Journal of Technology Informatics and Engineering (JTIE)
Vol. 4 No. 3 December 2025
E-ISSN: 2961-9068; P-ISSN: 2961-8215, Pages 590-612
DOI: https://doi.org/10.51903/jtie. v4i3.491

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

Haosen Xu[1], Yushan Chen*[2], Aron Med[3]
Email: yushanchen1029@gmail.com (2)
[1]*Electrical Engineering and Computer Science, University of California, Berkeley, CA, USA*
[2]*Service Design, Savannah College of Art and Design, GA, USA*
[3]*California Institute of the Arts, CA, USA*
*Corresponding Author

Abstract
Dark patterns (also called deceptive design patterns) are interface choices that steer or pressure users into unintended actions such as rushed purchases, unnecessary disclosures, or hard-to-cancel subscriptions. In e-commerce, many dark patterns are expressed directly in microcopy (e.g., button labels, banners, and inline messages), which makes text-only detection attractive for scalable auditing. This paper presents a fully reproducible experimental study on ec-darkpattern, a text-based dataset of e-commerce interface strings with balanced binary labels (1,178 dark pattern vs. 1,178 non-dark pattern) and seven dark pattern categories. We compare (i) a rule-based lexicon baseline, (ii) hashed n-gram linear models, (iii) a lightweight BERT-style bidirectional transformer encoder with word tokenization, (iv) a lightweight RoBERTa-style bidirectional transformer encoder with character tokenization, and (v) an LLM-style causal decoder trained as a classifier on the same inputs. On a fixed 80/10/10 split with seed 42, the best-performing model is a hashing + linear SVM baseline (F1=0.9437, ROC-AUC=0.9810), while the BERT-style encoder achieves F1=0.9038 (ROC-AUC=0.9695), the RoBERTa-style encoder achieves F1=0.8907 (ROC-AUC=0.9573), and the LLM-style decoder achieves F1=0.7884 (ROC-AUC=0.8808). These results should be interpreted as a controlled comparison under low-resource, no-pretraining conditions on a single fixed split, rather than as a general ranking of encoder-style versus decoder-style transformers. To support explainability, we generate token-level attributions using gradient-based saliency, summarize them as key phrases, and estimate explanation consistency via top-k token overlap on an exploratory 20-instance sample (mean Jaccard up to 0.7482 between the two character-based transformers). Finally, we curate an error-case library that links misclassifications to their most influential phrases. Within this short-microcopy setting, the findings show that lexical baselines are especially strong, while transformer directionality and tokenization change both accuracy and the stability of highlighted cues.

Keywords: *Dark Patterns, Deceptive Design, Interface Microcopy, E-Commerce, Text Classification.*

I. INTRODUCTION

Dark patterns are user interface (UI) designs that manipulate, coerce, or deceive people into actions that primarily benefit the service provider (Ashofi, 2023; Heraditya et al., 2026; Santoso & Yan, 2024; Wibowo & Santoso, 2024). The term was coined to name and document deceptive interface tactics in everyday products, and it has since become a central concept in HCI ethics, consumer protection, and privacy regulation (Brignull, 2010; Gray et al., 2018; Mathur et al., 2019). Unlike usability defects, dark patterns are often effective from the provider perspective:

Received on October, 2025; Revised on November, 2025; Accepted on December, 2025; Published on December, 2025

Journal of Technology Informatics and Engineering (JTIE)
Vol. 4 No. 3 December 2025
E-ISSN: 2961-9068; P-ISSN: 2961-8215, Pages 590-612
Journal Link: https://jtie.stekom.ac.id/index.php/jtie/index

they exploit cognitive biases, information asymmetries, and choice architecture to shift decisions such as purchases, subscriptions, or consent settings (Nouwens et al., 2020).

E-commerce microcopy is a particularly concentrated carrier of these tactics. Short snippets such as "Only 3 left," "Sale ends in 02:00," or "No thanks, I don't like saving money" can encode scarcity, urgency, social proof, or confirmshaming in a few tokens. Because microcopy is easy to scrape and log at scale, text-based detection has immediate practical value: it enables auditing, monitoring, and triage of potentially deceptive experiences without requiring full visual rendering or interaction replay. At the same time, microcopy is ambiguous. A phrase like "Out of stock" can be an honest status update, while "Low stock" can be a persuasive cue; similarly, star ratings can be truthful evidence or misleading social pressure depending on context. This ambiguity makes automatic detection a nontrivial NLP problem.

Prior research established the prevalence and diversity of dark patterns on the web. Large-scale measurement studies identified thousands of instances of dark patterns on shopping sites and proposed taxonomies that cover multiple pattern types and categories (Mathur et al., 2019). Complementary work analyzed dark patterns in mobile apps and consent interfaces, demonstrating that subtle UI choices measurably influence user behavior (Di Geronimo et al., 2020; Nouwens et al., 2020). These studies motivate automated detection pipelines that can support regulators, researchers, and practitioners. However, many detection systems are either multimodal (requiring screenshots or DOM structures) or are designed for specific contexts such as cookie banners (Soe et al., 2022). Text-only detection remains attractive because it is light-weight and portable across many e-commerce surfaces.

A second requirement for practical auditing is explainability. Even when a classifier achieves high F1, auditors need to understand why a string was flagged: which phrase expressed urgency, which wording induced guilt, and which token drove the prediction. Post-hoc explanation methods such as LIME and SHAP, as well as gradient-based attribution methods, can highlight influential features in text classification models (Ribeiro et al., 2016; Lundberg & Lee, 2017; Sundararajan et al., 2017). In the context of dark patterns, explanations are particularly important because the goal is often not only to detect, but also to support remediation: designers need actionable cues to rewrite microcopy into a more neutral form.

This paper focuses on automatic recognition and explanation of dark patterns using only interface microcopy and button text. We conduct full experimental evaluations on the ec-darkpattern dataset, which provides labeled e-commerce interface strings with both binary labels and a dark pattern category for positive instances (Yada et al., 2022). We compare three modeling paradigms that are commonly discussed in contemporary NLP practice: (1) feature-based linear models that

operate on surface n-grams, (2) transformer encoders that compute bidirectional contextual representations (BERT-style and RoBERTa-style), and (3) transformer decoders that operate with causal attention and are often associated with large language models (LLM-style) (Vaswani et al., 2017; Devlin et al., 2019; Liu et al., 2019). To keep the study fully reproducible within a single experimental pipeline, we train lightweight encoder and decoder transformers from scratch on ec-darkpattern, and we report all hyperparameters and random seeds. Accordingly, the comparison isolates architectural behavior under low-resource, no-pretraining conditions rather than comparing the fully pre-trained encoder and decoder families commonly used in practice.

Our study makes three concrete contributions. First, we provide a controlled single-split comparison of encoder- and decoder-transformer classifiers under low-resource, no-pretraining conditions, with strong classical baselines as reference points. Second, we implement token-level explanations using gradient-based attributions and report both qualitative highlights and an exploratory explanation-consistency metric based on top-k token overlap across models. Third, we curate an error case library that links high-confidence false positives and false negatives to their most influential phrases, enabling targeted data collection and guideline updates.

The remainder of the paper is organized as follows. The literature review synthesizes prior work on dark-pattern taxonomies and detection, as well as explainability methods in NLP. The methods section describes the ec-darkpattern dataset, preprocessing, models, training protocol, and evaluation metrics. The results section reports accuracy and ROC-AUC across models, analyzes performance by dark pattern category, and evaluates the consistency of explanations. The conclusion summarizes the main findings and provides recommendations for deploying microcopy-based dark pattern detection in auditing workflows.

Regulators and advocacy organizations increasingly treat deceptive interface design as a compliance and consumer harm issue rather than a mere usability concern. For example, privacy and consumer protection discussions around consent and subscription cancellation frequently cite UI friction, hidden choices, and manipulative wording as mechanisms that undermine informed decision making. Empirical evidence from consent banner studies shows that removing or hiding opt-out options can increase acceptance rates by large margins, underscoring why microcopy and button wording should be treated as a measurable risk factor (Nouwens et al., 2020). In practice, organizations need tools that can prioritize reviews across large page inventories and A/B tests, and provide concrete language-level explanations to support remediation.

From an NLP perspective, interface microcopy differs from conventional text classification settings such as sentiment analysis or topic classification. Microcopy strings are short, highly templated, and often contain numerals, symbols, and fragments that convey persuasive cues (e.g.,

"3,081 people viewed" or "Ends in 00:59"). These properties favor sparse lexical features and pattern matching, but they also produce domain-specific paraphrases, spelling variants, and cross-site templates that can benefit from contextual modeling. This tension makes the task a useful testbed for comparing model families under realistic constraints: the data are large enough for training, but the text is short enough that simple models remain competitive (Yada et al., 2022).

The comparison between encoder-style and decoder-style transformers is also motivated by deployment trends. Encoder models such as BERT and RoBERTa are widely used for classification because they process the full input bidirectionally and produce robust sentence representations (Devlin et al., 2019; Liu et al., 2019). In the standard encoder formulation, a dedicated [CLS] token attends to all positions and is used as the sentence representation for classification. Decoder-only LLMs are increasingly used through prompting or fine-tuning, including for safety and policy classification tasks. In a causal decoder, self-attention is left-to-right, and the classifier typically reads the final token state, which summarizes the sequence through a directional accumulation of prefix representations rather than through a dedicated bidirectional summary token. This asymmetry can matter in short microcopy, where evidence is often distributed across a few tokens and may require combining prefix and suffix cues. By training lightweight encoder and decoder classifiers under matched conditions, we directly measure how attention direction and tokenization affect performance and explanations on ec-darkpattern.

II. LITERATURE REVIEW

Research on dark patterns spans HCI, behavioral economics, and consumer protection. Early practitioner work cataloged recurring manipulative tactics and established a shared vocabulary for documenting deceptive interface choices (Brignull, 2010). Academic work later framed dark patterns as an ethical phenomenon in which user value is displaced by provider value, and it argued for studying them as recurring patterns rather than isolated incidents (Gray et al., 2018). This framing aligns with broader discussions of choice architecture, where design can steer decisions without explicit coercion (Thaler & Sunstein, 2008).

Because this study focuses on text-only dark pattern detection, the most relevant prior work lies at the intersection of dark pattern measurement, e-commerce microcopy, short-text classification, transformer-based text classification, and explanation evaluation (Mathur et al., 2019; Yada et al., 2022; Devlin et al., 2019; Liu et al., 2019; Ribeiro et al., 2016; Sundararajan et al., 2017). We therefore keep the remainder of the review tightly focused on studies that address deceptive design, microcopy classification, model comparison for text classification, and methods for interpreting classifier decisions.

A key step toward scalable measurement was the taxonomy introduced in a large-scale crawl of shopping websites. Mathur et al. (2019) analyzed thousands of e-commerce pages and identified instances of dark patterns grouped into categories such as scarcity, urgency, social proof, misdirection, obstruction, sneaking, and forced action. The taxonomy shows that many patterns rely on short linguistic constructs (e.g., countdowns, "only X left" claims, guilt-inducing refusal options), which motivates microcopy-based detection. Follow-up work also analyzed what makes a design pattern "dark" and emphasized normative dimensions such as autonomy reduction and information manipulation (Mathur et al., 2021).

Dark patterns were also studied in privacy and consent interfaces. Nouwens et al. (2020) scraped consent pop-ups and demonstrated experimentally that interface design changes substantially influence acceptance rates. This evidence supports treating manipulative design as a measurable risk factor rather than a purely subjective judgment. Technical surveys and audits of consent banners further emphasize that automated detection is difficult when patterns depend on layout, defaults, or multi-step flows rather than text alone.

Mobile ecosystems provide additional contexts. Di Geronimo et al. (2020) analyzed dark patterns in popular mobile applications and reported that many apps contained multiple types of dark patterns. Although mobile and web implementations differ, the underlying strategies often share linguistic signatures (e.g., confirmshaming), supporting the relevance of text-based signals across platforms.

Datasets are central to progress in automatic detection. The ec-darkpattern dataset was constructed specifically for text-only dark pattern detection in e-commerce and pairs labeled interface strings with a category label for positive examples (Yada et al., 2022). Yada et al. (2022) also provided baseline evaluations, showing that the dataset supports strong performance and is suitable as a benchmark for microcopy-based auditing. Because the dataset focuses on microcopy, it enables research into scalable language-level detection, even though it cannot capture purely visual deception.

Classical short-text classification approaches often rely on sparse lexical features, such as n-grams, and linear classifiers (Joachims, 1998; Weinberger et al., 2009). These models are computationally efficient and competitive on templated text. Linear SVMs have a long history in text classification because they scale to high-dimensional, sparse feature spaces and provide robust decision boundaries (Joachims, 1998). Feature hashing offers an efficient alternative to explicit vocabularies while preserving a high-capacity representation (Weinberger et al., 2009). For microcopy, n-grams directly capture phrases such as "limited time" or "only left" that act as strong lexical signatures.

Transformer models replaced recurrence with self-attention and enabled contextual representation learning (Vaswani et al., 2017). Encoder models such as BERT are trained with bidirectional objectives and are widely fine-tuned for classification tasks (Devlin et al., 2019). RoBERTa refined this approach and improved performance by modifying pretraining and optimization choices (Liu et al., 2019). Decoder-only transformers use causal attention and are commonly used for generation; they are also used for classification via prompting or fine-tuning, but their causal constraint changes how information is integrated across tokens.

Explainable AI methods provide tools for making detection systems audit-ready. Model-agnostic approaches such as LIME fit local surrogate models to highlight influential features (Ribeiro et al., 2016). SHAP connects feature attribution to Shapley values and provides a unifying framework for additive explanations (Lundberg & Lee, 2017). Gradient-based methods compute attributions from model derivatives, and integrated gradients provides axioms and a practical path integral approximation for deep networks (Sundararajan et al., 2017). In NLP, explanations are typically computed per token and then summarized as phrases.

Explanation quality is not guaranteed by attention weights or by visually plausible highlights. Attention weights can diverge from feature importance, which motivates the use of gradient- or perturbation-based explanations for faithful attributions (Jain & Wallace, 2019). Quantitative analysis of explanation stability and robustness further emphasizes that explanation evaluation should accompany accuracy reporting (Samek et al., 2019). In the dark pattern domain, explanation tools are particularly useful when they help auditors identify which microcopy terms drive model decisions and support remediation efforts.

In summary, prior work establishes the prevalence and potential harm of dark patterns, provides datasets and taxonomies for detection, and motivates the development of explainable, scalable auditing tools. The present study extends these directions by providing a controlled comparison of sparse lexical baselines, encoder-style transformers, and an LLM-style decoder classifier on ec-darkpattern, together with quantitative explanations, consistency, and an error-case library grounded in model attributions.

III. RESEARCH METHOD

This section describes the dataset, task formulation, preprocessing, model families, training protocol, explainability procedure, and evaluation metrics. All experiments were executed with a fixed random seed (42) and a deterministic train/validation/test split. The implementation used Python 3.11, scikit-learn 1.4.2, and PyTorch 2.5.1 (CPU). Because the study prioritizes exact reproducibility using a single fixed split, the resulting estimates—especially category-level results for rare classes—should be interpreted with caution.

A. Dataset

We used ec-darkpattern, a text-only dataset for automatic dark pattern detection in e-commerce (Yada et al., 2022). Each instance contains a page identifier (page_id), an interface string (text), a binary label (label), and a pattern category for positive instances (Pattern Category). The dataset contains 2,356 instances with exactly balanced binary labels: 1,178 dark-pattern texts and 1,178 non-dark-pattern texts. Positive instances belong to seven categories: Scarcity, Social Proof, Urgency, Misdirection, Obstruction, Sneaking, and Forced Action. The category distribution is skewed, with Scarcity (418) and Social Proof (312) as the largest categories and Forced Action (4) as the smallest. Table 1 summarizes dataset statistics, and Table 2 reports the category counts (Figure 1).

Table 1. Descriptive Statistics of Ec-Darkpattern Microcopy Strings

Statistic	Value
Instances (N)	2356
Unique page_id	1248
Mean characters per instance	42.87
Median characters per instance	25
Max characters per instance	857
Mean word tokens per instance	8.98
Median word tokens per instance	6

Table 2. Category Distribution (binary Label is 0 for Not Dark Pattern, 1 Otherwise)

Category	Count	BinaryLabel
Not Dark Pattern	1178	0
Scarcity	418	1
Social Proof	312	1
Urgency	210	1
Misdirection	195	1
Obstruction	27	1
Sneaking	12	1
Forced Action	4	1

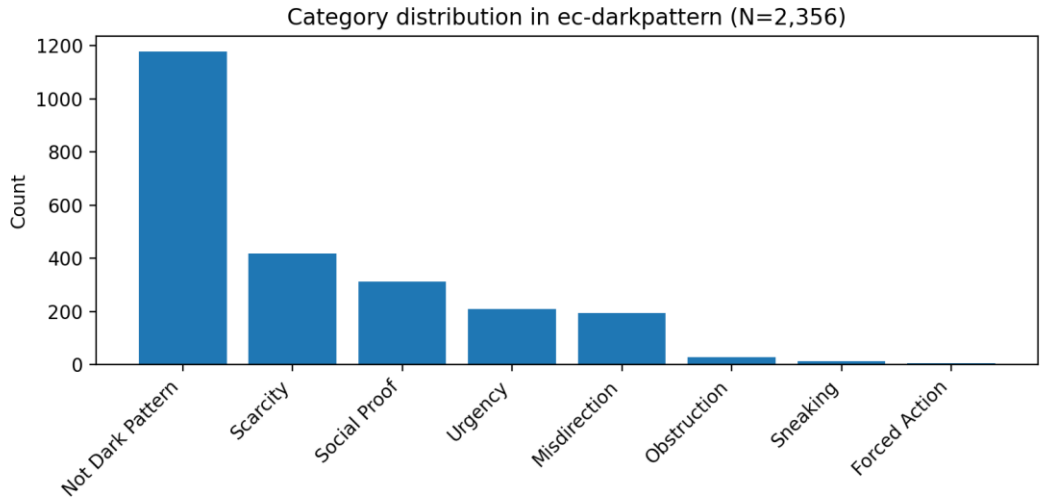

Figure 1. Category Distribution in Ec-Darkpattern (N=2,356)

Journal of Technology Informatics and Engineering (JTIE)
Vol. 4 No. 3 December 2025
E-ISSN: 2961-9068; P-ISSN: 2961-8215, Pages 590-612
Journal Link: https://jtie.stekom.ac.id/index.php/jtie/index

B. Task and Preprocessing

We formulated a binary classification: given an interface string x, predict y in {0,1} indicating whether x expresses a dark pattern. We used the Pattern Category label for category-level analysis of recall among positive examples. We applied minimal normalization: replacing line breaks with spaces, collapsing repeated whitespace, and trimming. We preserved punctuation and numerals because they encode persuasive cues (e.g., countdowns and quantities).

C. Data Split

We created a fixed 80/10/10 split (train/validation/test) stratified by the binary label. This produced 1,884 training instances, 236 validation instances, and 236 test instances, each with 50% positives. Rare categories were distributed unevenly across splits due to their small counts; Table 3 reports the category distribution per split. We chose a single fixed split to enable exact reproducibility and direct model-to-model comparison within one controlled pipeline. However, this design increases sensitivity to split composition, particularly for categories such as Forced Action. We therefore treat category-level recall estimates as descriptive for this split rather than as stable population estimates; repeated stratified splits or cross-validation would be a valuable extension. For neural models, we selected the checkpoint with the highest validation F1 and evaluated once on the held-out test set.

Table 3. Category Counts by Split (80/10/10; Stratified by Binary Label)

Category	Train	Val	Test	Total
Forced Action	1	1	2	4
Misdirection	162	21	12	195
Not Dark Pattern	942	118	118	1178
Obstruction	21	4	2	27
Scarcity	334	44	40	418
Sneaking	6	2	4	12
Social Proof	250	27	35	312
Urgency	168	19	23	210

Baseline 1: rule-based lexicon. We implemented a transparent rule baseline that flags strings matching a curated set of regular expressions for common cues in scarcity, urgency, social proof, misdirection/confirmshaming, and obstruction (e.g., "only [number] left", "limited time", "sale ends", "people viewed", and "no thanks"). If any rule matched, the model produced a dark pattern score of 0.9; otherwise it produced 0.1.

Baseline 2: hashed n-gram linear models. We used a hashing vectorizer with 1-2 word n-grams and 2^{18} feature dimensions (Weinberger et al., 2009). We trained two linear classifiers with SGD: a logistic-loss model (log_loss) and a hinge-loss model that approximates a linear SVM (Joachims, 1998). The logistic model outputs probabilities; the hinge model outputs a signed decision score that we used for ROC-AUC and thresholded at zero for F1.

D. Transformer Encoders

We implemented two lightweight bidirectional transformer encoders based on self-attention (Vaswani et al., 2017). Both used a [CLS] token and a linear classification head applied to the final [CLS] representation. The BERT-style encoder used word tokenization with a vocabulary learned from training data; the RoBERTa-style encoder used character tokenization to increase robustness to spelling and template variation (Devlin et al., 2019; Liu et al., 2019). Both encoders were trained from scratch on ec-darkpattern to keep the pipeline fully reproducible.

Table 4. Model Configurations and Training Settings Used in the Experiments

Model	Tokenization	Vocab/ Features	MaxLen	Architecture	Layers/Heads	d_model	FFdim	Dropout	Optimizer/LR	Training	Decision
Rule-Lexicon	Regex lexicon	-	-	Thresholded rule match	-	-	-	-	-	-	-
HV+SGD-LogReg	Hashing Vectorizer(1-2 grams)	2^18 features	-	Linear classifier	SGD log_loss	-	-	-	lr=default	max_iter=1000	thr=0.5
HV+SGD-SVM	Hashing Vectorizer(1-2 grams)	2^18 features	-	Linear classifier	SGD hinge	-	-	-	lr=default	max_iter=1000	thr=0.0 (decision)
BERT-style Encoder (word)	Word tokens (regex)	3404 vocab	48	Transformer encoder	layers=1; heads=4	d_model=64	ff=128	dropout=0.1	AdamW lr=2e-3	epochs=3; bs=64	thr=0.5
RoBERTa-style Encoder (char)	Character tokens	126 vocab	96	Transformer encoder	layers=1; heads=4	d_model=32	ff=64	dropout=0.1	AdamW lr=3e-3	epochs=3; bs=64	thr=0.5
LLM-style Decoder (char)	Character tokens	126 vocab	96	Causal Transformer decoder	layers=1; heads=4	d_model=32	ff=64	dropout=0.1	AdamW lr=3e-3	epochs=3; bs=64	thr=0.5

E. LLM-Style Decoder Classifier

Journal of Technology Informatics and Engineering (JTIE)
Vol. 4 No. 3 December 2025
E-ISSN: 2961-9068; P-ISSN: 2961-8215, Pages 590-612
Journal Link: https://jtie.stekom.ac.id/index.php/jtie/index

To represent decoder-only modeling, we implemented a causal transformer with an upper-triangular attention mask, ensuring that each token attends only to previous tokens. The decoder used the same character vocabulary and maximum length as the character encoder. For classification, we applied a linear head to the final non-padding token representation. Under causal masking, the hidden state at position t can encode only the prefix up to t; therefore, the final token state summarizes the sequence through a left-to-right accumulation of prefix representations. By contrast, the encoder classifiers use a dedicated [CLS] token whose final state attends bidirectionally to all positions, yielding a symmetric sentence representation.

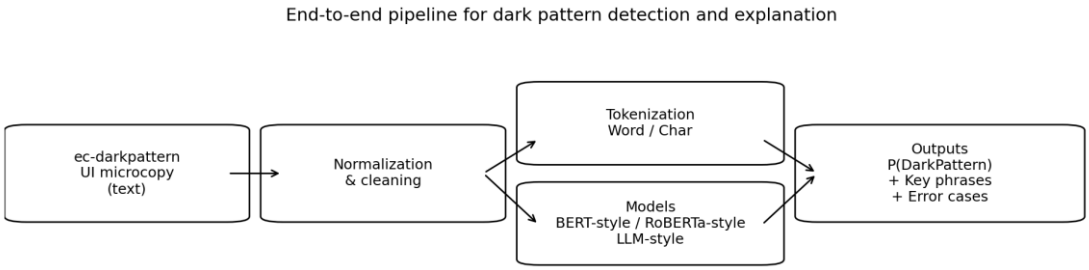

Figure 5. End-to-End Pipeline for Dark Pattern Detection and Explanation

Table 5. Test Performance Across Baselines and Transformer Models (Threshold=0.5 Unless Noted)

Model	Precision	Recall	F1	ROC-AUC
Rule-Lexicon	0.9844	0.5339	0.6923	0.7627
HV+SGD-LogReg	0.9646	0.9237	0.9437	0.9769
HV+SGD-SVM	0.9646	0.9237	0.9437	0.981
BERT-style Encoder (word)	0.8926	0.9153	0.9038	0.9695
RoBERTa-style Encoder (char)	0.8527	0.9322	0.8907	0.9573
LLM-style Decoder (char)	0.7724	0.8051	0.7884	0.8808

F. Tokenization and Hyperparameters

Word tokenization lowercased text and split alphanumeric tokens from punctuation; the resulting vocabulary contained 3,404 tokens, including special symbols. Sequences were [CLS] + tokens + [SEP], truncated/padded to 48 tokens. Character tokenization of lowercased text yielded 126 symbols, including special tokens; sequences were truncated/padded to 96 characters. All transformer models used one layer and four attention heads. The BERT-style encoder used a hidden size of 64, and the character models used a hidden size of 32. We trained transformers for three epochs with AdamW (Loshchilov & Hutter, 2019) and selected the best validation checkpoint. Table 4 reports all settings.

G. Explainability

We generated token-level explanations using gradient-based attribution (gradient×input) with respect to the positive-class logit. For character models, we aggregated character attributions into word-level tokens by summing scores over character spans corresponding to each regex token. This produced key phrases that are interpretable for auditors and designers; Figure 6 shows a representative example.

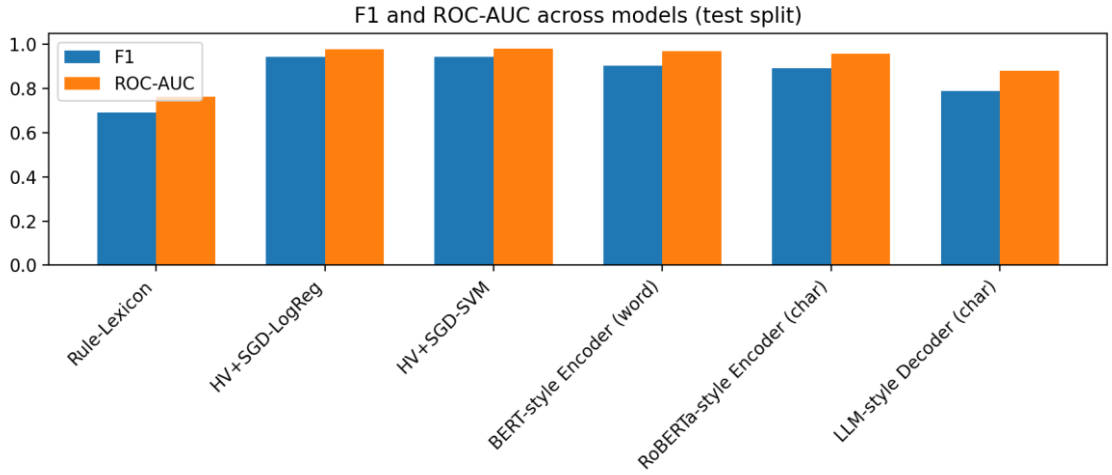

Figure 4. Comparison of F1 and ROC-AUC Across Models on the Test Split

H. Explanation Consistency

We quantified consistency between two models by computing the Jaccard overlap between their top-k token sets (k=5) on the same input. We computed this metric on N=20 randomly sampled test instances as an exploratory stability check, paired with manual inspection of the same cases, and reported the mean and standard deviation per model pair (Table 8). Because this metric captures overlap among highlighted tokens across models, it should be interpreted as a measure of explanation consistency/stability, not as a direct measure of explanation faithfulness.

I. Evaluation

We report precision, recall, F1-score, and ROC-AUC on the held-out test set. ROC-AUC was computed from continuous scores and interpreted as a measure of ranking quality (Fawcett, 2006). We also report category-level recall for each dark pattern category and specificity for non-dark microcopy (Table 6); given the very small counts for some categories, these values are descriptive. To support debugging, we compiled an error-case library by selecting the most confident false positives and false negatives for each neural model and attaching their top attribution tokens (Table 9). Figure 5 summarizes the end-to-end pipeline from microcopy strings to predictions and explanations.

Journal of Technology Informatics and Engineering (JTIE)
Vol. 4 No. 3 December 2025
E-ISSN: 2961-9068; P-ISSN: 2961-8215, Pages 590-612
Journal Link: https://jtie.stekom.ac.id/index.php/jtie/index

IV. RESULT AND DISCUSSION

This section reports empirical results on the held-out test split (n=236). Table 5 summarizes the main performance metrics across all methods, Figure 4 compares F1 and ROC-AUC, and Figure 2 shows ROC curves. We also report category-level recall and specificity (Table 6), confusion counts (Table 7), explanation consistency (Table 8), and representative high-confidence errors (Table 9).

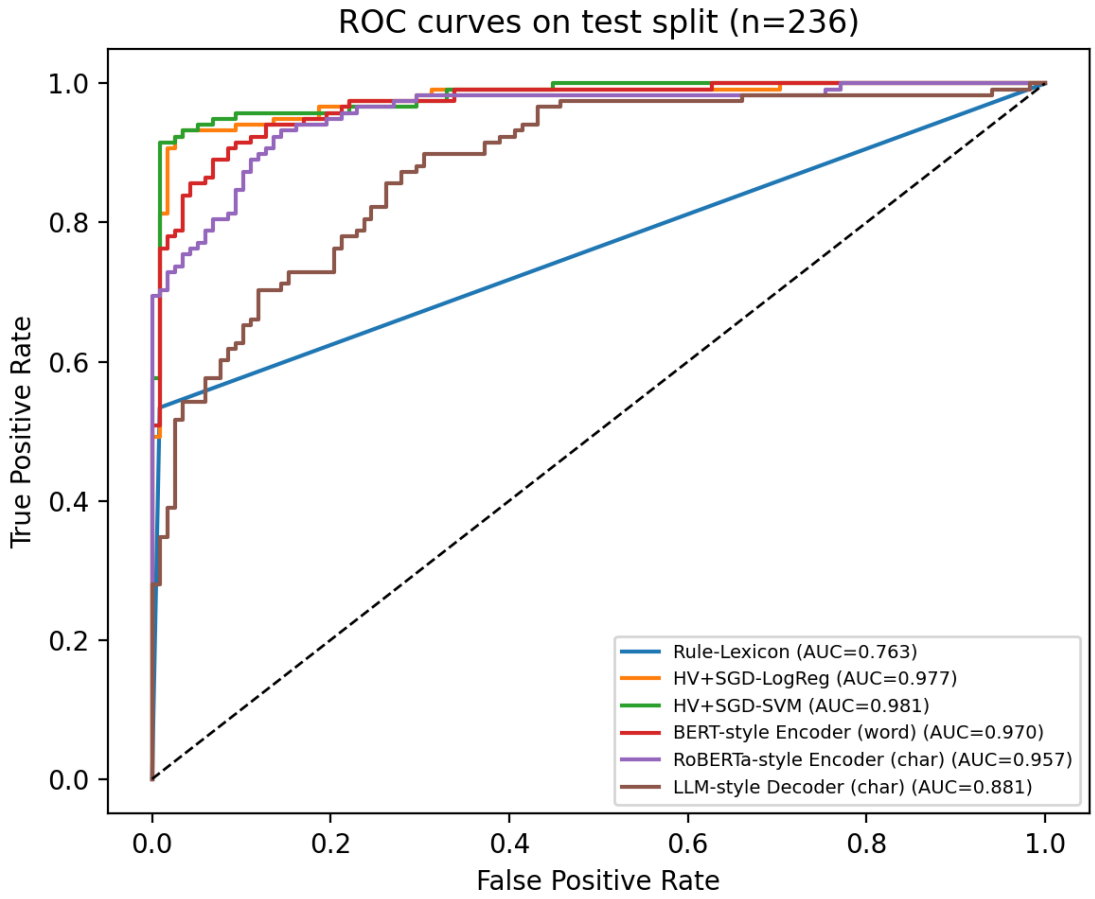

Figure 2. ROC Curves for All Models on the Test Split (n=236)

A. Overall Performance

The hashed n-gram linear models produced the strongest results. The HV+SGD-SVM baseline achieved F1=0.9437 and ROC-AUC=0.9810, and the HV+SGD-LogReg baseline achieved the same F1=0.9437 with ROC-AUC=0.9769. These results confirm that dark pattern microcopy contains highly predictive lexical cues that are effectively captured by sparse n-gram features. The rule-based lexicon baseline was substantially weaker (F1=0.6923), reflecting its limited coverage and inability to capture paraphrases beyond the curated patterns.

B. Transformer Performance

Among neural models, the BERT-style word encoder achieved F1=0.9038 (ROC-AUC=0.9695) and the RoBERTa-style character encoder achieved F1=0.8907 (ROC-AUC=0.9573). The LLM-style causal decoder achieved F1=0.7884 (ROC-AUC=0.8808). Under this from-scratch, low-resource setting, the encoder models consistently outperformed the decoder model, and the word-level encoder slightly outperformed the character-level encoder in overall F1. The gap between the best baseline and the BERT-style encoder was 0.0400 F1, while the gap between the BERT-style encoder and the LLM-style decoder was 0.1154 F1. Because the experiment intentionally excludes large-scale pretraining and uses a single fixed split, these results should be read as a controlled comparison under those constraints rather than as a general ranking of encoder-style versus decoder-style transformers. Because ROC-AUC remained high for the encoders, thresholds can be tuned to trade precision and recall for different auditing workloads, whereas the decoder's lower ROC-AUC indicates weaker ranking stability on this dataset.

C. ROC Analysis

Figure 2 shows that the best-performing models maintain high true positive rates at low false positive rates, consistent with their ROC-AUC values near 0.98. The transformer encoder curves are slightly below the linear baselines, but they still dominate the diagonal across most thresholds. In contrast, the decoder curve approaches the diagonal at higher false-positive rates, indicating that many positives and negatives received similar scores. This pattern aligns with the decoder's reduced F1 and suggests that causal attention limited the model's ability to integrate diagnostic tokens later in the string.

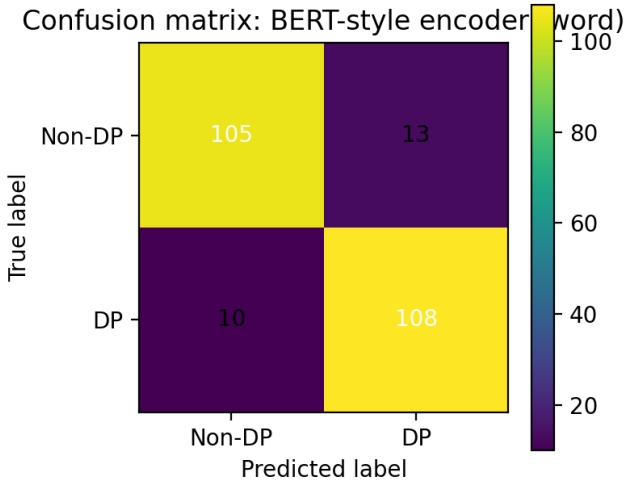

Figure 3. Confusion Matrix for the BERT-Style Encoder (Word Tokenization)

D. Confusion Patterns

Journal of Technology Informatics and Engineering (JTIE)
Vol. 4 No. 3 December 2025
E-ISSN: 2961-9068; P-ISSN: 2961-8215, Pages 590-612
Journal Link: https://jtie.stekom.ac.id/index.php/jtie/index

Figure 3 and Table 7 provide confusion statistics. For the BERT-style encoder, the test confusion matrix contained 108 true positives, 105 true negatives, 13 false positives, and 10 false negatives. This profile shows slightly higher recall than precision (recall=0.9153; precision=0.8926), indicating that errors were split between over-flagging some non-dark strings and missing some dark-pattern strings. In auditing practice, thresholds can be tuned to reduce false negatives at the cost of higher reviewer workload, guided by ROC-AUC.

E. Category-Level Performance

Table 6 reveals how categories contributed to errors. The BERT-style encoder reached perfect recall on Scarcity (1.0000) and strong recall on Social Proof (0.9143), Urgency (0.9130), and Misdirection (0.9167). Performance degraded on the small Sneaking subset (0.5000) and failed on Forced Action in this split (0.0000 on 2 examples). The RoBERTa-style character encoder achieved perfect recall on Forced Action (2/2) and Social Proof (1.0000), and it achieved the strongest Urgency recall (0.9565) among the neural models. However, Table 6 is based on very small per-category test counts for several classes (e.g., n_test=2 for Forced Action and Obstruction; n_test=4 for Sneaking), so these values should be interpreted as descriptive split-specific observations rather than stable estimates.

Table 7. Confusion Counts on the Test Split for Each Model (Threshold=0.5; Hinge Uses Score>=0)

Model	TP	TN	FP	FN
Rule-Lexicon	63	117	1	55
HV+SGD-LogReg	109	114	4	9
HV+SGD-SVM	109	114	4	9
BERT-style Encoder (word)	108	105	13	10
RoBERTa-style Encoder (char)	110	99	19	8
LLM-style Decoder (char)	95	90	28	23

Table 6. Descriptive Category-Level Recall for Dark Pattern Categories and Specificity for Non-Dark Microcopy (Test Split)

Category	MetricType	n_test	BERT_word	RoBERTa_char	LLM_decoder
Forced Action	Recall	2	0.0	1.0	0.5
Misdirection	Recall	12	0.9167	0.9167	0.9167
Not Dark Pattern	Specificity	118	0.8898	0.839	0.7627
Obstruction	Recall	2	1.0	1.0	0.0
Scarcity	Recall	40	1.0	0.95	0.9
Sneaking	Recall	4	0.5	0.75	0.5
Social Proof	Recall	35	0.9143	0.9714	0.7429
Urgency	Recall	23	0.913	0.8696	0.8261

F. Specificity on Non-Dark Microcopy

Table 6 reports specificity for Not Dark Pattern. The BERT-style encoder achieved specificity 0.8898, while the RoBERTa-style character encoder achieved 0.8390 and the LLM-style decoder achieved 0.7627. The lower specificity of the character-based models indicates more false positives on non-dark promotional text, suggesting that character tokenization benefits sensitivity to rare cues but can require stronger calibration or additional context signals to avoid over-flagging.

G. Explainability: Qualitative Highlights

Figure 6 illustrates token-level attributions for the example microcopy "FLASH SALE | LIMITED TIME ONLY Shop Now". The highlighted tokens included 'limited' and 'time', which are canonical urgency markers, and the attribution ranking aligned with the intuitive rationale that the message creates time pressure. This example demonstrates that gradient-based attributions can surface actionable phrases for rewriting, such as removing "limited time" or replacing it with a factual schedule.

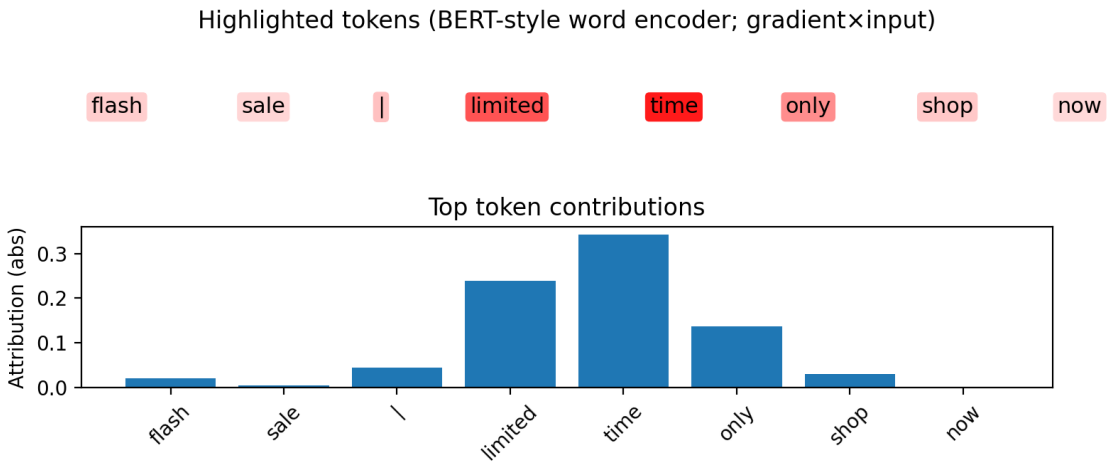

Figure 6. Token-Level Attribution Example (Gradient×Input) for an Urgency Microcopy String

H. Explainability: Consistency Across Models

Table 8 reports explanation consistency as Jaccard overlap between top-5 token sets across model pairs on 20 test instances. The two character-based models (RoBERTa-style encoder vs. LLM-style decoder) achieved the highest mean overlap (0.7482), which reflects their shared character tokenization and similar feature granularity. The BERT-style word encoder overlapped less with each character model (0.4164 with the character encoder; 0.4546 with the decoder), indicating that word tokenization and character tokenization highlight different rationales even when predictions agree. Because Table 8 is based on a 20-instance exploratory sample, the numerical differences should be read as indicative rather than definitive. Moreover, Jaccard overlap measures agreement in highlighted tokens across models; it does not by itself establish that those

Journal of Technology Informatics and Engineering (JTIE)
Vol. 4 No. 3 December 2025
E-ISSN: 2961-9068; P-ISSN: 2961-8215, Pages 590-612
Journal Link: https://jtie.stekom.ac.id/index.php/jtie/index

highlights are faithful causal explanations. In practice, this suggests that a deployment that mixes tokenizations should present explanations at a common word-phrase level to avoid confusing reviewers.

Table 8. Exploratory Explanation Consistency Measured as Jaccard Overlap of Top-5 Token Sets (N=20 Sampled Test Instances)

Model Pair	TopK	N_instances	Jaccard_mean	Jaccard_std
BERT-style Encoder (word) vs RoBERTa-style Encoder (char)	5	20	0.4164	0.3628
BERT-style Encoder (word) vs LLM-style Decoder (char)	5	20	0.4546	0.3457
RoBERTa-style Encoder (char) vs LLM-style Decoder (char)	5	20	0.7482	0.311

I. Error Case Analysis

Table 9 summarizes high-confidence false positives and false negatives for each neural model and attaches influential tokens. A recurring false-positive pattern was promotional language that resembled urgency or scarcity but could be legitimate, such as ordinary discount announcements without pressure cues. False negatives often occurred when dark pattern microcopy used indirect phrasing, euphemisms, or multi-clause descriptions that did not include canonical trigger words. For example, some obstruction messages described contact requirements or process steps without using explicit words such as 'cancel', which reduced the lexical signal for both linear and neural models. The attribution tokens in Table 9 provide concrete anchors for refining lexicons, collecting additional training data, or writing pattern-specific annotation guidelines.

J. Summary of Findings

On this single fixed split, the main determinant of performance was the degree to which the model family matched the microcopy structure. Linear n-gram models were strong because microcopy is short and contains repeated templates; transformer encoders improved contextual modeling but did not surpass the strongest sparse baseline under the from-scratch setting; and the causal decoder underperformed, consistent with its directional information flow and single-state classification design. Explanations were informative and comparatively stable within tokenization families, and the highest consistency was observed between the two character-level transformers. These summary statements should be interpreted within the low-resource, no-pretraining setting of this study.

K. Threshold Sensitivity

Although Table 5 reports metrics at a fixed probability threshold of 0.5 (and 0 for the hinge decision score), deployments are frequently audited, and thresholds are often tuned to meet operational constraints. We therefore tuned thresholds on the validation set by maximizing

validation F1, then applied them to the test set. This procedure improved the HV+SGD-LogReg model from F1=0.9437 to F1=0.9483 on the test set at a tuned threshold of 0.46, while other models changed marginally (e.g., the LLM-style decoder improved from F1=0.7884 to F1=0.7918 at threshold 0.47). These results confirm that ROC-AUC values near 0.98 translate into useful threshold flexibility for linear baselines, whereas threshold tuning does not close the performance gap for the causal decoder.

Table 9. Error Case Library: High-Confidence False Positives (FP) and False Negatives (FN) With Top Attribution tokens

Model	Error	Text	TrueLabel	PredProb	TopTokens
BERT-style Encoder (word)	FP	Frame out of stock in this colour	0	0.9916	in, this, stock, of, out
BERT-style Encoder (word)	FP	This chart should be used as a guide only. Due to each manufacturer's size scale incons...	0	0.9524	should, be, size, due, only
BERT-style Encoder (word)	FN	404 Orders	1	0.042	orders, 404
BERT-style Encoder (word)	FN	Add to Wish List(13 People Have Added)	1	0.171	people, add, 13, have, to
RoBERTa-style Encoder (char)	FP	The requested page could not be found	0	0.9004	found, could, page, requested, the
RoBERTa-style Encoder (char)	FP	★ ★ ★ ★ ★	0	0.8697	★, ★, ★, ★, ★
RoBERTa-style Encoder (char)	FN	Low Stock	1	0.0312	stock, low
RoBERTa-style Encoder (char)	FN	Sale ends soon	1	0.035	sale, soon, ends
LLM-style Decoder (char)	FP	If I emailed you how much I love each jersey you'd have an essay coming your way!	0	0.9494	if, an, your, emailed, '
LLM-style Decoder (char)	FP	To enable notifications, click 'Allow' when prompted.	0	0.9355	allow, notifications, prompted, ,, '
LLM-style Decoder (char)	FN	Low Stock	1	0.0188	stock, low
LLM-style Decoder (char)	FN	NO THANKS	1	0.0263	thanks, no

L. Performance Versus Microcopy Length

ec-darkpattern includes a minority of long interface strings (some exceeding several hundred characters). Because our compact transformer models used fixed maximum lengths (48 word tokens or 96 characters), long strings were truncated, potentially removing cues later in the text. We therefore analyzed test performance across character-length bins. For strings of 100 characters or fewer (214/236 test instances), all models achieved high F1, including HV+SGD-SVM (F1=0.9595 for <=50 chars and 0.9744 for 51-100 chars) and the BERT-style encoder (F1=0.9371

and 0.9048). For strings longer than 100 characters (22/236 instances), F1 dropped for every model, with the strongest baseline falling to 0.7368 and the causal decoder falling to 0.3333. This finding directly links sequence truncation and model design to error rates and supports a practical recommendation: when deploying microcopy detection, either increase the maximum length, chunk long strings, or pre-filter for the most salient span (e.g., the visible banner sentence) before classification.

Why linear baselines are strong. A notable outcome of Table 5 is that sparse n-gram models outperformed the from-scratch transformer encoders. This result is consistent with two properties of the task. First, many dark pattern strings contain near-template phrasing that repeats across sites (e.g., "Only X left", "People are viewing", "Hurry", "No thanks"). Second, the dataset is balanced but modest in size (1,884 training instances), which limits the benefit of high-capacity neural models without large-scale pretraining. In this setting, linear models can directly memorize and generalize over lexical templates with minimal overfitting, while transformers must learn both token embeddings and contextual composition from scratch. This should therefore be read as a result about short, templated microcopy under low-resource training, not as a general claim that sparse models dominate pre-trained transformers. This also explains why Yada et al. (2022) reported very high accuracy when using pre-trained transformers: pretraining supplies robust lexical and syntactic representations that are not available in our controlled from-scratch comparison.

Why the causal decoder underperformed. The decoder model shared the same character vocabulary and sequence length as the character encoder, but it imposed a causal mask and used the final token state as the classification representation. In the encoder models, the [CLS] token attends bidirectionally to all positions at every layer, and the other tokens can also exchange left and right context before their information is pooled into [CLS]. In the decoder classifier, by contrast, the hidden state at position t can encode only the prefix up to t, so the final state must summarize the whole sequence through a left-to-right accumulation of prefix representations. Although the last state can attend to all earlier tokens, earlier token states were formed without access to later evidence, so cross-token interactions are integrated less symmetrically than in the encoder setting. For short microcopy, where a decision may depend on combining an early promotional cue with a late refusal or action phrase, this asymmetry can be disadvantageous—especially in a small from-scratch model with limited capacity. This architectural difference is consistent with the decoder's lower ROC-AUC and its stronger degradation on long strings under truncation.

M. Explanation Differences Across Tokenizations

The attribution examples and the consistency scores in Table 8 also indicate that tokenization changes the granularity of explanations. Word models tended to highlight semantically meaningful units (e.g., 'limited', 'time', 'only'), while character models sometimes highlighted fragments that required aggregation to be interpretable. After aggregation to regex tokens, explanations became comparable, but overlaps between word and character explanations remained moderate. This suggests that explanation interfaces should normalize to word-level phrases even when the underlying model uses subword or character tokens.

V. CONCLUSION AND RECOMMENDATION

This paper presented a fully reproducible empirical study of text-only dark pattern detection and explanation using interface microcopy and button text. Using the ec-darkpattern dataset (2,356 labeled strings), we compared rule-based lexicon detection, hashed n-gram linear models, a lightweight BERT-style encoder, a lightweight RoBERTa-style encoder, and an LLM-style causal decoder classifier. We also implemented token-level explanations using gradient-based attributions, quantified explanation consistency via top-k token overlap, and curated an error case library to support targeted debugging. The study is intentionally narrow: it evaluates small models trained from scratch on a single fixed split, so the results isolate architectural behavior under low-resource, no-pretraining conditions rather than serving as a general comparison of all encoder- and decoder-style transformer systems.

Three main conclusions follow directly from the experimental results. First, dark pattern microcopy in ec-darkpattern contains strong lexical templates, and sparse n-gram models captured these templates extremely well. On the fixed test split, the HV+SGD-SVM and HV+SGD-LogReg baselines achieved the highest F1 (0.9437) and ROC-AUC values near 0.98. Second, transformer encoders trained from scratch performed well but did not surpass the strongest sparse baseline; the BERT-style word encoder reached F1=0.9038 and the RoBERTa-style character encoder reached F1=0.8907. Third, the LLM-style causal decoder underperformed both encoders and linear baselines (F1=0.7884). In this implementation, the decoder classifier uses the final token state under causal masking, whereas the encoder classifiers use a dedicated bidirectional [CLS] representation; this difference provides a concrete architectural account of the observed gap. Because the experiment uses a single split and excludes pretraining, these conclusions should be read as evidence about this controlled setting rather than as a universal ordering of model families.

For explainability, gradient-based attributions consistently highlighted canonical urgency and scarcity phrases and produced actionable key phrase summaries. Explanation consistency was highest between the two character-based transformers (mean Jaccard 0.7482 for top-5 tokens),

while overlaps between word and character explanations were moderate. Because the consistency metric is based on token-overlap over 20 sampled test instances, it should be interpreted as an exploratory measure of stability across models, not as a direct measure of explanation faithfulness. This result indicates that explanations are most stable within tokenization families and motivates presenting explanations at a unified phrase level when models differ internally.

For short, templated e-commerce microcopy like that in ec-darkpattern, several practical recommendations follow. (1) Start with strong lexical baselines. For microcopy auditing at scale, hashed n-gram linear models provide a strong accuracy-cost trade-off and can be retrained quickly when templates change. (2) Use transformer encoders when contextual generalization is required. Encoders can capture paraphrases and non-local cues, especially when combined with pretraining; they are also suitable when multi-language or domain transfer is expected. (3) Treat decoder-only classification carefully for short microcopy. If decoder-only LLMs are used via prompting or fine-tuning, evaluate them under strict held-out protocols and verify that they do not miss interactions distributed across the sequence. (4) Integrate explanations into review workflows. Token-level key phrases and an error case library accelerate human review and support remediation by highlighting the exact wording that triggered a flag. These recommendations are scoped to short, largely text-only microcopy and should be re-validated before being generalized to longer, richer, or multimodal interfaces.

Future work should extend microcopy detection in four directions while maintaining reproducibility. First, replace the single fixed split with repeated stratified splits or cross-validation to quantify variance and provide more stable category-level estimates. Second, incorporate pre-trained encoders and compare them under the same data protocol to quantify the contribution of pretraining relative to sparse baselines. Third, expand rare categories such as Forced Action and Sneaking through targeted data collection to enable stable category-level estimates and to reduce variance across splits. Fourth, evaluate explanation quality with human studies: stability and overlap are useful quantitative signals, but auditors ultimately need explanations that align with perceived manipulative intent and that support actionable rewrites. Combining text-based detection with lightweight structural signals (e.g., whether text is on a refusal button) may also improve precision without sacrificing scalability.

A final practical implication concerns handling long strings. Our length analysis showed that all models degraded on strings longer than 100 characters, and the causal decoder degraded sharply. In deployed systems, this limitation can be addressed by increasing the maximum length, splitting long UI text into sentences, or focusing on the most user-visible fragment. Because many dark patterns rely on short salient cues, careful segmentation can improve both detection and

explanation without increasing model size. Overall, ec-darkpattern provides a valuable benchmark for short-microcopy auditing, and the experimental artifacts in this study demonstrate how accuracy metrics and explanation metrics can be reported together for transparent, reviewer-centric dark pattern detection.

REFERENCES

Ashofi, A. A. (2023). The Impact of Data Security, Ease of Use, and Access Speed on User Trust in Mobile Banking Applications. *Journal of Management and Informatics*, *2*(3), 106–115. https://doi.org/10.51903/jmi.v2i3.148

Brignull, H. (2010). *Dark Patterns*. DarkPatterns.org. https://www.darkpatterns.org

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL-HLT 2019*, 4171–4186. https://aclanthology.org/n19-1423

Di Geronimo, L., Braz, L., Fregnan, E., Palomba, F., & Bacchelli, A. (2020). UI Dark Patterns and Where to Find Them. In *Proceedings of CHI 2020*, 1–14. https://doi.org/10.1145/3313831.3376600

Fawcett, T. (2006). An Introduction to ROC Analysis. *Pattern Recognition Letters*, 27(8), 861–874. https://doi.org/10.1016/j.patrec.2005.10.010

Gray, C. M., Kou, Y., Battles, B., Hoggatt, J., & Toombs, A. L. (2018). The Dark (Patterns) Side of UX Design. In *Proceedings of CHI 2018*, 1–14. https://doi.org/10.1145/3173574.3174108

Heraditya, N. C., Firmansyah, T. W., Yulianto, N. B., Faisal, S. A., & Supriyono. (2026). Implementing Odoo-Based ERP Sales and Inventory Modules (Case Study: UMKM Sirup Cap Manggis). *JUISI: Jurnal Ilmiah Sistem Informasi*, *5*(2), 1–15. https://doi.org/10.51903/ygw51693

Jain, S., & Wallace, B. C. (2019). Attention Is Not Explanation. *arXiv Preprint arXiv:1902.10186*. https://arxiv.org/abs/1902.10186

Joachims, T. (1998). Text Categorization With Support Vector Machines: Learning With Many Relevant Features. In *Proceedings of the 10th European Conference on Machine Learning (ECML 1998)*, 137–142. https://doi.org/10.1007/bfb0026683

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... Stoyanov, V. (2019). RoBERTa. *arXiv Preprint arXiv:1907.11692*. https://arxiv.org/abs/1907.11692

Loshchilov, I., & Hutter, F. (2019). Decoupled Weight Decay Regularization. In *ICLR*. https://arxiv.org/abs/1711.05101

Lundberg, S. M., & Lee, S.-I. (2017). A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems 30*, 4765–4774.

https://proceedings.neurips.cc/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html

Mathur, A., Acar, G., Friedman, M. J., Lucherini, E., Mayer, J., Chetty, M., & Narayanan, A. (2019). Dark Patterns at Scale. *Proceedings of the ACM on HCI*, 3(CSCW), Article 81. https://doi.org/10.1145/3359183

Mathur, A., Narayanan, A., & Chetty, M. (2021). What Makes a Dark Pattern... Dark? *arXiv Preprint arXiv:2101.04843*. https://arxiv.org/abs/2101.04843

Nouwens, M., Liccardi, I., Veale, M., Karger, D., & Kagal, L. (2020). Dark Patterns After the GDPR. In *Proceedings of CHI 2020*, 1–13. https://doi.org/10.1145/3313831.3376321

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). Why Should I Trust You? In *KDD*, 1135–1144. https://doi.org/10.1145/2939672.2939778

Samek, W., Montavon, G., Vedaldi, A., Hansen, L. K., & Müller, K.-R. (Eds.). (2019). *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Springer Cham. https://doi.org/10.1007/978-3-030-28954-6

Santoso, J. T., & Yan, S. (2024). A Hybrid Approach to Typo Correction in Indonesian Documents Using Levenshtein Distance. *Journal of Technology Informatics and Engineering*, *3*(2), 151–168. https://doi.org/10.51903/jtie.v3i2.184

Soe, W. H., Santos, C., & Slavkovik, M. (2022). Automated Detection of Dark Patterns. *arXiv Preprint arXiv:2204.11836*. https://arxiv.org/abs/2204.11836

Sundararajan, M., Taly, A., & Yan, Q. (2017). Axiomatic Attribution for Deep Networks. In *Proceedings of the 34th International Conference on Machine Learning*, 3319–3328. https://proceedings.mlr.press/v70/sundararajan17a.html

Thaler, R. H., & Sunstein, C. R. (2008). *Nudge: Improving Decisions About Health, Wealth and Happiness*. Yale University Press. https://yalebooks.co.uk/book/9780300146813/nudge

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention Is All You Need. In *Advances in Neural Information Processing Systems 30*, 5998–6008. https://arxiv.org/abs/1706.03762

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention Is All You Need. In *Advances in Neural Information Processing Systems 30*, 5998–6008. https://arxiv.org/abs/1706.03762

Wibowo, M. C., & Santoso, J. T. (2024). Utilizing PHPMyAdmin for System Design in Enterprise Administration. *Journal of Technology Informatics and Engineering*, *3*(2), 217–234. https://doi.org/10.51903/jtie.v3i2.193

Yada, Y., Feng, J., Matsumoto, T., Fukushima, N., Kido, F., & Yamana, H. (2022). Dark Patterns in E-Commerce: A Dataset and Its Baseline Evaluations. In *Proceedings of the 2022 IEEE International Conference on Big Data (Big Data 2022)*, 3015–3022. https://doi.org/10.1109/bigdata55660.2022.10020800