Biometrics NA, 1–11
NA NA

Approval policies for modifications to Machine Learning-Based Software as a Medical Device: A study of bio-creep

Jean Feng*, **Scott Emerson****, **and Noah Simon*****

Department of Biostatistics, University of Washington, Seattle, WA, USA

**email:* jeanfeng@uw.edu

***email:* semerson@uw.edu

****email:* nrsimon@uw.edu

Summary: Successful deployment of machine learning algorithms in healthcare requires careful assessments of their performance and safety. To date, the FDA approves locked algorithms prior to marketing and requires future updates to undergo separate premarket reviews. However, this negates a key feature of machine learning–the ability to learn from a growing dataset and improve over time. This paper frames the design of an approval policy, which we refer to as an automatic algorithmic change protocol (aACP), as an online hypothesis testing problem. As this process has obvious analogy with noninferiority testing of new drugs, we investigate how repeated testing and adoption of modifications might lead to gradual deterioration in prediction accuracy, also known as "biocreep" in the drug development literature. We consider simple policies that one might consider but do not necessarily offer any error-rate guarantees, as well as policies that do provide error-rate control. For the latter, we define two online error-rates appropriate for this context: Bad Approval Count (BAC) and Bad Approval and Benchmark Ratios (BABR). We control these rates in the simple setting of a constant population and data source using policies aACP-BAC and aACP-BABR, which combine alpha-investing, group-sequential, and gate-keeping methods. In simulation studies, bio-creep regularly occurred when using policies with no error-rate guarantees, whereas aACP-BAC and -BABR controlled the rate of bio-creep without substantially impacting our ability to approve beneficial modifications.

Key words: AI/ML-based SaMD; Alpha-investing; Gate-keeping; Group-sequential; Online hypothesis testing.

Due to the rapid development of artificial intelligence (AI) and machine learning (ML), the use of AI/ML-based algorithms has expanded in the medical field. As such, an increasing number of AI/ML-based Software as a Medical Device (SaMD) are seeking approval from the Center of Diagnostics and Radiologic Health (CDRH) at the US Food and Drug Administration (FDA). ML algorithms are attractive for their ability to improve over time by training over a growing body of data. Thus, rather than using a locked algorithm trained on a limited dataset, developers might like to train it further on a much more representative sample of the patient population that can only be obtained after deployment. To collect input on this regulatory problem, the FDA recently outlined a proposed regulatory framework for modifications to AI/ML-based SaMDs in a discussion paper (FDA, 2019).

Regulating evolving algorithms presents new challenges because the CDRH has historically only approved “locked” algorithms, i.e. algorithms that do not change after they are approved. This is a new regulatory problem because updating traditional medical devices and drugs is often logistically difficult whereas updating software is both fast and easy.

FDA (2019) proposes companies stipulate SaMD Pre-specifications (SPS) and an Algorithm Change Protocol (ACP). When listing the anticipated modifications in the SPS, it behooves the company to cast as wide a net as possible within FDA-imposed constraints. The ACP specifies how the company will ensure that their modifications are acceptable for deployment. Once the FDA approves the SPS and ACP, the company follows these prespecified procedures to deploy changes without further intervention. As such, we refer to the ACP in this paper as an “automatic ACP” (aACP). The aACP is the FDA’s primary tool for ensuring safety and efficacy of the modifications. However, specific aACP designs or requirements are noticeably absent from FDA (2019). This paper aims to address this gap.

A manufacturer has two potential motivations for changing an AI/ML-based SaMD: to advance public health and to increase their financial wealth. Modifications that improve performance and usability are encouraged. On the other hand, changes that do not and are deployed only for the sake of change itself have been used in the past to advance a manufacturer's financial interest and are contrary to public interest. Historically, such modifications have been used to 1) decrease competition because it is difficult for competitors to compare against an ever-changing benchmark; 2) file for a patent extension and keep prices artificially high; and 3) increase sales for a supposedly new and improved product (Gupta et al., 2010; Hitchings et al., 2012; Gottlieb, 2019). To prevent this type of behavior with drugs and biologics, the FDA regulates modifications through various types of bridging studies (International Conference on Harmonisation (1998)). Likewise, an aACP should only grant approval to modifications to AI/ML-based SaMD after ensuring safety and efficacy.

This paper provides a framework for designing and evaluating an aACP, considers a variety of aACP designs, and investigates their operating characteristics. We assume the manufacturer is allowed to propose arbitrary (and possibly deleterious) modifications, which include changes to model parameters, structure, and input features. For this manuscript, we focus on the setting of a constant population and data source, rather than more complicated settings with significant time trends. Throughout, we evaluate modifications solely in terms of their operating characteristics. Thus, the aACPs treat simple models and complex black-box estimators, such as neural networks and boosted gradient trees, the same. This parallels the drug approval process, which primarily evaluates drugs on their efficacy and safety with respect to some endpoints, even if the biological mechanism is not completely understood.

To our knowledge, there is no prior work that directly addresses the problem of regulating modifications to AI/ML-based SaMD, though many have studied related problems. In online hypothesis testing, alpha-investing procedures are used to control the online false discovery rate (FDR) (Foster and Stine, 2008; Javanmard and Montanari, 2015; Ramdas et al., 2017, 2018; Zrnic et al., 2018), which is important for companies that test many hypotheses over a long period of time (Tang et al. (2010)). We will consider aACPs that use alpha-investing to control online error rates; However, we will need to significantly adapt these ideas for use in our context. In addition, differential privacy methods (Blum and Hardt, 2015; Dwork et al., 2015) have been used to tackle the problem of ranking model submissions to a ML competition, where the submissions are evaluated using the same test data and models are submitted in a sequential and adaptive manner. Though that problem is related, those approaches cannot evaluate modifications that add previously-unmeasured covariates. Finally, online learning methods are a major motivation for studying this regulatory problem and can be used to automatically update the model (Shalev-Shwartz, 2012). However, rather than designing bespoke aACPs for online learning methods, we will consider approval policies for arbitrary modifications as a first step.

This paper evaluates the rates at which different policies make bad approvals as well as their rates of approving beneficial modifications. Due to the analogy between this problem and noninferiority testing of new drugs, we investigate how repeated testing of proposed modifications might lead to gradual deterioration in model performance, also known as "bio-creep" (Fleming, 2008). We compare simple aACPs that one might consider, but do not necessarily have error-rate guarantees, to policies that *do* provide error rate control. For the latter, we define two online error rates appropriate for this context—the expected Bad Approval Count (BAC) and Bad Approval and Benchmark Ratios (BABR)—and control them using policies aACP-BAC and aACP-BABR, respectively. In simulation studies, bio-creep frequently occurred when using the simple aACPs. By using aACP-BAC or -BABR instead, we significantly reduce the risk of bio-creep without substantially reducing the rate of model improvement. Based on these findings, we conclude that 1) bio-creep is a major concern when designing an aACP and 2) there are promising solutions for mitigating it without substantially hindering model improvements.

1. Motivating examples

We present examples of actual AI/ML-based medical devices and discuss possible modifications that manufacturers might consider. The examples are ordered by increasing regulatory complexity and risk. Throughout, we only discuss regulating modifications to the software and assume the intended use of the device remains constant.

1.1 *Blood tests using computer vision*

Sight Diagnostics has developed a device that collects and images blood samples to estimate complete blood count (CBC) parameters. They are evaluating the device in a clinical trial (ClinicalTrials.gov ID NCT03595501) where the endpoints are the estimated linear regression parameters (slope and intercept) between their CBC parameter estimates and gold standard.

The FDA requires locking the entire procedure, which includes blood collection, imaging, and the ML algorithm, prior to marketing. Nonetheless, the company might want to improve the accuracy of their test after obtaining regulatory approval. For instance, they can train more complex models that capture nonlinearities and interactions (between covariates and/or outcomes) or use a different FDA-approved device to image the blood sample. All these changes have the potential to improve prediction accuracy, though it is not guaranteed.

To regulate such modifications, we will need to define acceptable changes to endpoint values. This is not straightforward when multiple endpoints are involved: Do all endpoints have to improve? What if the model has near-perfect performance with respect to some endpoints and room for improvement for others? To tackle these questions, we must run both superiority and non-inferiority (NI) tests. Moreover, introducing NI tests prompts even more questions, such as how to choose an appropriate NI margin.

1.2 *Detecting large vessel occlusion from CT angiogram images of the brain*

ContaCT is a SaMD that identifies whether CT angiogram images of the brain contain a suspected large vessel occlusion. If so, it notifies a medical specialist to intervene. The manufacturer evaluated ContaCT using images analyzed by neuro-radiologists. The primary endpoints were estimated sensitivity and specificity. The secondary endpoint was the difference in notification time between ContaCT and standard-of-care. ContaCT achieved 87% sensitivity and 89% specificity and significantly shortened notification time.

Having obtained FDA approval (FDA, 2018), the company might want to improve ContaCT by, say, training on more images, extracting a different set of image features, or utilizing clinical covariates from electronic health records. This last modification type requires special consideration since the distribution of clinical covariates and their missingness distribution are susceptible to time trends.

1.3 *Blood test for cancer risk prediction*

GRAIL is designing a blood test that sequences cell-free nucleic acids (cfNAs) circulating in the blood to detect cancer early. They are currently evaluating this test in an observational study (ClinicalTrials.gov ID NCT02889978) where the gold standard is a cancer diagnosis from the doctor within 30 months. For time-varying outcomes, one may consider evaluating performance using time-dependent endpoints, such as those in Heagerty and Zheng (2005).

After the blood test is approved, GRAIL might still want to change their prediction algorithm. For example, they could collect additional omics measurements, sequence the cfNAs at a different depth (e.g. lower to decrease costs, higher to improve accuracy), or train the model on more data. Regulating modifications to this blood test is particularly difficult because the gold standard might not be observable in all patients, its definition can vary between doctors, and it cannot be measured instantaneously. In fact, the gold standard might not be measurable at all because test results will likely affect patient and doctor behavior.

2. Problem Setup

In this section, we provide a general framework and abstractions to understand the approval process for modifications to AI/ML-based SaMD. We begin with reviewing the approval process for a single AI/ML-based SaMD since it forms the basis of our understanding and is a prerequisite to getting modifications approved.

2.1 *AI/ML-based SaMD*

Formally, the FDA defines SaMD as software intended to be used for one or more medical purposes without being part of a hardware medical device. An AI/ML-system is software that learns to perform tasks by tracking performance measures. A SaMD must be approved for a specific indication, which describes the population, disease, and intended use. Here we only consider SaMDs whose predictions do not change the observed outcome; We leave SaMDs that affect the observed outcome (e.g. by recommending treatment) to future work.

Predictive accuracy is typically characterized by multiple endpoints, or co-primary endpoints (Offen et al., 2007; FDA, 2017). The most common endpoints for binary classifiers are sensitivity and specificity because they tend to be independent of disease prevalence, which can vary across subpopulations (Pepe, 2003). Additionally, we can evaluate endpoints over different subgroups to guarantee a minimum level of accuracy for each one.

We now define a model developer (the manufacturer) in mathematical terms. Let $\mathcal{X}$ be the support of the targeted patient population, where a patient is represented by their covariate measurements. Let $\mathcal{Y}$ be output range (possibly multivariate). Let $\mathcal{Q}$ be a family of prediction models $f : \mathcal{X} \mapsto \mathcal{Y}$. Each model f defines the entire pipeline for calculating the SaMD output, including feature extraction, pre-processing steps, and how missing data is handled. The model developer is a functional g that maps the training data $(X_T, Y_T) \in \mathcal{X}^n \times \mathcal{Y}^n$ to a function in $\mathcal{Q}$. Let $\mathcal{P}$ be the family of distributions for $X \times Y$. The performance of a model f on population $\mathbb{P} \in \mathcal{P}$ is quantified by the K-dimensional endpoint $m : \mathcal{Q} \times \mathcal{P} \mapsto \mathbb{R}^K$. For each endpoint m_k, we assume that a larger value indicates better performance.

2.2 *Modifications to AI/ML-based SaMD*

The proposed workflow in FDA (2019) for modifying an AI/ML-based SaMD iterates between three stages. First, the manufacturer proposes a modification by training on monitoring and/or external data and adds this to a pool of proposed modifications. Second, the aACP evaluates each candidate modification and grants approval to those satisfying some criteria. The most recently approved version is then recommended to doctors and patients. Finally, a new batch of monitoring data is collected, which can be used to evaluate and train future models. For simplicity, suppose these three stages are executed in the above order over a fixed grid of time points $t = 1, 2, ...$.

The model developer is allowed to propose arbitrary modifications in a sequential and possibly adaptive manner. For example, the modification can depend on all previously collected monitoring data as well as the set of approvals up to that time point. For generality, we represent each modification as an entirely separate model. Let filtration $\mathcal{F}_t$ be the sigma algebra representing the information up to time t, which includes observed monitoring data, proposed models, and aACP outputs up to time t. The model developer is a sequence of functionals $\{g_t : t = 1, 2, ...\}$, where g_t is a $\mathcal{F}_t$-measurable functional mapping to $\mathcal{Q}$. Let $\hat{f}_t$ be the realized model proposal at time t. In addition, suppose that each proposed model $\hat{f}_t$ has a maximum wait time Δ_t that specifies how long the manufacturer will wait for approval of this model, i.e. the model is no longer considered for approval after time $t + \Delta_t$.

Time trends are likely to occur in long-running processes, as found in long-running clinical trials and non-inferiority trials (Altman and Royston, 1988; Fleming, 2008). This includes changes to any component of the joint distribution between the patient population and the outcome, such as the marginal distributions of the covariates, their correlation structure, their prognostic values, and the prevalence of the condition. As such, let the joint distribution at time t of patients X_t and outcomes Y_t be denoted $\mathbb{P}_t$. The value of endpoint m for model f at time t is then $m(f, \mathbb{P}_t)$. More generally, we might characterize a model by the average endpoint value over time points t to $t+D-1$ for some $D \geq 1$. We denote the average endpoint using $m(f, \mathbb{P}_{t:t+D-1})$, where $\mathbb{P}_{t:t'}$ indicates a uniform mixture of $\mathbb{P}_t, ..., \mathbb{P}_{t'}$. Here D acts as a smoothing parameter; Larger D increases the smoothness of endpoint values.

Finally, this paper assumes that monitoring data collected at time t are representative of the current population $\mathbb{P}_t$. Of course, satisfying this criteria is itself a complex issue. We will not discuss the challenges here and instead refer the reader to Pepe (2003) for more details, such as selecting an appropriate sampling scheme, measuring positive versus negative examples, and obtaining gold standard versus noisy labels.

2.2.1 *Defining acceptable modifications.*

[Figure 1 about here.]

A fundamental building block for designing an aACP is defining when a modification is acceptable to a reference model. Our solution is to represent which modifications are acceptable using a directed graph between models in $\mathcal{Q}$. If there is a directed edge from model f to model f', then it is acceptable to update f to f'. This "acceptability graph" is parameterized by a pre-defined vector of non-inferiority margins $\epsilon \in \mathbb{R}_+^K$. An update from f to f' is acceptable if it demonstrates non-inferiority with respect to all endpoints and superiority in at least one (Bloch et al., 2001, 2007). So for a binary classifier where the endpoints are sensitivity and specificity, one may select the NI margins to encourage modifications that shift the model to a better ROC curve (Figure 1). An acceptability graph is formally defined below:

Definition 1: For a fixed evaluation window $D \in \mathbb{Z}^+$ and NI margin $\epsilon \in \mathbb{R}_+^K$, the acceptability graph at time t over $\mathcal{Q}$ contains the edge from f to f' if $m_k(f, \mathbb{P}_{t:t+D-1}) - \epsilon_k \leq m_k(f', \mathbb{P}_{t:t+D-1})$ for all $k = 1, ..., K$ and there is some $k = 1, ..., K$ such that $m_k(f', \mathbb{P}_{t:t+D-1}) > m_k(f, \mathbb{P}_{t:t+D-1})$. The existence of this edge is denoted $f \rightarrow_{\epsilon,D,t} f'$ and $f \nrightarrow_{\epsilon,D,t} f'$ otherwise.

In this paper, we assume D is fixed and use the notation $f \rightarrow_{\epsilon,t} f'$. For simplicity, Definition 1 uses the same NI margin across all models. In practice, it may be useful to let the margin depend on the reference model or the previously established limits of its predictive accuracy.

We obtain different graphs for different choices of ϵ. For instance, $\epsilon = 0$ means that a model is only acceptable if it is superior with respect to all endpoints, though this can be overly strict in some scenarios. Setting $\epsilon \neq 0$ is useful for approving modifications that maintain the value of some endpoints or have very small improvements with respect to some endpoints.

Finally, we define hypothesis tests based on the acceptability graph. In an ϵ-acceptability test, we test the null hypothesis that a model f' is not an ϵ-acceptable update to model f at time t, i.e. $H_0 : f \nrightarrow_{\epsilon,t} f'$. A superiority test is simply an ϵ-acceptability test where $\epsilon = 0$.

3. An online hypothesis testing framework

At each time point, we suppose an aACP evaluates which candidates to approve by running a battery of hypothesis tests. Since AI/ML-based SaMDs can be modified more easily and frequently compared to drugs, the aACP may run a large number of tests. To account for the multiplicity of tests, we will frame aACPs as online hypothesis testing procedures where the goal is to control the error rate over a sequence of tests.

Each aACP specifies a sequence of approval functions A_t for times $t = 1, 2, ...$ (Figure 2), where A_t is a $\tilde{\mathcal{F}}_t$-measurable function that outputs the index of the most recently approved model at time t (some value in $\{0, ..., t-1\}$). Filtration $\tilde{\mathcal{F}}_t$ is the sigma-algebra for monitoring data up to time t and proposed models and aACP outputs up to time $t-1$. The index of the latest approved model at time t is denoted $\hat{A}_t$. A model was approved at time t if $\hat{A}_t \neq \hat{A}_{t-1}$. Assuming companies are not interested in approving older models, we require $\hat{A}_t \geq \hat{A}_{t-1}$.

[Figure 2 about here.]

Different approval functions lead to different aACPs. In this paper, we only consider aACPs that evaluate candidate modifications using prospectively-collected monitoring data, i.e. data collected *after* the candidate modification has been proposed, as candidate modifications can train on all previously-collected monitoring data. The following are two simple aACPs that one may plausibly consider but do not provide error-rate guarantees:

aACP-Baseline approves any modification that demonstrates ϵ-acceptability to the initially approved model at a fixed level α. This can be useful when the initial model has high predictive accuracy. The manufacturer may also argue this is reasonable policy because the current laws only require a model to perform better than placebo, i.e. the standard of care without utilizing AI/ML-based SaMDs.

aACP-Reset approves any modification that demonstrates ϵ-acceptability to the currently approved model at some fixed level α. As opposed to aACP-Baseline, this policy encourages the model to improve over time.

4. Online error rates for aACPs

We define two online Type I error rates for this setting and describe aACPs that uniformly control these error rates over time. Manufacturers and regulators should select the error rate definition and aACP most suitable for their purposes. These aACPs achieve error rate control as long as their individual hypothesis tests are controlled at their nominal levels.

For both definitions, the error rate at time T is evaluated over the window $1 \vee (T - W)$ to T for some width $W \geq 1$. The hyperparameter W must be pre-specified and specifies different trade-offs between error control and speed: $W = \infty$ requires the strongest error rate control, but is overly strict in most cases, and $W = 1$ requires the weakest error control, but can lead to bad long-running behavior. The desired trade-off is typically in between these extremes.

4.1 *Bad approval count*

We define a bad approval as one where the modification is unacceptable with regards to *any* of the previously approved models. The first error rate is defined as the expected Bad Approval Count (BAC) within the current window of width W:

Definition 2: The expected bad approval count within the W-window at time T is

$$\mathrm{BAC}_W(T) = E\left[\sum_{t=1\vee(T-W)}^{T} \mathbb{1}\left\{ \exists t' = 1, \ldots, t-1 \text{ s.t. } \hat{f}_{\hat{A}_{t'}} \nrightarrow_{\epsilon,t} \hat{f}_{\hat{A}_t} \right\} \right].$$

This error rate captures two important ways errors can accumulate over time: bio-creep and the multiplicity of hypotheses. We discuss these two issues below.

When a sequence of NI trials is performed and the reference in each trial is the latest model that demonstrated NI, the performance of the approved models will gradually degrade over time; This phenomenon has been called bio-creep in previous work (Fleming, 2008). Bio-creep can also happen in our setting: Even if each approved model demonstrates superiority with respect to some endpoints and NI with respect to others, repeated applications of ϵ-acceptability tests can still lead to approval of strictly inferior models. The risk of bio-creep is particularly pronounced because the model developer can perform unblinded adaptations. To protect against bio-creep, Definition 2 counts it as a type of bad approval.

Second, when a long sequence of hypothesis tests is performed, the probability of a false rejection is inflated due to the multiplicity of hypotheses. Definition 2 accounts for multiplicity by summing the probabilities of bad approvals across the window. It is an upper bound for the probability of making any bad approval within the window, which is similar to the definition of family-wise error rate (FWER). In fact, we use the connection between FWER and BAC in the following section to design an aACP that controls this error rate.

4.1.1 *aACP to control bad approval counts.* We now present aACP-BAC, which uniformly controls $\text{BAC}_W(\cdot)$. An aACP is defined by its skeletal structure, which specifies the sequence of hypothesis tests run, and a procedure that selects the levels to perform the hypothesis tests. To build up to aACP-BAC, we i) first describe a simple aACP skeleton that launches a fixed sequence of group sequential tests (GSTs), ii) add gate-keeping to increase its flexibility, and iii) finally pair it with a sequence of $\tilde{\mathcal{F}}_t$-measurable functions $\{\alpha_t : t = 1, 2, ...\}$ for choosing the hypothesis test levels. The full algorithm is given in Algorithm 1 in Supporting Information. For now, we assume the distributions are constant and simply use the notation $\to_\epsilon$ in place of $\to_{\epsilon,t}$. We discuss robustness to time trends in a later section.

[Figure 3 about here.]

Let us first consider a simple aACP skeleton that compares each proposed model to previously approved models using a single hypothesis test (Figure 3). More specifically, at time t, it launches a group sequential ϵ-acceptability test with the null hypothesis

$$H_0 : \exists t' = 1, ..., t \text{ s.t. } \hat{f}_{\hat{A}_{t'}} \nrightarrow_{\epsilon} \hat{f}_t. \tag{1}$$

The number of interim analyses is the maximum wait time Δ_t and the critical values are chosen according to an alpha-spending function specified prior to launch (DeMets and Lan, 1994). At each time point, we also perform interim analyses for all active hypothesis tests (i.e. those not past their maximum wait time). The aACP approves $\hat{f}_j$ at time t if it demonstrates acceptability to $\hat{f}_{\hat{A}_1}, ..., \hat{f}_{\hat{A}_{t-1}}$. If multiple models are acceptable, it selects the latest one.

A drawback of this simple aACP skeleton is that it fails to adapt to new model approvals that occur in the middle of a group sequential test (GST). Consider the example in Figure 3, where a GST with null hypothesis $H^0_{\hat{f}_0 \nrightarrow_{\epsilon} \hat{f}_1}$ is launched at time $t = 1$ and a second GST with null hypothesis $H^0_{\hat{f}_0 \nrightarrow_{\epsilon} \hat{f}_2}$ is launched at time $t = 2$. If $\hat{f}_1$ is approved at time $t = 3$, this aACP cannot approve $\hat{f}_2$ since its GST only compares $\hat{f}_2$ to $\hat{f}_0$. Ideally, it could adapt to the new approval and add a test comparing $\hat{f}_2$ to $\hat{f}_1$.

aACP-BAC addresses this issue by evaluating proposed model $\hat{f}_t$ using a *family* of acceptability tests instead (Figure 4). In addition to the aforementioned test for the null hypothesis (1), this family includes acceptability tests to test each of the null hypotheses

$$H_{0,j} : \hat{f}_j \nrightarrow_{\epsilon} \hat{f}_t \text{ for } j = \hat{A}_t + 1, ..., t - 1. \tag{2}$$

As before, a model is approved at time t only if it demonstrates acceptability compared to all approved models up to time t. To control the online error rate, aACP-BAC controls the FWER for each family of tests using a serial gate-keeping procedure. Recall that gate-keeping tests hypotheses in a pre-specified order and stops once it fails to reject a null hypothesis (Dmitrienko and Tamhane, 2007). No alpha adjustment is needed in gate-keeping; It controls

FWER at α by performing all tests at level α. Here, the tests are naturally ordered by the index of the reference models, from oldest to latest. Moreover, this ordering maximizes the probability of approval, assuming the proposed models improve in predictive accuracy. We use the overall hierarchical rule to perform GSTs with gate-keeping Tamhane et al. (2018).

To uniformly control $\text{BAC}_W(\cdot)$ at α, aACP-BAC computes an over-estimate of $\text{BAC}_W(t)$ at each time t and selects level $\hat{\alpha}_t$ such that the over-estimate is bounded by α. Using a union bound like that in Bonferroni correction, it uses the over-estimate

$$\widehat{\text{BAC}}_W(t) = \sum_{t'=1}^{t} \hat{\alpha}_{t'} \mathbb{1}\left\{t - W \le t' + \Delta_{t'} \le t\right\} \tag{3}$$

and selects $\hat{\alpha}_t$ such that

$$\widehat{\text{BAC}}_W(t) \le \alpha. \tag{4}$$

See supporting information for a proof that aACP-BAC achieves the nominal rate.

Alternatively, we can think of aACP-BAC as an alpha-investing procedure (Foster and Stine, 2008) that begins with an alpha-wealth of α, spends it when a family of tests is launched, and earns it back when the family leaves the current window. Thus, aACPs that control BAC over window size $W = \infty$ inevitably have low power to approve later modifications because they only spend but never earn alpha-wealth. This is analogous to the so-called "alpha-death" issue that occurs in procedures that control online FWER (Ramdas et al., 2017). We sidestep the issue of alpha-death by selecting a reasonable value for W.

[Figure 4 about here.]

4.2 *Bad approval and benchmark ratios*

If the goal is to ensure that the SaMD improves on average and occasional drops in performance are tolerated, the approval policies for controlling BAC_W can be overly strict and unnecessarily conservative. There are two solutions to this problem. One approach (*reward-approach*) is to reward the company for each superior model by resetting the level alpha.

The FDA essentially uses this procedure right now, as each clinical trial resets the alpha-spending clock. Another idea (*FDR-approach*) is to draw on the false discovery rate (FDR) literature: These procedures control the expected proportion of false rejections rather than the FWER, which has higher power when some of the null hypotheses are false (Benjamini and Hochberg, 1995). This section defines a second online error rate based on these ideas.

The *reward-approach* punishes bad approvals and rewards the approval of superior models. To signify that a recent set of modifications has led to the creation of a superior model, we now define aACPs with an additional function that can label models as "benchmarks." By labeling a model as a benchmark, the aACP is claiming that it is *superior* to the previous benchmark. More formally, we define a benchmark function B_t as a $\tilde{F}_t$ measurable function that outputs the index of the latest benchmark model at time t. For $t = 0$, we have $B_0 \equiv 0$. We require benchmarks to be a previously approved model since superiority implies acceptability. Again, we use the hat notation to indicate the realized benchmark index. A bad benchmark is one in which $\hat{f}_{\hat{B}_{t-1}} \not\twoheadrightarrow_{0,t} \hat{f}_{\hat{B}_t}$. We do not compare against all previous benchmarks since $\not\twoheadrightarrow_{0,t}$ is a transitive property when the superiority graph is constant.

Based on the *FDR-approach*, we now introduce bad approval and benchmark ratios. An aACP needs to control both ratios to control the frequency of bad approvals and benchmarks.

Definition 3: For NI margin ϵ, the bad approval ratio within W-window at time T is

$$\mathrm{BAR}_W(T) = \frac{\sum_{t=1\vee(T-W)}^{T} \mathbb{1}\left\{\exists t' = 1, \ldots, t-1 \text{ s.t. } \hat{f}_{\hat{A}_{t'}} \not\twoheadrightarrow_{\epsilon,t} \hat{f}_{\hat{A}_t}\right\}}{1 + \sum_{t=1\vee(T-W)}^{T} \mathbb{1}\left\{\hat{B}_t \neq \hat{B}_{t-1}\right\}}. \tag{5}$$

The bad benchmark ratio within W-window at time T is

$$\mathrm{BBR}_W(T) = \frac{\sum_{t=1\vee(T-W)}^{T} \mathbb{1}\left\{\hat{f}_{\hat{B}_{t-1}} \not\twoheadrightarrow_{0,t} \hat{f}_{\hat{B}_t}\right\}}{1 + \sum_{t=1\vee(T-W)}^{T} \mathbb{1}\left\{\hat{B}_t \neq \hat{B}_{t-1}\right\}}. \tag{6}$$

Since only approved models can be designated as benchmarks, BAR_W is an upper bound for the proportion of bad approvals (this is approximate because the denominator is off by one).

The denominator in (5) was deliberately chosen to be the number of unique benchmarks rather than the number of approvals because the latter is easy to inflate artificially. We can simply propose models by alternating between two models that are ϵ-acceptable to each other. This strategy does not work for benchmarks because they require demonstrating superiority.

4.2.1 *aACP to control bad approval and benchmark ratios.* Instead of controlling the expectations of (5) and (6), we describe aACP-BABR for controlling the modified expected bad approval and benchmark ratios. These modified ratios are based on a similar quantity in the online FDR literature known as modified online FDR (Foster and Stine, 2008). We chose to control the modified versions because they can be controlled under less restrictive conditions and using relatively intuitive techniques (Ramdas et al., 2017). Moreover, Foster and Stine (2008) found that modified online FDR has similar long-running behavior to online FDR. We define modified expected bad approval and benchmark ratios below.

Definition 4: For NI margin ϵ, the modified expected bad approval ratio within W-window at time T is

$$\text{meBAR}_W(T) = \frac{E\left[\sum_{t=1\vee(T-W)}^{T} \mathbb{1}\left\{\exists t' = 1,\ldots,t-1 \text{ s.t. } \hat{f}_{\hat{A}_{t'}} \nrightarrow_{\epsilon,t} \hat{f}_{\hat{A}_t}\right\}\right]}{E\left[1 + \sum_{t=1\vee(T-W)}^{T} \mathbb{1}\left\{\hat{B}_t \neq \hat{B}_{t-1}\right\}\right]}. \tag{7}$$

The modified expected bad benchmark ratio within W-window at time T is

$$\text{meBBR}_W(T) = \frac{E\left[\sum_{t=1\vee(T-W)}^{T} \mathbb{1}\left\{\hat{f}_{\hat{B}_{t-1}} \nrightarrow_{0,t} \hat{f}_{\hat{B}_t}\right\}\right]}{E\left[1 + \sum_{t=1\vee(T-W)}^{T} \mathbb{1}\left\{\hat{B}_t \neq \hat{B}_{t-1}\right\}\right]}. \tag{8}$$

Next, we describe how aACP-BABR uniformly controls $\text{meBAR}_W(\cdot)$ and $\text{meBBR}_W(\cdot)$ at levels α and α', respectively (Algorithm 2). We begin with its skeleton and then discuss the alpha-investing procedure. Again, we assume the distributions $\mathbb{P}_t$ are constant.

aACP-BABR uses the acceptability tests from aACP-BAC to approve modifications and superiority tests to discover benchmarks. So at time t, in addition to launching a family of acceptability tests to evaluate model $\hat{f}_t$ for approval, aACP-BABR also launches a family of group-sequential superiority tests comparing $\hat{f}_t$ to models with indices $\left\{\hat{B}_{t-1}, ..., t-1\right\}$, which are executed in a gate-keeping fashion from oldest to latest. Let Δ'_t be the maximum wait time for the superiority tests, which can differ from the maximum wait time for acceptability tests. A model $\hat{f}_j$ is designated as a new benchmark at time t if it demonstrates superiority to models $\hat{f}_{\hat{B}_{j-1}}, ..., \hat{f}_{\hat{B}_{t-1}}$. If multiple benchmarks are discovered at the same time, the aACP can choose any of them (we choose the oldest one in our implementation).

aACP-BABR uses an alpha-investing procedure based on Ramdas et al. (2017) to control the error rates. Let the $\tilde{F}_t$-measurable function α'_t specify the level to perform superiority tests launched at time t. At time t, aACP-BABR constructs over-estimates of the error rates $\text{BAR}_W(t)$ and $\text{BBR}_W(t)$ and selects $\hat{\alpha}_t$ and $\hat{\alpha}_{t'}$ such that the over-estimates are no larger than the nominal levels. The over-estimates are

$$\widehat{\text{BAR}}_W(t) = \frac{\sum_{t'=1}^{t} \hat{\alpha}_{t'} \mathbb{1}\left\{t - W \le t' + \Delta_{t'} \le t\right\}}{1 + \sum_{t'=1\vee(t-W)}^{t} \mathbb{1}\{\hat{B}_{t'} \ne \hat{B}_{t'-1}\}} \tag{9}$$

$$\widehat{\text{BBR}}_W(t) = \frac{\sum_{t'=1}^{t} \hat{\alpha}'_{t'} \mathbb{1}\left\{t - W \le t' + \Delta'_{t'} \le t\right\}}{1 + \sum_{t'=1\vee(t-W)}^{t} \mathbb{1}\{\hat{B}_{t'} \ne \hat{B}_{t'-1}\}}. \tag{10}$$

It selects $\hat{\alpha}_t$ and $\hat{\alpha}'_t$ such that

$$\widehat{\text{BAR}}_j(t) \le \alpha \quad \forall j = 1, ..., W \tag{11}$$

$$\widehat{\text{BBR}}_j(t) \le \alpha' \quad \forall j = 1, ..., W. \tag{12}$$

(We consider all window sizes since we also need to over-estimate future errors $\text{BAR}_W(t')$ and $\text{BBR}_W(t')$ for $t' > t$.) So, aACP-BABR earns alpha-wealth when new benchmarks are discovered, which unites ideas from *FDR-approach* and *reward-approach.* See the supporting information for a proof that aACP-BABR provides the desired error control.

4.3 *Effect of time trends*

Time trends are likely to occur when an aACP is run for a long time. We now discuss how robust aACP-BAC and -BABR are to time trends. We consider levels of increasing severity: the distributions are relatively constant over time (*no-trend*), the distributions are variable but the acceptability graphs are relatively constant (*graph-constant*), and the acceptability graphs change frequently (*graph-changing*).

When the distributions are relatively constant over time, aACP-BAC and -BABR should approximately achieve their nominal error rates. Recall that the two aACPs perform paired T-tests by approximating the distribution $\mathbb{P}_{t:t+D-1}$ with monitoring data sampled from $\mathbb{P}_{j \vee j':t-1}$, which is reasonable when the distributions are relatively constant over time. When there are multiple endpoints, one can either choose a GST that rejects the null hypothesis when all endpoints surpass the significance threshold at the same interim time point or when the endpoints surpass their respective thresholds at any interim timepoint. The former approach is more robust to time trends with only modest differences in power (Asakura et al., 2014).

When the distributions are not constant but the acceptability graphs are, the GSTs have inflated error rates since they only guarantee Type I error control under the strong null. To handle heterogeneity in distributions over time, we can instead use combination tests, such as Fisher's product test and the inverse normal combination test, to aggregate results across time points (Fisher, 1932; Hedges and Olkin, 1985). Since we assumed that the acceptability graphs are constant, this tests the null hypothesis that the shared acceptability graph does not have a particular edge (i.e. $H_0 : f \nrightarrow_{\epsilon,\cdot} f'$); The alternative hypothesis is that the edge exists. Thus, we can replace GSTs with combination tests to achieve the desired error control.

The most severe time trend is where the acceptability graphs change frequently. Controlling error rates in this setting is difficult because previous data is not informative for future time points. In fact, even bad approvals are not well-defined since the relative performance of models changes over time, e.g. an approval at time t that looks bad at time $t+1$ might turn out to be a very good at time $t+2$. As such, we recommend checking that the acceptability graphs are reasonably constant to ensure proper use of aACP-BAC and -BABR. For example, one could track a moving average for the evaluation metrics of all previously proposed modifications and check that their values and/or their relative orderings are stable over time.

5. Cumulative utility of an aACP

Just as hypothesis tests are judged by their Type I error and power, aACPs should be judged by their rates for approving bad and good modifications. Taking a decision-theoretic approach, we characterize the rate of good approvals as the cumulative mean of an endpoint, which we refer to as "cumulative utility." This quantity is similar to "regret" in the online learning literature (Shalev-Shwartz, 2012).

Definition 5: The cumulative utility of an aACP with respect to endpoint m is

$$E\left[\frac{1}{T}\sum_{t=1}^{T} m\left(\hat{f}_{\hat{A}_t}, \mathbb{P}_t\right)\right]. \tag{13}$$

There is no single aACP that maximizes (13) for all possible model developers since we allow arbitrary unblinded adaptations. Instead, we suggest running simulation studies under probable model improvement rates to understand the cumulative utility under different aACP settings, such as window size W, NI margin ϵ, and monitoring data batch size.

6. Simulations

Through simulation studies, we evaluate the operating characteristics of the following aACPs:

(1) Blind: Approve all model updates

(2) Reset: Perform an acceptability test at level 0.05 against the last-approved model

(3) Baseline: Perform an acceptability test at level 0.05 against the initial model

(4) aACP-BAC at level $\alpha = 0.2$ with window $W = 15$

(5) aACP-BABR at level $\alpha = \alpha' = 0.2$ with window $W = 15$. The ratio of maximum wait times between the benchmark and approval was fixed at $\Delta'/\Delta = 2$.

(6) Fixed: Only approve the first model

The first three aACPs have no error rate guarantees but are policies one may consider; The others provide error rate control. In the first two simulations, we try to inflate the error rates of the aACPs. The next two study the cumulative utility of the aACPs when proposed models are improving on average. The last simulation explores the effects of time trends.

In the simulations below, we consider a binary classification task with sensitivity and specificity as metrics. We compare aACPs by plotting the metrics of the approved model over time. We test for acceptability/superiority using repeated confidence intervals (Cook, 1994) and Pocock alpha-spending functions (Pocock, 1977), where $\epsilon = 0.05$ for both endpoints. Supporting Information contains summary statistics of aACP performance (Table 1 and 2), simulation details (Section B), and sensitivity analyses to hyperparameters W and ϵ (Section C).

6.1 *Incremental model deterioration*

In this simulation, the proposed models deteriorate gradually. This can occur in practice for a number of reasons. For instance, a manufacturer might try to make their SaMD simpler, cheaper, and/or more interpretable by using fewer input variables, collecting measurements through other means, or training a less complex model. Even if their modifications are well-intended, the sponsor might end up submitting inferior models. A model developer can also inadvertently propose adverse modifications if they repeatedly overfit to the training data. Finally, a properly trained model can be inferior if the training data is not representative of future time points if, say, the biomarkers lose their prognostic value over time.

This simulation setup tries to induce bio-creep by submitting models that are acceptable to the currently approved model but gradually deteriorate over time. Each proposed model is worse by $\epsilon/2$ in one endpoint and better by $\epsilon/4$ in the other. By alternating between deteriorating the two endpoints, the manufacturer eventually submits strictly inferior models.

[Figure 5 about here.]

Bio-creep occurs consistently when using the aACP-Reset since it only compares against the most recently approved model (Figure 5). Both the sensitivity and specificity for the approved model at the final time point are significantly worse than the initial model. aACP-

BAC and aACP-BABR properly controlled the occurence of bio-creep since they require modifications to demonstrate acceptability with respect to *all* previously approved models.

6.2 *Periodic model deterioration and improvement*

Next we consider a simulation in which the proposed modifications periodically decline and improve in performance. This scenario is more realistic than the previous section since a manufacturer is unlikely to only submit bad modifications. More specifically, the proposed models monotonically improve in performance over the first fifteen time points and, thereafter, alternate between deteriorating and improving monotonically every ten time points.

As expected, aACP-Baseline had the worst error and cumulative utility. It performed like aACP-Blind and the performances of the approved models were highly variable over time (Figure 6). In contrast, the other aACPs displayed much less variability and the performances were generally monotonically increasing. aACP-Reset had the highest utility here because it performs hypothesis tests at a higher level alpha than aACP-BAC and aACP-BABR.

[Figure 6 about here.]

6.3 *Accumulating data model updates*

We now suppose the manufacturer automatically generates modifications by training the same model on accumulating monitoring data. In this simulation, the developer iteratively performs penalized logistic regression. Since model parameters are estimated with increasing precision, the expected improvement decreases over time and performance eventually plateaus. As such, we investigate aACP behavior over a shorter time period.

aACP-Blind approved good modifications the fastest (Figure 7). We find that the remaining aACPs, excluding aACP-fixed, are close in cumulative utility because less efficient aACPs can “catch up”: Even if an aACP fails to approve a small improvement, there will eventually be a proposal with sufficiently large improvement that is easy to discern. aACP-BABR often discovered one or no new benchmarks within a window and was unable to earn alpha-wealth much of the time because the models improved at a slow pace and performance plateaued over time. As such, aACP-BAC and -BABR behaved similarly in this simulation.

[Figure 7 about here.]

6.4 *Significant model improvements*

Next, we simulate a manufacturer that proposes models with large improvements in performance at each time point. Large improvements usually occur when the modifications significantly change the model, such as adding a highly informative biomarker or replacing a simple linear model with a complex one that accounts for non-linearities and interactions.

Since large improvements are relatively rare, we used a short total time. We designed the simulation to be less favorable for aACP-BAC and -BABR. The model developer proposes a modification that improves both endpoints by 4% compared to the most recently approved model. Therefore an aACP cannot catch up by simply waiting for large improvements.

[Figure 8 about here.]

As expected, Blind-aACP is the most efficient, followed by aACP-Baseline and aACP-Reset (Figure 8). The aACPs with error rate control are less efficient. For example, the performance of the final models approved by aACP-BABR and aACP-Reset differed by 4% on average. Unlike in previous simulations, there is a clear improvement in efficiency from using aACP-BABR over aACP-BAP. Since the models here improve at a fast pace, aACP-BABR earns enough alpha-wealth to discover new benchmarks with high probability.

6.5 *Robustness to time trends*

Finally, we evaluate the robustness of aACP-BAC and -BABR to time trends by simulating the three time trend severity levels from Section 4.3. We simulate the endpoints of the proposed models to follow a sinusoidal curve. In addition, the proposed model is always strictly inferior to the currently approved model on average. For the *graph-constant* setting, the sinusoids are aligned so that the proposed model is unacceptable at all time points. For the *graph-changing* setting, the sinusoids are offset by exactly half the period so that the proposed model is superior to the currently approved model at certain time points.

The error rates in the *graph-constant* and *no-trend* settings were similar (Table 2 in Supporting Information), which implies that aACP-BAC and -BABR still control error rates if the acceptability graphs stay relatively constant. However, they performed poorly in the *graph-changing* setting since bad modifications appeared superior at particular time points.

7. Discussion

In this work, we have presented and evaluated different policies for regulating modifications to AI/ML-based SaMDs. One of our motivations was to investigate the possibility of bio-creep, due to the parallels between this problem and noninferiority testing of new drugs. We found that the risk of bio-creep is heightened in this regulatory problem compared to the traditional drug development setting because software modifications are easy and fast to deploy. Nonetheless, we show that aACPs with appropriate online error-rate guarantees can sufficiently reduce the possibility of bio-creep without substantial sacrifices in our ability to approve beneficial modifications, at least in the specific settings discussed in this paper.

This paper only considers a limited scope of problems and there are still many interesting directions for future work. One direction is to develop more efficient aACPs, perhaps by spending alpha-wealth more judiciously, discovering benchmarks using a different procedure, sequestering monitoring data for repeated testing, or considering the special case with pre-specified modifications. Also, we have not considered aACPs that regulate modifications to SaMDs that are intended to treat and are evaluated based on patient outcomes.

Our results raise the interesting question regarding the general structure of the regulatory policy framework. Although aACP-BAC and -BABR mitigate the effect of bio-creep, they cannot provide indefinite error rate control without large sacrifices in cumulative utility. So if one desires both indefinite error rate control and fast approval of good modifications, perhaps the solution is not to use a fully automated approach. For example, human experts could perform comprehensive analyses every couple of years and the manufacturer could use an aACP in between to quickly deploy modifications.

Finally, we highlight that regulating modifications to AI/ML-based SaMDs is a highly complex problem. This paper has primarily focused on the idealized setting of a constant diagnostic environment. Our findings suggest that problems with bio-creep is more pervasive when modifications are designed to accommodate time trends in the patient population, available measurements, and bioclinical practice. It is crucial that we thoroughly understand the safety risks before allowing modifications in these more complex settings.

Acknowledgements We thank Ali Shojaie, Frederick A. Matsen IV, Holly Janes, Pang Wei Koh, and Qi Liu for helpful discussions and suggestions. This work was supported by NIH DP5OD019820.

Data Availability Statement Code to generate data and reproduce all tables and figures are available as Supporting Information as well as http://github.com/jjfeng/aACP.

References

Altman, D. G. and Royston, J. P. (1988). The hidden effect of time. *Stat. Med.* **7,** 629–637.

Asakura, K., Hamasaki, T., Sugimoto, T., Hayashi, K., Evans, S. R., and Sozu, T. (2014). Sample size determination in group-sequential clinical trials with two co-primary endpoints. *Stat. Med.* **33,** 2897–2913.

Benjamini, Y. and Hochberg, Y. (1995). Controlling the false discovery rate: A practical and powerful approach to multiple testing. *J. R. Stat. Soc. Series B Stat. Methodol.* **57,** 289–300.

Bloch, D. A., Lai, T. L., Su, Z., and Tubert-Bitter, P. (2007). A combined superiority and non-inferiority approach to multiple endpoints in clinical trials. *Stat. Med.* **26,** 1193–1207.

Bloch, D. A., Lai, T. L., and Tubert-Bitter, P. (2001). One-sided tests in clinical trials with multiple endpoints. *Biometrics* **57,** 1039–1047.

Blum, A. and Hardt, M. (2015). The ladder: A reliable leaderboard for machine learning competitions. *International Conference on Machine Learning* **37,** 1006–1014.

Cook, R. J. (1994). Interim monitoring of bivariate responses using repeated confidence intervals. *Control. Clin. Trials* **15,** 187–200.

DeMets, D. L. and Lan, K. K. (1994). Interim analysis: the alpha spending function approach. *Stat. Med.* **13,** 1341–52; discussion 1353–6.

Dmitrienko, A. and Tamhane, A. C. (2007). Gatekeeping procedures with clinical trial applications. *Pharm. Stat.* **6,** 171–180.

Dwork, C., Feldman, V., Hardt, M., Pitassi, T., Reingold, O., and Roth, A. (2015). The reusable holdout: Preserving validity in adaptive data analysis. *Science* **349,** 636–638.

FDA (2017). Multiple endpoints in clinical trials guidance for industry. Technical report.

FDA (2018). FDA permits marketing of clinical decision support software for alerting providers of a potential stroke in patients.

FDA (2019). US FDA artificial intelligence and machine learning discussion paper. Technical report.

Fisher, R. A. (1932). *Statistical Methods for Research Workers.* Oliver and Boyd, Edinburgh, 4th edition.

Fleming, T. R. (2008). Current issues in non-inferiority trials. *Stat. Med.* **27,** 317–332.

Foster, D. P. and Stine, R. A. (2008). α-investing: a procedure for sequential control of expected false discoveries. *J. R. Stat. Soc. Series B Stat. Methodol.* **70,** 429–444.

Gottlieb, S. (2019). FDA working to lift barriers to generic drug competition. Accessed: 2019-8-8.

Gupta, H., Kumar, S., Roy, S. K., and Gaud, R. S. (2010). Patent protection strategies. *J. Pharm. Bioallied Sci.* **2,** 2–7.

Heagerty, P. J. and Zheng, Y. (2005). Survival model predictive accuracy and ROC curves. *Biometrics* **61,** 92–105.

Hedges, L. V. and Olkin, I. (1985). *Statistical Methods for Meta-Analysis.* Academic Press.

Hitchings, A. W., Baker, E. H., and Khong, T. K. (2012). Making medicines evergreen. *BMJ* **345,** e7941.

International Conference on Harmonisation (1998). Ethnic factors in the acceptability of foreign clinical data (ICH E5).

Javanmard, A. and Montanari, A. (2015). On online control of false discovery rate.

Offen, W., Chuang-Stein, C., Dmitrienko, A., Littman, G., Maca, J., Meyerson, L., Muirhead, R., Stryszak, P., Baddy, A., Chen, K., Copley-Merriman, K., Dere, W., Givens, S., Hall, D., Henry, D., Jackson, J., Krishen, A., Liu, T., Ryder, S., Sankoh, A. J., Wang, J., and Yeh, C.-H. (2007). Multiple co-primary endpoints: Medical and statistical solutions: A report from the multiple endpoints expert team of the pharmaceutical research and manufacturers of america. *Drug Inf. J.* **41,** 31–46.

Pepe, M. S. (2003). *The Statistical Evaluation of Medical Tests for Classification and Prediction.* Oxford University Press.

Pocock, S. J. (1977). Group sequential methods in the design and analysis of clinical trials. *Biometrika* **64,** 191–199.

Ramdas, A., Yang, F., Wainwright, M. J., and Jordan, M. I. (2017). Online control of the false discovery rate with decaying memory. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30*, pages 5650–5659. Curran Associates, Inc.

Ramdas, A., Zrnic, T., Wainwright, M., and Jordan, M. (2018). SAFFRON: an adaptive algorithm for online control of the false discovery rate. *International Conference on Machine Learning* .

Shalev-Shwartz, S. (2012). Online learning and online convex optimization. *Foundations and Trends® in Machine Learning* **4,** 107–194.

Tamhane, A. C., Gou, J., Jennison, C., Mehta, C. R., and Curto, T. (2018). A gatekeeping procedure to test a primary and a secondary endpoint in a group sequential design with multiple interim looks. *Biometrics* **74,** 40–48.

Tang, D., Agarwal, A., O'Brien, D., and Meyer, M. (2010). Overlapping experiment infrastructure: More, better, faster experimentation. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '10, pages 17–26, New York, NY, USA. ACM.

Zrnic, T., Ramdas, A., and Jordan, M. I. (2018). Asynchronous online testing of multiple hypotheses.

Supporting Information Proofs, algorithms, and additional simulation results referenced in Sections 4 and 6 are available with this paper at the Biometrics website on Wiley Online Library.

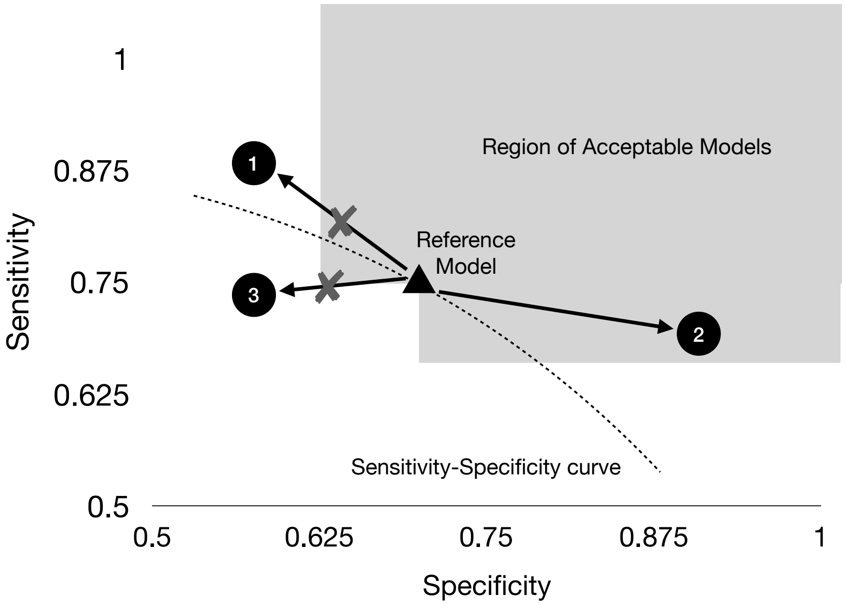

Figure 1. Example of an acceptability graph for binary classifiers evaluated on sensitivity and specificity. Given a reference model (triangle) and NI margin ϵ, a candidate model is acceptable if one endpoint is non-inferior and the other is superior compared to the reference model. The NI margin can be chosen to encourage approval of updates to a better ROC curve. Models in the shaded area are acceptable updates to the reference model. Model 3 is not acceptable since it is on a strictly inferior ROC curve. Model 1 and 2 are likely on better ROC curves, but 1 is not within the NI margin and is therefore not acceptable either.

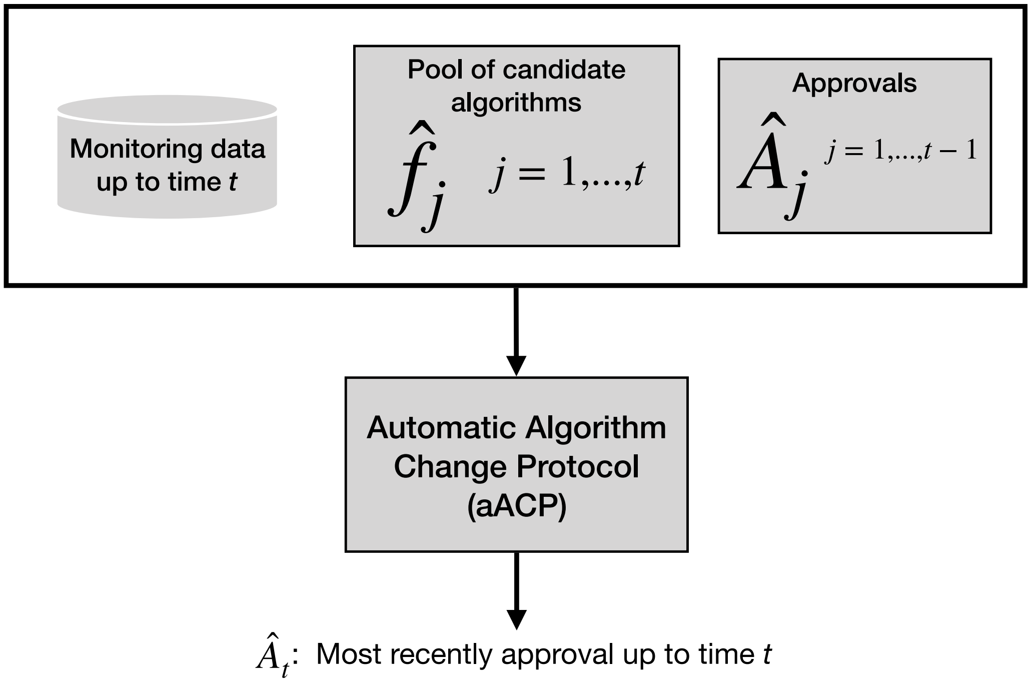
Monitoring data up to time t
Pool of candidate algorithms
$\hat{f}_j$ $j = 1,\ldots,t$
Approvals
$\hat{A}_j$ $j = 1,\ldots,t-1$
Automatic Algorithm Change Protocol (aACP)
$\hat{A}_t$: Most recently approval up to time t

Figure 2. An automatic Algorithm Change Protocol (aACP) outputs the index of the most recently approved model $\hat{A}_t$ at each time t. To do so, it evaluates the pool of candidate models against the pool of previously approved models using monitoring data collected up to that time.

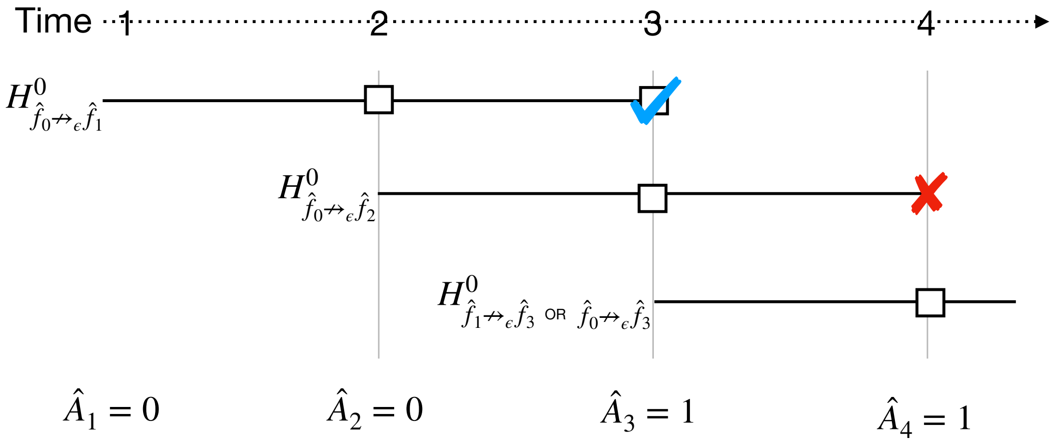
Time
1
2
3
4
$H^0_{\hat{f}_0 \not\rightarrow_\epsilon \hat{f}_1}$
$H^0_{\hat{f}_0 \not\rightarrow_\epsilon \hat{f}_2}$
$H^0_{\hat{f}_1 \not\rightarrow_\epsilon \hat{f}_3 \text{ OR } \hat{f}_0 \not\rightarrow_\epsilon \hat{f}_3}$
$\hat{A}_1 = 0$
$\hat{A}_2 = 0$
$\hat{A}_3 = 1$
$\hat{A}_4 = 1$

Figure 3. At each time point, this simple aACP launches a single group sequential test (GST) comparing the newly proposed model to previously approved models. Here, each model has a maximum wait time of $\Delta = 2$ and each interim analysis is represented by a square. A checkmark indicates that the null hypothesis is rejected and an "X" indicates that the interim analysis is not performed. The final interim analysis for $\hat{f}_2$ is not performed because its GST only compares $\hat{f}_2$ to $\hat{f}_0$ and not the newly approved model $\hat{f}_1$. Thus, $\hat{f}_2$ has no chance of being approved.

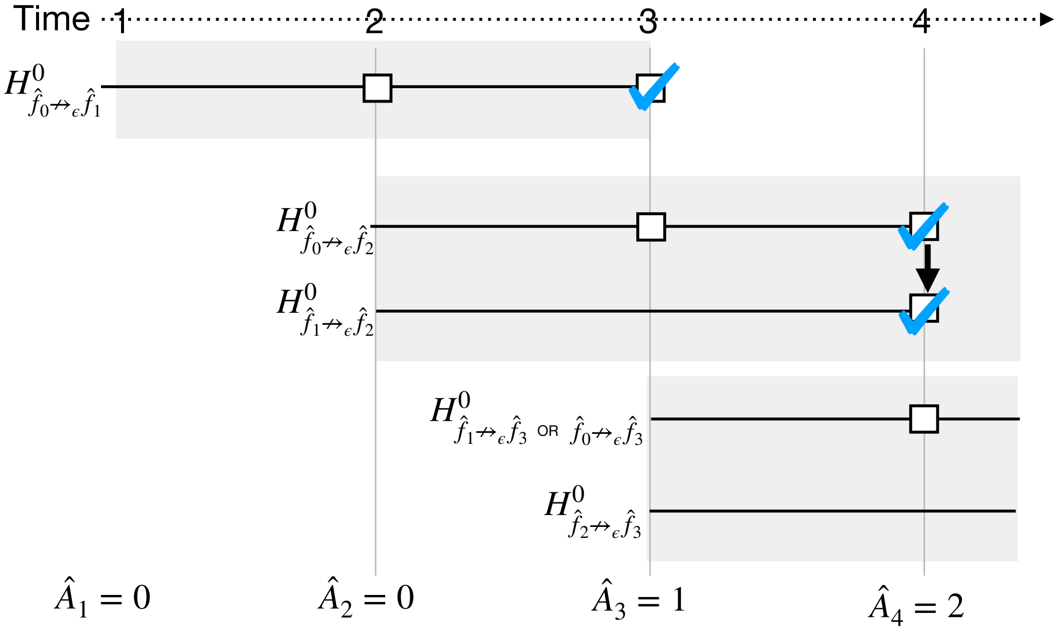
Time
1
2
3
4
$H^0_{\hat{f}_0 \nrightarrow_\epsilon \hat{f}_1}$
$H^0_{\hat{f}_0 \nrightarrow_\epsilon \hat{f}_2}$
$H^0_{\hat{f}_1 \nrightarrow_\epsilon \hat{f}_2}$
$H^0_{\hat{f}_1 \nrightarrow_\epsilon \hat{f}_3 \text{ OR } \hat{f}_0 \nrightarrow_\epsilon \hat{f}_3}$
$H^0_{\hat{f}_2 \nrightarrow_\epsilon \hat{f}_3}$
$\hat{A}_1 = 0$
$\hat{A}_2 = 0$
$\hat{A}_3 = 1$
$\hat{A}_4 = 2$

Figure 4. At each time point, this aACP launches a family of group sequential tests (shaded gray boxes) comparing the newly proposed model to previously approved models as well as other models that might be approved in the interim. Within each family, we test the hypotheses using a gatekeeping procedure, which provides a mechanism for comparing a candidate model to newly approved models in the interim. We use the same notation in Figure 3. An arrow between squares indicates that we rejected a null hypothesis and proceeded to the next test in the gatekeeping sequence.

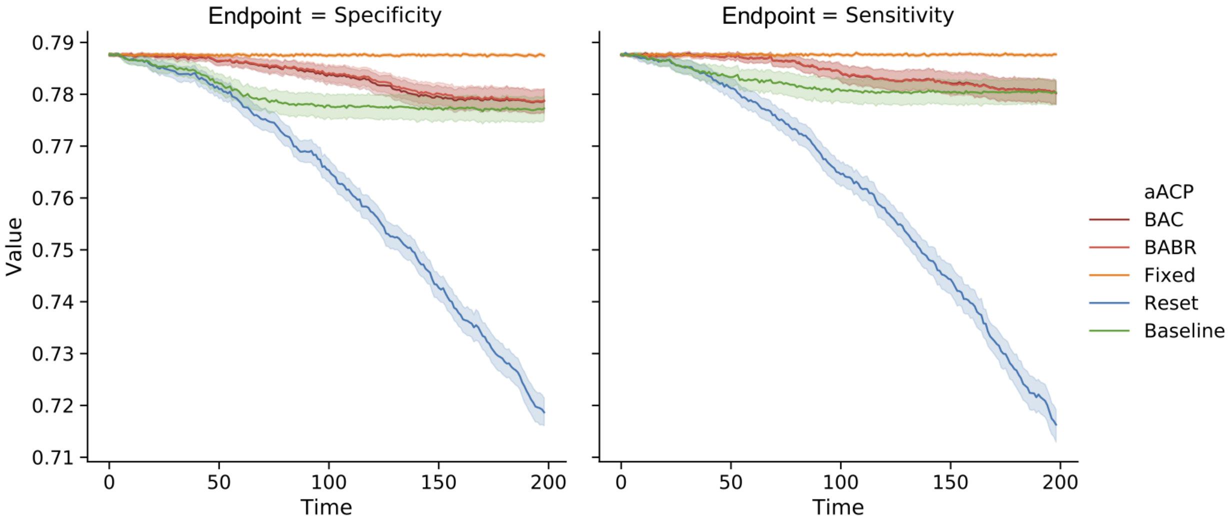

Figure 5. Comparison of the sensitivity and specificity of models approved by different aACPs when the proposed models are gradually deteriorating. (We omit aACP-Blind from this plot since it would obviously perform the worst.)

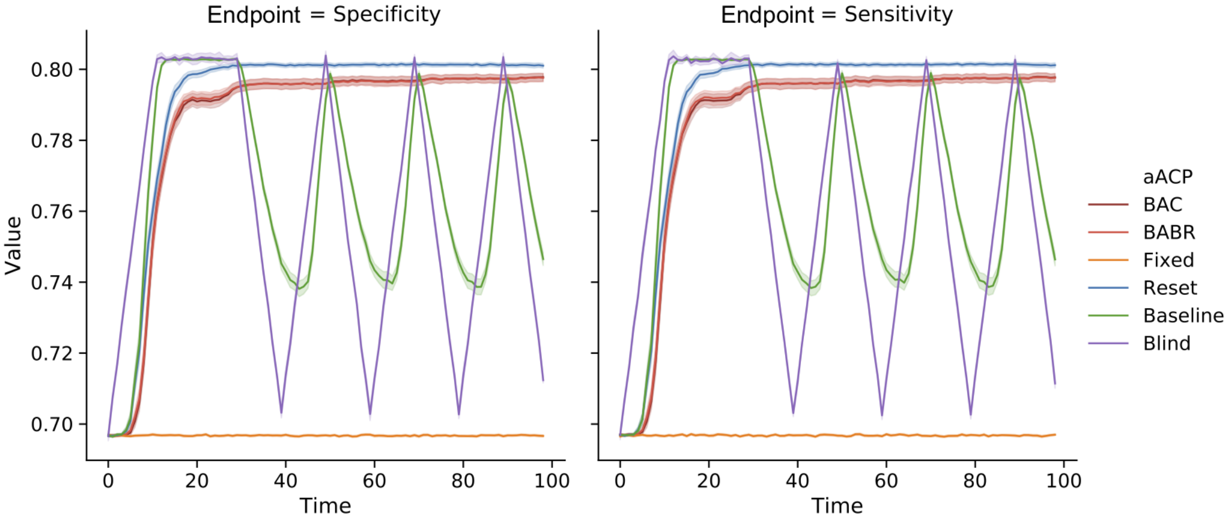

Figure 6. Comparison of the sensitivity and specificity of models approved by different aACPs when the proposed models periodically deteriorate and improve in performance.

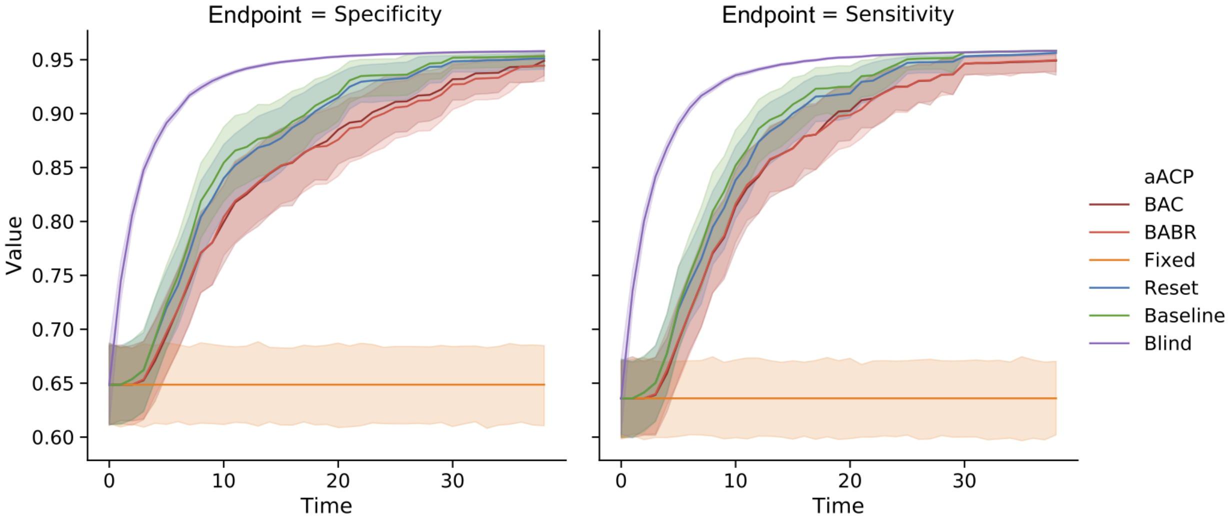

Figure 7. Comparison of the sensitivity and specificity of models approved by different aACPs when the model developer trains a logistic model on accumulating monitoring data.

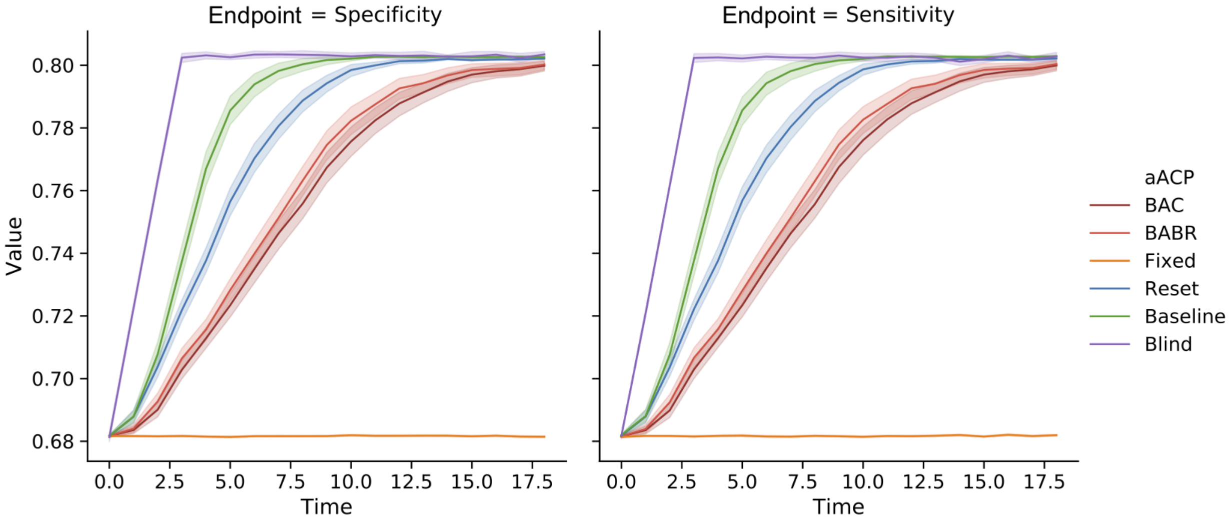

Figure 8. Comparison of the sensitivity and specificity of models approved by different aACPs when the model developer adaptively proposes a significantly better model than the currently approved model. We evaluate three different settings for aACP-BABR, where a larger index means that alpha-wealth is spent more greedily.