Efficient Methods for Selective Classification under Balanced Error Rate

Pedro A. F. Castro, Robson R. da Silva and Danilo Silva

Abstract— **Deploying deep learning models in safety-critical tasks, like medical diagnosis, demands classifiers that can abstain from high-uncertainty samples to mitigate errors, which are known as selective classifiers. However, while these tasks often exhibit class imbalance, most existing approaches are based on conventional metrics such as accuracy, which are unsuited for imbalanced data. Recent work has proposed an algorithm that minimizes the balanced error rate, which is an appropriate metric for this case. Yet, their solution presents high complexity and suffers from poor scalability, restricting the application range due to computational costs. This work establishes sufficient conditions for an optimal selective classifier under the balanced error rate and proposes three novel methods that are fast and highly scalable. Experimental results show that the methods match or outperform the state-of-the-art algorithm on synthetic and real-world imbalanced datasets.**

Keywords— **Machine Learning, Deep Learning, Imbalanced Data, Uncertainty Estimation, Selective Classification.**

I. Introduction

Deep learning models are well-established across numerous areas in industry and research [1], [2]. However, some of these areas concern safety-critical applications where a bad decision can lead to harmful consequences, such as a wrong medical diagnosis in the health field. Therefore, to safeguard against critical errors, it is important to endow classifiers with the option to reject samples when their prediction is not trustworthy. Selective Classification (SC), also known as classification with a reject option, allows this by trading off coverage, i.e., the proportion of predictions that are accepted, and selective risk, i.e., the performance obtained by the model over the accepted predictions.

Earlier works have addressed this problem by assigning explicit rejection costs [3]–[5]. The issue with this approach is that defining rejection costs can be difficult, especially when the consequences of a wrong decision are hard to quantify, such as in medical diagnosis. To avoid this, other authors tackle the problem by trading off risk and coverage directly as in [6], [7], where [6] derives an algorithm with a guaranteed desired risk level, while in [7] the goal is to enhance the uncertainty estimation. Our work follows the second approach due to its value in real-world applications.

In these safety-critical scenarios, one chronic problem is that the datasets often come with imbalanced classes, i.e., one class might be underrepresented, which yields models with poor performance in the minority classes, as showcased in [8]. On top of that, in the literature of selective classification [6], [7], [9], the papers usually focus on standard metrics such as the accuracy, which is unsuitable to deal with imbalanced data. To the best of our knowledge, the only exception is the work of [5], which takes into account suitable metrics to deal with imbalanced data, namely, the Balanced Error Rate (BER), and the Worst Group Error. The authors show that solutions designed for accuracy present suboptimal performance in BER scenarios and then derive an algorithm called CS-Plugin to address SC under the BER metric in long-tail distributions, which yields better performance. Nevertheless, their algorithm requires a hyperparameter search over $\mathbb{R}^K$ (with K being the number of classes), resulting in substantial computational complexity and poor scalability; moreover, its performance is quite sensitive to the grid over which the search is performed. Even in a binary scenario, i.e., with $K = 2$ (and applying the reparametrization proposed in [5], which reduces the search to a single hyperparameter in $\mathbb{R}$), using a sufficiently large grid makes the algorithm significantly slow.

Addressing the gaps in the literature discussed above, this work introduces three novel algorithms for SC under the BER metric. These algorithms are applicable to any probabilistic machine learning classifier without requiring retraining and are derived from sufficient conditions for an optimal selective classifier for the BER, which we establish in this work. The results showcased here illustrate that our methods achieve competitive performance compared to the one in [5], even outperforming it in some scenarios, with the advantage of being faster, easier to tune, and highly scalable.

In summary, we make the following key contributions in this paper:

- We make theoretical contributions to the field of SC by deriving sufficient conditions for an optimal *selective BER* classifier;
- We propose three simple but effective methods for dealing with SC under BER that are fast, highly scalable, and easy to tune;
- We display several tests performed over synthetic and real data to illustrate a variety of scenarios under imbalanced settings, showing that one of our algorithms always matches or outperforms the state-of-the-art (SOTA) solution in [5], while another of our algorithms, in certain scenarios, can outperform both of them.

II. Background

A. Classification

Consider the problem of classification with K classes. Let $\mathcal{X} = \mathbb{R}^n$ be the feature space and $\mathcal{Y} = \{1, 2, ..., K\}$ be the

Pedro A. F. Castro, Robson R. da Silva, and Danilo Silva are with the Department of Electrical and Electronic Engineering, Federal University of Santa Catarina, Florianópolis, Brazil, e-mails: pedro.a.f.castro@posgrad.ufsc.br, robson.ufsc.2001@gmail.com, danilo.silva@ufsc.br. This work was partially supported by FAPESC (grant 48/2021) and CNPq-Brazil (grants 130157/2024-4, 304619/2022-1 and 402516/2024-9).

label space. Let $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$ be random variables with a (potentially unknown) distribution $p_{X,Y}$. A classifier is a prediction function $h : \mathcal{X} \to \mathcal{Y}$. The risk of a classifier is defined as $R(h) \triangleq E\left[L(Y, h(X))\right]$, where $L : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}^{+}$ is a given loss[1] function.

The arguably most used loss function is the so-called $0/1$ loss, defined as

$$L_{0/1}(Y, h(X)) \triangleq \mathbb{1}[h(X) \neq Y], \tag{1}$$

where $\mathbb{1}$ is the indicator function. In this case, the corresponding risk $R(h) = P[h(X) \neq Y]$ is known as the error rate (and $1 - R(h)$ is the accuracy). The Bayes optimal classifier minimizing this risk is the well-known Maximum a Posteriori (MAP) classifier [10], given by

$$h_{\text{MAP}}(x) \triangleq \arg\max_{y \in \mathcal{Y}} P[Y = y \mid X = x]. \tag{2}$$

Since $p_{X,Y}$ is generally unknown, machine learning models are trained to estimate the posterior probability $p_{Y|X}$ and this estimate is used in (2). Whenever an estimate of a probability is used in place of the true one in an optimal expression, this is called a *plug-in* approach.

B. Selective Classification

A *selective classifier* [6] is a pair (h, s), where h is a classifier and $s : \mathcal{X} \to \{0, 1\}$ is a selection function. When a selective classifier is applied to an input $x \in \mathcal{X}$, the prediction $h(x)$ is accepted if $s(x) = 1$, otherwise the prediction is rejected. A selective classifier's *coverage*, $\phi(s) = E[s(X)]$, is the proportion of accepted predictions and its *selective risk*, $R(h, s) = E\left[L(Y, h(X)) \mid s(X) = 1\right]$, is the risk it incurs over the accepted predictions. Note that the conventional risk is equal to the selective risk when $\phi(s) = 1$.

Without loss of generality, we assume $s(x) = \mathbb{1}[g(x) \geq t]$, where $g : \mathcal{X} \to \mathbb{R}$ is a *confidence estimator*, which quantifies the model's confidence on each prediction, and $t \in \mathbb{R}$ is an acceptance threshold. Thus, a selective classifier may also be denoted by the triple (h, g, t). By varying t, it is possible to trade off performance and coverage; typically, improving the performance (reducing the selective risk) comes at the cost of rejecting more predictions, thereby reducing the coverage. This inverse relationship can be visualized through the Risk-Coverage (RC) curve, which plots the classifier's selective risk against its coverage [6] (see Fig. 1 as an example).

For the 0/1 loss, the Bayes optimal selective classifier, for any coverage, is given by $h = h_{\text{MAP}}$ and

$$g(x) \triangleq \max_{y \in \mathcal{Y}} P[Y = y \mid X = x] \tag{3}$$

which is known as Chow's rule [3].

In practice, coverage and selective risk can be evaluated empirically over a dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ drawn i.i.d. from $p_{X,Y}$, yielding the *empirical coverage* $\hat{\phi}(s; \mathcal{D}) = \frac{1}{N} \sum_{(x,\,y) \in \mathcal{D}} s(x)$ and the *empirical selective risk*

$$\hat{R}(h, s; \mathcal{D}) = \frac{\sum_{(x,\,y) \in \mathcal{D}} L(h(x), y) s(x)}{\sum_{(x,\,y) \in \mathcal{D}} s(x)}. \tag{4}$$

C. Classification under Balanced Error Rate

One of the most widely used metrics for evaluating classifiers on class-imbalanced data is the BER, which captures the importance of each class individually, weighting the error accordingly. The BER is defined as

$$\text{BER}(h) \triangleq \frac{1}{K} \sum_{y \in \mathcal{Y}} P[h(X) \neq y \mid Y = y] \tag{5}$$

and $1 - \text{BER}(h)$ is known as the balanced accuracy.

Previous work [11] has shown that the optimal classifier minimizing the BER is given by

$$h^{*}(x) = \arg\max_{y \in \mathcal{Y}} \frac{P[Y = y \mid X = x]}{P[Y = y]} \tag{6}$$

which we refer to as the BER classifier.

III. Selective classification under Balanced Error Rate

A. Problem Statement

We consider the problem of selective classification using the BER as the evaluation metric. However, in contrast to the conventional error rate, the BER is not naturally induced by some loss function, as required in the description of selective classification in Section II-B. Nevertheless, we can still define the corresponding selective risk, as it consists simply of conditioning the evaluation metric on the accepted predictions. Following [5], we define the *selective BER* (SBER) as

$$\text{SBER}(h, s) \triangleq \frac{1}{K} \sum_{y \in \mathcal{Y}} P[h(X) \neq y \mid Y = y,\, s(X) = 1]. \tag{7}$$

From a learning perspective, we focus on the *post-hoc* problem: given an estimator $\hat{f} : \mathcal{X} \to [0, 1]^{K}$ of the posterior probability $p_{Y|X}$ (obtained, e.g., by training a machine learning model), we wish to design a selective classifier (h, s) minimizing $\text{SBER}(h, s)$.

B. Theoretical Results

Let $\pi_y = P[Y = y]$, $y \in \mathcal{Y}$, denote the prior distribution of Y and, for any $x \in \mathcal{X}$, let $f_y(x) = P[Y = y \mid X = x]$, $y \in \mathcal{Y}$, denote the posterior distribution of Y given $X = x$. Additionally, for any selection function $s : \mathcal{X} \to \{0, 1\}$, let $\pi_y(s) = P[Y = y | s(X) = 1]$, $y \in \mathcal{Y}$, denote the post-rejection priors, i.e., the prior distribution of Y restricted to the accepted samples, and let $\pi(s) = (\pi_1(s), \ldots, \pi_K(s)) \in [0, 1]^{K}$.

For any $c > 0$, let $\mathcal{H}_{\text{SBER}}(c)$ denote the set of all selective classifiers (h, s) that minimize $\text{SBER}(h, s)$ under the constraint $\phi(s) \geq c$ and let $\Pi_{\text{SBER}}(c) = \{\pi(s) \in [0, 1]^{K} : (h, s) \in \mathcal{H}_{\text{SBER}}(c)\}$. Our main theoretical result is the following theorem.

Theorem 1: Suppose that $h : \mathcal{X} \to \mathcal{Y}$, $s : \mathcal{X} \to \{0, 1\}$, $g : \mathcal{X} \to \mathbb{R}$, $t \in \mathbb{R}$ and $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_K) \in [0, 1]^{K}$ are such

[1] The terminology here is from decision theory and statistical learning theory; it should not be confused with a *surrogate* loss function (such as the cross-entropy loss) used to train neural networks.

that

$$h(x) = \arg\max_{y \in \mathcal{Y}} \frac{f_y(x)}{\alpha_y} \tag{8}$$

$$s(x) = \mathbb{1}[g(x) \geq t] \tag{9}$$

$$g(x) = \max_{y \in \mathcal{Y}} \frac{f_y(x)}{\alpha_y} - \sum_{y \in \mathcal{Y}} \frac{f_y(x)}{\alpha_y} \tag{10}$$

$$\alpha_y = \pi_y(s), \text{ for all } y \in \mathcal{Y} \tag{11}$$

and let $c = \phi(s)$. In this case, if $\boldsymbol{\alpha} \in \Pi_{\text{SBER}}(c)$, then $(h, s) \in \mathcal{H}_{\text{SBER}}(c)$.

Proof: (Sketch.) We first show that SBER(h, s) equals the selective risk $R(h, s)$ induced by the loss function $L(y, \hat{y}) = \mathbb{1}[\hat{y} \neq y]/(K\alpha_y)$, where $\boldsymbol{\alpha} = \pi(s)$; then, we apply the optimality result in [12]. However, since that result does not allow $\boldsymbol{\alpha}$ to depend on s, we need to already start from some optimal $\boldsymbol{\alpha}$. ■

Note that, when $\phi(s) = 1$ (e.g., if $t = -\infty$), then $\alpha_y = \pi_y$ and (8) reduces to the optimal BER classifier in (6).

While Theorem 1 gives sufficient conditions for an optimal selective BER classifier, it does not provide an explicit construction, since $g(x)$ and $\boldsymbol{\alpha}$ are dependent on each other. Moreover, even if we find a selective classifier satisfying (8)–(11), we have no guarantee that $\boldsymbol{\alpha} \in \Pi_{\text{SBER}}(\phi(s))$. Nevertheless, we can use Theorem 1 as an inspiration to propose selective classifiers that approximately satisfy (8)–(11).

C. Proposed Methods

We propose three methods inspired by Theorem 1.

1) Static SBER Classifier: The first method consists of using (8)–(10) but setting $\alpha_y = \pi_y$. This method is optimal if $\pi_y(s) = \pi_y$ for all y, i.e., the priors on Y remain constant after rejection.

2) Filter-by-Class (FBC) SBER Classifier: This method attempts to enforce $\pi_y(s) \approx \pi_y$ by rejecting the same fraction of predictions from each predicted class. Similarly to the first method, it uses (8) and (10) and sets $\alpha_y = \pi_y$, but the selection function is changed to $s(x) = s(x|h(x))$, where $s(x|y) = \mathbb{1}[g(x) \geq t_y]$, $t_y \in \mathbb{R}$ is such that

$$E[s(X) \mid h(X) = y] = P[g(x) \geq t_y \mid h(X) = y] = c \tag{12}$$

and $c > 0$ is the target coverage. In other words, the acceptance threshold for each predicted class is chosen so that the induced coverage within that predicted class is equal to c. Naturally, this implies that $\phi(s) = c$.

3) Adaptive SBER Classifier: The last method uses (8)–(10) but computes α_y iteratively by alternating (11), (10) and (9), with t chosen at each iteration to satisfy some target coverage. The details are given in Algorithm 1. If the α_y's converge, the method results in a selective classifier satisfying (8)–(11). However, since we have no guarantee of convergence, we apply exponential smoothing to reduce potential oscillations. Note that the post-rejection priors in (11), as well as the coverage $\phi(s)$, have to be estimated at each iteration using a tuning set, so the method is susceptible to overfitting.

Algorithm 1: Adaptive SBER Classifier Tuning

```
Input: Posterior $f : \mathcal{X} \to [0,1]^K$, tuning set $\mathcal{D}$, target
       coverage $c > 0$
Parameters: Iterations $M$, smoothing factor $\beta$
Initialize $s(x) = 1$ for all $x \in \mathcal{X}$
for $m = 1, \dots, M$ :
    Compute, for all $k \in \mathcal{Y}$:
        $\hat{\pi}_k(s) = \frac{1}{\hat{\phi}(s;\mathcal{D})} \frac{1}{|\mathcal{D}|} \sum_{(x,y) \in \mathcal{D}} \mathbb{1}[y = k]\mathbb{1}[s(x) = 1]$
        $\alpha_k = \begin{cases} \hat{\pi}_k(s), & \text{if } m = 1, \\ \beta\alpha_k + (1-\beta)\hat{\pi}_k(s), & \text{otherwise} \end{cases}$
    Set $g(x)$ using (10) and $s(x)$ using (9), choosing $t$
      such that $\hat{\phi}(s;\mathcal{D}) \approx c$
return $\alpha_1, \dots, \alpha_K$
```

IV. Experiments with Synthetic Data

To illustrate our methods, we start with experiments with synthetic data, where the posteriors are known exactly.

A. Data Generation

We consider a conditional isotropic Gaussian distribution for X given Y with $n = 2$ features ($\mathcal{X} = \mathbb{R}^2$) and $K = 2$ classes ($\mathcal{Y} = \{0, 1\}$), where $X \sim \mathcal{N}(\mu_Y, \sigma_Y I)$ and I denotes the identity matrix. We fix $\mu_0 = [-1, 0]$, $\mu_1 = [1, 0]$, $\sigma_0^2 = 1$ and $\sigma_1^2 = 3$ and we vary π_1 (note that $\pi_0 = 1 - \pi_1$). We can interpret feature x_1 as informative and feature x_2 as less informative.

B. Experiments

We consider the following experiments:

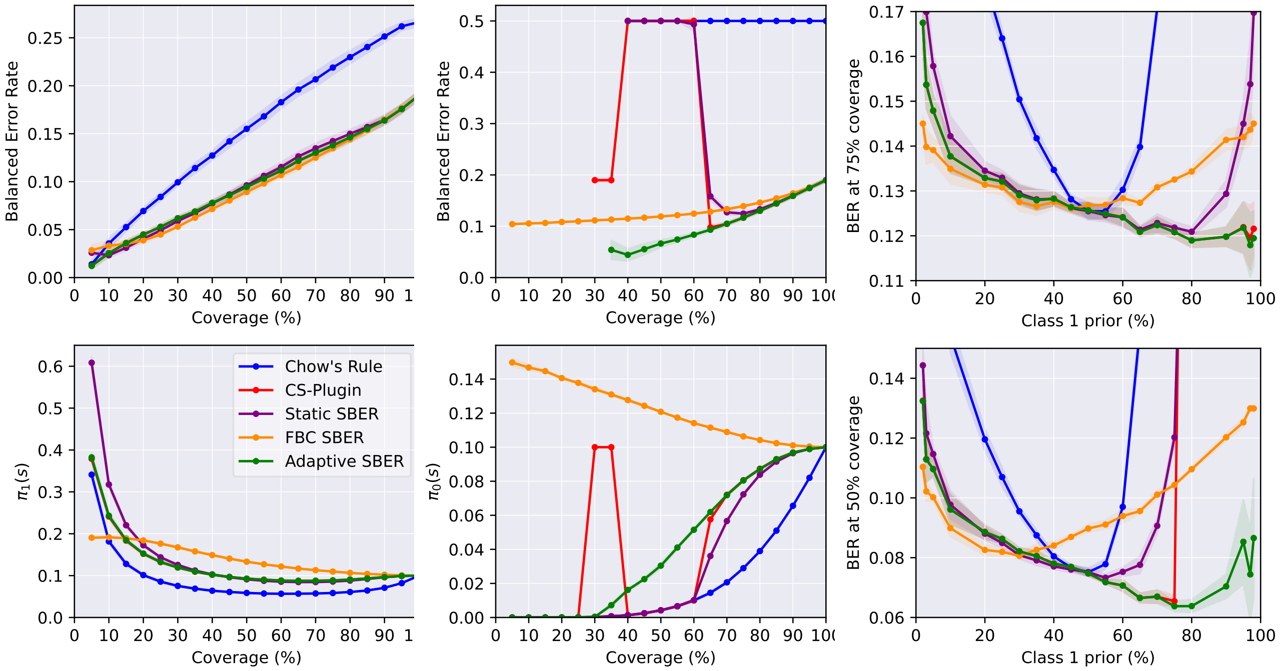
- Experiment 1: The positive class is set to be the minority class with $\pi_1 = 10\%$;
- Experiment 2: The positive class is set to be the majority class with $\pi_1 = 90\%$;
- Experiment 3: The prior of the positive class π_1 is varied from 2% to 98%.

For each experiment, we generate one tuning set and 5 test sets each with 50000 samples drawn i.i.d. from $p_{X,Y}$ and we report the mean and standard deviation of the SBER computed across the 5 test sets. To avoid unreliable statistics, we omit SBER values for which one of the classes has fewer than 50 samples (0.1% of the total) after rejection.

We also compare our methods to the SOTA algorithm in [5], using $M = 10$ iterations (as in [5]) and with the parameter $\hat{\lambda}_0$ tuned by searching over the grid $\{0, 1, \dots, 11\}$ (which includes the grid $\{1, 6, 11\}$ used in [5]). For our adaptive method, we set $M = 50$ and $\beta = 0.95$.

C. Results

In Figure 1a, the results of Experiment 1 are displayed, with the first graph exhibiting the RC curves, and the second, the post-rejection priors $\pi_1(s)$ over the coverage points. Firstly,



(a) Exp. 1 ($\pi_1 = 10\%$).

(b) Exp. 2 ($\pi_1 = 90\%$).

(c) Exp. 3 ($\pi_1 = \{2\%, ..., 98\%\}$).

Fig. 1: Results for all synthetic experiments, separated by columns. For each method, the curve shown corresponds to the mean performance across 5 trials, while the shaded region corresponds to 1 standard deviation. In (a) and (b), the top panel shows the RC curve, and the bottom panel shows the corresponding post-rejection prior of the minority class. In (c), the top and bottom panels show the BER at 75% and 50% coverage, respectively, as a function of the class 1 prior π_1.

we see that all methods designed for the BER yield better results than Chow's rule and that these methods show similar performance. The good performance of the Static SBER method can be attributed to the post-rejection priors remaining well-behaved, except at the end of the curve (coverage below 15%). The Adaptive SBER shows identical behavior to the CS-Plugin algorithm, while the FBC SBER method displays a slightly better performance across the entire RC curve, except at very low coverage values.

Figure 1b, which shows the results for the Experiment 2, exhibits a similar result for high coverage values, with all methods outperforming Chow's rule drastically and presenting similar BER values. However, as the coverage falls below 70%, both the CS-Plugin and the Static SBER's performance degenerates significantly, even following Chow's rule curve. The Adaptive SBER manages the priors better and yields better results among all algorithms until 35% coverage, but, for lower coverage values, this method reduces the minority class prevalence in a way that the number of samples for this class drops below 50, hence, the remaining points are not shown. Interestingly, although the FBC SBER method is not the one with the best results for high coverages, it shows a smooth behavior, maintaining a significant sample proportion for the minority class even at very low coverage, yielding reliable results in that scenario.

Figure 1c illustrates the results of Experiment 3, where the top panel shows the BER for 75% of coverage, while the bottom panel reports it at 50% coverage. Starting from the left on both graphs, we see that the behavior of all methods is similar to that in Experiment 1, with the FBC SBER outperforming every other algorithm, and that result is amplified as the prevalence decreases. Going to the opposite side, i.e., for high prevalence values, one can observe that the performance for the FBC SBER method degrades, while the SOTA algorithm and the Adaptive SBER yield better results. In the second panel, i.e., for 50% coverage, the behavior is similar except that, for higher prevalences (above 80%), all methods but the Adaptive SBER deteriorate abruptly.

It is worth mentioning that our methods are considerably faster to tune than the CS-Plugin. For instance, on an AMD Ryzen 5 3600 6-core CPU with 16GB RAM, running the entire Experiment 3 took on average 69 minutes for the CS-Plugin method. In contrast, the Adaptive SBER method took a fraction of the time, only 3.15 minutes, while, for the Static and FBC methods, the tuning times were negligible.

V. Experiments with Real Data

We now evaluate our methods using real-world data. In this case, the true posterior $f_y(x)$ in Sections III-B is replaced with an estimate by a machine learning model.

A. Dataset

We used the modified PatchCamelyon (PCam) benchmark dataset available on Kaggle[2], consisting of 96×96 pixel images (Figure extracted from hematoxylin and eosin (H&E)

[2] https://www.kaggle.com/competitions/histopathologic-cancer-detection

stained histopathological slides of lymph nodes and classified into tumorous and non-tumorous patches. It contains 220025 non-duplicate image samples. The dataset was randomly split into 80% training, 2% validation, and 18% testing. Each subset was independently imbalanced to a $9:1$ ratio between negative and positive classes, preserving all negative samples. The final sample counts are: 116362 (train), 2909 (validation), and 26182 (test).

B. Model Architecture and Training Details

Model training and evaluation were performed using PyTorch on a NVIDIA GeForce RTX 3070 GPU. We used the ResNet-34 [13] architecture, a widely adopted convolutional neural network for image classification tasks, initialized with ImageNet weights. Training was performed for 20 epochs using the Adam optimizer with an initial learning rate of 10^{-4} and a batch size of 64, using the cross-entropy surrogate loss function. The learning rate was dynamically adjusted via the ReduceLROnPlateau scheduler (factor=0.5, patience=2, min_lr=10^{-5}), monitoring the balanced accuracy on the validation set. To improve generalization, we applied simple data augmentation during training: random horizontal and vertical flips. The model at the best performing epoch was selected based on the balanced accuracy on the validation set.

C. Results

The Figure 2 displays the results on the PCam dataset, with the top panel exhibiting the RC curves of all methods while the bottom panel shows the post-rejection priors of the minority class to these methods. We observe that all methods except Chow's Rule perform well from 100% coverage down to approximately 80%, however, as coverage is reduced, we see that a growing performance gain is presented by the FBC SBER algorithm, with the selective risk close to 0 at 5% coverage. At the same time, the Static SBER, CS-Plugin, and Adaptive SBER start to degenerate. It can also be seen that, although the Adaptive SBER and the CS-Plugin worsen almost equally, at 30%, the CS-Plugin degenerates completely, while the Adaptive SBER remains reasonably well until around 10% of coverage.

VI. Conclusion

In safety-critical applications, SC plays a crucial role in mitigating potential errors. However, those tasks usually come with class-imbalanced data, and most solutions in the literature are based on unsuitable metrics such as accuracy. This work addresses SC under BER, where we establish sufficient conditions for an optimal selective classifier for this metric and derive three simple algorithms that are fast and highly scalable, applicable to any trained probabilistic model without requiring retraining. We displayed experiments on both synthetic and real data that show that the Adaptive SBER algorithm meets or surpasses the performance of the current SOTA algorithm across all scenarios tested, while the FBC SBER method is in some cases capable of outperforming both methods. In future work, we aim to combine both of our proposed algorithms into a single one and evaluate our methods in more real-world scenarios, including multi-class classification.

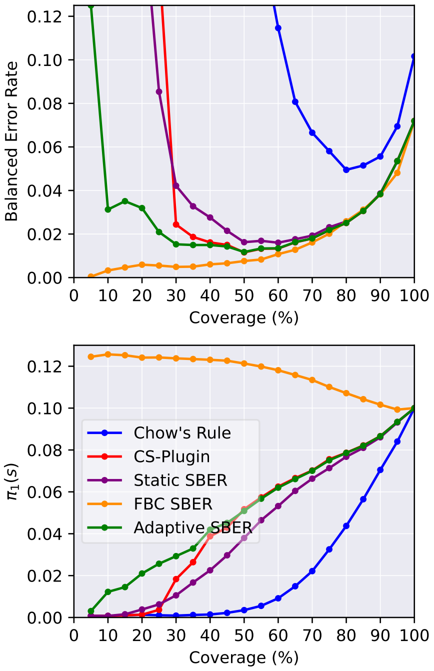


Fig. 2: RC curves and post-rejection priors of all methods in the real-world scenario with the PCam dataset.

References

[1] D. Kaul, H. Raju, and B. K. Tripathy, *Deep Learning in Healthcare*. Cham: Springer International Publishing, 2022, pp. 97–115. [Online]. Available: https://doi.org/10.1007/978-3-030-75855-4_6

[2] S. Grigorescu, B. Trasnea, T. Cocias, and G. Macesanu, "A survey of deep learning techniques for autonomous driving," *Journal of Field Robotics*, vol. 37, no. 3, pp. 362–386, 2020. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21918

[3] C. Chow, "On optimum recognition error and reject tradeoff," *IEEE Transactions on Information Theory*, vol. 16, no. 1, pp. 41–46, 1970.

[4] C. Cortes, G. DeSalvo, and M. Mohri, "Learning with rejection," in *Algorithmic Learning Theory*, R. Ortner, H. U. Simon, and S. Zilles, Eds. Cham: Springer International Publishing, 2016, pp. 67–82.

[5] H. Narasimhan, A. K. Menon, W. Jitkrittum, N. Gupta, and S. Kumar, "Learning to reject meets long-tail learning," in *The Twelfth International Conference on Learning Representations*, 2024. [Online]. Available: https://openreview.net/forum?id=ta26LtNq2r

[6] Y. Geifman and R. El-Yaniv, "Selective classification for deep neural networks," 2017. [Online]. Available: https://arxiv.org/abs/1705.08500

[7] Y. Geifman, G. Uziel, and R. El-Yaniv, "Bias-reduced uncertainty estimation for deep neural classifiers," *arXiv preprint arXiv:1805.08206*, 2018.

[8] K. Cao, C. Wei, A. Gaidon, N. Arechiga, and T. Ma, "Learning imbalanced datasets with label-distribution-aware margin loss," 2019. [Online]. Available: https://arxiv.org/abs/1906.07413

[9] L. F. P. Cattelan and D. Silva, "How to fix a broken confidence estimator: Evaluating post-hoc methods for selective classification with deep neural networks," 2024. [Online]. Available: https://arxiv.org/abs/2305.15508

[10] K. P. Murphy, *Probabilistic Machine Learning: An introduction*. MIT Press, 2022. [Online]. Available: probml.ai

[11] A. K. Menon, S. Jayasumana, A. S. Rawat, H. Jain, A. Veit, and S. Kumar, "Long-tail learning via logit adjustment," 2021. [Online]. Available: https://arxiv.org/abs/2007.07314

[12] V. Franc, D. Prusa, and V. Voracek, "Optimal strategies for reject option classifiers," *Journal of Machine Learning Research*, vol. 24, no. 11, pp. 1–49, 2023. [Online]. Available: http://jmlr.org/papers/v24/21-0048.html

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778.*, 2016. [Online]. Available: https://arxiv.org/abs/1512.03385