A Lightweight Meta-Feature Extraction Strategy for Deep Reinforcement Anomaly Detection

Simone Amirato[1,*,†], Fabrizio Angiulli[1,†], Fabio Fassetti[1,†] and Luca Ferragina[1,†]

[1]DIMES Dept., University of Calabria, 87036 Rende (CS), Italy.

Abstract

Reinforcement Learning (RL) describes a family of algorithms teaching an agent to determine a policy for interacting with its surrounding environment. Recently, the RL paradigm has been successfully applied to the anomaly detection problem by learning a meta-policy on a set of already labelled datasets. The meta-policy is subsequently actively applied to a flow of incoming unseen observations, representing the test environment. The interesting point of this approach is that one can apply the meta-policy without further tuning to a small number of meta-features that can be directly extracted from any new dataset.

For this kind of approach, a central question is the selection of a good set of meta-features. To date, two strategies have been explored, the first relying on meta-features defined in terms of the distances with the points that the expert has already labelled, and the second exploiting the direct and reverse nearest-neighbor rankings of these labelled points.

However, both strategies present the disadvantage that the worst-case cost of the meta-feature extraction procedure is, for each incoming point, linear in the dataset size (and, hence, quadratic as a whole). To alleviate the computational cost associated with the crucial meta-feature extraction step, in this work, we investigate a set of hybrid features that takes into account both distances and rankings, but has, for each incoming point, constant cost. Experiments highlight that the approach preserves accuracy while offering advantages in terms of resource consumption.

Keywords

Anomaly Detection, Reinforcement Learning, Active Learning, Meta-Learning, Green AI

1. Introduction

The Anomaly Detection task is one of the main discovery problems and its goal is to isolate objects in a dataset that are suspected of being generated by a different process with respect to the rest of the data.

The presence of anomalies is due to many reasons, like mechanical faults, fraudulent behavior, human errors, instrument errors, or simply through natural deviations in populations. We can distinguish three approaches to anomaly detection, namely supervised, semi-supervised, and unsupervised [1]. Supervised methods create a classifier after being trained on data labeled as normal and anomalous. In this setting usually, the classes are unbalanced since the anomalies are rare. Semi-supervised methods are trained with examples of just the normal class, thus they are also called one-class classifiers. Unsupervised methods take in input a dataset and try to find anomalies in it by assigning a score to each object. Several statistical, data mining and machine learning approaches have been proposed to detect outliers, namely, statistical-based [2], distance-based [3, 4, 5, 6], density-based [7, 8], reverse nearest neighbor-based [9, 10, 11], isolation-based [12], angle-based [13], SVM-based [14, 15], deep learning-based [16], and many others.

An alternative approach to classical outlier detection methods aimed at reducing false-positive rates is based on Active Anomaly Detection (AAD), which involves humans in the loop. There are several traditional anomaly detection scenarios where an analyst is involved in checking the top instances from

1st Workshop on Green-Aware Artificial Intelligence, 23rd International Conference of the Italian Association for Artificial Intelligence (AIxIA 2024), November 25–28, 2024, Bolzano, Italy

*Corresponding author.

†These authors contributed equally.

simone.amirato@dimes.unical.it (S. Amirato); f.angiulli@dimes.unical.it (F. Angiulli); f.fassetti@dimes.unical.it (F. Fassetti); luca.ferragina@unical.it (L. Ferragina)

0009-0009-8945-0182 (S. Amirato); 0000-0002-9860-7569 (F. Angiulli); 0000-0002-8416-906X (F. Fassetti); 0000-0003-3184-4639 (L. Ferragina)

Feature Type	Description	Cost
Detector	**D**: Score related to the unsupervised detector	
Binary	**B1**: indicator asserting the existence of an anomalous instance in the k-nearest neighbors	$\mathcal{O}(dn^2)$
Anomaly	**A1**: minimum distance between x_i and the set of instances currently labelled as anomalous	$\mathcal{O}(dBn)$
	A2: mean distance between x_i and the set of instances currently labelled as anomalous	$\mathcal{O}(dBn)$
	A3: the position occupied by the nearest known anomaly in the nearest neighbors list of x_i, that is $\min_{j \in H_A} R_{ij}$, where H_A is the set of the known anomaly indices	$\mathcal{O}(dn^2)$
	A4: the position $R_{\mu i}$ occupied by x_i in the nearest neighbors list of x_μ, where $\mu = \arg\min_{j \in H_A} R_{ij}$	$\mathcal{O}(dBn)$
Normality	**N1**: minimum distance between x_i and the set of instances currently labelled as normal	$\mathcal{O}(dBn)$
	N2: mean distance between x_i and the set of instances currently labelled as normal	$\mathcal{O}(dBn)$
	N3: the position occupied by the nearest known normal item in the nearest neighbors list of x_i, that is $\min_{j \in H_N} R_{ij}$, where H_N is the set of the known normal items indices	$\mathcal{O}(dn^2)$
	N4: the position $R_{\nu i}$ occupied by x_i in the nearest neighbors list of x_ν, where $\nu = \arg\min_{j \in H_N} R_{ij}$	$\mathcal{O}(dBn)$

Table 1
Meta-features adopted in *Meta–AAD*-based approaches.

a ranked list of anomalies to identify as many true anomalies as possible. In AAD this human feedback can be leveraged to help the system to identify more anomalies. In this work, we consider the AAD scenario in which the anomaly detector queries the analyst by selecting instances one at a time. In this scenario, the system can adjust the decision functions by leveraging the expert's knowledge gained from their responses to the queries. This human feedback can help the anomaly detector to promote the instances of interest and discourage the instances out of interest, showing the analyst more true anomalies in subsequent iterations.

A recent interesting improvement of AAD is the Active Anomaly Detection with Meta-Policy (*Meta–AAD*), which learns a meta-policy with the aim of optimizing the number of discovered anomalies keeping the same *budget B*, that is to say, the number of instances presented to the human for feedback.

Given a dataset $X = \{x_1, \ldots, x_n\} \subseteq \mathbb{R}^d$, at each iteration the policy selects an example x_i to submit to the expert that will assert whether x_i is actually an anomaly or not. The state of the policy is recorded in a vector $\hat{\mathbf{y}} \in \mathbf{R}^n$ such that, for each $j \in \{1, \ldots, n\}$, we have that the element y_j is equal to -1, if the item x_j has been submitted to the expert and reported as an anomaly, 1, if x_j has been submitted to the expert and reported as normal, 0, if x_j has not yet been submitted to the expert.

At the beginning of the policy, we have $y_j = 0$, for each $j \in \{1, \ldots, n\}$, and then at each iteration, the state of the item selected for the query is updated according to the expert feedback.

In AAD the instance presented to the human is that scoring the highest value of anomaly score and, due to feedback, the anomalous scores are adjusted to promote the anomalous instances to the top, thus the main goal is to make the top instance more likely to be anomalous so maximizing the immediate performance.

Thus, in [17], first the meta-policy is trained and any Deep Reinforcement Learning (DRL) algorithm can be used to this aim. In particular, authors adopt Proximal Policy Optimization (PPO) [18], a family of methods for reinforcement learning, in which a deep neural network is trained iteratively by sampling data through interaction with the environment, and by using these data for the optimization of a state function using stochastic gradient ascent.

Then, for all the instances, the meta-features are extracted and the probability of the meta-policy is computed. The instance with the highest probability is presented to the human.

A fundamental part of this process is the meta-features extraction. Formally, the aim is to define a function $g : X \times \mathbf{y} \to \mathbf{R}^{n \times l}$, where l is the number of meta-features, such that its image is as much as possible independent from the dataset. The framework allows flexible choices, in Table 1 are listed the main features adopted in literature. Among all these features an important role is played by the unsupervised anomaly detector **D**. Indeed, it is the only feature in the table that is computed before the training process and that is not updated after the expert's feedback. Its main goal is to provide a bootstrap phase to the active process in the initial phase when the other features are not stable yet. As for the other meta-features they can be initialized at any arbitrary value for all the points, so that at the very first step of the active process the algorithm cannot discriminate on their basis.

In [17], that is the first approach considering the *Meta–AAD* paradigm, authors propose the following set of $l = 6$ meta-features: **D** (in particular they adopt the Isolation Forest algorithm [12]), **B1**, **A1**, **A2**, **N1**, **N2**. It is worth noticing that all these features are distance-based, because of this, in [19], it is argued that this fact may cause some issues in the query selection phase due to the specific distance distribution of the dataset. In order to relieve this dependence, it is introduced an innovative approach, called *Meta–AAD–Rank*, for the choice of the meta-features based on nearest neighbor rankings rather than on distances. Specifically, this set of meta-features includes **D**, **B1**, **A3**, **A4**, **N3**, **N4** and also proposes the adoption of the CFOF algorithm [20, 21, 11] as anomaly detector since it is reverse neighbor based. Specifically, the CFOF algorithm is involved in the meta-feature **D**.

2. *Meta–AAD–Light*

Both the *Meta–AAD* and *Meta–AAD–Rank* require substantial memory, mainly due to the distances matrix calculations involved in certain meta-features.

Specifically, to compute the feature **B1**, **A3** and **N3** for a single item x_i, the values of the distances of x_i from all the other points in X are needed, thus the cost is $\mathcal{O}(dn)$ and $\mathcal{O}(dn^2)$ for the whole dataset.

Because of this, in order to reduce the computational cost, we propose a further set of features, called *Meta–AAD–Light*, that is obtained as a combination of the previous two. Specifically, the meta-features of *Meta–AAD–Light* are represented by **D**, **A1**, **A4**, **N1**, **N4**.

In addition, we introduce a novel binary feature, referred to as **B2**, which has a linear computational cost and is defined as follows:

> **B2**: indicator asserting whether x_i belongs to the k-nearest neighbors of any known anomalous instance

The computation of the *Meta–AAD–Light* meta-features does not require computing the distances (and the associate rankings) among all the data points, but only the information about the distances between the points already classified as normal or anomalous and all the other points. In this way, the memory occupation goes from $\mathcal{O}(dn^2)$ for *Meta–AAD* and *Meta–AAD–Rank* to $\mathcal{O}(dBn)$ for *Meta–AAD–Light* where B is the number of queries and it is usually very smaller than the size n of the dataset.

3. Experimental results

In our experiments we consider the tabular datasets *lympho*, *pima*, *satellite*, *satimage-2*, *shuttle*, *speech*, *wine*, from the ODDS repository [22]. Specifically, we adopt a *leave-one-out* strategy, meaning that for the evaluation phase on each dataset, we consider a meta-policy trained on all the others.

We fix a budget of $B = 100$ and we consider the following evaluation metrics, used also in [17, 19].

- The total number of anomalies a_B submitted to the expert in the B queries.
- The **Precision** (Prec), defined as
$$\frac{a_B}{B'}, \tag{1}$$
where $B' = \min(B, n_a)$, and n_a is the total number of anomalies in the dataset.

Dataset ($n \times d$)	*Meta–AAD*				*Meta–AAD–Rank*			
	$\mathbf{a_B}$	**Prec**	**N-AUC**	$\mathbf{CO_2}$	$\mathbf{a_B}$	**Prec**	**N-AUC**	$\mathbf{CO_2}$
lympho (148 × 18)	0.0000	0.0000	0.0000	- 1%	0.0000	0.0000	0.0000	- 2%
pima (768 × 8)	-0.1143	-0.1143	-0.2141	- 9%	-0.1014	-0.1014	-0.1913	-29%
satellite (6435 × 36)	0.0000	0.0000	0.0000	-74%	0.0000	0.0000	0.0000	-93%
satimage-2 (5803 × 36)	-0.0299	-0.0299	-0.0584	-71%	-0.0580	-0.0580	-0.1118	-92%
shuttle (49097 × 9)	0.0000	0.0000	0.0000	-87%	0.0000	0.0000	0.0000	-97%
speech (3686 × 400)	3.2500	3.2500	14.3000	-41%	0.8889	0.8889	2.4000	-75%
wine (129 × 13)	0.0000	0.0000	0.0000	0%	0.0000	0.0000	0.0000	- 1%

Table 2
Comparison between *Meta–AAD–Light* and *Meta–AAD* (left side of the table) and *Meta–AAD–Rank* (right side of the table) in terms of the considered evaluation metrics as well as the CO_2 emissions.

- The **Normalized AUC** (N-AUC), that is the area under the curve of a_s with varying s normalized by the number of the total number of anomalies in the dataset, in formula

$$\frac{2\sum_{s=1}^{B} a_s}{B'(B'+1)}. \tag{2}$$

Moreover, in this work, we add a new metric to take into account the environmental impact of methods. In particular, we compute the Kg of CO_2 emitted by exploiting the framework CodeCarbon [23].

In Table 2 we show the results of the experiments. The first column reports the employed datasets together with their number of samples and dimensions. Columns from 2 to 5 are associated with the comparison between *Meta–AAD–Light* and *Meta–AAD* and columns from 6 to 9 with the comparison between *Meta–AAD–Light* and *Meta–AAD–Rank*. Specifically, for each competitor C and for each of the four above-mentioned metrics M (namely, the number of anomalies, the precision, the normalized-AUC and the emitted CO_2), we compute the relative variation of *Meta–AAD–Light* with respect to C as $\frac{M(Meta\text{–}AAD\text{–}Light)-M(C)}{M(C)}$, where $M(Meta\text{–}AAD\text{–}Light)$ is the value of the metrics accomplished by *Meta–AAD–Light* and $M(C)$ the one accomplished by the competitor C.

We can note that, for the majority of the datasets, *Meta–AAD–Light* obtains the same detection performances of *Meta–AAD* and *Meta–AAD–Rank*, for two datasets, namely *pima* and *satimage-2*, it is possible to observe a slight degradation in the results, and for the dataset *speech* we even outperform the other two methods. As for the emissions of CO_2, as predicted by the theoretical discussion, *Meta–AAD–Light* is much less energy-consumptive. As expected, this is particularly emphasized for the three datasets with the largest cardinality, where, in two cases (*satellite* and *shuttle*), we obtain the same performance in terms of accuracy while almost halving the CO_2 emissions, and in one case (*satimage-2*) we pay a small loss in accuracy to obtain a great benefit in terms of emission.

4. Conclusion

In this paper, we introduced *Meta–AAD–Light*, a novel meta-features extraction strategy for meta-active anomaly detection. We prove that, differently from the already existing sets of meta-features, the computation of *Meta–AAD–Light* requires constant time for a single item and, then, linear for the whole dataset. Experiments confirm that while maintaining good performance in terms of accuracy, the proposed method notably reduces the CO_2 emissions, thus decreasing the environmental impact of the whole approach.

Acknowledgments

We acknowledge the support of the PNRR project FAIR - Future AI Research (PE00000013), Spoke 9 - Green-aware AI, under the NRRP MUR program funded by the NextGenerationEU.

References

[1] L. Ruff, J. R. Kauffmann, R. A. Vandermeulen, G. Montavon, W. Samek, M. Kloft, T. G. Dietterich, K. Müller, A unifying review of deep and shallow anomaly detection, Proc. IEEE 109 (2021) 756–795.

[2] L. Davies, U. Gather, The identification of multiple outliers, Journal of the American Statistical Association 88 (1993) 782–792.

[3] E. Knorr, R. Ng, V. Tucakov, Distance-based outlier: algorithms and applications, VLDB Journal 8 (2000) 237–253.

[4] F. Angiulli, C. Pizzuti, Outlier mining in large high-dimensional data sets, IEEE Trans. Knowl. Data Eng. 2 (2005) 203–215.

[5] F. Angiulli, S. Basta, C. Pizzuti, Distance-based detection and prediction of outliers, IEEE Trans. on Knowledge and Data Engineering 2 (2006) 145–160.

[6] F. Angiulli, F. Fassetti, DOLPHIN: an efficient algorithm for mining distance-based outliers in very large datasets, ACM Trans. Knowl. Disc. Data (TKDD) 3(1) (2009) Article 4.

[7] M. M. Breunig, H. Kriegel, R. Ng, J. Sander, Lof: Identifying density-based local outliers, in: Proc. Int. Conf. on Managment of Data (SIGMOD), 2000.

[8] W. Jin, A. Tung, J. Han, Mining top-n local outliers in large databases, in: Proc. ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD), 2001.

[9] V. Hautamäki, I. Kärkkäinen, P. Fränti, Outlier detection using k-nearest neighbour graph, in: International Conference on Pattern Recognition (ICPR), Cambridge, UK, August 23-26, 2004, pp. 430–433.

[10] M. Radovanović, A. Nanopoulos, M. Ivanović, Reverse nearest neighbors in unsupervised distance-based outlier detection, IEEE Transactions on Knowledge and Data Engineering 27 (2015) 1369–1382.

[11] F. Angiulli, CFOF: A concentration free measure for anomaly detection, ACM Transactions on Knowledge Discovery from Data (TKDD) 14 (2020) 4:1–4:53.

[12] F. Liu, K. Ting, Z.-H. Zhou, Isolation-based anomaly detection, TKDD 6 (2012).

[13] H.-P. Kriegel, M. Schubert, A. Zimek, Angle-based outlier detection in high-dimensional data, in: Proc. Int. Conf. on Knowledge Discovery and Data Mining (KDD), 2008, pp. 444–452.

[14] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, R. C. Williamson, Estimating the support of a high-dimensional distribution, Neural Computation (2001).

[15] D. M. J. Tax, R. P. W. Duin, Support vector data description, Mach. Learn. (2004).

[16] R. Chalapathy, S. Chawla, Deep learning for anomaly detection: A survey, 2019. arXiv:1901.03407.

[17] D. Zha, K. Lai, M. Wan, X. Hu, Meta-aad: Active anomaly detection with deep reinforcement learning, CoRR abs/2009.07415 (2020). arXiv:2009.07415.

[18] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, Proximal policy optimization algorithms, CoRR abs/1707.06347 (2017). arXiv:1707.06347.

[19] F. Angiulli, F. Fassetti, L. Ferragina, P. Papaleo, Meta-feature extraction strategies for active anomaly detection, in: Intelligent Data Engineering and Automated Learning–IDEAL 2021: 22nd International Conference, Springer, 2021, pp. 406–414.

[20] F. Angiulli, Concentration free outlier detection, in: European Conference on Machine Learning and Knowledge Discovery in Databases, Skopje, Macedonia, 2017.

[21] F. Angiulli, On the behavior of intrinsically high-dimensional spaces: Distances, direct and reverse nearest neighbors, and hubness, Journal of Machine Learning Research 18 (2018) 170:1–170:60.

[22] S. Rayana, Odds library, 2016. URL: http://odds.cs.stonybrook.edu.

[23] B. Courty, V. Schmidt, S. Luccioni, Goyal-Kamal, MarionCoutarel, B. Feld, J. Lecourt, LiamConnell, A. Saboni, Inimaz, supatomic, M. Léval, L. Blanche, A. Cruveiller, ouminasara, F. Zhao, A. Joshi, A. Bogroff, H. de Lavoreille, N. Laskaris, E. Abati, D. Blank, Z. Wang, A. Catovic, M. Alencon, M. Stęchły, C. Bauer, L. O. N. de Araújo, JPW, MinervaBooks, mlco2/codecarbon: v2.4.1, 2024. URL: https://doi.org/10.5281/zenodo.11171501. doi:10.5281/zenodo.11171501.