Decision Layer: Enhancing Multi-model, Multi Timescale Decisions on the Fly with Online Feedback

Meet Gandhi
Agniva Som
Suraj Satishkumar Sheth
Amrita Singh
mtgandhi@amazon.com
agnivsom@amazon.com
sursheth@amazon.com
amritsk@amazon.com
Amazon Ads
Bangalore, India

ABSTRACT

Rogue actors employ sophisticated automation techniques to mimic human browsing/click patterns and generate invalid (i.e., fraudulent or robotic) traffic on retail marketplaces to artificially inflate their key performance metrics at the expense of their legitimate competitors. To maintain a clean and fair advertising system, it is essential to identify and mitigate ad traffic that is invalid, i.e., fraudulent or coerced or unintended, driven by bad actors and ensure that advertisers do not get charged for invalid traffic (IVT). One major challenge for advertising systems is the absence of complete ground truth fraud labels, even in limited amounts, which makes it challenging to build one single overarching model for comprehensive IVT detection. This generally results in a suite of models, each trying to identify some specific bot modus operandi. While this approach has been beneficial to offer more robust protection to advertisers by catching a variety of bots, it also piled up potentially millions of dollars of lost revenue opportunities, with each algorithm contributing incrementally to false positive detection (i.e., incorrect removal of valid traffic). Hence, we propose to build a "model over models" that learns to maintain *true* IVT coverage of ad fraud detection system while simultaneously lowering the cost of false positives. In this paper, we present a few variations for the new system, trained with incomplete labels that are either high quality but delayed in availability or low quality but available faster. Our proposed online algorithm combines the best of both worlds. It continuously adapts to not only reduce false positive cost by a massive 37% (owing to strong delayed labels), but also to rapidly mitigate revenue loss spikes (owing to weak fast labels) associated with occasional IVT detection system failure scenarios. To this end, we show that the online algorithm has sub-linear regret.

Under review, Online and Adaptive Recommender Systems, August 2023, Long Beach, CA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

KEYWORDS

robot detection, digital advertising, online learning, online convex optimization

ACM Reference Format:
Meet Gandhi, Agniva Som, Suraj Satishkumar Sheth, and Amrita Singh. 2023. Decision Layer: Enhancing Multi-model, Multi Timescale Decisions on the Fly with Online Feedback. In *Proceedings of 3rd International Workshop on Online and Adaptive Recommender Systems (Under review, Online and Adaptive Recommender Systems).* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Sponsored Ads *aka* Sponsored Search *aka* Promoted Listings refer to performance advertising programs that enable advertisers to increase their product visibility and sales on popular e-commerce sites like eBay, Walmart, Flipkart, Amazon, Alibaba, etc. Because of the enormous revenue opportunity, Sponsored advertising is targeted by fraudsters to fulfil their pernicious motives like artificially inflating own earnings, depleting competitor budget, boosting search rankings, etc. Bad actors try to achieve these objectives by sending automated ad traffic to click on ads on sponsored advertising pages while mimicking human browsing behavior as much as possible. Invalid traffic (IVT) is defined as ad traffic that is either fraudulent or involuntary or non-human, and has no value to the advertiser. The role of Traffic Quality (TQ) is to detect and mitigate IVT, so that advertisers are charged only for traffic that is deemed to be valid by a highly precise and high coverage detection system. The goal of such a system is to maintain advertiser trust with comprehensive IVT discovery, and to simultaneously have minimal impact on the online marketplace revenue from incorrect invalidation.

Over time, invalid traffic creators have grown in the sophistication of fraud *modus operandi*. Continual explosion in the scale of IVT, variety of attack vectors and discovery of adversarial attack patterns prompted TQ to steadily respond to the IVT threat by building new algorithms. One major challenge for TQ in this business application is the absence of complete ground truth labels, even in limited amounts, to train complex models with supervision and to measure the true efficacy of any component algorithm. The dearth of labels has been a key inhibitor for TQ to build a single, overarching model for comprehensive IVT detection, since we are

unable to optimize any model training objective to cover for the "unknown and undetected" attack vectors. Hence, over the years, TQ developed a large number of algorithms–heuristic algorithms, machine learning models, deep neural networks, Javascript based client-side telemetry and security engineering based Forensics techniques. All these algorithms were built from a customer-backwards approach with a goal to mitigate every known instance of fraud *modus operandi* and to protect genuine advertiser interests.

Presently, the full suite of TQ invalid detection algorithms comprise of a large variety–some models detect IVT at the granularity of an ad click or impression [8], while other models generate lists of robotic (invalid) entities driving the IVT, like bot user accounts, devices, User Agents and IPs [1]. There is another important angle of variation among the several TQ system algorithms; some algorithms publish invalidation decisions in real-time (<5 milliseconds latency), while many others publish decisions offline ranging from several hours to several days of delay. Whenever any one TQ algorithm marks an ad event (click/impression) as invalid, the event is invalidated and dropped from advertiser budgeting and billing to err on the side of caution. Following the mushrooming of production algorithms for IVT detection, it is hard for TQ to continuously administer all underlying algorithms for optimal performance in the light of the perennially evolving valid and invalid ad traffic patterns. Moreover, each new algorithm deployed in production adds to the detection system's false positive rate (FPR) and revenue loss due to incorrect advertiser charge-back of human ad traffic. These FPR costs, however small in magnitude for one single algorithm, when aggregated over the entire TQ system, bloat up the size of lost revenue opportunity on the sponsored ads programs.

1.1 Need to rectify system decisions

As a consequence of the recurring addition of independent algorithms with so many varied flavors, TQ system has reached a point of diminishing returns with respect to the older generation algorithms. Many TQ algorithms have become less effective incrementally, given that newer generation models are more powerful and can often replicate the majority of detection by the more primitive algorithm. However, it is rarely the situation that any of the legacy algorithms have become entirely (or near 100%) redundant in regards to novel IVT pattern discovery. Deprecating any such legacy algorithm would evidently take a positive step toward lost revenue recovery, but this action will be in direct conflict with the TQ tenet to place advertiser customer interests first. As a solution to this significant missed revenue opportunity, we are proposing a single machine learning (ML) system that combines decisions from component TQ algorithms and predicts whether or not to invalidate the ad traffic event. Obviously, the simplest "model" to increase confidence (i.e., decrease FPR) in marked invalids is the naïve voting mechanism to invalidate an event only if $n(>= 2)$ algorithms have concurred the event as invalid. However, this simple rule drastically worsens the IVT capture rate, making it infeasible to apply in production without the risk of increasing advertiser exposure to IVT.

Current TQ system invalidates an ad event even if a single algorithm flags it as invalid, since greater importance is placed on advertiser protection from IVT than recouping lost revenue due to invalidation FPR. In this paper, we present a modeling framework that functions as a "model over models" (model over binary model decisions to be more precise) to shift to a more well-informed process in making the correct ad click and impression validation decisions. This framework (*aka* decision layer) has the sole aim to reduce overall system FPR, given a strict constraint that *true* IVT detection rate should not drop significantly. We describe further some of the unique and additional burden of challenges accompanying the overall FPR reduction objective. As desired, our decision layer system needs to adapt quickly and dynamically to incoming traffic and IVT trends. Algorithms occasionally misbehave due to any one (or more) among an abundance of known issues like upstream data corruption, faulty deployments, traffic shift due to sale events, bad model configuration etc. During these instances of a large scale TQ system failure where a single or a few algorithms go rogue, the decision layer is required to mitigate the resulting abnormal FPR spikes within a short time period. In order to train the decision layer model to rectify wrong TQ decisions, we wish to train a supervised model with strong human indicators (incomplete labels covering a very small but confident subset of human/good traffic). However, our best choice of confident human labels are available with a long delay (2 days), which is too slow to react to sudden spikes in FPR cost during a system failure scenario. To solve this problem, we propose an online algorithm in this paper that learns from delayed strong labels (to achieve high precision) and also updates model parameters using much faster but weak labels. These weak labels are less precise but directionally accurate to capture the ongoing trend. Our proposed algorithm combines three ideas from the literature, viz., ensemble learning, online convex optimization and mitigation of the effect of label noise induced by the weak labels and demonstrates that it can drastically reduce the FPR in both normal and system failure scenarios without materially deteriorating *true* IVT detection.

Section 2 discusses previous work on key ideas used in the paper. The algorithm and some of its properties are presented in Section 3, followed by performance comparison of the algorithm against other baselines in Section 4. Finally, we conclude and discuss future directions in Section 5.

2 RELATED WORK

This paper incorporates three key ingredients, viz., ensemble learning, online convex optimization and mitigation of the effect of label noise. We review prior work on these ideas in this section.

Ensemble models can be broadly categorized into decision fusion strategies, bagging, boosting and stacking. Fusion strategies, such as unweighted model averaging and majority voting work well when the expert algorithms are comparable [15, 28, 31]. However, these strategies lead to sub-optimal performance as they are biased learners [17]. Stacking, often referred to as *model blending*, is a meta-learning technique to consolidate the output of expert algorithms [32]. Unlike bagging [6], in stacking the models are different (e.g. not all decision trees) and fit on the same data set instead of on random samples of the training data set. Likewise, unlike boosting [10, 11], in stacking a single model learns how to best combine the predictions from contributing models, instead of a sequence of models that correct prior model predictions. As our interest is in

simply combining the expert algorithms, stacking is a natural fit for our setting. A detailed study on recent advances in ensemble methods can be found in [9, 12, 24, 33].

Online convex programming was introduced first by Zinkevich [35] and the worst case regret analysis was first proposed by Gordon [13]. Kalai and Vempala [19] proposed the "Follow the Leader" approach and showed that it leads to doing nearly as well as the best single decision in hindsight. Logarithmic regret algorithms for online convex optimization were first introduced and analysed in Hazan [14]. It was the seminal paper by Shalev-Shwartz and Singer [26] that put forth the primal-dual perspective of the online learning algorithm and introduced a general framework for the design and analysis of online algorithms.

It is well-known that deep neural networks when trained with noisy labels results in poor generalizability [3, 20, 34]. Unfortunately, popular regularization techniques, such as data augmentation [27], Dropout [30], Batch Normalization [16] and weight decay [21] cannot overcome this overfitting problem. Song et al. [29] provided a detailed survey on learning from noisy labels, where the methods are categorized into five buckets: Robust architecture, robust loss function, loss adjustment, sample selection and robust regularization. Menon et al. [23] proposed one such robust regularization technique in which a variant of standard gradient clipping reduces the label noise, which is one of the crucial components of our algorithm.

3 ALGORITHM OVERVIEW

In this section we briefly describe the Online Convex Optimization (OCO) setting, and subsequently propose our algorithm, Online clipped gradient descent using weak and strong labels.

3.1 Online convex optimization

Online Convex Optimization (OCO) can be thought of as a two player game between an adversary and a learner. Let T denote the total number of game iterations. At each iteration $t \in \{1, \ldots, T\}$, the learner chooses a point w_t from a convex set $\mathcal{K}$. After the learner commits to this choice, a convex loss function, $l_t \in \mathcal{L} : \mathcal{K} \rightarrow \mathbb{R}$ is revealed, where $\mathcal{L}$ is the space of loss function and $\mathbb{R}$ is the real line.

Let $\mathcal{A}$ be an algorithm for OCO, that maps the history up to time t to decide the decision point:

$$w_t^{\mathcal{A}} = \mathcal{A}(l_1, l_2, \ldots, l_{t-1}) \in \mathcal{K}.$$

The adversary enjoys an undue advantage of choosing an arbitrary set of loss functions $\{l_t\}_{t=1}^T$. However, the best algorithm is defined as choosing the best point in hindsight, $w \in \mathcal{K}$, fixed across all iterations [13]. Hence, we define the regret of algorithm $\mathcal{A}$ after T iterations as:

$$Regret_{\mathcal{A}}(T) = \sup_{\{l_1,\ldots,l_t\} \subseteq \mathcal{L}} \left\{ \sum_{t=1}^{T} l_t(w_t^{\mathcal{A}}) - \min_{w \in \mathcal{K}} \sum_{t=1}^{T} l_t(w) \right\}$$

3.2 Online clipped gradient descent using weak and strong labels

Online marketplaces log a host of *critical* and *non-critical* features associated with every ad click. *Non-critical* features include device type, page type, logged-in/non-logged-in status, customer membership status (Amazon Prime, Flipkart Plus) etc. *Critical* features are defined as the expert algorithm decisions on a binary scale. Next, we provide a primer on the labeling strategy for model training. We selected two candidate signals for the target variable. First, we use the retail orders on the marketplace to construct a binary label, wherein a session that placed the order and all its clicks in the corresponding order hour are marked as human. This data is generally available in near-real time. However, an order is not always an indisputable indicator of a human session, since the order can get canceled later on due to non-payment, could have been placed by the human part of a compromised account, etc. Henceforth, we refer to this low confidence human label as the noisy or *weak* label. The second labeling option is related to ad-attributed purchases, where the purchased product from the marketplace match the product category, brand etc. of the clicked ad. This signal is clearly more reliable but is available with a delay of more than a day. We refer to this signal as the clean or *strong label* hereon.

We propose a game play between the adversary and the learner, where the learner's action is to set the model weights at the start of every iteration. We start by initializing weights from some offline model. These model weights are updated *twice* every hour, once using the weak labels and once using the strong labels. Thus, the algorithm completes two game iterations every hour.

3.3 Theoretical results on regret bounds

We define $\|x\| = \sqrt{x \cdot x}$ and $D(x, y) = \|x - y\|$. All norms are l^2-norms unless specified otherwise. We define the projection $P(y) = \arg\min_{x \in \mathcal{K}} d(x, y)$. We have d_c number of critical, d_{nc} number of non-critical features and d number of total features, thus, $d = d_c + d_{nc}$ holds trivially. From a feature vector $\phi(x)$ (resp. model weight vector w), we extract its critical and non-critical subvectors as $\phi(x)_c$ (resp. w_c) and $\phi(x)_{nc}$ (resp. w_{nc}).

For the target variable $y \in \{+1, -1\}$, the logistic loss can be defined as $g(w) = -log(u)$, where $u = \sigma(y \cdot w^T \phi(x))$, the probability of the class corresponding to the label y. Here, σ is the sigmoid link function. As $\sigma' = \sigma(1 - \sigma)$, we get $\nabla g(w) = (-1/u \times u \times (1 - u) \times y) \cdot \phi(x)$, where all quantities except the feature vector $\phi(x)$ are real numbers and $\phi(x) \in \mathbb{R}^d$.

As a first step towards defining the algorithm, we propose the loss function as $l_t(w) = \langle w, z_t \rangle$, which is linear in w and hence a convex function. We define z_t for a single data point below, which can be averaged over all points in the batch to compute z_t for a batch. When clean labels are used, we define $z_t \doteq \nabla g_t(w_t) = (1 - u_t) y_t \cdot \phi(x_t)$, i.e., the gradient of the logistic loss function at time t, evaluated using the model weight vector at time t. When noisy labels are used, we define $z_t \doteq \Phi \circ \Psi(w_t, K) = \Phi(-((1/u_t \wedge K) \times u_t \times (1 - u_t) \times y_t) \cdot \phi(x_t))$, using a positive clipping constant K. Here, $(a \wedge b) = min(a, b)$ for $a, b \in \mathbb{R}$ and function $\Psi(w, K)$ computes the gradient of the partial Huberised loss at w_t by clipping only the first term in the gradient of the logistic loss function leaving the remaining terms untouched as mentioned in [23] [1]. Further, map Φ sets the vector components corresponding to the non-critical features as zero.

[1] Partial Huberised loss is not convex. We fix z_t as its gradient evaluated at w_t and define a convex loss function l_t, which is revealed after learner fixes the action w_t.

Lemma 1. *For the loss function $l_t(w) = \langle w, z_t \rangle$ defined on $w \in \mathcal{K}$, the following hold.*

(1) $\|z_t\| \leq K\sqrt{d}$ *for* $\forall t$.
(2) *Function l_t is Lipschitz continuous with Lipschitz constant $K\sqrt{d}$ for $\forall t$.*

Proof. For the clean label as target, $z \doteq \nabla g(w) = (1-u)y \cdot \phi(x)$. Hence, $\|z\|_\infty \leq 1$ as all features are binary. Using the inequality, $\|z\|_2 \leq \sqrt{d}\|z\|_\infty$, we get $\|z\| \leq \sqrt{d}$. For the noisy labels as the target, $z \doteq \Phi \circ \Psi(w, K)$, where K is the clipping constant. Hence, $z = \Phi(-((1/u \wedge K) \times u \times (1-u) \times y) \cdot \phi(x))$. As $\|z\|_\infty \leq K$, we get $\|z\| \leq K\sqrt{d}$.

2. We observe that $l_t(w)$ is differentiable for all $w \in \mathcal{K}$ as it is linear in w. Note that $\nabla l_t(w) = z_t$ and $\|z_t\| \leq K\sqrt{d}$ for $\forall t$. Also, the l^2-norm is the dual norm of itself. Thus, l_t is Lipschitz with respect to the l^2-norm (the norm in the primal space) as the norm of its gradient in the dual space (also the l^2-norm) is bounded. □

As a second step towards building the algorithm, we propose a regularizer and highlight some of its properties.

Lemma 2. *For positive real numbers η_c and η_{nc}, with $\eta_c > \eta_{nc}$, we define the regularizer, $R(w) \doteq \frac{1}{2\eta_c}\|w_c\|^2 + \frac{1}{2\eta_{nc}}\|w_{nc}\|^2$ for $w \in \mathbb{R}^d$; the following hold.*

(1) *R is strongly convex with parameter $1/\eta_c$.*
(2) *R is a Legendre function.*
(3) *The associated Fenchel dual $R^* : \{\nabla R(w) : w \in \mathbb{R}^d\} \to \mathbb{R}$ is:*

$$R^*(x) = \frac{\eta_c}{2}\|x_c\|^2 + \frac{\eta_{nc}}{2}\|x_{nc}\|^2 \tag{1}$$

(4) *The Bregman Divergence associated with the Legendre function R is:*

$$D_R(x, y) = \frac{1}{2\eta_c}\|x_c - y_c\|^2 + \frac{1}{2\eta_{nc}}\|x_{nc} - y_{nc}\|^2 \text{ for } x, y \in \mathbb{R}^d \tag{2}$$

(5) *The Bregman Divergence associated with the Fenchel dual R^* is:*

$$D_{R^*}(x, y) = \frac{\eta_c}{2}\|x_c - y_c\|^2 + \frac{\eta_{nc}}{2}\|x_{nc} - y_{nc}\|^2 \text{ for } x, y \in \mathbb{R}^d \tag{3}$$

Proof. 1. Any function f is η-strongly convex if and only if $\nabla^2 f(x) \geq \eta I$ for all $x \in \text{dom } f$ [5, page 11]. Here, $\nabla^2 f(x) - \eta I$ is positive semi-definite if all eigenvalues of $\nabla^2 f(x)$ be at least η for all x. The hessian of R is a diagonal matrix with $1/\eta_c$ at positions corresponding to the critical features and with $1/\eta_{nc}$ at positions corresponding to non-critical features. Thus, the minimum eigenvalue of the hessian of R is $1/\eta_c$ (as $\eta_c > \eta_{nc}$) and R is strongly convex with parameter $1/\eta_c$.

2. For any arbitrary sequence $\{x_t\} \in \mathcal{B}$, with $x_t \xrightarrow{t\to\infty} \partial B$, note that $\|\nabla R(x_t)\| \xrightarrow{t\to\infty} \infty$ for $\mathcal{B} = \mathbb{R}^d$. Also, as R is a continuous function, its domain $\mathbb{R}^d$ is convex, ∇R is continuous and R is strictly convex, its a Legendre function.

3. By the definition of the Fenchel dual [4, Definition 12.1],

$$\begin{aligned} R^*(x) &= \max_{w \in \mathbb{R}^d} \langle w, x \rangle - R(w) \\ &= \max_{w_c \in \mathbb{R}^{d_c}} \max_{w_{nc} \in \mathbb{R}^{d_{nc}}} w_c^T x_c - \frac{1}{2\eta_c} w_c^T w_c \\ &\quad + w_{nc}^T x_{nc} - \frac{1}{2\eta_{nc}} w_{nc}^T w_{nc} \end{aligned}$$

The equation achieves its maximum at $w_c = \eta_c x_c$ and $w_{nc} = \eta_{nc} x_{nc}$. By substituting them back, we get the desired result.

4. By the definition of Bregman Divergence [4, Definition 8.2],

$$\begin{aligned} D_R(x, y) &= R(x) - R(y) - \nabla R(y)^T (x - y) \\ &= \frac{1}{2\eta_c} x_c^T x_c + \frac{1}{2\eta_{nc}} x_{nc}^T x_{nc} - \frac{1}{2\eta_c} y_c^T y_c - \frac{1}{2\eta_{nc}} y_{nc}^T y_{nc} \\ &\quad - \frac{x_c^T y_c}{\eta_c} + \frac{y_c^T y_c}{\eta_c} - \frac{x_{nc}^T y_{nc}}{\eta_{nc}} + \frac{y_{nc}^T y_{nc}}{\eta_{nc}} \\ &= \frac{1}{2\eta_c}\|x_c - y_c\|^2 + \frac{1}{2\eta_{nc}}\|x_{nc} - y_{nc}\|^2 \end{aligned}$$

5. Similar as above. □

Finally, we present the algorithm, Online clipped gradient descent using weak and strong labels, which, at time t, simply chooses the action that *greedily* minimizes the total loss observed upto time t by virtue of "Follow the Regularized Leader" [22].

Lemma 3. *Starting with arbitrary w_0, iterates at any time $t > 0$ for the algorithm are driven by the recursive equation*

$$w_t = P(\nabla R^*(\nabla R(w_{t-1}) - z_{t-1})).$$

Proof. As the algorithm greedily minimizes the total observed loss so far,

$$\begin{aligned} w_t &\doteq \arg\min_{w \in \mathcal{K}} \sum_{s=1}^{t-1} l_s(w) + R(w) \\ &= \arg\max_{w \in \mathcal{K}} \left\langle -\sum_{s=1}^{t-1} z_s, w \right\rangle + R(w) \\ &= h(\theta_t) \end{aligned}$$

Here, $\theta_t = -\sum_{s=1}^{t-1} z_s$ and $h(\theta) = \arg\min_{w \in \mathcal{K}} \langle -\theta, w \rangle + R(w)$. We define $\widetilde{w}_t \doteq \arg\min_{w \in \mathbb{R}^d} \langle -\theta, w \rangle + R(w)$, the *unconstrained* minimizer of the total loss observed so far, then $w_t = h(\theta_t) = P(\widetilde{w}_t)$, where P is the projection function defined as $P(y) = \arg\min_{x \in \mathcal{K}} D(x, y)$ for the Bregman divergence associated with R as the distance metric [4, Lemma 8.13]. Thus, the projection lemma suggests to first find the unconstrained minimizer of the total loss observed so far and then project it to the convex set $\mathcal{K}$. As $\widetilde{w}_t$ is defined as the unconstrained minimizer of a convex loss function, its gradient evaluated at $\widetilde{w}_t$ should be zero. Hence, $\nabla(\langle -\theta_t, w \rangle + R(w))|_{\widetilde{w}_t} = 0$. Thus, $\widetilde{w}_t = (\nabla R)^{-1}(\theta_t) = \nabla R^*(\theta_t)$ [4, Proposition 12.3]. Hence,

$$\begin{aligned} w_t &= h(\theta_t) = P(\widetilde{w}_t) \\ &= P\big(\nabla R^*(\theta_t)\big) \\ &= P\big(\nabla R^*(\theta_{t-1} - z_{t-1})\big) \\ &= P\big(\nabla R^*(\nabla R(\widetilde{w}_{t-1}) - z_{t-1})\big) \end{aligned}$$

This recursive equation results in a *lazy* version of the algorithm. As described in Zinkevich [35], we replace $\widetilde{w}_{t-1}$ with w_{t-1} to get the *active* version of the algorithm. The geometric interpretation of the recursive equation is given in Figure 1. Here, the model weights, w_t, are iterates in the primal space $\mathcal{K}$, whereas θ_t are the iterates in the dual space $\mathbb{R}^d$. Note that the gradient descent procedure takes place in the dual space. At the start of an iteration, we project the model weights from the primal space to the dual

space ($\theta_{t-1} = \nabla R(w_{t-1})$) and then perform the gradient descent ($\theta_t = \theta_{t-1} - z_{t-1}$) in the dual space. Next, the dual iterates are transferred back to the primal space ($\widetilde{w}_t = \nabla R^*(\theta_t)$) and then projected to the convex set $\mathcal{K}$ ($w_t = P(\widetilde{w}_t)$). □

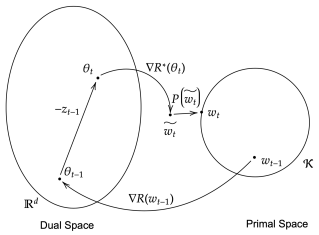

Figure 1: Graphical representation of the algorithm

It is easy to see this recursive equation results in Algorithm 1 for our choice of regularizer. Notice that the learning rates for the critical and non-critical parameter updates turn out to be the corresponding inverses of the regularizer parameters. Next, we state an upper bound on the *regret* of the algorithm in Theorem 1.

THEOREM 1. *For* $\max_{u \in \mathcal{K}} \|u\| \leq r$, *positive clipping constant* K, *and* d *number of features, an upper bound on the regret of the algorithm is:*

$$Regret(T) \leq 4rK\sqrt{dT}.$$

PROOF. As per [4, Lemma 9.5], for convex loss functions, $\{l_t\}_{t=1}^T$ defined over a convex set $\mathcal{K}$, and a Legendre function R defined over $\mathbb{R}^d$ as the regularizer, and $\forall u \in \mathcal{K}$,

$$\sum_{t=1}^{T} \langle w_t - u, z_t \rangle \leq D_R(u, w_1) - D_R(u, w_{T+1}) + \sum_{t=1}^{T} D_R(w_t, \widetilde{w}_{t+1}).$$

where D_R is defined in Lemma 2. Thus, $Regret(T)$

$$\begin{aligned}
&\leq \max_{u \in \mathcal{K}} \left(D_R(u, w_1) - D_R(u, w_{T+1}) + \sum_{t=1}^{T} D_R(w_t, \widetilde{w}_{t+1}) \right) \\
&\leq \max_{u \in \mathcal{K}} \left(D_R(u, w_1) + \sum_{t=1}^{T} D_R(w_t, \widetilde{w}_{t+1}) \right) \\
&= \max_{u \in \mathcal{K}} \left(D_R(u, w_1) + \sum_{t=1}^{T} D_{R^*}(\nabla R(\widetilde{w}_{t+1}), \nabla R(w_t)) \right) \\
&= \max_{u \in \mathcal{K}} \Big(\frac{1}{2\eta_c} \|u_c - w_{1,c}\|^2 + \frac{1}{2\eta_{nc}} \|u_{nc} - w_{1,nc}\|^2 \\
&\quad + \sum_{t=1}^{T} \frac{\eta_c}{2} \| - z_{t,c}\|^2 + \sum_{t=1}^{T} \frac{\eta_{nc}}{2} \| - z_{t,nc}\|^2 \Big) \\
&\leq \frac{2r^2}{\eta_c} + \sum_{t=1}^{T} \frac{\eta_c}{2} \|z_{t,c}\|^2 + \frac{2r^2}{\eta_{nc}} + \sum_{t=1}^{T} \frac{\eta_{nc}}{2} \|z_{t,nc}\|^2 \\
&\leq \frac{2r^2}{\eta_c} + \frac{\eta_c}{2} K^2 dT + \frac{2r^2}{\eta_{nc}} + \frac{\eta_{nc}}{2} K^2 dT \\
&\leq 4rK\sqrt{dT}
\end{aligned}$$

Here, $D_R(w_t, \widetilde{w}_{t+1}) = D_R^*(\nabla R(\widetilde{w}_{t+1}), \nabla R(w_t))$ [7, Lemma 5.1], $D_R(\cdot, \cdot) \geq 0$, $\theta_{t+1} = \theta_t - z_t$ with $\theta_t = \nabla R(\widetilde{w}_t)$, $\|a - b\| \leq 2 \times max(\|a\|, \|b\|)$, $\max_{u \in \mathcal{K}} \|u\| \leq r$, $\|z_t\| \leq K\sqrt{d}$ [Lemma 1], and used η_c and η_{nc} that maximises the expression. □

Upper bound on the regret of the online mirror descent (OMD) algorithm (for the online version of the stacking algorithm with one game iteration each day) is $2r\sqrt{dT}$ as the gradients are bounded by $\sqrt{d}$ [18, 25]. Thus, the proposed algorithm pays the price for clipping the gradients as well as for updating the critical and non-critical parameters using separate learning rates. We again emphasize the main result of the paper that the regret of the proposed algorithm is $O(\sqrt{T})$, which is the same as OMD.

4 RESULTS

4.1 Model performance metrics

We judge model performance on the basis of a few key business metrics. These metrics include (a) a proxy measure of false positive rate (FPR) that measures algorithm precision in catching bot traffic, and (b) robotic recall on a highly confident set of invalid ad clicks. We use two extremely precise robotic signals designed for general algorithm evaluation, using highly confident rules for invalid label assignment to a small portion of ad traffic. All ad clicks from a session having a large number of clicks in a single hour (robot signal R_1), or from an IP with a large click count in a week and extremely low purchase rate (robot signal R_2) are invalid with extremely high confidence. High coverage on these robot signals provides confidence on directionally positive progress made by the system in improving the recall of IVT detection.

To combine the expert algorithm decisions, the first step is to train a stacking logistic regression model using the strong labels [32]. This model is trained daily once on a period of N (chosen according to business objectives) consecutive days of digital advertising data from the sponsored advertising program of a major online retailer. We also trained a stacking logistic regression model on the same N days of advertising data from the same ad program using the weak labels. We compare our algorithm (referred to as the online algorithm, which, at time t, simply chooses the model weights that greedily minimizes the total loss observed upto time t) with these baselines on metrics defined above, along with an additional baseline, where the online algorithm is trained using only weak labels.

We compare the results for two periods: a normal period and a spike period where FPR of one of the expert algorithms suddenly increased because of a faulty model promotion in production. The results are shown in the Table 1 and Table 2 respectively. All metrics are relative to the production system. An algorithm having higher drop on FPR without significant drop on recall on R_1 and R_2 is preferred.

As expected, the stacking model trained using weak labels show inferior performance compared to the stacking model trained using the strong labels on both time periods. Weak labels lead to low FPR reduction (lower precision) and high drop on recall on robotic signals. The online algorithm trained using only weak labels has the highest FPR drop among all algorithms during the normal period. Nevertheless, it has significantly less FPR decrease during

Algorithm 1 Online Clipped Gradient Descent Using Weak and Strong Labels

Input: time horizon $2T$, initial model weights w_0, learning rates $\{\eta_c, \eta_{nc}\}$
for t = 0, 2, 4,..., $2T-2$ **do**:
  $z_t = \nabla(g_t(w_t))$
  $\widetilde{w}_{t+1,c} = w_{t,c} - \eta_c \times z_{t,c}$; $\widetilde{w}_{t+1,nc} = w_{t,nc} - \eta_{nc} \times z_{t,nc}$
  $w_{t+1} = P(\widetilde{w}_{t+1})$
  $z_{t+1} = \Phi \circ \Psi(w_{t+1}, K)$
  $\widetilde{w}_{t+2,c} = w_{t+1,c} - \eta_c \times z_{t+1,c}$; $\widetilde{w}_{t+2,nc} = w_{t+1,nc}$
  $w_{t+2} = P(\widetilde{w}_{t+2})$
End for

the spike period, providing strong motivation to update weights using both strong and weak labels. The proposed algorithm significantly reduces TQ system FPR (37% during the normal period and 51% during the spike period), by rectifying incorrect invalidation decisions without compromising recall on robotic signals. It also reduces 9% more FPR during the normal period and 20% more FPR during the spike period compared to the stacking model trained using the strong labels at similar recall on robotic signals.

Table 1: %Change in metrics over existing system for normal period

Algorithm	FPR	Recall on R1	Recall on R2
Stacking (strong labels)	-31.5%	-1.1%	**-0.6%**
Stacking (weak labels)	-25.3%	-4.1%	-1.1%
Online (weak labels)	**-37.2%**	**-0.9%**	-1.1%
Proposed Algorithm	-36.6%	**-0.9%**	-1.1%

Table 2: %Change in metrics over existing system for spike period

Algorithm	FPR	Recall on R1	Recall on R2
Stacking (strong labels)	-38.8%	-1.6%	-0.8%
Stacking (weak labels)	-22.1%	-1.9%	**-0.7%**
Online (weak labels)	-40.1%	-0.9%	-1.1%
Proposed Algorithm	**-50.8%**	**-0.7%**	-1.4%

We present ablation studies for the choices of the clipping constant K in Table 3 and Table 4. We observe that no single value of the clipping constant performs best on all metrics. Also, FPR reduction is significantly less for the unclipped version ($K = \infty$) signifying the importance of clipping the gradients while updating the weights using the weak labels. We observe that the algorithm with clipping constant as 5 is performs reasonably in reducing the overall FPR of the system without compromising recall on robotic coverage, and is thus our recommendation.

5 CONCLUSION AND FUTURE WORK

In this work, we compare a series of models trained using strong and weak labels to combine decisions of expert algorithms, achieving a large step function reduction in false positives without compromising on IVT detection recall. Our proposed online algorithm

Table 3: Ablation over clipping constant K for normal period

Clipping Constant, K	FPR	Recall on R1	Recall on R2
1	**-37.4%**	**-0.8%**	-1.2%
5	-36.6%	-0.9%	-1.1%
10	-33.9%	-0.9%	-1.0%
20	-25.9%	-1.1%	-0.8%
∞	-10.4%	-1.0%	**-0.4%**

Table 4: Ablation over clipping constant for spike period

Clipping Constant, K	FPR	Recall on R1	Recall on R2
1	-49.7%	-0.8%	**-1.3%**
5	-50.8%	**-0.7%**	-1.4%
10	-53.3%	-0.8%	-1.7%
20	**-53.4%**	-0.9%	-2.5%
∞	-36.1%	-1.0%	-2.3%

reduces system FPR up to a mammoth 37%. It has the same regret $\left(O(\sqrt{T})\right)$ as OMD, and is generic enough to be applied to scenarios with multiple labels of varied quality. We believe that future experiments on separate learning rates for weak and strong labels will help to further boost performance. We would also like to explore two-temperature logistic regression based on Tsallis divergence to update model weights using weak labels [2].

REFERENCES

[1] Rajat Agarwal, Anand Muralidhar, Agniva Som, and Hemant Kowshik. 2022. Self-supervised Representation Learning Across Sequential and Tabular Features Using Transformers. In *NeurIPS 2022 First Table Representation Workshop.* https://openreview.net/forum?id=wIIJlmr1Dsk
[2] Ehsan Amid and Manfred K. Warmuth. 2017. Two-temperature logistic regression based on the Tsallis divergence. *CoRR* abs/1705.07210 (2017). arXiv:1705.07210 http://arxiv.org/abs/1705.07210
[3] Devansh Arpit, Stanisław Jastrzundefinedbski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, and Simon Lacoste-Julien. 2017. A Closer Look at Memorization in Deep Networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70* (Sydney, NSW, Australia) *(ICML'17)*. JMLR.org, 233–242.
[4] Gabor Bartok, David Pal, Csaba Szepesvari, and Istvan Szita. 2011. *Online Learning - CMPUT 654.* Lecture notes.
[5] Dimitri Bertsekas, Angelia Nedic, and Asuman Ozdaglar. 2003. *Convex Analysis and Optimization.* Athena Scientific.
[6] Leo Breiman. 1996. Bagging predictors. *Machine Learning* 24 (1996).
[7] Sébastien Bubeck. 2011. Introduction to Online Optimization.

[8] Sharad Chitlangia, Anand Muralidhar, and Rajat Agarwal. 2022. Self Supervised Pre-training for Large Scale Tabular Data. In *NeurIPS 2022 First Table Representation Workshop*. https://openreview.net/forum?id=BXP02v4tZIL

[9] Thomas G. Dietterich. 2000. Ensemble Methods in Machine Learning. In *Multiple Classifier Systems*. Springer Berlin Heidelberg, Berlin, Heidelberg, 1–15.

[10] Yoav Freund and Robert E. Schapire. 1996. Experiments with a New Boosting Algorithm. In *Proceedings of the Thirteenth International Conference on International Conference on Machine Learning* (Bari, Italy) *(ICML'96)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 148–156.

[11] Jerome H. Friedman. 2001. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics* 29, 5 (2001), 1189 – 1232. https://doi.org/10.1214/aos/1013203451

[12] M.A. Ganaie, Minghui Hu, A.K. Malik, M. Tanveer, and P.N. Suganthan. 2022. Ensemble deep learning: A review. *Engineering Applications of Artificial Intelligence* 115 (oct 2022), 105151. https://doi.org/10.1016/j.engappai.2022.105151

[13] Geoffrey J. Gordon. 1999. Regret Bounds for Prediction Problems. In *Proceedings of the Twelfth Annual Conference on Computational Learning Theory* (Santa Cruz, California, USA) *(COLT '99)*. Association for Computing Machinery, New York, NY, USA, 29–40. https://doi.org/10.1145/307400.307410

[14] Elad Hazan. 2006. *Efficient algorithms for online convex optimization and their applications*. Ph. D. Dissertation. Princeton University.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. 770–778. https://doi.org/10.1109/CVPR.2016.90

[16] Sergey Ioffe and Christian Szegedy. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37* (Lille, France) *(ICML'15)*. JMLR.org, 448–456.

[17] Cheng Ju, Aurelien Bibaut, and Mark Laan. 2017. The Relative Performance of Ensemble Methods with Deep Convolutional Neural Networks for Image Classification. *Journal of Applied Statistics* 45 (04 2017). https://doi.org/10.1080/02664763.2018.1441383

[18] Sham M. Kakade, Shai Shalev-Shwartz, and Ambuj Tewari. 2012. Regularization Techniques for Learning with Matrices. *J. Mach. Learn. Res.* 13, 1 (jun 2012), 1865–1890.

[19] Adam Kalai and Santosh Vempala. 2005. Efficient algorithms for online decision problems. *J. Comput. System Sci.* 71, 3 (2005), 291–307. https://doi.org/10.1016/j.jcss.2004.10.016 Learning Theory 2003.

[20] Jonathan Krause, Benjamin Sapp, Andrew Howard, Howard Zhou, Alexander Toshev, Tom Duerig, James Philbin, and Li Fei-Fei. 2016. The Unreasonable Effectiveness of Noisy Data for Fine-Grained Recognition. In *Computer Vision – ECCV 2016*, Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling (Eds.). Springer International Publishing, Cham, 301–320.

[21] Anders Krogh and John Hertz. 1991. A Simple Weight Decay Can Improve Generalization. In *Advances in Neural Information Processing Systems*, J. Moody, S. Hanson, and R.P. Lippmann (Eds.), Vol. 4. Morgan-Kaufmann. https://proceedings.neurips.cc/paper_files/paper/1991/file/8eefcfdf5990e441f0fb6f3fad709e21-Paper.pdf

[22] Brendan McMahan. 2011. Follow-the-Regularized-Leader and Mirror Descent: Equivalence Theorems and L1 Regularization. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (Proceedings of Machine Learning Research, Vol. 15)*, Geoffrey Gordon, David Dunson, and Miroslav Dudík (Eds.). PMLR, Fort Lauderdale, FL, USA, 525–533. https://proceedings.mlr.press/v15/mcmahan11b.html

[23] Aditya Krishna Menon, Ankit Singh Rawat, Sashank J. Reddi, and Sanjiv Kumar. 2020. Can gradient clipping mitigate label noise?. In *International Conference on Learning Representations*. https://openreview.net/forum?id=rklB76EKPr

[24] D. Opitz and R. Maclin. 1999. Popular Ensemble Methods: An Empirical Study. *Journal of Artificial Intelligence Research* 11 (aug 1999), 169–198. https://doi.org/10.1613/jair.614

[25] Shai Shalev-Shwartz. 2007. *Online learning: theory, algorithms and applications*. Ph. D. Dissertation. The Hebrew University of Jerusalem.

[26] Shai Shalev-Shwartz and Yoram Singer. 2007. A primal-dual perspective of online learning algorithms. *Machine Learning* 69, 2 (2007), 115–142. https://doi.org/10.1007/s10994-007-5014-x

[27] Connor Shorten and Taghi M. Khoshgoftaar. 2019. A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data* 6, 1 (2019). https://doi.org/10.1186/s40537-019-0197-0

[28] Karen Simonyan and Andrew Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). http://arxiv.org/abs/1409.1556

[29] Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. 2022. Learning from Noisy Labels with Deep Neural Networks: A Survey. *IEEE Transactions on Neural Networks and Learning Systems* (2022).

[30] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research* 15, 56 (2014), 1929–1958. http://jmlr.org/papers/v15/srivastava14a.html

[31] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1–9. https://doi.org/10.1109/CVPR.2015.7298594

[32] David H. Wolpert. 1992. Stacked generalization. *Neural Networks* 5, 2 (1992), 241–259. https://doi.org/10.1016/S0893-6080(05)80023-1

[33] Seniha Esen Yuksel, Joseph N. Wilson, and Paul D. Gader. 2012. Twenty Years of Mixture of Experts. *IEEE Transactions on Neural Networks and Learning Systems* 23, 8 (2012), 1177–1193. https://doi.org/10.1109/TNNLS.2012.2200299

[34] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2021. Understanding Deep Learning (Still) Requires Rethinking Generalization. *Commun. ACM* 64, 3 (feb 2021), 107–115. https://doi.org/10.1145/3446776

[35] Martin Zinkevich. 2003. Online Convex Programming and Generalized Infinitesimal Gradient Ascent. In *Proceedings of the Twentieth International Conference on International Conference on Machine Learning* (Washington, DC, USA) *(ICML'03)*. AAAI Press, 928–935.