Prediction Guardian: A Distribution-based Real-time Anomaly Detection and Mitigation Framework for Online Advertising

Anonymous Author(s)

Abstract

Deep learning recommendation models (DLRMs) are integral to real-time ad ranking and delivery at massive scale. As foundation models—trained jointly across tasks and datasets and increasingly used to either replace or distill knowledge into task-specific DLRMs—become standard, the risk of prediction errors escalates. Frequent retraining of foundation models on fresh, multi-source data and continuous deployment to keep pace with shifting user behavior amplify exposure to silent data corruption during training. Additionally, real-time feature pipelines at inference time further increase vulnerability to upstream feature issues. These challenges make foundation models and student DLRMs prone to abrupt distribution shifts or extreme outliers that can propagate downstream and lead to advertiser budget overspend.

We present *Prediction Guardian*, an automated detection and mitigation framework designed to stabilize DLRM predictions by monitoring aggregate prediction distributions, which typically exhibit log-normal characteristics in healthy systems. By continuously comparing real-time distributions to historical baselines with robust metrics such as Jensen–Shannon divergence, the system rapidly identifies anomalies and applies targeted mitigations—including selective smoothing and zeroing out outliers—to prevent downstream harm. Prediction Guardian was evaluated across three corruption scenarios, leftward shifts (underprediction), rightward shifts (overprediction), and tail outlier spikes. The results demonstrate that Prediction Guardian detects prediction anomalies with over 0.80 precision and over 0.85 recall with a 16.7-minute average latency in simulation, and 0.75 precision and over 0.85 recall with a 18.3-minute average latency in historical backtesting. The proposed mitigation method reduced advertiser overcharge by over 90% for tail outliers and over 50% for moderate right-shift anomalies.

ACM Reference Format:
Anonymous Author(s). 2026. Prediction Guardian: A Distribution-based Real-time Anomaly Detection and Mitigation Framework for Online Advertising. In . ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

1.1 Background

Deep learning recommendation models (DLRMs) used in online advertising generate real-time predictions for billions of items each day. These predictions directly influence ad ranking and delivery, shaping downstream business goals such as user engagement and advertiser return on investment. Typically, DLRMs estimate the likelihood of user actions—such as clicks, conversions, or other engagement outcomes—and advertisers' campaign performance depends heavily on the accuracy and calibration of these predictions [7]. When prediction values become abnormally inflated, the system may overestimate expected engagement, causing advertisers to win auctions at prices that do not reflect true value. This mismatch can lead to sudden increases in cost-per-action and rapid, unexpected budget depletion.

Modern ad-serving stacks must support heterogeneous advertiser objectives, including click-through rate (CTR), conversion rate (CVR) for different optimization goals, and various quality metrics. To accommodate this diversity, production systems typically deploy multiple prediction tasks organized in multi-stage pipelines [32]. Traditional recommendation systems rely on hundreds of task-specific DLRMs trained and maintained independently [18, 22]. However, recent work on scaling laws for DLRMs [1] has fundamentally reshaped this design paradigm. For example, VIP5 [6] introduces multimodal foundation models that jointly incorporate visual, textual, and personalization signals within a unified architecture. Similarly, new model architectures have been proposed that exhibit efficient scaling behavior for DLRMs analogous to that observed in large language models [31]. These advances have enabled the emergence of foundation models for recommendation systems [16], which are trained jointly across multiple prediction tasks and diverse datasets. Such models can be deployed directly to serve live traffic [17, 32], or used as teacher models to improve individual DLRMs through knowledge distillation [16].

Despite their substantial benefits, foundation models introduce new vulnerabilities that can significantly degrade prediction quality. Prior works show that even a small number of adversarial or corrupted training samples can disproportionately corrupt foundation models [27, 30]. This risk is further amplified in industrial recommendation system settings, where foundation models must be retrained frequently on fresh data to adapt to continual shifts in user behavior, advertiser strategies, and market dynamics [5, 14, 19]. Frequent retraining on large volume of data from diverse sources, together with the continuous deployment of new model snapshots into production, substantially increases the likelihood of prediction errors—particularly when training data are corrupted or fail silently [21]. Moreover, online advertising systems update user and ad features in real time, making them especially susceptible to upstream feature corruption [26]. Failures in foundation models can either directly degrade the quality of live predictions if they're used for serving directly, or harm student DLRMs through distillation [13]. These faults often cause abrupt distributional shifts or extreme outliers in model predictions, which can propagate through

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
Conference'17, Washington, DC, USA
© 2026 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

downstream auction and delivery mechanisms and result in severe consequences, including large-scale advertiser budget overspend.

1.2 Related Work

1.2.1 Prediction Calibration System. Prediction calibration is a widely adopted approach in large-scale ranking systems to address systematic bias and mis-calibration in model predictions [3, 20]. The primary goal of calibration systems is to ensure that the aggregate of predicted probabilities (such as click-through rate or conversion rate) aligns closely with observed outcomes over defined traffic segments. This is typically achieved by adjusting model predictions after they're generated through calibration multipliers. Limitations of Prediction Calibration Systems include:

- **Latency:** Calibration systems rely on observed outcomes (e.g., conversions), which are only available after a significant delay (often hours or longer). This makes it too slow to respond to sudden, high-impact prediction anomalies that can cause immediate business harm.
- **Granularity:** Adjustments are typically made at a coarse segment level (e.g., by engagement surface or event type), which means localized or partial prediction corruption cannot be caught and mitigated.

1.2.2 Model Validator. The Model Validator is a proactive safeguard designed to evaluate the health of model versions prior to deployment [21, 24]. It operates by evaluating candidate model snapshots against representative data streams, ensuring that only models meeting predefined performance criteria are promoted to production. This process is effective at catching model training time errors, such as training data corruptions, before they can impact live systems.

Despite its utility, Model Validator is inherently limited to the pre-deployment phase. It cannot detect or mitigate issues that emerge after a model is serving live traffic in the inference stage, such as real-time feature corruption. Once deployed, the Model Validator has no visibility or control over the model's behavior in production, leaving the system exposed to runtime failures that can cause abrupt and severe prediction anomalies.

1.3 Foundational Stationarity in Model Prediction

Empirically, production-scale online advertising yield highly right-skewed model predicted probabilities for user–ad interactions that is well-approximated by a log-normal distribution [12] (Figure 1). Importantly, the log-normal distribution pattern remains consistent across different inference periods for well-functioning models. While seasonality can introduce varying degrees of pattern shifts between models and across inference periods, these shifts are expected to remain within defined magnitude thresholds for healthy predictions. Such statistical stability provides a robust foundation for monitoring and anomaly detection in production systems, as significant deviations from the expected distribution can serve as early indicators of model issues.

Here we offer an intuitive explanation for this stable log-normal distribution: Among the vast number of user-item pairs, only a small fraction of users are predicted to have a high likelihood of

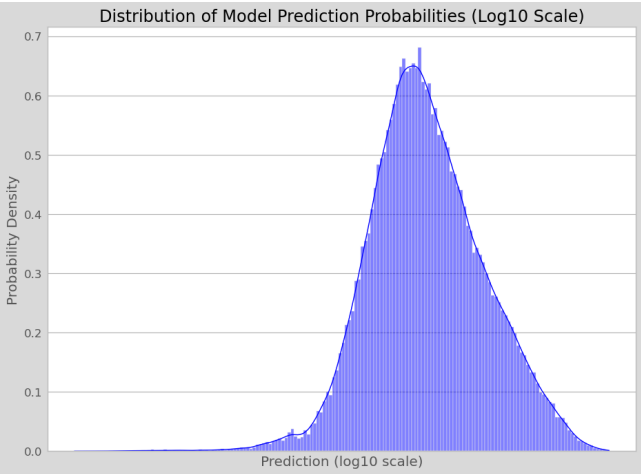


Figure 1: Empirical PDF of predictions generated by a real-world online advertising ranking model

interacting with a given item. As the predicted probability of interaction decreases, the number of users associated with those lower probabilities increases rapidly, until nearly all users are accounted for. This results in a cumulative distribution function (CDF) that closely resembles that of a log-normal distribution (Figure 2), with its derivative (the probability density function) appearing approximately Gaussian in log space.

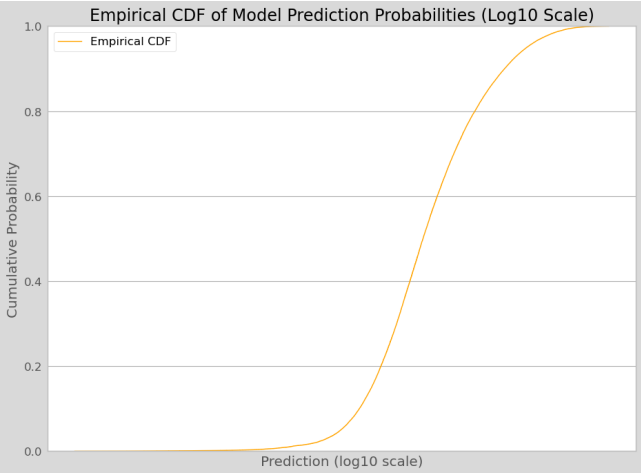


Figure 2: Empirical CDF of predictions generated by a real-world online advertising ranking model

1.4 Our Contributions

In this work, we introduce Prediction Guardian, a novel prediction abnormality auto detection and mitigation framework designed to enhance the robustness and stability of prediction outputs for DLRMs. Key contributions of this work include:

- **Distribution-based anomaly detection:** We propose a real-time monitoring system that leverages the statistical stability of aggregate prediction distributions. By continuously tracking the distribution of model outputs and comparing them to reference (healthy) distributions, our method can rapidly detect significant deviations or outliers indicative of underlying system failures without waiting for user engagement outcomes (e.g. clicks, conversions).
- **Automated mitigation:** Upon detection of anomalous prediction behavior, our framework employs automated mitigation strategies—such as hard and soft zero-out techniques—to adjust or filter corrupted predictions, thereby effectively preventing the damage from these outliers.
- **Empirical validation:** We present extensive empirical evaluations of Prediction Guardian on simulated datasets and offline backtests using historical incident data. Our results show that the framework can detect and mitigate critical prediction anomalies within minutes, significantly reducing business impact compared to existing solutions.
- **Generalizability:** While our primary focus is on online advertising systems, the principles and techniques developed in this work are broadly applicable to any large-scale recommendation or ranking system that relies on real-time model predictions.

Prediction Guardian and Prediction Calibration are complementary techniques for robust ranking systems. Prediction Guardian rapidly detects and corrects sudden prediction shifts by aligning outputs with a historically healthy distribution, while Prediction Calibration ensures long-term accuracy by matching aggregate predictions to real outcomes. In our current design, Prediction Guardian is placed before Calibration so that they offer the combined benefits of immediate prediction anomaly protection, as well as accurate prediction alignment with realized outcomes.

Similarly, Prediction Guardian complements Model Validator by covering different stages: Model Validator addresses issues during pre-deployment, and Prediction Guardian safeguards against anomalies post-deployment. Together, they offer end-to-end protection against prediction errors.

2 Proposed Solution: Distribution-based Guardrail

In order to enforce prediction robustness of DLRMs, Prediction Guardian is a post-hoc guardrail that monitors reference (healthy) and real-time distribution data to detect and correct sudden model prediction distribution drifts by leveraging unlabeled prediction data. Prediction Guardian functions as an independent post-processing layer, positioned outside of ranking models and applied prior to other post-processing steps like calibration (Figure 3).

Briefly, the guardrail stabilizes model predictions by:

(1) Creating estimators of real-time and reference model prediction distributions
(2) Comparing the real-time and reference distribution data to measure distribution shifts
(3) Smoothing out-of-distribution prediction values only when distribution shifts are detected

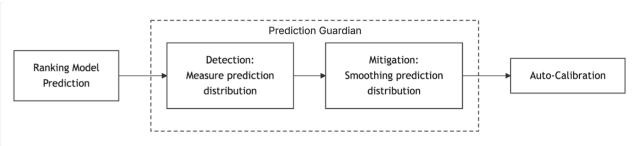


Figure 3: Prediction Guardian detects prediction distribution shifts and mitigates out-of-distribution prediction ranges via smoothing

Formally, consider a model M that takes input X_i and predicts the probability of an event as

$$p_i = M(X_i) \in [0, 1].$$

Furthermore, let $P_{M,t}$ denote the distribution of predictions of model M at time t, and let $P_{M,\text{ref}}$ denote the reference distribution of predictions of model M.

Step 1 constructs $P_{M,t}$ and $P_{M,\text{ref}}$, the real-time and reference distributions, respectively. Step 2 compares $P_{M,t}$ and $P_{M,\text{ref}}$ to alert on significant prediction distribution shifts causing service disruptions. Using the signal from Step 2, Step 3 automatically isolates the out-of-distribution prediction ranges in $P_{M,t}$ and smooths out predictions belonging to these ranges as mitigation. In practice, experts supervise Step 3 to guard against false positives.

2.1 Prediction Distribution Estimation

Prediction Guardian operates on distribution of ranking model predictions to detect and mitigate issues during service disruption events. In order to estimate the distribution, Prediction Guardian bins prediction values from model M into discrete prediction buckets. Furthermore, the system aggregates streamed prediction data to construct $P_{M,t}$ and $P_{M,ref}$, probability mass functions indicating the likelihood a ranking model's prediction falls into a prediction bucket.

Motivated by the property that ranking model outputs approximate a log-normal distribution, Prediction Guardian leverages a logarithmic, prediction binning scheme $B : [0, 1] \to \mathbb{Z}_{\leq 0}$ defined as

$$B(p) = \lfloor 10 \cdot \log_{10}(p) \rfloor$$

Under this binning scheme, a prediction p is defined as *belonging to* the prediction bucket $B(p)$. For instance, a prediction of $p = 0.01$ belongs to prediction bucket -20. While this particular binning model is used for Prediction Guardian based on prior information about the underlying density, the technique is generalizable across other popular binning models [8, 10].

Moreover, Prediction Guardian aggregates real-time and reference prediction data for a model M using the defined binning scheme to maintain $P_{M,t}$ and $P_{M,ref}$. In practice, $P_{M,t}$ is constructed every 5 minutes whereas $P_{M,ref}$ is a historical reference distribution constructed daily from a rolling 14-day window. Maintaining both real-time and historical, reference distributions enables real-time continuous monitoring of model predictions.

2.2 Prediction Distribution Drift Detection

Monitoring reference and real-time distributions is a popular approach to unsupervised drift detection [2]. In order to detect prediction distribution shifts, Prediction Guardian compares distributions $P_t(M)$ and $P_{\text{ref}}(M)$ described above to evaluate drift for a model M. Specifically, at each time t, Prediction Guardian evaluates a statistical robustness score $S(t)$ defined as

$$S(t) = D_{\text{JS}}(P_t(M) \parallel P_{\text{ref}}(M)) \tag{1}$$

and considers an anomaly as

$$S(t) > \delta(M) \tag{2}$$

where $D_{\text{JS}}(\cdot\|\cdot)$ is the Jensen–Shannon divergence and $\delta(M)$ is a predefined threshold for a model M. In practice, $\delta(M)$ is determined through backtesting on ground-truth anomaly events and the statistical robustness score $S(t)$ is calculated in 5-minute intervals.

Prediction Guardian uses the Jensen–Shannon (JS) divergence [15] over other popular divergences such as Kullback–Leibler (KL) divergence to measure statistical robustness. The JS divergence between P and Q is defined as

$$D_{\text{JS}}(P \parallel Q) = \frac{1}{2} D_{\text{KL}}\left(P \parallel \frac{P+Q}{2}\right) + \frac{1}{2} D_{\text{KL}}\left(Q \parallel \frac{P+Q}{2}\right). \tag{3}$$

Leveraging JS divergence provides a few key advantages for Prediction Guardian:

- **Robustness to non-overlapping supports between P and Q:** KL divergence on P and Q requires that the support of P (i.e., set of values where P has non-zero probability) should be a subset of the support of Q and is used widely for detecting drifts in distributions [23, 25]. In the context of prediction distribution shifts where outlier predictions may be present real-time but not when a model is healthy, this key requirement is often violated and points to a limitation of the KL divergence for the Prediction Guardian use-case. On the other hand, this limitation is not present in the JS divergence as it relies on the average distribution, $\frac{P+Q}{2}$, and its support will be the union of the supports of P and Q [15].
- **Boundedness on divergence output values:** The JS divergence has a lower bound of 0 and an upper bound of 1 when using $\log_2(x)$ in the KL divergence computations. As a result, it is easier to interpret the value of the JS divergence and compare results across different scenarios. This boundedness property is key for thresholding and interpreting the statistical robustness score across the many hundreds of models used in complex real-world ranking systems for digital advertising.

2.3 Out-of-Distribution Prediction Smoothing

Importance weighting is a fundamental technique used in machine learning to compare distribution shifts [29]. Given a source and a target distribution, importance weighting techniques can be used to compare and re-weight the source distribution to match the target distribution [9]. To abate prediction issues after successful detection, Prediction Guardian compares $P_{M,t}$ and $P_{M,\text{ref}}$ to re-weight out-of-distribution prediction ranges.

More formally, for a prediction bucket b, define

$$f(b) = P_{M,t}(B = b), \quad g(b) = P_{M,\text{ref}}(B = b), \tag{4}$$

where $f(b)$ and $g(b)$ denote the probability that a prediction from model M belongs to b for $P_{M,t}$ and $P_{M,\text{ref}}$, respectively. Let S denote the set of all prediction buckets, and let b_q denote the prediction bucket at the q-th quantile of the distribution, where $q \in [0, 1]$. The set of prediction buckets greater than the q-th quantile is then given by

$$S_q = \{b \in S \mid b > b_q\}. \tag{5}$$

In order to handle prediction spike incidents during model serving—where predictions shift toward high buckets—q_{high} is chosen to be a number larger than 0.5 so that $S_{q_{\text{high}}}$ contains only high prediction buckets. Prediction Guardian then performs a "soft zero-out" to remove predictions in high buckets to stabilize the distribution. This mechanism probabilistically suppresses predictions in out-of-distribution high buckets, reducing the likelihood of unhealthy predictions winning auctions and preventing advertisers from being overcharged due to inflated scores. Specifically, a smoothing multiplier $\epsilon \approx 10^{-8}$ is randomly applied with probability w_b to each model prediction belonging to prediction bucket $b \in S_{q_{\text{high}}}$. Here w_b is defined as

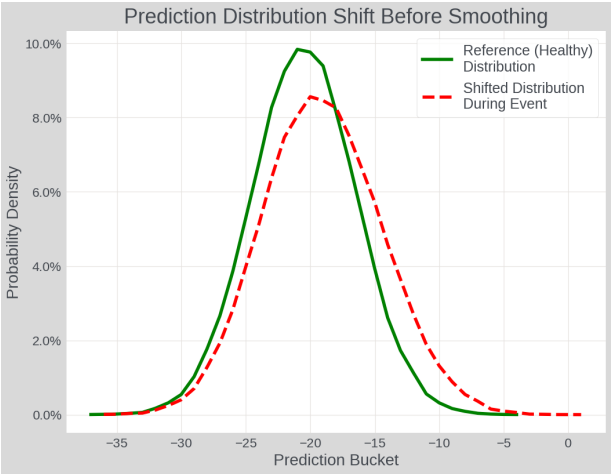
$$w_b = \begin{cases} 1 - \dfrac{g(b)}{f(b)}, & \text{if } f(b) > g(b), \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

The detection algorithm described in Section 2.2 also identifies prediction drop incidents, in which the prediction mass shifts toward low buckets. To avoid exacerbating such cases, no zero-out is applied for any bucket $b \notin S_{q_{\text{high}}}$. Moreover, Prediction Guardian does not apply positive prediction boosts: the system is explicitly designed to mitigate prediction overestimation and associated advertiser overcharge. Mitigating prediction overestimation is prioritized over mitigating prediction underestimation due to the potential loss of advertisers' trust caused by advertiser overcharge. However, mitigating underestimated prediction is a future work to better align predictions with realized outcomes.

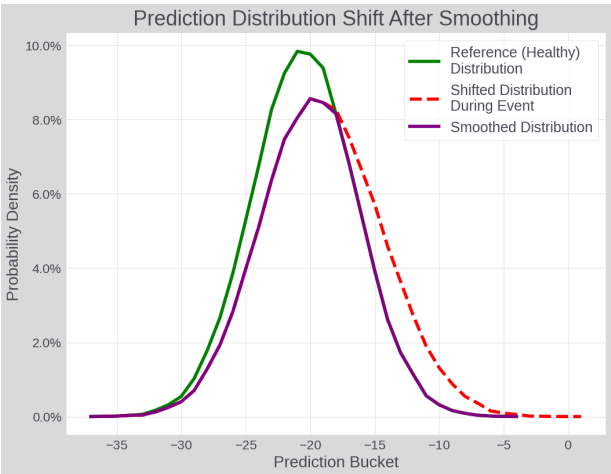
Figure 4 shows a theoretical distribution shift scenario. The green curve represents the reference distribution whereas the red dotted curve represents a shifted distribution due to ranking model prediction issues in real-time. In order to correct this distribution drift, Prediction Guardian smooths out the out-of-distribution prediction buckets resulting in the purple curve. As an example, for prediction bucket -15 where $f(-15) = 5.5\%$ and $g(-15) = 4\%$, the prediction range has a greater frequency of predictions than expected. Consequently, Prediction Guardian will aim to smooth out

$$1 - \frac{4}{5.5} = 27.3\%$$

of predictions belonging to prediction bucket -15. Prediction Guardian will execute a similar procedure bucket-wise across all prediction buckets.



(a) Unhealthy prediction distribution before smoothing



(b) Unhealthy prediction distribution adjusted after smoothing

Figure 4: Prediction Guardian smooths out only out-of-distribution prediction ranges by using the healthy distribution (green) as the reference

By detecting and smoothing out unhealthy predictions, Prediction Guardian improves reliability for digital advertisers through:

- **Increased stability in prediction outputs:** During service disruption incidents driven by abnormally high model predictions, the distribution $P_{M,t}$ is smoothed to match $P_{M,ref}$ and have a more stable prediction output.
- **Reduced negative impact to advertiser experience:** Ads corresponding to unhealthy predictions no longer win in auctions, which prevents advertisers from getting overcharged.

3 System Design

The Prediction Guardian system employs a dual-component architecture comprising an offline detection system and an online mitigation system (so-called Ads Serving Clients), designed to operate at scale across hundreds of ranking models serving live traffic while maintaining sub-minute response times to prediction anomalies.

The detection system aims to identify prediction outliers in near real time, expose model robustness metrics as critical health indicators to downstream consumers such as mitigation systems, and provide real-time intermediate prediction distribution histograms for analysis.

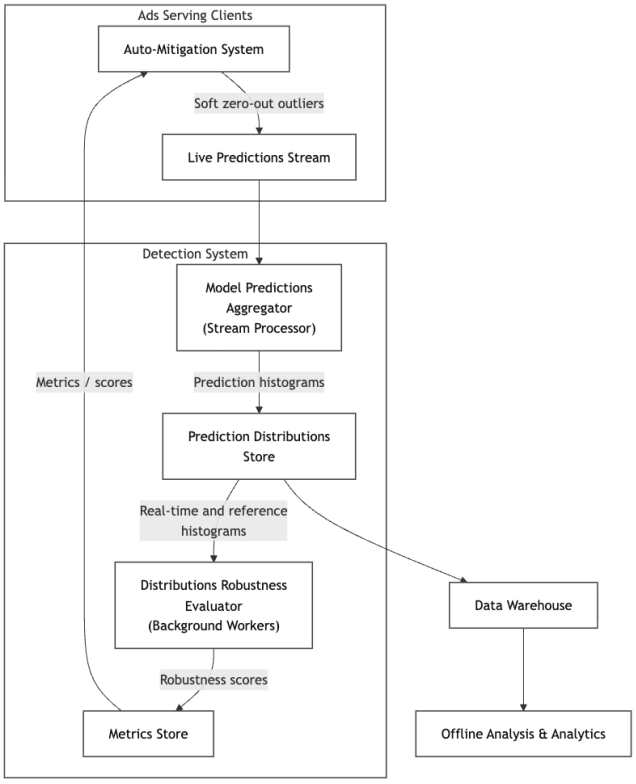


Figure 5: System design for the dual-component architecture of Prediction Guardian

The detection and mitigation system consists of five major components:

Model Predictions Aggregator. This component functions as a stream processor, subscribing to unsampled model prediction events from ads serving clients at 1-minute intervals. It aggregates events across various breakdowns, including model type (CTR or CVR models predicting different user engagement outcomes), model ID, snapshot ID (a specific version of a model as trained on data up to a particular point in time), serving regions, and prediction distribution buckets. The system supports idempotent processing, duplicate event handling, and reprocessing of historical data, all while ensuring privacy compliance for user-identifiable information.

Output format:

```
<Timestamp, ModelType, ModelId, SnapshotId,
Region, PredictionBucket>, AdsCount
```

$PredictionBucket = \lfloor 10 \cdot \log_{10}(\text{prediction}) \rfloor$

Prediction Distributions Store. Online relational database supporting low-latency, near-real-time access for querying prediction distribution histograms, with privacy-compliant storage and rich SQL-like query interface. The database offers easy extensibility for adding additional breakdowns and accommodating reprocessing of historical data. The online database is combined with a data warehouse system for longer retention (3+ years) and offline analysis & analytics purposes.

Distribution Robustness Evaluator. Python-based background worker service which periodically queries histograms from Prediction Distributions Store to build test curve from latest predictions (minutes of data) and control curve from historical predictions (days/weeks of data). Control data is cached to avoid repeated data queries and processing. The JS divergence algorithm is implemented to evaluate robustness metrics per model type and ID to quantify the curve shifts.

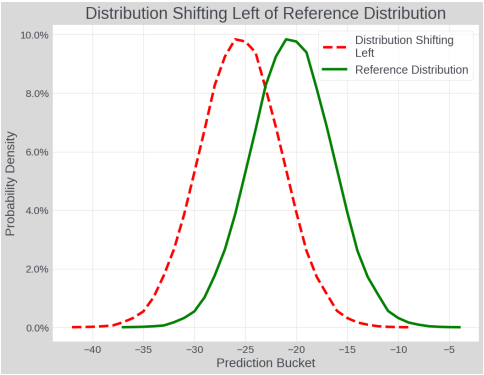
Metrics Store. Non-relational key-value store that saves and distributes evaluated prediction robustness scores to a large number of Ads Serving clients with low latency and per minute update frequency. Writes are also globally replicated to be available to all regions instantly.

Auto-Mitigation System. Receives periodic updates from Metric Store to apply mitigation strategies in real time to predictions generated by models in response to user requests. Mitigation strategies include zero-out/overwriting unhealthy predictions at model type, model ID or advertiser campaign level. Mitigation system ensures limited time-to-live (TTL) to these metrics, ensuring that stale or incorrect metrics are automatically removed, to guard against detection infra issues.

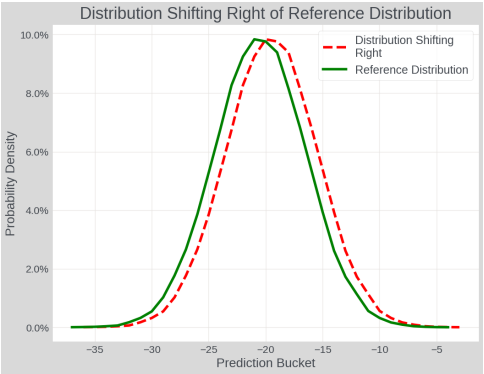
4 Evaluation and Results

Prediction Guardian's capabilities in detection and mitigation were assessed on the three most common scenarios of prediction distribution corruptions as seen in Figure 6:

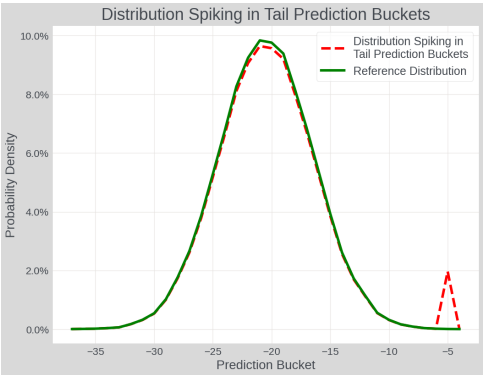
- **Real-time Distribution Shifting Left of Reference Distribution:** When ranking models start underpredicting the probability of an event, the model's prediction distribution shifts to the left of the reference distribution. Consistently underpredicting ads performance leads to undercharging advertisers, which causes negative business impact as a result.
- **Real-time Distribution Shifting Right of Reference Distribution:** In contrast, when ranking models start overpredicting probabilities of an event, the model's prediction distribution shifts to the right of the reference distribution. Overpredicting performance of ads often leads to overcharging advertisers, causing loss of advertiser trust in ad platforms.
- **Real-time Distribution Spiking in Tail Prediction Buckets:** Ranking model predictions become concentrated at tail prediction buckets corresponding to high prediction ranges. This can lead to a small number of ads to expend a large amount of an advertiser's budget and causes an inefficient experience for advertisers.



(a) Prediction distribution shifting to the left relative to reference distribution due to model underprediction



(b) Prediction distribution shifting to the right relative to reference distribution due to model overprediction



(c) Prediction distribution spiking in tail prediction buckets due to extreme model overprediction

Figure 6: Common scenarios of distribution corruptions found during service disruption events of model predictions

For evaluating detection capabilities of Prediction Guardian, the focus was on all three scenarios. However, for evaluating mitigation capabilities for Prediction Guardian, the focus was primarily on scenario 2 and scenario 3 since the intent of the mitigation system is

to reduce overpredictions that lead to advertiser overcharge impact. For both detection and mitigation systems, evaluations were done by conducting experiments on simulated datasets and monitoring performance during service disruption events.

4.1 Evaluation Dataset Construction

Experiments for Prediction Guardian were conducted using datasets that captured distribution corruptions in ranking models. These datasets were created from both simulation data and real-world historical service disruption events.

4.1.1 Simulated Datasets for Distribution Corruptions. From Table 1, live service disruption events occur infrequently and in uncontrolled manners. As a result, it is essential to have a curated time-series dataset based on simulation data of the three classes of distribution corruption.

To construct the simulated datasets, we first extracted reference distributions from a data warehouse containing production logs of 15 ranking models (10 CVR and 5 CTR) spanning 180 days. Reference distributions were built by aggregating prediction outputs from periods verified to be free of known service disruptions or anomalies, ensuring that only "healthy" model behavior was captured. This selection process involved cross-referencing incident logs to exclude outlier periods. The chosen models represent a diverse set of advertiser objectives and traffic segments, providing a robust foundation for simulating realistic prediction corruptions.

To simulate prediction corruptions, we applied targeted anomaly injection functions to the healthy reference distributions. Formally, from a reference collection R of healthy model predictions of model M, a corruption function h_M would be applied on R to obtain a corrupted distribution of model predictions $h_M(R)$.

Examples of corruption functions that were used in experiments included:

- h_M would be defined as randomly selecting 10% of prediction values in R and multiplying them by constant $0 < c < 1$ to shift the distribution left of reference distribution
- h_M would be defined as randomly selecting 10% of prediction values in R and multiplying them by a constant $c > 1$ to shift the distribution right of reference distribution

Table 1: A comparison of event counts based on different scenario types using simulated and historical datasets

Scenario ID	Scenario Type	Dataset Type	Event Count
LS	Left distribution shift	Simulation	50
LH	Left distribution shift	Historical Backtest	25
RS	Right distribution shift	Simulation	50
RH	Right distribution shift	Historical Backtest	55
TS	Tail outlier spike distribution shift	Simulation	50
TH	Tail outlier spike distribution shift	Historical Backtest	20

- h_M would be defined as randomly selecting 5% of prediction values of R and setting them to the 99th percentile prediction output of model M to create spikes in tail prediction buckets

The parameter choices in the aforementioned corruption functions were informed by analysis of past service disruptions, ensuring that the simulated corruptions closely resemble real-world failure modes and stress-test the detection and mitigation capabilities of the system.

Given a model M and corruption function h_M, the following process was used to simulate the disruption events from distribution shifts given a reference prediction dataset:

(1) For each 5-minute time period t, construct a distribution snapshot from all predictions output at time t to form the reference distribution
(2) Construct n time intervals $\{[s_1, e_1), \ldots, [s_n, e_n)\}$ to denote periods when simulated service disruptions would occur. Generally, these time periods would be on the order of hours to reflect the duration of service disruption events.
(3) Apply h_M to prediction values for each time interval $[s_i, e_i)$ and simulate impact to downstream components of ads auctions, specifically logging advertiser charge before and after anomalies would occur.

Following the above process, a time-series dataset comprising of prediction data, service disruption time intervals data, and advertiser charge data is constructed for each type of prediction distribution corruption for various ranking models. Prediction Guardian was evaluated on each instance of distribution corruption in this simulated dataset.

4.1.2 Historical Backtesting Dataset for Service Disruption Issues. The historical backtesting dataset was constructed by aggregating prediction distribution patterns from 80 ranking models, including both click-through rate (CTR) and conversion rate (CVR) models, and also noting time intervals when service disruptions occurred for models. The selected models span a diverse range of advertiser objectives and traffic segments providing a comprehensive evaluation ground for Prediction Guardian. This diversity ensures that the assessment covers various business scenarios and model behaviors encountered in production.

Prediction data was sourced from production logs and internal monitoring systems over a continuous 180-day period, ensuring coverage of a wide range of operational conditions and business cycles. Each model's dataset was systematically audited for unhealthy prediction behavior using a combination of automated anomaly detection algorithms on overcharge or prediction fluctuations as well as manual review. Automated systems flagged intervals with significant deviations in prediction distributions, which were then further examined by domain experts to confirm the presence of genuine anomalies.

Service disruption events were defined as intervals $[s_i, e_i)$ where prediction distributions showed statistically significant deviations from healthy baselines, determined by historical thresholds and corroborated by incident logs and business impact reports (e.g., changes in advertiser spend or auction outcomes). Annotation involved cross-referencing automated alerts with manual review and

logging of business metrics, marking only intervals with clear evidence of model failure. Multiple domain experts validated annotations to minimize false positives and ensure consistency.

Following this process, a time-series dataset comprising of prediction data, service disruption time intervals data, and advertiser charge data is constructed for each ranking model based on past live service disruption events. While the historical backtesting dataset provides robust coverage of real-world service disruption events, some limitations exist. For instance, potential gaps in incident reporting and manual annotation may lead to missing service disruption events.

4.2 Prediction Guardian Detection Evaluation

Prediction Guardian is designed to detect anomalies in DLRMs for online advertising. This section presents a comprehensive evaluation of the detection system, detailing the methodology, metrics, and results across a broad spectrum of anomaly types and operational scenarios.

4.2.1 Evaluation Method of Detection Framework. In order to evaluate performance of the detection framework, the key metrics used were:

- **Precision**: Precision is defined as the number of 5-minute intervals where Prediction Guardian metrics detect a true issue compared to the total number of 5-minute intervals where Prediction Guardian metrics breached the defined threshold.
- **Recall**: Recall is defined as the number of 5-minute intervals where Prediction Guardian metrics detect a true issue compared to the total number of 5-minute intervals where a true issue existed.
- **Time-to-detection (TTD)**: TTD is defined as the average time (in minutes) it takes for the system to detect a model prediction issue after the issue has started.

Prediction Guardian's detection capabilities were evaluated for precision, recall, and TTD by applying the technique to both simulated and historical distribution datasets covering all three common distribution corruption types. For each 5-minute interval period, the detection framework was evaluated on whether it could signal an issue had occurred during the provided service disruption time intervals in the various datasets.

4.2.2 Prediction Guardian Detection Results. Table 2 presents the summary of Prediction Guardian detection results.

For left distribution shifts, the precision and recall figures are similar across both simulated and historical event datasets with precision ranging between 0.803 and 0.823 and recall ranging between 0.859 and 0.909.

For right distribution shifts, the recall ranges between 0.86 and 0.908 and is similar across both datasets, but precision figures are dissimilar. Simulation data indicates a higher precision of 0.914 while datasets based on historical service disruption events have a lower precision of 0.750. This discrepancy in precision comes from instances of spurious low-volume traffic periods in historical data which were not present in simulation data. Specifically, lower volume traffic leads to more noisy distributions, which increases false positives in our detection techniques. Future work will target making Prediction Guardian robust to these sparse data scenarios.

Table 2: A comparison of evaluation metrics across different simulations and service disruption events

Scenario ID	Precision	Recall	Median TTD (minutes)
LS	0.823	0.859	21*
LH	0.803	0.909	20
RS	0.914	0.908	18*
RH	0.750	0.860	25
TS	1.00	0.950	11*
TH	1.00	1.00	10

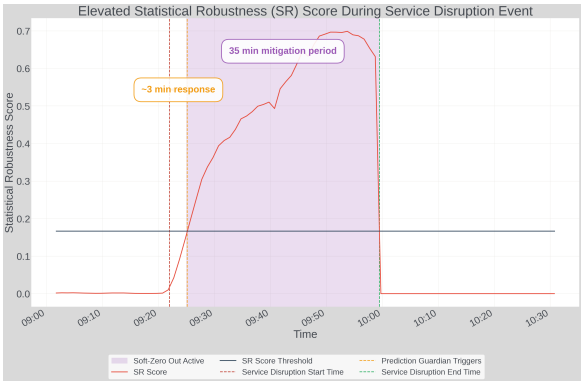
*TTD values for simulation datasets are normalized to account for production latency, enabling fairer comparison with historical event datasets.

For distribution shifts with tail outliers, the precision and recall figures are also similar across both simulated and historical event datasets with precision of 1.00 and recall ranging between 0.95 and 1.00. These figures suggest that distribution-based detection methods are able to capture extreme outliers very effectively. One such example is noted in Section 4.2.3 in greater detail.

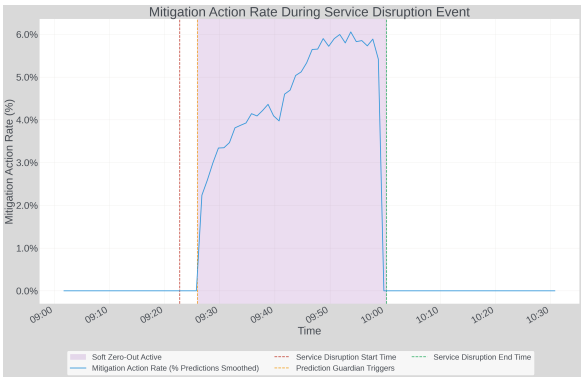
Furthermore, achieving a Time-To-Detection (TTD) of 15-25 minutes demonstrates that the framework is capable of identifying anomalies or shifts in model prediction distributions quickly. This rapid response is particularly advantageous in production environments where early detection can prevent downstream errors and mitigate potential business impact. In contrast, alternative approaches such as Prediction Calibration typically exhibit longer detection times on the order of hours due to delayed conversion [28], which may delay corrective actions and increase the risk of undetected model drift.

4.2.3 A Case Study on A Successful Detection Scenario. Figure 7 illustrates the statistical robustness (SR) score of a single service disruption event between 9:22 AM and 10:00 AM attributed to feature drift causing unhealthy model tail prediction outliers. At 9:25 AM, the robustness score exceeded the predefined threshold, prompting the Prediction Guardian guardrail to identify the model anomaly with a time-to-detection (TTD) of 3 minutes.

The guardrail remained effective from 9:25 AM to 10:00 AM, applying a soft-zeroing mechanism to 4% of the model's predictions on average in order to mitigate effects of overprediction as seen in Figure 7b. As the model distribution drift worsens from 9:40 AM to 10:00 AM, the SR score for the model elevates further and more predictions are smoothed by Prediction Guardian. Following the diagnosis implementation of a root-cause fix for the underlying feature issue, the statistical robustness score dropped below the threshold at 10:00 AM, indicating model prediction recovery. Consequently, the Prediction Guardian guardrail ceased its intervention, as the model returned to a stable operational state.



(a) Statistical robustness score remains elevated above threshold from 9:25 AM until 10:00 AM



(b) Prediction Guardian smooths out-of-distribution model predictions during the 35-minute mitigation period

Figure 7: Statistical robustness score of a model elevates during a service disruption event, prompting a 35-minute mitigation period before model recovery

4.3 Prediction Guardian Mitigation Evaluation

Advertiser overcharge during service disruptions can lead to significant financial impact and loss of trust. Prediction Guardian mitigation framework aims to reduce these negative advertiser impacts by adjusting high prediction values. This section presents a comprehensive evaluation of the mitigation system, detailing the methodology, metrics, and results across a broad spectrum of anomaly types and operational scenarios.

4.3.1 Evaluation Method of Mitigation Framework. Prediction Guardian's mitigation capability was evaluated on how effectively it could reduce advertiser overcharge by adjusting over-predicted values during a service disruption event. For this evaluation, we defined a key performance metric, **Advertiser Overcharge Reduction (%)**. Advertiser overcharge (AO) is defined as the excess revenue observed during an incident relative to a stable week-over-week baseline.

$$\text{AO} = R_{\text{incident}} - R_{\text{baseline}}$$

Table 3: Advertiser overcharge reduction rates for Prediction Guardian mitigation, comparing simulated and historical service disruption events across different distribution shift scenarios

Scenario ID	Median Advertiser Overcharge Reduction (%)
RS	59.4%
RH	56.3%
TS	92.7%
TH	91.0%

where R_{incident} and R_{baseline} are the respective revenue from incident and stable periods. AO is typically positive during prediction spike incidents, which are the main focus of our mitigation framework. The advertiser overcharge reduction metric is defined as the change in advertiser overcharge after the Prediction Guardian mitigation triggers over the entire time range of the service disruption:

$$\text{AO Reduction } (\%) = \frac{\text{AO}_{\text{before}} - \text{AO}_{\text{after}}}{\text{AO}_{\text{before}}} \times 100$$

For easier baseline comparison across different tested events and simulations, this metric is represented as a percent change.

Given that the scope of the mitigation system is for targeting over-predicted values, we consider the following scenarios: distribution shifting to right of reference and spikes in tail prediction buckets. For each time range where the distribution corruptions were introduced in the datasets, the advertiser overcharge reduction was recorded before and after the mitigation system took effect.

4.3.2 Prediction Guardian Mitigation Results. Table 3 presents the summary of Prediction Guardian mitigation results.

For right distribution shifts, advertiser overcharge reduction is consistent across both simulated and historical service disruption event datasets, ranging from 56.3% to 59.4%. In contrast, tail outlier spike distributions yield higher reduction rates of 91.0% to 92.7%, which are also consistent across both datasets.

These findings indicate that Prediction Guardian is more effective at mitigating advertiser overcharge when prediction outliers are concentrated in specific prediction buckets, as opposed to general rightward distribution shifts. Oftentimes, such rightward distribution shifts are caused by localized anomalies (e.g., mis-prediction on a group of advertisers, unhealthy predictions from specific advertiser clusters, etc.) that cause the global model's distribution to shift to the right. Enforcing robustness on localized anomalies by training on localized distributions would improve effectiveness of Prediction Guardian.

5 Conclusions and Future Works

The advent of foundational models in online advertising has significantly enhanced the predictive capabilities of recommendation systems, but has also introduced new vulnerabilities, particularly in the form of distribution shifts that can undermine the reliability of ad delivery for advertisers [27, 30]. Prediction Guardian addresses these challenges by providing a framework for real-time detection and mitigation of prediction anomalies in deep learning recommendation models (DLRMs). The detection system enables

rapid identification and diagnosis of model prediction issues, while the mitigation system effectively reduces the impact of erroneous predictions on advertiser outcomes.

While the primary focus of this work is online advertising, the principles and techniques developed for Prediction Guardian are broadly applicable to other large-scale recommendation and ranking systems that rely on real-time model predictions. Our evaluation demonstrates that Prediction Guardian achieves high precision and recall in both detecting and mitigating major prediction anomalies, with rapid time-to-detection and effective reduction of adverse impacts. However, challenges remain in handling localized anomalies, low-sample scenarios, and mitigation for under-prediction cases, motivating the following future works:

- **Addressing localized anomalies in model prediction distributions**: Distribution shifts may occur globally, affecting all outputs of a model, or locally, impacting only specific segments such as individual hosts or hardware clusters [4]. Effective adaptation to sudden model prediction drifts requires understanding and responding to the locality of these shifts [11]. The current framework monitors distribution patterns at a global level, which limits its ability to precisely target localized anomalies. Future development will focus on extending the system to support fine-grained detection and mitigation across dimensions such as cluster regions and hardware types.
- **Expanding mitigation to handle under-prediction**: The present mitigation strategy primarily addresses over-prediction (e.g., distribution right-shift, outlier spikes, etc.) by reducing high prediction values through a soft zero-out approach. However, under-prediction (e.g., distribution left-shift) degrades model performance and business outcomes. Future work will generalize re-weighting techniques to boost low-prediction ranges based on distribution data, with careful attention to maintaining the fidelity of boosting multipliers and preventing unintended advertiser overcharge.
- **Extending detection to support sparse data**: Currently, Prediction Guardian acts as a post-hoc guardrail, detecting and mitigating issues after model deployment. Applying the framework earlier—during experimentation or rollout—could proactively prevent prediction problems, but poses challenges in low-sample scenarios where noise may mimic anomalies. Overcoming this will require modifying algorithms and introducing adaptive thresholding to handle both distributional properties and limited data, ensuring reliable anomaly detection in data-sparse settings.

References

[1] Newsha Ardalani, Carole-Jean Wu, Zeliang Chen, Bhargav Bhushanam, and Adnan Aziz. 2022. Understanding scaling laws for recommendation models. *arXiv preprint arXiv:2208.08489* (2022).

[2] Cloudera Fast Forward Labs. 2021. *Inferring Concept Drift Without Labeled Data*. https://concept-drift.fastforwardlabs.com/

[3] Yewen Fan, Nian Si, and Kun Zhang. 2022. Calibration matters: Tackling maximization bias in large-scale advertising recommendation systems. *arXiv preprint arXiv:2205.09809* (2022).

[4] Joao Gama and Gladys Castillo. 2006. Learning with local drift detection. In *International conference on advanced data mining and applications*. Springer, 42–55.

[5] João Gama, Indrė Žliobaitė, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. 2014. A survey on concept drift adaptation. *ACM computing surveys (CSUR)* 46, 4 (2014), 1–37.

[6] Shijie Geng, Juntao Tan, Shuchang Liu, Zuohui Fu, and Yongfeng Zhang. 2023. Vip5: Towards multimodal foundation models for recommendation. *arXiv preprint arXiv:2305.14302* (2023).

[7] Xinran He, Junfeng Pan, Ou Jin, Tianbing Xu, Bo Liu, Tao Xu, Yanxin Shi, Antoine Atallah, Ralf Herbrich, Stuart Bowers, et al. 2014. Practical lessons from predicting clicks on ads at facebook. In *Proceedings of the eighth international workshop on data mining for online advertising*. 1–9.

[8] Brad Jackson, Jeffrey D Scargle, David Barnes, Sundararajan Arabhi, Alina Alt, Peter Gioumousis, Elyus Gwin, Paungkaew Sangtrakulcharoen, Linda Tan, and Tun Tao Tsai. 2005. An algorithm for optimal partitioning of data on an interval. *IEEE Signal Processing Letters* 12, 2 (2005), 105–108.

[9] Masanari Kimura and Hideitsu Hino. 2024. A short survey on importance weighting for machine learning. *arXiv preprint arXiv:2403.10175* (2024).

[10] Kevin H Knuth. 2006. Optimal data-based binning for histograms. *arXiv preprint physics/0605197* (2006).

[11] Łukasz Korycki and Bartosz Krawczyk. 2023. Adversarial concept drift detection under poisoning attacks for robust data stream mining. *Machine Learning* 112, 10 (2023), 4013–4048.

[12] Sébastien Lahaie and David M Pennock. 2007. Revenue analysis of a family of ranking rules for keyword auctions. In *Proceedings of the 8th ACM Conference on Electronic Commerce*. 50–56.

[13] Hayeon Lee, Rui Hou, Jongpil Kim, Davis Liang, Sung Ju Hwang, and Alexander Min. 2023. A study on knowledge distillation from weak teacher for scaling up pre-trained language models. *arXiv preprint arXiv:2305.18239* (2023).

[14] Jian Liang, Ran He, and Tieniu Tan. 2025. A comprehensive survey on test-time adaptation under distribution shifts. *International Journal of Computer Vision* 133, 1 (2025), 31–64.

[15] Jianhua Lin. 2002. Divergence measures based on the Shannon entropy. *IEEE Transactions on Information theory* 37, 1 (2002), 145–151.

[16] Liang Luo, Yuxin Chen, Zhengyu Zhang, Mengyue Hang, Andrew Gu, Buyun Zhang, Boyang Liu, Chen Chen, Chengze Fan, Dong Liang, et al. 2025. Meta Lattice: Model Space Redesign for Cost-Effective Industry-Scale Ads Recommendations. *arXiv preprint arXiv:2512.09200* (2025).

[17] Ning Ma, Mustafa Ispir, Yuan Li, Yongpeng Yang, Zhe Chen, Derek Zhiyuan Cheng, Lan Nie, and Kishor Barman. 2022. An online multi-task learning framework for google feed ads auction models. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 3477–3485.

[18] Kelong Mao, Jieming Zhu, Liangcai Su, Guohao Cai, Yuru Li, and Zhenhua Dong. 2023. FinalMLP: an enhanced two-stream MLP model for CTR prediction. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 37. 4552–4560.

[19] Kiran Kumar Matam, Hani Ramezani, Fan Wang, Zeliang Chen, Yue Dong, Maomao Ding, Zhiwei Zhao, Zhengyu Zhang, Ellie Wen, and Assaf Eisenman. 2024. {QuickUpdate}: a {Real-Time} personalization system for {Large-Scale} recommendation models. In *21st USENIX Symposium on Networked Systems Design and Implementation (NSDI 24)*. 731–744.

[20] H Brendan McMahan, Gary Holt, David Sculley, Michael Young, Dietmar Ebner, Julian Grady, Lan Nie, Todd Phillips, Eugene Davydov, Daniel Golovin, et al. 2013. Ad click prediction: a view from the trenches. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. 1222–1230.

[21] Meta Engineering. 2024. *Meta's Approach to Machine Learning Prediction Robustness*. https://engineering.fb.com/2024/07/10/data-infrastructure/machine-learning-ml-prediction-robustness-meta/ Accessed: 2025-01.

[22] Maxim Naumov, Dheevatsa Mudigere, Hao-Jun Michael Shi, Jianyu Huang, Narayanan Sundaraman, Jongsoo Park, Xiaodong Wang, Udit Gupta, Carole-Jean Wu, Alisson G Azzolini, et al. 2019. Deep learning recommendation model for personalization and recommendation systems. *arXiv preprint arXiv:1906.00091* (2019).

[23] Raquel Sebastião and João Gama. 2007. Change detection in learning histograms from data streams. In *Portuguese Conference on Artificial Intelligence*. Springer, 112–123.

[24] TensorFlow. 2024. The Evaluator TFX Pipeline Component. https://www.tensorflow.org/tfx/guide/evaluator. Accessed: 2025-12-23.

[25] Tamraparni Dasu Shankar Krishnan Suresh Venkatasubramanian and Ke Yi. 2006. An Information-Theoretic Approach to Detecting Changes in Multi-Dimensional Data Streams. *Proc. Symp. on the Interface of Statistics, Computing Science, and Applications* 12, 2 (2006).

[26] Ramanathan Vishnampet, Rajesh Shenoy, Jianhui Chen, and Anuj Gupta. 2024. Root Causing Prediction Anomalies Using Explainable AI. *arXiv preprint arXiv:2403.02439* (2024).

[27] Alexander Wan, Eric Wallace, Sheng Shen, and Dan Klein. 2023. Poisoning language models during instruction tuning. In *International Conference on Machine Learning*. PMLR, 35413–35425.

[28] Yifan Wang, Peijie Sun, Min Zhang, Qinglin Jia, Jingjie Li, and Shaoping Ma. 2023. Unbiased delayed feedback label correction for conversion rate prediction.

In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2456–2466.

[29] Makoto Yamada, Taiji Suzuki, Takafumi Kanamori, Hirotaka Hachiya, and Masashi Sugiyama. 2013. Relative density-ratio estimation for robust distribution comparison. *Neural computation* 25, 5 (2013), 1324–1370.

[30] Wenhan Yang, Jingdong Gao, and Baharan Mirzasoleiman. 2023. Better safe than sorry: Pre-training clip against targeted data poisoning and backdoor attacks. *arXiv preprint arXiv:2310.05862* (2023).

[31] Buyun Zhang, Liang Luo, Yuxin Chen, Jade Nie, Xi Liu, Daifeng Guo, Yanli Zhao, Shen Li, Yuchen Hao, Yantao Yao, et al. 2024. Wukong: Towards a scaling law for large-scale recommendation. *arXiv preprint arXiv:2403.02545* (2024).

[32] Qianqian Zhang, Xinru Liao, Quan Liu, Jian Xu, and Bo Zheng. 2022. Leaving no one behind: A multi-scenario multi-task meta learning approach for advertiser modeling. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*. 1368–1376.