Deep Anomaly Detection and Search via Reinforcement Learning* (Student Abstract)

Chao Chen[1], Dawei Wang[2], Feng Mao[2], Zongzhang Zhang[1], Yang Yu[1]

[1] National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China
[2] Alibaba Group, Hangzhou 310052, China
chenc@lamda.nju.edu.cn, {david.wdw, maofeng.mf}@alibaba-inc.com, {zzzhang, yuy}@nju.edu.cn

Abstract

Semi-supervised anomaly detection is a data mining task which aims at learning features from partially-labeled datasets. We propose Deep Anomaly Detection and Search (DADS) with reinforcement learning. During the training process, the agent searches for possible anomalies in unlabeled dataset to enhance performance. Empirically, we compare DADS with several methods in the settings of leveraging known anomalies to detect both other known and unknown anomalies. Results show that DADS achieves good performance.

Introduction

Anomaly Detection (AD) (Chandola, Banerjee, and Kumar 2009) is a classical data mining task which aims at detecting data instances that significantly deviate from the majority. In this work, we focus on semi-supervised AD, where labels from only part of the training dataset are available.

Semi-supervised AD is a specific type of AD where the training dataset is composed of a small labeled dataset and a large unlabeled dataset. Existing methods generally face the following two challenges. Firstly, some methods rely heavily on some prior data distribution assumptions, such as the cluster assumption (Chapelle, Scholkopf, and Zien 2009). Therefore, their results are closely related to how well these assumptions are met. Secondly, unlabeled datasets naturally contain anomalies, also known as contamination. However, some methods are not robust to contamination.

Deep Reinforcement Learning (RL) combines deep learning with RL. Deep RL algorithms can be divided into three categories: value-based, policy-based, and actor-critic (e.g., SAC (Haarnoja et al. 2018)). There is little research on the application of RL to AD, but we believe that once the strengths of RL in balancing exploration and exploitation are fully exploited, it will become a powerful tool for AD.

The main contributions of this work can be summarized as improvements in scenarios where testing set contains unknown anomaly classes. With the help of RL, DADS integrates the search of unknown anomalies and reducing the contamination of unlabeled dataset into one model.

Our Method

In this section, we introduce our method called DADS. Consider a semi-supervised AD scenario, where an anomaly dataset $\mathcal{D}^a$ and an unlabeled dataset $\mathcal{D}^u$ is available. Our aim is to find an anomaly scoring function $\phi(\cdot)$, such that $\phi(s_i) > \phi(s_j)$, where s_i is abnormal and s_j is normal. In particular, we consider multi-dimensional dataset.

Figure 1 is an illustration of DADS. Before introducing the SAC-based agent and the anomaly search environment, we define the state space and action space of RL. State space is the whole training dataset $\mathcal{D} = \{\mathcal{D}^a, \mathcal{D}^u\}$, with each $s_t \in \mathcal{D}$ sampled at time step t be a state. Action space is a continuous space within range $[0, 1]$, with value corresponding to anomaly score of input data.

SAC-Based Agent

Agent takes the current data as input and returns the corresponding anomaly score to the environment. In the design of the agent, we use the RL algorithm SAC, which adds an extra entropy term to the original target of RL. With the help of entropy regularization, agent is encouraged to explore more unseen states, thus improving the search efficiency.

After training, we get $\pi^*(\theta)$. For every single data s, $\pi^*(s;\theta)$ is used as anomaly score.

Anomaly Search Environment

The environment of DADS is divided into three parts. With these key components, DADS can not only leverage limited anomalies, but also achieve robustness to contamination of unlabeled dataset.

Hierarchical Datasets Tailored for Anomaly Search We divide the whole training dataset into three interconnected datasets: anomaly dataset $\mathcal{A}$, temporary dataset $\mathcal{T}$, and unlabeled dataset $\mathcal{U}$, which is composed of anomalies, possible anomalies and unlabeled data respectively. As is illustrated in Figure 1, three blue boxes correspond to three datasets. For each sampled data s_t, if the action is larger than TH_{score}, it will be put into a dataset according to the red arrow, else the target dataset is determined by the green arrow.

During training, $\texttt{conf}(s_t)$ was set to record how many consecutive times a data is judged as an anomaly. Any unlabeled data judged as abnormal consecutively for TH_{conf} times will be placed into anomaly dataset, which is actually a process of

*Corresponding author: Zongzhang Zhang. This work is supported by the NSFC (No. 62276126) and Alibaba Group through Alibaba Research Fellowship Program.

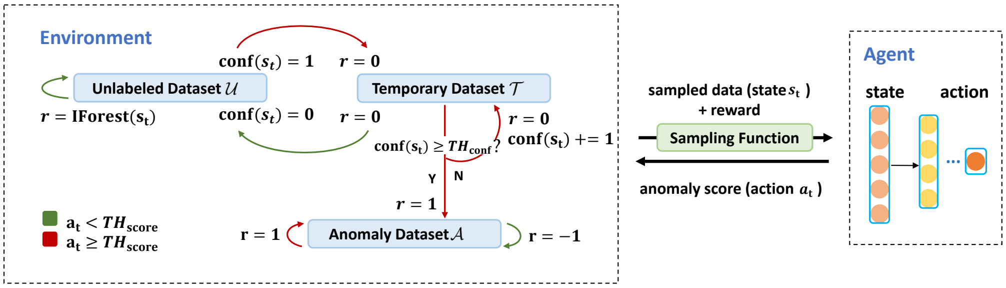


Figure 1: An illustration of our method DADS. See text for details.

searching possible anomalies and reducing the contamination of unlabeled dataset.

Ensemble-based Sampling Function The sampling strategy can be summarized as two stages. In the first stage, the environment chooses a certain dataset from three inner datasets $\{\mathcal{A}, \mathcal{T}, \mathcal{U}\}$ according to a predefined probability distribution. After that, the environment samples a batch of data from the selected dataset and chooses the one with the highest unsupervised anomaly score, which is calculated by averaging several unsupervised AD methods including isolation forest, OCSVM, and HBOS.

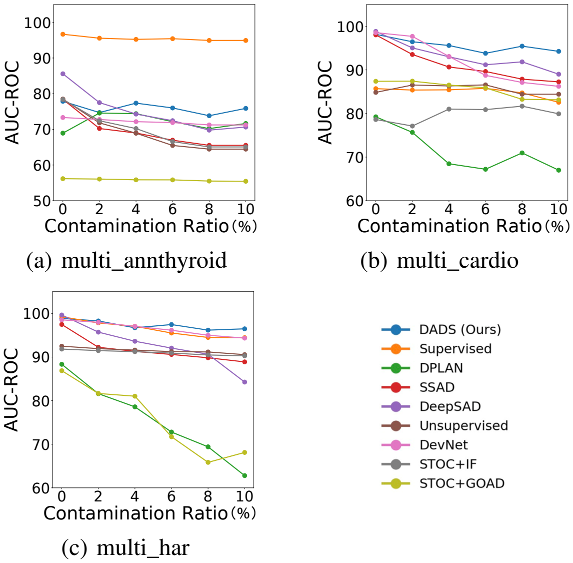
Though the transition function cannot be written directly, we still claim that it is suitable for applying RL. Every action will have a direct impact on the dataset to which the current data belongs. Further, it will affect the composition of $\{\mathcal{A}, \mathcal{T}, \mathcal{U}\}$, and finally influence the next sampled data.

Reward Function for Anomaly Detection The reward function of the environment is designed based on both supervised and unsupervised rewards.

For s_t coming from $\mathcal{A}$, the agent is asked to make a correct judgment, which is referred to as supervised reward. If current data s_t comes from $\mathcal{T}$, the agent will receive a reward when s_t is added into $\mathcal{A}$. For s_t sampled from $\mathcal{U}$, to enable the agent to learn data distribution with the help of unsupervised methods, the environment will give an unsupervised reward using Isolation Forest, which is written as $\texttt{IForest}(s_t)$.

Experiments

To test the ability of DADS in detecting unknown anomalies, we select 3 datasets with multiple anomaly classes. For each dataset we choose one anomaly class as known and leave others as unknown, and the number of known anomalies is set to be 10% of total anomalies. To further verify whether DADS is robust to contamination of unlabeled dataset, we adjust the percentage of anomalies in unlabeled data(contamination ratio) from 0% to 10%. 7 different AD methods are used for comparison, including one supervised method XGBoost, one unsupervised method Isolation Forest and five semi-supervised methods. We report average Area Under Receiver Operating Characteristic Curve (AUC-ROC) over 10 random seeds, which measures the area under ROC curve.



(a) multi_annthyroid

(b) multi_cardio

(c) multi_har

Figure 2: AUC-ROC of DADS and baselines.

Detailed results are shown in Figure 2. Although inferior to the supervised method in annthyroid (we hypothesis this maybe due to the close distribution of normal and abnormal data), DADS still achieves the best overall performance. In addition, all methods except DADS show varying degrees of decline as the contamination ratio increases, which proves the robustness of DADS to contamination.

Conclusion

This paper presents an RL-based semi-supervised tabular AD method DADS. With the help of hierarchical search mechanism and ensemble-based sampling function, DADS performs well in our experiments.

References

Chandola, V.; Banerjee, A.; and Kumar, V. 2009. Anomaly detection: A survey. *ACM Computing Surveys*, 41(3): 1–58.

Chapelle, O.; Scholkopf, B.; and Zien, A. 2009. Semi-supervised learning. *IEEE Transactions on Neural Networks*, 20(3): 542–542.

Haarnoja, T.; Zhou, A.; Hartikainen, K.; Tucker, G.; Ha, S.; Tan, J.; Kumar, V.; Zhu, H.; Gupta, A.; Abbeel, P.; et al. 2018. Soft actor-critic algorithms and applications. *arXiv:1812.05905*.