Semi-Supervised Anomaly Detection Via Neural Process

Fan Zhou, Guanyu Wang, Kunpeng Zhang, Siyuan Liu, and Ting Zhong

***Abstract*—Many deep (semi-) supervised neural network-based methods have been proposed for anomaly detection, tackling the issue of limited labeled data. They have shown good performance but still face two major challenges. First, insufficient labeled data limits their flexibility. Second, measuring the uncertainty of the prediction, especially when dealing with objects deviating largely from training data, has not been well studied. Another common reason preventing them from prevailing is that they learn a determined function to make predictions from the input. This usually makes the predicted results uncertain and lacks robustness. To address these problems, we propose a novel framework, incorporating the neural process into the semi-supervised anomaly detection paradigm and efficiently using unlabeled data and a handful of labeled data in training. Different from other methods, ours is equivalent to modeling the distribution of functions representing anomalous patterns according to the labeled data rather than learning a single determined function for anomaly detection. Our approach improves the flexibility and robustness under the condition of insufficient training data, and can measure the uncertainty of prediction results. Extensive experiments under real-world datasets demonstrate that our proposed method can significantly improve anomaly detection performance compared to several cutting-edge benchmarks.**

***Index Terms*—Anomaly detection, neural networks, neural process, probabilistic models, semi-supervised learning.**

I. Introduction

ANOMALIES, a.k.a. outliers, exceptions, deviants, peculiarities, etc., are data samples significantly inconsistent with the normal ones [1], [2]. The existence of anomalies in data often indicates unusual data generation or some possibly unexpected faults, which can potentially lead to severe consequences, e.g., threats to critical infrastructures. Thus, identifying anomalies that can undermine valuable information from the data from massive observations becomes a desirable and critical task and provide better actionable suggestions for subsequent decision-making. Due to its extreme importance and wide appearance in many applications, anomaly detection (AD) has been extensively studied in a broad range of domains, spanning from financial surveillance, credit fraud and online misinformation identification to epidemic control, health care, medical risk management, and network/cyber-intrusion recognition [2], [3].

Researchers have made great efforts to develop numerous anomaly detection methods. However, current studies fail to detect anomalies with high dimensionality and/or intricate relations due to the curse of dimensionality and highly non-linear feature relations [4], [5]. In recent years, we have witnessed the exceptional success of deep learning in discovering such intricate relations in high-dimensional data [6]. Meanwhile, deep learning-enabled anomaly detection, i.e., deep anomaly detection, has also emerged [3], [7]. As obtaining a large number of labeled data in practical applications becomes costly, most existing deep anomaly detection models mainly shift their focus toward unsupervised methods. In many real-world situations, however, we may have access to some verified (i.e., labeled) normal or abnormal samples in addition to the unlabeled data that could be manually labeled by domain experts.

A typical process of deep semi-supervised anomaly detection is that it first learns a parameterized function through neural networks and tunes such a function via gradient descent based on the given labeled and unlabeled training data. Then the trained model is applied to estimate the label or anomaly scores of the target data. However, existing semi-supervised anomaly detection models often face the following challenges. *First*, in semi-supervised anomaly detection tasks, limited labeled data usually provides the most prior knowledge for the model. At the same time, many learning algorithms (in particular deep neural networks) generally need a large number of labeled data to obtain a single proper function. When the training data is insufficient, the trained model is difficult to detect anomalies effectively, which limits their flexibility in real AD applications.

Second, the assumption that data within the same class (normal or abnormal) have similar patterns does not hold in the context of anomaly detection. In particular, anomalous data could vary significantly from each other in different dimensions due to various reasons. This adds extreme uncertainties to anomaly detection. Model uncertainty, also called epistemic uncertainty, means variance in a model's predictive distribution arising from the ignorance of the true model parameters for a given input [8].

Manuscript received 17 June 2022; revised 8 January 2023; accepted 8 April 2023. Date of publication 13 April 2023; date of current version 15 September 2023. This work was supported in part by National Natural Science Foundation of China under Grants 62072077 and 62176043, and in part by Natural Science Foundation of Sichuan Province under Grant 2022NSFSC0505. Recommended for acceptance by Y. Tong. *(Corresponding author: Ting Zhong.)*

Fan Zhou and Guanyu Wang are with the University of Electronic Science and Technology of China, Chengdu, Sichuan 610054, China (e-mail: fan.zhou@uestc.edu.cn; wgy05001@gmail.com).

Kunpeng Zhang is with the Department of Decision, Operations & Information Technologies, University of Maryland, College park, MD 20742 USA (e-mail: kpzhang@umd.edu).

Siyuan Liu is with the Department of Supply Chain and Information Systems, Pennsylvania State University, State College, PA 16802 USA (e-mail: siyuan@psu.edu).

Ting Zhong is with the University of Electronic Science and Technology of China, Chengdu, Sichuan 610054, China, and also with the Kashi Institute of Electronics and Information Industry, Kashi, Xinjiang 844199, China (e-mail: zhongting@uestc.edu.cn).

Digital Object Identifier 10.1109/TKDE.2023.3266755

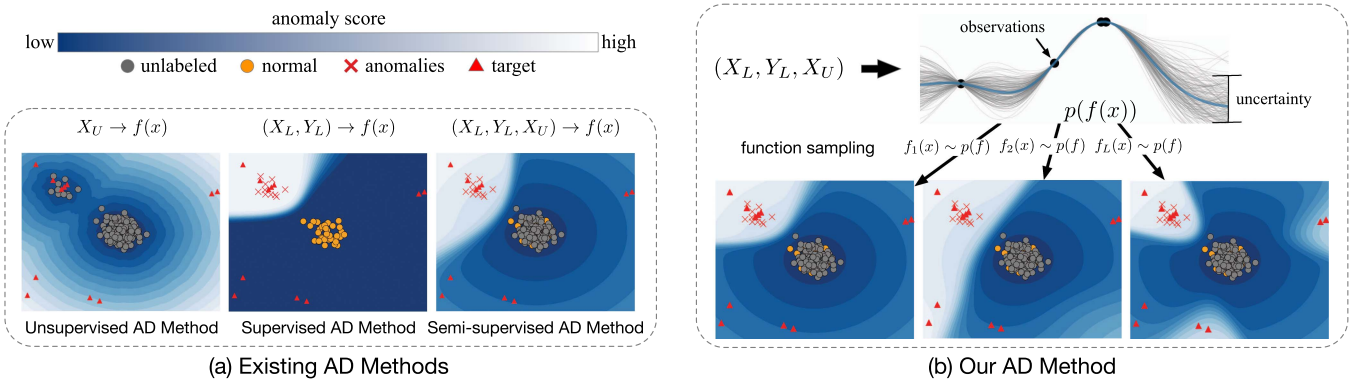


Fig. 1. Illustration of the comparisons between our method and previous anomaly detection (AD) models. We use $\mathbf{X}_U$, $\mathbf{X}_L$, and $\mathbf{Y}_L$ to denote *unlabeled* data, *labeled* data, and *labels*, respectively. The training data consists of unlabeled data (grey), labeled normal samples (orange), and anomalies (red cross). The task of anomaly detection methods is to detect target anomalies (red triangle). (a) Shows the decision boundaries of the various learning paradigms of existing anomaly detection methods at testing time. Existing models mainly detect anomalies by estimating a function f, limiting the flexibility when detecting novel anomalies. (b) Our method estimate the distribution of f to construct all possible functions that meet the observations, which allows our model to detect novel anomalies more flexibly and measure the uncertainty of predictions.

The uncertainty of the model is reflected in its confidence regarding the prediction results. Note that sometimes the confidence mentioned here is mistakenly regarded as the probability predicted by the model. For example, an AD model can be uncertain even if it clearly judges (predict with a high probability) that a data point is normal or abnormal. Although deep AD models have achieved promising results, there are usually few labeled data, and the causes of data exceptions are very different, which may easily lead to misjudgment. They are also unsure whether the results are trustworthy due to the difficulty of interpreting the estimates generated by deep neural networks. For example, in the case of classification, a model (function) produces a result that might be highly uncertain. In this situation, we often seek help from human beings for accurate classification. This is very common in the medical domain, especially when the training data is insufficient. The learned model can make incorrect judgments and requires physicians to correct them. Thus, learning one determined function through the training data and directly applying it to the unseen testing set may not be appropriate and can even result in completely wrong predictions. Because one fixed "so-called" optimal function has limited expressive power to capture these uncertainties, even the function is realized by deep models. Fortunately, researchers have developed some statistical models to address this issue, such as the Gaussian process (GP) [9]. However, with the increase in data volume and dimensionality, it is computationally costly to learn an optimal GP. In addition, the approximating ability of GP-based models is limited and cannot break through the performance bottleneck compared with deep neural networks.

Neural processes (NPs) [10], which refers to a series of supervised learning methods and combines the advantages of neural networks with the Gaussian process, is capable of rapid adaptation to new observations and, therefore, might be a potential solution to address above-mentioned uncertainties. It requires certain modifications for anomaly detection tasks, given that NP was originally designed for supervised classification tasks. In particular, it does not leverage unlabeled data effectively for training, and its objective is to separate two groups in binary classification. In anomaly detection tasks, we need to not only distinguish normal from abnormal groups but also estimate the uncertainty of predictions, i.e., multiple predicted anomaly scores from sampled posterior distributions rather than one score from one fixed function.

In this article, we introduce a novel NP-based anomaly detection (AD) framework. Unlike the existing AD methods, we use the observational data to learn a distribution P over functions $f : \mathbf{X} \to \mathbf{Y}$ rather than one single function while capturing the uncertainty over predictions. As shown in Fig. 1(a), the existing models approximate a single function with limited training data. As the parameters of a trained model are fixed, the model is not flexible enough to detect anomalies different from those in the training set. As shown in Fig. 1(b), our method learns the distribution over functions, i.e., all possible functions with the fit of limited observations. We can sample multiple functions to perform anomaly detection. Each function has the same detection result for the observed training data but may generate different results for the test data, which can help measure the uncertainty of the detection. Our method not only fully leverages the prior knowledge provided by the training data but also provides more flexibility to capture uncertainties for better anomaly detection.

We further instantiate our framework into a model called SNPAD. SNPAD is composed of three neural networks: the first and second networks are used to learn the distribution P and latent features of input data, respectively, and the third network is a discriminator that decides whether the input data is normal or abnormal with the sampled functions $f \sim P$ and the learned latent features. All these networks can be easily implemented by deep neural networks. In contrast to other NP-based probabilistic models, our method can leverage both labeled and unlabeled data for training and produces anomaly scores for each data instance with uncertainty accounted for. In other words, our model can be easily adapted into supervised, unsupervised and semi-supervised learning scenarios.

Overall, we make the following contributions:

- We propose a novel anomaly detection method based on NPs. Compared with other methods, we combine the flexibility of NPs when there are few training samples, and our method is more reasonable when used in anomaly detection

tasks. To our knowledge, this is among the first NP-based method for deep semi-supervised anomaly detection.
- A novel anomaly detection method, namely SNPAD, is instantiated from our proposed framework. SNPAD can effectively detect anomalies, and can estimate the uncertainty w.r.t. the predictions. SNPAD is composed of several fully connected neural networks, making our method computationally efficient during training and evaluation.
- Extensive empirical results on multiple benchmark datasets demonstrate that (i) SNPAD outperforms the state-of-the-art benchmark methods in terms of two standard evaluation metrics: AUC-ROC and Recall; and (ii) SNPAD is consistently more robust than other AD methods, and can accurately estimate the uncertainty of anomaly predictions.

The remainder of this article is organized as follows. Section II gives a detailed literature review of related work. In Section III, we present the details of our proposed framework SNPAD. We evaluate SNPAD on several benchmark datasets and compare it to the state-of-the-art methods while examining the model interpretability, robustness and parameter sensitivity. Section V concludes this work and points out potential future directions.

II. Related Work

Our work is mainly related to literature in three aspects: anomaly detection, Gaussian process, and neural process.

A. Anomaly Detection

Many studies are focusing on different aspects of anomaly detection and particular data structures, such as graph AD [11], time series AD [12], and tabular AD [13]. Since this work proposes a general AD model, we only review the most relevant studies.

Traditional anomaly detection methods aim to detect anomalies based on the intrinsic properties of the data. They can be divided into several categories, including distance-based (e.g., KNN [14]), density-based (e.g., LOF [15]), statistics-based (e.g., COPOD [16]), ensemble learning-based (e.g., IForest [17], RHF [18] and XGBOD [19]), and learning-based models (e.g., one-class SVM [20]). Although effective, traditional anomaly detection methods have not achieved satisfactory results due to the curse of dimensionality and the deficiency in capturing non-linear relations.

Deep anomaly detection models are widely studied in recent years, most of which are unsupervised [21], [22], [23], [24], [25] due to the limited labeled data in most real-world anomaly detection applications. A few works also focus on semi-supervised AD [26], [27], [28], [29], [30]. They primarily fall into two categories: having labeled normal samples only versus both labeled normal and labeled abnormal samples are available. Unlike conventional semi-supervised models in binary classification tasks where the data within a class are assumed similar, the abnormal ones might be very different, i.e., anomalies arise due to various reasons. Therefore, semi-supervised deep AD approaches are usually redesigned to find a compact description of the normal class while correctly discriminating the labeled anomalies.

B. Gaussian Process and Model Uncertainty

Gaussian Process (GP) is a well-known probabilistic methodology under the Bayesian framework. It has been successfully applied to many regression and classification tasks [9], [31]. Similar to neural networks, GP is a trending topic in the field of machine learning. The objective of neural network models is to fit a single function from a large amount of data. On the contrary, GP models are to learn a distribution over a family of functions, which are constrained by an assumption on the functional form of the covariance between two data points. Given the prominence of neural networks and GP models witnessed by many recent tasks, combining the two might be a worthwhile pursuing direction for learning tasks. Methods like [32], [33] are quite close to GPs, but incorporate neural networks as a 'pre-process' component for the input data. Deep GPs share some commonalities with neural networks at the concept level, as they stack GPs to obtain learning powers [34], [35]. MC dropout method [36] developed a new theoretical framework casting dropout training in deep neural networks as approximate Bayesian inference in deep GPs. In the context of anomaly detection, some methods have incorporated GPs and achieved decent performance [37], [38]. However, GPs are computationally expensive, and the available kernels are usually restricted by their functional forms.

C. Neural Process

Neural Process (NP) is a class of neural latent variable methods that model the distribution over functions [10], [39]. NP combines the strength of neural networks and the Gaussian Process, which allows the deep model to approximate a collection of labeled data flexibly with high precision. NP mainly concentrates on low-dimensional function regression and uncertainty estimation [40]. Meanwhile, there have been a growing number of studies on improving the expressiveness of the vanilla NP model. For example, ANP [41] introduces a self-attention NP to alleviate the underfitting of NP; CONVCNP [42] models the translation equivariance in data and extends task representation into a function space. SNP [43] incorporates a temporal state-transition model of stochastic processes and extends its modeling capabilities to dynamic stochastic processes.

NP has been successfully applied to various tasks, including regression, classification, and image completion. However, there are few NP methods for anomaly detection. To our knowledge, our method is among the first study that leverages the principle of NP to address the challenges in anomaly detection.

III. Methodology

In this section, we first formally define the studied anomaly detection (AD) problem and describe how to tackle it with existing deep anomaly detection models from the view of the neural

process that motivates this paper. Subsequently, we present the details of the proposed model SNPAD and its analysis.

A. Problem Statement

This work studies the semi-supervised anomaly detection problem [27], [28]. Given a set of n samples $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, where $\mathbf{x}_i \in \mathbb{R}^D$ is a D-dimensional data instance. Assume that we can access to a limited number m of labeled samples $\mathbf{X}_L = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m\}$ in addition to the massive unlabeled data $\mathbf{X}_U = \{\mathbf{x}_{m+1}, \mathbf{x}_{m+2}, \ldots, \mathbf{x}_n\}$, i.e., $\mathbf{X} = \mathbf{X}_L \cup \mathbf{X}_U$ and $m \ll n$. We use $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n\}$ to denote the true labels of all the samples but the labels for $\mathbf{X}_U$ are unavailable. For the labeled data $\mathbf{X}_L$, we have the corresponding labels $\mathbf{Y}_L = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_m\}$ with $\mathbf{y}_i \in \{-1, \tilde{}1\}$. For convenience, we use $\mathbf{O} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_m$ to denote the pairs of labeled observations.

Anomaly detection aims to learn a model $f : \mathbf{X} \to \mathbb{R}$ that assigns anomaly scores $\mathbf{s}_{1:n}$ to the data $\mathbf{X}$, where a higher score means a higher probability of anomaly, i.e., $\|\mathbf{s}_i\|_2^2 > \|\mathbf{s}_j\|_2^2$ if $\mathbf{x}_i$ is an anomaly and $\mathbf{x}_j$ is a normal data [27], [28].

B. Motivation

Existing anomaly detection models only learn one single fixed model f, which may not be able to learn correct parameters for unknown anomalous data due to the insufficient labeled data and the performance bottleneck caused by data distribution. To address this issue, we introduce the neural processes framework to model the AD function distribution, which allows us to sample various anomaly scoring functions for AD and estimate the prediction uncertainty while improving model robustness.

According to the Kolmogorov Extension Theorem [44], a pair of input data $\mathbf{x}_i \in \mathbf{X}$ and its label $\mathbf{y}_i \in \mathbf{Y}$ is associated with an instantiation (f) of a stochastic process (P) from which the probability that $\mathbf{x}_i$ is anomalous can be determined [9]:

$$p(\mathbf{y}_{1:n}|\mathbf{x}_{1:n}, \mathbf{O}) = \int p(f|\mathbf{O}) \prod_{i=1}^{n} p(\mathbf{y}_i|f, \mathbf{x}_i) df, \tag{1}$$

which can be reformulated according to the Bayes' rule as:

$$\begin{aligned} &p(\mathbf{y}_{1:n}, \mathbf{s}_{1:n} \mid \mathbf{x}_{1:n}, \mathbf{O}) \\ &= p(\mathbf{y}_{1:n} \mid \mathbf{s}_{1:n}, \mathbf{x}_{1:n}, \mathbf{O}) p(\mathbf{s}_{1:n} \mid \mathbf{x}_{1:n}, \mathbf{O}) \\ &= p(\mathbf{y}_{1:n} \mid \mathbf{s}_{1:n}, \mathbf{x}_{1:n}, \mathbf{O}) \int p(\mathbf{s}_{1:n} \mid f, \mathbf{x}_{1:n}, \mathbf{O}) p(f|\mathbf{O}) df \\ &= p(\mathbf{y}_{1:n} \mid \mathbf{s}_{1:n}) \int p(\mathbf{s}_{1:n} \mid f, \mathbf{x}_{1:n}) p(f \mid \mathbf{O}) df. \end{aligned} \tag{2}$$

Since the labels $\mathbf{y}_i$ can be easily determined by AD score $\mathbf{s}_{1:n}$, we will focus on how to learn the anomaly scores and the distribution of f from the data.

To sum up, we perform anomaly detection in a two-step manner. First, we regard the latent anomaly scores as a stochastic process that depends on the input data. Then we use neural networks to learn the latent representations for scoring functions $f \sim P$ and the input data $\mathbf{X}$. The second step is to generate anomaly scores for prediction with the learned representations.

C. SNPAD

In the following, we will introduce how to calculate the anomaly scores based on labeled observations $\mathbf{O}$ and complete data $\mathbf{X}$ via maximizing the likelihood $p(\mathbf{s}_{1:n}|\mathbf{x}_{1:n}, \mathbf{O})$. Then, we provide the training objectives of SNPAD for both labeled and unlabeled data, followed by the model architecture for implementing SNPAD.

We consider the predictions as a stochastic process depending on the input data:

$$\begin{aligned} p(\mathbf{s}_{1:n}|\mathbf{x}_{1:n}, \mathbf{O}) &= p(f_\omega(\mathbf{x}_{1:n}) = \mathbf{s}_{1:n}|\mathbf{x}_{1:n}, \mathbf{O}) \\ &= \int p(\mathbf{s}_{1:n}|\omega, \mathbf{x}_{1:n}) p(\omega|\mathbf{O}) d\omega, \end{aligned} \tag{3}$$

where ω denotes the parameters of f. This can be achieved by Gaussian Process (GP) [9] that designs a kernel function to construct the covariance matrix among input data and calculates the Gaussian distribution of $p(\omega|\mathbf{O})$ with the labeled observations $\mathbf{O}$. Despite its promising utility in estimating function uncertainty, GP is computationally expensive in high-dimensional scenarios.

Motivated by recent advances in Neural Process (NP) [9], we use neural networks to obtain a high-dimensional vector $\mathbf{r}$ and parameterize the anomaly scoring functions. In this way, the latent variables $\mathbf{r}$ that capture the model uncertainty determines the randomness of the neural networks. Besides, it allows us to sample one f for AD prediction rather than using fixed anomaly scores each time.

As computing the integral of model parameters ω in (3) becomes to learn the latent representation $\mathbf{r}$ of the AD functions, we have the following evidence lower bound (ELBO) according to the variational Bayes:

$$\begin{aligned} &\log p(\mathbf{s}_{1:n} \mid \mathbf{x}_{1:n}, \mathbf{O}) = \log \int p(\mathbf{s}_{1:n} \mid \mathbf{x}_{1:n}, \mathbf{r}, \mathbf{O}) p(\mathbf{r} \mid \mathbf{O}) d\mathbf{r} \\ &= \log \int p(\mathbf{s}_{1:n} \mid \mathbf{x}_{1:n}, \mathbf{r}) p(\mathbf{r} \mid \mathbf{O}) d\mathbf{r} \\ &\geq \mathbb{E}_{q(\mathbf{r}|\mathbf{O})} \log \frac{p(\mathbf{r} \mid \mathbf{O}) p(\mathbf{s}_{1:n} \mid \mathbf{r}, \mathbf{x}_{1:n})}{q(\mathbf{r}|\mathbf{O})} \\ &= \mathbb{E}_{q(\mathbf{r}|\mathbf{O})} \left[\log p(\mathbf{s}_{1:n} \mid \mathbf{r}, \mathbf{x}_{1:n}) - \log \frac{q(\mathbf{r} \mid \mathbf{O})}{p(\mathbf{r} \mid \mathbf{O})} \right], \end{aligned} \tag{4}$$

where $\mathbf{s}_{1:n}$ contains the anomaly scores of both labeled and unlabeled data. Because our model train the labeled data $\mathbf{X}_L$ and unlabeled data $\mathbf{X}_U$ separately (cf. Section III-C1), we decompose above optimization objective as:

$$\begin{aligned} \log\, p(\mathbf{s}_{1:n}|\mathbf{x}_{1:n}, \mathbf{O}) &\geq \mathbb{E}_{q(\mathbf{r}|\mathbf{O})} \left[\sum_{i=1}^{m} \log p(\mathbf{s}_i|\mathbf{r}, \mathbf{x}_i) \right] \\ &+ \mathbb{E}_{q(\mathbf{r}|\mathbf{O})} \left[\sum_{j=m+1}^{n} \log p(\mathbf{s}_j|\mathbf{r}, \mathbf{x}_j) \right] - \mathbb{E}_{q(\mathbf{r}|\mathbf{O})} \left[\log \frac{q(\mathbf{r}|\mathbf{O})}{p(\mathbf{r}|\mathbf{O})} \right], \end{aligned} \tag{5}$$

where the first and second terms are used to train the labeled data $\mathbf{X}_L$ and unlabeled data $\mathbf{X}_U$, respectively, and the third term is

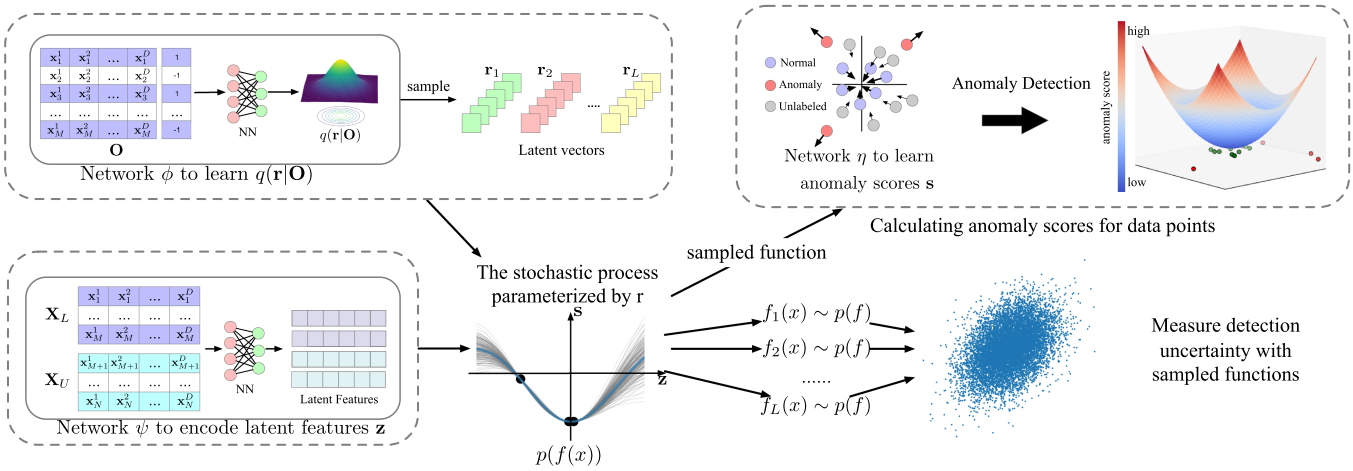


Fig. 2. The overall architecture of implementing SNPAD. We use a neural network ϕ to learn the latent representation vectors $\mathbf{r}$ used to sample the AD functions. The autoencoder network ψ encodes the latent features of both labeled and unlabeled data. Finally, another neural network η learns the anomaly scores for anomaly detection.

Algorithm 1: Training of SNPAD.

Input:
Training data $\mathbf{X} = \mathbf{X}_L \cup \mathbf{X}_U$ and labels $\mathbf{Y}_L$;
Output:
Predicted anomaly scores $\mathbf{s}$ for input data;
Learned optimal model parameters.
1: **for** each batch **do**
2: Learn posterior $q(\mathbf{r}|\mathbf{O})$ via network ϕ;
3: Sample $\mathbf{r}$ from $q(\mathbf{r}|\mathbf{O})$;
4: Learn latent features $\mathbf{z}$ via network ψ;
5: Compute anomaly scores $\mathbf{s}$ with $\mathbf{r}$ and $\mathbf{z}$ via η;
6: Calculate the log-likelihood loss $\mathcal{L}$ via (8);
7: Update SNPAD model via Adam.
8: **end for**

the KL divergence between the true and learned distributions regarding the function latent space $\mathbf{r}$.

1) Training SNPAD: Now we discuss the details of training the labeled and unlabeled data, and provide the overall training objective of our model. Algorithm 1 outlines the process of training SNPAD.

Training on $\mathbf{X}_L$. Recall that our model increases $||\mathbf{s}_i||_2^2$ of anomalous data $\mathbf{x}_i$ while reducing $||\mathbf{s}_j||_2^2$ of normal data $\mathbf{x}_j$. This goal is achieved by learning the AD function distribution from the labeled data $\mathbf{X}_L$. Let L denote the number of $\mathbf{r}$ sampled from $q(\mathbf{r}|\mathbf{O})$, the likelihood term of labeled set in (5) can be reformulated as:

$$\mathcal{L}_1 = -\frac{1}{m}\sum_{i=1}^{m}\sum_{l=1}^{L}(||\mathbf{s}_i^l||_2^2)^{\mathbf{y}_i} \propto \mathbb{E}_{q(\mathbf{r}|\mathbf{O})}\left[\sum_{i=1}^{m}\log p(\mathbf{s}_i|\mathbf{r},\mathbf{x}_i)\right], \tag{6}$$

where $\mathbf{y}_i$ denotes the label, i.e., $\mathbf{y}_i = 1$ for a normal data and $\mathbf{y}_i = -1$ for an anomaly, and the anomaly score $\mathbf{s}_i^l$ is output by the scoring network η.

Training on $\mathbf{X}_U$. As for unsupervised learning on unlabeled data $\mathbf{X}_L$, we train the model to reduce the anomaly scores of all unlabeled data, as the anomaly scores of normal data should drop faster than anomalies [13], [23], [27]. Therefore, the likelihood term of the unlabeled set in (5) can be rewritten as:

$$\begin{aligned}\mathcal{L}_2 &= \frac{1}{(n-m)L}\sum_{j=m+1}^{n}\sum_{l=1}^{L}||\mathbf{s}_j^l||_2^2 \\ &\propto -\mathbb{E}_{q(\mathbf{r}|\mathbf{O})}\left[\sum_{j=m+1}^{n}\log p(\mathbf{s}_j|\mathbf{r},\mathbf{x}_j)\right].\end{aligned} \tag{7}$$

Overall Objective. Thus, the goal of SNPADis to minimize the following loss function:

$$\mathcal{L} = \mathcal{L}_1 + \alpha\cdot\mathcal{L}_2 - \beta\cdot D_{\mathrm{KL}}\left(q\left(\mathbf{r}|\mathbf{O}\right)\|p\left(\mathbf{r}|\mathbf{O}\right)\right), \tag{8}$$

where D_{KL} is the KL divergence between distributions $p(\mathbf{r}|\mathbf{O})$ and $q(\mathbf{r}|\mathbf{O})$ in (5), α and β are the hyper-parameters controlling the weight of different terms.

2) Implementation: We implement the proposed framework using three neural networks as illustrated in Fig. 2. The *first* network ϕ is used to capture the posterior distribution $q(\mathbf{r}|\mathbf{O})$ given the labeled observations $\mathbf{O}$. The *second* network ψ is designed as a feature learner to extract latent features $\mathbf{z}$ from the input data $\mathbf{X}_L \cup \mathbf{X}_U$. The *third* network η combines the sampled $\mathbf{r}$ and latent features $\mathbf{z}_i$ to learn anomaly scores and perform anomaly detection.

Learning Posterior $q(\mathbf{r}|\mathbf{O})$. Specifically, we encode $\mathbf{r}$ as a Gaussian distribution, where $\mu_{\mathbf{r}}$ and $\sigma_{\mathbf{r}}$ are the mean and variance of $\mathbf{r}$. It allows us to sample different $\mathbf{r}$ with the reparameterization trick [45]:

$$\mathbf{r} = \mu_{\mathbf{r}} + \sigma_{\mathbf{r}}\cdot\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I}_K). \tag{9}$$

Learning Latent $\mathbf{z}$ *of Data.* We use an autoencoder neural network ψ to encode the latent representations of the data $\mathbf{X}$ (both labeled and unlabeled). The learned latent vectors $\mathbf{z}$ contain the

features of data and provide concise information for generating anomaly scores.

Anomaly Scoring and Detection. Network η is to learn the anomaly scores $\mathbf{s}_{1:n}$ for the data, which is parameterized by the learned latent function representation $\mathbf{r}$ and the latent features $\mathbf{z}_i$ as:

$$\mathbf{s}_i = \eta(\mathbf{r}, \mathbf{z}_i; \mathcal{W}_\eta), \tag{10}$$

where $\mathcal{W}_\eta$ denote the parameters.

Our model finally outputs a T-dimensional vector $\mathbf{s}_i$ for each data as the anomaly score instead of directly predicting the label $\mathbf{y}_i$ for each data $\mathbf{x}_i$ following previous deep AD models [23], [27]. During training, the model tries to maximize the discrepancy of anomaly scores between abnormal samples and normal ones, i.e., $||\mathbf{s}_i||_2^2 > ||\mathbf{s}_j||_2^2$.

D. Model Analysis

1) Interpretability: Like other AD methods, SNPAD produces an anomaly score for each data point in the test set. However, as observations are usually insufficient in AD tasks, previous deep models might generate inaccurate anomaly scores for data objects far from the training data. In addition, most AD methods cannot capture model uncertainty. In contrast, our model is sampling-based, which constructs the posterior distribution $q(\mathbf{r}|\mathbf{O})$ with the limited labeled data, and samples $\mathbf{r} \sim q(\mathbf{r}|\mathbf{O})$ to train more robust AD model. The sampled $\mathbf{r}$ capture different global "anomaly patterns", enabling SNPAD to be adaptive in various anomaly distributions. As anomaly scores constitute the measurement of the prediction results of the model, SNPAD is able to measure the uncertainty of the anomaly detection results while assessing the difficulty of detecting anomalies in the data. For example, SNPAD may estimate an accurate anomaly score with high confidence for testing data whose distribution is identical to the training data. On the contrary, SNPAD will have larger uncertainty regarding the AD predictions if the i.i.d. assumption does not hold. We will provide experimental results in the next section.

2) Complexity: Our model displays some fundamental properties of GPs with several fully connected neural networks. More importantly, our model performs anomaly detection in a computationally efficient way. The core computation of training SNPAD is the forward and backward propagation: (1) given M labeled observations, the network ϕ for estimating $q(\mathbf{r}|\mathbf{O})$ consumes $\mathcal{O}(M)$; (2) given N training points, the network ψ for learning latent features $\mathbf{z}$ consumes $\mathcal{O}(N)$; and (3) with $\mathcal{L}$ sampled $\mathbf{r}$ and learned $\mathbf{z}$, network η consumes $\mathcal{O}(LN)$ for estimating anomaly features $\mathbf{s}$. Therefore, the running time complexity of our SNPAD is $\mathcal{O}(n_{\text{epoch}} * n_{\text{batch}} * (M + N + LN))$, where n_{epoch} and n_{batch} refer to the number of training epochs and batch size of training the neural networks, respectively. Overall, our SNPAD is linear with the data size in contrast to the $\mathcal{O}((M + N)^3)$ running time of classic GPs.

IV. Experiments

We first describe datasets, baselines and experiment settings, and provide the experimental results and discussions.

TABLE I
Statistics of Datasets

Dataset	# Instances	# Dims.	%Outliers
Annthyroid	7,200	6	7.42%
Pima	768	8	35%
Mnist	7603	100	9.2%
WBC	278	30	5.6%
Pendigits	6870	16	2.27%
Satellite	6435	36	32%
Cardio	1831	21	9.6%
Lympho	148	18	4.1%
Speech	3686	400	1.65%
Mammography	11183	6	2.6%
Forestcover	286048	10	0.9%
Thyroid	3772	6	2.5%
Nasa	46464	9	1.89%

A. Experimental Settings

1) Datasets: Our experiments are conducted on 13 benchmark datasets from various domains. Their basic descriptive statistics are summarized in Table I.

Cardio consists of the measurements of fetal heart rate (FHR) and uterine contraction (UC) features on cardiotocograms classified by expert obstetricians. *Lympho* is a multi-class dataset with four classes, in which two classes are considered anomalies. *WBC* records the measurements for breast cancer cases, in which the data points in the malignant class are considered anomalies and those in the benign class are normal. *Thyroid* has 15 categorical and 6 real attributes, in which the hyperfunction class is treated as anomalous. *Speech* is provided by the Speech Processing Group at Brno University of Technology, which consists of 3,686 segments of English speech spoken with different accents. The majority of data corresponds to the American accent, and only 1.65% corresponds to one of seven other accents, which are treated as anomalies. The *Pima* dataset is from the UCI machine learning repository for binary classification tasks. In this dataset, all patients are females at least 21 years old of Pima Indian heritage. We predict whether or not a patient has diabetes based on specific diagnostic measures in the dataset. Note that these labels of 1's are considered outliers. *Satellite* is a multi-class classification dataset in which three classes with the smallest number of instances are treated as abnormal. *Mnist* is a dataset of handwritten digits which has over 7,000 examples. These digits have been size-normalized and centered in a fixed-size image. In this dataset, the digit-zero class is treated as usual, and over 700 digits sampled from other classes are treated as anomalies. *Optigits* is an optical recognition of the handwritten digits dataset. The instances of digits 1-9 are treated as normal, and digits 0 are considered outliers. *Mammography* is publicly available in the openML. It has 11,183 samples with 260 outliers. *Annthyroid* from the UCI machine learning repository is a classification dataset, including 15 categorical and 6 actual attributes. Its hyperfunction and subnormal classes are considered abnormal, and others are normal. *Pendigits* (Pen-Based Recognition of Handwritten Digits) is a multi-class classification dataset. It

has 10 classes and 16 integer attributes. These digits contain 250 samples from 44 writers.

For all datasets, missing values are replaced with the mean in the corresponding feature following previous studies [28], [46]. All datasets can be accessed via the following link: http://odds.cs.stonybrook.edu/.

2) Baselines: We compare our proposed method with unsupervised methods COPOD [16], iForest [17] and KNN [14], several supervised and semi-supervised methods including XGBoost [47], FCN, NP [10] and the deep AD model DeepSAD [27].

COPOD is a parameter-free unsupervised outlier detection method with high explainability and efficiency. It constructs the empirical copula through statistical methods and then predicts the tail probabilities of each data point. iForest is designed for processing high-dimensional data, which selects valuable attributes through the kurtosis coefficient and constructs an independent forest for anomaly detection. KNN is an AD method based on the distance of a point from its kth nearest neighbor. XGBoost is a classic supervised classification method that has been widely used in various AD tasks and ensure good performance. FCN is a classifier composed of three-layer fully connected neural networks. NP is a supervised model which combines neural networks and the Gaussian process for classification, which is adapted for anomaly detection here. DeepSAD is a state-of-the-art semi-supervised DAD method compared to previous semi-supervised AD models.

3) Parameter and Implementation Details: We set the number of layers for each network to three for the deep learning models, and the learning rate is 0.0003. We divide each dataset into a training set, a validation set, and a test set according to the scale of 6:2:2. Besides, we divide the training set into the labeled data and unlabeled data in a ratio of 1:4. The default setting of hyper-parameters of SNPAD are: $L = 1$, $\alpha = 0.2$, $\beta = 0.1$. We train our model with Adam [48].

Our implementation of COPOD, iForest and KNN is consistent with the APIs of PyOD, which is a comprehensive and scalable Python toolkit for detecting outlying objects in the multivariate data [49]. We keep the default setting of COPOD. As recommended in the previous study [17], we set the number of trees to 100 and the sub-sampling size to 256 for iForest. The implementation of XGBoost is available via the scikit-learn Python package. We set the number of neighbors to 5 in KNN. For XGBoost, we set the learning rate to 0.1, the number of estimators to 100, and the maximum depth to 3. FCN and NP are supervised deep models, we set the learning rate to 0.003 and the weight decay to 0.01, and the number of layers of each neural network is set to 3. For DeepSAD, we set $\lambda = 10^{-6}$ and equally weight the unlabeled and labeled examples by setting $\eta = 1$, as recommended in the original work [27].

All the models are tuned to the best performance on the validation set. For the DAD models, we set the learning rate as 0.0003 and weight decay as 0.01. The mini-batch size is probed using commonly used options, i.e., $\{8, 16, 32, 64, 128, 256, 512\}$.

4) Evaluating Metrics: We use AUC-ROC and Recall as metrics to evaluate the performance of all methods. AUC-ROC summarizes the ROC curve of true positives against false positives. AUC-ROC is commonly used in anomaly detection [16], [23], [23], [27]. Since users may pay more attention to whether anomalous data is correctly identified, we also report the Recall results to measure the anomaly detection performance.

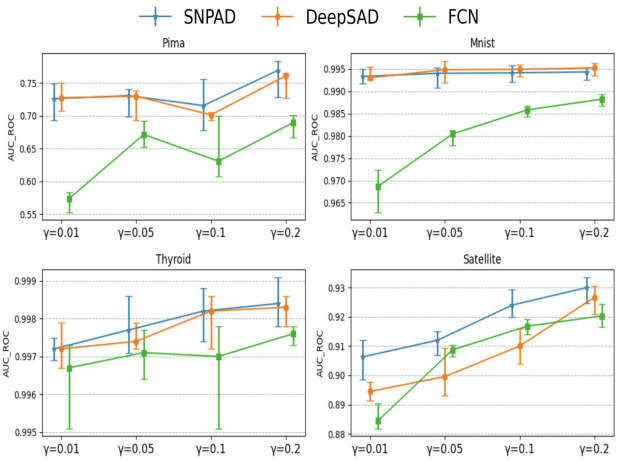


Fig. 3. Results w.r.t different ratios of labeled data. We increase the ratio of labeled data γ in the training set and report the corresponding avg. AUC-ROC performance on *Pima*, *Mnist*, *Thyroid* and *Satellite*.

B. Performance Comparisons

Table II summarizes the performance comparison between our SNPAD and baselines on the benchmark datasets, which demonstrates that our proposed model SNPAD achieves the best performance in most datasets regarding AUC-ROC. Notably, limited by the unsupervised anomaly detection paradigm, COPOD, iForest and KNN have the worst performance due to the lack of prior knowledge. Among the supervised classifiers, XGBoost has achieved ideal results on most datasets, and even the best on *Satellite* but cannot perform well on *Lympho* and *Speech*. Benefiting from the non-linear structure, deep supervised classifiers can accurately perform anomaly detection. On most datasets, NP performs better than FCN, which shows the effectiveness of the NP framework in anomaly detection, although it is usually inferior to DeepSAD and our SNPAD. Compared with unsupervised and supervised methods, the semi-supervised methods DeepSAD and SNPAD achieve better performance, mainly because they can better utilize unlabeled data for anomaly detection. Our model outperforms DeepSAD in most cases because our method takes advantage of the neural process, which proves that our approach is suitable for semi-supervised anomaly detection.

Table III shows the recall performance of anomalies on several datasets, which further demonstrates the effectiveness of our SNPAD in detecting abnormal data.

C. Impact of the Labeled Data

Fig. 3 shows the AUC-ROC results w.r.t different ratios γ of available labeled anomalies. The performance of our method and baselines generally increases with an increasing ratio of labeled anomalies since more labeled data improve the model performance. In *Pima*, surprisingly, the performance of the

TABLE II
AUC-ROC RESULTS OF SNPAD AND BASELINES

Dataset	COPOD	iForest	KNN	XGBoost	FCN	NP	DeepSAD	SNPAD
Annthyroid	0.7861±0.01	0.8513±0.02	0.7407±0.01	0.9917±0.03	0.9902±0.02	0.9921±0.02	0.9911±0.01	**0.9925**±0.01
Pima	0.6697±0.04	0.6744±0.04	0.6491±0.03	0.7578±0.08	0.6922±0.03	0.7502±0.02	0.7644±0.02	**0.7657**±0.02
Mnist	0.7908±0.02	0.8077±0.02	0.8121±0.01	0.9897±0.01	0.9889±0.01	0.9905±0.01	**0.9916**±0.01	0.9906±0.01
WBC	0.9626±0.02	0.9079±0.04	0.9349±0.02	0.9796±0.01	0.9969±0.01	0.9957±0.01	0.9965±0.01	**1.0000**±0.00
Pendigits	0.9024±0.02	0.9403±0.02	0.7099±0.01	0.9907±0.02	0.9918±0.01	0.9988±0.01	**1.0000**±0.00	**1.0000**±0.00
Satellite	0.6684±0.02	0.7091±0.03	0.6995±0.01	**0.9528**±0.01	0.9334±0.01	0.9255±0.01	0.9123±0.01	0.9149±0.01
Cardio	0.9284±0.02	0.9270±0.01	0.7540±0.02	0.9926±0.02	0.9950±0.01	0.9968±0.01	0.9969±0.01	**0.9983**±0.01
Lympho	0.9827±0.01	0.9914±0.01	0.9353±0.03	0.8061±0.09	0.9632±0.01	0.9941±0.01	**1.0000**±0.00	**1.0000**±0.00
Speech	0.4980±0.03	0.4715±0.06	0.4981±0.03	0.6230±0.07	0.7048±0.03	0.8269±0.02	0.8486±0.02	**0.8522**±0.04
Mammography	0.9179±0.02	0.8627±0.02	0.8557±0.04	0.9465±0.02	0.9455±0.02	0.9696±0.01	0.9737±0.01	**0.9757**±0.01
Forestcover	0.8831±0.01	0.8686±0.02	0.7992±0.01	0.9999±0.01	0.9991±0.01	0.9995±0.01	**1.0000**±0.00	**1.0000**±0.00
Thyroid	0.9328±0.01	0.9813±0.01	0.9525±0.01	0.9981±0.01	0.9964±0.01	0.9969±0.01	0.9972±0.01	**0.9977**±0.01
Nasa	0.9955±0.01	0.9985±0.01	0.9134±0.01	0.9999±0.01	0.9994±0.01	0.9999±0.01	**1.0000**±0.00	**1.0000**±0.00

TABLE III
RECALL PERFORMANCE OF SNPAD AND BASELINES. FOR DIFFERENT DATA, WE SET THE DETECTION RESULTS AS ANOMALIES IN DIFFERENT RATIOS, AND EVALUATE THE RESULTS WITH RECALL METRIC

Dataset	Ratio	FCN	DeepSAD	SNPAD
Speech	0.02	0.3750	**0.6250**	**0.6250**
Thyroid	0.03	0.8919	0.9459	**0.9730**
Mnist	0.1	0.9107	0.9179	**0.9357**
Cardio	0.1	0.9428	0.9429	**0.9571**
Pima	0.3	0.5421	0.5514	**0.5701**

methods drops with more labeled data from $\gamma = 0.05$ to $\gamma = 0.1$. This phenomenon happens due to the scattered and dissimilar distributions of anomalies. When the added labeled anomalies have very different anomalous behaviors and carry information conflicting with the other labeled anomalies for optimization, they may degrade the models' detection performance. When the ratio of labeled training data is low, FCN shows poor performance, demonstrating the disadvantages of supervised classifiers in anomaly detection with few labeled data.

We can draw the conclusion that semi-supervised AD methods perform better than supervised AD methods when the labeled training data is insufficient. In addition, our method is more flexible and achieves promising performance, especially when the ratio of labeled data is low. This is because the proposed NP-based method can effectively estimate the detection function distribution rather than relying on a single AD model as in previous models.

D. Robustness

We now investigate the robustness of AD methods by increasing the pollution ratio γ_p of the training data with unlabeled anomalies. We vary the contamination level from 0 to 0.2, with the ratio of available labeled training data fixed to 0.1. We conduct this experiment on SNPAD, DeepSAD and unsupervised method iForest.

The AUC-ROC performance w.r.t different anomaly contamination levels are presented in Fig. 4. According to the results, the performance of all these anomaly detection methods decreases with the increasing pollution ratio in most datasets. However,

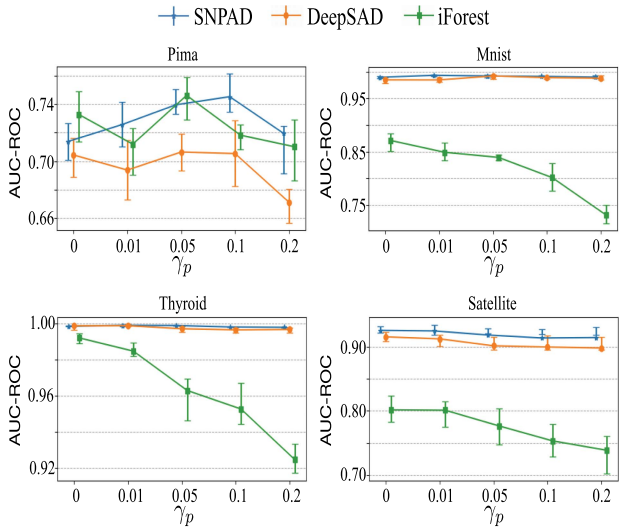


Fig. 4. Results w.r.t different contamination rates. We report the average AUC-ROC performance at various pollution ratios γ_p from 0 to 0.2 on *Pima*, *Mnist*, *Thyroid* and *Satellite*.

SNPAD is consistently more robust than other methods. When γ_p is large, for example, iForest becomes incomparable. By contrast, SNPAD is more capable of exploiting the limited prior knowledge to perform well in a noisy environment.

E. Uncertainty Measurement

Our model is designed based on NPs, which allows it to be more flexible when labeled training data is insufficient and, more importantly, measure the uncertainty of anomalies. To verify it, we specifically designed the following experiments. First, we limit the ratio of available labeled training data to 0.02. Then, we set the dimension B of latent features $\mathbf{z}$ to 1, which limits the model's performance but can effectively visualize the experimental results. Subsequently, we use the sigmoid function to set the value range of $\mathbf{z}$ from 0 to 1 and set L=100, which means we will predict 100 possible anomaly scores for all the features $\mathbf{z}$ to measure the uncertainty.

Fig. 5 shows the results of this experiment. The horizontal and vertical axes represent the magnitude of $\mathbf{z}$ and probability $p(\mathbf{Y} = -1)$, respectively. The red and green points denote $\mathbf{z}$

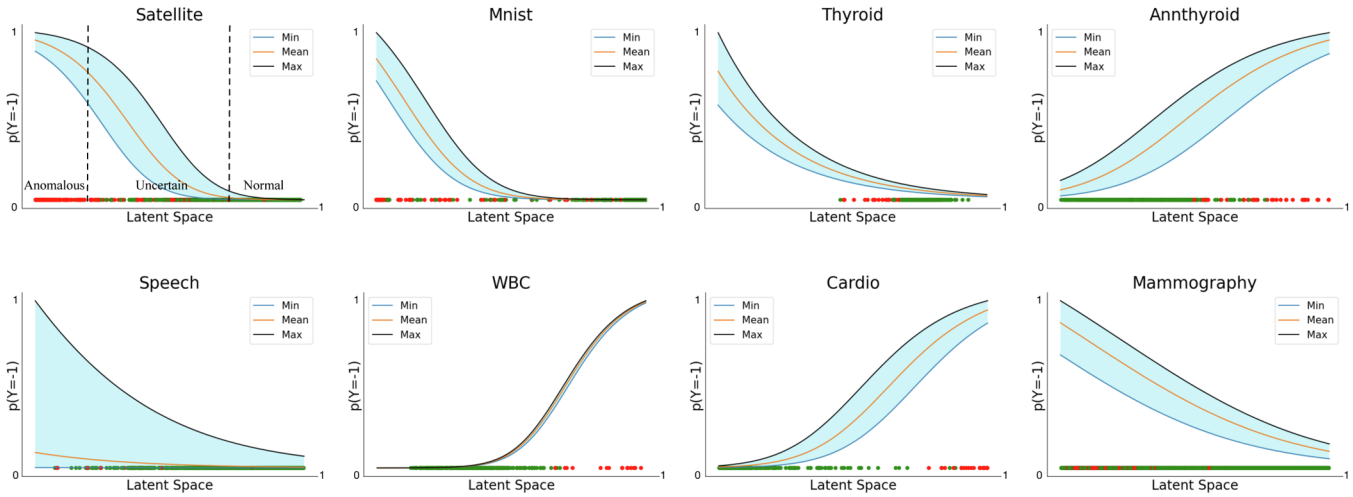


Fig. 5. Results on estimating uncertainty. We set the dimension of latent features $\mathbf{z}$ to 1 and trained our model on various datasets. We combine each feature with 100 sampled $\mathbf{r}$ and record the curve corresponding to the maximum, mean, and minimum values. The light blue area indicates the uncertainty of the prediction results. In addition, we record $\mathbf{z}$ corresponding to normal and anomalous points in the test set, which are represented by green and red points, respectively.

of anomalies and normal data in the testing set. We draw the maximum, mean, and minimum values of 100 prediction results of all the features $\mathbf{z}$ and use the blue area to represent the uncertainty of anomaly at each point. Take the *Satellite* for example, when $\mathbf{z}$ is close to 0, the corresponding probability $p(\mathbf{Y} = -1)$ of multiple detection results is significant. When $\mathbf{z}$ is close to 1, $p(\mathbf{Y} = -1)$ is small, which means that our model is quite sure that the data is abnormal if $\mathbf{z}$ is small – in contrast, the data is normal when $\mathbf{z}$ is large. When $\mathbf{z}$ is in between, different detection results differ greatly, which means that the data corresponding to this part of $\mathbf{z}$ may be normal or abnormal. In other words, the model is not confident in the detection results but can measure the value range of the probability $p(\mathbf{Y} = -1)$ for these data points. The learned $\mathbf{z}$ of test data points prove that our model can learn to estimate the results and uncertainty simultaneously.

Note that the performance of SNPAD varies significantly on different datasets. For the data where the AD methods (not limited to ours) can accurately detect the anomalies such as *WBC* (cf. Table II), our model is quite sure about the prediction results, i.e., with very low uncertainty (or high confidence) w.r.t. the prediction results. For the case where it is difficult to identify the abnormal data, e.g., *Speech*, which has high dimensions and irregular distribution of anomalies, our method generates considerable uncertainty regarding the predictions. This result demonstrates the ability of our method to quantify the AD uncertainty in addition to the superior performance in identifying the anomaly data. This nature is very important especially for critical applications such as healthcare and autonomous driving because a reliable AD model should not only output an accurate prediction but also describe whether the outcome should be trusted.

Recall that our model uses $\mathbf{r}$ to parameterize different anomaly scoring functions. Therefore, we can sample multiple $\mathbf{r}$ to perform different anomaly detection. For each data point in *Pima* and *Satellite*, we conduct experiments using different values of

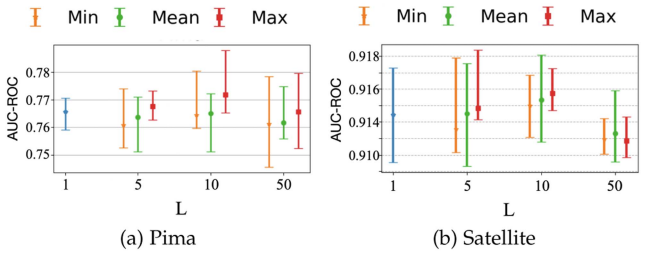


Fig. 6. Performance w.r.t different numbers of L. We report the avg. AUC-ROC with these final scores on *Pima* and *Satellite*.

$L \in \{1, 5, 10, 50\}$, and record the minimum, mean and maximum of the L detection results for each data point as the final results. Fig. 6 shows that our model performs better when we take a maximum of 10 detection results for each data point. This result indicates that our method is valuable in anomaly detection tasks, which helps to improve the detection results.

F. Effect of the Prior

We implement a variant of SNPAD, called SNPAD*, which replaces the sampling process in SNPAD. As we optimize SNPAD with the assumption that the prior distribution $p(\mathbf{r})$ is a standard normal distribution, during the testing time, we randomly sample $\mathbf{r}$ from $\mathcal{N}(0, I)$ instead of learned $\mathcal{N}(\mu_r, \sigma_r^2)$. Tables IV and V show the AUC-ROC and Recall performance comparison between SNPAD and SNPAD*. Although the performance is not better, SNPAD* still shows promising anomaly detection ability. On some datasets such as *Thyroid*, SNPAD* shows better performance, which demonstrates the rationality of estimated $q(\mathbf{r}|\mathbf{O})$.

G. Effect of the Log-Likelihood

In Section III-C1, we introduced our log-likelihood function for outputting detection results. By comparison, we present a

TABLE IV
AUC-ROC PERFORMANCE OF SNPAD AND SNPAD*. WE EVALUATE THE DETECTION RESULTS WITH THE AUC-ROC METRIC ON THE FOLLOWING VARIOUS DATASETS

Dataset	SNPAD	SNPAD*
Annthyroid	**0.9925**±0.01	0.9893±0.01
Pima	**0.7657**±0.02	0.7626±0.02
Mnist	**0.9906**±0.01	0.9842±0.01
WBC	**1.0000**±0.00	0.9982±0.01
Pendigits	**1.0000**±0.00	0.9983±0.01
Satellite	0.9149±0.01	**0.9181**±0.01
Cardio	**0.9983**±0.01	0.9974±0.01
Lympho	**1.0000**±0.00	**1.0000**±0.00
Speech	0.8522±0.04	**0.8739**±0.03
Mammography	**0.9757**±0.01	0.9528±0.01
Forestcover	**1.0000**±0.00	0.9982±0.01
Thyroid	0.9977±0.01	**0.9989**±0.01
Nasa	**1.0000**±0.00	0.9971±0.01

TABLE V
RECALL PERFORMANCE OF SNPAD AND SNPAD*

Dataset	Ratio	SNPAD	SNPAD*
Speech	0.02	**0.6250**±0.02	0.5000±0.02
Thyroid	0.03	**0.9730**±0.01	**0.9730**±0.01
Mnist	0.1	**0.9357**±0.01	0.9107±0.01
Cardio	0.1	**0.9571**±0.01	0.9429±0.02
Pima	0.3	**0.5701**±0.03	0.5607±0.03

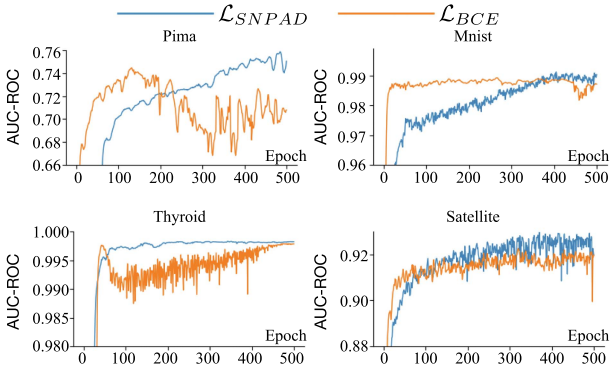


Fig. 7. Results w.r.t different log-likelihood functions of our model. We report the AUC-ROC performance of the validation set with different log-likelihood functions at the output layer.

popular choice of log-likelihood functions in most probabilistic models. The logistic log-likelihood is also a binary cross-entropy (BCE) loss for classification tasks. We make the labels of normal and anomalies as 0 and 1, respectively, and predict the label $\hat{\mathbf{y}}_i$ directly and use the logistic log-likelihood as follows:

$$\log p(\mathbf{y}_i|\mathbf{r}_i, \mathbf{x}_i) = \mathbf{y}_i \log \hat{\mathbf{y}}_i + (1 - \mathbf{y}_i) \log(1 - \hat{\mathbf{y}}_i). \quad (11)$$

Fig. 7 shows the AUC-ROC performance of validation set on four datasets with different log-likelihood functions. We can conclude that, compared with BCE loss, our method performs better and is more suitable for anomaly detection tasks.

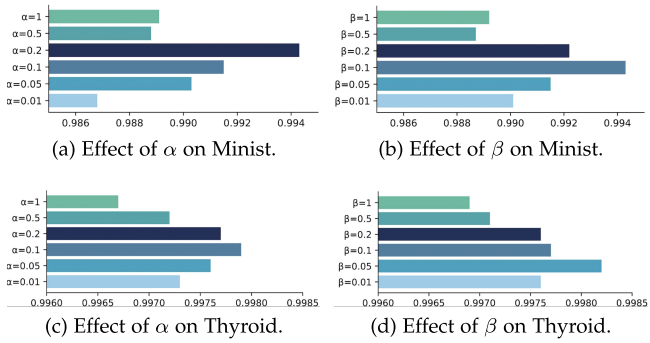


Fig. 8. SNPAD sensitivity analysis w.r.t. α and β. We report the average AUC-ROC values with standard deviations for different values of α and β.

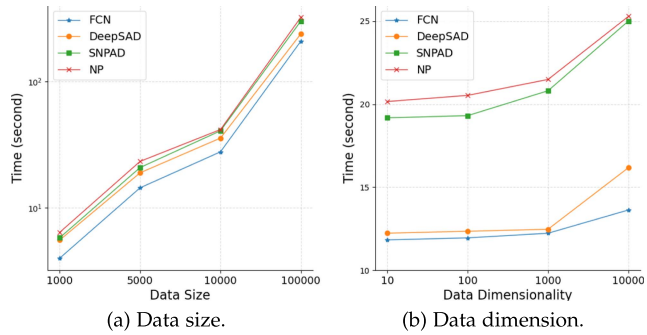


Fig. 9. Comparisons of model running time.

H. Parameter Sensitivity

We run experiments under the same experimental settings except varying the value of hyper-parameters $\alpha \in \{0.01, \ldots, 1\}$ and $\beta \in \{0.01, \ldots, 1\}$. As shown in Fig. 8, SNPAD achieves the best results at different hyper-parameters across datasets. Usually, α and β are set to a small value to weaken the influence of the unlabeled data and the prior constraint.

I. Efficiency

Now we examine the models' scalability. Following [16], [28], we generate synthesis datasets with different data sizes and dimensions and compare the running time of SNPAD and baselines, which allows us to investigate models' complexity and efficiency. Except for the varying data size and dimensions, we keep the default settings for all models.

Fig. 9 summarizes the running time of models, which shows that SNPAD has a linear time complexity w.r.t both data sizes and data dimensions as the analysis in Section III-D. Although SNPAD and NP need more time to estimate the distribution of AD functions, their time complexity is on the same order of magnitude as FCN and DeepSAD, which only evaluate a single predictor for anomaly detection. Our SNPAD is faster than NP, because the original NP splits the dataset into a context set and a target set and approximates the conditional prior using the variational posterior constructed by the context set. In SNPAD, the prior distribution $p(\mathbf{r}|\mathbf{O})$ is calculated analytically, avoiding the intensive computation for posterior estimation as in NP.

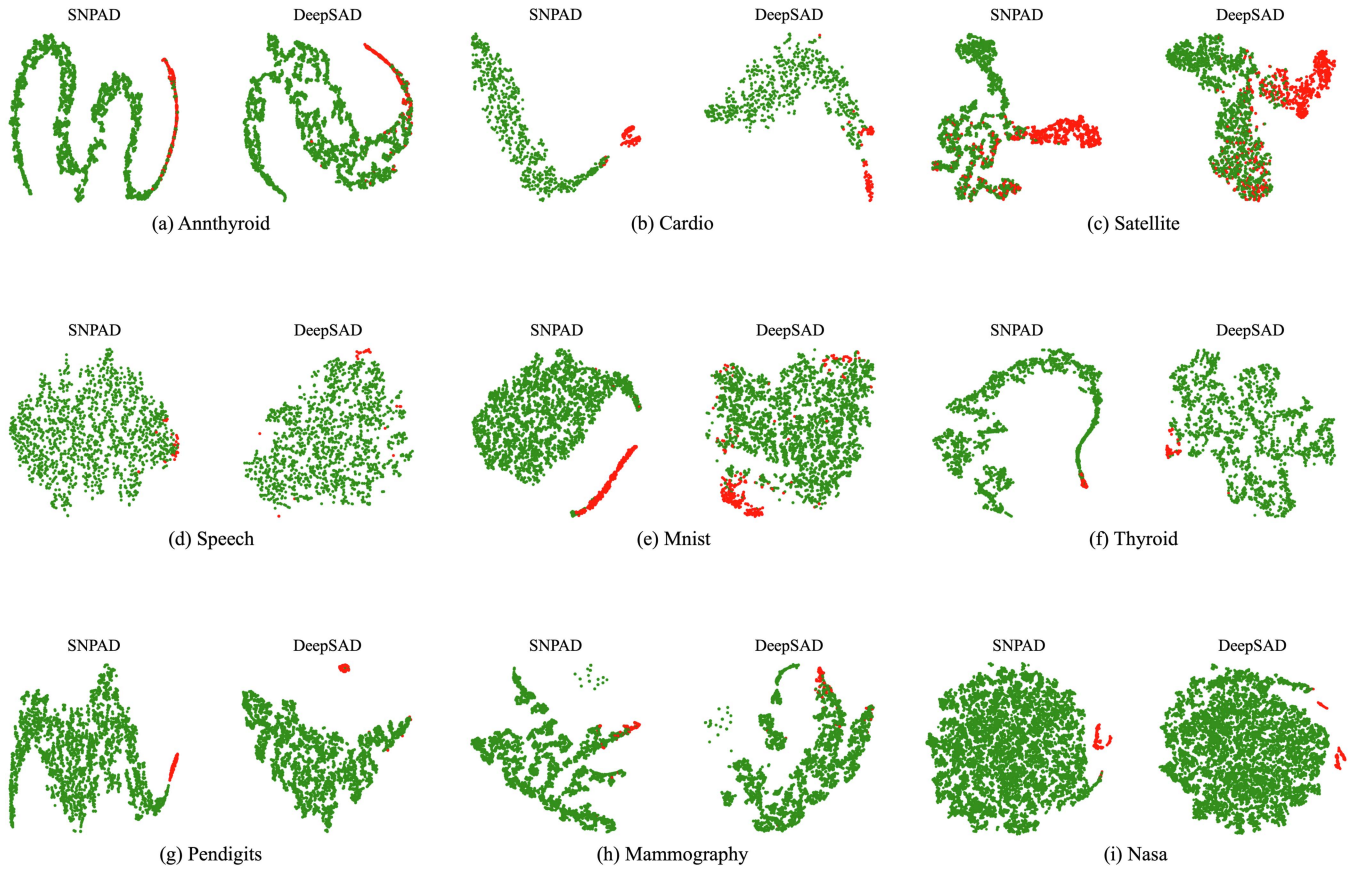


Fig. 10. Visualization of the learned latent representations on various datasets using t-SNE. Each point is the learned anomaly features of normal instances (green) and anomalies (red) in the test set. We can notice that our method, SNPAD performs better in terms of disentangling green and red points.

J. Learning Interpretability

Finally, we try to interpret the performance of SNPAD on anomaly feature representation learning. We plot the learned latent space of anomaly features for our SNPAD and DeepSAD in Fig. 10 using t-SNE [50]. The objective of AD models is to distinguish normal data (green) from anomalies (red). This experiment is conducted under the default experiment settings.

As shown in Fig. 10, given a small amount of labeled data, both models can effectively distinguish anomalies from normal instances. On some data sets, such as *Cardio*, *Mnist* and *Thyroid*, our model performs much better because the red points are clearly separated from green points. On other datasets, such as *Annthyroid*, *Speech*, and *Mammography*, our method SNPAD also makes green and red points less entangled, indicating the distinguished representations are learned well in SNPAD. However, we also note that in some cases (e.g., *Pendigits* and *Nasa*), the two models are very close due to the similar AD results achieved by the methods.

V. Conclusion

Many semi-supervised anomaly detection methods have been proposed, and all are facing one common issue: constrained by the training data to a certain extent. We presented SNPAD, a novel anomaly detection method that incorporates neural processes into the semi-supervised learning paradigm. We overcome the deficiency that most existing anomaly detection models aim to learn one anomaly scoring function by constructing a distribution over all possible functions. Our model can detect anomalies flexibly even when the training data is insufficient and can measure the uncertainty of the predictions. Extensive experiments conducted under multiple datasets have shown that our proposed model is more effective and efficient in anomaly detection than state-of-the-art methods.

References

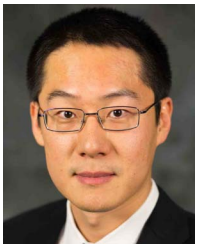
[1] W. Hu, J. Gao, B. Li, O. Wu, J. Du, and S. Maybank, "Anomaly detection using local kernel density estimation and context-based regression," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 2, pp. 218–233, Feb. 2020.
[2] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, pp. 1–58, 2009.
[3] G. Pang, C. Shen, L. Cao, and A. V. D. Hengel, "Deep learning for anomaly detection: A review," *ACM Comput. Surv.*, vol. 54, no. 2, pp. 1–38, 2021.
[4] H. Qin, X. Zhan, and Y. Zheng, "CSCAD: Correlation structure-based collective anomaly detection in complex system," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 5, pp. 4634–4645, May 2023.
[5] M. Yamada et al., "Ultra high-dimensional nonlinear feature selection for big biological data," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 7, pp. 1352–1365, Jul. 2018.
[6] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[7] L. Ruff et al., "A unifying review of deep and shallow anomaly detection," *Proc. IEEE*, vol. 109, no. 5, pp. 756–795, May 2021.
[8] H. Choi, E. Jang, and A. A. Alemi, "Waic, but why? generative ensembles for robust anomaly detection," 2018, *arXiv:1810.01392*.
[9] C. K. Williams and C. E. Rasmussen, *Gaussian Processes for Machine Learning*, vol. 2. Cambridge, MA, USA: MIT Press 2006.
[10] M. Garnelo et al., "Neural processes," in *Proc. Int. Conf. Mach. Learn. Workship*, 2018.
[11] X. Ma et al., "A comprehensive survey on graph anomaly detection with deep learning," *IEEE Trans. Knowl. Data Eng.*, early access, Oct. 8, 2021, doi: 10.1109/TKDE.2021.3118815.
[12] S. Schmidl, P. Wenig, and T. Papenbrock, "Anomaly detection in time series: A comprehensive evaluation," *Proc. VLDB Endowment*, vol. 15, no. 9, pp. 1779–1797, 2022.
[13] S. Han, X. Hu, H. Huang, M. Jiang, and Y. Zhao, "Adbench: Anomaly detection benchmark," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2022, pp. 32142–32159.
[14] S. Ramaswamy, R. Rastogi, and K. Shim, "Efficient algorithms for mining outliers from large data sets," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2000, pp. 427–438.
[15] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying density-based local outliers," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2000, pp. 93–104.
[16] Z. Li, Y. Zhao, N. Botta, C. Ionescu, and X. Hu, "COPOD: Copula-based outlier detection," in *Proc. IEEE Int. Conf. Data Mining*, 2020, pp. 1118–1123.
[17] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *Proc. Int. Conf. Des. Mater.*, 2008, pp. 413–422.
[18] A. Putina, M. Sozio, D. Rossi, and J. M. Navarro, "Random histogram forest for unsupervised anomaly detection," in *Proc. IEEE Int. Conf. Des. Mater.*, 2020, pp. 1226–1231.
[19] Y. Zhao and M. K. Hryniewicki, "XGBOD: Improving supervised outlier detection with unsupervised representation learning," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, 2018, pp. 1–8.
[20] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Comput.*, vol. 13, no. 7, pp. 1443–1471, 2001.
[21] J. Chen, S. Sathe, C. Aggarwal, and D. Turaga, "Outlier detection with autoencoder ensembles," in *Proc. SIAM Int. Conf. Data Mining*, 2017, pp. 90–98.
[22] S. Hawkins, H. He, G. Williams, and R. Baxter, "Outlier detection using replicator neural networks," in *Proc. Int. Conf. Data Warehousing Knowl. Discov.*, 2002, pp. 170–180.
[23] L. Ruff et al., "Deep one-class classification," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 4393–4402.
[24] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs, "Unsupervised anomaly detection with generative adversarial networks to guide marker discovery," in *Proc. Int. Conf. Inf. Process. Med. Imag.*, 2017, pp. 146–157.
[25] C. Zhou and R. C. Paffenroth, "Anomaly detection with robust deep autoencoders," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2017, pp. 665–674.
[26] G. Pang, L. Cao, L. Chen, and H. Liu, "Learning representations of ultrahigh-dimensional data for random distance-based outlier detection," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2018, pp. 2041–2050.
[27] L. Ruff et al., "Deep semi-supervised anomaly detection," in *Proc. Int. Conf. Learn. Representations*, 2020.
[28] G. Pang, C. Shen, and A. van den Hengel, "Deep anomaly detection with deviation networks," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 353–362.
[29] Y. Xu, J. Ding, L. Zhang, and S. Zhou, "DP-SSL: Towards robust semi-supervised learning with a few labeled samples," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2021, pp. 15895–15907.
[30] Y. Zhou, X. Song, Y. Zhang, F. Liu, C. Zhu, and L. Liu, "Feature encoding with autoencoders for weakly supervised anomaly detection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 6, pp. 2454–2465, Jun. 2022.
[31] C. E. Rasmussen, "Gaussian processes in machine learning," in *Summer School on Machine Learning*. Berlin, Germany: Springer, 2003, pp. 63–71.
[32] R. Calandra, J. Peters, C. E. Rasmussen, and M. P. Deisenroth, "Manifold gaussian processes for regression," in *Proc. Int. Joint Conf. Neural Netw.*, 2016, pp. 3338–3345.
[33] W. Huang, D. Zhao, F. Sun, H. Liu, and E. Chang, "Scalable gaussian process regression using deep neural networks," in *Proc. Int. Joint Conf. Artif. Intell.*, 2015, pp. 3576–3582.
[34] A. Damianou and N. D. Lawrence, "Deep gaussian processes," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2013, pp. 207–215.
[35] H.-M. Lu, J.-S. Chen, and W.-C. Liao, "Nonparametric regression via variance-adjusted gradient boosting gaussian process regression," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 6, pp. 2669–2679, Jun. 2021.
[36] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1050–1059.
[37] N. Li, X. Wu, H. Guo, D. Xu, Y. Ou, and Y.-L. Chen, "Anomaly detection in video surveillance via gaussian process," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 29, no. 06, pp. 1555011:1–1555011:25, 2015.
[38] K.-W. Cheng, Y.-T. Chen, and W.-H. Fang, "Video anomaly detection and localization using hierarchical feature representation and gaussian process regression," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 2909–2917.
[39] M. Garnelo et al., "Conditional neural processes," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 1704–1713.
[40] C. Louizos, X. Shi, K. Schutte, and M. Welling, "The functional neural process," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 8743–8754.
[41] H. Kim et al., "Attentive neural processes," in *Proc. Int. Conf. Learn. Representations*, 2019.
[42] J. Gordon, W. P. Bruinsma, A. Y. Foong, J. Requeima, Y. Dubois, and R. E. Turner, "Convolutional conditional neural processes," in*Proc. Int. Conf. Learn. Representations*, 2019.
[43] G. Singh, J. Yoon, Y. Son, and S. Ahn, "Sequential neural processes," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 10254–10264.
[44] B. Øksendal, "Stochastic differential equations," in *Stochastic Differential Equations*. Berlin, Germany: Springer, 2003, pp. 65–84.
[45] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *Proc. Int. Conf. Learn. Representations*, 2014.
[46] X. Zhu, S. Zhang, Z. Jin, Z. Zhang, and Z. Xu, "Missing value estimation for mixed-attribute data sets," *IEEE Trans. Knowl. Data Eng.*, vol. 23, no. 1, pp. 110–121, Jan. 2011.
[47] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 785–794.
[48] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, 2015.
[49] Y. Zhao, Z. Nasrullah, and Z. Li, "PyOD: A python toolbox for scalable outlier detection," *J. Mach. Learn. Res.*, vol. 20, pp. 96:1–96:7, 2019.
[50] L. Van Der Maaten, "Accelerating T-SNE using tree-based algorithms," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 3221–3245, 2014.

Fan Zhou received the BS degree in computer science from Sichuan University, China, in 2003, and the MS and PhD degrees from the University of Electronic Science and Technology of China, in 2006 and 2012, respectively, where he is currently a full professor with School of Information and Software Engineering. His research interests include machine learning, spatio-temporal data management, graph learning, social network mining and knowledge discovery.

Guanyu Wang received the BS degree from the University of Electronic Science and Technology of China. He is currently working toward the MS degree. His research interests include anomaly detection, recommender systems, machine learning and data mining.

Kunpeng Zhang received the PhD degree in computer science from Northwestern University, USA. He is an assistant professor with Robert H. Smith School of Business, University of Maryland, College Park. He is interested in large-scale data analysis, with particular focuses on social data mining, image understanding via machine learning, social network analysis, and causal inference. He has published papers in the area of social media, artificial intelligence, network analysis, and information systems on various conferences and journals.



Siyuan Liu received the first PhD degree from the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, and the second PhD degree from the University of Chinese Academy of Sciences. He is an assistant professor with Smeal College of Business, Pennsylvania State University. His research interests include spatial and temporal data mining, social networks analytics, and data-driven behavior analytics.

Ting Zhong received the BS degree in computer application and the MS degree in computer software and theory from Beijing Normal University, Beijing, China, respectively, in 1999 and 2002, respectively, and the PhD degree in information and communication engineering from the University of Electronic Science and Technology of China, Chengdu, China, in 2009. She is a full professor with the School of Information and Software Engineering, University of Electronic Science and Technology of China. Her research interests include machine learning, social network analysis, and mobile computing.