A Nearest Neighbor Open-set Classifier based on Excesses of Distance Ratios

Matthys Lucas Steyn[1,2],*
Tertius de Wet[1],
Bernard De Baets[2]
and
Stijn Luca[2]
Department of Statistics and Actuarial Science,
Stellenbosch University[1]
Department of Data Analysis and Mathematical Modelling,
Ghent University[2]

June 24, 2022

Abstract

This paper proposes an open-set recognition model that is based on the use of extreme value statistics. For this purpose, a distance ratio is introduced that expresses how dissimilar a target point is from the known classes by considering the ratio of distances locally around the target point. It is shown that the class of generalized Pareto distributions with bounded support can be used to model the peaks of the distance ratio above a high threshold. The resulting distribution provides a probabilistic framework to perform open-set recognition. Furthermore, we describe a numerical procedure to estimate the hyperparameters of our model. This procedure is based on a new objective function that considers both the fit of the generalized Pareto distribution and the misclassification error of the known classes. Our method is applied to three image data sets and an audio data set showing that it outperforms similar open-set recognition and anomaly detection methods.

Keywords: Classification, Extreme Value Theory, Generalized Pareto Distribution, Open-set Recognition

*This research received funding from the Flemish Government under the "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen" programme.

1 Introduction

Supervised classification is the process of assigning a sample to one of two or more known categories. This approach assumes that each sample $\boldsymbol{x} \in \boldsymbol{X}$ belongs to one of the classes denoted by $y_l \in \mathcal{Y}, l = 1, ..., L$, where the training data are labeled—each instance $\boldsymbol{x}_i$ is associated with a class label y_i. Traditional classifiers are constructed under a closed-set assumption. In other words, it is assumed that a predefined number of classes is present in the data. However, the available information of $\mathcal{Y}$ may be incomplete during training. Therefore, any sample from an unseen category will be misclassified as one of the known categories. Open-set recognition (OSR) methods generalize classification methods to detect these new classes during prediction on the test set whilst maintaining a good classification performance on the data from known classes (Scheirer et al., 2013).

The assumption of OSR is that additional classes that were unknown during training arise during prediction on the test set. Broadly, an OSR model must perform two tasks: the model must classify a sample from any of the known classes into the correct class and detect whether it is an observation from an unknown class. It is assumed that no information about the unknown classes is available during training. Recently, models based on extreme value theory (see Bendale and Boult, 2016; Jain et al., 2014; Scheirer et al., 2014) have shown to be useful at performing probabilistic OSR. This class of models adjusts the posterior class probabilities to allow the model to detect samples that do not belong to any of the known classes. Generally, a two-stage approach is used. Initially, a supervised classifier is trained using the available data. Extreme value theory is then used to adjust the posterior class probabilities such that samples that are extreme in some sense with respect to the training data are classified as 'unknown' (Scheirer et al., 2011).

In this paper we propose an OSR model based on the extreme values of a distance ratio. Using the open-set nearest neighbor (OSNN) classifier of Júnior et al. (2017) as a point of departure, a distance ratio based on the k closest points locally around a target point is derived. The proposed distance ratio expresses the dissimilarity of a target point with respect to the reference data. It is shown that the generalized Pareto distribution can be used to approximate the distribution of the distance ratio. This distribution is then used to

detect samples from unseen categories. Sections 2.1 and 2.2 give a brief overview of open-set recognition and extreme value theory, respectively. In Section 3, the proposed distance ratio is derived, and it is shown that the upper tail of the distance ratio can be approximated with the generalized Pareto distribution. We propose a numerical method to estimate the parameters of the generalized Pareto distribution with maximum likelihood estimation. Furthermore, a new approach to select the optimal number of nearest neighbors of the distance ratio and the optimal threshold of the generalized Pareto distribution is provided. We apply our method to four data sets in Section 4 and show that it outperforms similar OSR methods.

2 Related Work

Open-set recognition shares characteristics with other methods such as novelty detection, out-of-distribution detection and zero-shot learning (Geng et al., 2018). Models for novelty detection commonly use a one-class approach and are trained using the data considered as 'normal'. Samples are then classified as 'normal' or 'anomalous' during prediction on the test set (Pimentel et al., 2014). Although one-class classifiers can be used to detect unknown categories, the classification accuracy of the model with respect to the known classes will be inferior to that of supervised classification models (Jain et al., 2014). Out-of-distribution detection is the process of detecting distributional shifts in the data. The distributional shift may indicate the existence of unknown classes or that the distribution of the known classes changed (Yang et al., 2021). Zero-shot learning include methods that learn a semantic embedding using the known classes which is then transferred to the unknown classes during testing. This process assumes that some semantic information about the unknown classes is available during training (Geng et al., 2018).

2.1 Open-set Recognition

The main assumption of OSR models is that no information about the unknown classes is available during training (Scheirer et al., 2014). Therefore, data from the known classes

are used to define regions in the data where new classes are expected to be located, *i.e.*, regions with large open-space risk. Extreme value theory provides a method to determine regions in a feature space where unknown classes are expected to be located by using only the data of the known classes (Scheirer et al., 2011). The resulting distribution is used to classify samples as known or unknown with respect to the available classes.

Discriminative approaches for OSR assume that there exists a feature space where the samples from new classes are linearly separable from that of the known classes. The 1-vs-set machine of Scheirer et al. (2013) extends the linear support vector machine algorithm by defining two hyperplanes parallel to the original decision boundary such that observations from known categories are distributed within the slab of the two hyperplanes. This model was extended by Scheirer et al. (2014) to include non-linear kernels. Similarly, Bendale and Boult (2016) proposed the OpenMax model which discriminates between known and unknown classes by using the features of the penultimate layer of a neural network. For each class, the distances between the correctly classified training data and the corresponding class mean vector are used to fit an extreme value distribution. These distributions are then used to adjust the SoftMax layer for OSR.

Scheirer et al. (2013) provided a theoretical definition of open space $\mathcal{O}$ and open-set risk. Open space $\mathcal{O}$ is the measure space that is *far* from the known data. Performing classification in $\mathcal{O}$ incurs the open-set risk of classifying a sample from an unknown class as one of the known classes (Geng et al., 2018). Furthermore, Scheirer et al. (2014) defined a compact-abating probability model—the probability of class membership decreases as a sample moves away from the data of the known classes. Using the compact-abating probability model framework, Scheirer et al. (2014) showed that thresholding the model to detect unknown classes limits open-set risk. Specifically, if all the posterior class probabilities of the compact-abating probability model are smaller than the threshold, the target is regarded as a sample from an unknown class.

Hence, under this framework, probabilistic OSR consists of formulating a compact-abating probability model and selecting a threshold for discriminating between known and unknown classes. Note that conventional, numerical approaches cannot be used to estimate the

threshold because there are no data of the unknown classes. Therefore, a common approach is to use extreme value theory to determine the extreme boundaries of the data space of the known classes (Scheirer et al., 2011).

Methods such as the PI-SVM of Jain et al. (2014), the OpenMax of Bendale and Boult (2016) and the extreme value machine of Rudd et al. (2018) use the Fisher–Tippet theorem of extreme value theory to estimate the distribution of the maximum or minimum of a distance measure. Furthermore, the `libmr` module is used to estimate the parameters of the extreme value distribution. These approaches differ from our proposed method as follows. The PI-SVM and OpenMax methods fit a distribution for each class and the extreme value machine fits a distribution for each sample. The models proposed by Vignotto and Engelke (2018) also fit a distribution for each sample. However, both the generalized extreme value and generalized Pareto distributions are used. Our proposed method fits a single generalized Pareto distribution, irrespective of the sample size and number of known classes. Moreover, by definition, our distance ratio is bounded, which ensures that the shape parameter of the generalized Pareto distribution is negative.

Alternatively, some methods consist of estimating a score that measures the dissimilarity between a sample and the data of the known classes. A sample with a dissimilarity score above a selected threshold is regarded as a sample from an unknown class. The OSNN method of Júnior et al. (2017) estimates a dissimilarity score as the ratio of the distances of the two nearest neighbors around a target that are from different classes. Firstly, the OSNN method classifies a target as unknown if the first two nearest neighbors are from different classes. If the two nearest neighbors are from the same class, the distance ratio between the nearest neighbor and the nearest neighbor from another class is computed. The target is classified as unknown if the distance ratio exceeds a high threshold. As Júnior et al. (2017) noted, the distance ratio is located in $[0, 1]$. Using this result, they propose a numerical procedure to select the optimal threshold, T, by using only the data of the known classes. This is achieved by randomly selecting half of the classes as 'known' and treating the other classes as 'unknown'. The training data of the known classes are used as the reference data, and the distance ratio is computed for the known and unknown

classes in the validation data set. The optimal threshold is selected as the $T \in [0.5, 1]$ for which the accuracy of the validation data is a maximum.

2.2 Extreme Value Theory

In this section, we briefly review some of the main concepts of extreme value theory—a field of statistics concerned with estimating the tails of a distribution. The theory of extremes is applied in numerous fields, such as financial risk management (McNeil et al., 2015) and road safety estimation (Zheng et al., 2014). More recently, extreme value theory has shown to be useful for OSR (Geng et al., 2018), anomaly detection (Talagala et al., 2020a,b), and to evaluate the robustness of deep learning models (Weng et al., 2018).

An effective approach to model the extremes of a distribution is the peaks-over-threshold approach. Consider a sample of independent and identically distributed random variables denoted by $\{X_1, ..., X_n\}$ and define the conditional excesses above a sufficiently high threshold t by $S = X - t \mid X > t$. Using the Pickands–Balkema–de Haan theorem (McNeil et al., 2015), the distribution of S can be approximated with the generalized Pareto distribution, denoted by

$$P(S > s) \approx \bar{G}_{\tau,\gamma}(s) \equiv \left(1 - \frac{s}{\tau}\right)^{-\frac{1}{\gamma}}, 1 - \frac{s}{\tau} > 0, \gamma < 0, \tau > 0. \tag{1}$$

Note that in (1) we only consider the $\gamma < 0$ case. Furthermore, we parameterize the distribution in terms of γ and $\tau = -\sigma/\gamma$. The extreme value index, γ, determines the rate at which the tail of a distribution decays. Specifically, the case $\gamma < 0$ is known as the Weibull-type generalized Pareto distribution, and implies that the underlying random variable has a finite upper bound (see Beirlant et al., 2004, chap. 2). As will be shown in Section 3.2, our proposed method uses this type of generalized Pareto distribution to perform OSR.

The parameters of the generalized Pareto distribution are usually estimated via an iterative maximum likelihood procedure (Beirlant et al., 2004; del Castillo and Serra, 2015). Although not analytically tractable, del Castillo and Serra (2015) showed that, if $-1 < \gamma < 0$, the log-likelihood of the generalized Pareto distribution has a global

maximum. Furthermore, since it must hold that $1-\frac{s}{\tau} > 0$, it follows that $\tau > \max_{j=1,...,n_e} s_j$, where $\{s_j \mid j = 1, ..., n_e\}$ is a sample of n_e excesses above the threshold t.

The Pickands–Balkema–de Haan theorem enables one to model the upper tail of a random variable without any knowledge of the underlying form of F. If the conditions of the Pickands–Balkema–de Haan theorem are satisfied, then $\bar{F}(x)$ can be approximated as $\bar{F}(t) \cdot \bar{G}_{\tau,\gamma}(s)$ for large x. Hence, the upper tail of F can be parameterized through the limiting result.

Note that the peaks-over-threshold method assumes that the threshold is close to the limiting maximum of X. If the threshold is too low, the generalized Pareto distribution is not a good approximation for the excesses above the threshold leading to biased parameter estimates. Conversely, if the threshold is close to the maximum of X, then a small sample is used to estimate the parameters of the generalized Pareto distribution, leading to a large variance. Although this tuning parameter controls the goodness-of-fit of the model, there is no widely-adopted general threshold selection procedure (Beirlant et al., 2004). Graphical methods to select the threshold such as Hill-type plots have been proposed (see Davison and Smith, 1990; Beirlant et al., 2004). Alternatively, some methods use asymptotic theory to approximate the optimal t so that the mean squared error of the extreme value index is a minimum (Beirlant et al., 2005). In Section 3.3 we provide a numerical approach to select the optimal threshold based on the quantile-quantile plot of the generalized Pareto distribution.

3 Methodology

In this section, we outline a new, probabilistic method to perform OSR. Generalizing the OSNN method proposed by Júnior et al. (2017), a distance ratio is defined which measures the dissimilarity between a target point and a reference data set. We demonstrate how to estimate the distribution of the distance ratio in regions of extreme dissimilarity with the generalized Pareto distribution. This distribution provides a probabilistic approach to detect whether a target point is significantly dissimilar from a reference data set.

3.1 Distance Ratio

Consider a sample of coordinates $\{(\boldsymbol{x}_i, y_i) \mid i = 1, 2, ..., n\}$ where each $\boldsymbol{x}_i$ is a sample from a random vector $\boldsymbol{X} \in \mathbb{R}^p$ and $y_i \in \mathcal{Y} = \{1, 2, ..., L\}$ is the corresponding class label. Let $\boldsymbol{x}^\star$ be a vector of interest referred to as the target point. For a choice of k, denote the set of k closest points around $\boldsymbol{x}^\star$ by $\mathcal{N}(\boldsymbol{x}^\star) = \{(\boldsymbol{x}_{(u)},\ y_{(u)}) \mid u = 1, ..., k\}$ with corresponding Euclidean distances $\{d_{(1)}(\boldsymbol{x}^\star) \leq ... \leq d_{(k)}(\boldsymbol{x}^\star)\}$. Denote the average Euclidean distance between $\boldsymbol{x}^\star$ and the set of points in $\mathcal{N}(\boldsymbol{x}^\star)$ by $\bar{d}(\boldsymbol{x}^\star)$, where

$$\bar{d}(\boldsymbol{x}^\star) = \frac{1}{k} \sum_{u=1}^{k} d_{(u)}(\boldsymbol{x}^\star). \tag{2}$$

The neighborhood $\mathcal{N}(\boldsymbol{x}^\star)$ is used to find the majority vote of the class labels—denoted by $y^\star = \arg\max_{y \in \mathcal{Y}} \sum_{u=1}^{k} \mathbb{I}(y_{(u)} = y)$. If the majority vote is not unique, $y^\star$ is assigned the class label of the closest modal class to $\boldsymbol{x}^\star$. Next, the k closest points around $\boldsymbol{x}^\star$ that are not from class $y^\star$ are identified. Denote the average distance between these points and $\boldsymbol{x}^\star$ by

$$\bar{d}^c(\boldsymbol{x}^\star) = \frac{1}{k} \sum_{u=1}^{k} d^c_{(u)}(\boldsymbol{x}^\star). \tag{3}$$

The distance ratio is then computed as

$$r(\boldsymbol{x}^\star) = \frac{\bar{d}(\boldsymbol{x}^\star)}{\bar{d}^c(\boldsymbol{x}^\star)}. \tag{4}$$

The distance ratio in (4) provides a measure of dissimilarity between $\boldsymbol{x}^\star$ and a reference data set, $\{(\boldsymbol{x}_i, y_i) \mid i = 1, 2, ..., n\}$. If $\boldsymbol{x}^\star$ is close to a known class, then $\bar{d}(\boldsymbol{x}^\star)$ is small relative to $\bar{d}^c(\boldsymbol{x}^\star)$, and the distance ratio is close to zero. As $\boldsymbol{x}^\star$ moves away from the data of the known classes, both $\bar{d}(\boldsymbol{x}^\star)$ and $\bar{d}^c(\boldsymbol{x}^\star)$ increase such that $r(\boldsymbol{x}^\star)$ increases. Hence, assuming that data from any unknown class are far away from those of the known classes, a target point $\boldsymbol{x}^\star$ should be classified as 'unknown' if $r(\boldsymbol{x}^\star)$ is above a high threshold. Moreover, if $k = 1$, then the distance ratio reduces to that of the OSNN method in Júnior et al. (2017).

We use the generalized Pareto distribution discussed in Section 2.2 to estimate the distribution of the upper tail of the distance ratio. Let $r(\boldsymbol{x}_i)$, $i = 1, 2, ..., n$, denote the

distance ratios that are computed from a training data set of size n, where $r(\boldsymbol{x}_i), i = 1, ..., n$, are samples from a random variable R. The excesses of R above a large threshold t, denoted by $S = R - t \mid R > t$, can be approximated with the generalized Pareto distribution in (1). Hence, the probability $P(R > r)$ is approximated with

$$P(R > r) = P(R > t) \cdot P(S > r - t) \approx P(R > t) \cdot \bar{G}_{\tau,\gamma}(r - t). \tag{5}$$

The probability $P(R > t)$ is estimated as the proportion of sample excesses above the threshold t, and the excess probability $\bar{G}_{\tau,\gamma}(r - t)$ can be computed after estimating γ and τ. If $P(R > r)$ in (5) is small, then it indicates that r is in the tail of the distribution of R. Therefore, a sample is classified as 'unknown' if $P(R > r) < \alpha$, where α is a tuning parameter. Hence, the following decision rule is used to classify the vector $\boldsymbol{x}^\star$ as either one of the known classes or as 'unknown'.

$$\hat{y} = \begin{cases} y^\star = \arg\max_{y \in \mathcal{Y}} \sum_{y_u \in \mathcal{N}(\boldsymbol{x}^\star)} \mathbb{I}(y_u = y) & , \text{ if } P(R > r(\boldsymbol{x}^\star)) \geq \alpha \\ \text{'unknown'} & , \text{ otherwise.} \end{cases} \tag{6}$$

The parameter α can be interpreted as the probability that a sample from a known class is classified as unknown. Note that $P(R > r)$ is estimated from the data of the known classes. Using the decision rule in (6) implies that approximately $100 \cdot \alpha\%$ of the data of the known classes will be classified as unknown. There is a payoff between the classification performance of the known classes and the ability to detect unknown classes. If the unknown classes are perfectly separable from the known classes, then α should be close to zero. However, if the unknown classes overlap with the known classes, α should be increased to restrict the data space of the known classes.

3.2 Parameter Estimation

Before $P(R > r)$ can be approximated, the parameters in the right-hand side of (5) must be estimated. In this section, we argue that the distance ratio is bounded between 0 and 1. This result is then used to estimate γ and τ where we restrict $\gamma < 0$.

Consider the denominator in (4). By the definition of $r(\boldsymbol{x}^\star)$, the denominator is computed by removing the samples in $\mathcal{N}(\boldsymbol{x}^\star)$ where $y_u = y^\star$ and replacing them with samples farther away from $\boldsymbol{x}^\star$. The average distance between $\boldsymbol{x}^\star$ and this modified set must be larger than the average distance between $\boldsymbol{x}^\star$ and its k closest points. Consequently, it must hold that $r(\boldsymbol{x}^\star) \leq 1$. Moreover, the Euclidean distance between two vectors is non-negative which implies that the average of k distances is always non-negative. Hence, the distance ratio in (4) is bounded between 0 and 1.

Since the underlying random variable R is bounded, it guarantees that R is in the domain of attraction of the Weibull type extreme value distributions. For a sample of n_e excesses, denoted by $\{s_j \mid j = 1, ..., n_e\}$, the parameters of the generalized Pareto distribution are estimated by maximizing the log-likelihood, *i.e.*,

$$(\hat{\gamma}, \hat{\tau}) = \underset{-1<\gamma<0;\tau>\max_{j=1,\dots,n_e}(s_j)}{\arg\max} \left[-n_e \log(-\gamma \cdot \tau) - \frac{1+\gamma}{\gamma} \cdot \sum_{j=1}^{n_e} \log(1 - {}^{s_j}/_{\tau}) \right]. \tag{7}$$

Recall that if $-1 < \gamma < 0$, then the log-likelihood of the generalized Pareto distribution has a global maximum. Furthermore, it is required that $\tau > \max_{j=1,\dots,n_e}(s_j)$ to ensure that $\log(1 - {}^{s_j}/_{\tau})$ exists. The Newton–Raphson method is commonly used to solve the optimization in (7). Setting the partial derivatives of the log-likelihood equal to zero leads to the following system of equations:

$$\hat{\gamma} = \frac{1}{n_e} \sum_{j=1}^{n_e} \log(1 - {}^{s_j}/_{\hat{\tau}}) \tag{8}$$

$$0 = W_{\hat{\tau},\hat{\gamma}} = \left(\frac{1}{(-\hat{\gamma})} - 1 \right) \sum_{j=1}^{n_e} \frac{s_j}{\hat{\tau} - s_j} - n_e. \tag{9}$$

Note that only an initial value for $\hat{\tau}$, denoted by $\hat{\tau}_0$, is required. An initial value for $\hat{\gamma}$ is computed from (8) using $\hat{\tau} = \hat{\tau}_0$. The parameters are sequentially updated with

$$\hat{\tau}_{i+1} = \hat{\tau}_i - \frac{W_{\hat{\tau}_i,\hat{\gamma}_i}}{\frac{\partial}{\partial\tau} W_{\hat{\tau}_i,\hat{\gamma}_i}}, \text{ and } \hat{\gamma}_{i+1} = \frac{1}{n_e} \sum_{j=1}^{n_e} \log(1 - {}^{s_j}/_{\hat{\tau}_{i+1}}),$$

where $\frac{\partial}{\partial\tau} W_{\tau,\gamma}$ is the partial derivative of $W_{\tau,\gamma}$ with respect to τ.

In the case $\gamma \leq -1$, the maximum likelihood estimates do not exist. However, in most real-world applications, as discussed in del Castillo and Serra (2015), it is reasonable to assume that $\gamma > -1$. Since the likelihood is globally maximized, the starting value for τ only affects the number of iterations required for convergence. Furthermore, the distance ratio is bounded at unity which implies that $\max_{j=1,...,n_e}(s_j) < 1 - t$. Therefore, the initial value for τ can be specified as $\hat{\tau}_0 = 1 - t$, where t is the threshold used in the peaks-over-threshold method.

Using a training data set denoted by $\{(\boldsymbol{x}_i, y_i) \mid i = 1, ..., n_{train}\}$, a leave-one-out approach is used to compute $r(\boldsymbol{x}_i), i = 1, ..., n_{train}$—the distance ratio of $\boldsymbol{x}_i$ with respect to $\{(\boldsymbol{x}_j, y_j) \mid j = 1, ..., n_{train}, j \neq i\}$. For a choice of t, the sample excesses, denoted by $\{s_i \mid i = 1, ..., n_e\}$ where $S = R - t \mid R > t$, are computed and the parameters of the generalized Pareto distribution are estimated by maximizing the log-likelihood.

3.3 Selection of Hyperparameters

In this section, we propose a numerical approach to select k and t via cross-validation by considering both the goodness-of-fit of the generalized Pareto distribution and the increase in the misclassification error of the known classes after adjusting the closed-set predictions with the OSR model. After selecting k and t, the parameters of the generalized Pareto distribution are estimated from the training data.

Assume k and t are given, and that the parameters of the generalized Pareto distribution for the random variable $S = R - t \mid R > t$ are known. For the validation data set $\{(\boldsymbol{x}_i, y_i) \mid i = 1, ..., n_{val}\}$, the distance ratios are computed from (4) using the training data to search for the k-nearest neighbors. Note that an initial, closed-set prediction of $\boldsymbol{x}_i$ is $y_i^\star$, which is the usual prediction of the k-nearest neighbors classifier. Each $y_i^\star$ is then updated to $\hat{y}_i$ using (6).

Furthermore, assume that n_e of the distance ratios of the validation set exceed the threshold, and denote this set by $\{s_j \mid j = 1, ..., n_e\}$. Both the sets $\{\hat{y}_i \mid i = 1, ..., n_{val}\}$ and $\{s_j \mid j = 1, ..., n_e\}$ are used to validate the model. The former set is used to validate the classification performance of the OSR model and the latter is used to measure the

goodness-of-fit of the generalized Pareto distribution. Since no data from the unknown classes are available, it is not possible to measure the ability of the model to detect unknown classes. However, it is possible to measure the classification accuracy of the known classes after adjusting $y^\star$ to $\hat{y}$. To measure the error of classifying a known sample as unknown, the loss function in (10) below is used:

$$E(k,t) = \frac{1}{n_{val}} \sum_{i=1}^{n_{val}} \mathbb{I}(y_i^\star = y_i \cap \hat{y}_i \neq y_i). \tag{10}$$

Moreover, it must be determined whether the estimated generalized Pareto distribution is a good approximation for the set of excesses, $\{s_j \mid j = 1, ..., n_e\}$. This criterion is measured via the quantile-quantile plot of $\{s_j \mid j = 1, ..., n_e\}$ against the quantiles of the generalized Pareto distribution. For a sample of n_e excesses, the quantiles of the generalized Pareto distribution at probability $\frac{j}{n_e+1}$, for $j = 1, 2, ..., n_e$, can be estimated with

$$\hat{Q}\left(\frac{j}{n_e+1}\right) = \hat{\tau} \cdot \left(1 - \left(1 - \frac{j}{n_e+1}\right)^{-\hat{\gamma}}\right) \tag{11}$$

If, for a chosen t, the generalized Pareto distribution provides a good approximation for the excesses $\{s_j \mid j = 1, ..., n_e\}$, then the quantile-quantile plot between the quantiles obtained from (11) and the observed excesses should approximately be a straight line. Let the order statistics of the excesses be denoted by $s_{(j)}, j = 1, ..., n_e$, where $s_{(1)} < .... < s_{(n_e)}$. We define a measure for the goodness-of-fit of the generalized Pareto distribution on $\{s_j \mid j = 1, ..., n_e\}$ as the correlation between the coordinates $(s_{(j)}, \hat{Q}(\frac{j}{n_e+1})), j = 1, ..., n_e$ which is given in (12) below:

$$C(k,t) = \frac{\sum_{j=1}^{n_e} s_{(j)} \cdot \hat{Q}\left(\frac{j}{n_e+1}\right) - n_e^{-1} \sum_{j=1}^{n_e} s_{(j)} \sum_{j=1}^{n_e} \hat{Q}\left(\frac{j}{n_e+1}\right)}{n_e \cdot \hat{\sigma}_S \cdot \hat{\sigma}_{\hat{Q}}}. \tag{12}$$

In (12), the statistics $\hat{\sigma}_S$ and $\hat{\sigma}_{\hat{Q}}$ are the sample standard deviations of $\{s_{(j)},\ j = 1, ..., n_e\}$ and $\{\hat{Q}(\frac{j}{n_e+1}),\ j = 1, ..., n_e\}$, respectively. The objective function that we consider to select k and t combines the objective functions in (10) and (12), and is defined in (13) below:

$$O(k,t \mid \lambda) = (1-\lambda)C(k,t) - \lambda E(k,t). \tag{13}$$

Algorithm 1 Cross-validation to select optimal k and t.

Require: Set of thresholds $\mathcal{T}$, set of integers $\mathcal{K}$ of nearest neighbors, data set $(\mathbb{X}, \mathbb{Y})$, number of parts M, probability threshold α, and weight λ.

Partition $(\mathbb{X}, \mathbb{Y})$ into M parts $\{(\mathbb{X}^m,\ \mathbb{Y}^m) \mid m = 1, 2, ..., M\}$.
for $m = 1, 2, ..., M$ **do**:
    Set $(\mathbb{X}^{train}, \mathbb{Y}^{train}) = \left\{\bigcup_{w=1, w\neq m}^{M} (\mathbb{X}^w, \mathbb{Y}^w)\right\}$ and $(\mathbb{X}^{val}, \mathbb{Y}^{val}) = (\mathbb{X}^m, \mathbb{Y}^m)$.
    for $k \in \mathcal{K}$ **do**:
        Compute $r(\boldsymbol{x})_{train}$ for each $\boldsymbol{x} \in \mathbb{X}^{train}$ using its k nearest neighbors.
        Compute the distance ratio, $r(\boldsymbol{x})_{val}$, and the closed-set prediction, $y^\star$, for each $\boldsymbol{x} \in \mathbb{X}^{val}$ using $\mathbb{X}^{train}$ to search for the k nearest neighbors.
        for $t \in \mathcal{T}$ **do**:
            Estimate γ and τ using threshold t and the distance ratios of the training data, $r(\boldsymbol{x})_{train}$.
            Compute the excesses of the validation data, $s_i = r(\boldsymbol{x})_{val} - t$, for each $\boldsymbol{x} \in \mathbb{X}^{val}$ where $r(\boldsymbol{x})_{val} > t$, and the revised prediction, $\hat{y}$, for each $\boldsymbol{x} \in \mathbb{X}^{val}$.
            Compute $O_m(k, t \mid \lambda)$ using (13).
        end for
    end for
end for
Compute $O(k, t \mid \lambda) = \frac{1}{M} \sum_{m=1}^{M} O_m(k, t \mid \lambda)$ for each k and t.
return $(\hat{k}, \hat{t}) = \arg\max_{k\in\mathcal{K}, t\in\mathcal{T}} O(k, t \mid \lambda)$.

We consider a grid search over different values for k and t to estimate the optimal hyperparameters that maximize the objective in (13). This procedure is performed using cross-validation, the steps of which are given in Algorithm 1. Note that if λ is large, k and t are selected by only considering the misclassification rate. This may result in a poor choice for the threshold of the generalized Pareto distribution, denoted by t. Consequently, a poor choice of the threshold would lead to poor estimates of γ and τ. Hence, λ should be restricted to a small value, *i.e.* $\lambda < 0.5$. This will ensure that the objective in (13) ultimately measures the goodness-of-fit of the generalized Pareto distribution to the set of excesses.

Note that α in Algorithm 1 is user-specified and can be interpreted as the acceptable false-positive rate. Furthermore, we recommend choosing $\lambda < \alpha$ as a rule-of-thumb. Note that all other parameters and hyperparameters are selected with Algorithm 1. It only required to specify the sets $\mathcal{K}$ and $\mathcal{T}$. The set $\mathcal{K}$ is specified by considering the general procedures for the k-nearest neighbors classifier. Moreover, $\mathcal{T}$ in Algorithm 1 must be

restricted to ensure that there is a sufficient number of excesses to perform maximum likelihood estimation. We specify the range of t in terms of the order statistics of $r(\boldsymbol{x}_i), \boldsymbol{x}_i \in \mathbb{X}^{train}$. Since the generalized Pareto distribution is used to estimate the upper tail of R, the lowest threshold that can be considered is the median of $r(\boldsymbol{x})$. The upper bound for t can be specified as the q^{th} quantile of $r(\boldsymbol{x})$ such that $q < 1 - \frac{n_e}{n_{train}}$, where $n_e \in \mathbb{Z}, n_e < \frac{n_{train}}{2}$, is the required number of excesses above t. Once k and t are selected via Algorithm 1, the full training set is used to estimate γ and τ as discussed in Section 3.2.

3.4 Data Manipulation and Computational Complexity

It is well known that distance-based approaches are computationally inefficient and subject to the curse of dimensionality. To make the proposed method suitable for big data, it is required to use appropriate dimension reduction or feature extraction techniques before applying Algorithm 1. As will be discussed in Section 4, we apply our method on large image data sets. One approach, used in all our applications, is to use a neural network to extract a lower-dimensional feature space where the data of the known classes are optimally separated. Training is initialized by fitting a neural network on the training data of the known classes. After training the neural network, the output of the penultimate layer of the neural network is extracted for each training sample, resulting in a data set of dimension $n_{train} \times L$. We use this feature space together with Algorithm 1 to fit our model. This feature extraction method is similar to the one adopted in Bendale and Boult (2016). However, Bendale and Boult (2016) only use the correctly classified training data whereas in our case all the training data are used. Note that the number of classes, L, is generally much smaller than the dimension of the raw data. Consequently, the distances are less contaminated by the curse of dimensionality and the computational complexity is reduced.

The computational complexity of our proposed model can be examined by considering the complexity of the k-nearest neighbors algorithm. Using 'big-O' notation, the complexity of predicting n_{val} samples from a training data set of size n_{train} is at most $O(n_{train} n_{val}(L + 2k))$. Note that the complexity of computing $P(R > r(\boldsymbol{x}^\star))$ is dominated by that of the k-nearest neighbors search and is therefore negligible.

The complexity of Algorithm 1 follows in a similar manner. Note that the complexity of computing the objective in (13) for each t, steps 7 to 10 of Algorithm 1, is dominated by the computations in steps 4 to 6. Consequently, the computational complexity of Algorithm 1 for one part, m, is approximately $O((n_{train}^2 k + n_{train} n_{val} k)(L + 2k))$.

Remark that the code provided in the supplementary material uses the `numba` Python module from Lam et al. (2015) to reduce the time complexity. The training and prediction steps of the proposed method are executed in parallel by using `numba`. A further discussion of the computational complexity of our method is provided in the supplemental document.

3.5 Toy Example

In this section, we provide a two-dimensional example of our proposed model and compare it with the OSNN model of Júnior et al. (2017). We show that $k > 1$ leads to a smoother decision boundary and that the generalized Pareto distribution accurately models the tail of $r(\boldsymbol{x})$. The full details of this example can be found in the supplementary material.

Figure 1 displays the training and test data, respectively. Note there are three classes for training and seven classes for testing. The four new classes in the test data are all labeled 'unknown'. The optimal estimates returned by Algorithm 1 were $\hat{k} = 15$ and $\hat{t} = 0.194$, which produced the parameter estimates $\hat{\gamma} = -0.0137$ and $\hat{\tau} = 9.7787$.

The decision boundaries of our model and the OSNN model are given in Figure 2.

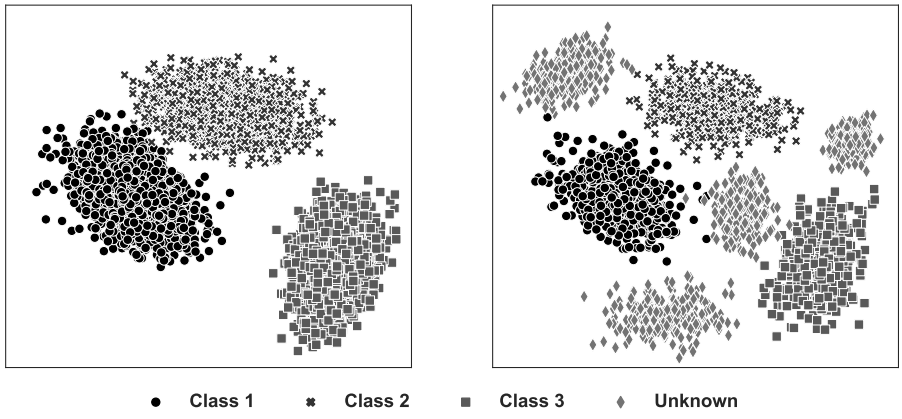

Figure 1: **Left**: Training data set. **Right**: Test data set.

Note that the black points indicate misclassified samples whereas the gray points indicate correctly classified samples. The optimal threshold for the OSNN method was $T = 0.201$ which is significantly smaller than the lower bound of 0.5 proposed by Júnior et al. (2017). Figure 2 demonstrates that using $k > 1$ nearest neighbors reduces the variance of the decision boundary which reduces the degree of overfitting. Furthermore, it was found that the false-positive rate of the test data was approximately $\alpha = 2\%$. The threshold selected with Algorithm 1 can be interpreted as one that maximizes the goodness-of-fit of the generalized Pareto distribution whilst maintaining a false-positive rate of approximately α.

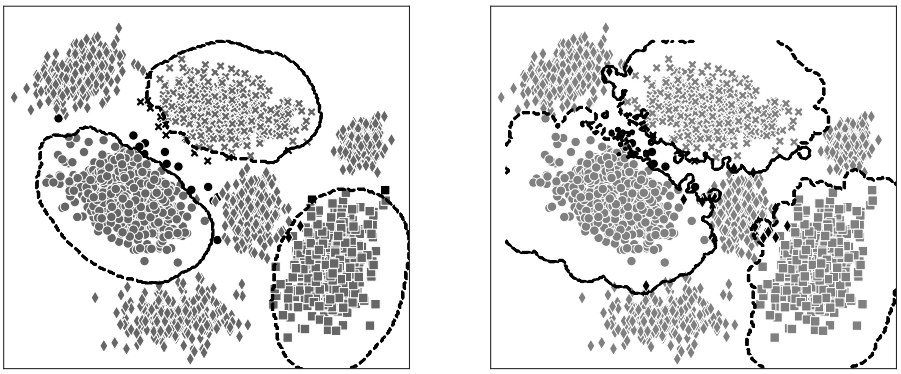

Figure 2: **Left**: Decision boundary produced by our model. **Right**: Decision boundary produced by the OSNN model.

4 Experimental Analyses

In this section, we apply our method to four image data sets. We refer to our method as the k-OSNN method, and compare it to the OSNN method of Júnior et al. (2017), the OpenMax method of Bendale and Boult (2016), and the k-nearest neighbors outlier detection (KNN-OD) method of the PyOD library of Zhao et al. (2019). Ultimately, we compare the performance of our method with the OSNN method. However, the comparisons between the OSNN and KNN-OD methods are used to investigate the effect of k on the OSR performance. Moreover, we make comparisons between the k-OSNN and KNN-OD methods

to investigate the effect of using extreme value theory for OSR. Lastly, the OpenMax method is compared to the k-OSNN method since it was designed to be applied using the same neural network-based feature-extraction method used in all our applications.

4.1 Implementation Details

The data sets used in our experiments are described below.

MNIST digits database (Lecun et al., 1998): This data set that consists of 60 000 grayscale images of handwritten digits from 0-10. There are 50 000 images in the training data and 10 000 images in the test data. A simulation was performed by repeating the following steps 10 times for each partition. We randomly selected 8, 6, 4, and 3 classes as the known classes and selected the training data of these classes. The MNIST test data labels were relabeled such that all the classes that were not selected as 'known' during training were labeled as 'unknown'. We report on the average performance of the 10 repetitions for each of the data partitions considered.

Cifar-10 and Cifar-100 (Krizhevsky, 2009): The Cifar-10 data set contains 60 000 images of 10 classes. Each class has 5 000 images in the training set and 1 000 images in the test set. Cifar-100 contains images of 100 classes where each class has 500 images in the training set and 100 images in the test set. The Cifar-10 training data set was used for training. For testing, the Cifar-10 and Cifar-100 test data sets were combined. All 100 classes in the Cifar-100 test data were considered to be 'unknown'.

DeepWeeds (Olsen et al., 2019): The DeepWeeds data set consists of 17 509 images of eight weed species native to northern Australia. For each weed species, images that do not belong to any of the eight species were collected and labeled 'negative'. This resulted in a total of 9 106 images labeled 'negative'. The images of the eight known weed species were partitioned into 6 427 training images, 715 validation images, and 1 261 test images. All images labeled 'negative' were added to the test data and considered as 'unknown'.

BirdCall (Cornell Lab of Ornithology, 2020): The sound files of the BirdCall data set were processed into Mel Spectrogram images using five-second clips. The training and validation data contain 45 000 and 4 584 Mel Spectrograms, respectively, of 49 bird species

(known classes). The test data contain 10 000 Mel Spectrograms of the 49 known species and 95 132 Mel Spectrograms of bird calls from 85 unknown species.[1]

A convolutional neural network was trained on each of the training data sets. The respective model architectures, training procedures, and weight tensors are given in the supplementary material.

The k-OSNN method was trained using Algorithm 1. Predictions on the test data were made using the decision rule in (6). Note that in our applications we combined the OSNN and OSNN$^{\text{CV}}$ methods in Júnior et al. (2017). The steps outlined in Júnior et al. (2017) were used to train the OSNN method. Moreover, the $\hat{k}$ returned by Algorithm 1 was used for the KNN-OD method. For all applications of the OpenMax method, we used the hyperparameters that produced the best results on the test data. This would not be valid in real-word applications. We simply used this approach to compare the k-OSNN method with the best possible OpenMax model. The code to reproduce all the applications and further descriptions of the implementation details are given in the supplementary material.

4.2 Performance Measures

To measure the *openness* of the problem, the measure proposed in Scheirer et al. (2013) was used. Let L_{train} and L_{test} be the number of class labels available for training and testing, respectively, and let L_{target} represent the number of classes to be recognized during testing. Scheirer et al. (2013) defined the *openness* of a problem by

$$\mathcal{O} = 1 - \sqrt{\frac{2 \times L_{train}}{L_{test} + L_{target}}}.$$

In our experiments, $L_{train} < L_{target} = L_{test}$.

The accuracy, open-set micro-F1, and open-set macro-F1 scores were used to evaluate the OSR models on the test data. The open-set micro- and macro-F1 scores, defined in Júnior et al. (2017), are extensions of the F1-score for the open-set, multi-class setting. This is achieved by micro- or macro-averaging the precision and recall of the known classes. Note

[1]Users of `xenocanto.org` are acknowledged for collecting the bird calls and making this data available.

that the precision and recall of the class labeled 'unknown' are not used in the computation of the micro- and macro-F1 scores, which is different from the conventional micro- and macro-F1 scores. Lastly, the precision and recall of the class labeled 'unknown' were used to investigate the models' ability to detect unknown classes.

4.3 Results

The results of the four applications are discussed in this section.

4.3.1 MNIST Data Set

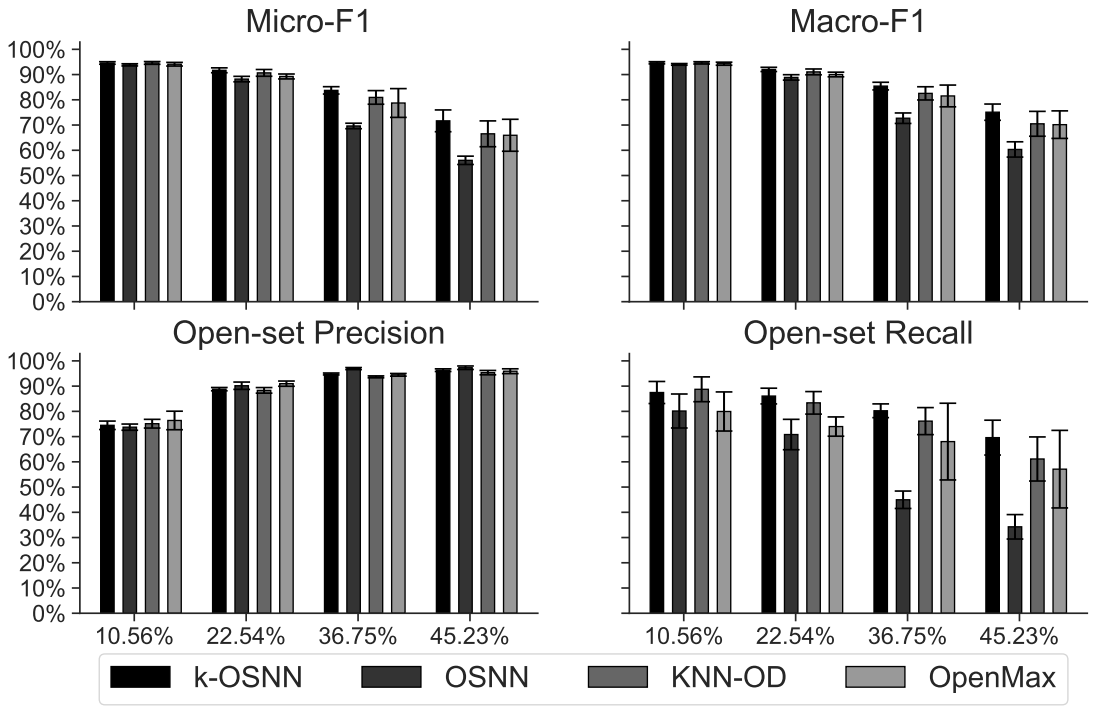

Figure 3: Results for the MNIST data set. The performance measure of each method is given on the vertical axes and the *openness* is given on the horizontal axes. The legend indicates the different methods.

Recall that for the MNIST data set four splits of 'known' and 'unknown' classes were considered. These splits were $(8, 2), (6, 4), (4, 6)$ and $(3, 7)$, resulting in $\mathcal{O} = 10.56\%$, 22.54%, 36.75% and 45.23%. The results in Figure 3 summarize the average performance measures of each split for the 10 repetitions. The standard deviations of the performance

measures over the 10 repetitions are plotted on each bar. The improvement of the k-OSNN model over the OSNN model becomes clear as $\mathcal{O}$ increases, indicating that $k > 1$ is required. Furthermore, our model tends to outperform the OpenMax and KNN-OD models.

4.3.2 Cifar-10 and Cifar-100 Data Sets

Statistic	k-**OSNN**	**OSNN**	**OpenMax**	**KNN-OD**
Accuracy	77.04	75.22	46.16	55.88
Micro-F1	75.73	73.88	47.31	59.95
Macro-F1	75.76	73.92	49.88	62.56
Recall	80.17	77.66	40.10	40.38
Precision	76.50	75.41	50.64	65.23

Table 1: Results on the Cifar-10 and Cifar-100 data sets. Values are given as percentages. Note that $L_{train} = 10$ and $L_{target} = L_{test} = 110$ such that $\mathcal{O} = 69.85\%$.

Table 1 shows the results on the Cifar-10 and Cifar-100 data sets. It is clear that the k-OSNN method outperforms the other methods based on all performance measures.

4.3.3 DeepWeeds Data Set

Statistic	k-**OSNN**	**OSNN**	**OpenMax**	**KNN-OD**
Accuracy	82.71	69.42	80.47	61.97
Micro-F1	54.71	41.54	48.18	35.63
Macro-F1	61.55	45.71	51.84	36.93
Recall	82.28	66.64	81.23	58.49
Precision	97.69	98.00	96.00	97.46

Table 2: Results on the DeepWeeds data set. Values are given as percentages. Note that $L_{train} = 8$ and $L_{target} = L_{test} = 9$ such that $\mathcal{O} = 5.72\%$. However, 87.84% of the test data is from the unknown class.

Table 2 gives the results of the DeepWeeds application from which it is clear that the k-OSNN model outperformed the other methods based on all the performance measures. This indicates that it is the optimal model in terms of its ability to detect unknown classes and to maintain a low false-positive rate.

4.3.4 BirdCall Data Set

Table 3 shows the results obtained on the test data. It is clear that the k-OSNN model has a high performance when compared to the other models. Again, all performance measures are the highest for the k-OSNN method. However, note that the OSNN method also performed exceptionally well on this data set. This may indicate that $k = 1$ is optimal for this data set.

Statistic	**k-OSNN**	**OSNN**	**OpenMax**	**KNN-OD**
Accuracy	88.58	88.12	82.48	30.03
Micro-F1	55.66	54.55	34.34	17.67
Macro-F1	58.90	57.13	35.95	19.84
Recall	89.93	89.47	86.06	24.81
Precision	97.30	97.26	94.17	94.62

Table 3: Results on the BirdCall data set. Values are given as percentages. Note that $L_{train} = 49$ and $L_{target} = L_{test} = 134$ such that $\mathcal{O} = 39.53\%$.

5 Discussion

In this paper, we built upon the OSNN method of Júnior et al. (2017) by providing a distance ratio that relies on the $k > 1$ nearest neighbors and using extreme value theory to estimate an optimal threshold to discriminate between known and unknown classes. Furthermore, we provided a new, computational approach to estimate the optimal number of nearest neighbors and threshold of the generalized Pareto distribution. Our method achieved a good performance on four large data sets where it compared favourably with similar methods. Our approach is motivated by the Pickands–Balkema–de Haan theorem, which provided the required theoretical framework to perform estimation.

Júnior et al. (2017) discussed that a drawback of the OSNN method is that only the nearest neighbor of a target point is used. Consequently, the score is severely affected by outliers and has a high sample variability. Furthermore, we point to an additional drawback of the approach to select the threshold of the OSNN method. The threshold is explicitly dependent on the classes removed from the training set and labeled as 'unknown' in the validation set. There is no way to guarantee that the same threshold is optimal when all

the classes must be recognized *and* additional unknown classes must be detected. Although the distance ratio of the OSNN method is bounded, it is difficult to determine the expected false-positive rate of the model for a given threshold because the threshold does not have a probabilistic interpretation. Nevertheless, the OSNN method still performed well in most of the applications. Our extensions in the k-OSNN model clearly addressed these issues, which is demonstrated by the superior model performance.

Classifying a sample as unknown if the probability that it is from a known class is below α provides an intuitive approach to discriminate between known and unknown classes. Recall that the distance ratio is computed and the parameters of the generalized Pareto distribution are estimated using only data from the known classes. Therefore, α can be considered as the probability of classifying a sample from a known class as unknown—the expected false positive rate. This interpretation is used when specifying α. Furthermore, Algorithm 1 requires that α is specified, and α is used to determine $\hat{y}$. Therefore, k and t are dependent on α. This simplifies the hyperparameter tuning procedure because only α must be specified.

A challenging problem in extreme value theory is to select an optimal threshold for the generalized Pareto distribution. Although this problem has received considerable attention, there is still no consensus on an optimal threshold selection method. Our numerical approach provides a new method based on a mixture between the goodness-of-fit of the generalized Pareto distribution on the observed excesses and the classification performance of the model. One challenge faced by our approach is that one must specify λ in the objective. This parameter should be close to zero, and be only increased if the optimal threshold returned by Algorithm 1 is too low. Increasing λ when $\hat{t}$ is low will increase the importance of the classification performance in the objective function. Since a low threshold would result in a high false-positive rate, increasing λ in this case should put more weight on the increase in the false-positive rate, and ultimately choose a higher threshold. The software implementation of Algorithm 1 has been optimized to provide efficient speeds on large data sets. We refer to the supplementary material for additional information on the efficiency of our method.

Acknowledgement

This research received funding from the Flemish Government under the "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen" programme.
The authors report there are no competing interests to declare.

SUPPLEMENTARY MATERIAL

Implementations of experiments: The code to reproduce the tables and figures in this paper can be found in the supplementary material. This zip folder contains all the code of the paper as well as a descriptive README to install the necessary dependencies.

References

Beirlant, J., Dierckx, G., and Guillou, A. (2005), "Estimation of the extreme-value index and generalized quantile plots," *Bernoulli*, 11, 949–970, URL `http://www.jstor.org/stable/25464774`.

Beirlant, J., Goegebeur, Y., Segers, J., and Teugels, J. (2004), *Statistics of Extremes: Theory and Applications*, Probability and Statistics, 1st ed., New York: John Wiley and Sons.

Bendale, A. and Boult, T. (2016), "Towards open set deep networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1563–1572.

Cornell Lab of Ornithology (2020), "Cornell birdcall identification," data acquired from `https://www.kaggle.com/c/birdsong-recognition/data`. Online; accessed May 2021.

Davison, A. C. and Smith, R. L. (1990), "Models for exceedances over high thresholds," *Journal of the Royal Statistical Society. Series B (Methodological)*, 52, 393–442, URL `http://www.jstor.org/stable/2345667`.

del Castillo, J. and Serra, I. (2015), "Likelihood inference for generalized Pareto distribution," *Computational Statistics and Data Analysis*, 83, 116–128.

Geng, C., Huang, S., and Chen, S. (2018), "Recent advances in open set recognition: A survey," *CoRR*, http://arxiv.org/abs/1811.08581.

Jain, L., Scheirer, W., and Boult, T. (2014), "Multi-class open set recognition using probability of inclusion," in *Computer Vision – ECCV 2014*, vol. 8691, pp. 393–409.

Júnior, P., de Souza, R., Werneck, R., Stein, B., Pazinato, D., de Almeida, W., Penatti, O., Torres, R., and Rocha, A. (2017), "Nearest neighbors distance ratio open-set classifier," *Machine Learning*, 106, 359–386.

Krizhevsky, A. (2009), *Learning Multiple Layers of Features from Tiny Images*.

Lam, S., Pitrou, A., and Seibert, S. (2015), "Numba: A llvm-based python jit compiler," in *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*, pp. 1–6.

Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998), "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, 86, 2278–2324.

McNeil, A., Frey, R., and Embrechts, P. (2015), *Quantitative Risk Management: Concepts, Techniques and Tools*, Princeton series in finance, Princeton, NJ, USA: Princeton University Press.

Olsen, A., Konovalov, D., Philippa, B., Ridd, P., Wood, J., Johns, J., Banks, W., Girgenti, B., Kenny, O., Whinney, J., Calvert, B., Azghadi, M., and White, R. (2019), "Deepweeds: A multiclass weed species image dataset for deep learning," *Scientific Reports*, 9, URL `https://doi.org/10.1038/s41598-018-38343-3`.

Pimentel, M., Clifton, D., Clifton, L., and Tarassenko, L. (2014), "A review of novelty detection," *Signal Processing*, 99, 215–249.

Rudd, E., Jain, L., Scheirer, W., and Boult, T. (2018), "The extreme value machine," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40, 762–768.

Scheirer, W., Jain, L., and Boult, T. (2014), "Probability models for open set recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36, 2317–2324.

Scheirer, W., Rocha, A., Micheals, R., and Boult, T. (2011), "Meta-recognition: The theory and practice of recognition score analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33, 1689–1695.

Scheirer, W., Rocha, A., Sapkota, A., and Boult, T. (2013), "Toward open set recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35, 1757–1772.

Talagala, P., Hyndman, R., and Smith-Miles, K. (2020a), "Anomaly detection in high-dimensional data," *Journal of Computational and Graphical Statistics*, 1–15.

Talagala, P. D., Hyndman, R. J., Smith-Miles, K., Kandanaarachchi, S., and Muñoz, M. A. (2020b), "Anomaly detection in streaming nonstationary temporal data," *Journal of Computational and Graphical Statistics*, 29, 13–27, URL `https://doi.org/10.1080/10618600.2019.1617160`.

Vignotto, E. and Engelke, S. (2018), "Extreme value theory for open set classification – gpd and gev classifiers," URL `https://arxiv.org/abs/1808.09902`.

Weng, T.-W., Zhang, H., Chen, P.-Y., Yi, J., Su, D., Gao, Y., Hsieh, C.-J., and Daniel, L. (2018), "Evaluating the robustness of neural networks: An extreme value theory approach," *arXiv preprint arXiv:1801.10578*.

Yang, J., Zhou, K., Li, Y., and Liu, Z. (2021), "Generalized out-of-distribution detection: A survey," URL `https://arxiv.org/abs/2110.11334`.

Zhao, Y., Nasrullah, Z., and Li, Z. (2019), "Pyod: A python toolbox for scalable outlier detection," *Journal of Machine Learning Research*, 20, 1–7.

Zheng, L., Ismail, K., and Meng, X. (2014), "Freeway safety estimation using extreme value theory approaches: A comparative study," *Accident Analysis & Prevention*, 62, 32–41.