider that trajectories can differ in terms of spatial segments that comprise them and/or the time period they occur.

As follows we establish the adopted data representation and the main theoretical aspects of the unsupervised anomaly detection task.

A. Trajectory representation

Broadly speaking, a trajectory consists of a sequence of GPS points (i.e., latitude, longitude and timestamp) generated by a moving object on a monitoring system. Below we formally define it.

Definition 2.1 (Trajectory): A trajectory $\mathbf{T}_m \in \mathcal{T}$ with size L_m is defined as a finite ordered sequence

$$\mathbf{T}_m \triangleq \left(\boldsymbol{q}_1^{(m)}, \boldsymbol{q}_2^{(m)}, \cdots, \boldsymbol{q}_l^{(m)}, \cdots, \boldsymbol{q}_{L_m}^{(m)}\right), \tag{1}$$

where $\boldsymbol{q}_l^{(m)} = \left(q_{l,1}^{(m)}, q_{l,2}^{(m)}, q_{l,3}^{(m)}\right)$ is a *location point*, such that $q_{l,1}^{(m)}, q_{l,2}^{(m)}, q_{l,3}^{(m)}$ are respectively the l-th latitude, l-th longitude, and l-th timestamp of the trajectory $\mathbf{T}_m$. Furthermore, we have $q_{l_0,3} < q_{l_1,3}$, for all $l_0 < l_1$, which ensures a temporal ordered sequence of points.

B. Problem statement

Given a set of trajectories, the goal of a trajectory anomaly detection model is to find trajectories that are significantly different from the majority, considered to be normal. In other words, let $\mathcal{T} = \{\mathbf{T}_n\}_{n=1}^{N}$ be a set of trajectories of moving objects in a GPS monitoring system. The task of trajectory anomaly detection is to create a model from the available trajectories to evaluate the anomaly degree of any given trajectory $\mathbf{T}$.

In this work, we follow an unsupervised anomaly detection procedure, which does not require the trajectories to be previously labeled as normal or anomalous. We detail such an approach as follows.

C. Unsupervised anomaly detection

Unsupervised anomaly detection approaches (or anomaly detection over noisy data [16]) makes two assumptions over the data. The first one is that the dataset contains a large number of normal elements and relatively few anomalies. The second assumption is that the abnormal data is generated by a different probability distribution [17].

After the training step, the determination if a sample $\boldsymbol{x}$ is normal or abnormal can be made by a decision system H as follows:

$$H(\boldsymbol{x}, \phi) = \begin{cases} 0 \text{ (normal)} & \text{if} \quad A(\boldsymbol{x}) < \phi, \\ 1 \text{ (abnormal)} & \text{if} \quad A(\boldsymbol{x}) \geq \phi, \end{cases} \tag{2}$$

where ϕ is a predefined threshold and $A(\cdot)$ is an *anomaly score*. The threshold ϕ is a value that separates abnormal from normal data samples. In the context of supervised and semi-supervised anomaly detection, this value is usually chosen by using a validation set that contains known anomalous samples [18], [19]. After that, metrics such as accuracy and F_1-score are computed to judge the quality of the models.

Alternatively, and more common to the unsupervised learning setup, we can judge the model quality without the choice of a single value for the threshold ϕ. This can be done by finding the receiver operating characteristic (ROC) curve, which indicates the relation between the false positive rate and the true positive rate as the threshold is changed. A practical metric to summarize the information provided by the ROC curve is the area under the curve (AUC or AUROC) [20].

III. Classical anomaly detection techniques

Anomaly detection algorithms can be classified in several groups based on distance, probability, reconstruction, and information theory [21]. In this section, we describe two of most know techniques used in anomaly detection problems.

A. Anomaly detection using the LOF algorithm

Local outlier factor (LOF, [14]) is an unsupervised distance-based anomaly detection algorithm. The anomaly score in LOF is computed by comparing the *local density* of a sample to the surrounding neighborhood. The local density is inversely correlated with the average distance from the point to its neighborhood.

Let $\mathcal{X}$ be a set of data points. The set of K-nearest neighbors of $\boldsymbol{x} \in \mathcal{X}$ is denoted by $\mathcal{N}_{\boldsymbol{x}}^{k}$ and defined as $\mathcal{N}_{\boldsymbol{x}}^{K} \triangleq k\mathrm{NN}\left(K, \boldsymbol{x}, \mathcal{X} \setminus \{\boldsymbol{x}\}\right)$, where $k\mathrm{NN}\left(\cdot,\cdot,\cdot\right)$ is the result of a K-nearest neighbor query [22], [23].

Then, we define the K-distance neighborhood $\mathrm{KD}(\boldsymbol{x})$ of a sample $\boldsymbol{x} \in \mathcal{X}$ as $\mathrm{KD}(\boldsymbol{x}) \triangleq \max_{\boldsymbol{u} \in \mathcal{N}_{\boldsymbol{x}}^{K}} \|\boldsymbol{x} - \boldsymbol{u}\|$, where $\|\cdot\|$ is the Euclidean distance.

We use the above to define the reachability distance $\mathrm{RD}(\boldsymbol{x}, \boldsymbol{u})$ of $\boldsymbol{x}$ with respect to another sample $\boldsymbol{u} \in \mathcal{X}$ as $\mathrm{RD}(\boldsymbol{x}, \boldsymbol{u}) \triangleq \max\left\{\mathrm{KD}(\boldsymbol{x}), \|\boldsymbol{x} - \boldsymbol{u}\|\right\}$.

The local reachability density of $\boldsymbol{x}$ with respect to $\boldsymbol{u}$ is then denoted by $\mathrm{LRD}(\boldsymbol{x}, \boldsymbol{u})$ and defined as

$$\mathrm{LRD}(\boldsymbol{x}) \triangleq \frac{K}{\sum_{\boldsymbol{u} \in \mathcal{N}_{\boldsymbol{x}}^{K}} \mathrm{RD}(\boldsymbol{x}, \boldsymbol{u})}.$$

Using all the previous definitions, the LOF anomaly score can be finally formalized as the average ratio of local reachability densities with respect to $\boldsymbol{x}$ and its K-neighborhood:

$$A(\boldsymbol{x}) = \frac{1}{K} \sum_{\boldsymbol{u} \in \mathcal{N}_{\boldsymbol{x}}^{K}} \frac{\mathrm{LRD}(\boldsymbol{x})}{\mathrm{LRD}(\boldsymbol{u})}. \tag{3}$$

Note that the above score measures the local density deviation of a given data point with respect to its neighbors.

B. Gaussian mixture model for anomaly detection

One way of computing an anomaly score A in Eq. (2) is to use a probability density estimator. This approach first trains the density estimator $p(\cdot)$ and then uses the negative log-likelihood of each testing data as an anomaly score, i.e.,

$$A(\boldsymbol{x}) = -\ln p(\boldsymbol{x}). \tag{4}$$

In such a context, the Gaussian mixture model (GMM) is a common choice. A GMM uses a linear combination of Gaussian density functions to approximate an unknown probability distribution. The parameters of each the component are usually adjusted using the Expectation Maximization (EM, [24]) algorithm.

Consider a data set $\mathcal{D} = \{\boldsymbol{x}_n\}_{n=1}^{N}$, where $\boldsymbol{x}_n \in \mathbb{R}^D$. We assume that the points from $\mathcal{D}$ are generated in an i.i.d. fashion from an underlying density $p(\boldsymbol{x})$. Furthermore, suppose that $p(\boldsymbol{x})$ is defined as a finite mixture model with K components:

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x} | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{5}$$

where $\mathcal{N}(\boldsymbol{x} | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ is a multivariate Gaussian density with mean vector $\boldsymbol{\mu}_k$ and covariance matrix $\boldsymbol{\Sigma}_k$; $\{\pi_k\}_{k=1}^{K}$ are the mixture weights, which are restricted to be non-negative and sum up to 1, i.e., $\sum_{k=1}^{K} \pi_k = 1$. The mixture weights represent the probability that a randomly selected data point $\boldsymbol{x}$ was generated by the component k. After the optimization of the GMM parameters via the EM algorithm, Eq. (5) can be directly applied as an anomaly score for new data points.

It is worth noting that flow-based generative models constitute a flexible alternative to density estimation with standard techniques such as the GMM. In the next section we detail the flow-based models used in this work.

IV. Probability density estimation via normalizing flows

NF models are powerful tools for estimating complicated probability densities [25], [26]. Two merits of these models are the exact inference and log-likelihood evaluation [26]. The latter is specially valuable in the context of anomaly detection.

Let $\boldsymbol{x} \in \mathbb{R}^D$ be a random vector with unknown distribution $p(\boldsymbol{x})$. In the most general flow-based model, the generative process is defined as [27]

$$\boldsymbol{h} \sim p(\boldsymbol{h}), \tag{6}$$

$$\boldsymbol{x} = g(\boldsymbol{h}), \tag{7}$$

where $\boldsymbol{h}$ is a latent (unobserved) variable and $p(\boldsymbol{h})$ is a simple and known distribution, e.g., a multivariate Gaussian. The function $g(\cdot)$, called *bijective*, is an invertible function such that $g^{-1}(\boldsymbol{x}) = f(\boldsymbol{x}) = \boldsymbol{h}$. If the transformation $f(\cdot)$ is considered to be a composition of K successive mappings and we apply the change of variables rule, the log-likelihood of the random variable $\boldsymbol{x}$ can be written as [27]

$$\ln p_K(\boldsymbol{z}_K) = \ln p_0(\boldsymbol{z}_0) - \sum_{k=1}^{K} \ln \left| \det \frac{\partial \boldsymbol{z}_k}{\partial \boldsymbol{z}_{k-1}} \right|, \tag{8}$$

where $\boldsymbol{x} \triangleq \boldsymbol{z}_K \sim p_K(\boldsymbol{z}_K)$, $\boldsymbol{h} \triangleq \boldsymbol{z}_0 \sim p_0(\boldsymbol{z}_0)$ and $\boldsymbol{z}_k = f_k(\boldsymbol{z}_{k-1}), \forall k = 1, 2, \ldots, K$.

The usual training criterion of flow-based generative models is simply the negative log-likelyhood over the training set $\mathcal{X}$:

$$L(\mathcal{X}) = -\frac{1}{|\mathcal{X}|} \sum_{\boldsymbol{x} \in \mathcal{X}} \ln p(\boldsymbol{x}). \tag{9}$$

We summarize the evaluated NF models as follows.

A. Real NVP

Real-valued non-volume preserving (Real-NVP, [28]) is a type of NF that uses a bijection called *coupling layer* that transforms only some input dimensions via functions that depend on the untransformed dimensions. If $1 : d$ denotes the sequential indexes of the d untransformed dimensions, the components of the layer output $\boldsymbol{y}$ are given by

$$\boldsymbol{y}_{1:d} = \boldsymbol{x}_{1:d}, \tag{10}$$

$$\boldsymbol{y}_{d+1:D} = \boldsymbol{x}_{d+1:D} \odot \exp\left(\sigma(\boldsymbol{x}_{1:d})\right) + \mu(\boldsymbol{x}_{1:d}), \tag{11}$$

where $\sigma, \mu : \mathbb{R}^d \to \mathbb{R}^{D-d}$ respectively represent scale and translation functions parametrized by neural networks, and $\odot$ is the element-wise product operator. The elements in each flow are permuted to different orders, allowing all of the inputs to have a chance to be altered.

The Jacobian matrix of the above described transformations can be calculated using

$$\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}} = \begin{bmatrix} \mathbf{I}_d & \mathbf{0}_{d \times (D-d)} \\ \frac{\partial \boldsymbol{y}_{d+1:D}}{\partial \boldsymbol{x}_{1:d}} & \mathrm{diag}\left(\exp\left(\sigma(\boldsymbol{x}_{1:d})\right)\right) \end{bmatrix}, \tag{12}$$

where $\mathbf{I}_d \in \mathbb{R}^{d \times d}$ is the d-order identity matrix, $\mathbf{0}_{d \times (D-d)} \in \mathbb{R}^{d \times (D-d)}$ is a zero matrix and $\mathrm{diag}\left(\exp\left(\sigma(\boldsymbol{x}_{1:d})\right)\right) \in \mathbb{R}^{(D-d) \times (D-d)}$ is a diagonal matrix whose elements are equal

to the vector $\exp(\sigma(\boldsymbol{x}_{1:d}))$. The Jacobian matrix in Eq. (12) is triangular, thus, its determinant is a simple product of the diagonal terms:

$$\det \frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}} = \prod_{j=1}^{D-d} \exp\left(\sigma(\boldsymbol{x}_{1:d})\right)_j = \exp\left(\sum_{j=1}^{D-d} \sigma(\boldsymbol{x}_{1:d})_j\right). \quad (13)$$

Since the computation of the Jacobian determinant of the mentioned transformations does not involve calculating the inverse of the functions $\sigma(\cdot)$ and $\mu(\cdot)$, such functions can be arbitrarily complex, usually a deep neural network [28]. All the model parameters (i.e., the networks' weights) are jointly optimized via maximization of the Eq. (9) via stochastic gradient descent methods.

B. Masked autoregressive flow (MAF)

We can decompose any joint density $p(\boldsymbol{x})$ of high-dimensional data into a product of one-dimensional conditionals using the chain rule of probabilities:

$$p(\boldsymbol{x}) = \prod_{d=1}^{D} p(x_d|x_1, x_2, \cdots, x_{d-1}) = \prod_{d=1}^{D} p(x_d|\boldsymbol{x}_{1:d-1}). \quad (14)$$

The Masked Autoregressive Flow (MAF, [10], [29]) uses the above autoregressive constraint to model the probability density whose conditionals are parameterized as single Gaussians. Thus, the d-th conditional probability is given by

$$p(x_d|\boldsymbol{x}_{1:d-1}) = \mathcal{N}\left(x_d|\mu_d(\boldsymbol{x}_{1:d-1}), \left(\exp\left(\alpha_d(\boldsymbol{x}_{1:d-1})\right)\right)^2\right), \quad (15)$$

where $\mu_d, \alpha_d : \mathbb{R}^{d-1} \mapsto \mathbb{R}$ are two unconstrained scalar functions that compute the mean and log-standard deviation of the d-th conditional given all previous variables. The bijective transformation of MAF generates each y_d conditioned on the past dimensions $\boldsymbol{y}_{1:d-1}$,

$$y_d = x_d \exp\left(\alpha_d(\boldsymbol{y}_{1:d-1})\right) + \mu_d(\boldsymbol{y}_{1:d-1}). \quad (16)$$

As a consequence of the autoregressive nature of this transformation, the dimension d of the resulting variable $\boldsymbol{y}$ depends only on the $1:d$ dimensions of the input variable $\boldsymbol{x}$. Thus, the Jacobian matrix of this transformation is triangular [30] and its determinant is equal to the product of its diagonal terms:

$$\det \frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}} = \prod_{d=1}^{D} \exp\left(\alpha_d(\boldsymbol{y}_{1:d-1})\right) = \exp\left(\sum_{d=1}^{D} \alpha_d(\boldsymbol{y}_{1:d-1})\right). \quad (17)$$

As in the RealNVP, the functions $\mu_d(\cdot)$ and $\alpha_d(\cdot)$ can be arbitrarily complex. In the MAF model, these functions are implemented by an efficient feedforward network called Masked Autoencoder for Distribution Estimation (MADE, [29]) that takes $\boldsymbol{x}$ as input and outputs the means and log-standard deviations for all dimensions in a single network pass.

The nature of MAF transformations allows more flexible generalizations when compared to the RealNVP model. As one can see, if for the first $j \leq d$ dimensions we fix $\mu_j = \alpha_j = 0$ and apply the MAF transformations into the other $j > d$ dimensions, the MAF structure becomes equivalent to the RealNVP. Besides, we can see the coupling layer of the RealNVP as a special case of the MAF transformation [10].

V. Proposed Methodology

We can compute an anomaly score for a sequential data type sample either directly or by first computing scores for local subsections and then aggregating them. These subsections are called pattern fragments, segments, sliding windows, motifs, or n-grams [31]. In Definition 5.1 we present a formal description of these objects.

Definition 5.1 (trajectory segment): Given a trajectory $\mathbf{T}_m$ with length L_m, the segment $\mathbf{S}_i$ of $\mathbf{T}_m$ with a user-defined length W is a finite ordered sequence of location points, denoted by

$$\mathbf{S}_i^{(m)} \triangleq \left(\boldsymbol{q}_i^{(m)}, \boldsymbol{q}_{i+1}^{(m)}, \cdots, \boldsymbol{q}_{i+W}^{(m)}\right). \quad (18)$$

where $W \leqslant L_m$ and $1 \leqslant i \leqslant L_m - W + 1$.

Segment-based techniques usually perform better when compared to direct detection methods [31]. Furthermore, they enable handling large sequences with different lengths. As follows we detail our proposal, named aggregated anomaly detection with normalizing flows (GRADINGS), which consists in three main steps.

In the first step, GRADINGS transforms the set of trajectories into a set of trajectory segments. Thus, given a set of trajectories $\mathcal{T} = \{\mathbf{T}_m\}_{m=1}^{M}$, the transformed set is defined by

$$\mathcal{X} = \bigcup_{m=1}^{M} \left\{\boldsymbol{x}_n = \delta\left(\mathbf{S}_i^{(m)}\right)\Big|_{i=1}^{L_m-W+1}\right\}, \quad (19)$$

where $\boldsymbol{x}_n \in \mathbb{R}^D$, $1 \leqslant n \leqslant N = \sum_{n=1}^{N}(L_m - W)$, and $\delta(\cdot)$ is a function that flattens a $W \times 3$-segment into a D-dimensional row vector, where $D = 3W$, i.e.,

$$\delta\left(\mathbf{S}_i^{(m)}\right) = \left(q_{i,1}^{(m)}, q_{i,2}^{(m)}, q_{i,3}^{(m)}, \cdots, q_{i+W,1}^{(m)}, q_{i+W,2}^{(m)}, q_{i+W,3}^{(m)}\right).$$

The second step consists in estimating the distribution $p(\cdot)$ from the available trajectory segments. This step is performed by using one of the NF generative models described in Section IV. At this point, the GRADINGS is able to compute the anomaly degree for any trajectory segment, denoted by $\alpha\left(\mathbf{S}_i^{(m)}\right)$:

$$\alpha\left(\mathbf{S}_i^{(m)}\right) = -\ln p\left(\delta\left(\mathbf{S}_i^{(m)}\right)\right). \quad (20)$$

In the last step, we aggregate the anomaly scores of the segments to compute a single anomaly score for the trajectory. More specifically, given a trajectory $\mathbf{T}_m$, its anomaly score, denoted by $A(\mathbf{T}_m)$, can be computed using an aggregation function φ that combines the anomaly degree of each segment $\mathbf{S}_i^{(n)}$ in the trajectory $\mathbf{T}_m$, i.e.,

$$A(\mathbf{T}_m) = \varphi\left(\left\{\alpha\left(\mathbf{S}_i^{(m)}\right)\right\}_{i=1}^{L_m-W+1}\right). \quad (21)$$

Possible choices for the aggregation function φ includes the median or the average.

VI. Experiments

To assess the performance of the proposed methodology, we conduct experiments comparing GRADINGS when using either Real NVP or MAF estimators against standard LOF and GMM anomaly detectors with real world data.

A. Data set description

We consider the version 1.3 of the Microsoft GeoLife data set [11]–[13], comprised of real trajectory data measured from 182 users over a period of five years (from April 2007 to August 2012), which is equivalent to 17621 trajectories. For 73 users, the transportation mode is labeled, such as driving, taking a bus, riding a bike and walking. Each trajectory represents a complete trip from departure to arrival location.

In our experimental setting, we use a subset of the data that consists of the trajectories located in Beijing, China, made using car (126 trajectories) or bus (365 trajectories). We define two different scenarios. In the first one, called CAR $\times$ BUS, we use the car trajectories as in-distribution data (i.e., as "normal" patterns) and the bus trajectories as out-of-distribution data (i.e., as "anomalies"). In the second one, called BUS $\times$ CAR scenario we switch the roles: the bus trajectories act as in-distribution data and the car trajectories are seen as out-of-distribution samples. For each scenarios we use segments with length correspondent to 10, 20, and 30 location points, accounting a total of 6 data sets. All of these data sets have 230632 segments of car trajectories and 850082 segments of bus trajectories.

The timestamp information of the trajectory data is firstly converted to the hour of the week (e.g. Tuesday, 12:30, is equal to 36.5 if we consider the Monday as the start of the week) and then encoded into two variables using $\left(\sin\left(2\pi\frac{hour}{168}\right), \cos\left(2\pi\frac{hour}{168}\right)\right)$. This encoding ensures that similar periodic times are close in the input space, even in different weeks (e.g. Sunday, 23:59 is close to Monday, 00:00).

B. Results and discussion

We report results for individual segments scores and full trajectories scores. In the latter, we consider both the average and the median as score aggregation functions φ (see Eq. (21)).

We train all the models on the normal data and then apply them to unseen normal samples as well as abnormal data samples. The normal data have been partitioned into two folds, the first one with 80% of the data for the training, and the other 20% is grouped with the abnormal data to compute the evaluation metrics.

For the MAF and RealNVP models we use 10 flows of neural networks as bijective functions, with the MADE structure in the case of the MAF model. Each network has two hidden hidden layers, each one with 32 neurons. Both models were trained for 300 epochs. A grid search with 5-fold cross-validation is used to perform the hyper-parameter tuning using the training data for the GMM model. The K value of the LOF algorithm was determined using the heuristic presented in [14].

The ROC curves and the correspondent AUROC values are presented in Figs. 1 and 2. In addition, we present in Table I the the false positive rate obtained when we fix a true positive rate of 80%, named the FPR80 metric.

In all evaluated pair scenario-variant the NF-based solutions performed better in terms of AUCROC. In most of them, the GRADINGS framework with the MAF model was the best.

TABLE I
FALSE POSITIVE RATES OBTAINED WHEN WE FIX A TRUE POSITIVE RATE OF 80% (FPR80) FOR ALL EXPERIMENTAL SCENARIOS.

			Model			
Scenario	*Variant*	*Length*	MAF	RealNVP	GMM	LOF
CAR $\times$ BUS	segment	10	**0.423**	0.643	0.698	0.719
		20	**0.498**	0.640	0.653	0.688
		30	**0.608**	0.652	0.699	0.727
	average	10	0.342	**0.335**	0.376	0.465
		20	**0.272**	0.435	0.500	0.550
		30	**0.361**	0.577	0.556	0.622
	median	10	**0.245**	0.375	0.308	0.481
		20	**0.247**	0.335	0.353	0.419
		30	**<u>0.201</u>**	0.361	0.315	0.462
BUS $\times$ CAR	segment	10	0.603	**0.592**	0.597	0.684
		20	**0.510**	0.633	0.682	0.692
		30	**0.489**	0.517	0.631	0.689
	average	10	**0.252**	0.310	0.482	0.712
		20	**0.529**	0.601	0.635	0.704
		30	**0.311**	0.555	0.622	0.732
	median	10	**0.226**	0.330	0.761	0.771
		20	**0.190**	0.294	0.744	0.819
		30	**<u>0.055</u>**	0.328	0.564	0.747

In terms of FPR80, the MAF also achieved better results in 16 out of 18 evaluations, with the RealNVP being slightly better in the others. It is important to highlight that the use of a segment aggregation strategy considerably increased the performance concerning the AUROC in all experiments. Particularly, models with the median as the aggregation function achieved the best results in terms of AUROC and FPR80.

In terms of the segment length, when using the median as the aggregation function, we can see that the performance is inversely proportional to the size of the segment. On the other hand, using the average as the aggregation function, the performance decreases as the segment size increases. Since the average score is more sensitive to outliers, we hypothesize that the increase of the segment size may cause more outliers to appear in the same pattern. The results that consider only the individual segments do not show any specific behavior with respect to the segment length.

In summary, the obtained results indicate the importance of both main ingredients of the proposed GRADINGS framework: (*i*) the NF-based density estimation; and (*ii*) the aggregation of the individual segments degrees into a single trajectory anomaly score. Furthermore, we have also verified that, in general, the combination of the autoregressive MAF model, the median aggregation function and a larger (e.g. 30) segment length representation offers the best performance.

VII. CONCLUSION AND FURTHER WORK

Anomaly detection is a challenging task with important practical applications. In the context of trajectory data, GPS measurements are usually widely available. However, the high dimensional patterns and the lack of labeled data hinder the application of standard techniques.

In this work we have proposed GRADINGS, an unsupervised density estimation methodology that includes flexible

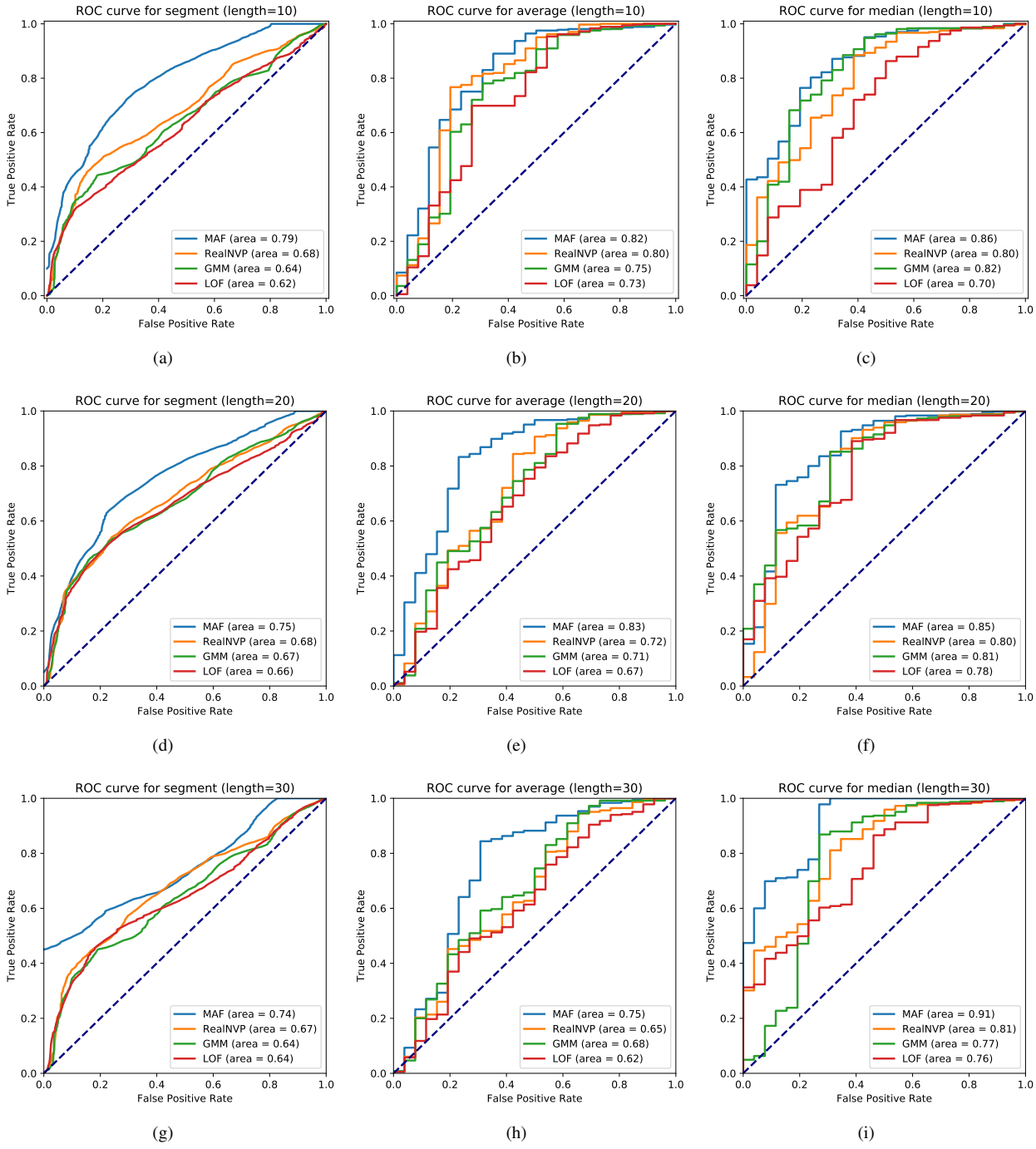

Fig. 1. Anomaly detection results for CAR × BUS scenario. ROC curves and respective AUROC values for segments (left column) and for trajectories, using average (middle column), and median (right column). The rows represent the segment lengths – 10 (a, b, c), 20 (d, e, f), and 30 (g, h, i). The dashed line indicates a completely random detector.

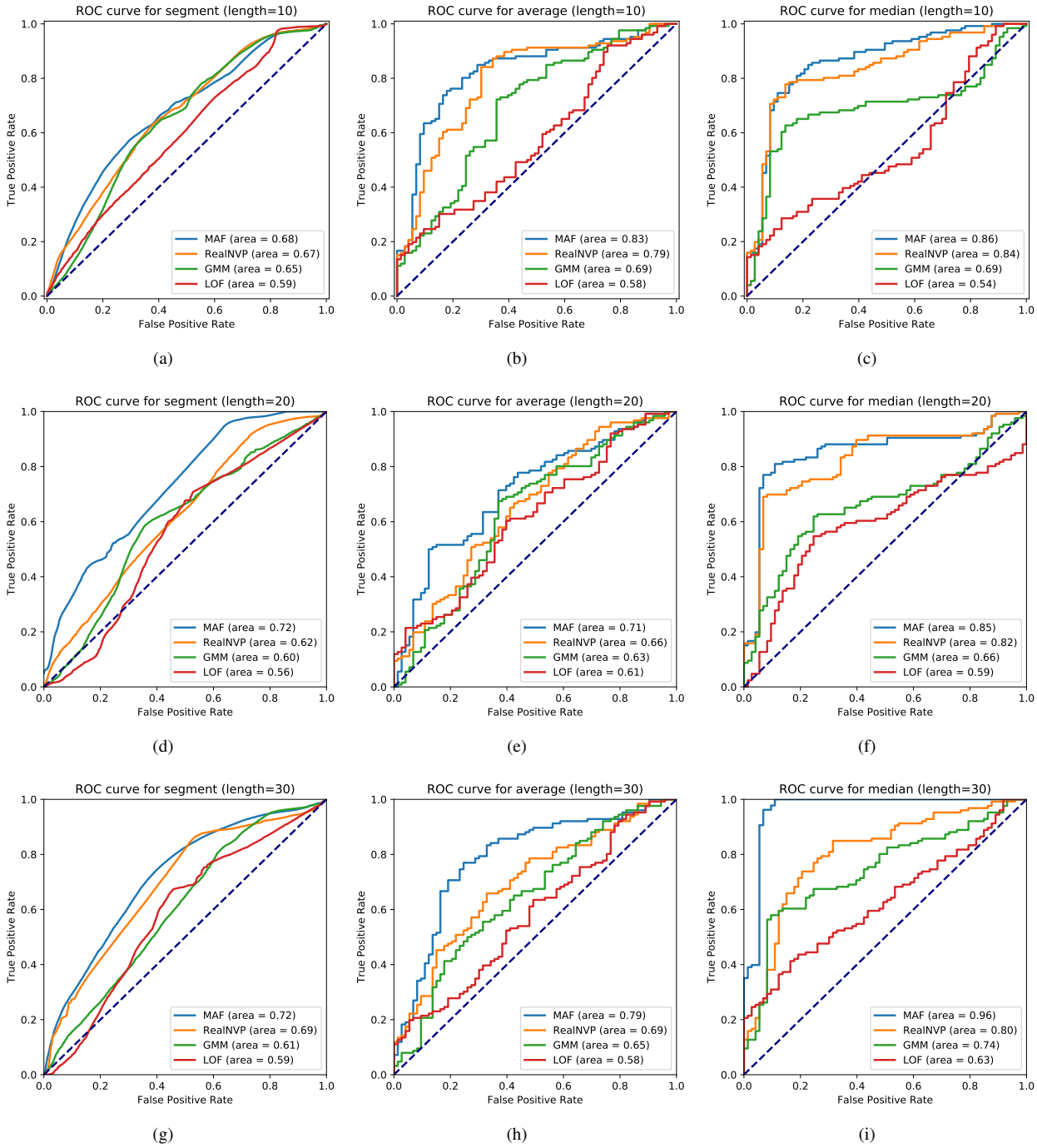

Fig. 2. Anomaly detection results for BUS × CAR scenario. ROC curves and respective AUROC values for segments (left column) and for trajectories, using average (middle column), and median (right column). The rows represent the segment lengths – 10 (a, b, c), 20 (d, e, f), and 30 (g, h, i). The dashed line indicates a completely random detector.

normalizing flows, more specifically the Real NVP and the MAF structures. GRADINGS aggregates the analytical log-likelihood values of trajectory segments into a single robust anomaly score, which enables the use of trajectories with distinct lengths. The empirical results obtained using real world data showed promising performance compared to the LOF and GMM baselines, specially when considering the autoregressive MAF-based version.

The present research outcome encourages us to pursue additional NF approaches for trajectory anomaly detection. For instance, future work shall evaluate the use of convolution-based flows, such as the so-called Glow [26], which can handle data with multiple channel representation. Models with more complex invertible transformations, such as the recently proposed [32]–[34], are also worthy subjects of future investigations.

ACKNOWLEDGMENT

The authors thank the financial support of FUNCAP SPU 8789771/2017 and UFC-FASTEF 31/2019.

REFERENCES

[1] Y. Zheng, "Trajectory data mining: an overview," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 6, no. 3, p. 29, 2015.

[2] 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.

[3] C. C. Aggarwal, "Outlier analysis," in *Data mining*. Springer, 2015, pp. 237–263.

[4] G. G. Hazel, "Multivariate Gaussian MRF for multispectral scene segmentation and anomaly detection," *IEEE transactions on geoscience and remote sensing*, vol. 38, no. 3, pp. 1199–1211, 2000.

[5] A. Basharat, A. Gritai, and M. Shah, "Learning object motion patterns for anomaly detection and improved object detection," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2008, pp. 1–8.

[6] L. Li, R. J. Hansman, R. Palacios, and R. Welsch, "Anomaly detection via a gaussian mixture model for flight operation and safety monitoring," *Transportation Research Part C: Emerging Technologies*, vol. 64, pp. 45–57, 2016.

[7] D. Rezende and S. Mohamed, "Variational inference with normalizing flows," in *International Conference on Machine Learning*, 2015, pp. 1530–1538.

[8] M. Yamaguchi, Y. Koizumi, and N. Harada, "AdaFlow: Domain-adaptive density estimator with application to anomaly detection and unpaired cross-domain translation," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 3647–3651.

[9] T. Iwata and Y. Yamanaka, "Supervised anomaly detection based on deep autoregressive density estimators," *arXiv preprint arXiv:1904.06034*, 2019.

[10] G. Papamakarios, T. Pavlakou, and I. Murray, "Masked autoregressive flow for density estimation," in *Advances in Neural Information Processing Systems*, 2017, pp. 2338–2347.

[11] Y. Zheng, L. Zhang, X. Xie, and W.-Y. Ma, "Mining interesting locations and travel sequences from gps trajectories," in *Proceedings of the 18th international conference on World wide web*. ACM, 2009, pp. 791–800.

[12] Y. Zheng, Q. Li, Y. Chen, X. Xie, and W. Ma, "Understanding mobility based on GPS data," in *UbiComp 2008: Ubiquitous Computing, 10th International Conference, UbiComp 2008, Seoul, Korea, September 21-24, 2008, Proceedings*, 2008, pp. 312–321. [Online]. Available: https://doi.org/10.1145/1409635.1409677

[13] Y. Zheng, X. Xie, and W. Ma, "Geolife: A collaborative social networking service among user, location and trajectory," *IEEE Data Eng. Bull.*, vol. 33, no. 2, pp. 32–39, 2010. [Online]. Available: http://sites.computer.org/debull/A10june/geolife.pdf

[14] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: identifying density-based local outliers," in *ACM sigmod record*, vol. 29, no. 2. ACM, 2000, pp. 93–104.

[15] S. Agrawal and J. Agrawal, "Survey on anomaly detection using data mining techniques," *Procedia Computer Science*, vol. 60, pp. 708–713, 2015.

[16] E. Eskin, "Anomaly detection over noisy data using learned probability distributions," in *Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000), Stanford University, Stanford, CA, USA, June 29 - July 2, 2000*, 2000, pp. 255–262.

[17] K. Leung and C. Leckie, "Unsupervised anomaly detection in network intrusion detection using clusters," in *Computer Science 2005, Twenty-Eighth Australasian Computer Science Conference (ACSC2005), Newcastle, NSW, Australia, January/February 2005*, 2005, pp. 333–342. [Online]. Available: http://crpit.scem.westernsydney.edu.au/abstracts/CRPITV38Leung.html

[18] M. Schmidt and M. Simic, "Normalizing flows for novelty detection in industrial time series data," *CoRR*, vol. abs/1906.06904, 2019. [Online]. Available: http://arxiv.org/abs/1906.06904

[19] N. Davis, G. Raina, and K. Jagannathan, "A Framework for End-to-End Deep Learning-Based Anomaly Detection in Transportation Networks," *arXiv e-prints*, p. arXiv:1911.08793, Nov 2019.

[20] C. X. Ling, J. Huang, and H. Zhang, "AUC: A better measure than accuracy in comparing learning algorithms," in *Advances in Artificial Intelligence, 16th Conference of the Canadian Society for Computational Studies of Intelligence, AI 2003, Halifax, Canada, June 11-13, 2003, Proceedings*, 2003, pp. 329–341. [Online]. Available: https://doi.org/10.1007/3-540-44886-1_25

[21] M. A. F. Pimentel, D. A. Clifton, L. A. Clifton, and L. Tarassenko, "A review of novelty detection," *Signal Processing*, vol. 99, pp. 215–249, 2014. [Online]. Available: https://doi.org/10.1016/j.sigpro.2013.12.026

[22] N. Roussopoulos, S. Kelley, and F. Vincent, "Nearest neighbor queries," in *Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data, San Jose, California, USA, May 22-25, 1995*, 1995, pp. 71–79. [Online]. Available: https://doi.org/10.1145/223784.223794

[23] D. Papadias, *Nearest Neighbor Query*. Boston, MA: Springer US, 2009, pp. 1890–1890. [Online]. Available: https://doi.org/10.1007/978-0-387-39940-9_245

[24] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 39, no. 1, pp. 1–22, 1977. [Online]. Available: https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.2517-6161.1977.tb01600.x

[25] Z. Xiao, Q. Yan, and Y. Amit, "A Method to Model Conditional Distributions with Normalizing Flows," *arXiv e-prints*, p. arXiv:1911.02052, Nov 2019.

[26] D. P. Kingma and P. Dhariwal, "Glow: Generative flow with invertible 1x1 convolutions," in *Advances in Neural Information Processing Systems*, 2018, pp. 10 215–10 224.

[27] D. J. Rezende and S. Mohamed, "Variational inference with normalizing flows," in *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, 2015, pp. 1530–1538. [Online]. Available: http://proceedings.mlr.press/v37/rezende15.html

[28] L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using Real NVP," in *Proceedings of the 5th International Conference on Learning Representations*, 2017.

[29] M. Germain, K. Gregor, I. Murray, and H. Larochelle, "Made: Masked autoencoder for distribution estimation," in *International Conference on Machine Learning*, 2015, pp. 881–889.

[30] D. P. Kingma, T. Salimans, and M. Welling, "Improving variational inference with inverse autoregressive flow," *CoRR*, vol. abs/1606.04934, 2016. [Online]. Available: http://arxiv.org/abs/1606.04934

[31] M. Gupta, J. Gao, C. C. Aggarwal, and J. Han, "Outlier detection for temporal data: A survey," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 9, pp. 2250–2267, 2014. [Online]. Available: https://doi.org/10.1109/TKDE.2013.184

[32] C.-W. Huang, D. Krueger, A. Lacoste, and A. Courville, "Neural autoregressive flows," *arXiv preprint arXiv:1804.00779*, 2018.

[33] J. B. Oliva, A. Dubey, M. Zaheer, B. Poczos, R. Salakhutdinov, E. P. Xing, and J. Schneider, "Transformation autoregressive networks," *arXiv preprint arXiv:1801.09819*, 2018.

[34] C. Durkan, A. Bekasov, I. Murray, and G. Papamakarios, "Neural spline flows," in *Advances in Neural Information Processing Systems*, 2019, pp. 7509–7520.