Universiteit
Leiden
The Netherlands

Reliable and fair machine learning for risk assessment

Pereira Barata, A.P.

Citation

Pereira Barata, A. P. (2023, April 5). *Reliable and fair machine learning for risk assessment*. *SIKS Dissertation Series*. Retrieved from https://hdl.handle.net/1887/3590289

Version:	Publisher's Version
License:	Licence agreement concerning inclusion of doctoral thesis in the Institutional Repository of the University of Leiden
Downloaded from:	https://hdl.handle.net/1887/3590289

Note: To cite this publication please use the final published version (if applicable).

Chapter 3

Crosslier Detection

Finding anomalous entries is a difficult task with real-world consequences. Consider the following example. Transit of wasteful materials within the EU is highly regulated through a system of permits. Waste processing costs vary greatly depending on the waste category of a permit. Therefore, companies may have a financial incentive to allege transporting waste with erroneous categorisation (i.e., label-noisy samples). Our goal is to assist inspectors of the ILT in selecting potentially manipulated permits for further investigation. For this purpose, we introduce the concept of *crosslier*, of which the definition follows.

Definition 3.1 – ***Crosslier***

> A crosslier is a sample of which (a) the category label is swapped and (b) a proportion of its features is more similarly valued to the features of the samples of the newly-swapped category.

To detect crossliers, we propose the eXPOSE method. Moreover, to facilitate the targeting of crossliers by inspector, we define the *crosslier diagram*.

Definition 3.2 – ***Crosslier diagram***

> A crosslier diagram is a visualisation tool specifically designed for domain experts to easily assess crossliers.

We compare eXPOSE against traditional detection methods in various benchmark datasets. By evidencing the superior performance of our method in targeting these instances of interest, we provide an answer to RQ2(a): given data with label noise, how can noisy-samples be adequately detected?

The current chapter corresponds to the following publication:
Pereira Barata, A., Takes, F. W., van den Herik, H. J., and Veenman, C. J. (2021). The eXPose approach to crosslier detection. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 2312–2319. IEEE

3.1 Crossliers and Miscategorisation

Within the EU, economic proliferation and globalisation have resulted in a increase of transnational waste transportation. The nowadays established List of Waste provides EU member-states with waste categorisation, which promotes appropriate waste handling, particularly relevant for hazardous waste [European Commission, 2018a]. Since transportation of waste poses serious health and environmental risks, all movement of waste must be priorly noticed through a system of permits [European Commission, 2018b]. In the Netherlands, the entity responsible for permit compliance is the ILT. Inspectors must evaluate and determine whether a permit (a) is likely to be compliant and requires no further inspection, or (b) raises concern and requires investigation.

Since different waste categories are encompassed by specific regulations with dissimilar processing costs, companies may have an economic incentive to purposefully miscategorise their waste. Hence, targeting such cases is of utmost importance to the inspectors of the ILT. Given high volume and velocity of data, however, inspectors cannot adequately assess all permits. Therefore, automatic methods are required.

Under the current problem scenario, the usually most-effective supervised learning approaches to instance targeting [Choudhary and Gianey, 2017] are not applicable since no historical labels for misconduct are available. Unsupervised learning techniques are also not suited, given the unspecificity of the retrieved instances. Here we note that for anomaly detection methods, outlyingness alone does not translate to the desired targets, and we further mention the difficulty of detecting samples in high-dimensional data [Venkatesh and Anuradha, 2019].

With respect to data-quality assurance techniques, we remark that they mostly depend on variable distribution assumptions and concentrate on random errors [Liu et al., 2016]. We focus on instances in which the category label and category-correlated feature values have been altered. In other words, our goal is to pinpoint samples with *non-random* changes in feature values which mask the true underlying category label.

To address the current problem of manipulation, we propose the following three contributions:

1. the concept of a *crosslier*: a deviating instance resulting from potentially intentional category manipulation;
2. the EXPOSE method for crosslier detection, by computing the crosslier score of a sample given its category;
3. the *crosslier diagram*: a visualisation tool which allows easy assessment of crossliers.

Albeit motivated by a waste transportation problem, our proposed contributions are intrinsically domain-agnostic and therefore applicable to other fields. Within a dataset with category labels, a crosslier is an instance of which the combination of (1) its set of feature values and (2) the category label are disharmonious.

We consider a crosslier to be a *special* case of an outlier defined as follows.

Definition 3.3 – ***Outlier***

An outlier is a sample of which the feature values differ significantly from those of the other samples.

By special case, we mean that crossliers are outlying instances with *specific* characteristics. More precisely, a crosslier is a specific outlier with some connection regarding a category label; that is, it is a sample of a category which lies *across* other categories.

For completion, here we remark that the terms (a) crosslier and (b) outlier are both a form of (c) anomaly, which is defined as follows.

Definition 3.4 – ***Anomaly***

An anomaly is a sample which, given its features, class label, domain knowledge, or any combination of the three, is significantly different from the remainder of the samples. It is used to broadly refer to a data point which stands out from the dataset.

The relationship between the three terms is depicted in Fig. 3.1. As shown, all crossliers are outliers and all outliers are anomalies, but not all anomalies are outliers and not all outliers are crossliers.

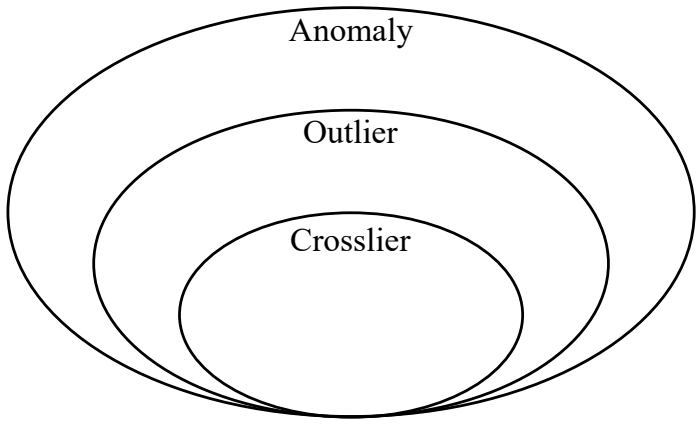

Figure 3.1: **Anomaly, outlier, and crosslier.** Diagram depicting the relationship between the three terms.

The chapter structure follows: Section 3.2 states our problem formally; Section 3.3 discusses past work related to ours; Section 3.4 elaborates our approach in detail; Section 3.5 describes ours experimental setup; Section 3.6 refers to our results; Section 3.7 discusses our method; and Section 3.8 concludes this work and suggests future research directions.

3.2 Problem Description

Given a category-labelled dataset, we defined (in Definition 3.1) a *crosslier* as a sample of which (a) the category label is swapped and (b) a proportion of its features is more similarly valued to the features of samples of the newly-swapped category. To put it simply, we assume that feature values might have been manipulated to mask the true category label.

To detect crossliers, we propose *crosslyingness* as a rankable property expressed as a function, in which the instance with the highest crosslyingness with respect to a category is the most likely crosslier.

Definition 3.5 – ***Crosslyingness***

> Crosslyingness is a rankable property indicative of the degree to which a sample is considered a crosslier.

Accordingly, either (1) crossliers fall within the cluster of some other category, or (2) crossliers lie across other categories. To illustrate, we present Fig. 3.2; four different categories $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$, and $\mathcal{D}$ are denoted, with crossliers as $\mathcal{A}^*$, $\mathcal{B}^*$, $\mathcal{C}^*$, and $\mathcal{D}^*$.

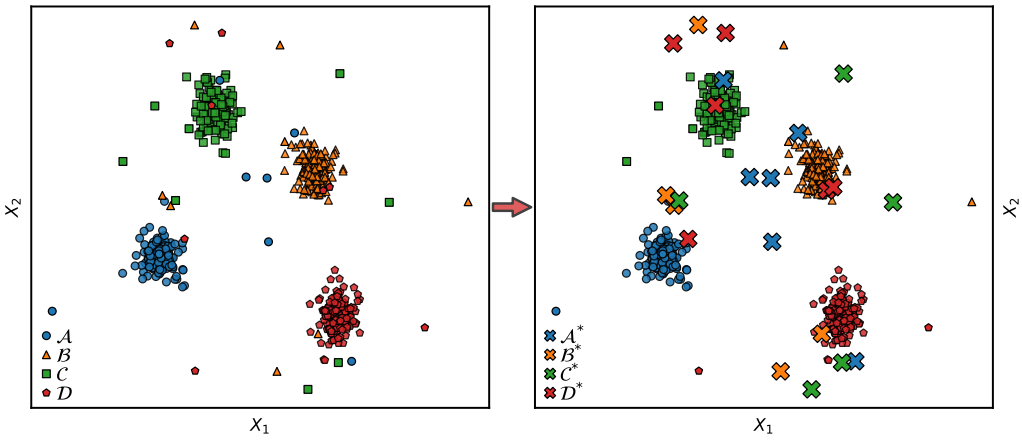

Figure 3.2: **Crosslier detection.** Samples with features X_1 and X_2, pertaining to either category $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$, or $\mathcal{D}$ (left). Crossliers are marked as crosses (right).

Formally, let $\mathcal{D}$ be a distribution of random variables $(X, Z) \in \mathcal{X} \times \mathcal{Z}$, where $\mathcal{X} \subseteq \mathbb{R}^m$, $\mathcal{Z} = \{z_1, z_2, \ldots, z_q\}$, and $z \in \mathcal{Z}$ is one of the q different category labels. Let also $(x_1, z_1), \ldots, (x_n, z_n)$ represent the samples drawn from $\mathcal{D}$. Our goal is to find, for each unique category $z \in \mathcal{Z}$, a function $f_z(x)$ which scores the crosslyingness of $x_i \in X$ with $z_i = z$.

3.3 Related Work

In this section, we provide a brief overview of three techniques typically used to address anomaly detection problems. In this sense, we consider a crosslier to be a particular type of data anomaly, with specific characteristics as described in the previous sections.

We report on previous work which applied supervised and semi-supervised learning techniques (Section 3.3.1), unsupervised learning methods (Section 3.3.2), and data quality assurance techniques (Section 3.3.3). We further disclose their non-applicability to our scenario.

3.3.1 Supervised and Semi-supervised Learning

In the presence of labels indicative of previously-recognised noncompliance, the problem can be approached as a supervised learning task. Three examples are: (1) detecting insurance fraud [Subudhi and Panigrahi, 2020]; (2) exposing deceitful telecommunication users [Li et al., 2018]; and (3) identifying irregular heart beat patterns [Vollmer et al., 2017]. The choice of algorithm is rather diverse. We mention three of them: (1) SVMC [George and Vidyapeetham, 2012]; (2) multilayer perceptron [Mulongo et al., 2020]; and (3) random forest [Alazzam et al., 2019].

For the case where both labelled and unlabelled instances are available, a semi-supervised learning approach is suitable [Chapelle et al., 2006]. This framework can, as an example, make use of clustering algorithms assuming that data points within the same cluster probably share the same label [Xiang and Min, 2010]. Another approach to improve on the selection of inspection targets is to consider the unlabelled instances as pertaining to the negative class (i.e., the class which is not of interest) [Jacobusse and Veenman, 2016]. Here, the assumption is that the incidence of inspection targets within the unlabelled data is small enough as to be made negligible towards learning. Yet, our data does not possess target labels, making these techniques inapplicable.

3.3.2 Unsupervised Learning

A straightforward alternative is to find deviating cases through *anomaly detection* techniques using unsupervised methods. The assumption is that the most probable samples to target are the ones that differ in an extreme way from all others in their category (i.e., outliers). Such techniques have been applied to system intrusion detection [Zanero and Savaresi, 2004], maritime traffic anomaly flagging [Vespe et al., 2012], and image curation [Liu et al., 2014], amongst others. Four examples of the successful algorithms used are: (1) isolation forest (IF) [Liu et al., 2012]; (2) local outlier factor (LOF) [Breunig et al., 2000]; (3) nearest-neighbour [Amer and Goldstein, 2012]; and (4) k-means clustering [Muniyandi et al., 2012].

There are at least three intrinsic obstacles with unsupervised methods. The first obstacle is their dependency on distance metrics (Minkowski measures) to define outlyingness, which makes them sensitive to feature scaling. The second obstacle arises when dealing with high-dimensional data [Liu et al., 2017], particularly when attempting to estimate densities empirically [Santos et al., 2019a]. The third obstacle is that, through manipulation of only a proportion of features — as per the problem description (see Section 3.2)— target samples (crossliers) may not stand out. To illustrate, we present Fig. 3.3.

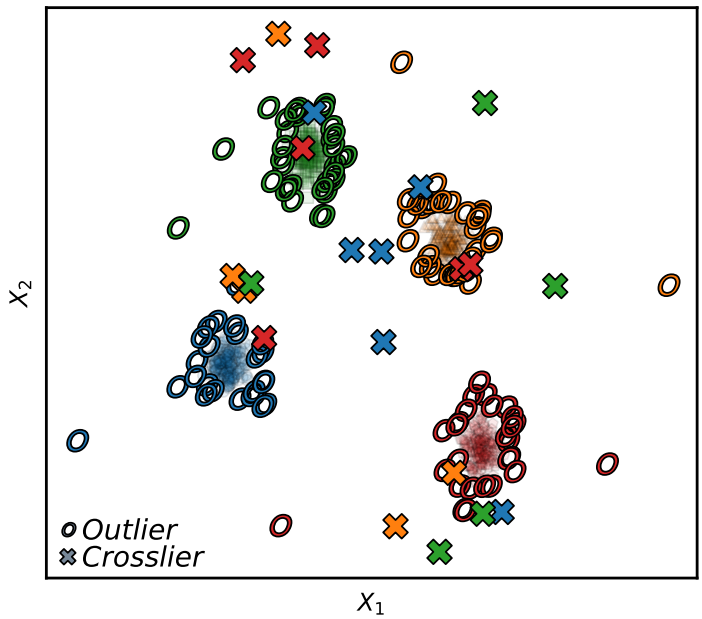

Figure 3.3: **Distinction between outlier and crosslier.** Four-category example from Fig. 3.2. Crossliers are marked as crosses and outliers are denoted as circles. Transparency values for data clusters have been raised for visualisation.

Fig. 3.3 builds on the example in Fig. 3.2 by applying the IF algorithm as per [Liu et al., 2008]. Here we see how data points flagged as outliers *do not* represent the target crossliers; hence, we should consider the distribution of categories when marking instances as crosslying. The issue with using traditional anomaly detection methods towards finding crossliers is evidently illustrated: most flagged instances are arguably not outlying with respect to their clusters. The insensitivity shown makes anomaly detection methods precarious to address our problem. Ultimately, not all outliers are crossliers since not all of them possess the specific category-related characteristics we seek.

3.3.3 Data Quality Assurance

By considering an outlier to be anomalous, and therefore an inspection candidate, one could argue that the abnormal values by which outlyingness is attributed can be caused by erroneous data entries on the permit category. Here, data quality assurance techniques can be used for detection [Bonner et al., 2015]. Typical methods involve, for example, assumptions over feature distributions [Mariet et al., 2016] and cross-referencing datasets for dependency-matching or constraint-mining [Rekatsinas et al., 2017, Chu et al., 2013]. Our scenario does not allow for reliable cross-dataset linkage due to the lack of entity identifiers. Furthermore, despite the existence and usage of both univariate and multivariate constraints, the constraints are not generated with respect to an ulterior task. In other words, the assumptions over feature distributions need not hold for the category distributions we are interested in.

In summary, the current literature is ill-equipped to adequately address our issue of discriminating towards crosslying instances, which translate to permits of interest to inspectors.

3.4 The EXPOSE Method

Here, we detail the proposed EXPOSE for the detection of crossliers. As defined in Section 3.2, the aim is to find a function $f_z(x)$ that determines the crosslier score of sample $x \in X$ with category label z. The EXPOSE method is data-driven in the sense that it uses a learning function to obtain the scores for a dataset with category labels. Since the whole dataset is category-labelled by definition, all samples can obtain a crosslier score. We follow a supervised learning approach, where the crosslying score is determined per category on a left out part in order to obtain an independent score. As a result, we need to optimise several learners as in a CV setup. Therefore, these learned functions must be calibrated to make the scores comparable among each other.

Below, we first describe the setup to obtain the learners in a supervised way. We then elaborate on the model selection and model calibration steps per data subset based on CV. The learners collectively yield the overall crosslier score function. We finalise the method section with the crosslier diagram, a tool to visualise crosslier scores and pinpoint suspect samples.

3.4.1 Classification Setup

Consider the distribution $\mathcal{D}$ defined in Section 3.2. For a fixed category z, $(x_1, y_1), \ldots, (x_n, y_n)$ are samples of $\mathcal{D}$ in which

$$y_i = \begin{cases} 1, \text{ if } z_i = z \\ 0, \text{ otherwise} \end{cases} \tag{3.1}$$

Given $\mathcal{D}$ and a loss function $\mathcal{L}$, the task of the learner is to find a function $f \in \mathcal{F}$ through empirical risk minimisation [Vapnik, 2013]:

$$\arg\min_{f \in \mathcal{F}} \hat{\mathcal{R}}_{\mathcal{D},\mathcal{L},f} \tag{3.2}$$

where

$$\hat{\mathcal{R}}_{\mathcal{D},\mathcal{L},f} = \frac{1}{n} \cdot \sum_{i=1}^{n} \mathcal{L}(f(x_i), y_i) \tag{3.3}$$

Depending on the chosen learner, the curse of dimensionality is addressed by incorporating either regularisation, feature selection, or both protocols in the learning task [Sharma et al., 2017]. These protocols also alleviate overfitting and promote classifier robustness by reducing the complexity of the final model [Gupta et al., 2016].

All regularisation parameters given prior to the learning task can be optimally retrieved through hyperparameter optimisation [Claesen and De Moor, 2015, Bergstra and Bengio, 2012]. The learners to be applied within a specific problem can also be optimally selected.

3.4.2 Model Selection

A model is selected based on classification performance. For each candidate learner that is applicable to a problem and their respective hyperparameters, the estimated classification performance is measured in terms of AUC through CV [Flach, 2016]. The choice of CV strategy is dependent on $\mathcal{D}$, as the appropriate number of folds and splitting strategy relate to $\mathcal{Z}$ and the respective $P(y)$, as well as sample size n. Model calibration is also subject to the CV strategy, detailed further.

Formally, consider the dataset D, with distribution $\mathcal{D}$. For a given $k \in \{1, 2, \ldots, K\}$, $K > 1$, let test set D_k^{ts} and training set D_k^{tr} be independent and identically distributed subsets of D such that

$$\bigcap_{k=1}^{K} D_k^{ts} = \emptyset, \bigcup_{k=1}^{K} D_k^{ts} = D, \text{and } D_k^{tr} = D \setminus D_k^{ts} \tag{3.4}$$

Fixing on k, we define test and training sets D_ℓ^{ts} and D_ℓ^{tr}, respectively, as independent and identically distributed subsets of D_k^{tr}, for $\ell \in \{1, 2, \ldots, L\}$ and $L > 1$, such that

$$\bigcap_{\ell=1}^{L} D_\ell^{ts} = \emptyset, \bigcup_{\ell=1}^{L} D_\ell^{ts} = D_k^{tr}, \text{and } D_\ell^{tr} = D_k^{tr} \setminus D_\ell^{ts} \tag{3.5}$$

Given D and sets of learners $\{\Psi_1, \Psi_2, \ldots, \Psi_r\}$ with hyperparameters $\{\phi_1, \phi_2, \ldots, \phi_p\}$, the final model is selected by maximising the estimated AUC with K and L folds, comprised of learner ${}^*\Psi$ and hyperparameters $\phi_k \in \{\phi_1, \phi_2, \ldots, \phi_K\}$. AUC is directly linked to crosslyingness, as detailed ahead.

Learner ${}^*\Psi$ and hyperparameters ϕ_k are used to generate the crosslier scores. Since EXPOSE generates crosslier scores from a collection of models learned on independent data subsets to avoid overfitting, the output of each model is not comparable across models. We enforce model comparability through model calibration.

3.4.3 Crosslier Score

To transform the output of uncalibrated models into a calibrated output, Platt scaling [Platt et al., 1999] is used. The original output $\hat{y}$ of a learned model given input x thus becomes the estimated posterior probability $\hat{P}(y|x)$. Given z, the crosslier score function f_z is defined as the information content [Jones, 1979] of a sample x from category z:

$$f_z(x) = -\log_2 \hat{P}(y|x) \tag{3.6}$$

The choice of $-\log_2$ translates to: (1) the score difference between samples with low and high posterior probabilities are augmented; and (2) scores are easily interpretable, in which a posterior 1 returns a score 0, and a posterior 0.5 returns 1. Heuristically, samples with crosslier score greater than 1 can be considered crossliers and are rankable by crosslyingness according to their respective crosslier scores. The estimated AUC model performance relates to the crosslier scores. By definition, poor-performing models output calibrated posterior probabilities close to 0.5. Therefore, the crosslier scores will lie close to 1 for all samples. With high AUC models, the range of crosslier scores is allowed to widen.

Formally, let x_k and y_k represent the variable values of samples $(x, y) \in D_k^{ts}$ for a given k. The estimated posterior is then given as

$$\hat{P}(y|x) = \bigcup_{k=1}^{K} \hat{P}(y_k|x_k) \tag{3.7}$$

in which,

$$\hat{P}(y_k|x_k) = \frac{1}{L} \cdot \sum_{\ell=1}^{L} [f_\ell^k({}_\phi^* \Psi_k^{tr}(x_k))] \tag{3.8}$$

where ${}_\phi^* \Psi_k^{tr}(x_k)$ is the output of ${}^*\Psi$ learned on $(x, y) \in D_k^{tr}$ with hyperparameters ϕ_k, given input x_k, and f_ℓ^k is the sigmoid function with parameters α^* and β^*

$$f_\ell^k(u) = \frac{1}{1 + e^{-(\alpha^* + \beta^* \cdot u)}} \tag{3.9}$$

in which

$$\alpha^*, \beta^* = \arg\min_{\alpha,\beta} - \sum_{(x,y) \in D_\ell^{ts}} [\mu \cdot log(p) + (1 - \mu) \cdot log(1 - p)] \tag{3.10}$$

where

$$\mu = \begin{cases} \frac{(\sum_{y \in D_\ell^{ts}} y) + 1}{(\sum_{y \in D_\ell^{ts}} y) + 2}, \text{ if } y = 1 \\ (|D_\ell^{ts}| - (\sum_{y \in D_\ell^{ts}} y) + 2)^{-1}, \text{ otherwise} \end{cases} \tag{3.11}$$

and

$$p = \frac{1}{1 + e^{-(\alpha + \beta \cdot {}_\phi^* \Psi_\ell^{tr}(x))}} \tag{3.12}$$

In Eq. 3.12, ${}_\phi^* \Psi_\ell^{tr}(x)$ is the output of ${}^*\Psi$ learned on $(x, y) \in D_\ell^{tr}$ with hyperparameters ϕ_k, given input $x \in D_\ell^{ts}$.

3.4.4 Crosslier Diagram

At the basis of the crosslier diagram (see Definition 3.2) lies an interactive tool which discriminates individual samples based on their crosslier score. Existing tools such as box, swarm, and violin plots were not suited since: (1) box plots do not present all samples that might be relevant crossliers; (2) swarm plots do not function well for a large number of samples; and (3) violin plots do not exhibit any samples in their output.

The diagram is a mapping of the output of $f_z(x)$ onto a horizontal axis where x are samples of category z. To each plotted sample we add a Gaussian-generated vertical value so that even if two or more samples have the same crosslier score they do not entirely overlap.

Finally, the crosslier diagram can display related domain-specific information of a sample by hovering over it. In the context of real-world transportation data, we present the crosslier diagram (Fig. 3.4) in the upcoming Section 3.6 as part of our experimental results.

3.5 Experiments

In this section, we describe our experiments. Two setups are considered, viz. (a) waste transportation setup and (b) benchmark setup. Within the first setup, EXPOSE is applied to the waste permit dataset (Section 3.5.1). In the second setup, we compare our method to anomaly detection methods in a controlled environment (Section 3.5.2). The resources described in this section are made available online [Pereira Barata, 2020].

3.5.1 Waste Transportation Setup

In this section, we discuss: (1) data; (2) learners; and (3) selection and calibration of the best model.

Data

The dataset was generated and provided by the ILT. It represents solicitations of waste transportation events across Europe (2009–2015), encompassing a total of $876,311$ waste transportations. Each row represents an individual transportation event. Several rows are linked by a permit identifier, where permits are the units of interest to inspectors of the ILT. We followed an aggregation strategy with respect to permit identifiers. The aggregation process produced $11,740$ instances, each with a waste category (out of 20 total different waste categories) and 49 variables which were a mixture of numerical and nominal features.

Learners

We experimented with (a) linear and (b) non-linear learners to find the best performing model for each waste category. First, an elastic net-regularised logistic regression (LR) learner was deployed, with hyperparameters λ and ϵ referring to the regularisation coefficient, and the ratio of $L1$ to $L2$-regularisation, respectively. Besides its broad usage and proven efficacy [Rosario, 2004, Wang, 2005, Mok et al., 2010], advantages of this learner are, for example: its calibrated output probabilities (hence, not requiring any further calibration); and its resilience to overfitting given low complexity and regularisation [Kleinbaum and Klein, 2010].

Second, a non-linear gradient boosted tree framework (XGBC) was considered [Friedman, 2001], with 100 additive trees. Each tree was allowed a maximum depth of 3 with regularisation parameter $\lambda = 1$. This learner is widely accepted as a state-of-the-art solution to supervised problems [Pafka, 2019] in terms of scalability, robustness to noisy samples, and classification performance.

Selection and calibration

To select and calibrate the best model, we applied nested-CV in a stratified manner [Stone, 1974] with $K = 10$, $L = 10$ as described in Section 3.4. Stratification is selected to ensure that each category is represented in each fold with the same relative frequency as in the full dataset. A grid-search [Chan and Treleaven, 2015] was applied to find the optimal set of LR regularisation parameters λ and ϵ. Each parameter was set to one of 21 distinct values, in ranges $[10^{-3}, 10^{3}]$ logarithmic and $[0, 1]$ linear, respectively, for a total of 441 sets of candidate hyperparameters. Since XGBC is relatively insensitive to hyperparameter changes, as shown in the experimental results of [Xia et al., 2017], we did not perform hyperparameter optimisation for this classifier. The best model for each category was used to generate the crosslier scores and crosslier diagrams (Section 3.6).

3.5.2 Benchmark Setup

In this section, we discuss: (1) data; (2) preprocessing; (3) crosslier synthesis; and (4) evaluation.

Data

Twenty binary classification datasets were retrieved from *openML*: an open, organised, and online ecosystem for machine learning [Vanschoren et al., 2014]. They are real-world datasets from different domains. Target classes were treated as the categories $\mathcal{Z}$. Table 3.1 summarises each dataset with identifier ID, n instances, and m features of which u are numeric. The datasets were chosen such that n, m, and u are heterogeneous across datasets.

Preprocessing

Numeric features values were scaled to a $[0, 1]$ range to accommodate feature scale-sensitive methods. Non-numeric features were $\{0, 1\}$-binarised per unique value.

Table 3.1: Datasets retrieved for crosslier simulations.

ID	n	m	u	ID	n	m	u
446	200	7	6	40705	959	44	42
40	208	60	60	31	1000	20	7
1495	250	6	0	1494	1055	41	41
53	270	13	13	40706	1124	10	0
40710	302	14	5	1462	1372	4	4
59	351	34	34	1504	1941	33	33
40690	512	9	0	1487	2534	72	72
1063	522	21	21	1485	2600	500	500
335	554	6	0	41143	2984	144	8
1510	569	30	30	41144	3140	259	259

Crosslier synthesis

To simulate a real-world scenario, crossliers were synthesised by replacing category labels and feature values. Different proportions of both label and feature manipulation were considered extensively. The proportion of label-swapped samples for each category per dataset was $\rho_y \in \{.01, .05, .1, .15, .2, .25, .3, .35, .4\}$. To recreate the scenario in which feature values are manipulated to simulate another category, samples which were label-swapped had a proportion of their feature values replaced. The proportion of randomly-selected features to have their values replaced was $\rho_x \in \{0, .05, .1, .15, .2, .25, .3, .35, .4\}$.

Replacement values were drawn from univariate distributions with parameters estimated from the features of the category being mimicked, modelled as either: (a) the normal distribution $\mathcal{N}(\hat{\mu}, \hat{\sigma})$ for numeric features, where $\hat{\mu}$ is the estimated mean and $\hat{\sigma}$ is the estimated standard deviation; or (b) the multinomial distribution with estimated event probabilities $\{\hat{p}_1, \hat{p}_2, \ldots, \hat{p}_\pi\}$ where π is the number of unique feature values, otherwise. Crossliers were generated 10 times with different random initialisation seeds for all datasets per configuration (ρ_x, ρ_y) to account for randomness. Both categories per dataset were corrupted with crossliers before any method was applied.

Methods

The EXPOSE method was compared to two well-established anomaly detection methods: LOF and IF, mentioned in Section 3.3.2. The previously-established methods were not designed to detect crossliers.

To promote a reasonable comparison, EXPOSE was applied with a single set of learner and hyperparameters and no optimised model selection was performed. The model selected was a tree-based gradient boost learner and default hyperparameters of 100 trees of maximum depth 3 with regularisation $\lambda = 1$ [Chen and Guestrin, 2016]; calibration values K and L were set to 10. LOF neighbourhood size was set to 20 and IF number of trees was set to 100.

Evaluation

The crosslier scores of EXPOSE were generated as in Section 3.4; the anomaly scores of the anomaly detection methods were generated category-wise. For each category, crosslier detection performance was measured in AP [Liu and Özsu, 2009], a common measure in anomaly detection assessment [Xu et al., 2018]. Accordingly, the targets are the crossliers in each category. The performance of both categories in each configuration (ρ_x, ρ_y) were jointly averaged per dataset, and across initialisation seeds.

3.6 Results

Here, we present findings relative to both experimental setups: (*a*) EXPOSE applied to the real-world scenario of waste transportation in the inspection domain; and (*b*) EXPOSE compared to other anomaly detection methods in a controlled environment with benchmark datasets.

3.6.1 Waste Transportation

When applied to the waste transportation data, we show firstly the estimated AUC performances yielded by both candidate models LR and XGBC. The next step was presenting the crosslier diagrams of waste categories to the inspectors for assessment. Waste category 4 (waste from textile industries) was not shown due to insufficient number of instances.

Model performance and selection

Table 3.2 shows the estimated AUC performances and measured standard deviations yielded during the model selection step of EXPOSE, which were used to select the best model per category for crosslier detection. Values in bold indicate the highest performance per category of which the model was chosen.

Table 3.2: Model performances per waste category.

Category	LR	XGBC
1	0.983 ± 0.008	$\mathbf{0.985 \pm 0.010}$
2	0.868 ± 0.044	$\mathbf{0.919 \pm 0.037}$
3	0.868 ± 0.020	$\mathbf{0.908 \pm 0.027}$
—	—	—
5	0.672 ± 0.092	$\mathbf{0.755 \pm 0.082}$
6	0.740 ± 0.038	$\mathbf{0.794 \pm 0.037}$
7	0.776 ± 0.016	$\mathbf{0.821 \pm 0.015}$
8	0.798 ± 0.026	$\mathbf{0.856 \pm 0.025}$
9	0.867 ± 0.047	$\mathbf{0.915 \pm 0.047}$
10	0.737 ± 0.032	$\mathbf{0.788 \pm 0.035}$
11	0.815 ± 0.021	$\mathbf{0.896 \pm 0.016}$
12	0.860 ± 0.032	$\mathbf{0.897 \pm 0.031}$
13	0.609 ± 0.063	$\mathbf{0.720 \pm 0.062}$
14	0.776 ± 0.034	$\mathbf{0.817 \pm 0.024}$
15	0.841 ± 0.019	$\mathbf{0.883 \pm 0.016}$
16	0.695 ± 0.016	$\mathbf{0.753 \pm 0.019}$
17	0.845 ± 0.023	$\mathbf{0.889 \pm 0.022}$
18	0.894 ± 0.015	$\mathbf{0.921 \pm 0.015}$
19	0.806 ± 0.014	$\mathbf{0.851 \pm 0.013}$
20	0.719 ± 0.024	$\mathbf{0.779 \pm 0.027}$

XGBC provided the best performance for all categories and was selected to generate the crosslier diagrams. For clarity, AUC does not measure the performance of crosslier detection since no crosslier labels exist in this real-world problem.

Crosslier diagrams

In Fig. 3.4 the crosslier diagrams with scores generated by the selected model XGBC are shown. For demonstration purposes, we show crosslier diagrams of four waste categories: (1) exploration and treatment of minerals; (2) agriculture, food preparation, and processing; (9) waste from photography industry; and (18) human or animal healthcare. In addition, the interactive aspect of the diagram is represented for a sample of waste category 9, in which its permit identifier (ID 4358) and crosslier score (1.41) are shown.

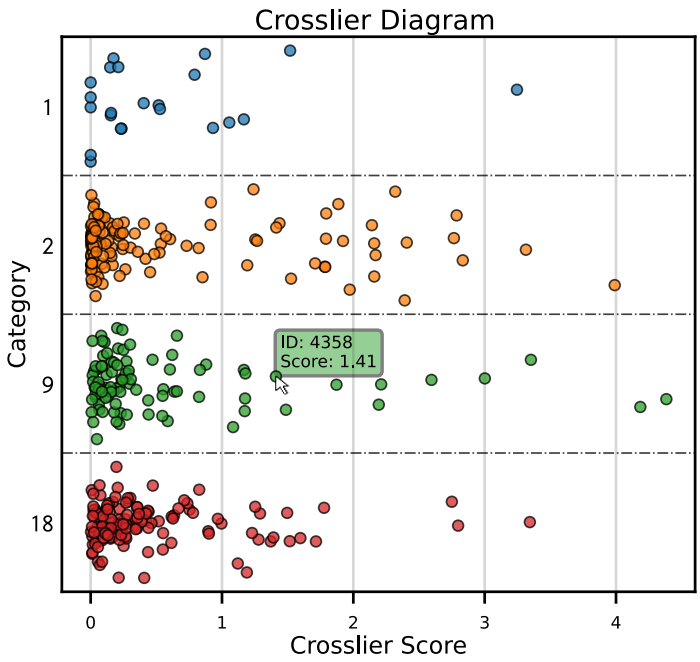

Figure 3.4: **Crosslier diagrams of four waste categories.** Hovering over an instance highlights its identifier (4358) and crosslier score (1.41).

Inspection domain

The inspectors of ILT were provided with the crosslier diagrams. They analysed the permit cases across waste categories according to the given crosslier scores. Their assessment was that the authenticity of most of the high-scoring permits was sufficiently doubtful and that further investigation was necessary to establish compliance. All in all, the crosslier diagram was considered a valuable expansion of their tool set, especially when compared to spreadsheet analysis.

3.6.2 Benchmark

The outcome of our experiments with respect to controlled crosslier detection is to be seen in Fig. 3.5. We present the results for the three methods: EXPOSE, LOF, and IF. Fig. 3.5 shows the mean (AP) across 20 datasets, for 81 configurations of (ρ_x, ρ_y), each with 10 random initialisations of crosslier synthesis.

Lighter (darker) cell tones indicate higher (lower) values of performance. Each number indicates the yielded AP performance for each (ρ_x, ρ_y) configuration with which we experimented. For every possible setting (i.e., heatmap cell), EXPOSE yielded a higher mean performance than any of the other methods. The differences in performance diminish as both ρ_x and ρ_y increase.

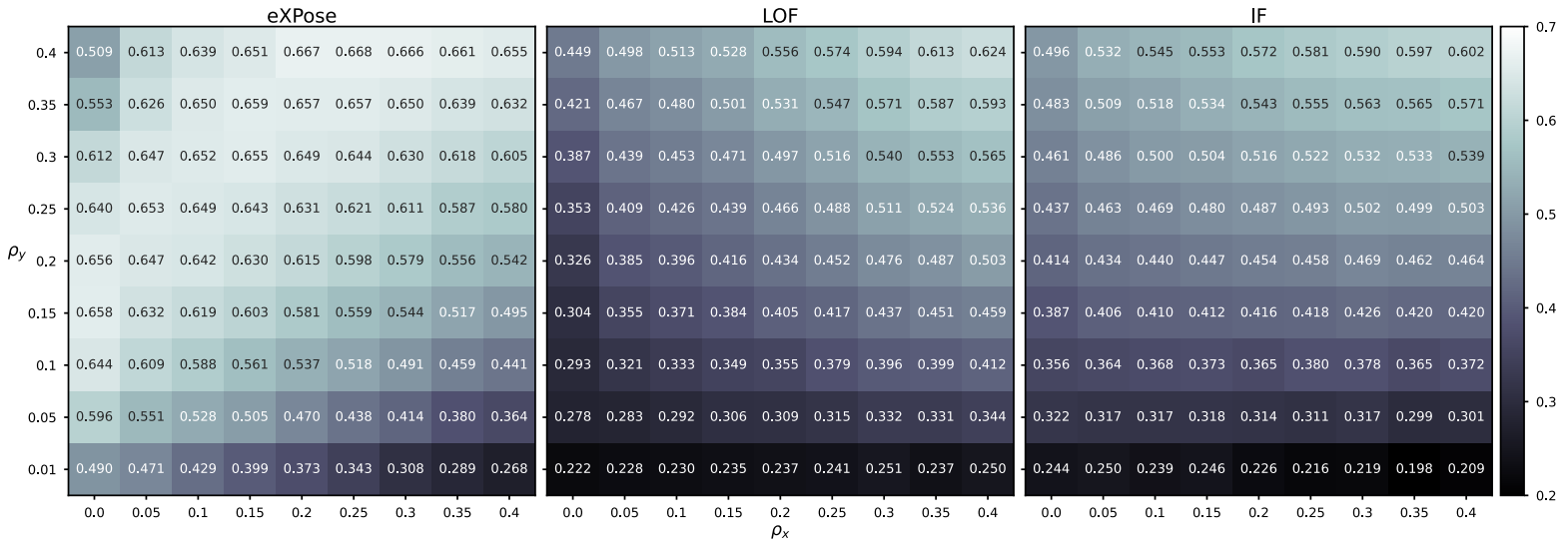

Figure 3.5: **Crosslier detection performance across different methods.** Heatmaps depict the AP scores for our method (EXPOSE), LOF, and IF. Performance values were averaged across datasets and random initialisations. In the vertical axis, ρ_y is the proportion of samples which have been category-swapped. In the horizontal axis, ρ_x denotes the proportion of features (in category-swapped samples) of which the values were replaced.

Note that to perform a correct comparison, EXPOSE was not subject to any optimisation: the model selection step was reduced to a single learner with a single set of default hyperparameters. When deployed onto a real-world scenario, model selection should be applied to select the best possible learner and hyperparameter configuration, as described in Section 3.4.

3.7 Discussion

The EXPOSE method is evidently better at detecting crossliers through the exploitation of category models, when compared to standard anomaly detection methods. This was expected, as crossliers are defined based on their feature values in a category-wise manner. High dimensionality and feature dependence are also better dealt with through the appropriate selection of learner with adequate feature selection and regularisation protocols.

The implementation of the EXPOSE method is to be seen as a wrapper over different components: at its core, it is a data-driven category-modelling method using learner functions. Score calibration is applied and, even though a selected model might have a low AUC, the crosslier scores are —we argue— reliable.

For low AUC values, the crosslier scores will tend to cluster at 1 (corresponding to the posterior 0.5). In this sense, EXPOSE will not *expose* a sample unless its respective category is well modelled (high AUC). This property ensures adequate precision of the sample *exposed* and is of particular relevance when dealing with sensitive Inspectorate domains where wrongly-targeting instances has negative outcomes. Assuming sensible feature values and category labels, a high AUC depends only on the learner and hyperparameters selected.

3.8 Chapter Conclusion

In the present work, we (1) defined a specific type of data anomaly, which we term *crosslier*, (2) introduced the EXPOSE method to *crosslier detection*, and (3) designed the *crosslier diagram*, a visualisation tool to represent crossliers evidently. We showed that conventional anomaly detection methods (LOF and IF) are ill-suited for crosslier detection when compared to eXPose.

Although domain-insensitive, EXPOSE produced valuable domain-specific insights into the problem scenario of targeting potentially fraudulent permits of waste transportation across European countries. We defined *crosslier* as an instance which is more similar to other categories than its own; in other words, it is a sample which likely carries company misconduct.

Extensive preprocessing and optimisation steps were performed which culminated in well-performing (high AUC) models of waste categories. Accordingly, the feature values collected in the waste permits allow for suitable differentiation. This finding shows that administrative data allow for compliance checking. After presenting the crosslier diagrams to the inspectors, their assessment was on par with the expected workings of our EXPOSE method: (1) detected crossliers were considered suspicious, and (2) were marked for further inspection. We remark that these cases had gone undetected in standard permit review operations. So, the crosslier diagram was considered by the inspectors a beneficial extension to current methods.

One clear limitation of our experimental setup is, however, that no direct link can be made between (a) hyperparameter optimisation towards AUC performance and (b) crosslier detection performance. While, by definition, it holds that higher classification performance enables more extreme crosslier scores than lower classification performance, the nature of the relationship between the 2 aforementioned points (a) and (b) should be empirically assessed. To this end, the work by [Van Rijn and Hutter, 2018] would prove invaluable towards efficiently selecting the set of hyperparameters over which the optimisation search should be performed.

As a different future research direction, we recommend close cooperation with the inspectors for the following three reasons: (1) by receiving their feedback on the inspected crosslying permits, our method is further validated; (2) we can use the inspected crosslying cases as labelled instances in a supervised learning scenario towards compliance/non-compliance modelling; and (3) EXPOSE is applicable to other problems within the Inspectorate, which further aids the inspectors.