The eXPose approach to crosslier detection

Pereira Barata, A.P.; Takes, F.W.; Herik, H.J. van den; Veenman, C.J.

2020 25th International Conference on Pattern Recognition (ICPR)
Milan, Italy, Jan 10-15, 2021

The eXPose Approach to Crosslier Detection

António Pereira Barata
LIACS, Leiden University, the Netherlands
ILT, Ministry of Infrastructure and Water Management, the Netherlands
a.p.pereira.barata@liacs.leidenuniv.nl

Frank W. Takes
LIACS, Leiden University, the Netherlands
f.w.takes@liacs.leidenuniv.nl

H. Jaap van den Herik
LCDS, Leiden University, the Netherlands
h.j.van.den.herik@law.leidenuniv.nl

Cor J. Veenman
Data Science Department, TNO, the Netherlands
LIACS, Leiden University, the Netherlands
c.j.veenman@liacs.leidenuniv.nl

***Abstract*—Transit of wasteful materials within the European Union 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.**

Our goal is to assist inspectors in selecting potentially manipulated permits for further investigation, making their task more effective and efficient. Due to data limitations, a supervised learning approach based on historical cases is not possible. Standard unsupervised approaches, such as outlier detection and data quality-assurance techniques, are not suited since we are interested in targeting non-random modifications in both category and category-correlated features.

For this purpose we (1) introduce the concept of *crosslier*: an anomalous instance of a category which lies *across* other categories; (2) propose *eXPose*: a novel approach to crosslier detection based on supervised category modelling; and (3) present the *crosslier diagram*: a visualisation tool specifically designed for domain experts to easily assess crossliers. We compare eXPose against traditional outlier detection methods in various benchmark datasets with synthetic crossliers and show the superior performance of our method in targeting these instances.

***Index Terms*—crosslier, anomaly, detection, visualisation.**

I. Introduction

Within the European Union (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 [1]. Since transportation of waste poses serious health and environmental risks, all movement of waste must be priorly noticed through a system of permits [2]. In the Netherlands, the entity responsible for permit compliance is the Human Environment and Transport Inspectorate (ILT).

In the ILT, inspectors must evaluate and determine whether (1) a permit is likely to be compliant and requires no further inspection, or (2) a permit 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 [3] 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 outlier detection methods, outlyingness alone does not translate to the desired targets, and we further mention the difficulty of detecting outliers in high-dimensional data [4]; with respect to data-quality assurance techniques, we remark that they mostly depend on variable distribution assumptions and concentrate on random errors [5]. 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* approach to 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 crossliers to be a special case of outliers in the sense that they 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.

The paper structure follows: Sec. II states our problem formally; Sec. III discusses past work related to ours; Sec. IV elaborates our approach in detail; Sec. V describes ours experimental setup; Sec. VI refers to our results; Sec. VII discusses our method; and Sec. VIII concludes this work and suggests future research directions.

978-1-7281-8808-9/20/$31.00 ©2020 IEEE

2020 25th International Conference on Pattern Recognition (ICPR) | 978-1-7281-8808-9/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICPR48806.2021.9412644

II. Problem Statement

Given a category-labelled dataset, we define a *crosslier* as a sample of which the category label is swapped and a proportion of its features are 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. Accordingly, either (1) crossliers fall within the cluster of some other category, or (2) crossliers lie across other categories. To illustrate, we present Fig. 1; four different categories $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$, and $\mathcal{D}$ are denoted, with crossliers marked as $\mathcal{A}^*$, $\mathcal{B}^*$, $\mathcal{C}^*$, and $\mathcal{D}^*$.

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 $z_i = z$.

III. Related Work

In this section, we provide a brief overview of three techniques typically used to address anomaly detection problems. Hereinafter the term *anomaly* is used to broadly refer to a data point which, given its observed values an/or domain knowledge, stands out from the dataset. 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: (*A*) supervised and semi-supervised learning techniques; (*B*) unsupervised learning methods; and (*C*) data quality-related procedures. We further disclose their non-applicability to our scenario.

A. Supervised and Semi-supervised Learning

In the presence of labels indicative of previously-recognised non-compliance, the problem can be approached as a supervised learning task. Examples are: detecting insurance fraud [6], exposing deceitful telecommunication users [7], and identifying irregular heart beat patterns [8]. The choice of fitting algorithm is diverse: support vector machines (SVM) [9], neural networks [10], and random forests [11], to name a few. However, a common issue is class imbalance; i.e., the small ratio of positive to negative instances [12]. Some remedies to this issue are instance importance reweighting [13], under and over sampling [14], or a mixture of both [15].

For the case where both labelled and unlabelled instances are available, a semi-supervised learning approach is suitable. This framework can, as an example, make use of clustering algorithms assuming that data points within the same cluster probably share the same label [16]. Other authors focus on addressing sample bias to improve on the selection of inspection targets [17], using unlabelled instances as negative samples. The assumption is that the incidence of inspection targets is negligible within the unlabelled data. Our data does not possess target labels, making these techniques inapplicable.

B. Unsupervised Learning

A straightforward alternative is to find deviating cases through outlier detection techniques using unsupervised methods. The assumption is that the most probable samples to target are the ones that differ the most from all others in their category, i.e., outliers. Outlier detection techniques have been applied to system intrusion detection [18], maritime traffic anomaly flagging [19], and image curation [20], amongst others. Some examples of the algorithms used are one-class classifiers [21], isolation forests [22], nearest-neighbour [23], k-means clustering [24], and local outlier factor [25].

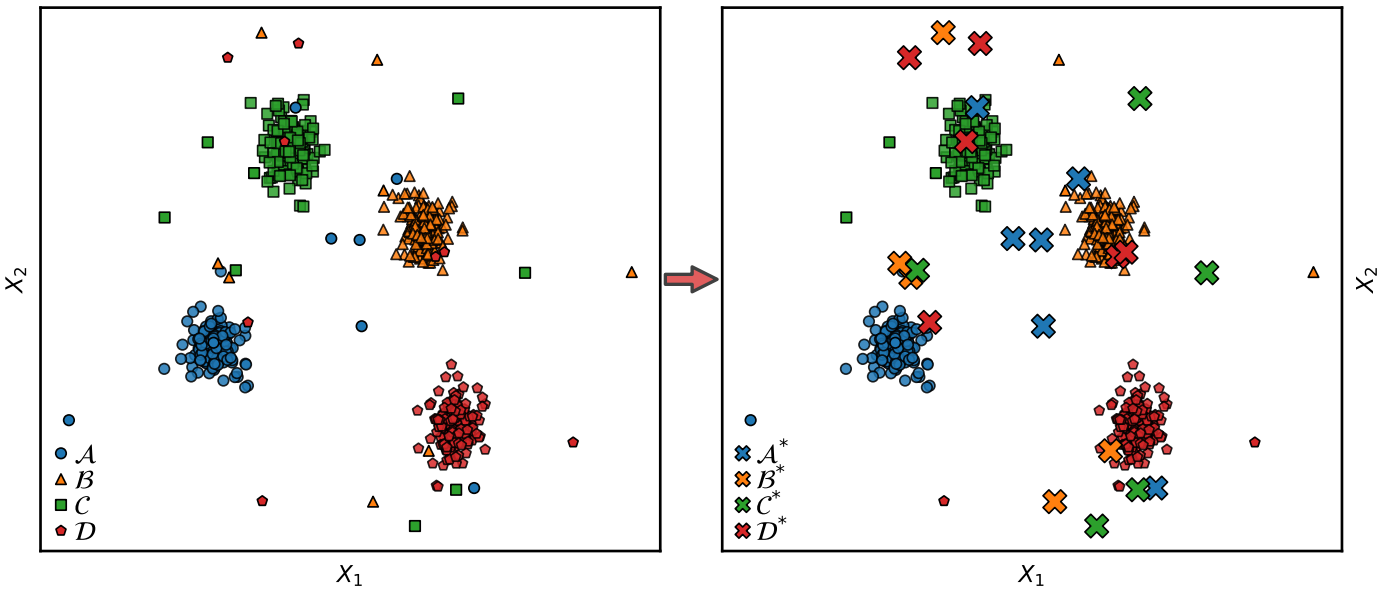

Fig. 1. **Crosslier detection.** Samples with features X_1 and X_2, pertaining to either category $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$, or $\mathcal{D}$ (left). Crossliers are marked as crosses (right).

There are at least three types of problem with most unsupervised methods. The main problem is their dependency on distance metrics (Minkowski measures) to define outlyingness, which makes them sensitive to feature scaling. A second problem arises when dealing with high-dimensional data [26], particularly when attempting to estimate densities empirically [27]. Thirdly, through manipulation of only a proportion of features as assumed in the problem description, target samples may not stand out as outliers.

Yet, outliers are not necessarily crossliers. To illustrate, we present Fig. 2, which builds on the example in Fig. 1 by applying an isolation forest algorithm as per [28]. We see how data points flagged as outliers do not represent the target crossliers; hence, we should take in consideration the distribution of categories when marking instances as crosslying. A second problem with the outlier detection method illustrated is that most flagged instances are arguably not outlying with respect to their clusters. The insensitivity shown makes outlier detection methods precarious to address our problem. Ultimately, not all outliers are crossliers: they do not possess the specific category-related characteristics we seek.

C. 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 anomaly detection [29]. Typical methods involve, for example, assumptions over feature distributions [30] and cross-referencing datasets for dependency-matching or constraint-mining [31], [32].

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 towards 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.

IV. The eXPose Approach

Here, we detail the proposed eXPose for the detection of crossliers. As defined in Sec. II, 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 cross-validation setup. Therefore, these learned functions must be calibrated to make the scores comparable among each other.

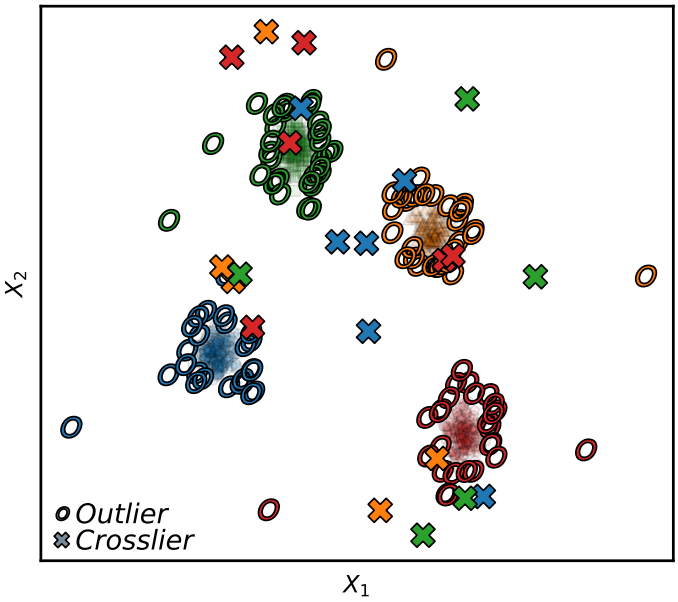

Fig. 2. **Distinction between outlier and crosslier.** Four-category example from Fig. 1. Crossliers are marked as crosses and outliers are denoted as circles. Transparency values for data clusters have been raised for visualisation.

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 cross-validation. The learners collectively yield the overall crosslier score function. We finalize the method section with the crosslier diagram, a tool to visualise crosslier scores and pinpoint suspect samples.

A. Classification Setup

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

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

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 [33]:

$$\arg\min_{f \in \mathcal{F}} \hat{\mathcal{R}}_{\mathcal{D},\mathcal{L},f} \tag{2}$$

where

$$\hat{\mathcal{R}}_{\mathcal{D},\mathcal{L},f} = \frac{1}{n} \cdot \sum_{i=1}^{n} \mathcal{L}(f(x_i), y_i) \tag{3}$$

Depending on the chosen learner, the curse of dimensionality is addressed by incorporating either regularisation, feature selection, or both protocols in the learning task [34]. These protocols also alleviate overfitting and promote classifier robustness by reducing the complexity of the final model [35].

All regularisation parameters given prior to the learning task can be optimally retrieved through hyperparameter optimisation [36], [37]. The learners to be applied within a specific problem can also be optimally selected.

B. 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 Area Under the receiver operating characteristic Curve (AUC) through cross-validation (CV) [38]. 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

$$\bigcap_{k=1}^{K} D_k^{ts} = \emptyset, \ \bigcup_{k=1}^{K} D_k^{ts} = D, \text{ and } D_k^{tr} = D \setminus D_k^{ts} \quad (4)$$

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

$$\bigcap_{\ell=1}^{L} D_\ell^{ts} = \emptyset, \ \bigcup_{\ell=1}^{L} D_\ell^{ts} = D_k^{tr}, \text{ and } D_\ell^{tr} = D_k^{tr} \setminus D_\ell^{ts} \quad (5)$$

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 address this through model calibration.

C. Crosslier Score

To transform the output of uncalibrated models into a calibrated output, Platt scaling [39] 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 [40] of a sample x from category z:

$$f_z(x) = -\log_2 \hat{P}(y|x) \quad (6)$$

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

$$\hat{P}(y|x) = \bigcup_{k=1}^{K} \hat{P}(y_k|x_k) \quad (7)$$

in which,

$$\hat{P}(y_k|x_k) = \frac{1}{L} \cdot \sum_{\ell=1}^{L} [f_\ell^k({}_\phi^*\Psi_k^{tr}(x_k))] \quad (8)$$

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^*

$$f_\ell^k(u) = \frac{1}{1 + e^{-(\alpha^* + \beta^* \cdot u)}} \quad (9)$$

in which

$$\alpha^*, \beta^* = \arg\min_{\alpha,\beta} - \sum_{(x,y) \in D_\ell^{ts}} [\mu \cdot log(p) + (1 - \mu) \cdot log(1 - p)] \quad (10)$$

where

$$\mu = \begin{cases} \dfrac{(\sum_{y \in D_\ell^{ts}} y) + 1}{(\sum_{y \in D_\ell^{ts}} y) + 2}, \text{ if } y = 1 \\ (|D_\ell^{ts}| - (\sum_{y \in D_\ell^{ts}} y) + 2)^{-1}, \text{ otherwise} \end{cases} \quad (11)$$

and

$$p = \frac{1}{1 + e^{-(\alpha + \beta \cdot {}_\phi^*\Psi_\ell^{tr}(x))}} \quad (12)$$

In (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 $x \in D_\ell^{ts}$.

D. Crosslier Diagram

At the basis of the crosslier diagram 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) in the upcoming Sec. VI as part of our experimental results.

V. Experiments

In this section, we describe our experiments. Two setups are considered in which eXPose is (*A*) applied to the waste permit dataset, and (*B*) compared to other anomaly detection methods. Resources described in (*B*) are made available [41].

A. Waste Transportation Setup

1) 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 permit instances, each with a waste category (out of 20 total different waste categories) and 49 variables which were a mixture of both numerical and nominal features.

2) Learners: we experimented with both linear and non-linear learners to find the best performing model for each waste category: An elastic net-regularised logistic regression learner (LR) 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 [42]–[44], 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 [45]. A non-linear gradient boosted tree framework (XGB) was considered [46], with 100 additive trees where 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 [47] in terms of scalability, robustness to noisy samples, and classification performance.

3) Selection and calibration: to select and calibrate the best model, we applied nested-CV in a stratified manner [48] with $K = 10$, $L = 10$ as described in Sec. IV. 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 [49] 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 XGB is relatively insensitive to hyperparameter changes, as shown in the experimental results of [50], we did not perform hyperparameter optimisation for this learner. The best model for each category was used to generate the crosslier scores and crosslier diagrams (Sec. VI).

B. Benchmark Setup

1) Data: 20 binary classification datasets were retrieved from *openML*: an open, organised, and online ecosystem for machine learning [51]. They are real-world datasets from different domains, and can be easily accessed through *openML*'s API. Target classes were treated as the categories $\mathcal{Z}$. Table I 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.

2) 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 I
DATASETS

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

3) Synthetic crossliers: to simulate a real-world scenario, crossliers were generated 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.

4) Methods: eXPose was compared to two well-established anomaly detection methods: local outlier factor (LOF) and isolation forest (IF), mentioned in Sec. III. 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 $\lambda = 1$ [52]; 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.

5) Evaluation: the crosslier scores of eXPose were generated as in Sec. IV; the anomaly scores of the anomaly detection methods were generated category-wise for every method. For each category, crosslier detection performance was measured in average precision (AP) [53], a common measure in outlier detection assessment [54]. 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 posteriorly across initialisation seeds.

VI. 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.

A. Waste Transportation

When applied to the waste transportation data, we show firstly the estimated AUC performances yielded by both candidate models LR (logistic regression learner) and XGB (gradient boosted tree-based learner). 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.

1) Model performance and selection: table II 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. Learner XGB 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.

2) Crosslier diagrams: in Fig. 3 the crosslier diagrams with scores generated by the selected model XGB 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.

TABLE II
Model Performance

Category	LR	XGB
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	$\mathbf{0.779 \pm 0.027}$

3) Inspection domain: the inspectors of ILT were provided with the crosslier diagrams of all waste categories excluding category 4. 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 strenuous spreadsheet analysis.

B. Benchmark

The outcome of our experiments with respect to controlled crosslier detection is to be seen in Fig. 4. We present the results for the three methods: eXPose, local outlier factor (LOF), and isolation forest (IF). Fig. 4 shows (a) the mean performance scores (AP) across 20 datasets, (b) for 81 different configurations of (ρ_x, ρ_y), each dataset-configuration pair with 10 different 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.

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 Sec. IV.

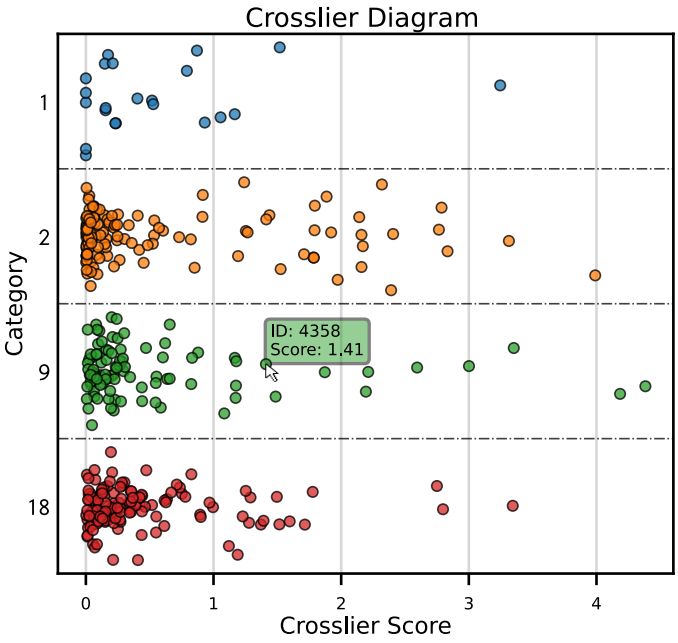

Fig. 3. **Crosslier diagrams of four waste categories.** Hovering over an instance highlights its identifier (4358) and crosslier score (1.41).

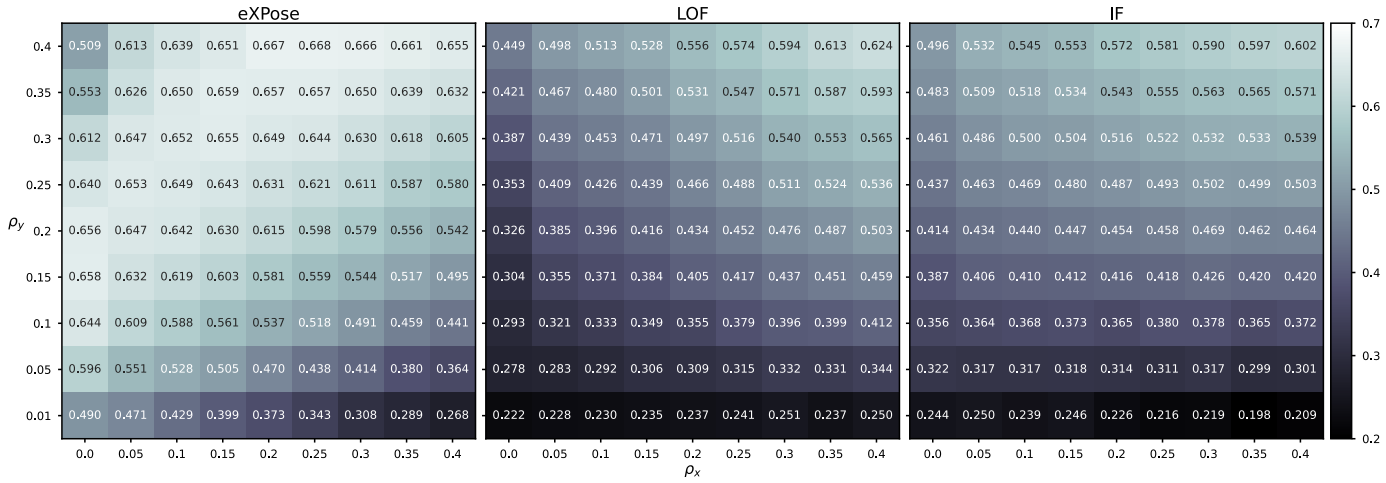

Fig. 4. **Crosslier detection performance across different methods.** Heatmaps depict the AP scores for our method (eXPose), local outlier factor (LOF), and isolation forest (IF). Performance values were averaged across datasets and random initialisations. In the vertical axis, ρ_y is the proportion 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.

VII. Discussion

The eXPose approach is evidently better at detecting crossliers through the exploitation of category models, when compared to standard outlier detection methods. This was expected, as crossliers are defined based on their feature values in a category-wise manner, instead of simply considering any outlying feature value. 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 approach is to be seen as a wrapper over different components: at its core, it is a data-driven category-modelling method using learner functions. Here, score calibration is applied, and even though a selected model might have a low AUC, the generated 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 performance). This is relevant 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 learner and hyperparameters selected.

VIII. Conclusion

In the present work, we (1) defined a specific type of data anomaly, which we term *crosslier*, (2) introduced the eXPose approach to crosslier detection, and (3) designed the *crosslier diagram*, a visualisation tool to represent crossliers evidently. We showed that conventional outlier 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 approach: (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.

In the future, close cooperation with the inspectors is highly recommended. By receiving their feedback on the inspected crosslying permits, our method will be further validated. Moreover, we can use those cases as labelled instances in a supervised learning scenario. A further direction is to apply eXPose to other real-world problems in other domains in order to investigate its general applicability and related outcomes.

Acknowledgment

The authors would like to thank Jasper van Vliet, as well as the entire team of the *Innovatie- en Datalab* within the Human Environment and Transport Inspectorate of the Netherlands Ministry of Infrastructure and Water Management, for their time and resources, which enabled the making of this work.

References

[1] (2020, July) Waste classification. [Online]. Available: https://ec.europa.eu/environment/waste/framework/list.htm.

[2] (2020, July) EU waste legislation. [Online]. Available: https://ec.europa.eu/environment/waste/legislation.

[3] R. Choudhary and H. Gianey, "Comprehensive review on supervised machine learning algorithms," in *Proceedings of the 2017 International Conference on Machine Learning and Data Science*, pp. 37–43.

[4] B. Venkatesh and J. Anuradha, "A review of feature selection and its methods," *Cybernetics and Information Technologies*, vol. 19, no. 1, pp. 3–26, 2019.

[5] J. Liu, J. Li, W. Li, and J. Wu, "Rethinking big data: a review on the data quality and usage issues," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 115, pp. 134–142, 2016.

[6] S. Subudhi and S. Panigrahi, "Use of optimized fuzzy c-means clustering and supervised classifiers for automobile insurance fraud detection," *Journal of King Saud University – Computer and Information Sciences*, 2017. [Online]. Available: https://doi.org/10.1016/j.jksuci.2017.09.010.

[7] R. Li, Y. Zhang, Y. Tuo, and P. Chang, "A novel method for detecting telecom fraud user," in *Proceedings of the 2018 International Conference on Information Systems Engineering*, pp. 46–50.

[8] M. Vollmer, P. Sodmann, L. Caanitz, N. Nath, and L. Kaderali, "Can supervised learning be used to classify cardiac rhythms?," in *2017 Computing in Cardiology*, pp. 1–4.

[9] A. George, "Anomaly detection based on machine learning: dimensionality reduction using PCA and classification using SVM," *International Journal of Computer Applications*, vol. 47, no. 21, pp. 5–8, 2012.

[10] J. Mulongo et al., "Anomaly detection in power generation plants using machine learning and neural networks," *Applied Aritifical Intelligence*, vol. 34, no. 1, pp 64–79, 2020.

[11] H. Alazzam, A. Alsmady, and A. Shorman, "Supervised detection of IoT botnet attacks," in *Proceedings of the 2019 International Conference on Data Science, E-Learning and Information Systems*, pp. 1–6.

[12] C. Veenman, "Data base investigation as a ranking problem," in *Proceedings of the 2012 European Intelligence and Security Informatics Conference*, pp. 225–231.

[13] N. García-Pedrajas and C. García-Osorio, "Boosting for class-imbalanced datasets using genetically evolved supervised non-linear projections," *Progress in Artificial Intelligence*, vol. 2, no. 1, pp. 29–44, 2013.

[14] R. Barandela, R. Valdovinos, J. Sánchez, and F. Ferri, "The imbalanced training sample problem: under or over sampling?," in *Proceedings of the Joint International Association for Pattern Recognition International Workshops, SSPR 2004 and SPR 2004*, pp. 806–814.

[15] G. Nguyen, A. Bouzerdoum, and S. Phung, "A supervised learning approach for imbalanced data sets," in *Proceedings of the 2008 International Conference on Pattern Recognition*, pp. 1–4.

[16] G. Xiang and W. Min, "Applying semi-supervised cluster algorithm for anomaly detection," in *Proceedings of the 2010 International Symposium on Information Processing*, pp. 43–45.

[17] G. Jacobusse and C. Veenman, "On selection bias with imbalanced classes," in *DS 2016: Discovery Science*, pp. 325–340, 2016.

[18] S. Zanero and S. Savaresi, "Unsupervised learning techniques for an intrusion detection system," in *Proceedings of the 2004 ACM Symposium on Applied Computing*, pp. 412–419.

[19] M. Vespe, I. Visentini, K. Bryan, and P. Braca, "Unsupervised learning of maritime traffic patterns for anomaly detection," in *Proceedings of the 2012 IET Data Fusion & Target Tracking Conference*, pp. 1–5.

[20] W. Liu, G. Hua, and J. Smith, "Unsupervised one-class learning for automatic outlier removal," in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3826–3833.

[21] J. Janssens, I. Flesch, and E. Postma, "Outlier detection with one-class classifiers from ML and KDD," in *Proceedings of the 2009 International Conference on Machine Learning and Applications*, pp. 147–153.

[22] F. Liu, K. Ting, and Z. Zhou, "Isolation-based anomaly detection," *ACM Transactions on Knowledge Discovery from Data*, vol. 6, no. 1, art. 3, 2012.

[23] A. Mennatallah and M. Goldstein, "Nearest-neighbor and clustering based anomaly detection algorithms for rapidminer." in *Proceedings of the 2012 RapidMiner Community Meeting and Conference*, pp. 1–12.

[24] A. Muniyandi, R. Rajeswari, and R. Rajaram, "Network anomaly detection by cascading k-means clustering and C4.5 decision tree algorithm," *Procedia Engineering*, vol. 30, pp. 174–182, 2012.

[25] M. Breunig, H. Kriegel, R. Ng, and J. Sander, 2000. "LOF: identifying density-based local outliers," in *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, pp. 93-–104.

[26] H. Liu, X. Li, J. Li, and S. Zhang, "Efficient outlier detection for high-dimensional data," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 12, pp. 2451–2461, 2018.

[27] J. Santos et al., "Fine scale genomic signals of admixture and alien introgression among asian rice landraces," *Genome Biology and Evolution*, vol. 11, no. 5, pp. 1358–1373, 2019.

[28] F. Liu, K. Ting, and Z. Zhou, "Isolation forest," *Proceedings of the 2008 IEEE International Conference on Data Mining*, pp. 413–422.

[29] S. Bonner et al., "Data quality assessment and anomaly detection via map/reduce and linked data: a case study in the medical domain," in *Proceedings of the 2015 IEEE International Conference on Big Data*, pp. 737–746.

[30] C. Pit-Claudel, Z. Mariet, R. Harding, and S. Madden, "Outlier detection in heterogeneous datasets using automatic tuple expansion," 2016. [Online]. Available: https://dspace.mit.edu/bitstream/handle/1721.1/101150/MIT-CSAIL-TR-2016-002.pdf

[31] T. Rekatsinas, X. Chu, I. Ilyas, and C. Re, "HoloClean: holistic data repairs with probabilistic inference," in *Proceedings of the 2017 Very Large Data Base Endowment*, vol. 10, no. 11, pp. 1190–1201.

[32] X. Chu, I. Ilyas, and P. Papotti, "Discovering denial constraints," in *Proceedings of the 2013 VLDB Endowment*, vol. 6, no. 13, pp. 1498–1509.

[33] V. Vapnik, *The Nature of Statistical Learning Theory*. Springer, 2000.

[34] N.Sharma, P. Verlekar, R. Ashary, and S. Zhiquan, "Regularization and feature selection for large dimensional data," 2017. [Online]. Available: https://arxiv.org/pdf/1503.03305.pdf

[35] A. Gupta, K. Gusain and B. Popli, "Verifying the value and veracity of extreme gradient boosted decision trees on a variety of datasets," in *Proceedings of the 2016 International Conference on Industrial and Information Systems (ICIIS)*, pp. 457–462.

[36] M. Claesen and B. Moor, "Hyperparameter search in machine learning," 2015. [Online]. Available: https://arxiv.org/abs/1502.02127

[37] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *Journal of Machine Learning Research*, vol. 13, no.10, pp. 281–305, 2012.

[38] P. Flach, "ROC analysis," in *Encyclopedia of Machine Learning and Data Mining*, pp. 869–875, 2010.

[39] J. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," *Advances in Large Margin Classifiers*, vol. 10, no.3, pp. 61–74, 1999.

[40] D. Jones, in *Elementary Information Theory*, Clarendon Press, Oxford, pp. 11–15, 1979.

[41] A. Barata, "Crosslier detection," in *GitHub Repository*. [Online]. Available: https://github.com /pereirabarataap/crosslier-detection.

[42] D. Rosaio, "Highly effective logistic regression model for signal (anomaly) detection," in *Proceedings of the 2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. V–817.

[43] Y. Wang, "A multinomial logistic regression modeling approach for anomaly intrusion detection," *Computers & Secutiry*, vol. 24, no. 8, pp. 662–674, 2005.

[44] M. Mok, S. Sohn, and Y. Ju, "Random effects logistic regression model for anomaly detection," *Experts Systems with Applications*, vol. 37, no. 10, pp 7162–7166, 2010.

[45] D. Kleinbaum and M. Klein, "Modeling Strategy Guidelines," in *Logistic Regression*, pp. 165–202, 2010.

[46] J. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001.

[47] S. Pafka, "Benchm-ml", in *GitHub Repository*. [Online]. Available: https://github.com/szilard/benchm-ml.

[48] M. Stone, "Cross-validatory choice and assessment of statistical predictions," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 36, no. 2, pp. 111–147, 1974.

[49] S. Chan and P. Treleaven, "Continuous model selection for large-scale recommender systems," in *Handbook of Statistics*, vol. 33, pp. 107–124, 2015.

[50] Y. Xia, C. Liu, Y. Li, and N. Liu, "A boosted decision tree approach using bayesian hyper-parameter optimization for credit scoring," *Expert Systems with Applications*, vol. 78, no. 1, pp. 225–241, 2017.

[51] J. Vanschoren, J. Rijn, B. Bischl, and L. Torgo, "OpenML: networked science in machine learning;" *SIGKDD Explorations*, vol. 15, no. 2, pp 49–60, 2013.

[52] T. Chen, and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the 2016 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794.

[53] E. Zhang and Y. Zhang, "Average Precision," in *Encyclopedia of Database Systems*, Springer, 2009.

[54] X. Xu, H. Liu, L. Li, and M. Yao, "A comparison of outlier detection techniques for high-dimensional data," *International Journal of Computational Intelligence Systems*, vol. 11, pp. 652–662, 2018.