SafePredict: A Meta-Algorithm for Machine Learning That Uses Refusals to Guarantee Correctness

Mustafa A. Kocak, David Ramirez, *Member, IEEE,* Elza Erkip, *Fellow, IEEE,* and Dennis E. Shasha, *Fellow, ACM*

Abstract—*SafePredict* is a novel meta-algorithm that works with any base prediction algorithm for online data to guarantee an arbitrarily chosen correctness rate, $1-\epsilon$, by allowing refusals. Allowing refusals means that the meta-algorithm may refuse to emit a prediction produced by the base algorithm so that the error rate on non-refused predictions does not exceed ϵ. The SafePredict error bound does not rely on any assumptions on the data distribution or the base predictor. When the base predictor happens not to exceed the target error rate ϵ, SafePredict refuses only a finite number of times. When the error rate of the base predictor changes through time SafePredict makes use of a weight-shifting heuristic that adapts to these changes without knowing when the changes occur yet still maintains the correctness guarantee. Empirical results show that (i) SafePredict compares favorably with state-of-the-art confidence-based refusal mechanisms which fail to offer robust error guarantees; and (ii) combining SafePredict with such refusal mechanisms can in many cases further reduce the number of refusals. Our software is included in the supplementary material.

1 Introduction

Machine learning and statistical inference are the primary building blocks for systems that predict the future from the past. Prediction algorithms have been tailored to fit various applications ranging from analytics [1], to health care [2], [3], to judicial decision making [4], [5]. One of the major concerns when utilizing prediction algorithms to automate risk-critical applications is reliability.

To guarantee an error rate for an overall prediction system, a meta-algorithm should refuse to make a prediction when the meta-algorithm infers that the base prediction algorithm is likely enough to be in error. The implications of refusing to make a prediction may vary according to the application of interest. For example, in a medical diagnosis system, refusing to make a prediction may result in the collection of more information about the patient or a request to a human expert to make a decision based on a more thorough evaluation.

Inspired by the prediction-with-expert-advice framework [6], we propose *SafePredict*, an online meta-algorithm that accepts or refuses predictions of a base algorithm depending on the previous performance of the base algorithm. SafePredict asymptotically bounds the error to the desired level without any assumption on the data or the base predictor. When the error rate of the base predictor varies over time, SafePredict adapts to those changes while preserving the error guarantee.

- *M. A. Kocak is with the Broad Institute of Harvard and MIT, Cambridge, MA, 02142. E-mail: mkocak@broadinstitute.org*
- *D. Ramirez and E. Erkip are with the Department of Electrical and Computer Engineering, NYU Tandon School of Engineering, Brooklyn, NY, 11201. E-mail: dar550@nyu.edu, elza@nyu.edu*
- *D. E. Shasha is with Courant Institute of Mathematical Sciences New York University, New York, NY, 10012. E-mail: shasha@courant.nyu.edu*

1.1 Prior Work

The idea of allowing meta-algorithms to refuse to make predictions to reduce the error rate was first introduced by Chow [7]. Chow mainly focused on the classification problem, where data points have an object-label pair independently sampled from a fixed and known probability distribution. Chow showed that the optimum error-refuse trade-off is achieved by a classifier that refuses to predict when the posterior probability of the estimated label given the object is less than a certain threshold.

The major challenge in applying Chow's work is the need to know the data distribution, which is rarely available. The mainstream assumption is that the available data points are independently sampled from a fixed but unknown distribution. Given these data points, the most common approach is to estimate the underlying distribution and to use it in the Chow framework. Estimating probability distributions in high dimensional and complex datasets can be harder even than classification itself, see e.g., [8]. An alternative approach uses confidence-based refusals. One can train a classification algorithm with an arbitrary confidence score (e.g., distance to the decision boundary) and refuse to make a prediction for points with low confidence scores. Examples of this line of work are [9], [10], [11], [12].

Although many practical applications of the refusal framework exist in the literature, e.g., [13], [14], [15], [16], theoretical analyses of the suggested methods are relatively rare. Some notable exceptions are found in Wegkamp et al. [17], [18], [19], El-Yaniv and Wiener [20], [21] and Cortes et al. [22] which approach the problem from a statistical learning theory perspective and suggest minimizing a linear combination of error and refuse probabilities. Alternatively, reliable agnostic learning, proposed by Kalai et al. [23], posits an error threshold and searches for the least refusing

predictor from a family of predictors that bounds the error to the error threshold, assuming such a predictor exists.

The related work discussed so far assumes a batch setup, i.e., the algorithm is trained on a fixed set of data points which are independently and identically distributed (i.i.d.). For a more comprehensive literature review of refusal algorithms in the batch setup, please see [18], [22], [24], [25] and the references therein.

In contrast to the batch setup, a meta-algorithm in the online learning framework observes the true outcome after a prediction and then modifies its future behavior. In the conformal prediction framework of Vovk et al. [26], a base algorithm generates confidence scores for each data point. Then the conformal predictor decides to predict or refuse based on these scores. A bound on the error probability and the independence between the error events are guaranteed under the assumptions that the data points are chosen from an exchangeable (essentially i.i.d.) distribution and the base predictor is invariant to the order of the observed data points. For recent developments regarding the classification with refusal problem in the conformal prediction framework we refer the reader to [27], [28], [29] and Chapter 3 of [26]. Unfortunately, in an online setting, the probability of making errors on consecutive predictions may be correlated or the data sequence may have a non-stationary or even an adversarial distribution. Thus, any guarantees that might hold in the independent and identically distributed setting do not directly carry over.

Another approach in the online setting starts with the KWIK (knows what it knows) framework [30] which removes all assumptions about the data distributions. Instead, KWIK assumes the existence of a perfect predictor (i.e., a predictor that is always correct) among a set of predictors and aims to find this perfect predictor with a minimum number of refusals. Sayedi et al. [31] extend this framework by allowing a fixed error budget k and characterizing the minimum number of refusals by keeping the number of errors below k. Finally, Zhang et al. [32] relax the perfect predictor assumption to an l-bias assumption (i.e., predictor makes at most l errors) and allow the algorithm to refuse in order to achieve an optimal error-refuse trade-off.

1.2 Contributions

This paper makes the following main contributions:

1) The SafePredict online meta-algorithm can work with any base prediction algorithm to provide an asymptotic error guarantee on the non-refused predictions, without making any assumption about the data or the base prediction algorithm(s).
2) All the variants of SafePredict meta-algorithm refuse at most a finite number of times when the base predictor's error rate is below the target error rate ϵ.
3) Adaptive SafePredict meta-algorithm uses weight-shifting and other conventional adaptive procedures to track the error rate of the base predictor in changing environments, thus reducing the number of refusals while preserving the error guarantee.
4) Experiments show that the above theoretical guarantees are achieved in practice and translate to better error performance than other refusal algorithms. The experiments also show that combining SafePredict with previous meta-algorithms can lead to yet fewer refusals in many cases.

The rest of the paper is organized as follows. Section 2 presents the problem and provides a brief introduction to the exponentially weighted average forecasting (EWAF) [33], [34] expert advice framework. Section 3 introduces SafePredict by recasting EWAF as a randomized refusal meta-algorithm and proves its theoretical properties. Section 4 presents Adaptive SafePredict, a weight-shifting heuristic, to track changes in the error rate of the base algorithm and therefore reduce the number of refusals. Section 5 presents experiments on real and synthetic data. Section 6 concludes our work.

1.3 Notation

A summary of the notation introduced throughout the paper is given in Table 1. For each quantity, we provide the notation, a brief description, and a reference to the section where it is first introduced.

TABLE 1: Summary of the Notation

Not.	Description	Def. in Sec.
α_t, β_t	Adaptivity parameters, $\alpha_t \leq w_{P,t+1} \leq \beta_t$.	4
ϵ	Target error rate.	2
η	Learning rate, $\eta > 0$.	2.1
ρ_T	Efficiency of the meta-algorithm, T^*/T.	2
D	Dummy predictor, always refuses, $\hat{y}_{D,t} = \varnothing$ and $l_{D,t} = \epsilon$ for all t.	3
l_t	Expected loss at time t.	2.1
L_T	Cumulative expected loss, $\sum_{t=1}^{T} l_t$.	2.1
$l_{P,t}$	Loss of P at time t, assume $l_{P,t} \in [0,1]$.	2
L_{P,T,t_0}	Partial cumulative loss of P from t_0 to T, $\sum_{t=t_0+1}^{T} l_{P,t}$. The third index drops if $t_0 = 0$	2
$L^*_{P,T}$	Expected cumulative loss for the meta-algorithm $\sum_{t=1}^{T} w_{P,t} l_{P,t}$.	2
P	Base predictor.	2
P_i	i^{th} expert in the ensemble, has weight $w_{P_i,t}$ and loss $l_{P_i,t}$.	2.1
T	Time horizon.	2
T^*	Expected value of the number of (non-refused) predictions, $\sum_{t=1}^{T} w_{P,t}$.	2
V^*	Variance of the number of (non-refused) predictions, $\sum_{t=1}^{T} w_{P,t}(1 - w_{P,t})$.	2
$w_{D,t}$	Weight of the dummy, also the probability of refusal, $1 - w_{P,t}$.	3
$w_{P,t}$	Probability of making a prediction at time t.	2
$\hat{y}_{P,t}$	Prediction of P at time t.	2
$\hat{y}_t$	Filtered prediction, $\hat{y}_t = \hat{y}_{P,t}$ (predict) or $\hat{y}_t = \varnothing$ (refuse) .	2

2 Problem Setup and Background

This section introduces the mathematical formulation of the online prediction problem with a refusing meta-algorithm. Our method makes use of the prediction-within-the-expert-advice framework.

2.1 Problem Formulation

We assume access to a base predictor that produces a label prediction on an observed object. We denote the base

predictor as P and a sequence of (object, label) pairs by $(x_1, y_1), (x_2, y_2), \ldots, (x_T, y_T)$ where T is an arbitrary horizon. At each time $t \in \{1, \ldots, T\}$ the base predictor does the following

1) Observes the object x_t.
2) Predicts the corresponding label $\hat{y}_{P,t}$.
3) Observes the true label y_t, and suffers the loss $l_{P,t}$.

In our formulation, we stay agnostic to the data sequence $(x_1, y_1), (x_2, y_2), \ldots, (x_T, y_T)$ and inner workings of the base predictor P (which may for example be composed of an ensemble of predictors). We assume *only* that the loss values are scaled to the unit interval, i.e., $0 \leq l_{P,t} \leq 1 \; \forall \; t$.

For example, in a traditional classification task, the labels y_t are drawn from a finite set and 0-1 loss is used for the loss function, i.e., $l_{P,t} = 0$ if $\hat{y}_{P,t} = y_t$ and $l_{P,t} = 1$ if $\hat{y}_{P,t} \neq y_t$.

Once the base predictor P is chosen, our goal is to design a meta-algorithm M which decides either to follow the prediction made by P *or* to refuse to make a prediction for each data point. We characterize this meta-algorithm by the following:

- **Parameter**: Target error/loss rate, $\epsilon \in (0, 1)$, is the maximum average loss over time the user has specified.
- **Input**: In full generality, the input of M at time t consists of $x_i, \hat{y}_{P,i} \; \forall \; i \in \{1, ..., t\}$, and $y_j, l_{P,j} \; \forall \; j \in \{1, ..., t-1\}$. Note these are all the observed quantities *before* the label y_t is revealed.
- **Output**: A randomized decision to predict (or refuse) at time t. We represent the output of M as a probability value $w_{P,t} \in [0, 1]$ that is used to decide the final prediction $\hat{y}_t$ as follows:

$$\hat{y}_t = \begin{cases} \hat{y}_{P,t} & \text{with prob. } w_{P,t} \\ \varnothing & \text{with prob. } 1 - w_{P,t} \end{cases},$$

  where $\varnothing$ denotes a refusal.

A pictorial representation of this general framework is presented in Figure 1.

This is a relaxed oblivious adversary model: the sequence of x_i, y_i, $\hat{y}_{P,i}$, and $w_{P,i}$ can be known to the adversary before generating x_{i+1}, y_{i+1}, but not $\hat{y}_i$. Intuitively, the adversary does not however know the output of the meta-algorithm.

The meta-algorithm M calculates $w_{P,t}$ deterministically based on the observed data and then makes a randomized decision at each time point to refuse or predict. In the following definitions, we consider this randomness as the sole source of randomness within M and compute all the expectations conditioned on the complete data sequence, i.e., $x_t, y_t, w_{P,t} \; \forall \; t$.

Because M makes a randomized decision at each time point, the number of (non-refused) predictions is a random variable and we compute the expected value T^* of this quantity as

$$T^* = \sum_{t=1}^{T} w_{P,t}.$$

We ascribe no loss from refusing to predict, so we define the expected cumulative loss of M as

$$L^*_{P,T} = \sum_{t=1}^{T} w_{P,t} l_{P,t}.$$

Finally, we define the error rate for this randomized meta-algorithm by normalizing the cumulative expected loss via the expected number of non-refused predictions, i.e., $L^*_{P,T}/T^*$.

Our top priority is to guarantee that the error rate of non-refused predictions made by the meta-algorithm does not exceed the target error rate ϵ as the number of predictions increases. Following nomenclature introduced by Tukey [35], see also [26], we call this property the *validity* of the algorithm. Our goal is to satisfy validity without making any assumptions about the data. An asymptotic definition for the validity is given below.

Definition 2.1. *A meta-algorithm M with a given target error ϵ is called* valid *if $T^* = O(1)$ or*

$$\limsup_{T^* \to \infty} \frac{L^*_{P,T}}{T^*} \leq \epsilon.$$

Next, among valid algorithms, we interpret the *efficiency* of an algorithm as the fraction of the predicted data points and define it as follows.

Definition 2.2. *The Efficiency of a meta-algorithm M is denoted by $\rho_T = T^*/T$ and M is called* efficient *if $T - T^* = o(T)$. Furthermore, meta-algorithm M is said to have the* finite refusal property *if $T - T^* = O(1)$.*

Note that $T - T^* = O(1)$ implies that there exists a finite T_0, such that the number of refusals is less than T_0 almost surely. This result directly follows from writing $T - T^*$ as $\sum_t (1 - w_{P,t})$ and applying the Borel-Cantelli lemma directly (see Theorem 3.9 from [36]). By contrast, $T^* = O(1)$ implies that we refuse almost all the time; thus efficiency approaches 0 as $T \to \infty$.

Though the discussion until now has defined validity and efficiency only asymptotically, we will derive explicit bounds on the excess error rate, i.e., $L^*_{P,T}/T^* - \epsilon$. Below, we analyze the efficiency asymptotically for theoretical purposes for arbitrary base predictors. Experiments show that the efficiency is high for predictors that have an error rate below ϵ.

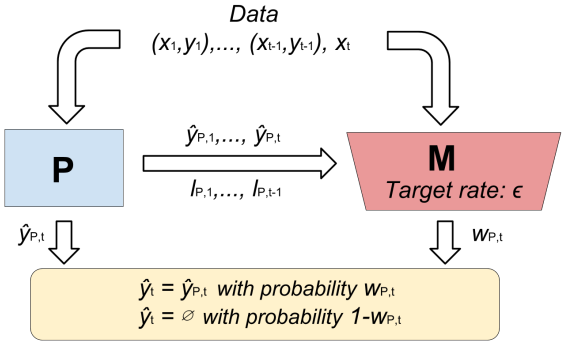


Fig. 1: *The meta-algorithm, represented by M, makes a prediction equivalent to the recommendation of the base predictor P or refuses to do so for data point t while guaranteeing a target rate ϵ.*

For the sake of succinctness in the sequel, we introduce the following notation. First, we compute the variance V^*

of the number of non-refused predictions with respect to the randomness of the meta-algorithm (i.e., $w_{P,t}$) as

$$V^* = \sum_{t=1}^{T} w_{P,t}\left(1 - w_{P,t}\right).$$

Next, we denote the cumulative loss for any predictor P (typically for the base predictor) as

$$L_{P,T} = \sum_{t=1}^{T} l_{P,t}.$$

We also introduce a third sub-index t_0 for any cumulative quantity to represent that the corresponding sum starts from $t_0 + 1$, e.g., for the cumulative loss of P,

$$L_{P,T,t_0} = \sum_{t=t_0+1}^{T} l_{P,t}.$$

2.2 Expert Advice and Exponentially Weighted Average Forecasting (EWAF)

The prediction-with-expert-advice framework pertains to an online prediction scenario in which one has access to a fixed set of experts (predictors). The goal is to combine these experts' predictions such that the difference between the cumulative losses of the combined predictor and the best expert in the set is minimized. This difference is called *regret*, and the algorithms that guarantee a regret that grows sublinearly with the number of predictions are defined as *Hannan consistent*. The appeal of the expert advice algorithms is the fact that the consistency can be guaranteed even in an adversarial setup. For a comprehensive treatment of the expert advice algorithms, we refer the reader to [6].

Exponentially weighted average forecasting (EWAF), also known as "multiplicative weights" or "randomized weighted majority," was first introduced by Littlestone and Warmuth [33] and by Vovk [34] for the expert advice prediction framework. Our meta-algorithm builds on EWAF to provide means to meet any target error rate. Among various expert advice algorithms, we focus on EWAF due to the simplicity of its analysis and strong theoretical guarantees.

Consider N predictors (experts) $P_1, P_2, \ldots, P_N$ each making a prediction $\hat{y}_{P_1,t}, \ldots, \hat{y}_{P_N,t}$ and each suffering a loss $l_{P_1,t}, \ldots, l_{P_N,t}$ at time t. At each time t, EWAF outputs one of these predictions as the combined prediction. To do so, EWAF starts with an initial probability distribution over the experts $(w_{P_1,1}, \ldots, w_{P_N,1})$ and a learning rate $\eta > 0$. At each time point t, this probability distribution is used to choose an expert and consequently, use the prediction of that expert. The probability distribution is updated after the true label and corresponding losses are revealed. By denoting the probability that the predictor P_i is chosen at time t with $w_{P_i,t}$, the prediction $\hat{y}_t$ is obtained by

$$\hat{y}_t = \hat{y}_{P_i,t} \text{ with prob. } w_{P_i,t}, \qquad i = 1, \ldots, N.$$

After the true label y_t is revealed, we update these probabilities by multiplying them by an exponential factor that is scaled with the corresponding loss value and the learning rate, i.e.,

$$w_{P_i,t+1} \quad = \quad \frac{w_{P_i,t} e^{-\eta l_{P_i,t}}}{\sum_{j=1}^{N} w_{P_j,t} e^{-\eta l_{P_j,t}}}. \tag{1}$$

The pseudo-code of the EWAF algorithm is given in Algorithm 1. Given that EWAF randomly selects a predictor according to a known distribution, we can compute the expected loss for the combined prediction at time t as

$$l_t = \sum_{i=1}^{N} w_{P_i,t} l_{P_i,t}$$

We denote the expected cumulative loss of the EWAF as

$$L_T = \sum_{t=1}^{T} l_t$$

and the cumulative loss of expert i, $\forall\, i$ as

$$L_{P_i,T} = \sum_{t=1}^{T} l_{P_i,t}.$$

Algorithm 1 Exp. Weighted Avg. Forecasting (EWAF)
Initial weights: $(w_{P_1,1}, w_{P_2,1}, \ldots, w_{P_N,1})$
Learning rate: η

1: **for** each $t = 1, 2, \ldots$ **do**
2: Follow expert P_i with probability $w_{P_i,t}$.
3: Update the weights :

$$w_{P_i,t+1} = \frac{w_{P_i,t} e^{-\eta l_{P_i,t}}}{\sum_{j=1}^{N} w_{P_j,t} e^{-\eta l_{P_j,t}}}$$

The following well-known theorem establishes an upper bound to the expected loss for EWAF, implying that the regret of EWAF is bounded by $O(\sqrt{T \log N})$.

Theorem 2.1. *(Theorem 2.2 [6]) For any learning rate $\eta > 0$, and the initial weights $(w_{P_1,1}, \ldots, w_{P_N,1})$, we get the following bound for the expected cumulative loss of the EWAF*

$$L_T \quad \leq \quad L_{P_i,T} - \frac{\log w_{P_i,1}}{\eta} + \frac{\eta T}{8} \tag{2}$$

for all $i = 1, \ldots, N$.

If we choose $w_{P_i,1} = 1/N$ for all i and minimize the right-hand-side (RHS) of the bound above with respective to η, we get,

$$L_T \quad \leq \quad L_{P_i,T} + \sqrt{\frac{T \log N}{2}} \tag{3}$$

which is achieved for $\eta = \sqrt{8 \log(N)/T}$.

The bound given in eq. (3) is optimal in the sense that there exists a matching lower bound in the worst case (see Chapter 3.7 [6]). Since the bound on the total sum of the expected loss holds for any predictor, it follows that the bound holds for the predictor with the least cumulative loss. Therefore, the bound in eq. (3) limits how far EWAF is from optimality in terms of the sum of the expected loss.

3 Reliable Prediction with Refusals

We now introduce our meta-algorithm, *SafePredict*, which uses EWAF to convert any given base predictor P into a refusing one with a guaranteed error bound. Furthermore, we provide a theoretical analysis to establish the asymptotic validity and efficiency of SafePredict. Then, we employ the

well-known doubling trick (see for instance Exercise 2.8 [6] or Theorem 7.7 [37]) to choose the learning rate and extend our validity analysis accordingly.

In order to achieve validity, we introduce a trivial predictor which can meet the target error rate by refusing to predict all the time. We refer to this particular predictor as the "dummy predictor" and denote it by D, i.e., $\hat{y}_{D,t} = \varnothing \; \forall \; t$. We also assume that the predictor D suffers a constant loss ϵ for each time t, i.e., $l_{D,t} = \epsilon$ and $L_{D,t} = \epsilon t \; \forall \; t$.

SafePredict is obtained by employing an instance of the EWAF algorithm that runs on the ensemble $\{D, P\}$, to decide either to refuse or predict. Therefore at each step, SafePredict follows the base predictor with probability $w_{P,t}$ and computes the prediction probability for the next round after observing the corresponding loss of P as

$$w_{P,t+1} \quad = \quad \frac{w_{P,t} e^{-\eta l_{P,t}}}{w_{P,t} e^{-\eta l_{P,t}} + w_{D,t} e^{-\eta \epsilon}}, \tag{4}$$

where $w_{D,t} = 1 - w_{P,t}$ stands for the dummy's weight. The pseudo-code for SafePredict is given in Algorithm 2.

Algorithm 2 SafePredict

Base predictor: P; *Initial weight*: $w_{P,1} \in (0,1)$
Learning rate: $\eta > 0$; *Target error rate*: $\epsilon \in (0,1)$

1: **for** each $t = 1, 2, \ldots$ **do**
2: Predict with probability $w_{P,t}$, refuse otherwise, i.e.,

$$\hat{y}_t = \begin{cases} \hat{y}_{P,t} & \text{with prob. } w_{P,t} \\ \varnothing & \text{otherwise} \end{cases}$$

3: Update the prediction probability:

$$w_{P,t+1} \quad = \quad \frac{w_{P,t} e^{-\eta l_{P,t}}}{w_{P,t} e^{-\eta l_{P,t}} + w_{D,t} e^{-\eta \epsilon}},$$

Note that Algorithm 2 requires no assumptions, e.g., i.i.d. assumptions, about the input data. The algorithm takes *only* the loss values as input.

3.1 Validity

To bound the average loss of SafePredict one might be inclined to simply apply the bound presented in Theorem 2.1 with $N = 2$, $P_1 = P$, $P_2 = D$, and the optimal learning rate to minimize the RHS of eq. (2) for $i = 2$, which results in

$$\frac{L^*_{P,T}}{T^*} \quad \leq \quad \epsilon + \frac{1}{T^*} \sqrt{\frac{T \log(1/w_{D,1})}{2}}. \tag{5}$$

The problem with the bound in eq. (5) is that if the prediction probability decreases too quickly (i.e., for $T^* \ll \sqrt{T}$) the rightmost term can easily become vacuous, i.e., the excess error does not vanish. In particular, the bound is not sufficient to guarantee the validity of the algorithm for the case in which T^* the number of predictions made approaches infinity slower than $\sqrt{T}$ where T is the number of data points presented, i.e., $T^* = \omega(1)$ and $T^* = O\left(\sqrt{T}\right)$.

Further, even though the bound given in eq. (3) is tight in the worst case (i.e., adversarial scenario), [38] introduces second-order adaptive techniques that perform better when the environment is more benign, e.g., pseudo-stationary. In particular, [38] provides variance-based bounds with data-dependent learning rates (see Corollary 3.5.1 below). In the following, we develop a simpler bound based on the techniques introduced in [39] to simplify our analysis in Section 4. The resulting bound has the same order of magnitude that we would have obtained using the methods of [38].

Particularly, in the following theorem, we present a refined bound for SafePredict that suggests a learning rate that decreases with the variance of the number of predictions, i.e., V^*, and guarantees the validity of our algorithm.

Theorem 3.1. *For any P, $\eta > 0$, $\epsilon < 1/2$, and $0 < w_{P,1} < 1$, SafePredict satisfies*

$$\frac{L^*_{P,T}}{T^*} \quad \leq \quad \epsilon - \frac{\log(w_{D,1})}{\eta T^*} + \frac{(1-\epsilon)^2 \eta V^*}{T^*}. \tag{6}$$

Consequently, by choosing the learning rate η to minimize the RHS of this bound, we get

$$\frac{L^*_{P,T}}{T^*} \quad \leq \quad \epsilon + (1-\epsilon) \frac{2\sqrt{\log(1/w_{D,1})\, V^*}}{T^*} \tag{7}$$

for $\eta^* = \frac{\sqrt{\log(1/w_{D,1})/V^*}}{1-\epsilon}$.

Before proving the statement given in the theorem, we define auxiliary quantities called "mix-loss" and "mixability gap", and present two important results from [39] about EWAF in terms of these quantities.

In the expert advice framework of Section 2.1, the *mix-loss* m_t is an alternative way of averaging the expert losses $l_{P_i,t}, \; \forall \; i$ using the weights $w_{P_i,t}, \; \forall \; i$. Formally, *mix-loss* is defined as

$$m_t = -\frac{1}{\eta} \log \left(\sum_{i=1}^{N} w_{P_i,t} e^{-\eta l_{P_i,t}} \right)$$

or equivalently $e^{-\eta m_t} = \sum_{i=1}^{N} w_{P_i,t} e^{-\eta l_{P_i,t}}$. Additionally, we denote the cumulative mix-loss as $M_T = \sum_{t=1}^{T} m_t$.

One can bound the cumulative mix-loss in terms of the losses of the individual experts by the following lemma.

Lemma 3.2. *([39], Lemma 1) For the learning rate $\eta > 0$, and the initial weights $(w_{P_1,1}, \ldots, w_{P_N,1})$, the cumulative mix-loss of the EWAF satisfies the following inequality*

$$M_T \quad \leq \quad L_{P_i,T} - \frac{\log w_{P_i,1}}{\eta} \tag{8}$$

for all $i = 1, \ldots, N$.

The bound on the cumulative mix-loss can be used to bound the cumulative expected loss by defining the *mixability gap* as the difference between these two types of averages and bounding them with classical concentration inequalities. In particular, the mixability gap, $\delta_t = l_t - m_t$, can be bounded using the following lemma.

Lemma 3.3. *([39], Lemma 4) The difference between the expected and mix losses obtained by EWAF algorithm is bounded by*

$$\delta_t = l_t - m_t \leq \eta \sum_i w_{P_i,t} \left(l_t - l_{P_i,t} \right)^2$$

for all t.

Armed with these two lemmas, we can continue with the proof of Theorem 3.1

Proof of Theorem 3.1. By definition of δ_t we have

$$L_T \quad = \quad M_T + \sum_{t=1}^{T} \delta_t.$$

Then by applying Lemma 3.2. with $P_i = D$, we get

$$\sum_{t=1}^{T} l_t \quad \leq \quad \epsilon T - \frac{\log(w_{D,1})}{\eta} + \sum_{t=1}^{T} \delta_t. \tag{9}$$

Next, we plug in the definition of the expected loss $l_t = w_{P,t} l_{P,t} + w_{D,t}\epsilon$, and divide both sides by the expected number of predictions,

$$\begin{aligned}
\sum_{t=1}^{T} w_{P,t} l_{P,t} &\leq \sum_{t=1}^{T} (1 - w_{D,t})\epsilon - \frac{\log(w_{D,1})}{\eta} + \sum_{t=1}^{T} \delta_t \\
L^*_{P,T} &\leq \epsilon T^* - \frac{\log(w_{D,1})}{\eta} + \sum_{t=1}^{T} \delta_t \\
\frac{L^*_{P,T}}{T^*} &\leq \epsilon - \frac{\log(w_{D,1})}{\eta T^*} + \frac{1}{T^*} \sum_{t=1}^{T} \delta_t. \qquad (10)
\end{aligned}$$

Finally, we obtain the desired result by bounding δ_t:

$$\begin{aligned}
\delta_t &\leq \eta \left(w_{P,t} (l_t - l_{P,t})^2 + w_{D,t} (l_t - \epsilon)^2 \right) && (11) \\
&= \eta \left(w_{P,t} w_{D,t}^2 + w_{D,t} w_{P,t}^2 \right) (l_{P,t} - \epsilon)^2 && (12) \\
&= \eta w_{P,t} w_{D,t} (l_{P,t} - \epsilon)^2 && (13) \\
&\leq \eta w_{P,t} w_{D,t} (1 - \epsilon)^2 && (14)
\end{aligned}$$

(11) follows from Lemma 3.3, while (12) and (13) follow from the definition of l_t and $w_{D,t} + w_{P,t} = 1$, respectively.

Then, by summing up both sides of eq. (14) over t, we get

$$\sum_{t=1}^{T} \delta_t \quad \leq \quad \frac{\eta V^*}{(1-\epsilon)^2}. \tag{15}$$

https://www.overleaf.com/project/5bcc82f8278cd86e7cfd27d3

The desired result for the first part of the theorem, eq. (6), follows by plugging eq. (15) into eq. (10). Finally, the learning rate η that minimizes the RHS of eq. (6) can be found by basic calculus, and by plugging in the optimal learning rate we obtain eq. (7). □

Note that the essential difference between the bounds in eq. (5) and (7) lies in the choice of the learning rate η. While the optimal learning rate for the conventional use of EWAF is on the order of $1/\sqrt{T}$, Theorem 3.1 suggests choosing η to be on the order of $1/\sqrt{V^*}$, which decreases much slower than $1/\sqrt{T}$, and leads to a sufficient condition for the validity of our algorithm.

Theorem 3.1 implies the excess error rate decreases with a rate $\sqrt{V^*}/T^*$, i.e., $L^*_{P,t}/T^* - \epsilon = O\left(\sqrt{V^*}/T^*\right)$. Since V^* is always less than or equal to T^*, this rate is bounded by

$$\frac{1}{T^*} \leq \frac{\sqrt{V^*}}{T^*} \leq \frac{1}{\sqrt{T^*}}.$$

Therefore our result implies that as more predictions are made (i.e., T^* increases) the bound becomes tighter and guarantees the validity of SafePredict algorithm.

Corollary 3.3.1. *For any P, $\epsilon < 1/2$, $0 < w_{P,1} < 1$, if one choose the learning rate η in the order of $1/\sqrt{V^*}$, i.e., $\eta = \Theta\left(1/\sqrt{V^*}\right)$, SafePredict is guaranteed to be valid.*

Unfortunately, selecting a learning rate η that depends on V^* is infeasible because V^* is unknown a priori. In Section 3.3 we describe a practical method for choosing a learning rate without knowledge of the expected number of refusals while still satisfying eq. (7) up to a constant factor. Having shown the validity of our algorithm, we now move to the efficiency.

3.2 Efficiency

In this section, we study the efficiency of the SafePredict meta-algorithm given in Algorithm 2. In contrast with validity which we addressed without requiring any assumptions on the base predictor, we must characterize the efficiency of a given algorithm with respect to the performance of the base predictor P. We show that SafePredict leads to efficient predictions if the base predictor P has an asymptotic error rate smaller than the target error rate ϵ, i.e., $\lim_{t\to\infty} L_{P,t}/t < \epsilon$.

The following lemma lower and upper bounds the probability of making a prediction at time $t+1$, i.e., $w_{P,t+1}$, in terms of the cumulative loss of the base predictor P.

Lemma 3.4. *The probability of making a prediction at time $t+1$ for SafePredict, namely $w_{P,t+1}$, satisfies the following*

$$1 - \frac{w_{D,1}}{w_{P,1}} e^{\eta(L_{P,t} - \epsilon t)} \leq w_{P,t+1} \leq \frac{w_{P,1}}{w_{D,1}} e^{\eta(\epsilon t - L_{P,t})}. \tag{16}$$

Proof. To prove the upper bound in eq. (16) we first note that the update rule given in eq. (4) can be written in terms of the mix-loss, and by induction we can obtain the prediction probability at time $t+1$ in terms of the cumulative losses M_t and $L_{P,t}$ (see (17) and (18) below), i.e.,

$$\begin{aligned}
w_{P,t+1} &= \frac{w_{P,t} e^{-\eta l_{P,t}}}{e^{-\eta m_t}} = w_{P,t} e^{\eta(m_t - l_{P,t})} && (17) \\
&= w_{P,1} e^{\eta(M_t - L_{P,t})} && (18) \\
&\leq w_{P,1} e^{\eta(L_{D,t} - \log(w_{D,1})/\eta - L_{P,t})} && (19) \\
&\leq \frac{w_{P,1}}{w_{D,1}} e^{\eta(\epsilon t - L_{P,t})}. && (20)
\end{aligned}$$

Next, (19) and (20) follows by applying Lemma 3.2 with $P_i = D$ to bound the cumulative mix loss in terms of the cumulative loss of the dummy, $L_{D,t} = \epsilon t$.

For the lower bound, we apply the same argument for $w_{D,t+1} = 1 - w_{P,t+1}$. □

Note that Lemma 3.4 implies that the probability of making a prediction decreases exponentially fast if the cumulative loss of the base predictor is increasing faster than the target error rate ϵ. Next, we exploit this fact to show that if the base predictor satisfies the desired error requirement (i.e., its average loss is already below ϵ), our algorithm achieves a finite expected number of refusals.

Theorem 3.5. *If $\limsup L_{P,T}/T = \epsilon' < \epsilon$ and $\eta T \to \infty$ in the limit $T \to \infty$, then the expected number of refusals made by SafePredict is finite, i.e.,*

$$\lim_{T\to\infty} T - T^* \quad = \quad \sum_{t=1}^{\infty} w_{D,t} \quad < \quad \infty. \tag{21}$$

Proof. Since $\epsilon' < \epsilon$, there exists a $t_0 < \infty$ such that for all $t > t_0$:

$$\frac{L_{P,t}}{t} \quad \leq \quad \epsilon' + \frac{\epsilon - \epsilon'}{2}. \tag{22}$$

Then,

$$\begin{aligned}
\sum_{t=1}^{\infty} w_{D,t} \;&=\; \sum_{t=1}^{t_0} w_{D,t} + \sum_{t=t_0+1}^{\infty} w_{D,t} && (23)\\
&\leq\; t_0 + \sum_{t=t_0+1}^{\infty} (1 - w_{P,t}) && (24)\\
&\leq t_0 + \frac{1 - w_{P,1}}{w_{P,1}} \sum_{t=t_0+1}^{\infty} e^{\eta t (L_{P,t}/t - \epsilon)} && (25)\\
&\leq t_0 + \lim_{t' \to \infty} \frac{w_{D,1}}{w_{P,1}} \sum_{t=t_0+1}^{t'} e^{\eta t (\epsilon' - \epsilon)/2} && (26)\\
&\leq t_0 + \lim_{t' \to \infty} \frac{w_{D,1}}{w_{P,1}} \frac{e^{\eta (t_0+1)(\epsilon' - \epsilon)/2} - e^{\eta t' (\epsilon' - \epsilon)/2}}{1 - e^{\eta (\epsilon' - \epsilon)/2}} && (27)\\
&\leq t_0 + \frac{w_{D,1}}{w_{P,1}} \frac{e^{\eta (t_0+1)(\epsilon' - \epsilon)/2}}{1 - e^{\eta (\epsilon' - \epsilon)/2}} \;<\; \infty, && (28)
\end{aligned}$$

where (24) follows from the fact that $w_{D,t} \leq 1$, (25) follows from Lemma 3.4, (26) follows from eq. (22), (27) follows from the sum of a geometric series and from the fact $e^{\eta(\epsilon' - \epsilon)/2} < 1$, finally (28) follows from the hypothesis $\eta T \to \infty$. □

Theorem 3.5 shows that when the base predictor is already valid, SafePredict is not only efficient but also satisfies the *finite refusal property*. Furthermore, as mentioned in Section 2, the finite refusal property implies a finite number of refusals almost surely beyond the expected value.

Note that the finite refusal property depends on the assumption that the predictor P can (eventually) predict accurately enough to obtain a valid algorithm. If the predictor is highly inaccurate, i.e., $l_{P,t} > \epsilon$ with high probability, then the SafePredict meta-algorithm will refuse almost all the time, as it should, and the hypothesis $\eta T \to \infty$. The meta-algorithm does not need to know in advance how well the base predictor will behave to achieve this desirable outcome. Further, we can make SafePredict adaptive when the base predictor sometimes is valid and sometimes is not, as we show in Section 4.

3.3 Choosing the Learning Rate

Corollary 3.3.1 and 3.5.1 establish the validity and efficiency of SafePredict once the learning rate η is chosen to be on the order of $1/\sqrt{V^*}$, but V^* cannot be known a priori. One classical method of addressing the issue of estimating unknown quantities in an online scenario is known as the *doubling trick* (cf. Chapter 2.3 of [6] or Theorem 7.7 of [37]). Sophisticated methods to choose η include the approach of Theorem 5 from [38] to provide the following validity bound:

Corollary 3.5.1 (to Theorem 5 of [38]). *For any P, $\epsilon < 1/2$, and $0 < w_{P,1} < 1$, SafePredict (Alg. 2) with a data dependent learning rate*

$$\eta_t = min\left\{\frac{1}{2}, \frac{0.894}{\sqrt{\sum_{i=1}^{t-1} (l_{P,i} - \epsilon)^2 w_{P,i} w_{D,i}}}\right\},$$

satisfies the following validity bound:

$$\frac{L^*_{P,T}}{T^*} \quad \leq \quad \epsilon + \frac{25}{T^*} + 4\frac{\sqrt{V^*}}{T^*}. \tag{29}$$

To keep the analysis in Section 4 simple, we nevertheless apply the doubling trick to our problem and derive a validity bound having the same order of magnitude as Corollary 3.5.1.

In an iterative fashion, the doubling trick starts with an initial estimate of the unknown quantity and compares it with the observed value of this quantity at each time step. When the measured value exceeds the estimate, the estimate is doubled and the algorithm is reset.

Procedurally, to estimate V^* for a fixed horizon T, start with an initial estimate of V_{est} (typically $V_{est} = 1$) and a running sum $V_{sum} = 0$. Then invoke Theorem 3.1. by replacing V^* with $V_{est} = 1$, i.e., $\eta = \sqrt{\log(1/w_{D,1})}/(1 - \epsilon)$ and run Algorithm 2. At each time t update V_{sum} by incrementing it by $w_{P,t+1} w_{D,t+1}$ and check if it exceeds the current estimate. If it does, do the following:

1) Double the estimated value $V_{est} \leftarrow 2V_{est}$.
2) Update the learning rate according to the new V_{est}, i.e., $\eta \leftarrow \eta/\sqrt{2}$.
3) Reset the prediction probability $w_{P,t+1} \leftarrow w_{P,1}$.
4) Reset the running sum $V_{sum} \leftarrow 0$.

The complete pseudo-code for the modified algorithm is given in Algorithm 3.

Algorithm 3 SafePredict with Doubling Trick

Base predictor: P; *Initial weight*: $w_{P,1} \in (0, 1)$
Target error rate: $\epsilon \in (0, 1)$

Initialize $t = 1$
for each $k = 1, 2, \ldots$ **do**
  Reset $w_{P,t} = w_{P,1}$, $V_{sum} = 0$, and

$$\eta = \sqrt{\log(1/w_{D,1})/(1 - \epsilon)^2 / 2^k}$$

  while $V_{sum} \leq 2^k$ **do**
    Predict with probability $w_{P,t}$, refuse otherwise,

$$\hat{y}_t = \begin{cases} \hat{y}_{P,t} & \text{with prob. } w_{P,t} \\ \varnothing & \text{otherwise} \end{cases}$$

    Update the prediction probability:

$$w_{P,t+1} \quad = \quad \frac{w_{P,t} e^{-\eta l_{P,t}}}{w_{P,t} e^{-\eta l_{P,t}} + w_{D,t} e^{-\eta \epsilon}},$$

    Compute $V_{sum} \leftarrow V_{sum} + w_{P,t+1} w_{D,t+1}$
    Increment t by 1, i.e., $t \leftarrow t + 1$

Theorem 3.6. *For any P, $\epsilon < 1/2$, and $0 < w_{P,1} < 1$, SafePredict with the doubling trick given in Algorithm 3 satisfies*

$$\frac{L^*_{P,T}}{T^*} \quad \leq \quad \epsilon + (1 - \epsilon) \frac{2\sqrt{2}}{\sqrt{2} - 1} \frac{\sqrt{\log(1/w_{D,1})\, V^*}}{T^*}. \tag{30}$$

Proof. Denote the time for the K^{th} reset of the algorithm with T_K, i.e., T_K is the largest τ such that $\sum_{t=1}^{\tau} w_{P,t} w_{D,t} \leq 2^K$. Additionally, assume $T_0 = 0$ and K^* is the integer that satisfies $T_{K^*-1} < T \leq T_{K^*}$.

Then, we can rewrite the sum

$$\begin{aligned} L^*_{P,T} - \epsilon T^* &= \sum_{t=1}^{T} w_{P,t} \left(l_{P,t} - \epsilon\right) \\ &\leq \sum_{K=1}^{K^*} \sum_{t=T_{K-1}+1}^{T_K} w_{P,t} \left(l_{P,t} - \epsilon\right). \end{aligned} \quad (31)$$

Next, we can bound each summand in the RHS using eq. (7) from Theorem 3.1, i.e., for all K we plug the estimated $V_{est} = 2^K$ value instead of V^* in the learning rate and the corresponding bound,

$$\sum_{t=T_{K-1}+1}^{T_K} w_{P,t} \left(l_{P,t} - \epsilon\right) \leq \ (1-\epsilon)\, 2\sqrt{2^K \log\left(1/w_{D,1}\right)}. \quad (32)$$

Combining eq. (31) and (32), we obtain

$$\begin{aligned} L^*_{P,T} - \epsilon T^* &\leq \ (1-\epsilon)\, 2 \sum_{K=1}^{K^*} \sqrt{2^K \log\left(1/w_{D,1}\right)} \\ &= \ (1-\epsilon)\, 2\sqrt{2^{K^*} \log\left(1/w_{D,1}\right)} \sum_{K=1}^{K^*} \sqrt{2^{K-K^*}} \\ &\leq \ (1-\epsilon)\, \frac{2\sqrt{2}}{\sqrt{2}-1} \sqrt{2^{K^*} \log\left(1/w_{D,1}\right)} \quad (33) \\ &\leq \ (1-\epsilon)\, \frac{2\sqrt{2}}{\sqrt{2}-1} \sqrt{\log\left(1/w_{D,1}\right) V^*} \quad (34) \end{aligned}$$

where (33) follows from the sum of a geometric series and (34) follows from the definition of K^*.

Finally, the desired result is obtained by dividing both sides by T^*. □

Both Corollary 3.5.1 and Theorem 3.6 suggest methods to bound the order of excess error rate to the same level ($\sqrt{V^*}/T^*$) as stated in Theorem 3.1 where we assumed the use of an optimal learning rate value for η^*.

Additionally, even though we decrease the learning rate η as T increases, it is always greater than $\eta^*/2$ of Theorem 3.1 and satisfies $\eta T \to \infty$. Thus, Theorem 3.5 holds for Algorithm 3 using the same arguments in the proof. We omit the details of the proof due to the space restrictions.

4 Adaptive SafePredict

We now consider the practical scenario in which the error rate of the base predictor changes over time. Our meta-algorithm should reduce the probability of making a prediction when the base predictor suffers a large error and predict more often when the base predictor does well. Inspired by the Fixed Share algorithm [40], we introduce Adaptive SafePredict, a weight-shifting extension to SafePredict. Adaptive SafePredict tracks changes in the error rate of the base predictor while preserving validity, thus improving efficiency even when there are periods of poor predictions.

The base predictor may endure spikes in its error rate for a variety of reasons. For example, most predictors have a high error rate at the beginning of the prediction task; their error rate decreases as they see more examples and learn from the mistakes. Sometimes, the underlying data distribution may abruptly change, and thus the performance of the base predictor degrades significantly until the predictor learns the new data distribution. Such scenarios might lead to long sequences of bad predictions for the base algorithm P, and force the prediction probability to tend to zero. To ensure that the prediction probability does not decrease too quickly due to a long sequence of bad predictions, we shift a small portion of the dummy's "weight" (refusal probability) towards the base predictor. This weight shift allows Adaptive SafePredict to recover quickly when the base predictor performs well again.

Proposals to make generic EWAF adaptive against such changing environments include Chapter 5.2 of [6]. A popular way among these techniques is called "sharing the weights" which leads to the *fixed share algorithm* [40]. The fixed share algorithm simply adds a mixing step to the weight update rule in the EWAF. In particular, following the notation from Section 2.1, the fixed share update rule becomes

$$w_{P_i,t+1} = \frac{\alpha}{N} + (1-\alpha)\, \frac{w_{P_i,t} e^{-\eta l_{i,t}}}{\sum_{j=1}^{N} w_{P_j,t} e^{-\eta l_{j,t}}}$$

for some $\alpha \in [0,1)$. The mixing step ensures that no weight falls below a predefined value α/N, and guarantees a sub-linear regret against roughly αT abrupt changes in the underlying statistics (e.g., the error rate). For a detailed analysis of the fixed share algorithm, please see [40], [41].

We apply a similar idea to SafePredict by modifying the EWAF update rule eq. (4) to guarantee that the prediction probabilities do not get too small or too large. The net result is to track the changes in the error rate efficiently while preserving the validity guarantees established in Section 3.1. To constrain the prediction probability at time $t+1$ to lie within an arbitrary interval $\alpha_t \leq w_{P,t+1} \leq \beta_t$, we modify our update rule as follows

$$w_{P,t+1} \ = \ \alpha_t + (\beta_t - \alpha_t)\, \frac{w_{P,t} e^{-\eta l_{P,t}}}{w_{P,t} e^{-\eta l_{P,t}} + w_{D,t} e^{-\eta \epsilon}}. \quad (35)$$

We call α_t and β_t the *adaptivity parameters* and let them change with time for the sake of generality. In Algorithm 4, we give the pseudo-code for Adaptive SafePredict with adaptivity parameters α_t and β_t.

A good setting for the adaptivity parameters provides resilience against changes while preserving validity. From our analysis in Section 3.1, we observe that validity is preserved as long as α_t is on the order of $1/T$, regardless of the choice of β_t. By noting $w_{P,t+1}$ is an increasing function of β_t, we choose $\beta_t = 1$ to maximize the efficiency of this adaptive algorithm (Alg. 4) under this restriction on α_t. In particular, we derive an upper bound to the probability of refusal that depends only on the loss sequence of P starting from an arbitrary time point t_0, i.e., L_{P,t,t_0} instead of $L_{P,t}$. As a result, efficiency increases whenever P starts to make better predictions after t_0.

4.1 Validity

To quantify the effect of the adaptivity parameters α_t, β_t on our validity guarantees, we first show that Adaptive

Algorithm 4 Adaptive SafePredict
Base predictor: P; *Initial weight*: $w_{P,1} \in (0,1)$
Learning rate: $\eta > 0$; *Target error rate*: $\epsilon \in (0,1)$
Min. prediction prob.: $\alpha_1, \ldots, \alpha_T \in [0,1)$
Max. prediction prob.: $\beta_1, \ldots, \beta_T \in (0,1]$

1: **for** each $t = 1, 2, \ldots$ **do**
2: Predict with probability $w_{P,t}$, refuse otherwise, i.e.,

$$\hat{y}_t = \begin{cases} \hat{y}_{P,t} & \text{with prob. } w_{P,t} \\ \varnothing & \text{otherwise} \end{cases}$$

3: Update the prediction probability:

$$w_{P,t+1} \;=\; \alpha_t + (\beta_t - \alpha_t)\,\frac{w_{P,t}e^{-\eta l_{P,t}}}{w_{P,t}e^{-\eta l_{P,t}} + w_{D,t}e^{-\eta\epsilon}}$$

SafePredict is equivalent to the usual EWAF with a larger set of experts. In particular, we consider a virtual ensemble, where each expert in the ensemble chooses to follow either the dummy predictor D, or the base predictor P. In the following lemma, we show that for an appropriately chosen set of initial weights, the EWAF on this virtual ensemble (i.e., Algorithm 1) becomes equivalent to Adaptive SafePredict as outlined in Algorithm 4. Next, we use this equivalence to extend our analysis from Section 3.

Lemma 4.1. *(Equivalence lemma) Suppose we have a base predictor P. Consider an ensemble of 2^T experts, $\mathcal{P} = \{P_0, P_1, \ldots, P_{2^T-1}\}$, defined as follows:*

- *Denote the t^{th} bit of the binary expansion of integer i with $b_{i,t}$, and define the notation $\bar{x} = 1 - x$.*
- *Fix the predictions and the losses of each expert P_i as follows:*

$$\hat{y}_{P_i,t} = \begin{cases} \varnothing & b_{i,t} = 0 \\ \hat{y}_{P,t} & b_{i,t} = 1 \end{cases} \text{ and } l_{P_i,t} = \begin{cases} \epsilon & b_{i,t} = 0 \\ l_{P,t} & b_{i,t} = 1 \end{cases}.$$

- *Set the initial weights for each expert as*

$$w_{P_i,1} = w_{P,1}^{b_{i,1}} w_{D,1}^{\bar{b}_{i,1}} \cdots \prod_{t=1}^{T-1} \alpha_t^{\bar{b}_{i,t}b_{i,t+1}} \bar{\alpha}_t^{\bar{b}_{i,t}\bar{b}_{i,t+1}} \beta_t^{b_{i,t}b_{i,t+1}} \bar{\beta}_t^{b_{i,t}\bar{b}_{i,t+1}}.$$

Then the EWAF algorithm (Alg. 1) using the expert ensemble $\mathcal{P}$ with the learning rate η is equivalent to Adaptive SafePredict (Alg. 4) using the base predictor P, in terms of the prediction probability

$$w_{P,t} = \sum_{i:b_{i,t}=1} w_{P_i,t}.$$

Proof. The proof is in the Supplementary Material. □

Because Lemma 4.1 reduces the adaptive algorithm to an instance of EWAF, we can obtain the following validity guarantee by modifying the proof of Theorem 3.1.

Corollary 4.1.1. *For any P, $\eta > 0$, $\epsilon < 1/2$, $0 < w_{P,1} < 1$, and $0 \le \alpha_t, \beta_t, \le 1$, $\forall t$ Adaptive SafePredict meta-algorithm given in Algorithm 4 satisfies*

$$\frac{L^*_{P,T}}{T^*} \;\le\; \epsilon - \frac{\log(w_{D,1}\Delta_T)}{\eta T^*} + \frac{(1-\epsilon)^2\,\eta V^*}{T^*}, \quad (36)$$

where Δ_T is defined as $\prod_{t=1}^{T-1}(1-\alpha_t)$.

By choosing the learning rate η to minimize the RHS of this bound, we get

$$\frac{L^*_{P,T}}{T^*} \;\le\; \epsilon + (1-\epsilon)\,\frac{2\sqrt{\log\left(1/(w_{D,1}\Delta_T)\right)V^*}}{T^*} \quad (37)$$

for $\eta^* = \frac{\sqrt{\log(1/(w_{D,1}\Delta_T))/V^*}}{1-\epsilon}$.

Proof. The proof of the corollary follows from the same steps given in the proof of Theorem 3.1, except instead of choosing $P_i = D$ while applying Lemma 3.2 in eq. (9), we choose $P_i = P_0$ from the equivalent virtual ensemble given in Lemma 4.1. Note that P_0 always refuses and is essentially identical to D, except its initial weight is

$$w_{P_0,1} \;=\; w_{D,1}\prod_{t=1}^{T-1}(1-\alpha_t) \;=\; w_{D,1}\Delta_T. \qquad \square$$

Corollary 4.1.2. *For $\alpha_t = \alpha < 1/2\ \forall t$, the validity bound given in eq. (37) becomes*

$$\frac{L^*_{P,T}}{T^*} \;\le\; \epsilon + (1-\epsilon)\,\frac{2\sqrt{V^*\left(\log\left(1/w_{D,1}\right) + T\alpha + T\alpha^2\right)}}{T^*}.$$

Thus setting $\alpha = O(1/T)$ is a sufficient condition to guarantee the validity of Adaptive SafePredict given in Alg. 4.

Proof. Proof directly follows from setting $\alpha_t = \alpha$ in eq. (37) and using Taylor series expansion of Δ_T,

$$\log(\Delta_T) \;=\; (T-1)\log(1-\alpha) \;\ge\; -T\left(\alpha + \alpha^2\right). \qquad \square$$

Corollary 4.1.2 implies that by choosing $\alpha_t = \alpha = O(1/T)$, we can preserve the same convergence rate for the excess error rate from Theorem 3.1, i.e., $L^*_{P,T}/T^* - \epsilon = O\left(\sqrt{V^*}/T^*\right)$. In other words, as long as α is small, the whole effect of the weight shifting on our validity bound can be interpreted as starting with a smaller initial refusal probability, by reducing $w_{D,1}$ by a multiplicative factor of $\Delta_T \approx e^{\alpha T}$ which is essentially a constant for $\alpha = O(1/T)$.

Furthermore, note that the bound given in eq. (37) depends only on α_t and is totally agnostic to the choice of β_t. Therefore, once α_t values are chosen to preserve the validity, i.e., on the order of $1/T$, we are free to choose β_t to maximize the prediction probability, and therefore efficiency, by choosing $\beta_t = 1$ at all time points t. In the next subsection, we analyze the efficiency of this special case and argue it provides resilience against changes in the data distribution.

4.2 Weight-Shifting SafePredict

In this section, we exploit a special case of Adaptive SafePredict to show that it provides resilience against changes in the data distribution while preserving the validity of the algorithm. This special case, obtained for $\alpha_t = \alpha = \Theta(1/T)$ and $\beta_t = 1$, is called *Weight-Shifting SafePredict*.

As motivated by Corollary 4.1.2, we first set α_t for all t equal to some constant α, where α is on the order of $1/T$. This will guarantee validity. Next, we choose to maximize $w_{P,t+1}$ over β_t in order to maximize the efficiency. Therefore, from eq. (35) we obtain $\beta_t = 1,\ \forall\ t$. Note that this particular choice of adaptivity parameters simplifies the update rule given in eq. (35) to

$$w_{P,t+1} \;=\; \alpha + (1-\alpha)\,\frac{w_{P,t}e^{-\eta l_{P,t}}}{w_{P,t}e^{-\eta l_{P,t}} + w_{D,t}e^{-\eta\epsilon}}. \quad (38)$$

An intuitive way of looking at this rule is that at each time point we use the EWAF rule first and then shift an α portion of the weight of the dummy to towards the base predictor P, thus performing *weight shifting*.

The following result implies an exponentially diminishing refusal probability in terms of the partial cumulative loss L_{P,t,t_0}, for an arbitrary $t_0 < t$, which implies that Weight-Shifting SafePredict can quickly recover to make predictions if the base predictor performs well starting time t_0.

Lemma 4.2. *The probability of refusing to predict at time $t+1$, $w_{D,t+1}$, by using the Weight-Shifting SafePredict (Algorithm 4 with $\beta_t = 1$ and $\alpha_t = \alpha$, $\forall$ t) satisfies the following inequality*

$$w_{D,t+1} \leq \frac{1-\alpha}{\alpha} e^{\eta\left(L_{P,t,t_0} - \epsilon'(t-t_0)\right)} \tag{39}$$

for $\epsilon' = \epsilon + \alpha/\eta$.

Proof. First, write the update rule eq. (38) in terms of the probability of refusal by noting $w_{D,t} = 1 - w_{P,t}$

$$\begin{aligned} w_{D,t+1} &= w_{D,t}\,(1-\alpha)\,\frac{e^{-\eta\epsilon}}{w_{P,t}e^{-\eta l_{P,t}} + w_{D,t}e^{-\eta\epsilon}} \\ &= w_{D,t}\,(1-\alpha)\,e^{\eta(m_t - \epsilon)} && (40) \\ &= w_{D,t_0+1}\,(1-\alpha)^{t-t_0}\,e^{\eta\left(\sum_{\tau=t_0+1}^{t} m_\tau - \epsilon(t-t_0)\right)} && (41) \\ &\leq w_{D,t_0+1} e^{\eta\left(\sum_{\tau=t_0+1}^{t} m_\tau - \epsilon'(t-t_0)\right)}. && (42) \end{aligned}$$

Where (40) follows by replacing the denominator with the definition of mix-loss from Section 3.1, (41) follows by recursing this update rule $t - t_0$ time, and the final step follows from the inequality $1 - \alpha \leq e^{-\alpha}$ for $0 \leq \alpha \leq 1$ and $\epsilon' = \epsilon + \alpha/\eta$.

By Lemma 4.1, the mix-loss suffered by Algorithm 4 is equal to the one suffered by the EWAF over the virtual ensemble described in the the lemma. Note that we can consider all the virtual experts that follow P from time t_0+1 to t as a single super-expert since they suffer the same loss sequence, namely $l_{P,t_0+1}, \ldots, l_{P,t}$, within this interval. By denoting this super expert as Q, we can compute its total weight at time $t_0 + 1$ as

$$w_{Q,t_0+1} = \sum_{i:b_{i,t_0+1}=1} w_{P_i,t_0+1} \;=\; w_{P,t_0+1},$$

where equality to w_{P,t_0+1} again follows from the Lemma 4.1. Then we can bound the sum in the eq. (42) using Lemma 3.2 for virtual expert Q,

$$\begin{aligned} \sum_{\tau=t_0+1}^{t} m_\tau &\leq L_{Q,t,t_0} - \frac{\log(w_{Q,t_0+1})}{\eta} \\ &= L_{P,t,t_0} - \frac{\log(w_{P,t_0+1})}{\eta}. && (43) \end{aligned}$$

Finally, we conclude the proof by employing eq. (43) in eq. (42) and noting $w_{P,t_0+1} \geq \alpha$ and $w_{D,t_0+1} \leq 1 - \alpha$, i.e.,

$$\begin{aligned} w_{D,t+1} &\leq w_{D,t_0+1} e^{\eta\left(L_{P,t,t_0} - \epsilon'(t-t_0)\right) - \log(w_{P,t_0+1})} \\ &= \frac{w_{D,t_0+1}}{w_{P,t_0+1}} e^{\eta\left(L_{P,t,t_0} - \epsilon'(t-t_0)\right)} \\ &\leq \frac{1-\alpha}{\alpha} e^{\eta\left(L_{P,t,t_0} - \epsilon'(t-t_0)\right)}. \end{aligned}$$

□

We note that the dominant term of the RHS of eq. (39) is the exponential term since the preceding term $(1-\alpha)/\alpha$ for $\alpha = \Theta(1/T)$ increases only linearly with T. Therefore, Lemma 4.2 implies that if P starts to do well at time t_0, i.e., if its error rate starting from t_0 becomes less than the target rate ϵ, the prediction probability will increase to 1 exponentially fast.

Algorithm 5 Weight-Shifting SafePredict with Doub. Trick
Base predictor: P; *Initial weight*: $w_{P,1} \in (0,1)$
Target error rate: $\epsilon \in (0,1)$; *Adaptivitiy Parameter*: $\alpha \in [0,1)$

Initialize $t = 1$
for each $k = 1, 2, \ldots$ **do**
  Reset $w_{P,t} = w_{P,1}$, $V_{sum} = 0$, and

$$\eta = \sqrt{-\log\left(w_{D,1}\,(1-\alpha)^{T-1}\right)/(1-\epsilon)^2\,/2^k}$$

  while $V_{sum} \leq 2^k$ **do**
    Predict with probability $w_{P,t}$, refuse otherwise,

$$\hat{y}_t = \begin{cases} \hat{y}_{P,t} & \text{with prob. } w_{P,t} \\ \varnothing & \text{otherwise} \end{cases}$$

    Update the prediction probability:

$$w_{P,t+1} \;=\; \alpha + (1-\alpha)\,\frac{w_{P,t}e^{-\eta l_{P,t}}}{w_{P,t}e^{-\eta l_{P,t}} + w_{D,t}e^{-\eta\epsilon}}$$

    Compute $V_{sum} \leftarrow V_{sum} + w_{P,t+1}w_{D,t+1}$
    Increment t by 1, i.e., $t \leftarrow t+1$

As in Section 3, the proposed learning rates for Adaptive SafePredict and therefore the Weight-Shifting SafePredict depend on V^*, and can be estimated using the doubling trick as described in Section 3.3. For the sake of completeness, pseudo-code for the Weight-Shifting SafePredict with the doubling trick is given in Algorithm 5. We can also extend the validity bound given in Corollary 4.1.2 by following the same steps we used in the proof of Theorem 3.6.

Corollary 4.2.1. *For any P, $\epsilon < 1/2$, $0 < w_{P,1} < 1$, and $0 \leq \alpha < 1$ Weight-Shifting SafePredict with the doubling trick given in Algorithm 5 satisfies*

$$\frac{L^*_{P,T}}{T^*} \;\leq\; \epsilon + (1-\epsilon)\,\frac{2\sqrt{2V^*}}{\sqrt{2}-1}\,\frac{\sqrt{\log(1/w_{D,1}) + T\alpha + T\alpha^2}}{T^*}.$$

Note that for $\alpha = 0$, Algorithm 5 reduces to the original SafePredict (Alg. 3). Increasing α, i.e., the adaptivity, increases the efficiency, since we increase the likelihood to make a prediction at each step. Furthermore, validity is guaranteed as long as α is on the order of $1/T$. In the next section, the impact of α is numerically evaluated.

5 Experiments

In this section, we investigate the performance of the proposed meta-algorithms on both synthetic and standard machine learning datasets.[1] In Section 5.1, we randomly generate loss sequences and verify the validity of our algorithms

1. For the sake of reproducibility, the code used to generate our results, tabulated results on the synthetic data, and the experiments on other datasets are provided in the supplementary material.

empirically for various loss statistics and various degrees of adaptivity. The experiments show that the Weight-Shifting SafePredict boosts the number of predictions in changing environments while preserving the validity of the algorithm.

In Section 5.2, we compare SafePredict with popular confidence-based refusal methods on three well-known and varied machine learning datasets, namely, MNIST digit recognition [42] (a vision application), IMDB sentiment analysis [43] (a natural language problem), and Reuters topic recognition [44] (text classification). SafePredict increases the efficiency relative to other refusal mechanisms while guaranteeing validity.

In the following, for the sake of brevity, we refer to Weight-Shifting SafePredict (Alg. 5) as simply SafePredict by noting that $\alpha = 0$ corresponds to the original SafePredict (Alg. 3).

5.1 Synthetic Data

In this subsection, we generate a binary sequence of loss-values with varying error probabilities. The subsection examines the validity and efficiency of SafePredict. In particular, we restrict the adaptivity parameter of the SafePredict to have the form $\alpha = k/T$, per Corollary 4.1.2, and observe the trade-off in choosing the parameter k.

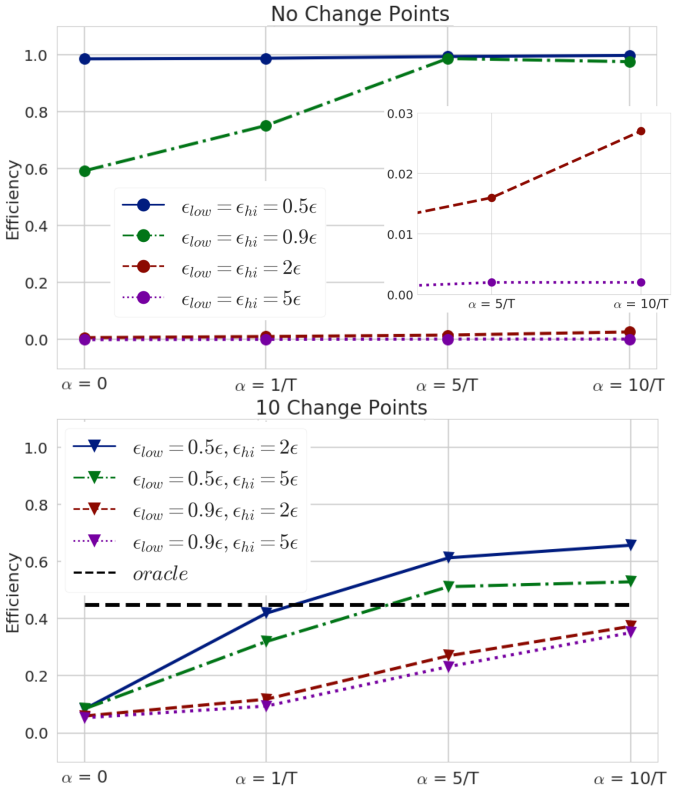


Fig. 2: *Efficiency Experiments on Synthetic Data: The efficiency (T^*/T) of SafePredict with respect to increasing choices of α. (top) If the base predictor has a constant error rate which is higher than the target, SafePredict almost always refuses. The number of predictions, in this case, increases with α. (bottom) On the other hand, when the error rate of the base predictor fluctuates around the target, the efficiency of SafePredict increases as α increases and achieves nearly the same efficiency as the oracle, which predicts if only if $\epsilon_t \leq \epsilon$. So, in all cases, asymptotic validity is preserved.*

Parameters: We fix the time horizon $T = 50000$, initial weight $w_{P,1} = 0.5$ and the target error rate $\epsilon = 0.05$. Then we evaluate our results for $\alpha \in \{0, 1/T, 5/T, 10/T\}$.

Data Generation: To evaluate the performance of the meta-algorithm, we assume the existence of a base predictor P with a time-varying error rate, and generate the loss sequence corresponding to its predictions randomly. To model the changes in the error-rate we employ a simple change point model. The statistical properties of the generated loss sequence $l_{P,1}, \ldots, l_{P,T}$ are characterized by the following three parameters: low error level (ϵ_{low}), high error level (ϵ_{hi}), and the number of changepoints ($numChange$). To generate a particular loss sequence, we first split the time horizon into $numChange+1$ non-overlapping, consecutive, equal length blocks. Then we assume the error rate of P to be constant within each block and alternates between ϵ_{low} and ϵ_{hi} for consecutive blocks. Formally, we generate each $l_{P,t}$ as an independent Bernoulli random variable as follows:

$$l_{P,t} = \begin{cases} 1 & \text{with prob. } \epsilon_t \\ 0 & \text{with prob. } 1 - \epsilon_t \end{cases},$$

where

$$\epsilon_t = \begin{cases} \epsilon_{low} & \text{if } \lceil t(numChange+1)/T \rceil \text{ is even} \\ \epsilon_{hi} & \text{otherwise} \end{cases}.$$

Results: We generate 12 distinct loss sequences with different $numChange$, ϵ_{low} and ϵ_{hi} values and evaluate the error rate and efficiency of Algorithm 5 for various values of α. The complete numerical results are presented in the supplementary material, but the key observations about the efficiency of SafePredict are summarized in Figures 2 and 3.

As a baseline for comparison, the results of an idealized oracle are also included. We assume the oracle has access to the true error rate of P, i.e., ϵ_t, at each time point and decides to predict if and only if $\epsilon_t \leq \epsilon$. In other words, its prediction probability is equal to

$$w_{P,t} = \begin{cases} 1 & \text{if } \epsilon_t \leq \epsilon \\ 0 & \text{otherwise} \end{cases}.$$

By contrast, in all our experiments, SafePredict does not know the number or location in time of the change points. So the oracle enjoys a significant advantage.

These simulations reveal two main issues:

1) Bound on Validity: Following Corollary 4.1.2, for all $\alpha = O(1/T)$, the excess error rate is $O\left(\sqrt{V^*}/T^*\right)$. However, the constants are hidden by the big-oh notation increase with α, and become significant when T^* is small. This effect can be observed most prominently in experiments where the error rate of the base algorithm is consistently higher than the target rate. In these cases, the oracle always refuses as it should whereas SafePredict refuses often but not always. Asymptotically, SafePredict is still valid, but its error rate may exceed the target for finite sequences. On the other hand, in the experiments where the base predictor achieves the target for significant periods of time, the excess error rate of SafePredict stays within 7% of the target error rate (below 0.0035 for $\epsilon = 0.05$) and the efficiency increases with α, see Table 1 in Supplementary Material.

2) Efficiency via adaptivity: As expected from the theoretical analysis in Section 4 and empirically observed in Figure 2, the efficiency of SafePredict increases with α. SafePredict

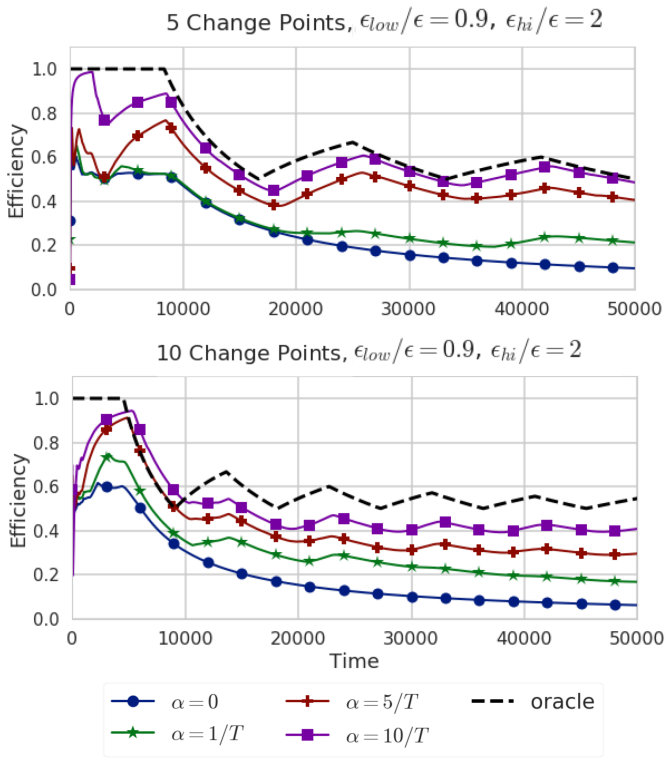


Fig. 3: *Synthetic Data, Evolution of Efficiency: Note $\alpha = 0$ corresponds to the original SafePredict (Alg. 3) and has no adaptivity. For $\alpha > 0$, SafePredict can track the change points and boost efficiency. Larger α implies better tracking. As the number of change points increases, SafePredict does a poorer job tracking the performance of the base predictor (relative to the oracle that knows the error rate), thus the efficiency drops. All the predictors in the figures are valid.*

performs nearly as well as the oracle, even though the oracle knows the true error probabilities and SafePredict does not. However, as the number of change points increases, SafePredict must refuse more to be able to adapt to the changes, and therefore suffers a drop in efficiency. As can be seen in Figure 3, the efficiency of SafePredict decreases with the number of change points while the tracking ability of SafePredict increases with α, sometimes approaching the efficiency achieved by the oracle.

5.2 Real Data

We now explore the validity and efficiency of SafePredict on the three datasets mentioned above: MNIST, IMDB and Reuters topic classification with neural nets as the base predictor. The neural nets are implemented using the Keras library in the R programming language.For a brief description of the datasets and the neural network architectures please refer to Table 1 of the supplementary material. Furthermore, to demonstrate the simplicity and the generality of our method, the supplementary material also includes a Python implementation using a random forest as the base predictor with six datasets from the UCI repository [45] along with both R and Python implementations of our algorithm.

We also compare SafePredict with a natural confidence-based refusal mechanism. This method is widely used in practice, e.g., [11], [46], [47], and similar methods are used as baselines in the literature, see e.g., [17], [48]. Furthermore, one can conveniently make a fair comparison with SafePredict since both are meta-algorithms that can be used on top of (almost) any predictor. Finally, we investigate a heuristically promising method of combining SafePredict with the confidence-based mechanism to improve efficiency.

For each of the datasets, we randomly permute the data and choose the first 10000 data points to use in our experiments. Artificial change points are introduced at every 2000 data points by applying a random label permutation[2] to all the data points after the change point, i.e., we effectively change the data distribution (the mapping from features to target) at each change point. Finally, we fix the target error rate as $\epsilon = 0.05$ in our experiments.

Neural networks enjoy high predictive accuracy for such high dimensional data. For the MNIST dataset, we use a simple three-layer feed-forward fully connected multilayer perceptron as on the intensity values of the 784 gray-tone pixels. For the IMDB and Reuters datasets, we used a word-of-bags representation of each data point, restricting ourselves to the most common 20000 words in the corpus as the input of the FastText architecture [49]. Please see the supplementary material for the implementation details. In the scope of this experiment, we retrain the base predictor once at every 200 data points on all the data points observed so far and use its predictions for the next 200 data points, to track changes in labeling.

The confidence-based refusal method we consider in this paper starts with a base predictor that outputs a confidence score for each prediction, and a refusal threshold. The meta-algorithm decides to refuse if the confidence score does not exceed the threshold value. In our experiments, we used the maximum probability estimate produced by the base predictor as the confidence score for each prediction. As in the case of retraining the base predictors, we update the refusal threshold once every 200 data points. In particular, to update the threshold at time t we re-train the base predictor using the first $t - 200$ data points and choose the smallest threshold (i.e., the one refuses the least) that gives an error rate smaller than ϵ over the non-refused predictions in the last 200 data points ($t - 200$ to $t - 1$).

We show the results using the SafePredict meta-algorithm (Alg. 5) with fixed parameters $\alpha = 0.0005$ $(5/T)$ and $w_{P,1} = 0.5$. SafePredict is used on top of the neural nets either by itself or in a pipeline base predictor then confidence-based refusal meta-algorithm then SafePredict. In the latter multi-meta-algorithm scenario, SafePredict uses the losses suffered by the confidence-based algorithm as input. When the confidence-based meta-algorithm refuses to make a prediction for data point at time t, SafePredict also refuses to predict and does not update the weights, i.e., ignores the data point at time t.

Finally, we include an "amnesic adaptive" version of the combined meta-algorithm and base predictor that considers excessive refusals of SafePredict as a sign that adaptation is needed. To do so, the amnesic adaptive version monitors the average prediction probability of the SafePredict over the predictions made by the base predictor within the last epoch

2. For the IMDB dataset, we simply toggled the labels instead of randomly permuting them since there are only two labels.

(i.e., 200 data points). If the average is less than 0.01 the amnesic adaptive version ignores all training data observed so far, and restart the base predictor (i.e., the neural net and the confidence-based meta-algorithm) using only the most recent 200 (epoch size) data points. The evaluation of efficiency (T^*/T) and error rate ($L^*_{P,T}/T^*$) versus time is plotted for each of these predictors in Fig. 4, 5, and 6.

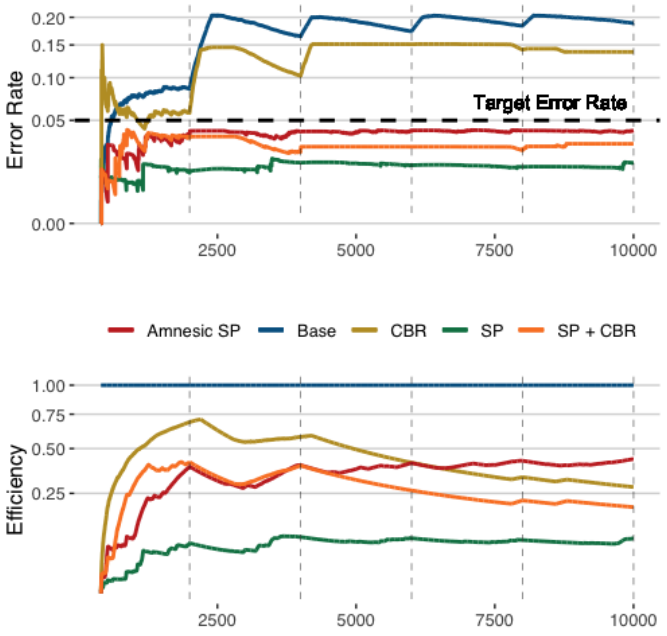


Fig. 4: *MNIST Dataset: Efficiency is 1.0 for the base predictor but lower for the various refusing meta-algorithms. However, the base predictor has a poor error rate (way over ϵ). All the SafePredict variants rapidly approach an error rate value below the target error rate* 0.05 *as they make predictions. The confidence-based meta-algorithm cannot guarantee asymptotic validity due to the changes in the underlying distribution (marked by vertical dashed lines). Two forms of adaptivity help reduce the number of refusals: weight-shifting especially with a high α value and amnesic adaptivity. Combining both leads to the highest efficiency while preserving validity.*

Experimental results lead to the following observations:

1) Validity: The confidence-based refusal mechanism fails to satisfy the validity requirement of keeping the error rate below ϵ (top subfigures of Figure 4, 5 and 6). The reason is that confidence-base refusal requires data points to be (at least approximately) exchangeable to achieve the required error guarantee. This assumption fails after the change point. On the other hand, SafePredict establishes validity by refusing when the base predictor cannot achieve the error rate without making any assumptions about the data points. Note that for the Reuters dataset (Figure 6), the base predictor never reaches the target error rate 0.05. Thus, SafePredict refuses to make a prediction almost all the time, $T^* = 16.44$.

2) Efficiency: Generally, the confidence-based refusal method has a higher efficiency than SafePredict due to the nature of the refusal strategies (confidence-based methods make stronger (e.g., i.i.d.) assumptions so predict more often, but fail to be valid when assumptions do not hold).

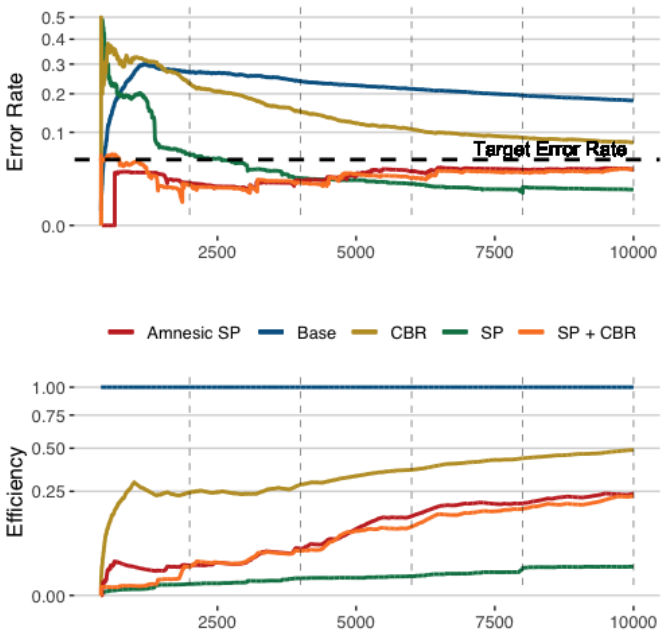


Fig. 5: *IMDB Dataset: Similar to Fig. 4, except in this case, the experiment consists of a randomly chosen 10000 data points used from the IMDB sentiment analysis dataset [43] instead of MNIST.*

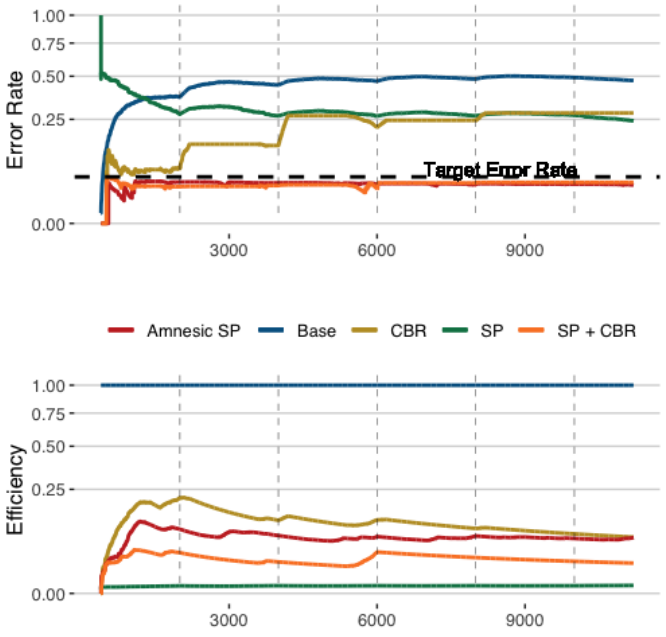


Fig. 6: *Reuters Dataset: Similar to Fig. 4, except randomly chosen 10000 data points used from Reuters dataset [44] instead of MNIST. Note the error rate for SafePredict (SP) is way above the target error rate for the first* $10,000$ *points. Ideally, SafePredict should refuse always. In fact, the expected number of predictions (T^*) is only* 16.44.

However, the discrepancy in the efficiency can be mitigated by employing the three-way pipeline: base predictor then confidence-based mechanism then SafePredict. This method implies a two-layered refusal mechanism, but as we discussed in Section 3.2 and 4.2, the second layer (SafePredict) will seldom refuse as long as the first layer (confidence-based algorithm) has a validity rate as high as the desired correctness rate. As can be seen from Figure 4-6, this method (SP+CBR) combines the best of confidence-based refusals and SafePredict by performing almost as efficiently as confidence-based refusals before the change point and achieving validity throughout.

3) Amnesic Adaptivity: SafePredict following the confidence-based predictor remains valid throughout, but refusals increase after the change point since the confidence-based mechanism is not valid anymore and thus causes excessive errors. In the amnesic adaptive variant, we use the excessive refusals to trigger an update of the base algorithm. Specifically, if the number of data points predicted by the confidence-based predictor but refused by SafePredict is large, the amnesic approach concludes that the confidence-based algorithm is no longer well-calibrated, so earlier data points should be ignored. We denote this adaptive method as "Amnesic CBR+SP". As seen in the plots, Amnesic CBR+SP gives the most favorable performance in our experiments by preserving validity thanks to SafePredict and achieving better efficiency after the change point by forcing the confidence-based algorithm to forget early data points.

6 Conclusion

We have introduced a meta-algorithm, SafePredict, that works with any base prediction algorithm (including ensembles of prediction algorithms) and asymptotically guarantees an upper bound on the error rate for non-refused predictions. The error guarantee achieved by SafePredict does not depend on any assumption on the data or the base prediction algorithm. To achieve this, we refined the regret notion from the expert advice framework and recast the exponentially weighted average forecasting algorithm to be used as a method to manage refusals.

To avoid too many refusals in changing environments, we introduced a weight-shifting heuristic that encourages predictions when the quality of the base predictor improves. We have also used an amnesic adaptation mechanism to improve versatility in the face of occasional change points. Our experiments show that these methods ensure validity even in challenging environments and seldom refuse whenever the base predictor achieves a high enough correctness rate.

Acknowledgments

Work supported in part by NYU Seed Grant, NYU WIRELESS, and the United States National Science Foundation grants CNS-1302336, MCB-1158273, IOS-1339362, and MCB-1412232. This support is greatly appreciated. We would also like to thank the editor and the reviewers for their suggestions and criticisms.

References

[1] E. Siegel, *Predictive Analytics: The Power to Predict Who will Click, Buy, Lie, or Die*. John Wiley & Sons, 2013.
[2] S. Dua, U. R. Acharya, and P. Dua, *Machine Learning in Healthcare Informatics*. Springer, 2014.
[3] B. Han, "Building a Better Disease Detective," *IEEE Spectr.*, vol. 52, no. 10, pp. 46–51, 2015.
[4] T. Harbert, "The Law Machine," *IEEE Spectr.*, vol. 50, no. 11, pp. 31–54, 2013.
[5] T. Simonite, "How to Upgrade Judges with Machine Learning." MIT Technology Review, Mar 2017.
[6] N. Cesa-Bianchi and G. Lugosi, *Prediction, Learning, and Games*. Cambridge University Press, 2006.
[7] C. Chow, "On Optimum Recognition Error and Reject Trade-off," *IEEE Trans. Inf. Theory*, vol. 16, no. 1, pp. 41–46, 1970.
[8] J. H. Friedman, "On Bias, Variance, 0/1-Loss, and the Curse-of-Dimensionality," *Data Mining and Knowledge Discovery*, vol. 1, no. 1, pp. 55–77, 1997.
[9] M. E. Hellman, "The Nearest Neighbor Classification Rule with a Reject Option," *IEEE Trans. on Systems Science and Cybern.*, vol. 6, no. 3, pp. 179–185, 1970.
[10] T. C. Landgrebe, D. M. Tax, P. Paclík, and R. P. Duin, "The Interaction Between Classification and Reject Performance for Distance-Based Reject-Option Classifiers," *Pattern Recognition Lett.*, vol. 27, no. 8, pp. 908–917, 2006.
[11] C. De Stefano, C. Sansone, and M. Vento, "To Reject or Not to Reject: That is the Question-an Answer in Case of Neural Classifiers," *IEEE Trans. Syst. Man Cybern. C, Appl. Rev.*, vol. 30, no. 1, pp. 84–94, 2000.
[12] M. Li and I. K. Sethi, "Confidence-based Classifier Design," *Pattern Recognition*, vol. 39, no. 7, pp. 1230–1240, 2006.
[13] W. J. Scheirer, L. P. Jain, and T. E. Boult, "Probability Models for Open Set Recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 11, pp. 2317–2324, 2014.
[14] M. Golfarelli, D. Maio, and D. Malton, "On the Error-Rreject Trade-off in Biometric Verification Systems," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 786–796, 1997.
[15] G. Fumera, I. Pillai, and F. Roli, "Classification with Reject Option in Text Categorisation Systems," in *12th Int. Conf. on Image Anal. and Process.* IEEE, 2003, pp. 582–587.
[16] M. C. Campi, "Classification with Guaranteed Probability of Error," *Mach. Learning*, vol. 80, no. 1, pp. 63–84, 2010.
[17] P. L. Bartlett and M. H. Wegkamp, "Classification with a Reject Option Using a Hinge Loss," *J. of Mach. Learning Research*, vol. 9, pp. 1823–1840, Aug, 2008.
[18] R. Herbei and M. H. Wegkamp, "Classification with Reject Option," *Canadian J. of Stat.*, vol. 34, no. 4, pp. 709–721, 2006.
[19] M. Yuan and M. Wegkamp, "Classification Methods with Reject Option Based on Convex Risk Minimization," *J. of Mach. Learning Research*, vol. 11, no. Jan, pp. 111–130, 2010.
[20] R. El-Yaniv and Y. Wiener, "On the Foundations of Noise-Free Selective Classification," *J. of Mach. Learning Research*, vol. 11, no. May, pp. 1605–1641, 2010.
[21] Y. Wiener and R. El-Yaniv, "Agnostic Selective Classification," in *Advances in Neural Inform. Process. Syst.*, 2011, pp. 1665–1673.
[22] C. Cortes, G. DeSalvo, and M. Mohri, "Learning with Rejection," in *Int. Conf. on Algorithmic Learning Theory*. Springer, 2016, pp. 67–82.
[23] A. T. Kalai, V. Kanade, and Y. Mansour, "Reliable Agnostic Learning," *J. of Comput. and Syst. Sci.*, vol. 78, no. 5, pp. 1481–1495, 2012.
[24] Y. Wiener, *Theoretical Foundations of Selective Prediction*, ser. PhD dissertation. Technion-Israel Inst. of Technology, Faculty of Comput. Sci., 2013.
[25] C. Zhang, W. Wang, and X. Qiao, "On Reject and Refine Options in Multicategory Classification," *J. of the Amer. Statistical Assoc.*, 2017.
[26] V. Vovk, A. Gammerman, and G. Shafer, *Algorithmic Learning in a Random World*. Springer Sci. & Bus. Media, 2005.
[27] C. Denis and M. Hebiri, "Consistency of Plug-in Confidence Sets for Classification in Semi-supervised Learning," *arXiv preprint arXiv:1507.07235*, 2015.
[28] M. A. Kocak, E. Erkip, and D. E. Shasha, "Conjugate Conformal Prediction for Online Binary Classification," in *32nd Conf. on Uncertainty in Artificial Intell.*, 2016, pp. 347–356.
[29] J. Lei *et al.*, "Classification with Confidence," *Biometrika*, vol. 101, no. 4, pp. 755–769, 2014.

[30] L. Li, M. L. Littman, and T. J. Walsh, "Knows What it Knows: a Framework for Self-Aware Learning," in *25th Int. Conf. on Mach. Learning*. ACM, 2008, pp. 568–575.

[31] A. Sayedi, M. Zadimoghaddam, and A. Blum, "Trading off Mistakes and Don't-Know Predictions," in *Advances in Neural Inform. Process. Syst.*, 2010, pp. 2092–2100.

[32] C. Zhang and K. Chaudhuri, "The Extended Littlestone's Dimension for Learning with Mistakes and Abstentions," *arXiv preprint arXiv:1604.06162*, 2016.

[33] N. Littlestone and M. K. Warmuth, "The Weighted Majority Algorithm," in *30th Annu. Symp. on Found. of Comput. Sci.* IEEE, 1989, pp. 256–261.

[34] V. Vovk, "Aggregating Strategies," in *Conf. on Computational Learning Theory*, 1990.

[35] J. W. Tukey, "Sunset Salvo," *The Amer. Statistician*, vol. 40, no. 1, pp. 72–76, 1986.

[36] K. Krickeberg, *Probability Theory*, ser. Adiwes Int. Series. Addison-Wesley Publishing Company, 1965.

[37] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of Machine Learning*. MIT Press, 2012.

[38] N. Cesa-Bianchi, Y. Mansour, and G. Stoltz, "Improved Second-order Bounds for Prediction with Expert Advice," *Machine Learning*, vol. 66, no. 2-3, pp. 321–352, 2007.

[39] S. De Rooij, T. Van Erven, P. D. Grünwald, and W. M. Koolen, "Follow the Leader if You can, Hedge if You must." *J. of Mach. Learning Research*, vol. 15, no. 1, pp. 1281–1316, 2014.

[40] M. Herbster and M. K. Warmuth, "Tracking the Best Expert," *Mach. Learning*, vol. 32, no. 2, pp. 151–178, 1998.

[41] D. Adamskiy, W. M. Koolen, A. Chernov, and V. Vovk, "A Closer Look at Adaptive Regret," in *Int. Conf. on Algorithmic Learning Theory*. Springer, 2012, pp. 290–304.

[42] Y. LeCun and C. Cortes, "MNIST Handwritten Digit Database," 2010. [Online]. Available: http://yann.lecun.com/exdb/mnist/

[43] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning Word Vectors for Sentiment Analysis," in *49th Annu. Meeting of the Assoc. for Computational Linguistics: Human Language Technologies*. Portland, Oregon, USA: Association for Computational Linguistics, June 2011, pp. 142–150. [Online]. Available: http://www.aclweb.org/anthology/P11-1015

[44] empty, "Reuters-21578 dataset," empty. [Online]. Available: http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html

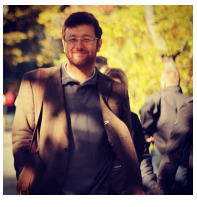
[45] M. Lichman, "UCI Machine Learning Repository," 2013. [Online]. Available: http://archive.ics.uci.edu/ml

[46] B. Zhang, Y. Zhou, and H. Pan, "Vehicle Classification with Confidence by Classified Vector Quantization," *IEEE Intell. Transp. Syst. Mag.*, vol. 5, no. 3, pp. 8–20, 2013.

[47] B. Hanczar and E. R. Dougherty, "Classification with Reject Option in Gene Expression Data," *Bioinformatics*, vol. 24, no. 17, pp. 1889–1895, 2008.

[48] C. Cortes, G. DeSalvo, M. Mohri, and S. Yang, "On-line Learning with Abstention," *arXiv preprint arXiv:1703.03478*, 2017.

[49] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of Tricks for Efficient Text Classification," *arXiv preprint arXiv:1607.01759*, 2016.

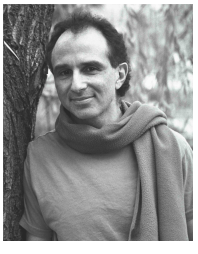


Mustafa A. Kocak is a computational biologist in Broad Institute of MIT and Harvard, His research interests include machine learning applications applications, information theory, and bio-statistics. Kocak received a B.Sc. in electrical engineering from Bilkent University (Ankara, Turkey) and his Ph.D. from NYU School of Engineering. Contact him at mkocak@broadinstitute.org.

David Ramirez received a B.S. with honors in Engineering Physics from Tecnologico de Monterrey (ITESM), and M.S. and Ph.D. degrees in Electrical and Computer Engineering from Rice University. He is currently a Postdoctoral Researcher at New York University and a Visiting Postdoctoral Researcher at Princeton University. His research interests are in wireless networks, communication theory, & optimization.

Elza Erkip received the B.S. degree in Electrical and Electronics Engineering from Middle East Technical University, Ankara, Turkey, and the M.S. and Ph.D. degrees in Electrical Engineering from Stanford University, Stanford, CA, USA. Currently, she is an Institute Professor in the Electrical and Computer Engineering Department at New York University Tandon School of Engineering, Brooklyn, NY, USA. Her research interests are in information theory, communication theory, and wireless communications.

Dr. Erkip is a member of the Science Academy of Turkey and is among Clarivate Highly Cited Researchers. She received the NSF CAREER award in 2001, the IEEE Communications Society WICE Outstanding Achievement Award in 2016, and the IEEE Communications Society Communication Theory Technical Committee (CTTC) Technical Achievement Award in 2018. Her paper awards include the IEEE Communications Society Stephen O. Rice Paper Prize in 2004, the IEEE Communications Society Award for Advances in Communication in 2013 and the IEEE Communications Society Best Tutorial Paper Award in 2019. She has been a member of the Board of Governors of the IEEE Information Theory Society since 2012 where she was the Society President in 2018. She was a Distinguished Lecturer of the IEEE Information Theory Society from 2013 to 2014.



Dennis E. Shasha is a Julius Silver Professor of Computer Science at the Courant Institute of New York University. His research interests include data science, biological computing, wireless communication and concurrent data structures. Shasha received a PhD in applied math from Harvard University. He is an ACM Fellow and the recipient of an INRIA Internaional Chair. He is co-editor in chief of Information Systems, and is or has been the puzzle columnist for several magazines, including CACM and Scientific American. Contact shasha@courant.nyu.edu