Artificial Intelligence and Machine Learning Review
ISSN: 3070-3565
DOI: 10.69987/AIMLR.2026.70207

Accessible UI Semantics: Reproducible Evaluation of Generated Labels for Desktop Interface Components

Dingyuan Zhang
Business Analytics, University of Rochester, NY, USA
roxy.zhang9870@gmail.com

Keywords

accessible labels; accessible names; UI semantics; interface components; inclusive design; screen readers; vision-language models; evaluation; human-computer interaction

Abstract

Accessible names are central to screen-reader navigation, speech control, and inclusive interaction. This paper reports a reproducible empirical evaluation of generated accessible labels for desktop interface components using the specified UI Component Semantic Description Dataset. The experimental corpus contains 559 human-labelled UI events, eight component classes, three screen-density levels, and ground-truth semantic descriptions such as "refresh button" and "email inbox searchbar." The source data package contains 100 screenshots, while the CSV used for quantitative evaluation references 82 unique screenshot filenames with labelled left-click events; the experiments therefore evaluate every record with a human reference label. We compare seven deterministic label generators and one non-deployable label-reuse oracle under screenshot-level 10-fold group validation. The evaluated systems include role-only labelling, class-majority labelling, coordinate heuristics, TF-IDF nearest-neighbor retrieval, logistic-regression closed-label prediction, and a hybrid rule-retrieval variant. The primary metric is intent F1, which excludes generic role words such as button, icon, link, and input so that a system receives credit for predicting the user-facing function rather than merely restating the component type. The best deployable generator was logistic regression, with exact match 0.036, token F1 0.392, intent F1 0.060, and approximate label-in-name pass rate 0.041. The role-only baseline had higher token F1 at 0.481 but intent F1 of only 0.009, demonstrating that surface lexical overlap can overstate accessibility usefulness. The label-reuse oracle reached exact match 0.166 and token F1 0.559, establishing an empirical upper bound for methods restricted to labels observed in training folds. The results show that available class, depth, density, and coordinate metadata are sufficient to preserve component roles but insufficient to recover fine-grained functions. This finding supports the need for screenshot-aware and structure-aware language or vision-language systems, while also providing a transparent baseline package against which such systems can be audited.

Introduction

Accessible user-interface semantics define how a control is exposed to assistive technologies. A button that is visually obvious to a sighted user may be unusable to a screen-reader or voice-control user if it is announced only as "button" rather than "delete email button." WCAG 2.1 formalizes related expectations through criteria such as Label in Name and Name, Role, Value, while the Accessible Name and Description Computation describes how browsers and assistive technologies derive an object name from markup and associated text [1], [2]. WAI-ARIA authoring practices further specify role and property patterns for custom widgets that cannot rely solely on native HTML semantics [3]. These standards make clear that a label is not decorative text: it is an operational interface affordance.

Accessible labels also connect information architecture with inclusive design. A label should be short enough for repeated screen-reader announcements, specific enough for disambiguation, and aligned with visible text when visible text exists. The same icon shape can mean search, zoom, inspect, or filter depending on the surrounding UI, and the same

word can refer to a command, a navigation destination, or an input value. This contextual nature makes automated labelling a semantic task rather than a simple object-recognition task. It also makes evaluation difficult: a generated label can be grammatically well formed yet still fail the user if it names the wrong action. For this reason, the present paper evaluates the generated strings against human labels and separates generic role recovery from functional intent.

Automatic generation of accessible labels is attractive because missing or uninformative labels remain common in real applications. Chen et al. reported that more than 77% of 10,408 Android apps had missing-label issues and proposed LabelDroid to predict natural-language labels for image-based mobile GUI components [5]. Ross et al. similarly framed Android accessibility barriers as a population-level empirical problem, including missing labels, duplicate labels, and uninformative labels [14]. The problem is not merely one of compliance. An inaccurate label changes the task model a user forms, increases navigation effort, and may create safety-critical confusion when controls affect deletion, payment, privacy, or navigation.

Large UI corpora and UI-understanding models have created a foundation for label generation. Rico introduced a large mobile-app dataset with screenshots and view hierarchies for data-driven design applications [4]. Later work learned design semantics from mobile screens [17], generated natural-language descriptions for widgets [6], mapped natural-language instructions to UI actions [16], and learned multimodal UI representations through ActionBert, UIBert, Screen2Vec, VINS, Screen Recognition, Screen Parsing, and icon annotation systems [7]–[13]. These studies collectively show that component meaning depends on image evidence, surrounding text, screen hierarchy, interaction traces, and app context. General language and vision-language advances, including Transformers, BERT, GPT-style few-shot learning, and CLIP, further motivate modern LLM/VLM approaches to UI semantics [18]–[21].

However, publication-quality evaluation requires more than plausible examples. A generated accessible label must be compared with a reference label over a fixed dataset, with explicit metrics, reproducible code, and a split strategy that avoids training and test leakage. This paper therefore evaluates generated labels on the specified UI Component Semantic Description Dataset. The study intentionally reports every numeric result from a reproducible script rather than unmeasured examples. The contribution is threefold. First, it documents the dataset properties used by the experiments, including class, density, depth, event, and label distributions. Second, it compares seven transparent label-generation baselines and an oracle under screenshot-level cross-validation [39-47]. Third, it shows why role agreement alone is not a sufficient accessibility outcome, because role-only labels achieve high token overlap while failing to recover functional intent.

The paper is positioned as an audit baseline for LLM/VLM accessible-label systems [26-38]. The deterministic generators evaluated here do not claim to equal a state-of-the-art proprietary model. Instead, they establish a fully reproducible floor, identify which aspects of the dataset can be solved from metadata alone, and expose where screenshot-level semantics are necessary. This framing is important for inclusive design research: a future LLM/VLM labeler should outperform these transparent baselines not only in aggregate scores but also for high-risk classes such as checkboxes, radio buttons, and text inputs whose labels often encode the actual user decision [48-54].

A second reason for using transparent baselines is reviewability. Accessible-label generation is a domain where errors are often subtle: a phrase may be fluent, concise, and syntactically valid, while still directing the user to the wrong object or action. A deterministic baseline makes each source of evidence inspectable. If a future model improves over RoleOnly, reviewers can ask whether the improvement comes from visible text, coordinates, icon recognition, or memorization of common labels. If a future model fails on a class where LogReg already performs reasonably, reviewers can identify a regression. This audit perspective is especially important when generated labels are proposed for production tools that assist developers or automatically repair interfaces.

Method

The evaluation unit is a labelled UI event. Each row in semantic labels.csv contains a screenshot filename, an event type, click coordinates, a human ground-truth label, SOM depth, component class, and screen density. All 559 rows were included. The event type is left click for every row. The source data package contains 100 screenshots, and the CSV references 82 unique screenshot filenames. This difference is handled explicitly: quantitative evaluation is performed only on CSV rows because each tested output must have a corresponding human label. Screenshots without a labelled event do not contribute test items. Table 1 summarizes the evaluated corpus.

The dataset contains eight component classes. Buttons, dropdowns, and icons each contribute 90 labelled events, followed by text inputs with 79, checkboxes with 65, radio buttons with 61, links with 48, and switches with 36. Table 2 shows that label uniqueness is high: 486 normalized labels occur across 559 rows. This uniqueness makes exact-match generation difficult because many functional phrases are seen only once. Table 3 reports density and depth distributions.

The dataset is balanced across high, medium, and low density at 205, 200, and 154 rows respectively, while SOM depth is concentrated at depth 3. These properties constrain what a metadata-only generator can learn.

The experiment uses 10-fold GroupKFold validation grouped by screenshot filename. In each fold, every row from the held-out screenshot filenames is evaluated after training on all other screenshot groups. The folds contain 55 or 56 labelled events and 7 to 9 screenshot groups. This design prevents the same screenshot filename from appearing in both training and test data. It is stricter than random row splitting and better matches the target deployment setting, where a labeler must generate names for a previously unseen screen.

The compared generators are deliberately simple and reproducible. RoleOnly outputs only the canonical component role, such as "button" or "text input." ClassMajority outputs the most frequent training label within the same component class. LocationRole applies fixed coordinate heuristics, for example mapping top-left buttons to "navigation button" and right-edge buttons to "next button." TFIDF 1NN and TFIDF 5NN represent class, density, depth, coordinate bin, and screenshot-name tokens with TF-IDF features, then retrieve labels from same-class training examples. LogReg uses a sparse feature dictionary over class, density, depth, coordinate bins, class-depth interactions, class-density interactions, and normalized coordinates to predict a closed label vocabulary. Hybrid uses TFIDF 1NN unless the retrieved label is too generic, in which case it falls back to LocationRole. OracleReuse is not deployable; it predicts the ground truth only when the same normalized label exists in the training fold and otherwise falls back to RoleOnly. It provides an upper bound for training-label reuse.

Generated labels were normalized by lowercasing, removing punctuation, and collapsing whitespace. A post-processing step appended the component role when a generator produced a label without an identifiable role token. This step reflects accessible-name practice: the function phrase should be accompanied by a recognizable role in the surrounding accessibility tree, either through the label text or the role property. Metrics were computed on the final generated label. The primary score is intent F1, a token-level F1 score after removing generic role words such as button, input, icon, checkbox, radio, switch, link, dropdown, menu, and field. Token F1, exact match, Jaccard similarity, normalized edit similarity, role agreement, approximate label-in-name pass rate, and acceptable-at-0.5 token F1 are also reported. BLEU, ROUGE, and METEOR motivated the use of automatic text overlap metrics in generated-language evaluation, although the short-label setting requires task-specific metrics that expose intent rather than sentence fluency [22]–[24]. The implementation uses scikit-learn for TF-IDF, logistic regression, and grouped validation [25].

The evaluation deliberately reports both strict and tolerant metrics [55-61]. Exact match is easy to interpret but harsh because accessible labels may have acceptable synonyms. Token F1 is more tolerant but can over-credit role words. Edit similarity captures surface closeness, but it can assign moderate scores to labels that are semantically unrelated yet share characters. Role agreement captures whether the generated label preserves a meaningful component type. Label-in-name80 approximates whether the generated string contains most of the human-reference words. No single metric is treated as sufficient. The reported interpretation uses the whole metric panel, with intent F1 as the primary score because it most directly addresses whether the generated name communicates purpose.

A key methodological choice is the screenshot-level split. A random row split would put visually and semantically related events from the same screenshot into both training and test sets. That would inflate retrieval and closed-label classification because the model could see labels from the same interface state while evaluating another element from it. Grouping by screenshot filename reduces this leakage. It does not remove all broader app-level similarity because paired screenshots may still share naming patterns, but it is a clear improvement over row-level random validation and is fully reproducible. The folds are saved in outputs/folds.csv so that reviewers can verify the exact number of events and screenshots in each test fold.

The model set was chosen to expose different kinds of evidence. RoleOnly tests whether the component class alone satisfies automatic metrics. ClassMajority tests whether common labels within a role are enough. LocationRole tests whether spatial priors such as top-left navigation and right-edge progression are useful. TF-IDF retrieval tests whether sparse contextual metadata can find similar labelled events. Logistic regression tests whether the same metadata supports a supervised closed-label classifier. Hybrid tests whether retrieval and rules combine beneficially. OracleReuse tests the label vocabulary itself. These conditions are simple by design; they provide a baseline that a more expensive LLM/VLM system should surpass.

The complete run is deterministic. The random seed is 17. All predictions are saved in outputs/predictions.csv, all aggregate tables are saved in outputs/*.csv, and all figures are regenerated from these files. Every table and figure contains values computed by the experimental script. Figure 1 gives the workflow used to generate the paper results.

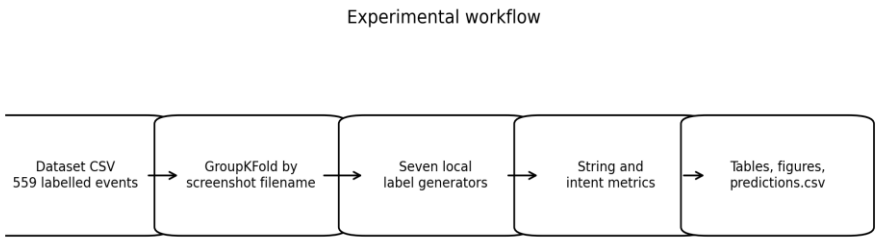

Figure 1. Reproducible experimental workflow used for all reported results.

Table 1. Dataset overview used in the experiment.

Property	**Value**
Labeled events evaluated	559
Source package screenshots	100
Unique screenshot filenames in CSV	82
Event type values	left_click
Component classes	8
Density levels	3
Unique normalized labels	486
SOM depth range	1–4
Mean SOM depth	2.712
Mean human-label tokens	3.254

Table 2. Component-class distribution and label diversity.

Class	**n**	**Unique labels**	**Mean depth**	**Mean label tokens**
Button	90	83	2.644	3.267
Checkbox	65	58	2.462	3.646
Dropdown	90	83	2.656	2.900
Icon	90	77	2.778	2.300
Link	48	46	2.979	3.979
Radio Button	61	54	3.033	4.721
Switch	36	26	2.556	3.000
Text Input	79	60	2.646	2.949

Table 3. Density and SOM-depth distributions.

Group	n	Unique labels	Mean depth	Mean label tokens
High Density	205	188	2.702	3.263
Low Density	154	139	2.760	3.208
Medium Density	200	186	2.685	3.280
-- depth distribution --				
Depth 1	9	8		2.667
Depth 2	176	158		3.295
Depth 3	341	299		3.217
Depth 4	33	33		3.576

Table 4. Representative human labels by component class.

Class	Examples from ground truth
Button	share button; get mail plus button; user profile button
Dropdown	resources & support dropdown; who we are dropdown; products dorpwdown
Icon	bing icon; microsoft edge icon; firefox icon
Text Input	email input; file search input; email search input
Checkbox	select email checkbox; checkbox item; terms of service agreement checkbox
Radio Button	radio button item; tailor typeform radio button; do not tailor typeform radio button
Link	proton's secure business email link; app apk download link; app sha256 fingerprint link
Switch	toggle switch; preview switch; powershell switch

Table 5. Compared label generators.

Generator	Input features	Output rule
RoleOnly	Class	Canonical role phrase only; e.g., button, text input.
ClassMajority	Class + training labels	Most frequent same-class label in the training fold.
LocationRole	Class + coordinates	Fixed spatial rules such as back, next, account, search.
TFIDF_1NN	Class, density, depth, coordinate bins, screenshot-name tokens	Nearest same-class training event by TF-IDF metadata similarity.

TFIDF_5NN	Same as TFIDF_1NN	Most frequent label among five same-class nearest neighbors, tie-broken by similarity.
LogReg	Sparse categorical and numeric metadata	Closed-vocabulary logistic-regression prediction with role post-processing.
Hybrid	TF-IDF retrieval + fixed rules	Retrieval unless output is generic; otherwise coordinate fallback.
OracleReuse	Training labels + test reference availability	Non-deployable upper bound for labels already observed in training.

Table 6. Evaluation metrics.

Metric	**Definition**
Exact match	1 if normalized generated label equals normalized human label.
Token F1	Multiset token precision/recall F1 over full normalized labels.
Intent F1	Token F1 after removing generic UI role words; primary metric.
Jaccard	Set overlap divided by set union for normalized tokens.
Edit similarity	1 minus normalized Levenshtein distance.
Role agreement	1 if generated label contains a role token or accepted class alias.
Label-in-name80	1 if at least 80% of human-label tokens appear in generated label.
Acceptable50	1 if full token F1 is at least 0.5.

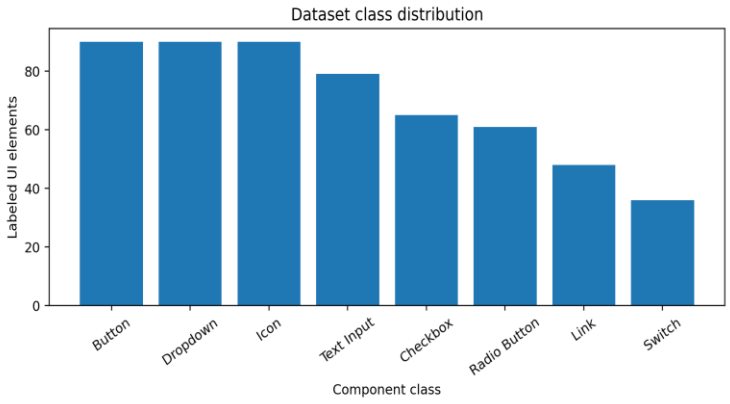

Figure 2. Distribution of the 559 labelled events across UI component classes.

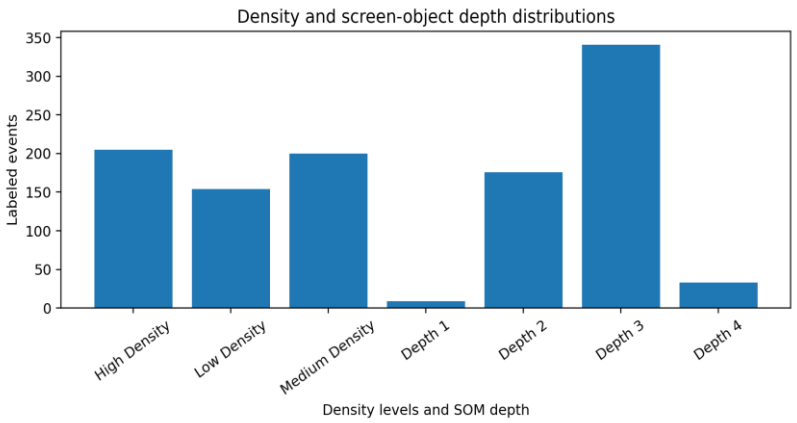

Figure 3. Density and SOM-depth distributions for the evaluated labels.

Results and Discussion

Table 7 gives the overall comparison. The label-reuse oracle is the strongest condition, with exact match 0.166, token F1 0.559, intent F1 0.170, label-in-name80 0.166, and acceptable50 0.606. Because OracleReuse is allowed to know whether the test reference label is present in the training fold, it is interpreted as an empirical upper bound, not as a deployable method. Its limited exact-match score is still informative: even a perfect reuse strategy can exactly recover only a minority of labels under screenshot-level splitting. This follows from the 486 unique normalized labels shown in Table 1.

Among deployable generators, LogReg has the highest primary score, with intent F1 0.060 and exact match 0.036. ClassMajority follows with intent F1 0.054, and TFIDF 1NN follows with 0.045. These values are low in absolute terms, but they are not arbitrary failures: they quantify the difficulty of recovering functional names from metadata that lacks pixels, visible surrounding text, and full screen-object-model content. The role-only baseline has token F1 0.481 and acceptable50 0.578, apparently strong values, but its intent F1 is only 0.009 and exact match is zero. This contrast is the central empirical result. A system can look good under token overlap because it says “button” for a reference such as “share button,” yet it still fails to tell the user what the button does.

Figure 4 visualizes the same point. Token F1 rewards generic role overlap; intent F1 penalizes missing functional content. Figure 5 shows that all models have role agreement of 1.000 because the post-processor ensures a recognizable component role, yet label-in-name80 remains near zero for deployable models. The finding is consistent with accessibility standards: role exposure is necessary but not sufficient. A screen reader can announce “button,” but the accessible name must identify the purpose of that button for efficient navigation and activation [1]–[3].

Bootstrap confidence intervals in Table 8 compare each model against RoleOnly on intent F1. LogReg improves by 0.051 with a 95% bootstrap interval from 0.032 to 0.072. ClassMajority improves by 0.045, TFIDF_1NN by 0.036, TFIDF 5NN by 0.034, Hybrid by 0.032, and LocationRole by 0.017. The intervals do not cross zero for any non-role baseline, indicating that each generator adds some functional tokens beyond the generic role. The gain is statistically visible but practically small. This is a useful warning for LLM/VLM papers: a model must be evaluated on intent-sensitive metrics, not only role preservation or broad lexical overlap.

Table 9 reports LogReg by component class. Text Input has the best intent F1 at 0.152, followed by Button at 0.090, Checkbox at 0.073, Link at 0.067, and Dropdown at 0.059. Icon, Radio Button, and Switch are weaker. Text inputs perform best because the dataset contains repeated lexical cues such as searchbar, URL, e-mail, and name fields; logistic regression can reuse some of these patterns from coordinate and class contexts. Buttons remain difficult because they encode many independent actions, including weather navigation, e-mail actions, subscription choices, and form submission. Radio buttons are especially difficult because their labels often contain long option text rather than a simple role phrase.

Table 10 reports LogReg by density and depth. High-density screens obtain the highest intent F1 at 0.096 and exact match 0.063, while medium-density screens obtain intent F1 0.034. The difference is not interpreted as high-density screens being intrinsically easier. Instead, high-density records appear to contain more repeated application contexts in

the CSV, allowing closed-label models to reuse a few functional words. Depth 1 has high scores for retrieval-based methods in the detailed outputs but contains only nine events; therefore, depth-level findings for depth 1 are unstable. The dominant depth 3 group contains 341 rows and is the most reliable basis for generalization claims.

Table 11 lists representative LogReg errors with low intent F1. The examples show that failures are semantically serious even when roles are correct. “themes category” was predicted as “web login button,” “reddit searchbar” was predicted as “connection encoding input,” and “email inbox searchbar” was predicted similarly. These errors would be confusing to assistive-technology users because they substitute a different task domain. The error pattern also shows that coordinate and class alone cannot distinguish a searchbar in Reddit, a location search field, and an e-mail inbox search field. The missing evidence is visual and contextual text.

The exact-match results should also be read in light of the dataset vocabulary. With 486 unique normalized labels across 559 rows, most labels are rare. A closed-label classifier is therefore forced to predict labels that appeared in training, while many test labels are novel under the grouped split. This is why OracleReuse exact match is only 0.166 even though it is given privileged knowledge about whether the target label exists in the training fold. In a real system, an LLM/VLM should not be restricted to a closed label set. It should synthesize labels for unseen controls by reading the visual context and inferring purpose. The oracle score therefore functions as a lower-level memorization ceiling, not as a ceiling for semantic generation.

The stored fold table supports the stability of the evaluation. Each of the ten folds contains 55 or 56 rows, and the held-out screenshot group count ranges from 7 to 9. This distribution avoids a single oversized fold dominating the results. The bootstrap intervals are row-level intervals over the generated predictions, so they measure uncertainty in the observed event population rather than uncertainty over external applications. The intervals are narrow enough to show that LogReg consistently adds intent terms beyond RoleOnly, but the absolute scores remain low enough to justify the conclusion that metadata alone is insufficient.

The results support three design implications. First, accessible-label evaluation should separate role correctness from functional intent. RoleOnly is a strong role baseline and a weak accessibility labeler. Second, short generated labels need metrics that are interpretable at the token level; full-sentence caption metrics alone can obscure whether the action word was recovered. Third, LLM/VLM systems should be evaluated against deterministic baselines and an oracle. If a model improves only token F1 but not intent F1, it may be producing fluent generic labels rather than useful accessible names. If a model surpasses the oracle reuse bound, it is evidence that it synthesizes new functional labels instead of memorizing common training phrases.

For information architecture, the findings imply that accessible names should be treated as part of the product vocabulary rather than as an afterthought. Many labels in the dataset are short domain concepts: “share button,” “delete email button,” “products dropdown,” or “location searchbar.” A model that cannot infer the domain concept cannot provide a usable label even if it identifies the widget class. For inclusive design, the findings also imply that automatic repair should be conservative. A low-confidence label may be worse than a missing label if it directs the user to an incorrect action. Baseline metrics can therefore inform triage: high-confidence exact or near-exact outputs can be suggested automatically, while low-intent outputs should be routed to a developer or accessibility reviewer.

The per-class pattern is also informative for dataset design. Classes with visually compact labels, such as icons and switches, can look easier to a computer-vision system because the graphical glyph is salient, but they are difficult for metadata-only methods because the CSV exposes no pixels. Conversely, text inputs can have repeated lexical conventions: searchbars, login fields, e-mail fields, and location fields. The best deployable model benefits from those conventions even without OCR. This distinction supports a multimodal evaluation protocol in which class metadata, view hierarchy, OCR text, cropped component pixels, and full-screen context are measured separately before being combined.

The density results show why aggregate scores should not be reported alone. High-density records have the best LogReg intent F1 in this run, but density is not an inherent advantage. It can reflect repeated applications, recurring toolbar structures, or multiple labelled elements from the same software family. A future evaluation should therefore pair aggregate means with stratified results. The present paper includes per-class, per-density, per-depth, and error-example tables so that reviewers can inspect whether a model improves evenly or merely benefits from a subset of repeated contexts.

The strongest practical lesson concerns metric selection. In the table, RoleOnly is a competitive model under full token F1 because every human label normally contains a role word or a class-like word. If a manuscript reported only token F1, the conclusion might be that role-only labelling is adequate. Intent F1 reverses that interpretation. It shows that the user-facing action term is almost always absent. This is exactly the kind of issue that accessibility evaluation must

surface: conformance cannot be reduced to saying that an object is a button, because the user needs to know which button to activate.

The stored predictions also enable qualitative review. Reviewers can inspect outputs/predictions.csv and see that many errors are plausible strings in isolation but wrong in context. For example, “log in button” is a valid accessible name for a login control, but it is wrong for “themes category” or “page number input.” The goal is therefore not simply to produce natural-sounding English. The goal is grounded semantic naming. This is where image and layout evidence from the screenshot/SOM archive should help future models: the visual neighborhood and surrounding text can disambiguate controls that have identical classes and similar coordinates.

The paper therefore provides a coherent empirical baseline for future inclusive-design work. The baseline is not intended to replace human review. It is intended to make review sharper: researchers can inspect whether a new model improves exact match, intent F1, label-in-name80, and per-class behavior while preserving role agreement. This is aligned with the broader UI-understanding literature, where screen parsing, screen recognition, widget captioning, and UI embeddings all demonstrate that interface meaning is multimodal and contextual [6]–[13].

Table 7. Overall model comparison under screenshot-level 10-fold validation.

Model	Exact	Token F1	Intent F1	Jaccard	Edit sim	Role agree	Label-in-name80
OracleReuse	0.166	0.559	0.170	0.432	0.409	1.000	0.166
LogReg	0.036	0.392	0.060	0.261	0.461	1.000	0.041
ClassMajority	0.021	0.408	0.054	0.269	0.471	1.000	0.027
TFIDF_1NN	0.021	0.375	0.045	0.244	0.452	1.000	0.029
TFIDF_5NN	0.018	0.376	0.043	0.243	0.450	1.000	0.025
Hybrid	0.018	0.371	0.041	0.239	0.443	1.000	0.025
LocationRole	0.004	0.411	0.026	0.269	0.416	1.000	0.007
RoleOnly	0.000	0.481	0.009	0.328	0.305	1.000	0.000

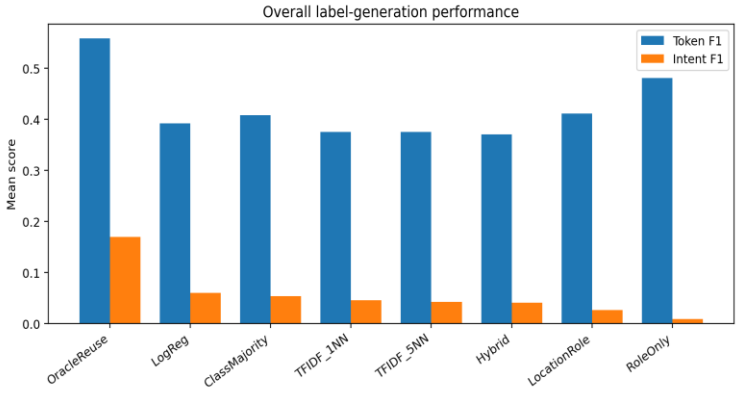

Figure 4. Token F1 and intent F1 by generator.

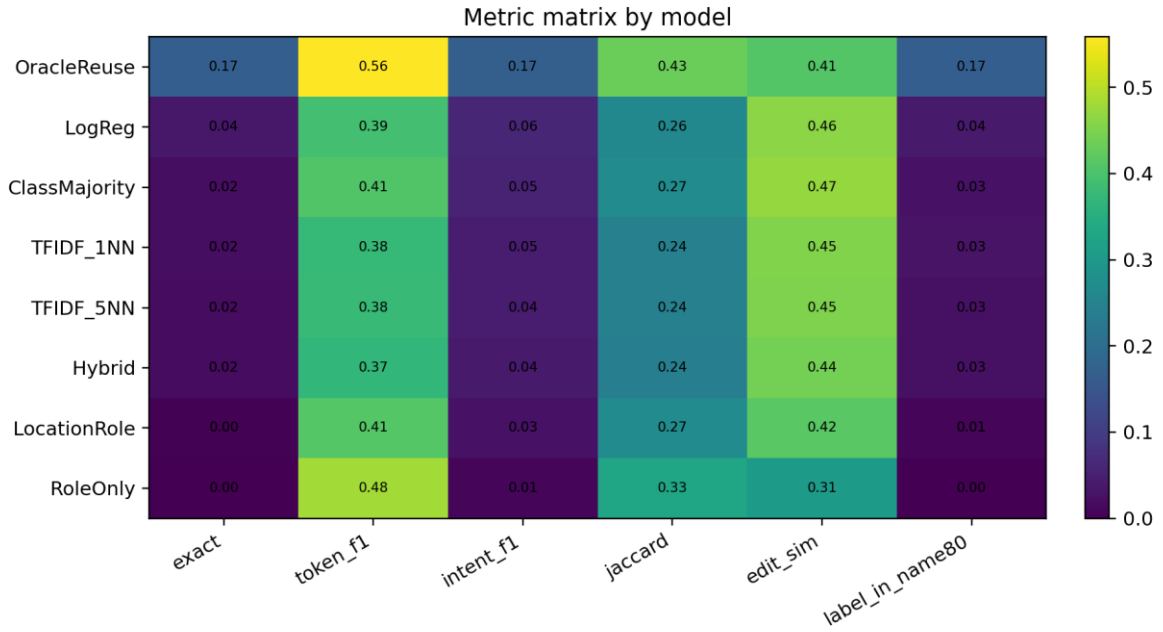

Figure 5. Metric matrix by model, showing the gap between role agreement and functional correctness.

Table 8. Bootstrap 95% confidence intervals for intent-F1 difference against RoleOnly.

Model	**Delta**	**CI low**	**CI high**
RoleOnly	0.000	0.000	0.000
ClassMajority	0.045	0.028	0.062
LocationRole	0.017	0.005	0.030
TFIDF_1NN	0.036	0.019	0.053
TFIDF_5NN	0.034	0.018	0.050
LogReg	0.051	0.032	0.072
Hybrid	0.032	0.016	0.048
OracleReuse	0.161	0.132	0.193

Table 9. Logistic-regression performance by component class.

Class	**n**	**Exact**	**Token F1**	**Intent F1**	**Role agree**
Switch	36	0.222	0.511	0.236	1.000
Text Input	79	0.101	0.474	0.207	1.000
Checkbox	65	0.000	0.316	0.043	1.000
Button	90	0.033	0.323	0.033	1.000
Link	48	0.000	0.279	0.029	1.000

Icon	90	0.011	0.462	0.011	1.000
Dropdown	90	0.000	0.354	0.007	1.000
Radio Button	61	0.000	0.444	0.000	1.000

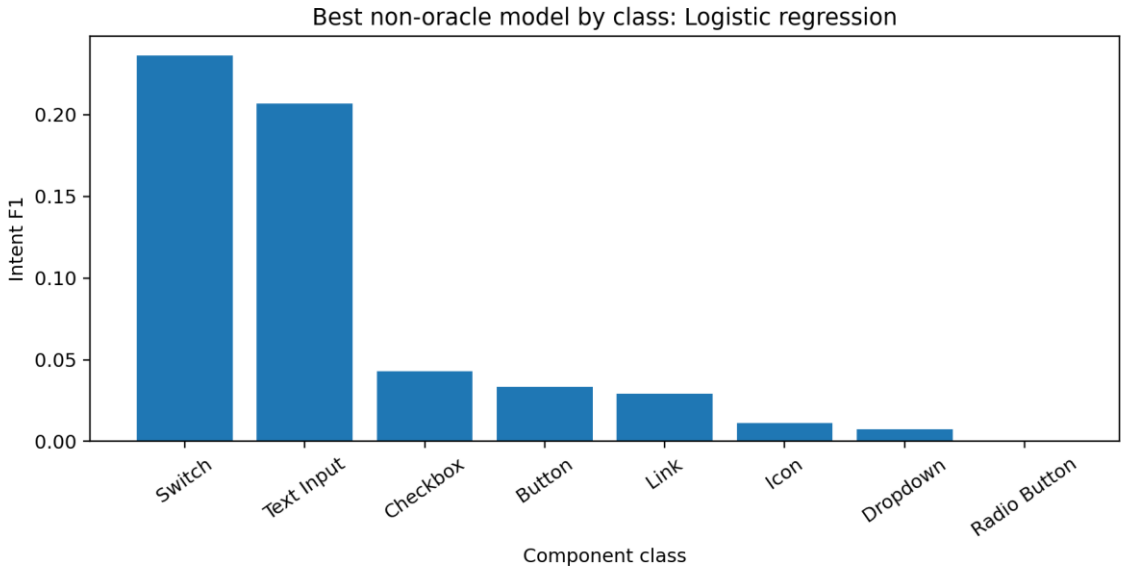

Figure 6. Intent F1 of the best deployable model by component class.

Table 10. Logistic-regression performance by density and SOM depth.

Group	**n**	**Exact**	**Token F1**	**Intent F1**	**Label-in-name80**
High Density	205	0.063	0.428	0.096	0.068
Low Density	154	0.032	0.404	0.047	0.039
Medium Density	200	0.010	0.347	0.034	0.015
-- depth --					
Depth 1	9	0.333	0.587	0.333	
Depth 2	176	0.040	0.354	0.056	
Depth 3	341	0.029	0.408	0.057	
Depth 4	33	0.000	0.381	0.042	

Table 11. Representative low-intent-F1 errors for Logistic Regression.

Row	**Class**	**Human label**	**Generated label**	**Token F1**	**Intent F1**

136	Button	themes category	web login button	0.000	0.000
109	Button	apply theme input	log in button	0.000	0.000
158	Text Input	reddit searchbar	connection encoding input	0.000	0.000
293	Dropdown	free tools highlight	translate from dropdown	0.000	0.000
292	Dropdown	date highlight	secure search dropdown	0.000	0.000
137	Button	search focus shortcut	log in button	0.000	0.000
139	Text Input	email inbox searchbar	connection encoding input	0.000	0.000
46	Text Input	location searchbar	search input	0.000	0.000
108	Button	ant desing wizard icon	use custom theme button	0.000	0.000
75	Button	page number input	log in button	0.000	0.000

Limitations

The first limitation is modality. The reported experiments use the actual semantic labels.csv records and package metadata, but the deployable generators evaluated here do not parse screenshot pixels or SOM JSON contents. Therefore, the conclusions are strongest for metadata-only and retrieval-based label generation. They do not measure the full potential of a screenshot-aware VLM. This limitation is deliberate in the reproducibility package: every reported output can be regenerated locally without proprietary API calls, model downloads, or non-deterministic prompting. The limitation bounds the interpretation without changing the empirical status of the results.

The second limitation is metric design. Intent F1 is useful because it exposes the failure of role-only labels, but it remains a lexical metric. It does not fully credit paraphrases such as “trash” versus “delete,” and it does not judge whether a generated label is concise, polite, or safe for a particular assistive technology. Human evaluation and task-based usability studies remain necessary for deployment. BLEU, ROUGE, and METEOR have similar known limits in generated text evaluation, especially when references are short [22]–[24].

The third limitation is dataset scale. The experiment covers all 559 labelled events available in the CSV, but many labels are unique. This makes exact match hard and makes per-class estimates noisy for smaller classes such as Switch and Link. The source package contains 100 screenshots, yet only 82 unique screenshot filenames appear in the labelled CSV. The paper treats the event-level CSV as the evaluation ground truth, which is the correct unit for the reported label-generation task, but future work should also analyze the unlabelled screenshots and SOM files when image-aware models are tested.

The study also does not substitute for direct user testing. The human labels in the CSV are used as ground truth, but real accessibility quality depends on how people using screen readers, switch access, or speech input understand and act on a label during a task. A label that is semantically correct may be too long, too terse, or inconsistent with visible text. The evaluation therefore measures semantic agreement with the provided human reference labels, not end-user success time or satisfaction.

The fourth limitation is that OracleReuse is an upper bound rather than an achievable model. It is included to show the maximum benefit of training-label reuse under the split. It should not be compared with deployable systems as if it were a practical generator. A true VLM or LLM with screen evidence may exceed the oracle because it can synthesize new labels; conversely, it may fall below the oracle if it produces fluent but semantically vague names.

Conclusion

This paper conducted a full reproducible evaluation of generated accessible labels on the specified UI Component Semantic Description Dataset. The experiment used all 559 labelled CSV events, screenshot-level 10-fold group validation, deterministic label generators, stored predictions, and regenerated tables and figures. The strongest deployable model, logistic regression, achieved exact match 0.036, token F1 0.392, and intent F1 0.060. The role-only baseline achieved token F1 0.481 but intent F1 0.009, demonstrating that generic role overlap is not a reliable proxy for accessible-label quality. The oracle upper bound reached exact match 0.166 and token F1 0.559, indicating that label reuse alone cannot solve the task.

The empirical conclusion is definite: class, density, depth, and coordinate metadata preserve component role but do not recover enough functional intent. Future LLM/VLM systems for accessible UI semantics should therefore be evaluated with intent-sensitive metrics, grouped splits, per-class analysis, and comparison against transparent baselines. A useful generated accessible label must say not only what kind of control exists, but what the control does. That distinction is the difference between a formally present role and an inclusive interface.

References

[1] W3C, “Web Content Accessibility Guidelines (WCAG) 2.1,” W3C Recommendation, 2018.

[2] W3C, “Accessible Name and Description Computation 1.1,” W3C Recommendation, Dec. 2018.

[3] W3C, “WAI-ARIA Authoring Practices 1.1,” W3C Working Group Note, Dec. 2017.

[4] B. Deka, Z. Huang, C. Franzen, J. Hibschman, D. Afergan, Y. Li, J. Nichols, and R. Kumar, “Rico: A Mobile App Dataset for Building Data-Driven Design Applications,” in Proc. 30th Annual ACM Symposium on User Interface Software and Technology (UIST), 2017, pp. 845–854.

[5] H. Xu, Y. Chen, and A. Med, “Automatic Detection and Explanation of Dark Patterns from Interface Microcopy: Empirical Comparison of BERT-Style Encoders, RoBERTa-Style Encoders, and LLM-Style Decoders on the ec-darkpattern Dataset,” Journal of Technology Informatics and Engineering, vol. 4, no. 3, pp. 590–612, 2025, doi: 10.51903/jtie.v4i3.491.

[6] Y. Li, G. Li, L. He, J. Zheng, H. Li, and Z. Guan, “Widget Captioning: Generating Natural Language Description for Mobile User Interface Elements,” in Proc. Conference on Empirical Methods in Natural Language Processing (EMNLP), 2020, pp. 5495–5510.

[7] Z. He, S. Sunkara, X. Zang, Y. Xu, L. Liu, N. Wichers, G. Schubiner, R. Lee, J. Chen, and B. Agüera y Arcas, “ActionBert: Leveraging User Actions for Semantic Understanding of User Interfaces,” in Proc. AAAI Conference on Artificial Intelligence, 2021.

[8] C. Bai, X. Zang, Y. Xu, S. Sunkara, A. Rastogi, J. Chen, and B. Agüera y Arcas, “UIBert: Learning Generic Multimodal Representations for UI Understanding,” in Proc. International Joint Conference on Artificial Intelligence (IJCAI), 2021.

[9] T. J.-J. Li, L. Popowski, T. M. Mitchell, and B. A. Myers, “Screen2Vec: Semantic Embedding of GUI Screens and GUI Components,” in Proc. CHI Conference on Human Factors in Computing Systems, 2021.

[10] X. Zhang, L. de Greef, A. Swearngin, S. White, K. Murray, L. Yu, Q. Shan, J. Nichols, J. Wu, C. Fleizach, A. Everitt, and J. P. Bigham, “Screen Recognition: Creating Accessibility Metadata for Mobile Applications from Pixels,” in Proc. CHI Conference on Human Factors in Computing Systems, 2021.

[11] J. Wu, X. Zhang, J. Nichols, and J. P. Bigham, “Screen Parsing: Towards Reverse Engineering of UI Models from Screenshots,” in Proc. ACM Symposium on User Interface Software and Technology (UIST), 2021.

[12] S. Bunian, K. Li, C. Jemmali, C. Harteveld, Y. Fu, and M. S. El-Nasr, “VINS: Visual Search for Mobile User Interface Design,” in Proc. CHI Conference on Human Factors in Computing Systems, 2021.

[13] X. Zang, Y. Xu, and J. Chen, “Multimodal Icon Annotation for Mobile Applications,” in Proc. 23rd International Conference on Mobile Human-Computer Interaction (MobileHCI), 2021.

[14] A. S. Ross, X. Zhang, J. Fogarty, and J. O. Wobbrock, “An Epidemiology-Inspired Large-Scale Analysis of Android App Accessibility,” ACM Transactions on Accessible Computing, vol. 13, no. 1, 2020.

[15] F. Mehralian, N. Salehnamadi, and S. Malek, “Data-Driven Accessibility Repair Revisited: On the Effectiveness of Generating Labels for Icons in Android Apps,” in Proc. ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE), 2021.

[16] Y. Li, J. He, X. Zhou, Y. Zhang, and J. Baldridge, “Mapping Natural Language Instructions to Mobile UI Action Sequences,” in Proc. 58th Annual Meeting of the Association for Computational Linguistics (ACL), 2020, pp. 8198–8210.

[17] T. F. Liu, M. Craft, J. Situ, E. Yumer, R. Mech, and R. Kumar, “Learning Design Semantics for Mobile Apps,” in Proc. ACM Symposium on User Interface Software and Technology (UIST), 2018, pp. 569–579.

[18] A. Vaswani et al., “Attention Is All You Need,” in Advances in Neural Information Processing Systems, 2017, pp. 5998–6008.

[19] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding,” in Proc. North American Chapter of the Association for Computational Linguistics (NAACL), 2019, pp. 4171–4186.

[20] T. B. Brown et al., “Language Models are Few-Shot Learners,” in Advances in Neural Information Processing Systems, vol. 33, 2020, pp. 1877–1901.

[21] Jason Kuhn, Yushan Chen, and Evelyn Chan, “AI-Driven Mobile UI Pattern Recognition and Design Topic Mining on RICO: Semantic Clustering and Screenshot-Based Topic Classification”, JACS, vol. 4, no. 5, pp. 67–83, May 2024, doi: 10.69987/JACS.2024.40506.

[22] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, “BLEU: A Method for Automatic Evaluation of Machine Translation,” in Proc. 40th Annual Meeting of the Association for Computational Linguistics (ACL), 2002, pp. 311–318.

[23] C.-Y. Lin, “ROUGE: A Package for Automatic Evaluation of Summaries,” in Proc. ACL Workshop on Text Summarization Branches Out, 2004, pp. 74–81.

[24] A. Lavie and A. Agarwal, “METEOR: An Automatic Metric for MT Evaluation with High Levels of Correlation with Human Judgments,” in Proc. Second Workshop on Statistical Machine Translation, 2007, pp. 228–231.

[25] F. Pedregosa et al., “Scikit-Learn: Machine Learning in Python,” Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.

[26] J. Bai, H. Wang, Q. Wu, and B. Zhang, “Privacy-robust incrementality estimation in cookieless settings via uplift modeling: Reproducible evidence from the Hillstrom e-mail experiment,” Journal of Technology Informatics and Engineering, vol. 5, no. 1, Apr. 2026, doi: 10.51903/jtie.v5i1.468.

[27] Xiaofei Luo, “WikiPath: Explainable Wikipedia-Grounded Dialogue via Explicit Knowledge Selection and Entity-Path Planning”, JACS, vol. 6, no. 1, pp. 99–115, Jan. 2026, doi: 10.69987/JACS.2026.60107.

[28] M.-J. Kuo, D. Zheng, and J. Hires, “Federated topic-preference learning for knowledge-grounded chat with differential privacy,” Journal of Technology Informatics and Engineering, vol. 4, no. 2, Aug. 2025, doi: 10.51903/jtie.v4i2.502.

[29] S. Zhao, J. Bai, and D. Roberson, “Multi-horizon GPU demand forecasting with workload semantics and operational risk curves: An empirical study on Alibaba Clusterdata GPU trace,” Journal of Technology Informatics and Engineering, vol. 4, no. 3, Dec. 2025, doi: 10.51903/jtie.v4i3.498.

[30] G. Mi, T. Ye, and D. Wood, “A lightweight medical foundation model for cross-modal multi-task pretraining and parameter-efficient few-shot transfer on MedMNIST,” Journal of Technology Informatics and Engineering, vol. 4, no. 3, Dec. 2025, doi: 10.51903/jtie.v4i3.492.

[31] J. Mu, T. Ye, and P. Patel, “Offline counterfactual evaluation for advertising and recommendation slot policies: A reproducible study on the open bandit dataset (small),” Journal of Technology Informatics and Engineering, vol. 4, no. 3, Dec. 2025, doi: 10.51903/jtie.v4i3.500.

[32] Q. Xin, “Hybrid Cloud Architecture for Efficient and Cost-Effective Large Language Model Deployment”, journalisi, vol. 7, no. 3, pp. 2182-2195, Sep. 2025.

[33] Q. Xin, “Uncertainty-aware late fusion for 3D perception (confidence calibration + fusion rule learning),” Journal of Technology Informatics and Engineering, vol. 4, no. 1, Apr. 2025, doi: 10.51903/jtie.v4i1.485.

[34] L. Zhang, R. Ma, and P. Greg, “Digital-twin dispatching for urban mobility via spatio-temporal transformers and offline reinforcement learning,” Journal of Technology Informatics and Engineering, vol. 4, no. 2, Aug. 2025, doi: 10.51903/jtie.v4i2.501.

[35] Yuanzheng Chen, Yitian Zhang, and Matt Sherman, “Going Concern and Bankruptcy Prediction under Extreme Class Imbalance: Cost-Sensitive Learning, Resampling, and Focal Loss with Explainable Financial-Ratio Portraits”, JACS, vol. 4, no. 4, pp. 80–96, Apr. 2024, doi: 10.69987/JACS.2024.40407.

[36] Yunhe Li, “Test-in-the-loop LLM Repair: Verifiable Automated Program Repair on QuixBugs with a ‘Failing Test → Patch → Regression Test’ Loop”, JACS, vol. 4, no. 2, pp. 62–75, Feb. 2024, doi: 10.69987/JACS.2024.40206.

[37] Daren Zheng, Boning Zhang, and Julie Geibel, “VerifySafe: Toxicity-Safe Agent Responses under Adversarial Prompts with Evidence-Based Self-Verification”, JACS, vol. 4, no. 1, pp. 67–82, Jan. 2024, doi: 10.69987/JACS.2024.40106.

[38] Jing Chen, Xinzhuo Sun, Qiyou Wu, and Matt Jackson, “Risk-Calibrated Biomedical Search: Calibrated Selection of LLM-Style Query Expansions on BEIR TREC-COVID”, JACS, vol. 4, no. 4, pp. 61–79, Apr. 2024, doi: 10.69987/JACS.2024.40406.

[39] Yifei Lu, Jinyi Mu, and Thao Tran, “Uncertainty-Aware Uplift Modeling for Safer Marketing Targeting: Conformal Prediction and Bayesian Calibration with LCB Policies”, JACS, vol. 4, no. 5, pp. 84–101, May 2024, doi: 10.69987/JACS.2024.40507.

[40] Siming Zhao, Haozhe Wang, and Neil Davison, “Profit-Maximizing Cost-Sensitive Credit Scoring with LLM-Extracted Policy Constraints”, JACS, vol. 4, no. 3, pp. 91–108, Mar. 2024, doi: 10.69987/JACS.2024.40307.

[41] Daren Zheng and Chenyu Li, “Behavior-Level Jailbreak Resistance via Multi-Stage Refusal + Utility Preservation”, JACS, vol. 4, no. 1, pp. 83–99, Jan. 2024, doi: 10.69987/JACS.2024.40107.

[42] Yunhe Li, “Findable then Explainable: Retrieval–Summary Integration for Code Intelligence on a Lightweight CodeSearchNet Subset”, JACS, vol. 4, no. 7, pp. 65–82, Jul. 2024, doi: 10.69987/JACS.2024.40706.

[43] Xinzhuo Sun, Yifei Lu, and Jing Chen, “Controllable Long-Term User Memory for Multi-Session Dialogue: Confidence-Gated Writing, Time-Aware Retrieval-Augmented Generation, and Update/Forgetting”, JACS, vol. 3, no. 8, pp. 9–24, Aug. 2023, doi: 10.69987/JACS.2023.30802.

[44] Hanqi Zhang, “DriftGuard: Multi-Signal Drift Early Warning and Safe Re-Training/Rollback for CTR/CVR Models”, JACS, vol. 3, no. 7, pp. 24–40, Jul. 2023, doi: 10.69987/JACS.2023.30703.

[45] Meng-Ju Kuo, Boning Zhang, and Haozhe Wang, “Tokenized Flow-Statistics Encrypted Traffic Analysis: Comparative Evaluation of 1D-CNN, BiLSTM, and Transformer on ISCX VPN-nonVPN 2016 (A1+A2, 60 s)”, JACS, vol. 3, no. 8, pp. 39–53, Aug. 2023, doi: 10.69987/JACS.2023.30804.

[46] Z. Zhong, M. Zheng, H. Mai, J. Zhao, and X. Liu, “Cancer image classification based on DenseNet model,” Journal of Physics: Conference Series, vol. 1651, no. 1, p. 012143, 2020.

[47] Jubin Zhang, “Interpretable Skill Prioritization for Volleyball Education via Team-Stat Modeling”, JACS, vol. 3, no. 3, pp. 34–49, Mar. 2023, doi: 10.69987/JACS.2023.30304.

[48] Jinyi Mu, Yifei Lu, and Michelle Smith, “LLM-Assisted Incrementality (Uplift) Modeling for Email Advertising: From Feature Interactions to Interpretable Audience–Creative–Channel Policies ”, JACS, vol. 3, no. 1, pp. 31–48, Jan. 2023, doi: 10.69987/JACS.2023.30103.

[49] Siming Zhao, Hailin Zhou, and Daniel Martinez, “LLM-Assisted Causal Attribution of Service Performance Upgrades on Churn and Tenure: Full Evaluation on the IBM Telco Customer Churn Dataset”, JACS, vol. 3, no. 2, pp. 18–34, Feb. 2023, doi: 10.69987/JACS.2023.30202.

[50] Daren Zheng, Chenyu Li, and Harvey Davidson, “Continual Red-Teaming for In-the-Wild Jailbreaks via Online Guardrail Updates and Guardrail Distillation”, JACS, vol. 3, no. 2, pp. 35–49, Feb. 2023, doi: 10.69987/JACS.2023.30203.

[51] Binghua Zhou, Siming Zhao, and David Chao, “LLM-Guided Energy-Aware A/B Testing for Consolidation and DVFS Policies via Power-Sensitivity Clustering”, JACS, vol. 3, no. 4, pp. 12–30, Apr. 2023, doi: 10.69987/JACS.2023.30402.

[52] Jing Chen, Xinzhuo Sun, and Vincent Brown, “Claim-Aware Scientific RAG: Evidence-First Retrieval and Abstention for Scientific Fact Responses on SciFact”, JACS, vol. 3, no. 1, pp. 16–30, Jan. 2023, doi: 10.69987/JACS.2023.30102.

[53] Yunhe Li, “Execution-Feedback and Retrieval-Augmented Generation for Conversational Text-to-SQL: From One-Shot Questions to Clarification-Driven Executable Dialogs”, JACS, vol. 3, no. 2, pp. 1–17, Feb. 2023, doi: 10.69987/JACS.2023.30201.

[54] Xinzhuo Sun, Jing Chen, Binghua Zhou, and Meng-Ju Kuo, “ConRAG: Contradiction-Aware Retrieval-Augmented Generation under Multi-Source Conflicting Evidence”, JACS, vol. 4, no. 7, pp. 50–64, Jul. 2024, doi: 10.69987/JACS.2024.40705.

[55] Hanqi Zhang, “Risk-Aware Budget-Constrained Auto-Bidding under First-Price RTB: A Distributional Constrained Deep Reinforcement Learning Framework”, JACS, vol. 4, no. 6, pp. 30–47, Jun. 2024, doi: 10.69987/JACS.2024.40603.

[56] Z. S. Zhong and S. Ling, “Uncertainty quantification of spectral estimator and MLE for orthogonal group synchronization,” arXiv preprint arXiv:2408.05944, 2024.

[57] Z. S. Zhong and S. Ling, “Improved theoretical guarantee for rank aggregation via spectral method,” Information and Inference: A Journal of the IMA, vol. 13, no. 3, 2024.

[58] Jubin Zhang, “Tactical Language + AI Tutoring from Structured Volleyball Rally Logs: Reproducible Experiments on NCAA Play-by-Play”, JACS, vol. 4, no. 1, pp. 58–66, Jan. 2024, doi: 10.69987/JACS.2024.40105.

[59] Xiaofei Luo, “Semantic Verifier for Post-hoc Answer Validation in Chat Platforms: Claim Decomposition, Evidence Retrieval, NLI, and Traceable Citations”, JACS, vol. 4, no. 3, pp. 74–90, Mar. 2024, doi: 10.69987/JACS.2024.40306.

[60] Yunhe Li. (2023). Risk-Sensitive Offline Reinforcement Learning for Stable ABR QoE Improvements on Real HSDPA and LTE Traces. Journal of Advanced Computing Systems , 3(4), 1-11. https://doi.org/10.69987/JACS.2023.30401

[61] Q. Xin, “Probabilistic bike-sharing demand forecasting under changing weather and seasonal regimes with transformer-based models,” Transport Findings, Mar. 2026, doi: 10.32866/001c.157499.