

When Offline Performance Misleads: A Leakage Audit of Completed-Session Features for Cart-Abandonment Prediction

Babak Rad Jahanbani^{1*}, Ahmet Sayar²

MSc Candidate, Information Technology, Bahçeşehir University, Istanbul, Türkiye¹

Department of Artificial Intelligence Engineering, Bahçeşehir University, Istanbul, Türkiye²

Abstract—Cart-abandonment models are often evaluated on session summaries created only after sessions end. Such tables can violate the information boundary of an intervention-time prediction task and inflate offline performance. This study audits 43,917 cart sessions reconstructed from the public Retail Rocket clickstream dataset. Feature-lineage and arithmetic-closure tests reveal two leakage mechanisms. First, transaction count is exactly recoverable from total, view, and add-to-cart counts. Second, final session age, mean inter-event interval, and recency from the last cart action are computed at a label-dependent completed-session endpoint. A third defect lies in the target: the session-level purchase label conflates the purchase of one item with the abandonment of another, and an item-level check finds this in 1,354 sessions (3.1%). Across five repeated stratified holdouts, XGBoost obtained a receiver operating characteristic area under the curve (ROC-AUC) of 0.9997 with the deterministic count proxy and 0.9947 with the original completed-session features. After the three endpoint-dependent timing variables were removed, ROC-AUC fell to 0.6065. Visitor-grouped and chronological splits reproduce the same pattern. A point-in-time evaluation at the first add-to-cart event, using only information observable at that moment, yields ROC-AUC between 0.57 and 0.65 across four models. These results do not establish a deployable predictor. They show that the near-perfect score described completed sessions rather than the intervention-time task, and they provide a rerunnable audit procedure, summarized as a nine-question checklist, for detecting such problems before model complexity or performance claims are interpreted.

Keywords—Cart abandonment; clickstream analytics; data leakage; temporal leakage; offline evaluation; point-in-time evaluation; feature validity; e-commerce

I. INTRODUCTION

A cart-abandonment prediction is operationally useful only when it is produced early enough to support an intervention [1], [2]. This requirement creates a strict information boundary: every model input must be available at the stated decision moment. Historical clickstream data, however, are often transformed into one row per completed session [3], [4], [5]. Counts, durations, recency variables, and the outcome label are then stored together and passed to a classifier. This representation is convenient. It can also silently convert a prospective prediction task into a retrospective description of a session whose outcome is already known, which is the defining form of data leakage [6].

The present study originated in a bounded replay prototype. The prototype passed pre-sessionized clickstream events

through Kafka, periodically aggregated them with Spark, stored evolving session documents in MongoDB, and applied tabular and hybrid Long Short-Term Memory (LSTM) models. It provides project context rather than evidence of point-in-time validity: the producer supplied a precomputed session key, and Spark used repeated bounded reads rather than checkpointed incremental processing. During offline validation, the scores were implausibly strong for a noisy behavioral outcome, a pattern that leakage commonly produces [7]. Logistic regression reached a receiver operating characteristic area under the curve (ROC-AUC) of 0.9969 over all reconstructed sessions, and an XGBoost benchmark restricted to cart sessions reached 0.9946. Rather than treating these values as evidence of model quality, the construction of every feature was traced back to the underlying event log. This inspection exposed both an exact arithmetic encoding of the target and a subtler dependence on the completed-session endpoint.

The study therefore addresses the following research question:

RQ: How can completed-session feature construction create misleading performance estimates in cart-abandonment prediction?

Five bounded contributions are made. First, an arithmetic target proxy is demonstrated: transaction count remains exactly reconstructable after the explicit transaction feature is removed. Second, endpoint-dependent timing variables are shown to incorporate either the label-generating transaction event or the complete negative history. Third, the session-level purchase label is shown to be ambiguous at a within-session decision point; the defect is verified at item level, and a cutoff-indexed alternative is defined. Fourth, a controlled feature ablation is reported under stratified, visitor-grouped, and chronological splits. Fifth, a point-in-time evaluation at the first add-to-cart event and a reusable audit procedure separate feature availability from mere historical observability.

The scope is deliberately narrow. The ablation is diagnostic: its reduced configuration still summarizes completed sessions and is not an online-valid predictor. The point-in-time evaluation estimates what is achievable with information available at a decision, but it remains an offline estimate. Neither evaluation measures a causal offline-to-online performance loss.

The remainder of this paper is organized as follows. Section II positions the study relative to purchase-intention prediction and leakage research. Section III defines the data, decision

*Corresponding author.

boundary, audit procedure, and experimental protocol. Section IV documents the detected leakage mechanisms. Section V reports the controlled results. Sections VI and VII discuss implications, limitations, and reproducibility, and Section VIII concludes the paper.

II. RELATED WORK

Two bodies of research are directly relevant: models that predict purchase or abandonment from clickstream behavior, and methodological work on leakage and evaluation integrity. Their intersection defines the gap addressed by this audit.

A. Purchase Intention and Cart Abandonment

Purchase-intention prediction has been studied using tabular, sequential, and deep-learning methods. Sakar et al. [3] addressed real-time intention prediction on online shopper sessions, comparing multilayer perceptrons with recurrent models. Working from a deliberately restricted observation window, Requena et al. [2] contrasted handcrafted trajectory features with deep models. Cart abandonment specifically was targeted by Rausch et al. [4] using clickstream-derived variables and supervised learning. More recent work has examined static and sequential information under class imbalance [5], and has treated leakage prevention as an explicit preparation step for post-add-to-cart data [8]. These studies share one feature: each reports performance on a session representation assembled after the session closed. None tests whether the variables in that representation were available at the decision the model is said to support. That question is prior to model selection, and it is the question audited here.

Other studies make the decision-time boundary more explicit. Esmeli et al. [1] generate features as interactions arrive and evaluate how early purchase intention can be recognized. Their evaluation, however, reports aggregate discrimination over a single session representation selected by the model's own prediction outcome. It does not report discrimination as a function of decision position, so comparable position-indexed figures are not available. More recently, Esmeli and Gokce studied the same post-add-to-cart population on two proprietary datasets. Using session-level price dynamics, category transitions, and geographic context, they reported ROC-AUC values of 0.990 and 0.969 [9]. Their session-aggregate features are defined over the complete browsing sequence, and no within-session prediction cutoff is stated. Those figures are therefore also aggregate rather than indexed by decision position. Fabra et al. similarly update behavioral profiles as new log events are observed [10]. Unlike those studies, the present work does not propose another early-prediction architecture. It asks a prior validity question: whether a conventional completed-session table represents the information that an early predictor could actually observe.

B. Leakage and Evaluation Integrity

Kaufman et al. define leakage as the use of information that would not legitimately be available in the prediction context and recommend separating learning-time data from prediction-time information [6]. Subsequent evidence indicates that leakage is a widespread source of irreproducible and overoptimistic findings across application domains [7]. The

TABLE I. CART-SESSION DATASET USED IN THE AUDIT

Property	Value
Cart sessions	43,917
Unique visitors	37,715
Abandoned sessions	31,985 (72.83%)
Purchased sessions	11,932 (27.17%)
Session boundary	30 min inactivity
Event types	view, add-to-cart, transaction

REFORMS recommendations therefore emphasize transparent reporting of data construction, splitting, preprocessing, and evaluation decisions [11]. In recommender systems, Ji et al. showed that offline evaluations that ignore the global timeline leak future interactions into training [12]. Concurrently with this study, Alzghoul et al. audited session-based navigation prediction on other public clickstream data and found that descriptors defined only after a session ends, such as completed session length, distort offline benchmarks; they also adopt a prefix-only protocol [13].

Evaluation integrity is particularly important when complex models appear to dominate simple baselines. Ferrari Dacrema et al. showed that several reported neural improvements in recommender systems disappeared under stronger reproduction and baseline practice [14]. Sculley et al. describe the related systems risk: undeclared data dependencies can turn local modeling decisions into hidden technical debt [15]. General leakage guidance is well established, but session-level arithmetic closure and label-dependent endpoints are rarely demonstrated together in a cart-abandonment case. The present study contributes that concrete, auditable link between feature construction and apparent model performance.

III. DATA AND AUDIT DESIGN

The audit was designed around one principle: feature validity is determined by information availability at the intended decision time, not by whether data were processed offline or through streaming infrastructure. This section defines the artifact, formal boundary, audit steps, and comparison protocol.

A. Dataset and Session Construction

The source is the public Retail Rocket dataset, which contains timestamped `view`, `addtocart`, and `transaction` events from an e-commerce site [16]. The preparation artifacts report 2,756,101 raw events collected over 137 days and 2,755,610 events after a heuristic traffic filter removed 491 events associated with 35 visitors. Sessions were then reconstructed offline using a 30-minute inactivity boundary. Sessions without an add-to-cart event were excluded from the cart-abandonment population.

The resulting modeling table contains 43,917 cart sessions from 37,715 visitors. A session is labeled purchased when at least one transaction event occurs within it. The class distribution is 31,985 abandoned sessions (72.83%) and 11,932 purchased sessions (27.17%). Table I summarizes the audited artifact.

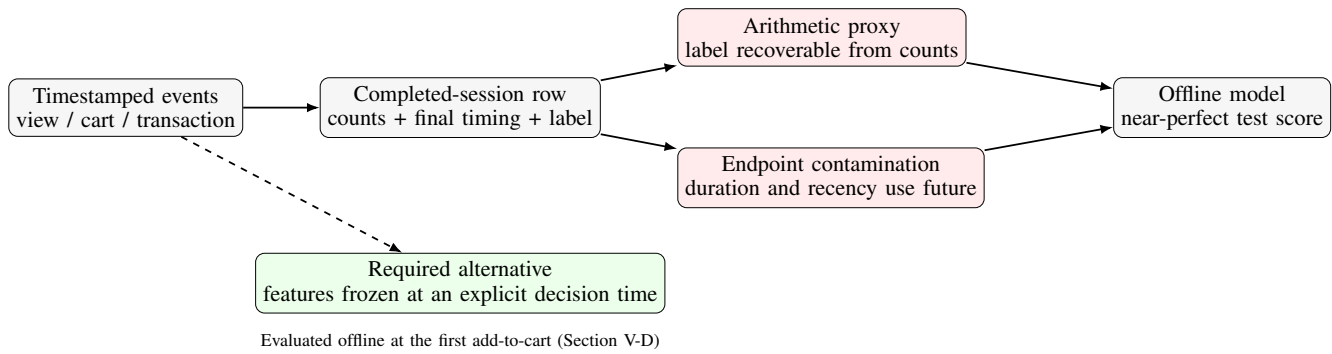


Fig. 1. How completed-session construction changes the prediction problem. The upper path is audited here. The lower path is the point-in-time design; Section V-D evaluates it offline at the first add-to-cart event.

B. Prediction-Time Information Boundary

Let session s consist of ordered events $e_{s,1}, \dots, e_{s,n}$. If scoring occurs after event $k < n$, a valid feature vector must depend only on the observed prefix:

$$\mathbf{x}_{s,k} = f(e_{s,1}, \dots, e_{s,k}). \quad (1)$$

A feature that violates Eq. 1 by depending on $e_{s,k+1}, \dots, e_{s,n}$ is invalid for that decision even when it is available in a historical file. Completed-session aggregation instead constructs $f(e_{s,1}, \dots, e_{s,n})$ after the endpoint has been observed. The audit therefore asks not whether a column exists, but whether its value could have been generated at the stated prediction time. Fig. 1 contrasts the audited completed-session path with the point-in-time construction required for a future online evaluation.

This distinction prevents a common category error. Offline evaluation is not automatically invalid: historical events can be truncated to emulate a real decision boundary. Conversely, streaming infrastructure does not guarantee validity if each emitted record already contains future-dependent aggregates.

C. Feature-Lineage Audit Procedure

The preprocessing notebook, original model notebook, corrected training scripts, and derived session table were inspected. Each feature was traced to the event types and timestamps from which it was constructed. The audit proceeded in five steps. First, the label event and prediction boundary were defined. Second, direct and algebraically reconstructable target fields were identified. Third, durations and recencies were checked for use of the final session event. Fourth, preprocessing parameters were checked for training-only estimation. Finally, models were retrained under controlled feature ablations.

This order is important. Removing the explicit transaction-count column is insufficient if the same quantity can be reconstructed from other columns. Likewise, removing transaction count does not repair a duration whose endpoint is the transaction timestamp.

Table II records the resulting feature lineage. For every variable it gives the construction formula, the events it reads, and the earliest moment at which its value exists. Completed-session variables exist only after the session has closed, that is,

30 minutes after its last event. Point-in-time variables exist at the cutoff $t_{s,k}$ defined in Section IV-C and are computed from events at or before that cutoff, a prefix that never contains a transaction.

D. Feature Configurations and Evaluation Protocol

Four feature configurations were constructed from the supplied cart-session table. The *target-proxy* condition contains total event, view, and add-to-cart counts, deliberately retaining the deterministic identity as a positive control. The *original completed-session* condition contains the seven features used in the cart-session benchmark: view count, add-to-cart count, view-to-cart ratio, unique items, final session age, mean inter-event interval, and final cart recency. The *timing-only* condition contains the three endpoint-dependent temporal variables. The *counts-only* condition contains view count, add-to-cart count, view-to-cart ratio, and unique items.

The counts-only condition is not claimed to be online valid because the counts still summarize completed histories. It is a conservative diagnostic ablation that removes the two specifically identified channels: the exact count identity and completed-endpoint timing. Table III records the validity judgment for each feature group.

Logistic regression and XGBoost [17] carry the primary evidence. Both were evaluated on every configuration across five stratified 80/20 holdouts with seeds 11, 22, 33, 42, and 55. The same configurations were then repeated under two splits that respect repeat visitors and time. The visitor-grouped split places all sessions of a visitor on one side of an 80/20 partition, drawn once per seed. The chronological split trains on the earliest 80% of sessions by start time and tests on the latest 20%; this partition is fixed. A decision tree and random forest [18] were evaluated with seed 42 only, to reproduce the four-model comparison of the original artifact. All four models were evaluated under identical conditions in the point-in-time evaluation of Section V-D. The hybrid LSTM of the original project appears only as development context (Table IV) and supports no conclusion in this paper.

Class weighting was used for logistic regression and the tree models; XGBoost used the negative-to-positive class ratio as `scale_pos_weight`. Continuous features were capped at their 99th percentile using thresholds estimated from the

TABLE II. FEATURE LINEAGE: CONSTRUCTION, SOURCE EVENTS, AND EARLIEST AVAILABILITY

Variable	Construction	Source events	Earliest availability
<i>Completed-session table (one row per session s with events $e_{s,1}, \dots, e_{s,n}$ at times $t_{s,1}, \dots, t_{s,n}$)</i>			
event_count	n	all	session end
view_count, addtocart_count	number of view / add-to-cart events	view, add-to-cart	session end
transaction_count	number of transaction events; never used as a feature	transaction	session end
view_to_cart_ratio	view_count / addtocart_count	view, add-to-cart	session end
unique_items_count	number of distinct item identifiers	all	session end
session_age_seconds	$t_{s,n} - t_{s,1}$	all, incl. transaction	session end
inter_event_interval_mean	mean gap between consecutive events, $(t_{s,n} - t_{s,1}) / (n - 1)$	all, incl. transaction	session end
recency_last_cart_seconds	$t_{s,n}$ minus time of the last add-to-cart	all, incl. transaction	session end
Label y_s	$\mathbb{I}[\text{transaction_count}_s > 0]$ (Eq. 3)	transaction	session end
<i>Point-in-time prefix at cutoff $t_{s,k}$ (events with timestamp $\leq t_{s,k}$ only)</i>			
View, add-to-cart, and distinct-item counts;	as above, over the prefix	view, add-to-cart	$t_{s,k}$
view-to-cart ratio			
Seconds since first event, first add-to-cart, last	$t_{s,k}$ minus the respective timestamp	view, add-to-cart	$t_{s,k}$
add-to-cart, previous event			
Mean inter-event interval	mean gap within the prefix	view, add-to-cart	$t_{s,k}$
Events after first add-to-cart	number of prefix events after the first add-to-cart ($= k$)	view, add-to-cart	$t_{s,k}$
Repeated views of added items	prefix views of items already added to the cart	view, add-to-cart	$t_{s,k}$
Hour of day, day of week	of $t_{s,k}$ (UTC)	—	$t_{s,k}$
Label $y_{s,k}$	any transaction after $t_{s,k}$ in the same session (Eq. 4)	transaction	session end

TABLE III. FEATURE VALIDITY RELATIVE TO A PRE-OUTCOME SCORING BOUNDARY

Feature/group	Audit result	Reason
Transaction count	Invalid	Directly defines the label.
Total + view + cart counts	Invalid jointly	Transaction count is recovered exactly by subtraction.
Final session age	Invalid	Uses the completed endpoint and includes transaction timing for positive sessions.
Mean interval over full session	Invalid	Includes intervals to later events and potentially the transaction.
Recency to final event	Invalid	The reference endpoint is unknown during an active session.
Completed view/cart/item counts	Hindsight-dependent	No exact label identity, but actions after a possible decision time are included.
Training-only scaling/clipping	Valid preprocessing	Parameters are estimated without test-partition information.

training partition. Logistic regression was standardized through a training-only pipeline implemented with scikit-learn [19].

ROC-AUC was used to assess ranking performance. PR-AUC was also reported because positive-class prevalence was 27.17%. Accuracy, precision, recall, and F1 were reported for the seed-42 reproduction at a threshold of 0.5. Standard deviations across holdouts are descriptive stability measures, not confidence intervals from independent samples. Under the fixed chronological split both models are deterministic, so a single value is reported.

IV. LEAKAGE MECHANISMS

The evidence emerged in stages. The initial all-session score triggered the audit. The cart-session benchmark removed the explicit arithmetic identity but retained endpoint-dependent timing. The controlled reanalysis then separated these mechanisms under a common protocol. Table IV classifies every evaluation by purpose. *Diagnostic* evaluations locate leakage in completed-session features; their scores describe completed sessions and are not predictive performance. Only the *decision-time* evaluation estimates performance with information available at a decision, and it remains an offline estimate.

The main project report records ROC-AUC values of 0.6540 for logistic regression, 0.6860 for XGBoost, and 0.6654 for a hybrid LSTM after temporal variables were recomputed without transaction events. These results are useful corroborating evidence: the near-perfect performance disappears, and the more complex sequential model does not outperform the corrected tree baseline. They are not merged into the controlled ablation because they still summarize each complete non-transaction history. Excluding transaction events moves the endpoint, but the features still depend on an endpoint that becomes known only after the session has closed. The training scripts for that path are retained, but the reported values predate the present protocol and were not regenerated under its repeated-holdout design with training-only preprocessing. The workshop-stage version of this study therefore treats them as development context rather than primary evidence.

A. Arithmetic Target Reconstruction

The event vocabulary contains only three categories, while the session table stores total events, views, add-to-cart actions, and transaction count. Consequently,

$$\text{transaction_count} = \text{event_count} - \text{view_count} - \text{addtocart_count}. \quad (2)$$

Eq. 2 holds for all 43,917 rows in the supplied table. The binary label also satisfies

$$y_s = \mathbb{I}[\text{transaction_count}_s > 0] \quad (3)$$

for every row. The label therefore remains deterministically available after the explicit transaction column is removed, provided that the three component counts remain. This is an illegitimate-feature leakage mechanism in the sense of Kaufman et al. [6]. It explains why a linear classifier achieved an apparently exceptional score in the initial validation.

TABLE IV. DEVELOPMENT CHRONOLOGY, EVALUATIONS, AND THEIR PURPOSE

Stage	Population and model	ROC-AUC	What the stage revealed	Role
Initial signal validation	1,761,640 sessions; logistic regression, five-fold cross-validation	0.9969	Total count and component counts recover transaction count; scaling was fitted before cross-validation.	Audit trigger
Cart-session benchmark	43,917 cart sessions; XGBoost	0.9946	Removing total count did not remove completed-endpoint timing contamination.	Historical benchmark
Reported non-transaction timing	Logistic regression, XGBoost, and hybrid LSTM	0.654–0.686	Transaction events were excluded from timing, but features still used the completed non-transaction history.	Supporting context
Controlled audit	Same cart-session table; four feature configurations; repeated holdouts	0.606–1.000	Separates the deterministic count proxy, endpoint timing, and residual completed-session counts under one protocol.	Primary evidence (diagnostic)
Split robustness	Same configurations; visitor-grouped and chronological splits	0.610–1.000	The pattern does not depend on overlapping random splits; variability is larger.	Diagnostic check
Point-in-time evaluation	43,745 sessions at the first add-to-cart; four models; prefix features	0.57–0.65	Performance with information available at the decision, using the label of Eq. 4.	Decision-time estimate (offline)

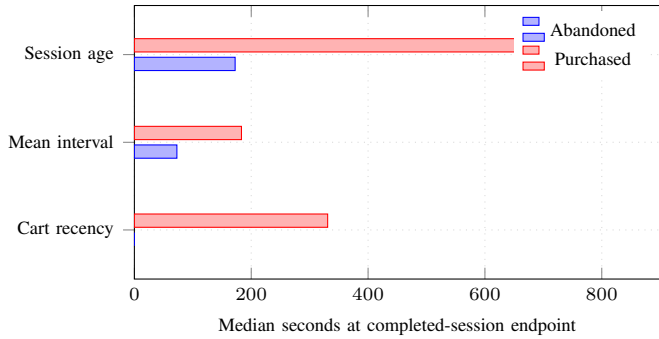


Fig. 2. Class-conditional medians of endpoint-dependent timing features. The differences may contain genuine behavior, but the completed endpoint is unavailable at a pre-outcome decision time.

B. Completed-Endpoint Timing Contamination

The later cart-session benchmark excluded `event_count` and therefore removed the exact identity. It nevertheless retained final session age, mean inter-event interval over the complete sequence, and recency from the last cart action to the final event. For purchased sessions, these calculations include the transaction event. For abandoned sessions, the endpoint is the final observed event before inactivity closes the session. The endpoint is therefore post-decision and selected differently by outcome.

Fig. 2 shows the class-conditional medians. Purchased sessions have a median completed-session age of 821.03 seconds, compared with 172.44 seconds for abandoned sessions. Median recency from the last cart action is 330.82 seconds for purchased sessions and zero for abandoned sessions. Moreover, 60.6% of abandoned sessions have zero cart recency because their final event is an add-to-cart action, compared with 2.1% of purchased sessions.

The distributional differences do not prove that every unit of predictive performance is caused by leakage. They show that a model can exploit information generated after the intended intervention point. A feature can be behaviorally meaningful and still be invalid for the claimed use case.

C. Label Ambiguity at a Decision Time

The first two mechanisms concern features. A third defect is located in the target itself and is therefore not repaired by any feature ablation.

The session-level rule assigns $y_s = 1$ when session s contains at least one transaction event. This is well defined for a completed session, but it does not describe the outcome of a specific cart decision. A visitor may purchase one item and, in the same session, add a second item to the cart and leave without buying it. The session-level rule records this as a purchase, although one cart action was in fact abandoned.

The audited artifact contains a measurable instance of the problem. By timing, 172 of the 43,917 cart sessions contain a transaction that occurs *before* the first add-to-cart event, and 1,573 contain a transaction strictly between the first and last add-to-cart action. Because 23 sessions satisfy both conditions, 1,722 distinct sessions (3.9%) are affected by timing. Timing alone is an upper bound, because a session may eventually buy every item it carted. An item-level check therefore compared carted and purchased item identifiers. Of the 1,722 sessions, 753 contain at least one carted item that was never purchased in the session. Across all purchased sessions, 1,354 (3.1% of cart sessions; 11.3% of purchased sessions) combine a purchase with a carted item that was never bought. The session-level rule labels these sessions purchased, although at least one cart action ended in abandonment.

A label consistent with an intervention-time task must be tied to the decision point rather than to the session:

$$y_{s,k} = \mathbb{I}[\exists e_{s,j} : \text{type}(e_{s,j}) = \text{transaction} \wedge t_{s,j} > t_{s,k}], \quad (4)$$

where $t_{s,k}$ is the cutoff timestamp. The outcome then belongs to the pair (session, decision point), and a session that has already converted by $t_{s,k}$ leaves the risk set rather than contributing a positive example.

Example construction. The cutoff is anchored at the first add-to-cart event $e_{s,a}$ of each session: $t_{s,k}$ is the timestamp of the k -th event after $e_{s,a}$, and $k = 0$ denotes $e_{s,a}$ itself. The observation window contains every event of s with timestamp at most $t_{s,k}$. The outcome window is the remainder of the same session, which ends after 30 minutes of inactivity; the prediction horizon is therefore finite and bounded by the session. A session enters the risk set at cutoff k only if it has at least k events after $e_{s,a}$ and no transaction at or before $t_{s,k}$. Later add-to-cart actions do not create new anchors; they enter the features of later cutoffs. The label is defined at session level after the cutoff: any transaction in the outcome window counts,

TABLE V. SEED-42 RESULTS WITH THE ORIGINAL COMPLETED-SESSION
FEATURE SET

Model	Acc.	Prec.	Recall	F1	ROC-AUC	PR-AUC
Logistic regression	.777	.591	.580	.585	.825	.599
Decision tree	.841	.642	.941	.763	.934	.834
Random forest	.941	.908	.871	.889	.984	.962
XGBoost	.964	.912	.963	.936	.995	.988

regardless of item or number of transactions. Item-level targets are left to future work. Under these rules, the 172 sessions whose transaction precedes the first add-to-cart never enter the risk set. The $k = 0$ set contains 43,745 sessions with a positive rate of 26.88%. Cutoffs $k \in \{0, 1, 2, 3, 5, 10\}$ yield 91,036 examples in total; this paper reports $k = 0$.

This defect is easy to overlook because it produces no implausible score. It does not inflate performance in the manner of Sections IV-A and IV-B; it silently changes the question being answered. The ablation confirms that it does not drive the results. Re-running the protocol of Table VI without the 1,722 timing-ambiguous sessions, or without the 1,354 item-verified sessions, leaves XGBoost at ROC-AUC 0.995 or higher with the proxy and original features and at 0.869 with timing alone. Counts-only ROC-AUC falls from 0.606 to between 0.578 and 0.595. The defect is reported because the same session table supports both formulations, and only one of them corresponds to a decision a retailer could act on.

D. Preprocessing Leakage

Two weaker implementation issues were also found. In the initial all-session notebook, standardization was fitted before cross-validation. In the original cart-session notebook, 99th-percentile clipping thresholds were estimated on the full dataset before the train/test split. Both choices expose test-distribution information to training, although their likely effect is smaller than that of the feature-level mechanisms. In the controlled reanalysis, clipping thresholds and scaling parameters were learned separately within each training partition.

V. RESULTS

The results answer four questions. Can the original near-perfect pattern be reproduced under isolated preprocessing? How does it change across the audited feature groups? Does it depend on the split protocol? What is achievable at a real decision point? The first three evaluations are diagnostic; the fourth is a decision-time estimate (Table IV).

A. Reproduction of the Original Benchmark

Using the original completed-session feature set and seed 42, the reanalysis reproduces the historical pattern closely. XGBoost reaches ROC-AUC 0.9948 and PR-AUC 0.9876; random forest reaches ROC-AUC 0.9837; decision tree reaches 0.9340; and logistic regression reaches 0.8249. The historical XGBoost value was 0.9946. The small difference is consistent with training-only clipping and software-version variation. Table V shows that, when interpreted without a feature audit, the task appears exceptionally easy and nonlinear tree ensembles appear decisively superior.



Fig. 3. Mean ROC-AUC across five repeated stratified holdouts. Counts-only remains a completed-session diagnostic, not an online evaluation.

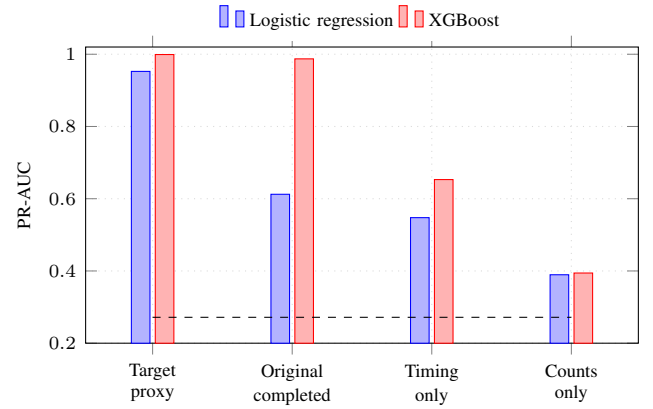


Fig. 4. Mean PR-AUC across five repeated stratified holdouts. The dashed line marks positive-class prevalence.

B. Ablation Across Leakage Pathways

Fig. 3 reports mean ROC-AUC across the five holdouts. The target proxy yields 0.9944 ± 0.0004 for logistic regression and 0.9997 ± 0.0001 for XGBoost. The original completed-session feature set retains an XGBoost score of 0.9947 ± 0.0005 after the exact count identity is removed. Timing features alone still yield 0.8688 ± 0.0035 . Once the three endpoint-dependent timing variables are removed, XGBoost falls to 0.6065 ± 0.0024 , while logistic regression reaches 0.6061 ± 0.0035 .

Fig. 4 shows the same inflation pattern for precision-recall performance. XGBoost falls from 0.9992 ± 0.0002 with the deterministic proxy to 0.9872 ± 0.0010 with the original completed-session set, 0.6530 ± 0.0106 with timing alone, and 0.3943 ± 0.0032 with counts alone. Positive-class prevalence is 0.2717, so the final condition retains modest signal but nothing resembling the near-perfect benchmark. Table VI provides the exact means and standard deviations used in Fig. 3 and 4.

C. Robustness to Visitor-Grouped and Chronological Splits

The stratified holdouts share visitors: 37,715 visitors contribute 43,917 sessions, and the holdouts ignore temporal drift. Table VII therefore repeats the ablation under visitor-grouped and chronological splits, with all other settings unchanged.

TABLE VI. REPEATED-HOLDOUT ABLATION (MEAN $\pm$ STANDARD DEVIATION)

Feature configuration	Model	ROC-AUC	PR-AUC
Target proxy	Logistic regression	.9944 $\pm$.0004	.9525 $\pm$.0042
	XGBoost	.9997 $\pm$.0001	.9992 $\pm$.0002
Original completed	Logistic regression	.8306 $\pm$.0036	.6123 $\pm$.0082
	XGBoost	.9947 $\pm$.0005	.9872 $\pm$.0010
Timing only	Logistic regression	.8006 $\pm$.0032	.5476 $\pm$.0094
	XGBoost	.8688 $\pm$.0035	.6530 $\pm$.0106
Counts only	Logistic regression	.6061 $\pm$.0035	.3895 $\pm$.0049
	XGBoost	.6065 $\pm$.0024	.3943 $\pm$.0032

TABLE VII. ABLATION UNDER VISITOR-GROUPED AND CHRONOLOGICAL SPLITS (ROC-AUC). STRATIFIED VALUES ARE IN TABLE VI

Feature configuration	Model	Visitor-grouped	Chronological
Target proxy	Logistic regression	.9934 $\pm$.0009	.9949
	XGBoost	.9997 $\pm$.0001	.9997
Original completed	Logistic regression	.8322 $\pm$.0053	.8423
	XGBoost	.9945 $\pm$.0011	.9965
Timing only	Logistic regression	.8040 $\pm$.0082	.8107
	XGBoost	.8713 $\pm$.0045	.8764
Counts only	Logistic regression	.6099 $\pm$.0170	.6200
	XGBoost	.6118 $\pm$.0111	.6178

The pattern is the same. XGBoost reaches 0.9997 with the target proxy under both splits, and 0.9945 (visitor-grouped) and 0.9965 (chronological) with the original completed-session features. Counts-only ROC-AUC lies between 0.610 and 0.620 for both models. The visitor-grouped standard deviations are larger than the stratified ones, up to ± 0.017 for counts-only logistic regression against ± 0.0035 . The stratified protocol therefore understates uncertainty, but the conclusions of Section V-B do not depend on it. Chronological scores are equal to or slightly higher than stratified scores in every configuration.

D. Point-in-Time Evaluation at the First Add-to-Cart

The counts-only configuration still summarizes completed sessions. To estimate what is achievable at an actual decision point, the cutoff-indexed label of Eq. 4 was evaluated at $k = 0$, the moment of the first add-to-cart event (Section IV-C). Thirteen features were computed from the observed prefix only (Table II, lower part). All four models used the settings of Table IX, with training-only capping and scaling, under the visitor-grouped and chronological splits.

Table VIII reports the results for 43,745 sessions with a positive rate of 26.88%. ROC-AUC lies between 0.57 and 0.65. Logistic regression reaches 0.630 (visitor-grouped) and 0.642 (chronological), and XGBoost reaches 0.622 and 0.633. The depth-limited decision tree is highest at 0.646 and 0.647, and the random forest is lowest at 0.582 and 0.568. PR-AUC ranges from 0.34 to 0.43 against a prevalence of 0.27. These values are close to the counts-only diagnostic and far from the completed-session scores. They are offline estimates under a point-in-time protocol. They are not online results, and no intervention was tested.

TABLE VIII. POINT-IN-TIME EVALUATION AT THE FIRST ADD-TO-CART EVENT ($k = 0$; 43,745 SESSIONS; POSITIVE RATE 0.269)

Model	Visitor-grouped		Chronological	
	ROC-AUC	PR-AUC	ROC-AUC	PR-AUC
Logistic regression	.6301 $\pm$.0065	.3778	.6415	.3861
XGBoost	.6218 $\pm$.0065	.3925	.6326	.3967
Decision tree [†]	.6458	.4319	.6467	.3863
Random forest [†]	.5816	.3773	.5677	.3416

Logistic regression and XGBoost: mean over five seeds. [†]Seed 42.

E. Interpretation of Model Complexity

Under the original feature set, XGBoost appears vastly superior to logistic regression (0.9947 against 0.8306). Under counts only, their mean ROC-AUC values differ by 0.0004. With four aggregate count features, this convergence is consistent with a feature ceiling: the remaining inputs offer little structure that a nonlinear model can exploit. The convergence therefore does not show that model choice is irrelevant, nor that XGBoost cannot benefit from richer valid features. It shows only that the large advantage observed with the original features depended on variables that violated the intended information boundary. The point-in-time evaluation, with thirteen prefix features, points the same way: logistic regression and XGBoost differ by about 0.01, and no model approaches the completed-session scores.

The ablation should not be interpreted as a precise causal decomposition. Removing a feature block changes the fitted function and can remove legitimate behavioral signal together with invalid information. The defensible conclusion is structural: the near-perfect result does not survive when variables tied to the completed endpoint are unavailable. Feature validity therefore had to be established before model rankings could be interpreted.

VI. DISCUSSION

The audit changes the scientific meaning of the original project. What first appeared to be a high-performing prediction system is better understood as evidence that the data representation answered a different, retrospective question.

A. What Completed Sessions Actually Predict

The first failure mode is deterministic. Removing the obvious label column does not prevent leakage when the same label-generating count can be recovered algebraically from other inputs. Feature importance is a weak defense because a nonlinear model can distribute the identity across several columns. The relevant test is arithmetic closure: whether the target or a target-defining event count can be reconstructed from the remaining feature set.

Eq. 2 is trivial once written down: with a three-event vocabulary, the identity holds by construction. Its significance lies elsewhere. The identity passed through an ordinary pipeline of session aggregation, model training, and reporting without being noticed, and it produced a score that looked publishable. Concurrent work reports comparable post-session descriptors in other clickstream benchmarks [13]. This suggests that the pattern is not specific to one prototype, although the present study does not test other datasets.

The second failure mode is temporal. Final duration and recency variables can be commercially meaningful, yet they describe what the complete session looked like after its outcome was known. A pre-outcome model asks what was observable at a specified event or clock time. Completed-session construction silently shifts the observation boundary to the endpoint and makes that endpoint outcome-dependent.

There is no universal ROC-AUC threshold above which leakage must exist. Nevertheless, an implausibly strong score for a noisy behavioral outcome should trigger a feature-lineage audit rather than immediate optimization. A high offline score is not by itself proof of leakage. Esmeli and Gokce report ROC-AUC of 0.990 on a comparable post-add-to-cart task, and their reported feature set contains neither an arithmetically closed count group nor endpoint-anchored timing variables [9]. The claim advanced here is correspondingly narrower. In the audited artifact the specific mechanisms were located, and the score did not survive their removal. The strongest scientific result is therefore not the near-perfect score; it is the demonstration that the score did not measure the intended intervention-time task.

B. Practical Leakage-Audit Checklist

Before a session-level prediction result is accepted, nine questions should be answered:

- What exact event or time defines the prediction moment?
- Which event creates the label?
- Does the label describe the outcome of the specific decision, or only of the session that contains it?
- Can that event count be reconstructed from other counts?
- Does any duration, recency, or average use the final session event?
- Is the endpoint selected differently for positive and negative cases?
- Are clipping, scaling, feature selection, and imputation fitted using training data only?
- Does performance remain plausible after suspicious feature groups are removed?
- Can the experiment be rerun from code and preserved outputs?

The checklist is intentionally simple. None of the nine questions requires additional computation; each can be answered from the data dictionary, the preparation code, and the definition of the intended intervention. Leakage often survives because attention is directed toward architecture and hyperparameters while the semantics of a few columns remain unexamined. The checklist was derived from, and applied to, a single artifact. Its completeness, reliability, and false-positive behavior on independent datasets have not been evaluated, so it is offered as a structured starting point rather than a validated detection instrument.

C. Implications for E-Commerce Experiments

The intended business action should be specified before data are aggregated. A discount displayed immediately after the first cart action, a reminder issued after two minutes of inactivity, and a follow-up email sent after session expiration are three different prediction tasks. Each permits a different feature set and requires a different outcome horizon. Treating them all as “cart-abandonment prediction” obscures these boundaries.

A genuine in-session study should assign every training example an explicit cutoff timestamp, an observation window ending at that cutoff, and an outcome window beginning afterward. Features should be regenerated from each observed prefix, and labels should be assigned according to Eq. 4 rather than at the session level. This design is more expensive than storing one row per completed session, but it prevents the target event from participating in its own prediction and allows performance to be studied as a function of decision time.

D. Cost and Pipeline Integration

At the scale of this study the cost is modest. For six cutoffs, the point-in-time construction turned 43,917 sessions into 91,036 examples, 2.1 times as many rows. It took about 13 minutes with single-threaded pandas on a laptop, including cleaning and sessionization of the raw log. Building the completed-session table from the same log took about 10 minutes. The cost grows with the number of cutoffs and with events per session, not with model complexity.

The same design fits streaming and machine-learning pipelines. A stream processor can maintain the prefix features incrementally, one event at a time, using event-time session windows and watermarks. Each emitted feature vector carries its cutoff timestamp. Labels are attached later, when the outcome window closes. Training sets are then assembled by point-in-time joins on session and cutoff time, the operation that feature stores provide. The main difficulty is discipline rather than computation: every feature must carry the time at which it became available, and every join must respect that time.

E. Threats to Validity

All findings come from one public dataset, one sessionization scheme with a 30-minute inactivity boundary, and a limited set of feature definitions. The claims are therefore restricted to the audited artifact. The mechanisms are plausible elsewhere, but their prevalence and magnitude in other datasets, event taxonomies, or session definitions were not measured. The stratified holdouts share observations, and some visitors contribute multiple sessions. Visitor-grouped and chronological splits (Table VII) leave the conclusions unchanged but show larger variability, so the stratified standard deviations understate uncertainty.

The counts-only condition is not a leakage-free production benchmark because its counts are computed after the complete session. Likewise, timing-only discrimination may combine genuine behavior with invalid future information, so the experiment cannot assign every unit of AUC causally to leakage. The main project report describes a later recomputation that excludes transaction events, but it also remains a completed-history evaluation. Those scores were produced before the

present protocol was defined and were not regenerated under its five-seed, training-only preprocessing design. The 0.654–0.686 results are therefore treated as supporting context only. The point-in-time evaluation covers one cutoff and four standard models; it is an offline estimate, and no live deployment or intervention experiment was performed.

VII. REPRODUCIBILITY STATEMENT

The primary audit has been re-executed from the 43,917-row derived cart-session table and a standalone reanalysis script. A submission package should include that table, the exact five seed values, generated metric files, figure inputs, and an environment specification. Before model fitting, the script should assert the count identity in Eq. 2 and the target identity in Eq. 3. Clipping and scaling parameters must then be learned from each training partition only.

The event-level preparation path has since been re-executed from the raw file. Re-running the original cleaning rules on the 2,756,101 raw events reproduces the reported figures exactly: 35 suspected automated visitors and 491 events are removed, leaving 2,755,610 events. The result matches the retained cleaned artifact row for row on visitor identifier, timestamp, event type, and item identifier. The removed events comprise 431 views and 60 add-to-cart actions and no transactions, indicating that the heuristic filter discarded no converting visitor. The controlled comparisons were independently re-executed under the protocol of Table IX using scikit-learn 1.8.0, XGBoost 3.3.0, NumPy 2.3.3, and pandas 2.3.2. Nineteen of the twenty reported cells in Tables V and VI reproduced within ± 0.0007 . The exception is the timing-only XGBoost PR-AUC, which returned 0.6563 against the reported 0.6530. This cell has the largest reported standard deviation (± 0.0106), and the difference lies within one standard error of the reported mean. The published values are retained, and the re-execution values are recorded in the accompanying provenance table rather than substituted for them. The additions in this version (split robustness, item-level label verification, and the point-in-time evaluation) were produced by separate scripts that reuse the same settings. A further script rebuilds the cart-session table from the raw events byte for byte and recomputes every descriptive figure quoted in the paper.

A provenance table mapping each published figure to the script that produces it is distributed with the submission package. One figure in the underlying project report, the log of intermediate LSTM development iterations, is recorded there as untraced: no sweep code for those configurations was preserved, and no value from that table is used as evidence in this paper. Table IX summarizes the settings required to rerun the controlled comparisons. The objective is not to optimize a state-of-the-art classifier, but to test whether the interpretation changes when invalid feature groups are removed.

The deterministic identities, class-conditional summaries, seed-42 four-model benchmark, and five-holdout logistic regression and XGBoost ablation are reproducible when the described aggregate table and audit script are supplied. The cleaned event stream and the training scripts used for the later non-transaction timing experiment are also retained, but those scores were generated under an earlier single-split protocol and were not regenerated for this paper. The paper therefore

TABLE IX. MODEL AND PREPROCESSING SETTINGS IN THE AUDIT

Component	Setting
Data split	Stratified 80/20 and visitor-grouped 80/20 (seeds 11, 22, 33, 42, 55); chronological 80/20 by session start
Outlier handling	Upper 99th-percentile cap fitted on each training partition
Logistic regression	Balanced class weights; 2,000 maximum iterations; training-only standardization
Decision tree	Balanced class weights; maximum depth 8; seed 42 (reproduction and point-in-time evaluation)
Random forest	100 trees; balanced class weights; seed 42 (reproduction and point-in-time evaluation)
XGBoost	100 trees; class-ratio weight; histogram tree method
Primary metrics	ROC-AUC and PR-AUC
Secondary metrics	Accuracy, precision, recall, and F1 at threshold 0.5

excludes them from its primary tables and restricts its central claims to results obtained under one common rerun protocol.

VIII. CONCLUSION

Completed-session tables can make cart-abandonment prediction appear much easier than the intended intervention-time task. In the audited artifact, one group of counts reconstructed the transaction label exactly, while completed-endpoint timing preserved near-perfect XGBoost performance after that identity was removed. Across the diagnostic configurations, mean XGBoost ROC-AUC was 0.9997 with the deterministic proxy, 0.9947 with the original completed-session features, and 0.6065 after endpoint-dependent timing was removed. Visitor-grouped and chronological splits reproduced this pattern. At the first add-to-cart event, where models saw only the observed prefix, ROC-AUC lay between 0.57 and 0.65 across four models.

The conclusion is methodological rather than algorithmic: an offline score is not interpretable until every feature is mapped to the stated decision time. Within the audited artifact, the near-perfect score measured completed sessions, not the intervention-time task. The next steps are to extend the point-in-time evaluation to later cutoffs and item-level targets, and to test the mechanisms on other datasets and session definitions. Throughout, a model must be tested only on information that could have been observed when a real intervention was still possible.

ACKNOWLEDGMENT

The authors thank Bahçeşehir University's Department of Artificial Intelligence Engineering for the computational resources and course environment (Big Data and Analytics) that supported this study.

DECLARATIONS

Conflict of Interest

The authors declare no conflict of interest.

Funding

This research received no external funding.

Declaration on Generative AI

This project was developed with the assistance of generative AI systems (Claude, Anthropic; ChatGPT, OpenAI) over a period of approximately seven months. Their use included iterative assistance with analysis code, exploratory discussion of candidate interpretations of observed patterns, and drafting and formatting support for the manuscript, including its conversion to the journal's template. Throughout the project, the authors directed the research questions and overall audit design, executed and validated all code against the underlying data, and critically evaluated, tested, and verified every interpretive claim before its inclusion in the manuscript (see the Reproducibility Statement for details on the independent re-verification of the quantitative results). The authors assume full responsibility for the accuracy, originality, and scientific validity of the manuscript. No generative AI system is listed as an author.

REFERENCES

- [1] R. Esmeli, M. Bader-El-Den, and H. Abdullahi, "Towards early purchase intention prediction in online session based retailing systems," *Electronic Markets*, vol. 31, no. 3, pp. 697–715, 2021.
- [2] B. Requena, G. Cassani, J. Tagliabue, C. Greco, and L. Lacasa, "Shopper intent prediction from clickstream e-commerce data with minimal browsing information," *Scientific Reports*, vol. 10, p. 16983, 2020.
- [3] C. O. Sakar, S. O. Polat, M. Katircioglu, and Y. Castro, "Real-time prediction of online shoppers' purchasing intention using multilayer perceptron and lstm recurrent neural networks," *Neural Computing and Applications*, vol. 31, pp. 6893–6908, 2019.
- [4] T. M. Rausch, N. D. Derra, and L. Wolf, "Predicting online shopping cart abandonment with machine learning approaches," *International Journal of Market Research*, vol. 64, no. 1, pp. 89–112, 2022.
- [5] F. Waldmann, G. Nápoles, and Y. Salgueiro, "Prediction of cart abandonment using imbalanced clickstream data in online shopping," 2025, advance online publication, *Electronic Commerce Research*.
- [6] S. Kaufman, S. Rosset, C. Perlich, and O. Stitelman, "Leakage in data mining: Formulation, detection, and avoidance," *ACM Transactions on Knowledge Discovery from Data*, vol. 6, no. 4, pp. 15:1–15:21, 2012.
- [7] S. Kapoor and A. Narayanan, "Leakage and the reproducibility crisis in machine-learning-based science," *Patterns*, vol. 4, no. 9, p. 100804, 2023.
- [8] H. Suleiman, E. Idreis, Y. Allama, D. Naser, and L. Alawneh, "Post-add-to-cart purchase prediction using machine learning," in *Proceedings of the 17th International Conference on Information and Communication Systems*. New York, NY, USA: Association for Computing Machinery, 2026.
- [9] R. Esmeli and A. Gokce, "An analysis of consumer purchase behavior following cart addition in e-commerce utilizing explainable artificial intelligence," *Journal of Theoretical and Applied Electronic Commerce Research*, vol. 20, no. 1, p. 28, 2025.
- [10] J. Fabra, P. Álvarez, and J. Ezpeleta, "Log-based session profiling and online behavioral prediction in e-commerce websites," *IEEE Access*, vol. 8, pp. 171 834–171 850, 2020.
- [11] S. Kapoor, E. Cantrell, K. Peng, T. H. Pham, C. A. Bail, O. E. Gundersen, J. M. Hofman, J. Hullman, M. A. Lones, M. M. Malik, P. Nanayakkara, R. A. Poldrack, I. D. Raji, M. Roberts, M. J. Salganik, M. Serra-Garcia, B. M. Stewart, G. Vandewiele, and A. Narayanan, "Reforms: Consensus-based recommendations for machine-learning-based science," *Science Advances*, vol. 10, no. 18, p. eadk3452, 2024.
- [12] Y. Ji, A. Sun, J. Zhang, and C. Li, "A critical study on data leakage in recommender system offline evaluation," *ACM Transactions on Information Systems*, vol. 41, no. 3, pp. 75:1–75:27, 2023.
- [13] M. Alzghoul, M. Baklizi, M. Alkhazaleh, O. Qtaish, and H. Alshawabkeh, "A leakage-audited prefix-only protocol for session-based consumer navigation prediction in e-commerce clickstream data," *Ingénierie des Systèmes d'Information*, vol. 31, no. 8, pp. 2215–2229, 2026.
- [14] M. Ferrari Dacrema, P. Cremonesi, and D. Jannach, "Are we really making much progress? a worrying analysis of recent neural recommendation approaches," in *Proceedings of the 13th ACM Conference on Recommender Systems*. New York, NY, USA: Association for Computing Machinery, 2019, pp. 101–109.
- [15] D. Sculley, G. Holt, D. Golovin, E. Davydov, T. Phillips, D. Ebner, V. Chaudhary, M. Young, J.-F. Crespo, and D. Dennison, "Hidden technical debt in machine learning systems," in *Advances in Neural Information Processing Systems 28*. Red Hook, NY, USA: Curran Associates, Inc., 2015, pp. 2503–2511.
- [16] Retail Rocket, "Retailrocket recommender system dataset," Kaggle dataset, 2015, accessed July 2026. [Online]. Available: <https://www.kaggle.com/datasets/retailrocket/ecommerce-dataset>
- [17] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: Association for Computing Machinery, 2016, pp. 785–794.
- [18] L. Breiman, "Random forests," *Machine Learning*, vol. 45, pp. 5–32, 2001.
- [19] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.