Heterogeneity-Stratified Bootstrap Oversampling for Training a Spoiled Food Detector

Pertami J. Kunz*, Abdelhak M. Zoubir†
*Graduate School Computational Engineering, *†Signal Processing Group
Technische Universität Darmstadt
Darmstadt, Germany
Email: *pertami.kunz@ieee.org, †zoubir@ieee.org

***Abstract*—We propose the Heterogeneity-Stratified Bootstrap (HSBoot), a stratification method that gives higher resampling probabilities to the sample points in the less homogeneous regions. We demonstrate its advantage in the case of training a detector by oversampling the under-represented class in an imbalanced data set. We took a case study of a spoiled food detector in form of an electronic nose. The performance metrics were calculated on the out-of-bag test set as well as on measurements collected from another sensor.**

***Index Terms*—Oversampling, Bootstrap, Out of Bag Bootstrap, Detection, Classification, IoT**

I. Introduction

It is often the case that the class distribution in a dataset is not equal. The learning process to develop effective decision boundaries to support the decision-making process is called imbalanced learning [12]. Some machine learning algorithms such as naive Bayes classifier [8, Chapter 4], decision trees [10], quadratic discriminant analysis [7], and neural network [2] are often biased towards the majority classes than the minority target class, such that there is a higher misclassification rate in the target instances [9]. Using a balanced training set may help. This may be created with an artificially equal class distribution by oversampling the instances in the minority class.

The simplest oversampling method is by randomly duplicating instances in the minority class [11]. Other techniques include the Synthetic Minority Oversampling Technique (SMOTE) [4] and its variations [6], the Adaptive synthetic sampling approach for imbalanced learning (ADASYN) [9], and data augmentation [18].

In this paper, we propose an oversampling method based on the stratified resampling with replacement, or stratified bootstrapping. It is well known that stratification reduces variance [15]. In regression problems, the stratified bootstrap was proven to be robust against outliers [13]. The proposed stratification is not only done based on the class labels but also the heterogeneity in the feature space. The proposed method is applied to train a spoiled food detector.

Anosmic or visually impaired people may fail to recognise spoiled food in their fridge or pantry. A smart detector that can be trained to recognise such hazards is needed. Intelligently choosing a method to train the algorithm may reduce dependencies on the complexity and the amount of data collected. IoT devices do not always support collection of long data segments. Bootstrap techniques help in estimating statistical characteristics of interest in the case of limited samples [19]–[21].

The measurements in this study were collected using the BME688 (Bosch Sensortec) [3]. It is a metal oxide-based sensor that detects gases by adsorption and subsequent oxidation or reduction on its sensitive layer. It is capable of measuring volatile organic compounds (VOCs) in the surrounding air. The metal oxide layer of the gas sensor at different temperatures allows measurements with different sensitivities, thereby creating unique fingerprints for different gas compositions. In other words, the sensor acts as an electronic nose that can distinguish different gas compositions by their unique digital fingerprints. However, it needs to first learn about the different gases. A trained classification model can then be deployed on a microcontroller that will take the readings from the BME688, which in our case is the Adafruit HUZZAH32 - ESP32 Feather board [1].

Given the measurement set $\mathcal{X} = \{(x_j, y_j)\}_{j=1}^{N}$, where the total number of samples N can be broken down according to the class labels,

$$N = N_0 + N_1, \quad N_0 >> N_1, \tag{1}$$

and the subscripts 0 and 1 indicate fresh and spoiled, respectively, we would like to report the performance metrics with the proposed oversampling method on the Out-of-Bag Bootstrap (or Out-of-Bootstrap, OOB) test set [5], [14] as well as on measurements resulting from another sensor.

Next, Section II elaborates the proposed oversampling algorithm, a dummy example to illustrate it, and its application on the case study. Section III includes the qualitative results and finally Section IV concludes the paper and discusses the future direction.

The work of Pertami J. Kunz is supported by the Graduate School CE within the Centre for Computational Engineering at Technische Universität Darmstadt.

979-8-3503-3959-8/23/$31.00 ©2023 IEEE

II. Methodology

Suppose the feature space can be divided into K disjoint grids. We propose using the heterogeneity measure

$$H(C,k) = -\sum_{c=1}^{C} \frac{a_{ck}}{N} \log\left(\frac{a_{c,k}}{\sum_{c=1}^{C} a_{c,k}}\right), \quad k = 1,\ldots,K, \tag{2}$$

where a_{ck} is the number of sample points that belong to class c in grid k such that the grids with no sample points or with homogeneous sample points will have $H(C,k) = 0$. This was inspired by the homogeneity score in clustering [16] and here we treat each grid as one cluster. The resampling probability distribution is then modified such that the homogenous area becomes less likely to be selected and the samples in the heterogeneous area are more favored. Contrary to the other oversampling methods previously discussed, where the treatments are focused on the minority class, this method also affects the majority class.

Algorithm 1 summarises the proposed method applied for oversampling. A simple example with two classes (Class $*$ and Class $\circ$) with 6 and 8 instances is illustrated in Table I. The two-dimensional feature space is divided into $K = 18$ grids. Originally, the stratified resampling probability distribution is uniform with $p_*(k) = 1/6$ and $p_\circ(k) = 1/8$, $k = 1,\ldots,K$ (Ia). The heterogeneity measure is calculated (Ib) and as we can see the probabilities for both classes in the more heterogeneous area increased while those in the more homogenous area are reduced (Ic).

Algorithm 1 Oversampling with Heterogeneity-Stratified Bootstrap (HSBoot)

Step 1 Divide the sample domain into K grids.

Step 2 For each grid k, $k = 1,\ldots,K$, calculate the heterogeneity measure $H(C,k)$ (Eq. 2)

Step 3 Modify the resampling probability distribution $p_c(k)$ for each class c, $c = 1,\ldots,C$, in cluster k,

$$p'_c(k) = \frac{\frac{1}{N_c} + \gamma H(C,k)}{1 + \gamma \sum_k a_{ck} H(C,k)}, \tag{3}$$

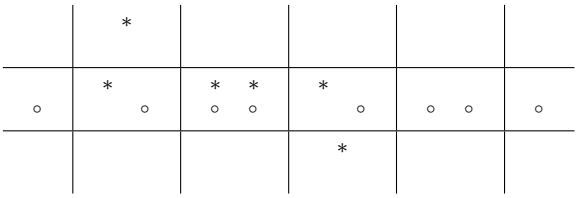
where γ is a hyperparameter constant that determines the influence of the heterogeneity, a_{ck} is the number of samples in grid k that belongs to class c, and the denominator is to make sure that the sum equals 1.

Step 4 HSBoot resampling and oversampling: Sample N_0 instances with replacement from Class 0 (non-target) and likewise N_0 instances from Class 1 (target) with the updated distribution p'_c. Let this sample set with $2N_0$ instances be $\mathcal{X}_{\text{train}}$.

Step 5 Out-of-Bootstrap (OOB) test set: take the observations from $\mathcal{X}$ that do not make it to $\mathcal{X}_{\text{train}}$ to be the bootstrap test set, $\mathcal{X}_{\text{test}}$.

The problem with using the OOB test set is that the remaining minority instances that did not make it to the training will be even smaller in proportion to the majority instances.

TABLE I: Dummy example of $C = 2$ classes with respectively 6 and 8 sample points, and $K = 18$ grids.



(a) Random resampling probability distribution $p_c(k)$

	.167							
.125	.167	.125	.167	.167	.125	.125	.125	.125
				.167				

(b) Heterogeneity measure $H(C,k)$

0	0	0	0	0	0
0	.099	.198	.099	0	0
0	0	0	0	0	0

(c) Updated resampling probability distribution $p'_c(k)$

	.105							
.078	.167	.140	.229	.229	.167	.140	.078	.078	.078
					.105			

Therefore, we also took measurements from a different sensor that was measuring the same specimens at around the same time, as an additional test set.

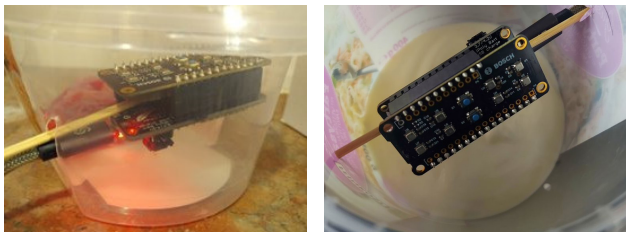
The gas compositions of the following specimens were measured:

1) Fresh Chicken: a piece of fresh, raw chicken
2) Yoghurt: fresh yoghurt
3) Beef: a piece of fresh raw beef
4) Coffee: a handful of coffee beans
5) Mix (target): a piece of spoiled raw chicken, mixed with some fresh vegetables
6) Rotten Chicken (target): a piece of spoiled raw chicken

Each specimen was placed in a plastic container together with the board that was suspended a few centimeters above the base of the container (see Fig. 1). The containers were not completely sealed off, hence the sensor still has a slight exposure to the atmosphere in the well-ventilated room where the experiments took place.

For the purpose of this paper, we took only 500 measurements for each of the specimens mentioned. We used two predictive features, the temperature and gas resistance. The original data were in degree Celcius and Ohms, respectively, then normalised such that they have 0 to 1 range.

Fig. 2 shows the gas measurements of the 6 specimens. The



(a) (b)

Fig. 1: Training the electronic nose with different specimens.

first 4 specimens are non-target (fresh) and the last 2 are the target (spoiled) classes. Fig. 3 shows the gas measurements from a different sensor. The new test set is left as is (i.e. not resampled, hence unbalanced).

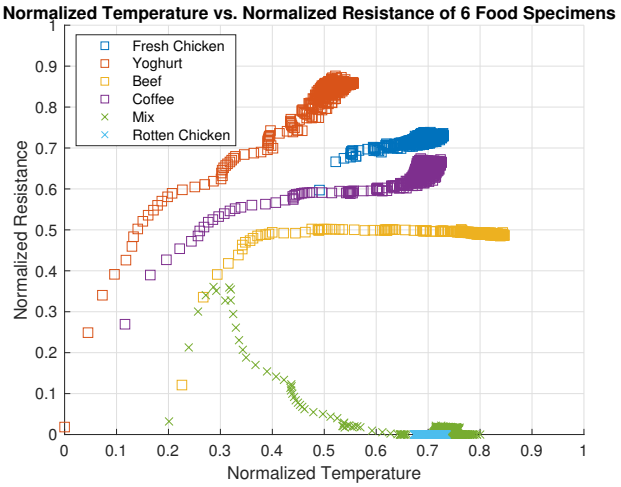


Fig. 2: The measurements for the training set

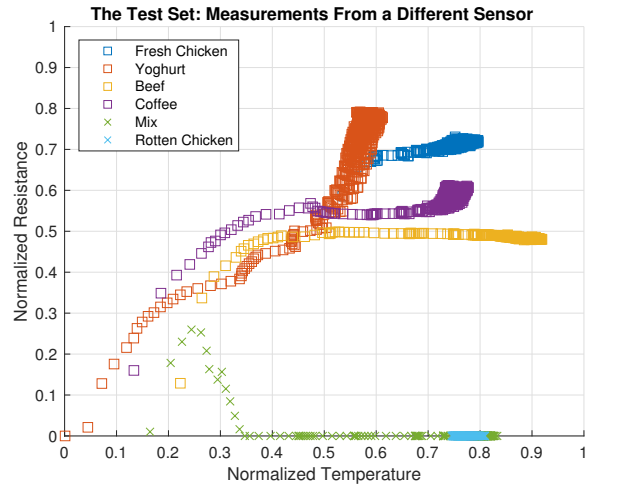


Fig. 3: The measurements as the additional test set

The random resampling probability for Class 0 is $1/(4\times500)$ and for Class 1 (target) is $1/(2\times500)$. After taking the heterogeneity into account, the probability distribution was updated, as illustrated in Fig. 4.

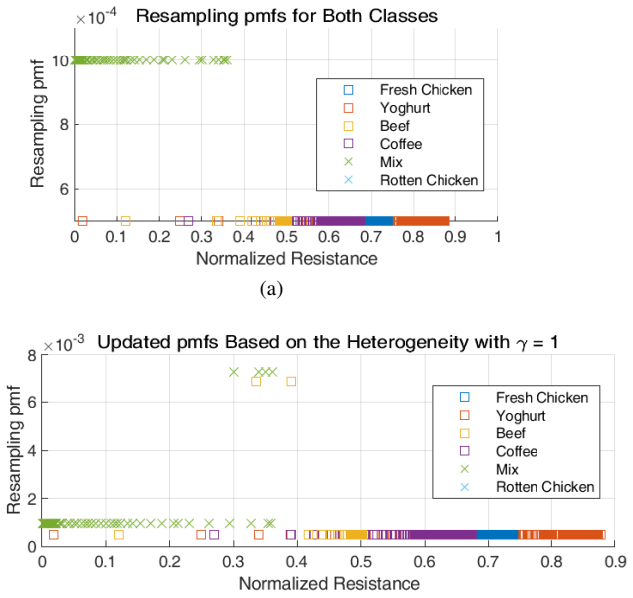


Fig. 4: Illustration of the updated resampling PMFs

In the experiment, we tried different λ: $\lambda = 0.1$, $\lambda = 1$, and $\lambda = 5$, and compared the performance of our method with the random oversampling. In the next section we report the performance of the trained models based on different algorithms: naive Bayes classifier [8, Chapter 4], decision trees [10], quadratic discriminant analysis [7], and neural network [2]. For each of the algorithm, the hyperparameters were optimised with leave-out 0.3 cross validation.

III. Results and Discussion

We list in Table II the following classification metrics, where TP is the true positives, TN the true negatives, FP the false positive, and FN the false negatives [17]:

1) Sensitivity (SN) or true positive rate (TPR) or recall (REC) = TP / (TP + FN)
2) Specificity (SP) or true negative rate (TNR) = TN / (TN + FP)
3) Accuracy (ACC) = (TP + TN) / (TP + TN + FN + FP)
4) Error rate (ERR) = (FP + FN) / (TP + TN + FN + FP)
5) Precision (PREC) = TP / (TP + FP)
6) F1 score (F1) = 2 * PREC * REC / (PREC + REC)

We observe that the proposed resampling method might slightly reduce the sensitivity and increase the error rate, but the specificity, accuracy, precision, and F1 score are constantly better, which is favorable in an unbalanced data set. We also observe that higher λ seems to improve the metrics up to certain point, then there is a diminishing return.

TABLE II: Average out of 500 trials for SN (sensitivity, or true positive rate or recall), SP (specificity, or true negative rate), ACC (accuracy), ERR (error rate), PREC (precision), F1 (F1 score, harmonic mean of precision and recall)

(a) Naive Bayes

	Random oversampling, on OOB test set	**Proposed HS-Bootstrap,** on OOB test set			Random oversampling, on another sensor test set	**Proposed HS-Bootstrap,** on another sensor test set		
		$\lambda = 0.1$	$\lambda = 1$	$\lambda = 5$		$\lambda = 0.1$	$\lambda = 1$	$\lambda = 5$
SN	**.9945**	.9917	.9917	0.9905	.9842	**.9850**	.9797	.9701
SP	.9990	.9991	.9994	**.9995**	.9944	.9945	.9960	**.9961**
ACC	.9983	.9966	**.9968**	.9965	.9910	**.9914**	.9906	.9874
ERR	**.0017**	.0034	.0032	.0035	.0090	**.0086**	.0094	.0126
PREC	.9944	.9982	.9988	**.9990**	.9888	.9890	.9920	**.9921**
F1	.9944	.9949	**.9952**	.9947	.9863	**.9868**	.9855	.9806

(b) Decision Tree

	Random oversampling, on OOB test set	**Proposed HS-Bootstrap,** on OOB test set			Random oversampling, on another sensor test set	**Proposed HS-Bootstrap,** on another sensor test set		
		$\lambda = 0.1$	$\lambda = 1$	$\lambda = 5$		$\lambda = 0.1$	$\lambda = 1$	$\lambda = 5$
SN	.9989	.9978	.9982	**.9990**	**.9991**	.9982	.9980	.9983
SP	.9980	.9984	**.9987**	**.9987**	.9932	.9933	.9950	**.9957**
ACC	.9981	.9982	.9985	**.9988**	.9952	.9952	.9960	**.9965**
ERR	.0019	.0018	.0015	**.0012**	.0048	.0048	.0040	**.0035**
PREC	.9893	.9967	.9973	**.9974**	.9867	.9869	.9901	**.9914**
F1	.9940	.9972	.9977	**.9982**	.9928	.9929	.9940	**.9948**

(c) Quadratic Discriminant

	Random oversampling, on OOB test set	Proposed HS-Bootstrap, on OOB test set			Random oversampling, on another sensor test set	Proposed HS-Bootstrap, on another sensor test set		
		$\lambda = 0.1$	$\lambda = 1$	$\lambda = 5$		$\lambda = 0.1$	$\lambda = 1$	$\lambda = 5$
SN	.9960	.9914	.9957	**.9976**	.9978	.9930	**.9956**	.9951
SP	.9967	.9975	**.9979**	.9978	.9909	**.9937**	.9932	.9921
ACC	.9966	.9955	.9971	**.9977**	.9932	.9934	**.9940**	.9931
ERR	.0034	.0045	.0029	**.0023**	.0068	.0066	.0060	**.0069**
PREC	.9826	.9951	**.9958**	.9957	.9822	.9821	**.9868**	.9846
F1	.9891	.9942	.9957	**.9966**	.9899	.9900	**.9911**	.9897

(d) Neural Network with 2 hidden layers, with 8 and 4 fully connected outputs for each hidden layer, respectively.

	Random oversampling, on OOB test set	Proposed HS-Bootstrap, on OOB test set			Random oversampling, on another sensor test set	Proposed HS-Bootstrap, on another sensor test set		
		$\lambda = 0.1$	$\lambda = 1$	$\lambda = 5$		$\lambda = 0.1$	$\lambda = 1$	$\lambda = 5$
SN	**.9803**	.9433	.9440	.9086	**.9761**	.9642	.9364	.9021
SP	.9723	.9985	.9989	**.9990**	.9449	.9540	.9698	**.9717**
ACC	.9735	.9801	**.9806**	.9688	.9553	.9574	**.9587**	.9485
ERR	.0265	.0199	**.0194**	.0312	.0447	.0426	**.0413**	.0515
PREC	.9663	.9969	.9977	**.9978**	.9338	.9377	.9504	**.9518**
F1	.9751	.9970	.9978	**.9982**	.9576	.9614	.9666	**.9683**

IV. Conclusion and Future Work

We proposed a novel stratified bootstrap method based on the heterogeneity of instances in certain regions of the feature space. This was then applied for oversampling a minority class in an unbalanced dataset when training a spoiled food detector. The proposed method constantly improved the specificity, accuracy, precision, and F1 score of the detector as compared to those by using random oversampling.

In the future, it would be interesting to replace the grids k with adaptive region of interests as far as the heterogeneity is concerned, as well as to develop a method to optimise the value of λ.

REFERENCES

[1] Esp32. https://www.espressif.com/en/products/socs/esp32, 2023.

[2] Chris M Bishop. Neural networks and their applications. *Review of scientific instruments*, 65(6):1803–1832, 1994.

[3] Bosch Sensortec. *BME688 Digital Low Power Gas, Pressure, Temperature and Humidity Sensor with AI*, 7 2022. Rev. 1.1.

[4] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[5] Bradley Efron and Robert Tibshirani. Improvements on cross-validation: the 632+ bootstrap method. *Journal of the American Statistical Association*, 92(438):548–560, 1997.

[6] Alberto Fernández, Salvador Garcia, Francisco Herrera, and Nitesh V Chawla. Smote for learning from imbalanced data: progress and challenges, marking the 15-year anniversary. *Journal of artificial intelligence research*, 61:863–905, 2018.

[7] Ronald A Fisher. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188, 1936.

[8] Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.

[9] Haibo He and Edwardo A Garcia. Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, 21(9):1263–1284, 2009.

[10] Bin Li, J Friedman, R Olshen, and C Stone. Classification and regression trees (cart). *Biometrics*, 40(3):358–361, 1984.

[11] Charles X Ling and Chenghui Li. Data mining for direct marketing: Problems and solutions. In *Kdd*, volume 98, pages 73–79, 1998.

[12] Yunqian Ma and Haibo He. Imbalanced learning: foundations, algorithms, and applications. 2013.

[13] Samuel Müller and AH Welsh. Outlier robust model selection in linear regression. *Journal of the American Statistical Association*, 100(472):1297–1310, 2005.

[14] J Sunil Rao and Robert Tibshirani. The out-of-bootstrap method for model averaging and selection. *University of Toronto*, 1997.

[15] Jonathan NK Rao and CFJ Wu. Bootstrap inference with stratified samples. Technical report, WISCONSIN UNIV-MADISON MATHEMATICS RESEARCH CENTER, 1984.

[16] Andrew Rosenberg and Julia Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, pages 410–420, 2007.

[17] Takaya Saito and Marc Rehmsmeier. The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PloS one*, 10(3):e0118432, 2015.

[18] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48, 2019.

[19] Abdelhak M Zoubir and Boualem Boashash. The bootstrap and its application in signal processing. *IEEE signal processing magazine*, 15(1):56–76, 1998.

[20] Abdelhak M Zoubir and D Robert Iskander. *Bootstrap techniques for signal processing*. Cambridge University Press, 2004.

[21] Abdelhak M Zoubir and D Robert Iskander. Bootstrap methods and applications. *IEEE Signal Processing Magazine*, 24(4):10–19, 2007.