TracInAD: Measuring Influence for Anomaly Detection

Accepted for Oral Presentation at IJCNN 2022

Hugo Thimonier[1], Fabrice Popineau[1], Arpad Rimmel[1], Bich-Liên Doan[1], Fabrice Daniel[2]

[1]Université Paris-Saclay, CNRS, CentraleSupélec, Laboratoire Interdisciplinaire des Sciences du Numérique
[2]LUSIS, AI Department

This work was supported by the **Chaire Lusis** led by CentraleSupelec and the Lusis company.

Anomaly Detection

Anomaly detection (AD) as a research direction has caught more and more attention in the recent years:

- Well-suited for applications where classes are **imbalanced** (*e.g.* fraud detection, intrusion detection . . . etc.).
- Effective for tasks where **no labels are available**.

Contribution

In the present work we propose a **novel Anomaly Detection** method based on **influence measures** which can serve to **augment any deep anomaly detection** .

What is Anomaly Detection ?

Standard Supervised Approach to Classification

For the vanilla binary classification case one usually considers the following set-up:

- A training set $D_n^{train} = \{(x_i, y_i), x_i \in \mathcal{X}, y_i \in \{0,1\}\}_{i=1}^n$ composed of **samples belonging to both classes,** $y_i = 0$ **and** $y_i = 1$**.**
- The goal is to directly learn a classifier using the training set

$$f : \mathcal{X} \rightarrow \{0,1\}$$

Anomaly Detection

Standard approaches to AD:

- Training set D_n^{train} solely composed of *normal* samples.

$$D_n^{train} = \{(x_i, y_i), y_i = 0\}_{i=1}^n$$

  where $x_i \in \mathcal{X} \subseteq \mathbb{R}^d, y_i \in \mathcal{Y} = \{0, 1\}$.
- Most AD methods aim at **characterizing the distribution of the *normal* samples** $(y = 0)$, $\mathbb{P}_{y=0}$.
- Samples that belong to a **low probability region** of the normal distribution are then flagged as **anomalous** $(y = 1)$.

Different types of AD

Three approaches to AD:

- **One-Class Classification** e.g. OCSVM [Schölkopf et al. (1999)], SVDD [Tax and Duin (2004)], Deep-SVDD [Ruff et al. (2018)].
- **Reconstruction-Based Methods** e.g. VAE, Autoencoder, RaPP [Kim et al. (2020)] . . . etc.
- **Self-Supervised Methods** e.g. GOAD [Bergman and Hoshen (2020)], NeutralAD [Qiu et al. (2021)], Internal Contrastive Learning methods [Shenkar and Wolf (2022)].

Our approach can serve to augment any deep methods from **all three categories.**

Set-Up

Consider the following set-up:

- Consider f_θ a **deep model** parametrized by $\theta \in \Theta \subseteq \mathbb{R}^p$.
- **Parameters** θ are obtained by minimizing a loss function $\ell : \Theta \times \mathcal{X} \to \mathbb{R}$ over the training set.

$$\theta^* = \arg\min_{\theta \in \Theta} \sum_{x \in \mathcal{D}_{train}} \ell(\theta, x).$$

Influence (1)

Definition (Influence)

The influence of a sample x on a test sample x' is the **difference in the loss** for the sample x' **incurred by having included** x **in the training set**. Formally, the influence function of a sample x on the test sample x' is:

$$IF(x, x') = \ell(\theta, x') - \ell(\theta_{-x}, x') \tag{1}$$

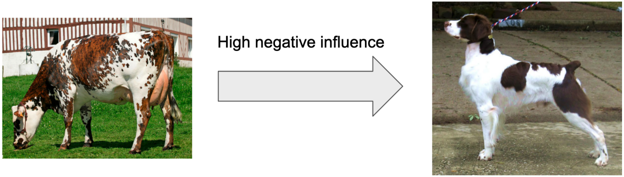
where $\theta_{-x} = \arg\min_{\theta \in \Theta} \sum_{z \in \mathcal{D}_{train} \setminus \{x\}} \ell(\theta, z)$.

Influence (2)

- Influence was first proposed for **explicability purposes**.
- It allows to identify the samples which contributed to reducing the loss of a sample and those that contributed to increasing its loss.
- It can help understand **why some samples were misclassified**, especially for image datasets.

Influence (2)

- Influence was first proposed for **explicability purposes**.
- It allows to identify the samples which contributed to reducing the loss of a sample and those that contributed to increasing its loss.
- It can help understand **why some samples were misclassified**, especially for image datasets.



Dog classified as cow

TracIn, Pruthi et al. (2020)

Based on a first-order approximation, Pruthi et al. (2020) propose `TracIn`, a novel estimation of the Influence function given in (1).

- **Parameters** θ are obtained by minimizing a loss function through an **iterative optimization process**.
- **Optimizer**: SGD with step size η_t at iteration t.
- θ_t denotes the obtained parameters after iteration t.
- B_t a minibatch of size b at iteration t.

TracIn, Pruthi et al. (2020)

TracIn

The influence of sample x on sample x' is estimated by

$$\texttt{TracIn}(x, x') = \frac{1}{b} \sum_{t:x \in B_t} \eta_t \nabla \ell(\theta_t, x) \cdot \nabla \ell(\theta_t, x') \quad (2)$$

where $\nabla \ell(\theta_t, x')$ denotes the gradient of the loss function evaluated for the sample x' w.r.t. the parameter θ_t.

TracInAD (1)

In an **unsupervised set-up** involving β-Variational Autoencoders, Kong and Chaudhuri (2021) show that the **self-influence behaviour differs between normal samples and anomalies.**

TracInAD (1)

In an **unsupervised set-up** involving β-Variational Autoencoders, Kong and Chaudhuri (2021) show that the **self-influence behaviour differs between normal samples and anomalies.**

Hypothesis

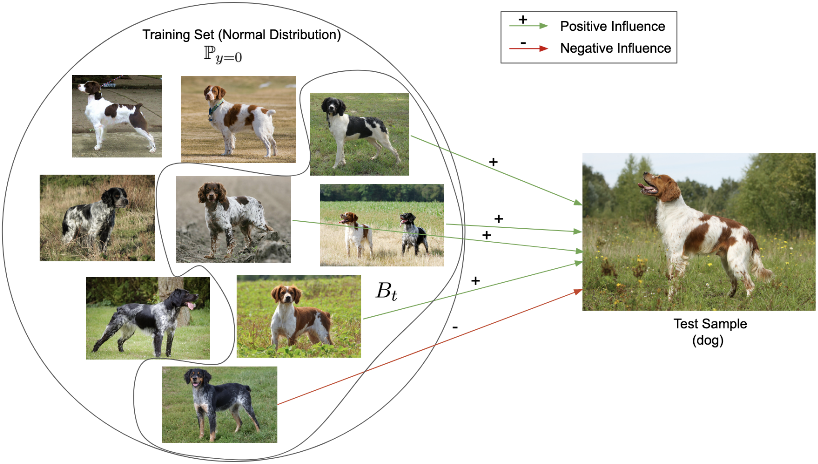
Not only do self-influence behaviours differ between normal samples and anomalies, but **the influence of normal points on anomalies should significantly differ from the influence of normal points on normal points.**

Intuition

On average, **normal samples** should have a **positive influence** on other **normal samples** (*i.e.* help reduce the loss).

Intuition

On average, **normal samples** should have a **positive influence** on other **normal samples** (*i.e.* help reduce the loss).

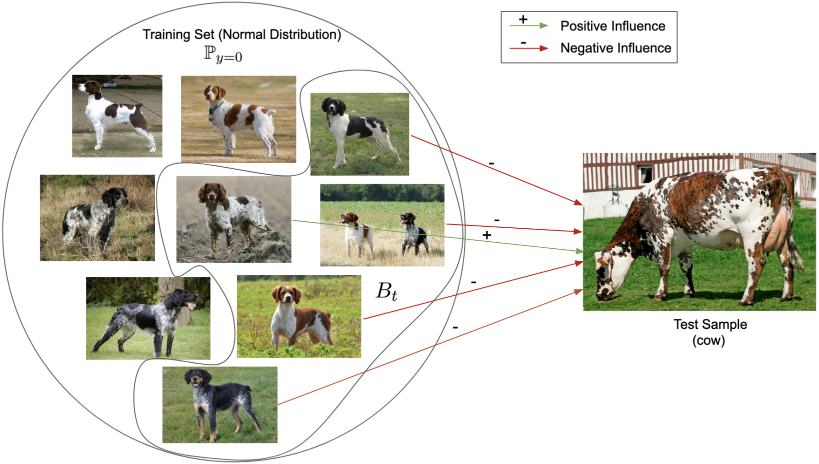


Intuition

on the contrary, on average, **normal samples** should have a **negative influence** on **anomaly samples** (*i.e.* contribute to increase the loss).

Intuition

on the contrary, on average, **normal samples** should have a **negative influence** on **anomaly samples** (*i.e.* contribute to increase the loss).



TracInAD (2)

Consider the following procedure:

- Train a deep AD model using **only *normal* samples.**
- In inference, the anomaly score is the average influence of a subsample of the training set.

TracInAD (3)

Formally:

- Consider f_θ a **deep AD model** parametrized by $\theta \in \Theta \subseteq \mathbb{R}^p$.
- $\{t_1, \dots, t_k\}$ **checkpoints** at which parameters are saved (*e.g.* one epoch).
- $\texttt{TracInCP}(x, x') = \sum_{i=1}^{k} \eta_{t_i} \nabla \ell(\theta_{t_i}, x') \cdot \nabla \ell(\theta_{t_i}, x)$ a more **computationally efficient influence estimation.**
- B_t a **random subsample of the training set** of fixed size m.

TracInAD (4)

TracInAD

The anomaly score for sample x' is set as

$$\begin{aligned} \texttt{TracInAD}(x') &= \frac{1}{m} \sum_{x \in B_t} \texttt{TracInCP}(x, x') \\ &= \frac{1}{m} \sum_{x \in B_t} \sum_{i=1}^{k} \eta_i \nabla \ell(\theta_{t_i}, x) \cdot \nabla \ell(\theta_{t_i}, x') \end{aligned}$$

Experiments

We experiment on 4 baseline tabular datasets with a **reconstruction-based AD method** based on a VAE. We obtain **competitive results** on several datasets.

Method	Dataset							
	Arthythmia		Thyroid		KDD		KDDRev	
	F_1 Score	σ	F_1 Score	σ	F_1 Score	σ	F_1 Score	σ
OC-SVM	45.8		38.9		79.5		83.2	
E2E-AE	45.9		11.8		0.3		74.5	
LOF	50.0		52.7		83.8		81.6	
DAGMM	49.8		47.8		93.7		93.8	
GOAD	52.0	2.3	74.5	1.1	98.4	0.2	98.9	0.3
NeuTraL AD	60.3	1.1	76.8	1.9	99.3	0.1	99.1	0.1
Shenkar et al.	**61.8**	1.8	76.8	1.2	**99.4**	0.1	**99.2**	0.3
`TracIn AD`	54.6	2.1	**77.6**	5.4	82.1	0.6	98.8	0.3

TABLE I
ANOMALY DETECTION ACCURACY

Conclusion (1)

We proposed a novel method which:

- Includes influence measures.
- Can be applied on **any deep AD method**.
- Shows competitive results with SOTA methods.
- But displays however higher standard deviation.

Conclusion (2)

A few questions still remain:

- Are there other ways to aggregate the influence scores ? (e.g. $\max$ instead of the mean)
- How much is `TracInAD` affected by contaminated data (i.e. presence of anomalies in the training set) ?

Thank you for your attention !
Questions ?

For more details please visit: https://arxiv.org/abs/2205.01362

References I

Bergman, L. and Hoshen, Y. (2020). Classification-based anomaly detection for general data. In *International Conference on Learning Representations*.

Kim, K. H., Shim, S., Lim, Y., Jeon, J., Choi, J., Kim, B., and Yoon, A. S. (2020). Rapp: Novelty detection with reconstruction along projection pathway. In *International Conference on Learning Representations*.

Kong, Z. and Chaudhuri, K. (2021). Understanding instance-based interpretability of variational auto-encoders. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W., editors, *Advances in Neural Information Processing Systems*, volume 34, pages 2400–2412. Curran Associates, Inc.

Pruthi, G., Liu, F., Sundararajan, M., and Kale, S. (2020). Estimating training data influence by tracing gradient descent.

References II

Qiu, C., Pfrommer, T., Kloft, M., Mandt, S., and Rudolph, M. (2021). Neural transformation learning for deep anomaly detection beyond images. In Meila, M. and Zhang, T., editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 8703–8714. PMLR.

Ruff, L., Vandermeulen, R., Goernitz, N., Deecke, L., Siddiqui, S. A., Binder, A., Müller, E., and Kloft, M. (2018). Deep one-class classification. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4393–4402. PMLR.

References III

Schölkopf, B., Williamson, R., Smola, A., Shawe-Taylor, J., and Platt, J. (1999). Support vector method for novelty detection. In *Proceedings of the 12th International Conference on Neural Information Processing Systems*, NIPS'99, page 582–588, Cambridge, MA, USA. MIT Press.

Shenkar, T. and Wolf, L. (2022). Anomaly detection for tabular data with internal contrastive learning. In *International Conference on Learning Representations*.

Tax, D. and Duin, R. (2004). Support vector data description. *Machine Learning*, 54:45–66.