Anomaly Detection in High-dimensional Data Based on Autoregressive Flow

Yanwei Yu[1], Peng Lv[2], Xiangrong Tong[2], and Junyu Dong[1]

[1] Department of Computer Science and Technology, Ocean University of China
{yuyanwei,dongjunyu}@ouc.edu.cn
[2] School of Computer and Control Engineering, Yantai University
lvpeng4869@outlook.com,txr@ytu.edu.cn

Abstract. Anomaly detection of high-dimensional data is an important but yet challenging problem in research and application domains. Unsupervised techniques typically rely on the density distribution of the data to detect anomalies, where objects with low density are considered to be abnormal. The state-of-the-art methods solve this problem by first applying dimension reduction techniques to the data and then detecting anomalies in the low dimensional space. However, these methods suffer from inappropriate density estimation modeling and decoupled models with inconsistent objectives. In this work, we propose an effective Anomaly Detection model based on Autoregressive Flow (ADAF). The key idea is to unify the distribution mapping capability of flow-based models with the neural density estimation power of autoregressive models. We design an autoregressive flow-based model to infer the latent variables of input data by minimizing the combination of latent error and neural density. The neural density of input data can be estimated naturally by ADAF, along with the latent variable inference, rather than through an additional stitched density estimation network. Unlike stitching decoupled models, ADAF optimizes the same network parameters simultaneously by balancing latent error and neural density estimation in a unified training fashion to effectively separate the anomalies out. Experimental results on six public benchmark datasets show that, ADAF achieves better performance than state-of-the-art anomaly detection techniques by up to 20% improvement on the standard F_1 score.

Keywords: Anomaly detection, flow-based model, neural density estimation, deep learning.

1 INTRODUCTION

Anomaly detection is a fundamental and hence well-studied problem in many areas, such as cyber-security [26], manufacturing [19], system management [16], and medicine [7]. Anomaly detection, also known as outlier detection, is to identify the objects that significantly differ from the majority of objects in the data space. In general, normal data is large and consistent with certain distribution, while abnormal data is small and discrete; therefore anomalies are residing in low density areas.

Although great progress has been made in anomaly detection in the past few decades, anomaly detection for high-dimensional data is still a huge challenge. Due to the dimensional disaster, it is increasingly difficult for traditional density estimation models to implement density estimation in the original data space. But unfortunately for a real-world problem, the dimensionality of data could be very large. To address this challenge, a two-step framework is usually applied into high-dimensional data [5, 12]. It first performs dimensionality reduction on high-dimensional data and then detect anomalies in the low-dimensional space. In recent years, deep learning has achieved great success in anomaly detection [6]. Generative adversarial networks (GANs) [13] and autoencoder [30] and their variants have been widely used for anomaly detection, such as variational autoencoder (VAE) [1], and adversarial autoencoder (AAE) [21]. The core idea of these methods is to encode input data into a low dimensional representation, and then decode the low dimensional representation into the original data space by minimizing the reconstruction error. In this process, the essential features of the original data are extracted in latent data space through training autoencoder, without noise and unnecessary features. Several recent studies have applied this structure into practical problems. For example, DAGMM [31] combines deep autoencoder and Gaussian mixture model (GMM) in anomaly detection. However, the real-world data may not only have high dimensions, but also lack a clear predefined distribution (e.g., GMM). Manual parameter adjustment is also required in GMM when modeling the density distribution of input data, which has a serious impact on detection performance. Additionally, all these methods based on two steps have two main limitations: (1) the loss of information in original data is caused by the irreversible dimensionality reduction. (2) the decoupled models of dimensionality reduction and density estimation are easily trapped in local optima during training.

Recently, several flow-based models are proposed to generate data and have proved to be successful in many fields, such as Parallel WaveNet [20] for speech synthesis, and Glow [17] and NICE [9] for image generation. Flow-based models map original data to a latent space so as to make the transformed data conform to a factorized distribution, i.e., resulting in independent latent variables. This is a revertible non-dimensional reduction process, meaning that there is no loss of information. Compared with GANs and VAEs, which have shown great success in the field of high-dimensional data anomaly detection, flow-based models have not received much attention. Nevertheless, flow-based models possess the following advantages: First, flow-based models perform exact latent variable inference and log-likelihood evaluation. VAEs can only infer the approximate value of the latent variable corresponding to the input data point after encoding. GANs have no encoder at all to infer the latent variable. In reversible generative models like Glow [17], exact inference of latent variables can be achieved without approximation, and the exact log-likelihood of the data also can be optimized, instead of a lower bound of it. Second, flow-based models are efficient to parallelize for both inference and synthesis, such as Glow [17] and RealNVP [10]. Third, there is significant potential for memory savings. Computing gradients in reversible neural networks requires a certain amount of memory, instead of linear in their depth. The fourth is natural neural density estimation. Autoregressive models and normalizing flows are the main members of the family of neural density estimation. The neural density of input data can be estimated while inferring latent variable.

In this paper, we propose an effective Anomaly Detection method based on Autoregressive Flow-based generative model, called ADAF, which is a deep learning framework that addresses the aforementioned challenges in anomaly detection from high-dimensional datasets. ADAF is a *neural density estimation* model, which unifies the distribution mapping capacity of flow-based model with the density estimation power of autoregressive model to provide a neural density estimation of high-dimensional data for effectively identifying anomalies. First, we design an autoregressive flow-based model to infer the latent variables of input data by minimizing the combination of latent error and sample neural density. Second, neural density of input data can be estimated naturally by ADAF, which is totally different from traditional surrounding point-based density estimation. The neural density of a data point is calculated directly along with the latent variable inference and log-likelihood evaluation, rather than through an additional stitched density estimation network. Finally, ADAF is an absolute end-to-end model that optimizes both latent error and neural density estimation simultaneously for the same network parameters, which avoids getting into local optima.

We conduct comprehensive experiments on six public benchmark datasets to valuate the effectiveness of our proposed model. ADAF is significantly better than state-of-the-art methods by up to 20% improvement in standard F_1 score for anomaly detection. It is worth noting that ADAF achieves better results with fewer training samples compared to existing methods based on deep learning.

To summarize, we make the following contributions:

- We propose a deep anomaly detection model based on autoregressive flow for anomaly detection from high-dimensional datasets.
- We propose to combine the latent error and neural density together to optimize latent variable inference and log-likelihood estimation simultaneously in autoregressive flow model for effectively identifying anomalies.
- We conduct extensive evaluations on six benchmark datasets. Experimental results demonstrate that our method significantly outperforms state-of-the-art methods.

2 RELATED WORK

In recent years, varieties of studies focus on anomaly detection in data mining and machine learning [11]. Distance-based model [18] detects anomalies through global density criterion. Density-based methods [4, 27] uses local relative density as anomaly criterion to detect anomalies. Several studies [15, 25] apply KDE into density-based local outlier detection to improve the detection accuracy. However, such methods rely on an appropriate distance metric, which are only feasible for handling low-dimensional data, but not for anomaly detection of high dimensional data. One-class classification approaches trained by normal data are widely used for anomaly detection, such as one-class SVMs [8] and SVDD [22]. The core of these methods is to find a decision boundary that separates abnormal data from normal data. Another category of anomaly detection framework is mainly based on reconstruction errors to determine whether a sample is anomalous, such as conventional Principal Component Analysis (PCA), kernel PAC, and Robust PCA (RPCA) [5, 14].

Recently, varieties of anomaly detection methods based on deep neural networks are proposed to detect anomalies [6]. GANs, Autoencoder and their variants have been widely used in anomaly detection, especially for high-dimensional data anomaly detection. The variational autoencoder is used directly for anomaly detection by using reconstruction error in [1]. Inspired by RPCA [5], Zhou et al. [30] propose a Robust Deep Autoencoder (RDA), and use the reconstruction error to detect anomalies for high-dimensional data. AnoGAN [3] uses a Generative Adversarial Network [13] to detect anomalies in the context of medical images by reconstruction error. In a follow-up work, f-AnoGAN [23] introduces Wasserstein GAN [2] to improve AnoGAN to be adaptable to real-time anomaly detection applications. However, these methods only consider reconstruction errors as anomaly criterion, thus the performance of these methods is limited in detecting anomalies.

Deep structured energy based model (DSEBM) [29] directly simulates the data distribution through the deep architectures to detect data anomalies. DSEBM integrates Energy-Based Models (EBMs) with various types of datasets, including spatial data, static data, and sequential data. DSEBM has two anomaly criteria to identify anomalies: the energy score (DSEBM-e) and the reconstruction error (DSEBM-r). Deep Autoencoding Gaussian Mixture Model (DAGMM) [31] consists of a compression network and an estimation network. The compression network reduces the dimensionality of input samples through a deep autoencoder, prepares their low-dimensional representations from the reduced space and reconstruction error features, and provides the representations to the subsequent estimation network. Estimation networks take feeds and predict their likelihood/energy in the framework of a Gaussian Mixture Model (GMM). These models first reduce the dimensionality of the data, and then detect anomalies in the low-dimensional space through the energy model or GMM. As GANs are able to model the complex high-dimensional distributions of real-world data, and Adversarially Learned Anomaly Detection (ALAD) is a GAN based methods [28], which considers both data distribution and reconstruction error. ALAD derives adversarially learned features for the anomaly detection task based on bi-directional GANs, and then uses reconstruction errors based on these adversarially learned features to separate out anomalies.

Our proposed method is most related to DAGMM. However, unlike DAGMM, ADAF uses an autoregressive flow-based model to accurately extract independent latent variables. And ADAF directly obtain the neural density estimation of the original data with latent variable mapping, rather than a predefined GMM distribution. Most importantly, ADAF can independently estimate the neural density of a data point without having to rely on other constraints, such as distance or density from other data points, and show a powerful ability of anomaly detection with few training samples.

3 Autoregressive Flow-based Anomaly Detection Model

3.1 Normalizing flows

Flow refers to the data "flowing" through a series of bijections (revertible mapping), and finally maps to a suitable representation space. Normalizing means that the variable integral of the representation space is 1, which meets the definition of probability distribution function.

Given an observed data $x \in X$, an explicit invertible non-linear transformation $f : \mathbb{R}^d \to \mathbb{R}^d$ of a simple tractable distribution $p_Z(z)$ (e.g., an isotropic Gaussian distribution) on a latent variable $z \in Z$, $X = f(Z)$ and $Z = f^{-1}(X)$, the change of variable formula defines a model distribution on X by:

$$p_X(x) = p_Z(f^{-1}(x))|det(\frac{\partial f^{-1}(x)}{\partial x})|, \tag{1}$$

where $\frac{\partial f^{-1}(x)}{\partial x}$ is the Jacobian of f at x. The transformation f is typically chosen so that it is invertible and its Jacobian determinant is easy to compute.

Therefore, the probability density function of the model given a data can be calculated from a log probability:

$$\log(p_X(x)) = \log(p_Z(f^{-1}(x))) + \log(|det(\frac{\partial f^{-1}(x)}{\partial x})|). \tag{2}$$

3.2 Autoregressive density estimation

Autoregressive density estimation uses the chain rule of probability to learn the joint probability density by decomposing it into the product of one-dimensional conditional probability density. Given an observation x which contains d attributes, its joint probability density is calculated as follows:

$$p(x) = \prod_{i=1}^{d} p(x_i|x_{1:i-1}), \tag{3}$$

Formally, the generation of the variable x_i in the i-th dimension depends only on the previously generated variable $x_{1:i-1}$, that is:

$$p(x_i|x_{1:i-1}) = \mathcal{N}(x_i|\mu_i, (\exp(\alpha_i))^2),\ \mu_i = g_{\mu_i}(x_{1:i-1}),\ \alpha_i = g_{\alpha_i}(x_{1:i-1}), \tag{4}$$

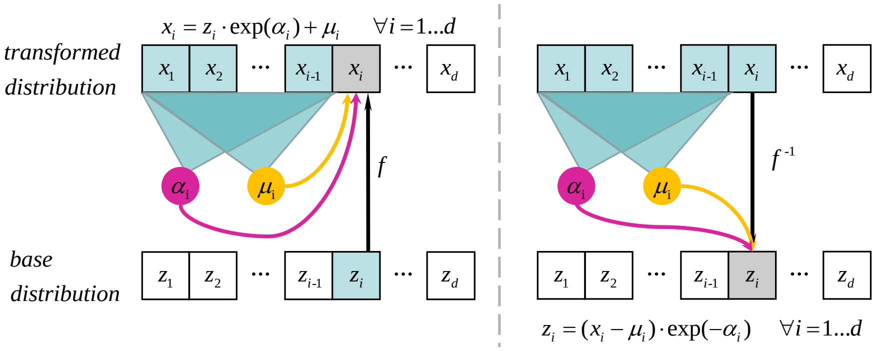

$x_i = z_i \cdot \exp(\alpha_i) + \mu_i \quad \forall i = 1...d$
transformed distribution
x_1
x_2
x_{i-1}
x_i
x_d
α_i
μ_i
f
base distribution
z_1
z_2
z_{i-1}
z_i
z_d
f^{-1}
$z_i = (x_i - \mu_i) \cdot \exp(-\alpha_i) \quad \forall i = 1...d$

Fig. 1. Specific process of single model. The gray cells are the cells that are currently being calculated, and the blue cells represent the cells on which they depend.

where g_{μ_i} and g_{α_i} are functions that compute the mean and log standard deviation of the i-th attribute given all previous variables. Autoregressive probability density has two parameters: mean μ_i and log standard deviation α_i.

We use the recursive operation of the above Eq. (3) and Eq. (4) to generate data:

$$x_i = z_i \exp(\alpha_i) + \mu_i, \ z_i \sim \mathcal{N}(0, 1), \tag{5}$$

where $z = (z_1, z_2, ..., z_d)$ is the vector of random numbers the model uses internally to generate data.

3.3 Anomaly Detection based on Autoregressive Flow (ADAF)

Single Module From Eq. (5), we can see that the autoregressive model provides an alternative characterization as a transformation f from the space of random numbers Z to the space of data X. We express this model as $X = f(Z)$. Given data point x which contains d dimensions, we can get z by the following reverse operation:

$$z_i = (x_i - \mu_i) \exp(-\alpha_i), \mu_i = g_{\mu_i}(x_{1:i-1}), \alpha_i = g_{\alpha_i}(x_{1:i-1}), \tag{6}$$

The specific process of a single module is shown in Figure 1. The figure on the left is the generation process f of x. For any distribution x_i, it is calculated from α_i, μ_i and z_i, which means that x_i depends on all previous variables (i.e., $x_1, \ldots, x_{i-1}$) and corresponding z_i. The figure on the right is the inverse generation process f^{-1} of z. For any distribution z_i, it is obtained from α_i, μ_i and x_i, which means that z_i also only depends on all previously generated variables (i.e., $x_1, \ldots, x_{i-1}$).

Because of autoregressive structure, the Jacobian of f^{-1} is triangular by design. We can calculate its absolute determinant as follows:

$$|det(\frac{\partial f^{-1}(x)}{\partial x})| = \exp(-\sum_{i=1}^{d} \alpha_i),\ \alpha_i = g_{\alpha_i}(x_{1:i-1}). \tag{7}$$

Therefore, the autoregressive model can be equivalently regarded as a normalizing flow, which can calculate density $p(x)$ by substituting Eq. (6) and (7) into Eq. (2):

$$\log(p_X(x)) = \log(p_Z(f^{-1}(x))) + \log(\exp(-\sum_{i=1}^{d} \alpha_i)). \tag{8}$$

Multiple Modules We improve the model fit by stacking multiple instances of the single model into a deeper flow:

$$x = f_K \circ ... \circ f_2 \circ f_1(z), \tag{9}$$

$$z = f_1^{-1} \circ ... \circ f_{K-1}^{-1} \circ f_K^{-1}(x), \tag{10}$$

where x is the input data for d dimensions, K is the number of single module, f_i represents an autoregressive module, z is the latent variable.

Combining Eq. (7), (8), and Eq. (10), then sample neural density can be further inferred by:

$$\begin{aligned} D(x) &= -\log(p_X(x)) \\ &= -[\log(p_Z(\prod_{k=1}^{K} f_i^{-1}(x))) + \sum_{k=1}^{K}[\log(\exp(-\sum_{i=1}^{d} \alpha_{ki}))]], \end{aligned} \tag{11}$$

where p_Z is a simple tractable distribution (e.g., an isotropic Gaussian distribution).

Objective Function Given a dataset of N instances, which contain d attributes. The objective function guides ADAF training is constructed as follows:

$$\mathcal{J}(\mu, \alpha) = \frac{1}{N}\sum_{j=1}^{N} L(x^j, z^j) + \frac{\lambda}{N}\sum_{j=1}^{N} D(x^j). \tag{12}$$

This objective function includes two components.

- $L(x^j, z^j)$ is the latent error, which is the error between input data x^j and its latent data z^j. Latent data is the key information of the input data, so we expect the value of latent error is as low as possible. In practice, we use L_2-norm for this purpose, as $L(x^j, z^j) = \|x^j - z^j\|_2^2$.

- $D(x^j)$ is the sample neural density of input data. By minimizing negative log-likelihood density estimations, we can better fit the observed data to high-density space. We optimize the combination of neural density and latent error until the two reach a equilibrium, which makes our objective function better serve the objective of anomaly detection.
- λ is the coefficient parameter in ADAF, which controls the objective to be biased towards latent error or neural density.
- $\mathcal{J}(\mu, \alpha)$, μ and α represent all related parameters μ_i and α_i in the model.

Although our objective function consists of two components, it is totally different from DAGMM. In our objective function, the latent error and the neural density together optimize the same network parameters, which is a thorough end-to-end model. DAGMM is also an end-to-end training model, but the two parts of its objective function optimize different network parts, respectively. Therefore, our model is an absolute end-to-end framework that jointly optimizes latent error and neural density estimation simultaneously. More specifically, we use stochastic gradient descent to optimize the objective during training. Finally, the latent error and the sample neural density are used as anomaly criteria to detect anomalies. That is, a data sample has a higher latent error and sample neural density value, it is more likely to be an anomaly.

4 Experiments

In this section, we use six public benchmark datasets to evaluate the effectiveness and robustness of ADAF in anomaly detection. The code of the baseline methods is available at GitHub[3] released by ALAD. The code of our ADAF can be available at GitHub[4].

Table 1. Statistics of the public benchmark datasets

Dataset	#Dimensions	#Instances	Anomaly ratio (ρ)
Thyroid	36	3,772	0.025
KDDCUP	118	494,021	0.2
SpamBase	58	3485	0.2
Arrhythmia	274	432	0.15
KDDCUP-Rev	118	121,597	0.2
Cardiotocography	22	2068	0.2

[3] https://github.com/houssamzenati/Adversarially-Learned-Anomaly-Detection

[4] https://github.com/1246170471/ADAF

4.1 Datasets

We conduct experiments on six public datasets in the field of anomaly detection: KDDCUP, Thyroid, Arrhythmia, KDDCUP-Rev, SpamBase, and Cardiotocography. The details of the datasets are shown in Table 1.

- **Thyroid**: Thyroid is from UCI Machine Learning Repository[5] thyroid disease classification dataset, which contains samples of 36 dimensions. There are 3 classes in original dataset. As hyperfunction is a minority class, we treat hyperfunction as anomaly class in our experiment.
- **KDDCUP**: The KDDCUP 10% dataset from UCI Machine Learning Repository is a network intrusion dataset, which originally contains 41 dimensions. 34 of them are continuous data, and another 7 represent categories. We use one-hot representation to encoder them, and eventually obtain a 118-dimensional dataset. As 20% of them are marked as "normal" and meanwhile others are marked as "attack", and "normal" samples constitute a small portion, therefore, we treat "normal" samples as anomalies in our experiment.
- **SpamBase**: SpamBase is from UCI Machine Learning Repository, which collects spam emails filed by postmaster and individuals and non-spam emails from filed work and personal emails. We treat the spam emails as outliers, and the anomaly ratio is 0.2.
- **Arrhythmia**: Arrhythmia dataset is also obtained from the UCI Machine Learning Repository. This dataset contains 274 attributes, 206 of them are linear valued and the rest are nominal. The smallest classes, including 3, 4, 5, 7, 8, 9, 14 and 15, are combined to form the anomaly class, and the rest of the classes are combined to form the normal class.
- **KDDCUP-Rev**: This dataset is an abbreviated version extracted from KDDCUP. We retain all "normal" data in this dataset, and randomly draw "attack" samples to keep the anomaly ratio as 0.2. As "attack" data is in minority part, we treat "attack" data as anomalies.
- **Cardiotocography**: Cardiotocography is also from UCI Machine Learning Repository which related to heart diseases. This dataset contains 22 attributes, and the instances in the dataset are classified by three expert obstetricians into 3 classes: normal, suspect, or pathological. Normal instances are treated as inliers and the remaining as outliers.

4.2 Baseline Methods

We compare our method with the following traditional and state-of-the-art deep learning methods:

- **OC-SVM** [8]: One Class Support Vector Machines (OC-SVM) is a classic kernel method for novelty detection that only use normal data to learn a decision boundary. We adopt the widely used radial basis function (RBF) kernel. In our experiments, we assume that the abnormal proportion is known. We

[5] https://archive.ics.uci.edu/ml/

set the parameter ν to the anomaly proportion, and set γ to $1/m$, where m is the number of input features.

- **DSEBM** [29]: Deep Structured Energy Based Models(DSEBM) is a deep learning method for anomaly detection. They tackle the anomaly detection problem by directly modeling the data distribution with deep architectures. DSEBM contains two decision criteria for performing anomaly detection: the energy score (**DSEBM-e**) and the reconstruction error (**DSEBM-r**).
- **DAGMM** [31]: Deep Autoencoding Gaussian Mixture Model (DAGMM) is a state-of-the-art method for anomaly detection, which consists of two major components: a compression network and an estimation network. The compression network performs dimensionality reduction for input samples by a deep autoencoder, and feeds the low-dimensional representations with the reconstruction error to the subsequent estimation network. The estimation network takes the feed, and predicts their likelihood/energy in the framework of GMM.
- **AnoGAN** [24]: AnoGAN is a GAN-based method for anomaly detection. AnoGAN is trained with normal data, and using it to recover a latent representation for each input test data. AnoGAN uses both reconstruction error and discrimination components as the anomaly criterion. Reconstruction error ensures how well the GAN is able to reconstruct the data via the generator, while the discrimination component considers a score based on the discriminator. There are two approaches for the anomaly score in the original paper and we choose the best variant in our tasks.
- **ALAD** [28]: Adversarially Learned Anomaly Detection (ALAD) is also a state-of-the-art method based on bi-directional GANs, which derives adversarially learned features for the anomaly detection task. ALAD uses reconstruction error based on these adversarially learned features to determine if a data sample is anomalous.

4.3 Experiment Configuration

The configurations of baselines used in experiments follows their original configurations. We follow the setting in [29, 31] with completely clean training data: in each run, we take $\tau\%$ of data by randomly sampling for training with the rest (1-$\tau\%$) reserved for testing, and only data samples from the normal data are used for training models. Specifically, for our ADAF and all baselines, we set τ=50 in KDDCUP and KDDCUP-Rev, τ=80 in other datasets. Without special statement, we set λ to 1 by default.

We set different K values (i.e., the number of distribution mappings) on different datasets in our network structure. K is set to 4 in KDDCUP and KDDCUP-Rev, K=8 in Cardiotocography, K=16 in Arrhythmia, K=16 in SpamBase, and K=10 in Thyroid. See our code for more detailed network structure settings.

4.4 Evaluation Metrics

We consider average precision, recall, and F_1 score to quantify the results. We choose a threshold based on the anomaly ratio in the test set. For example, if the the anomaly ratio in the test set is ρ, the top ρ data of the objective function value is marked as anomalies.

The precision and recall are defined as follows: $Precision = \frac{|G| \cap |R|}{|R|}$ and $Recall = \frac{|G| \cap |R|}{|G|}$, where G denotes the set of ground truth anomalies in the dataset, and R denotes the set of anomalies reported by the methods. F_1 score is defined as follows: $F_1 = \frac{2 * Precision * Recall}{Precision + Recall}$.

Table 2. Average precision, recall, and F_1 from ADAF and all baselines. For each metric, the best result is shown in bold.

Method	KDDCUP			Thyroid		
	Precision	Recall	F_1	Precision	Recall	F_1
OC-SVM	0.7457	0.8523	0.7954	0.3639	0.4239	0.3887
DSEBM-r	0.8744	0.8414	0.8575	0.0400	0.0403	0.0403
DSEBM-e	0.2151	0.2180	0.2170	0.1319	0.1319	0.1319
DAGMM	0.9297	0.9442	0.9369	0.4766	0.4834	0.4782
AnoGAN	0.8786	0.8297	0.8865	0.0412	0.0430	0.0421
ALAD	0.9427	0.9577	0.9501	0.3196	0.3333	0.3263
ADAF	**0.9877**	**0.9926**	**0.9901**	**0.5102**	**0.5321**	**0.5209**

Method	Arrhythmia			KDDCUP-Rev		
	Precision	Recall	F_1	Precision	Recall	F_1
OC-SVM	0.5397	0.4082	0.4581	0.7148	0.9940	0.8316
DSEBM-r	0.4286	0.5000	0.4615	0.2036	0.2036	0.2036
DSEBM-e	0.4643	0.4645	0.4643	0.2212	0.2213	0.2213
DAGMM	0.4909	0.5078	0.4983	0.9370	0.9390	0.9380
AnoGAN	0.4118	0.4375	0.4242	0.8422	0.8305	0.8363
ALAD	0.5000	0.5313	0.5152	0.9547	0.9678	0.9612
ADAF	**0.7172**	**0.7171**	**0.7171**	**0.9895**	**0.9941**	**0.9918**

Method	SpamBase			Cardiotocography		
	Precision	Recall	F_1	Precision	Recall	F_1
OC-SVM	0.7440	0.7972	0.7694	0.7366	0.6848	0.7051
DSEBM-r	0.4296	0.3085	0.3574	0.5584	0.5467	0.5365
DSEBM-e	0.4356	0.3185	0.3679	0.5564	0.5367	0.5515
DAGMM	0.9435	0.7233	0.7970	0.5024	0.4905	0.4964
AnoGAN	0.4963	0.5313	0.5132	0.4446	0.4360	0.4412
ALAD	0.5344	0.5206	0.5274	0.5983	0.5841	0.5911
ADAF	**0.8381**	**0.8393**	**0.8387**	**0.7435**	**0.7432**	**0.7433**

4.5 Effectiveness Evaluation

First, we valuate the overall effectiveness of our proposed model compared with all baseline methods on six benchmark datasets. We repeat 20 runs for all methods on each dataset and the average precision, recall, and F_1 score are shown in Table 2.

From Table 2, we can see that ADAF is significantly better than all baselines in terms of average precision, recall, and F_1 score on six datasets. On the KDDCUP and KDDCUP-Rev, ADAF achieves 4% and 2.4% improvement in standard F_1 score compared to state-of-the-art ALAD, reaching over 98% in all terms of precision, recall and F_1 score. On Thyroid and Arrhythmia, ADAF significantly performs better than state-of-the-art DAGMM and ALAD by over 4.2% and 20.1% improvement in standard F_1 score. On SpamBase and Cardiotocography, ADAF is 4.1% and 3.7% better than DAGMM and OC-SVM methods, respectively. The reasons why ADAF is better than DAGMM may be attributed as: (1) ADAF obtains latent variables based on a reversible flow model. There is no loss of dimensional information in the reversible process, and exact latent variables can be obtained. DAGMM uses an autoencoder to obtain the latent variables, which is an irreversible dimensionality reduction operation and will inevitably lose the information of the original input data; (2) ADAF uses a neural density estimator for density estimation instead of Gaussian mixture model. Deep neural density estimation is superior to Gaussian mixture model, because GMM is a parameter estimation that refers to the process of using sample data to estimate the parameters of the selected distribution, while neural density estimator compute the probability density jointly combining with the generation of latent variables. Additionally, GMM also needs to manually select the number of mixed Gaussian models, which is very tricky in the absence of domain knowledge.

For AnoGAN, it adopts adversarial autoencoder to recover a latent representation for each input data, and uses both reconstruction error and discrimination components as the anomaly criterion, but AnoGAN does not make full use of the low-dimensional representation. Although ALAD can simulate the distribution of data well when the experimental data is large enough, it also ignores the consideration of latent representation. Another potential reason why our method is better than all baselines is that we use an autoregressive flow model to obtain the latent variables and neural density of input data at the same time without dimensionality reduction, avoiding the loss of information.

4.6 Performance w.r.t. Training Set

Second, we investigate the impact of different training data on ADAF and all baselines. We use τ% of the normal dataset as the training set for all methods. We repeat the experiments on Arrhythmia and KDDCUP datasets 20 times and report the average results in Table 3 and Table 4.

As we can see, only when the training data is 30%, our results are slightly lower than DSEBM-e on Arrhythmia. In all other cases, our ADAF significantly outperforms than all baselines in terms of precision, recall and F_1 score on both

Table 3. Performance comparison w.r.t. training ratio on Arrhythmia

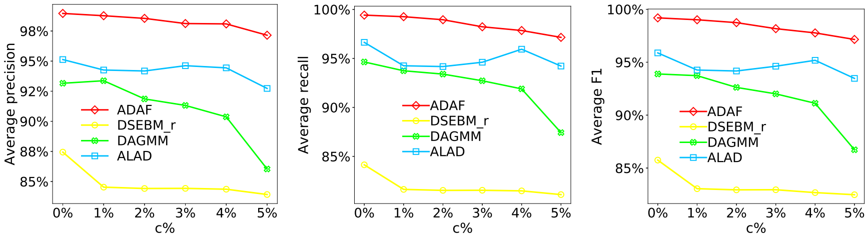
Ratio	ADAF			ALAD			DAGMM			DSEBM-e			DSEBM-r			AnoGAN		
τ%	Precision	Recall	F_1	Precision	Recall	F_1	Precision	Recall	F_1	Precision	Recall	F_1	Precision	Recall	F_1	Precision	Recall	F_1
30%	0.4607	0.4747	0.4676	0.4641	0.5250	0.4926	0.3750	0.4500	0.4091	0.4583	0.5500	0.5000	0.3542	0.4250	0.3864	0.2917	0.3500	0.3182
40%	0.5024	0.5252	0.5135	0.4634	0.5278	0.4935	0.3902	0.4444	0.4156	0.4634	0.5278	0.4935	0.3902	0.4444	0.4156	0.3415	0.3889	0.3636
50%	0.5539	0.5707	0.5621	0.5000	0.5312	0.5152	0.3824	0.4062	0.3939	0.5000	0.5312	0.5152	0.4118	0.4375	0.4242	0.3529	0.3750	0.3636
60%	0.5808	0.5808	0.5808	0.4643	0.4643	0.4643	0.4643	0.4643	0.4643	0.4643	0.4643	0.4643	0.4286	0.4286	0.4286	0.4286	0.4286	0.4286
70%	0.6286	0.6363	0.6315	0.3810	0.4000	0.3902	0.4286	0.4500	0.4390	0.4286	0.4500	0.4390	0.3810	0.4000	0.3902	0.4286	0.4500	0.4390
80%	0.7172	0.7171	0.7171	0.3571	0.4167	0.3846	0.3571	0.4167	0.3846	0.4286	0.5000	0.4615	0.4286	0.5000	0.4615	0.3571	0.4167	0.3846

Table 4. Performance comparison w.r.t. training ratio on KDDCUP

Ratio	ADAF			ALAD			DAGMM			DSEBM-e			DSEBM-r			AnoGAN		
τ%	Precision	Recall	F_1	Precision	Recall	F_1	Precision	Recall	F_1	Precision	Recall	F_1	Precision	Recall	F_1	Precision	Recall	F_1
10%	0.9873	0.9938	0.9906	0.9576	0.9727	0.9651	0.9234	0.9382	0.9308	0.1121	0.1142	0.1131	0.8535	0.8233	0.8381	0.9166	0.8362	0.8667
20%	0.9896	0.9942	0.9919	0.9554	0.9691	0.9622	0.9041	0.9171	0.9106	0.1322	0.1333	0.1332	0.8472	0.8166	0.8316	0.8590	0.8590	0.8590
30%	0.9863	0.9889	0.9876	0.9513	0.9513	0.9513	0.9290	0.9437	0.9363	0.0830	0.0840	0.0830	0.8732	0.8403	0.8564	0.8344	0.8476	0.8409
40%	0.9888	0.9895	0.9892	0.9466	0.9625	0.9545	0.9469	0.9628	0.9548	0.1311	0.1332	0.1321	0.8745	0.8422	0.8576	0.8343	0.8344	0.8344
50%	0.9833	0.9941	0.9887	0.9513	0.9664	0.9588	0.9315	0.9464	0.9389	0.2151	0.2180	0.2170	0.8744	0.8414	0.8575	0.9472	0.8163	0.8630
60%	0.9890	0.9959	0.9925	0.9502	0.9624	0.9563	0.9448	0.9570	0.9509	0.0401	0.0411	0.0410	0.8756	0.8399	0.8573	0.8496	0.8605	0.8550

Arrhythmia and KDDCUP. As the ratio of training data increases, the performance of our model is getting better and better on both datasets, especially on Arrhythmia ADAF achieves a significant improvement. The performance of ALAD and AnoGAN on KDDCUP dataset is relatively stable, and has some fluctuations on Arrhythmia. From Table 4, DSEBM-e that uses energy score as detection criterion is not suitable for KDDCUP. This is because the data distribution of KDDCUP is more complicated than that of the energy model. The experimental results of ALAD, DSEBM-r and AnoGAN are similar because they all use the reconstruction error as the criterion for anomaly detection. Although the results of DAGMM also increases with the increase of training data, our ADAF is far superior to DAGMM, even using less training data.

In summary, this experiment confirms that our ADAF can achieve better results with fewer training samples compared to state-of-the-art baselines.



(a) Precision (b) Recall (c) F_1 score

Fig. 2. Anomaly detection results on contaminated training data on KDDCUP

4.7 Robustness Evaluation

Finally, we evaluate the robustness of our ADAF compared to the baselines on KDDCUP. We only use 10% of the normal data as the training set for our ADAF, and meanwhile we mix c% of samples from the anomalous data into the training set. In term of ALAD, DSEBM and DAGMM, we select 50% of the normal data as the training set, while mixing c% of samples from anomaly data into their training set.

Figure 2 shows the average precision, recall, and F_1 score results of ADAF, DSEBM-e, DAGMM and ALAD with different contaminated training data. When the contamination ratio c increases from 1% to 5%, the average precision, recall, and F_1 score of all methods decrease. However, we also observe that our model is only affected slightly and maintains an extremely robust performance. As c% increases, the performance of DAGMM declines sharply, but the impact on DSEBM-r and ALAD is not very significant. This may be because the GMM model in DAGMM is more sensitive to noise compared to the reconstruction error used in DSEBM-r and ALAD. Nevertheless, our ADAF is still significantly better than all baseline methods.

5 CONCLUSION

In this paper, we propose an Anomaly Detection model based on Autoregressive Flow (ADAF) for detecting anomalies in high-dimensional data. ADAF uses an autoregressive flow to obtain the latent variable, which holds the key information of the original input data. Because of the reversibility of flow model, the latent variables completely inherit the essential information of the original input data. Unlike the traditional two-step methods, ADAF is an absolute end-to-end framework that jointly optimizes the latent error and probability density estimation simultaneously. Finally, both latent error and neural density are used as decision criteria in anomaly detection. Our experimental results on public benchmark datasets show that ADAF is significantly better than state-of-the-art methods by up to 20% improvement on the standard F_1 score.

Acknowledgments. This work is partially supported by the National Natural Science Foundation of China under grant Nos. 61773331, U1706218 and 61572418, and the Natural Science Foundation of Shandong Province under grant No. ZR2018ZB0852.

References

1. An, J., Cho, S.: Variational autoencoder based anomaly detection using reconstruction probability. Special Lecture on IE **2**, 1–18 (2015)
2. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein generative adversarial networks. In: ICML. pp. 214–223 (2017)
3. Baur, C., Wiestler, B., Albarqouni, S., Navab, N.: Deep autoencoding models for unsupervised anomaly segmentation in brain mr images. In: International MICCAI Brainlesion Workshop. pp. 161–169. Springer (2018)
4. Breunig, M.M., Kriegel, H.P., Ng, R.T., Sander, J.: Lof: identifying density-based local outliers. In: ACM sigmod record. vol. 29, pp. 93–104. ACM (2000)
5. Candès, E.J., Li, X., Ma, Y., Wright, J.: Robust principal component analysis? Journal of the ACM (JACM) **58**(3), 11 (2011)
6. Chalapathy, R., Chawla, S.: Deep learning for anomaly detection: A survey. arXiv:1901.03407 (2019)
7. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: A survey. ACM computing surveys (CSUR) **41**(3), 15 (2009)
8. Chen, Y., Zhou, X.S., Huang, T.S.: One-class svm for learning in image retrieval. In: ICIP. pp. 34–37. Citeseer (2001)
9. Dinh, L., Krueger, D., Bengio, Y.: Nice: Non-linear independent components estimation. arXiv preprint arXiv:1410.8516 (2014)
10. Dinh, L., Sohl-Dickstein, J., Bengio, S.: Density estimation using real nvp. arXiv preprint arXiv:1605.08803 (2016)
11. Domingues, R., Filippone, M., Michiardi, P., Zouaoui, J.: A comparative evaluation of outlier detection algorithms: Experiments and analyses. Pattern Recognition **74**, 406–421 (2018)
12. Erfani, S.M., Rajasegarar, S., Karunasekera, S., Leckie, C.: High-dimensional and large-scale anomaly detection using a linear one-class svm with deep learning. Pattern Recognition **58**, 121–134 (2016)

13. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: NeurIPS. pp. 2672–2680 (2014)
14. Günter, S., Schraudolph, N.N., Vishwanathan, S.: Fast iterative kernel principal component analysis. Journal of Machine Learning Research **8**(Aug), 1893–1918 (2007)
15. Hu, W., Gao, J., Li, B., Wu, O., Du, J., Maybank, S.J.: Anomaly detection using local kernel density estimation and context-based regression. IEEE Transactions on Knowledge and Data Engineering (2018)
16. Keller, F., Muller, E., Bohm, K.: Hics: high contrast subspaces for density-based outlier ranking. In: ICDE. pp. 1037–1048. IEEE (2012)
17. Kingma, D.P., Dhariwal, P.: Glow: Generative flow with invertible 1x1 convolutions. In: NeurIPS. pp. 10215–10224 (2018)
18. Knorr, E.M., Ng, R.T., Tucakov, V.: Distance-based outliers: algorithms and applications. The VLDB Journal **8**(3-4), 237–253 (2000)
19. Liu, F.T., Ting, K.M., Zhou, Z.H.: Isolation forest. In: ICDM. pp. 413–422. IEEE (2008)
20. Oord, A.v.d., Li, Y., Babuschkin, I., Simonyan, K., Vinyals, O., Kavukcuoglu, K., Driessche, G.v.d., Lockhart, E., Cobo, L.C., Stimberg, F., et al.: Parallel wavenet: Fast high-fidelity speech synthesis. arXiv preprint arXiv:1711.10433 (2017)
21. Ravanbakhsh, M., Nabi, M., Mousavi, H., Sangineto, E., Sebe, N.: Plug-and-play cnn for crowd motion analysis: An application in abnormal event detection. In: WACV. pp. 1689–1698. IEEE (2018)
22. Ruff, L., Vandermeulen, R., Goernitz, N., Deecke, L., Siddiqui, S.A., Binder, A., Müller, E., Kloft, M.: Deep one-class classification. In: ICML. pp. 4393–4402 (2018)
23. Schlegl, T., Seeböck, P., Waldstein, S.M., Langs, G., Schmidt-Erfurth, U.: f-anogan: Fast unsupervised anomaly detection with generative adversarial networks. Medical image analysis **54**, 30–44 (2019)
24. Schlegl, T., Seeböck, P., Waldstein, S.M., Schmidt-Erfurth, U., Langs, G.: Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In: IPMI. pp. 146–157. Springer (2017)
25. Schubert, E., Zimek, A., Kriegel, H.P.: Generalized outlier detection with flexible kernel density estimates. In: SDM. pp. 542–550. SIAM (2014)
26. Tan, S.C., Ting, K.M., Liu, T.F.: Fast anomaly detection for streaming data. In: IJCAI (2011)
27. Yan, Y., Cao, L., Rundensteiner, E.A.: Scalable top-n local outlier detection. In: KDD. pp. 1235–1244. ACM (2017)
28. Zenati, H., Romain, M., Foo, C.S., Lecouat, B., Chandrasekhar, V.: Adversarially learned anomaly detection. In: ICDM. pp. 727–736. IEEE (2018)
29. Zhai, S., Cheng, Y., Lu, W., Zhang, Z.: Deep structured energy based models for anomaly detection. ICML **48** (2016)
30. Zhou, C., Paffenroth, R.C.: Anomaly detection with robust deep autoencoders. In: KDD. pp. 665–674. ACM (2017)
31. Zong, B., Song, Q., Min, M.R., Cheng, W., Lumezanu, C., Cho, D., Chen, H.: Deep autoencoding gaussian mixture model for unsupervised anomaly detection. ICLR (2018)