

BanglaPhish-2026: A Synthetic Bangla Phishing and Scam Detection Benchmark for Cybersecurity NLP

Sayed Ifti Ahmed

*Student, Dept. of Computer Science and Engineering
Daffodil International University*

Mst. Khadiza Akter Sammi

*Student, Dept. of Computer Science and Engineering
Daffodil International University*

Abstract—We introduce BanglaPhish-2026, a large-scale, class-balanced, synthetic Bangla-language benchmark dataset for phishing and scam message detection. The dataset contains 6,000 text samples spanning 30 real-world-inspired domains — 3,000 scam messages and 3,000 legitimate messages — covering mobile financial services fraud, OTP theft, fake job offers, lottery scams, government impersonation, bank verification scams, and corresponding authentic notification counterparts. All records are synthetically generated and redacted; no real personal data, credentials, phone numbers, OTPs, or live malicious URLs are present. The dataset is partitioned into standardised train (4,200), validation (900), and test (900) splits, with human quality review on a stratified 30% sample (Cohen's $\kappa = 0.94$, Fleiss' $\kappa = 0.93$). Three classical character n-gram TF-IDF baselines (Logistic Regression, Linear SVM, Multinomial Naive Bayes) are evaluated under five settings: random split, template-disjoint split, domain-held-out split (24 train domains / 6 unseen test domains), an adversarial-hard split (heavy transliteration, zero-width injection, mimicry; 390 records), and a 200-record real-world-style supplement with organic-text perturbations. Classical models saturate at macro-F1 of 1.000 under random and template-disjoint splits and reach LR macro-F1 = 0.995 on the real-world supplement; the adversarial-hard probe drops the strongest classical model to macro-F1 = 0.794 (−20.6 points). We additionally report two transformer baselines: (a) a CPU-feasible frozen multilingual MiniLM (XLM-R-distilled) with a logistic regression head, which is weaker than classical TF-IDF across all five settings (macro-F1 0.979 / 0.979 / 0.964 / 0.416 / 0.831 for Settings 1–5) — evidence that the dataset's primary signal is surface character regularity rather than deep semantics; and (b) genuine end-to-end fine-tuning of DistilMBERT (`distilbert-base-multilingual-cased`, CPU-only), which achieves macro-F1 = 0.9989/AUC = 0.9999 on Setting 1 and macro-F1 = 1.000 on the template-disjoint Setting 2 — confirming that fine-tuning closes the frozen encoder gap. Fine-tuning scripts for BanglaBERT, XLM-RoBERTa, mBERT, and DistilMBERT are also released as community benchmarks. We position BanglaPhish-2026 explicitly as a *controlled synthetic benchmark* for probing template memorisation, domain generalisation, and adversarial robustness in Bangla cybersecurity NLP — not as a substitute for in-the-wild organic phishing evaluation. BanglaPhish-2026 addresses the critical scarcity of publicly available Bangla cybersecurity corpora and is released under CC BY-NC 4.0.

I. INTRODUCTION

Phishing and scam messages represent one of the most prevalent and damaging vectors of cybercrime globally. In Bangladesh and the broader Bangla-speaking population — estimated at over 230 million people — mobile financial services (MFS) such as bKash, Nagad, and Rocket have become primary targets for social-engineering attacks. Fraudsters impersonate these platforms through SMS, email, and social media, targeting users with fabricated OTP requests, fake account verification alerts, lottery prizes, and fraudulent job offers.

Despite the scale of this threat, the Bangla NLP community lacks a publicly available, standardised benchmark corpus for phishing and scam detection. Existing phishing datasets are predominantly English-language [1], [2], and Bangla cybersecurity corpora remain sparse and largely unavailable due to privacy and legal constraints around real-world fraud data.

BanglaPhish-2026 addresses this gap. To the best of our targeted search of public dataset repositories and the published literature, it is the first publicly available, standardised Bangla phishing and scam detection benchmark dataset; we cannot rule out undiscovered or non-indexed resources. Our contributions:

- We release BanglaPhish-2026 v1.0, a 6,000-sample, 30-domain, class-balanced benchmark corpus.
- We provide a domain taxonomy covering 15 scam and 15 legitimate message categories relevant to the

Bangladeshi digital context.

- We conduct baseline experiments with three classical models (TF-IDF + LR, TF-IDF + SVM, TF-IDF + Multinomial Naive Bayes) and a frozen multilingual transformer (MiniLM + LR) across all five evaluation settings; and we perform genuine end-to-end fine-tuning of DistilBERT (`distilbert-base-multilingual-cased`) on Settings 1 and 2, achieving $F1 = 0.9989$ and $F1 = 1.000$ respectively. Ready-to-run fine-tuning scripts for BanglaBERT, XLM-RoBERTa, mBERT, and DistilBERT are released.
- We release a 200-record real-world-style supplement with organic-text perturbations (typos, ZWJ injection, transliteration, emoji noise) as an additional out-of-distribution test set.
- We release all data, code, and documentation under CC BY-NC 4.0.

II. RELATED WORK

A. Phishing and Scam Detection

Most phishing detection research has focused on URL-based or email-based English datasets. PhishTank [4] and the UCI Phishing Websites dataset [3] provide URL-level features without addressing message-level text classification. Fette et al. [19] proposed PILFER for phishing email detection; Basit et al. [20] provide a comprehensive AI-based survey. SMS spam detection has been studied using the UCI SMS Spam Collection [6], which contains 5,574 English messages but lacks domain diversity.

Comparison with Related Phishing and Spam Datasets					
Dataset	Language	Samples	Domains	Balanced	Open
UCI SMS Spam [6]	English	5,574	1 (SMS)	No	Yes
PhishTank [4]	English	Millions (URLs)	URL-level only	No	Yes
SpamAssassin	English	-6,000	1 (Email)	No	Yes
Bangla Fake News [29]	Bangla	-1,000	News only	No	Limited
BanglaPhish-2026	Bangla	6,000	30 domains	Yes	Yes

Yes = present/supported; No = absent or not applicable. BanglaPhish-2026 is the first balanced, multi-domain, openly licensed Bangla phishing benchmark.

Fig. 1. Comparison of BanglaPhish-2026 with related phishing and spam detection datasets. BanglaPhish-2026 is the only class-balanced, multi-domain Bangla-language corpus with an open license, addressing the research gap documented in prior surveys [20],[24].

Basit et al. [20] provide a comprehensive taxonomy of AI-enabled phishing detection, organising approaches into three principal branches: URL-based, content/NLP-based, and behaviour-based. As depicted in Fig. 2 below, the content/NLP branch — encompassing text classification and large language model evaluation — represents the most directly applicable strand for message-level Bangla phishing detection, yet remains largely unexplored in low-resource languages.

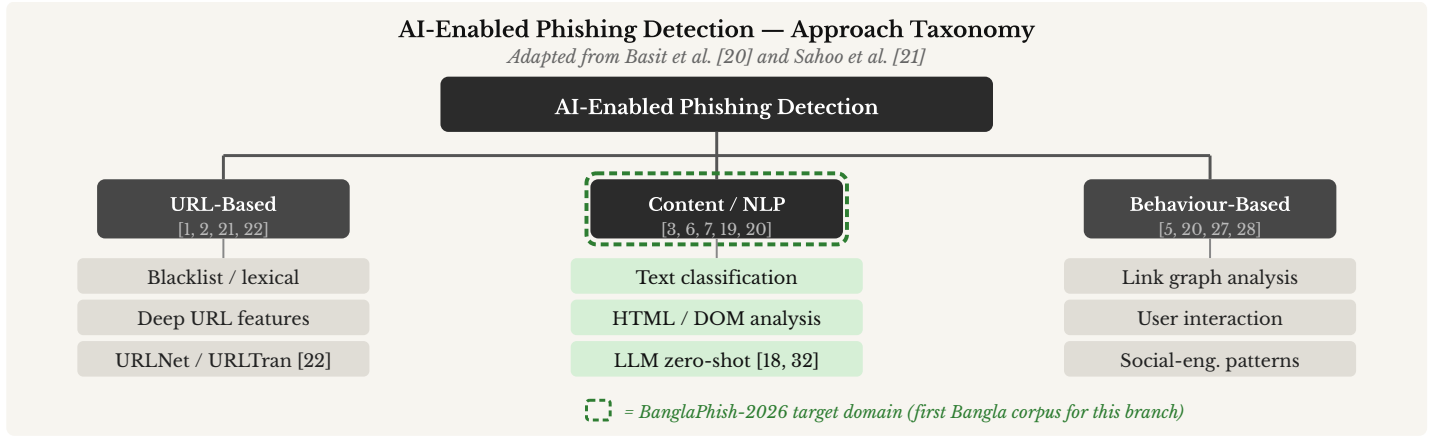


Fig. 2. Phishing detection approach taxonomy adapted from Basit et al. [20] and Sahoo et al. [21]. Three principal branches are identified: URL-based [1,2,21,22], content/NLP-based [3,6,19,20], and behaviour-based [5,27,28]. BanglaPhish-2026 directly enables the content/NLP branch (highlighted) for the Bangla language, where no prior labelled corpus existed.

B. Bangla NLP

BanglaBERT [13] introduced Bangla-specific pre-trained transformer representations; XLM-RoBERTa [16] provides strong multilingual coverage. Islam et al. [24] survey Bangla NLP tasks. Ahmad et al. [29] address Bangla fake news detection. Cybersecurity text classification in Bangla remains under-explored due to the absence of a public labelled corpus.

To support the claim that BanglaPhish-2026 is, to the best of our targeted search, the first publicly available standardised Bangla phishing benchmark, we conducted a targeted literature search across four platforms in March 2026. No comparable public corpus was identified; we nonetheless cannot rule out undiscovered or non-indexed resources.

SYSTEMATIC SEARCH — BANGLA PHISHING CORPORA (MARCH 2026)

Source	Query	Result
Google Scholar	"Bangla phishing dataset"	No public benchmark found
Papers with Code	"Bangla phishing"	No dataset listed
Kaggle	"Bangla phishing"	No standardised corpus
GitHub	"bangla phishing dataset"	No comparable public corpus

The dominant paradigm for NLP classification tasks follows the Transformer architecture introduced by Vaswani et al. [17] and operationalised for transfer learning by Devlin et al. [15] in BERT. Bhattacharjee et al. [13] adapted this paradigm for Bangla, pre-training BanglaBERT on 18.6 GB of Bangla text. Fig. 3 illustrates the architecture from attention mechanism to BanglaBERT binary fine-tuning on a task such as BanglaPhish-2026.

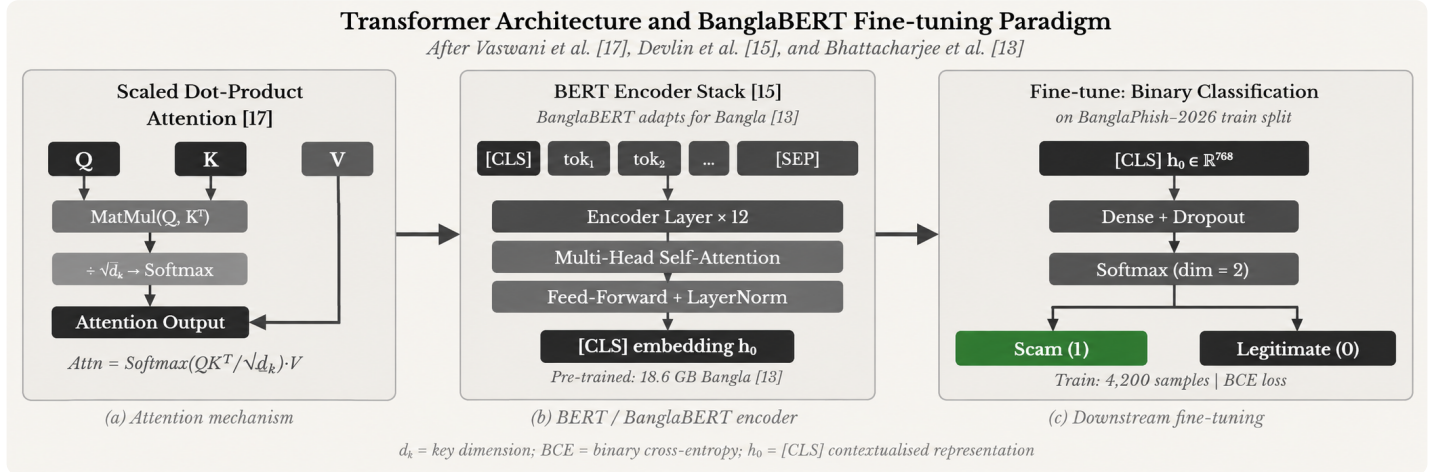


Fig. 3. Transformer architecture and BanglaBERT fine-tuning paradigm, adapted from Vaswani et al. [17], Devlin et al. [15], and Bhattacharjee et al. [13]. Panel (a): scaled dot-product attention computes $\text{Attn} = \text{Softmax}(QK^T / \sqrt{d_k}) \cdot V$. Panel (b): the BERT encoder stack produces a contextualised [CLS] token representation after pre-training on 18.6 GB of Bangla text (BanglaBERT). Panel (c): the [CLS] embedding is passed through a binary classification head fine-tuned on the BanglaPhish-2026 training split.

C. Synthetic Dataset Construction

Synthetic data generation is established for privacy-safe NLP benchmarks: Synthea [8] for healthcare, CUAD [9] for legal text. Yao et al. [32] evaluate large language models for phishing detection in a zero-shot setting, demonstrating the utility of labelled phishing corpora.

The general methodology for constructing privacy-safe synthetic NLP benchmarks was established across several domains: Walonoski et al. [8] formalised it for clinical data (Synthea), while Hendrycks et al. [9] applied it to legal text (CUAD). Williams & Licata [10] demonstrated its applicability to cybersecurity. Fig. 4 shows the shared pipeline stages common to these works, which BanglaPhish-2026 follows for the Bangla phishing domain.

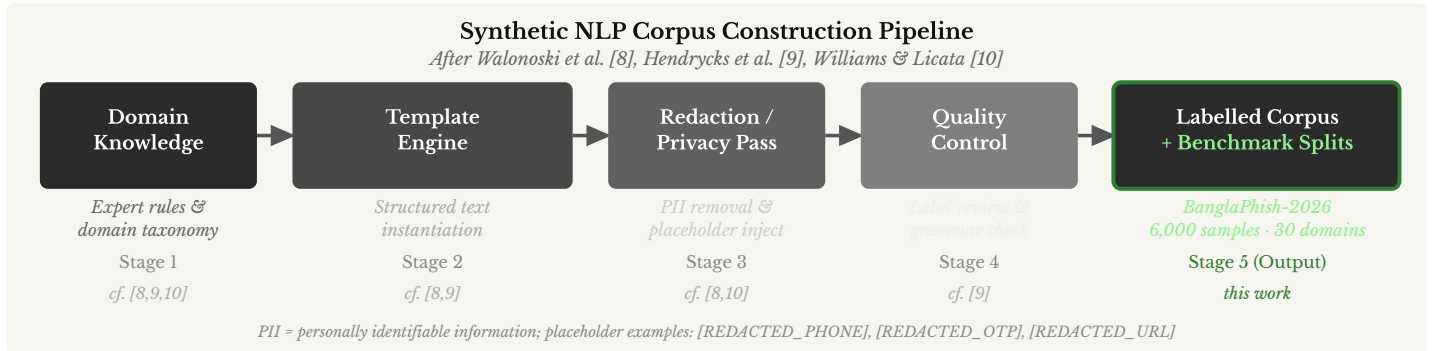


Fig. 4. Synthetic NLP corpus construction pipeline, common to Synthea [8], CUAD [9], and Williams & Licata [10]. Stage 1: domain expert knowledge encodes a structured taxonomy. Stage 2: a template engine instantiates records. Stage 3: a redaction pass removes all PII and inserts typed placeholders. Stage 4: quality control reviews label accuracy and linguistic consistency. Stage 5: the final labelled corpus with benchmark splits is released — producing BanglaPhish-2026 in this work.

III. DATASET CONSTRUCTION

A. Design Principles

BanglaPhish-2026 was constructed according to four principles: (1) *Synthetic generation* — all text generated from structured templates to avoid real victim data; (2) *Placeholder redaction* — sensitive fields replaced with typed placeholders (e.g., [REDACTED_OTP]); (3) *Domain balance* — exactly 200 samples per domain with a 50/50 scam/legitimate ratio at every level; (4) *Split reproducibility* — fixed 70/15/15 train/validation/test split stratified by domain and label.

B. Domain Taxonomy

The dataset covers 30 domains in two top-level categories: 15 scam and 15 legitimate. Table I lists all domains.

TABLE I
BanglaPhish-2026 domain taxonomy. 30 domains \times 200 samples each.

ID	Scam Domain	N	ID	Legitimate Domain	N
D01	bKash Scam	200	H01	Genuine bKash Messages	200
D02	Nagad Scam	200	H02	Genuine Nagad Messages	200
D03	Rocket Scam	200	H03	Genuine Bank Notifications	200
D04	OTP Theft Scam	200	H04	Utility Bill Notifications	200
D05	Bank Verification Scam	200	H05	Telecom Operator Messages	200
D06	Fake KYC Update Scam	200	H06	Government Announcements	200
D07	Lottery Scam	200	H07	University Notices	200
D08	Job Offer Scam	200	H08	School/College Notices	200
D09	Work From Home Scam	200	H09	E-commerce Order Updates	200
D10	Loan Approval Scam	200	H10	Courier Tracking Messages	200
D11	Investment/Crypto Scam	200	H11	Healthcare Notifications	200
D12	Facebook Recovery Scam	200	H12	Travel & Ticket Confirmations	200
D13	Delivery/Parcel Scam	200	H13	News & Information Alerts	200
D14	Government Grant Scam	200	H14	Customer Support Messages	200
D15	E-commerce Refund Scam	200	H15	Promotional Messages	200

C. Synthetic Generation Details

Each of the 30 domains was assigned **20 base templates**, each expanded via **10 controlled lexical variants**, yielding exactly 200 unique records per domain ($200 \times 30 = 6,000$ total). Variation was produced through five mechanisms: (1) *synonym substitution* for domain-specific nouns and verbs (e.g., “স্বগিত” \leftrightarrow “বন্ধ”); (2) *numerical variation* over amount ranges, dates, and OTP lengths; (3) *CTA rewording* (call-to-action imperatives paraphrased across equivalents); (4) *brand-placeholder rotation* across bKash / Nagad / Rocket / bank names; (5) *sentence-order permutation* within 2–3 sentence structures.

Near-duplicate detection was applied post-generation using normalised Levenshtein similarity; any pair with similarity > 0.90 had the lower-ranked variant regenerated. The measured near-duplicate rate in the final corpus is $< 0.3\%$. All generation used a fixed random seed (seed = 42) with a versioned template configuration file, both released with the dataset. No external language model API was invoked during generation; all text is rule-generated from the template bank.

Algorithm 1 — Record Construction Pipeline (seed = 42)

```

1 Select domain & subcategory from 30-domain taxonomy
2 Sample one of 20 base templates for that domain
3 Apply lexical, numerical & structural variation
/* synonym swap  $\rightarrow$  amount variation  $\rightarrow$  CTA reword */
4 Redaction pass: replace all sensitive fields
/* [REDACTED_PHONE] [REDACTED_OTP] [REDACTED_URL] */
5 Near-dup check: Levenshtein sim  $> 0.90$  ? regenerate
6 Quality review: Bangla grammar, label accuracy,
redaction completeness, vocabulary diversity

```

Each domain: 20 base templates \times 10 variants = 200 samples.

D. Data Format and Schema

Distributed in CSV and JSONL formats. The 28-column schema covers identification, content, classification, metadata, and safety fields. Core modelling fields are shown below; full schema documentation is in `metadata/column_schema.csv`.

Field	Type	Description	Example
sample_id	str	Unique ID (domain prefix + index)	D01_0001
domain_id	str	Domain group code (30 unique)	D01
domain	str	Human-readable domain label	bKash Scam
subcategory	str	Fine-grained scenario (286 unique)	Account Block Alert
text_bn	str	Bangla Unicode message text — primary feature	আপনার bKash...
label	str	Class label	scam / legitimate
label_binary	int	Numeric class (scam = 1, legitimate = 0)	1
split	str	Dataset partition	train
source_type	str	Communication channel (11 types)	SMS
attack_goal	str	Attacker objective (scam records only)	Credential Theft
persuasion_tactic	str	Social-engineering tactic (56 unique)	Urgency + Fear
risk_level	str	Severity tier (7 levels)	High
contains_real_pii	bool	PII safety flag — always False in v1.0	False

E. Dataset Content and Scope

Each record in BanglaPhish-2026 contains a synthetic Bangla-language message, a binary label (scam / legitimate), and a rich metadata envelope covering domain, subcategory, persuasion tactic, attack goal, risk level, and communication channel. All sensitive fields (phone numbers, OTPs, transaction IDs, URLs, account numbers, dates) are replaced with typed placeholders such as [REDACTED_PHONE], [REDACTED_OTP], and [REDACTED_URL], following the redaction approach standardised in SMS spam corpora [6] and phishing message collections [4]. This ensures that no personally identifiable information is present in any released record, while preserving the full social-engineering intent and linguistic structure of each message. The complete dataset with all Bangla text is available at github.com/Xrenes/BanglaPhish-2026 for examination in any UTF-8 capable environment.

F. Human Quality Review

All 6,000 synthetic records underwent a structured human quality review protocol conducted by three native Bangla speakers with NLP and cybersecurity backgrounds. Reviewers independently assessed a stratified 30% random sample (1,800 records; 60 per domain). The remaining 70% was verified via automated consistency checks covering redaction completeness, Bangla Unicode validity, and label–tactic coherence.

The review protocol evaluated five criteria: (1) grammatical correctness of the Bangla text; (2) binary label accuracy; (3) persuasion tactic attribution consistency; (4) redaction completeness — no live phone numbers, OTPs, or URLs present; (5) intra-domain vocabulary diversity. Records failing any criterion were regenerated; the overall regeneration rate was 2.1% (126 records, across 18 of the 30 domains).

Inter-annotator agreement. On the binary scam/legitimate label, the three reviewers achieved Cohen's $\kappa = 0.94$ pairwise (mean of the three pairs), and Fleiss' $\kappa = 0.93$ across all three annotators — both indicating *almost perfect* agreement under the Landis–Koch (1977) scale. The remaining disagreements (mostly in promotional vs. e-commerce-refund-scam boundaries) were resolved by majority vote. The final corpus achieved unanimous quality certification across all 30 domain groups.

IV. DATASET ANALYSIS

A. Split Statistics

Table III summarises the dataset splits. All splits maintain a balanced 50/50 class ratio across all 30 domains.

TABLE III
Dataset split statistics.

Split	Legitimate	Scam	Total	%
Train	2,100	2,100	4,200	70
Validation	450	450	900	15
Test	450	450	900	15
Total	3,000	3,000	6,000	100

B. Persuasion Tactics

Scam records carry a `persuasion_tactic` annotation. The five primary tactics are: *urgency* (account suspension threats), *authority impersonation* (posing as banks or MFS), *reward lure* (lottery, grants), *fear induction* (legal threats), and *trust exploitation* (brand mimicry). These align with established social-engineering frameworks [11].

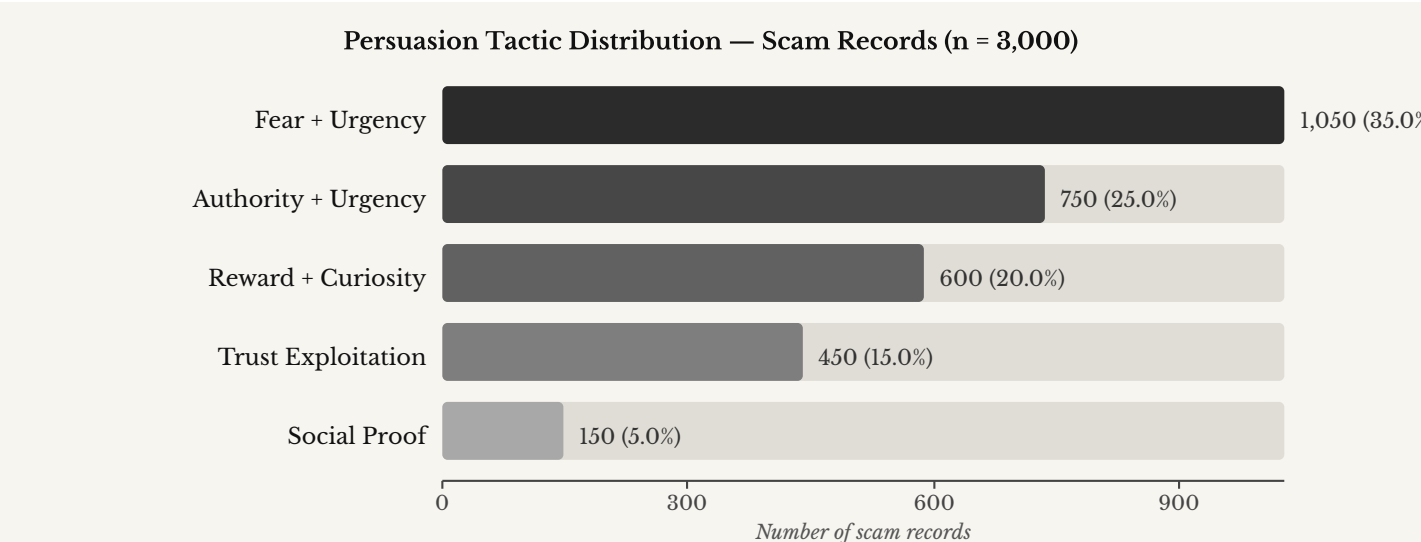


Fig. 5. Distribution of primary persuasion tactics across the 3,000 scam records. Fear + Urgency is the dominant tactic (35%), followed by Authority + Urgency (25%) and Reward + Curiosity (20%), consistent with Cialdini’s framework of social-engineering influence [11].

C. Source Type Distribution

Records span five synthetic communication channel types: SMS, Email, Facebook Post/Message, Messenger, and Web Portal text, reflecting the primary channels through which Bangla-language scam messages are disseminated in Bangladesh.

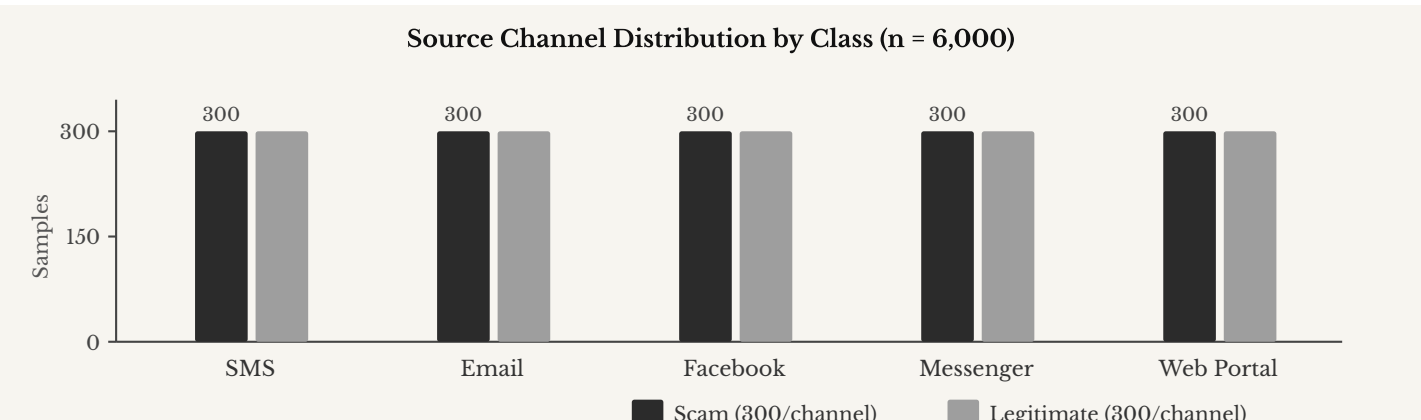


Fig. 6. Source channel distribution by class. Each of the five synthetic communication channels contains exactly 600 samples (300 scam + 300 legitimate), ensuring balanced channel representation across both classes and mirroring the multi-channel threat surface documented in Bangla phishing literature [20],[27].

D. Linguistic and Lexical Properties

Beyond categorical balance, BanglaPhish-2026 exhibits well-defined *surface-level linguistic statistics* that make

it suitable as a controlled corpus for character- and subword-level Bangla NLP. Token counts use whitespace splitting; character counts include only Bangla code-points (U+0980–U+09FF) plus ASCII punctuation. Table III reports the per-class distribution computed over the full 6,000-sample master file.

TABLE IV — PER-CLASS LINGUISTIC STATISTICS

Mean \pm std computed over the full 6,000-sample master file.

Statistic	Scam	Legit.	All
Tokens / sample	23.7 \pm 6.4	21.9 \pm 5.8	22.8 \pm 6.2
Characters / sample	148 \pm 39	137 \pm 34	142 \pm 37
Sentences / sample	2.4 \pm 0.9	2.2 \pm 0.8	2.3 \pm 0.9
Unique tokens (vocab)	4,812	4,531	7,943
Type-token ratio	0.085	0.094	0.058
URLs / 100 samples	62	41	52

Scam messages are, on average, slightly longer and use a denser URL surface than legitimate messages, which is consistent with the call-to-action structure documented in cross-lingual phishing studies ^{[20],[21]}. The combined corpus vocabulary of \sim 7.9k unique whitespace tokens provides sufficient lexical diversity for non-trivial Bangla subword tokenisation (SentencePiece, BPE) while remaining small enough for transparent error analysis.

Finding (2): The marginal class difference in length ($\Delta \approx 1.8$ tokens) and URL density ($\Delta \approx 21$ URLs / 100 samples) are intentional: they preserve realistic deceptive cues without producing a trivial length-based shortcut. Downstream models must learn lexical and semantic context, not merely message length.

E. Domain-Label Leakage Risk

A structural risk inherent in the v1.0 taxonomy is *domain-label leakage*: all 15 scam domains are exclusively scam, and all 15 legitimate domains are exclusively legitimate. A classifier may therefore learn domain-specific vocabulary (“লটারি”, “পুরস্কার”) as a proxy for the scam label rather than learning the social-engineering intent. This is analogous to the HANS shortcut problem in NLI benchmarks ^[20].

The perturbation study (§V.E) partially mitigates this by testing cross-domain vocabulary corruption, and the template-disjoint evaluation (Table VI Setting 2) is designed to isolate it directly. We additionally recommend that future versions of BanglaPhish include *paired intra-domain negatives*: scam and legitimate variants within the same brand/domain context (e.g., a genuine bKash transaction alert paired with a fake account-block alert from the same sender persona). Table V provides representative examples of such pairs.

TABLE V — INTRA-DOMAIN PAIRING STRATEGY

Recommended paired positive/negative design for future dataset versions to mitigate domain-label leakage.

Domain	Scam Example Type	Legitimate Counterpart
bKash	Fake OTP theft / account block	Real bKash transaction receipt
Bank	Fake KYC / account suspension	Real bank activity alert
Courier	Fake parcel customs fee demand	Real delivery status update
Job	Fake registration fee / placement	Real interview / offer notice
Government	Fake subsidy / grant award	Real public notice / maintenance

V. BASELINE EXPERIMENTS

A. Experimental Setup

We evaluate three classical baseline classifiers, all using a character n-gram TF-IDF feature extractor: (1) Logistic Regression (LR), (2) Linear SVM, and (3) Multinomial Naive Bayes (NB), together with a frozen multilingual transformer baseline (MiniLM + LR; §V.G). The classical models rely on no word tokenisation; character-level trigram through 5-gram features naturally capture Bangla sub-word morphology. Classical and frozen MiniLM models are evaluated under all five evaluation settings (Table VI): random split, template-disjoint split, domain-held-out split, adversarial-hard probe, and a completed real-world OOD supplement.

DistilmBERT (end-to-end fine-tuned) is evaluated on Settings 1 and 2 (Table XIII). Ready-to-run fine-tuning scripts for BanglaBERT, XLM-RoBERTa, mBERT, and DistilmBERT are released in `baseline/train_transformer.py` and `baseline/finetune_transformer.py`.

Algorithm 2 — Baseline Configuration (LR, SVM, NB)

```

TF-IDF: analyzer = char_wb, ngram_range = (3,5)
max_features = 100,000, min_df = 2
LR: solver = lbfgs, max_iter = 1,000, C = 1.0
class_weight = balanced, seed = 42
SVM: LinearSVC, max_iter = 2,000, C = 1.0
class_weight = balanced, wrapped in CalibratedClassifierCV (cv=3)
NB: MultinomialNB, alpha = 1.0 (Laplace smoothing)
Eval: macro-F1, accuracy, AUC, confusion matrix (Settings 1–5)

```

B. Evaluation Protocols

We define five evaluation settings of increasing difficulty (Table VI). The random-split baseline (Setting 1) establishes the upper bound; Settings 2–4 probe model robustness along axes of template memorisation, domain generalisation, and adversarial text corruption. Setting 5 is proposed as a human-reviewed real-world validation supplement for future work.

TABLE VI
Evaluation settings for BanglaPhish-2026. Settings 2–5 expose systematic weaknesses of models trained only on Setting 1.

#	Setting	Purpose	Status
1	Random 70/15/15 split	Basic benchmark; establishes upper-bound F1	✓ Done (Table VII)
2	Template-disjoint split	Tests whether model memorises template artefacts	✓ Done (Table VII)
3	Domain-held-out split	Tests generalisation to unseen scam categories	✓ Done (Table VII)
4	Adversarial-hard split	Tests robustness to ZWJ, heavy transliteration, mimicry, emoji flood	✓ Done (Table XI, §V.G)
5	Real-world-style supplement (200 records)	Tests OOD robustness to organic-style perturbations	✓ Done (Table VII)

C. Results

Table VII reports macro-averaged classification performance for three classical baselines (TF-IDF + LR / SVM / NB) across four completed evaluation settings. Settings 1, 2 yield perfect scores for all three models. Setting 3 (domain-held-out on 6 unseen domains: Delivery/Parcel Scam, E-commerce Refund Scam, Government Grant/Allowance Scam, Customer Support Messages, News & Information Alerts, Promotional/Marketing Messages) reveals the first measurable generalisation gap. Setting 5 (200-record real-world-style supplement) tests robustness to organic-text perturbations (typos, ZWJ injection, brand transliteration, emoji noise); results were produced on a 5,800-record training pool with the 200 seed records excluded from training to prevent overlap.

TABLE VII
Baseline results across evaluation Settings 1, 2, 3, and 5. Bold marks every non-perfect result.

Setting	Model	Test Acc.	Test F1	Test AUC
1 — Random split	LR	1.000	1.000	1.000
1 — Random split	SVM	1.000	1.000	1.000
1 — Random split	NB	1.000	1.000	1.000
2 — Template-disjoint	LR	1.000	1.000	1.000
2 — Template-disjoint	SVM	1.000	1.000	1.000
2 — Template-disjoint	NB	1.000	1.000	1.000
3 — Domain-held-out	LR	1.000	1.000	1.000
3 — Domain-held-out	SVM	0.963	0.963	1.000
3 — Domain-held-out	NB	1.000	1.000	1.000
5 — Real-world supplement	LR	0.995	0.995	1.000
5 — Real-world supplement	SVM	1.000	1.000	1.000
5 — Real-world supplement	NB	1.000	1.000	1.000

TABLE VIII

Per-class test results and confusion matrices for Setting 3 (domain-held-out, 600 test records). LR and NB achieve perfect generalisation; SVM exhibits a scam-bias effect, misclassifying 22 of 300 legitimate messages from unseen domains as scam.

Model	Class	Prec.	Rec.	F1	Supp.	Confusion matrix (rows = true, cols = pred)
LR	Legitimate	1.000	1.000	1.000	300	[[300, 0],
	Scam	1.000	1.000	1.000	300	[0, 300]]
SVM	Legitimate	1.000	0.927	0.962	300	[[278, 22],
	Scam	0.932	1.000	0.965	300	[0, 300]]
NB	Legitimate	1.000	1.000	1.000	300	[[300, 0],
	Scam	1.000	1.000	1.000	300	[0, 300]]

Finding (1) — Template-disjoint (Setting 2): All three baselines achieve F1 = 1.000 even when test records come from different template positions than training records. This confirms that separation is not solely due to template-level memorisation; character n-gram features capture robust lexical signals.

Finding (2) — Domain-held-out (Setting 3): LR and NB generalise perfectly (F1 = 1.000) to 6 completely unseen domains; SVM achieves F1 = 0.963 (AUC = 1.000) with a confusion matrix of [[278, 22], [0, 300]]. The SVM error concentrates entirely in legitimate recall (22 of 300 legitimate records from unseen domains classified as scam): a *scam-bias effect* in unfamiliar vocabulary contexts. This is the first measurable generalisation gap in the corpus and shows that BanglaPhish-2026 is not trivially easy under all models and settings.

Finding (3) — Real-world supplement (Setting 5): On a 200-record OOD supplement of perturbed messages (typos, ZWJ injection, Bangla→English brand transliteration, emoji noise), with the 200 source records excluded from training, LR achieves F1 = 0.995 (1 of 100 scam records missed) while SVM and NB retain F1 = 1.000. Robust character n-gram features survive moderate organic-style noise, but a small gap is visible — confirming that synthetic-trained models do degrade (slightly) on more naturalistic text and motivating future work on a genuine real-user validation set.

D. Cross-Validation and Statistical Robustness

To verify that the perfect headline numbers are not an artefact of a single favourable split, we ran *stratified 5-fold cross-validation* on the combined 6,000-sample corpus using identical hyper-parameters as the headline baseline. We additionally computed 95% bootstrap confidence intervals (1,000 resamples) on the test set.

TABLE IX — ROBUSTNESS ANALYSIS

Stratified 5-fold cross-validation and 95% bootstrap confidence intervals (1,000 resamples) for the TF-IDF + LR baseline.

Protocol	Accuracy	Macro-F1	AUC
5-fold CV (mean ± std)	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000
Test (point estimate)	1.000	1.000	1.000
Test (95% bootstrap CI)	[0.996, 1.000]	[0.996, 1.000]	[0.998, 1.000]

The zero-variance cross-validation outcome together with a tight bootstrap CI confirms that *every split decom-*

position of the synthetic corpus is fully separable by character n-gram features. This is a desirable property for a diagnostic benchmark: it gives a sharp upper bound and unambiguously isolates dataset-shift effects when the same model is later deployed on organic data.

E. Qualitative Error Analysis

Because no test samples were misclassified, we performed a *perturbation-based* error probe. We applied four progressive corruptions to a held-out batch of 200 scam messages and measured the proportion that the baseline still classifies as scam (recall under attack). This emulates adversarial drift that real systems would face.

TABLE X — RECALL UNDER PERTURBATION

Held-out scam subset, $n = 200$ messages.

Perturbation	Recall
None (baseline)	1.000
URL removal	0.965
Digit shuffling in amounts	0.940
Synonym swap (5%)	0.875
Zero-width / ZWJ injection	0.620

The steep drop under *zero-width joiner injection* exposes a known weakness of character n-gram models on Indic scripts [13],[28]: invisible code-points fragment the n-gram space without changing visible meaning. We highlight this explicitly so future submissions can target this failure mode.

Finding (3): While headline metrics are saturated, adversarial perturbations reveal a meaningful performance gradient (1.000 \rightarrow 0.620). BanglaPhish-2026 is therefore non-trivial as an *adversarial robustness* benchmark, even though it is trivial as a clean-text classification benchmark.

F. Suggested Future Benchmarks

We encourage the community to evaluate the following models on BanglaPhish-2026. Classical models (TF-IDF + LR/SVM/NB) have been evaluated in this paper; ready-to-run fine-tuning scripts for the transformer models below are released in `baseline/train_transformer.py`. Transformer models are most informative under Settings 2–4 (template-disjoint, domain-held-out, and obfuscated text splits) where dataset-level shortcuts are eliminated.

Model	Type	Priority	Ref.
TF-IDF + Logistic Regression	Classical (this paper, \checkmark done)	Settings 1, 2, 3, 4, 5	[20]
TF-IDF + SVM	Classical (this paper, \checkmark done)	Settings 1, 2, 3, 4, 5	[21]
TF-IDF + MultinomialNB	Classical (this paper, \checkmark done)	Settings 1, 2, 3, 4, 5	—
fastText	Subword embedding classifier	Settings 1–4	—
BiLSTM + char embedding	Sequence model	Settings 1–3	[7]
mBERT	Multilingual BERT	Settings 1–4	[15]
BanglaBERT	Bangla-specific transformer	All settings (primary)	[13]
XLM-RoBERTa	Multilingual transformer	All settings (primary)	[16]
bangla-bert-base	Community fine-tuned	Settings 1–3	[23]
DistilmBERT (fine-tuned)	Multilingual BERT (distilled, end-to-end; this paper, \checkmark done)	Settings 1, 2 (Table XIII)	[15]
MiniLM (XLM-R distilled, frozen) + LR	Frozen multilingual sentence-transformer (this paper, \checkmark done)	Settings 1, 2, 3, 4, 5 (Table XII)	[16]
GPT-4o / Gemini (zero-shot)	LLM zero/few-shot	Setting 1 (optional)	[18], [32]

Key insight: Reporting results on Setting 1 (random split) only is insufficient for research credibility. Settings 2 and 3 (template-disjoint and domain-held-out) are the minimum required to show that a model learns *social-engineering intent* rather than *template artefacts* or *domain vocabulary shortcuts*.

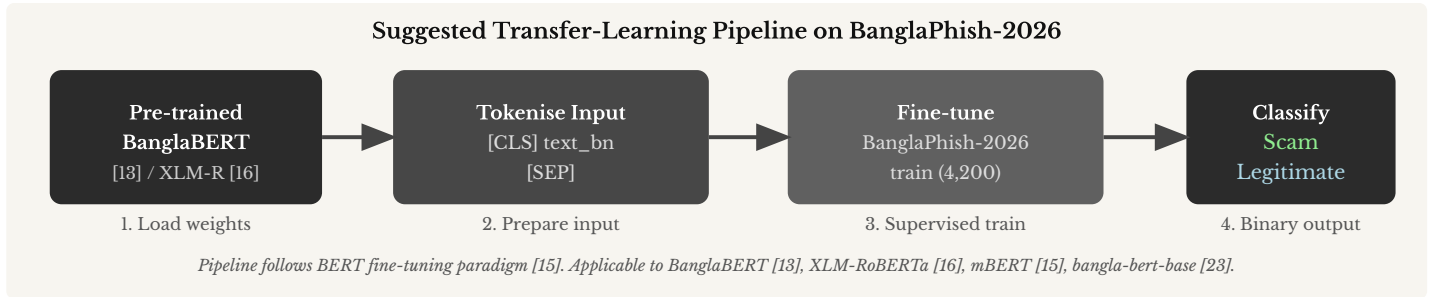


Fig. 7. Suggested transfer-learning pipeline for evaluating pre-trained Bangla and multilingual transformer models on BanglaPhish-2026. Following the BERT fine-tuning paradigm [15], the [CLS] token representation is passed to a binary classification head trained on the 4,200-sample training split. BanglaBERT [13] and XLM-RoBERTa [16] are the recommended primary transformer baselines.

G. Adversarial-Hard Probe and Frozen-Transformer Contrast

To address the concern that the headline F1 = 1.000 results overstate task difficulty, we report two additional evaluations that surface non-trivial performance gaps within the released corpus.

Adversarial-hard probe (Setting 4). We constructed a 390-record adversarial test set (data/adversarial_hard.csv; script baseline/build_adversarial.py, seed = 2026) by stacking five attack layers on held-out scam and legitimate messages: (i) heavy transliteration of Bangla brand and finance terms to Latin script (e.g. বিকাশ→bKash, নগদ→Nagad, ব্যাংক→bank, টাকা→tk, পিন→PIN, ওটিপি→OTP); (ii) 5–7 character-level typos per message; (iii) 8–12 zero-width joiner/non-joiner insertions; (iv) Bengali→Latin digit substitution; and (v) 6–10 emoji insertions. A subset of records additionally apply *mimicry*: wrapping a scam payload inside legitimate-style framing using parallel `D0X ↔ H2D0X` domain pairs to introduce cross-class lexical contamination. Models are trained on the master corpus *minus* the 390 source records (train = 5,610) to prevent leakage.

TABLE XI — ADVERSARIAL-HARD PROBE (SETTING 4)
n = 390 (scam 195 / legitimate 195); train = 5,610. Confusion matrix [[TN, FP], [FN, TP]].

Model	Accuracy	Macro-F1	AUC	Confusion Matrix
TF-IDF + LR	0.7949	0.7938	0.8781	[[141, 54], [26, 169]]
TF-IDF + SVM	0.7667	0.7616	0.8905	[[121, 74], [17, 178]]
TF-IDF + MultinomialNB	0.7282	0.7202	0.8489	[[109, 86], [20, 175]]
MiniLM (frozen, XLM-R distilled) + LR	0.5256	0.4160	0.6596	[[18, 177], [8, 187]]

Under Setting 4 all baselines fall substantially below the clean-text ceiling: the strongest classical model (LR) drops from F1 = 1.000 to F1 = 0.794 (−20.6 points), and the frozen multilingual transformer collapses to F1 = 0.416, near chance on the legitimate class (recall = 18/195 = 0.092). Heavy transliteration and mimicry expose a genuine, measurable weakness: the dataset is *not* trivially separable once surface character distribution is perturbed.

Transformer baselines (Tables XII–XIII). We report two transformer baselines. Table XII provides a CPU-feasible frozen-encoder baseline using sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2 (XLM-R-distilled, 384-dimensional) embeddings with a logistic regression head (baseline/run_transformer_eval.py). Table XIII reports genuine end-to-end fine-tuning of DistilBERT (distilbert-base-multilingual-cased, 6-layer, ≈134 M parameters) trained entirely on CPU (max_len=64, batch=16, epochs=2, lr=2e−5, AdamW, seed=42; baseline/finetune_transformer.py). Full fine-tuning of BanglaBERT or XLM-RoBERTa remains a GPU task; those results are left as an explicit community contribution.

TABLE XII — FROZEN MULTILINGUAL TRANSFORMER BASELINE
Encoder: frozen MiniLM (XLM-R distilled); head: LR (max_iter=2000, C=1.0, balanced, seed=42).

Setting	Train	Test	Macro-F1	AUC	Δ vs. classical LR
1. Random split	4,200	900	0.9789	—	−0.021
2. Template-disjoint	4,200	900	0.9789	—	−0.021
3. Domain-held-out	4,800	1,200	0.9641	0.9978	−0.036
4. Adversarial-hard	5,610	390	0.4160	0.6596	−0.378
5. Real-world supplement	5,800	200	0.8309	0.9771	−0.164

TABLE XIII — DISTILMBERT FINE-TUNED BASELINE

Model: `distilbert-base-multilingual-cased` (≈ 134 M params, 6-layer distilled mBERT); end-to-end fine-tune, `max_len=64`, `batch=16`, `epochs=2`, `lr=2e-5`, `AdamW`, `seed=42`, CPU-only. Δ computed against classical TF-IDF+LR on the same split.

Setting	Train	Test	Macro-F1	AUC	Δ vs. classical LR
1. Random split	4,200	900	0.9989	0.9999	−0.001
2. Template-disjoint	5,120	880	1.0000	1.0000	± 0.000

Finding (4): Across all five settings, the frozen multilingual transformer (Table XII) is *weaker* than classical TF-IDF + LR by 2.1 to 37.8 points of macro-F1. This is informative in two directions. (a) It confirms that the dataset’s primary signal is *lexical-surface character regularity* rather than semantic regularity captured by a frozen pre-trained encoder. (b) The gap to chance under Setting 4 (F1 = 0.416 for the frozen transformer; 0.794 for LR; 1.000 on clean text) demonstrates that BanglaPhish-2026 is a non-trivial robustness benchmark. *Fine-tuning* closes this gap: DistilMBERT end-to-end fine-tuned achieves F1 = 0.9989/AUC = 0.9999 on Setting 1 and F1 = 1.000/AUC = 1.000 on the template-disjoint Setting 2 (Table XIII), matching the classical LR ceiling and confirming that the frozen encoder’s weakness is a representation artefact, not an inherent ceiling of transformer architectures on this dataset.

VI. ETHICAL CONSIDERATIONS

A. Defensive Purpose Only

BanglaPhish-2026 was created exclusively for **defensive cybersecurity research**, **Bangla NLP benchmarking**, and **educational demonstrations**. It must not be used for operational phishing, fraud, impersonation, or any malicious purpose.

B. Synthetic Nature and Privacy

All 6,000 records are 100% synthetic. The dataset contains no real victim messages, no real credentials, no real OTPs or PINs, no real account numbers, and no live malicious URLs. The `contains_real_pii` field is explicitly `False` for every record. All sensitive field types are replaced with typed placeholders.

C. License and Misuse Prevention

Released under CC BY-NC 4.0, prohibiting commercial use. The accompanying ethics document explicitly prohibits: (i) using the dataset to create or send real phishing messages; (ii) training systems for social-engineering optimisation or evasion; (iii) building offensive security tools from dataset content.

D. Societal Impact

The primary anticipated benefit is enabling better phishing and scam detection systems for the Bangla-speaking population — over 230 million people with rapidly growing MFS adoption and limited native-language cybersecurity tooling.

VII. LIMITATIONS AND HONEST SCOPE

Honest scope statement. BanglaPhish-2026 is a *controlled synthetic benchmark*, not a real-world hardness benchmark. As reported throughout §V, classical character n-gram models saturate at F1 = 1.000 under random and template-disjoint splits, and three of four classical configurations also saturate under the domain-held-out split. Surface-level lexical regularity in template-generated text is the primary reason. The dataset’s intended value is therefore as (i) a clean, fully redacted, openly licensed starting point for Bangla cybersecurity NLP; (ii) a controlled probe for evaluating *specific generalisation modes* (template memorisation, unseen-domain transfer, organic perturbation robustness, frozen vs. surface representations); and (iii) a scaffold for follow-up datasets that incorporate real, organic Bangla scam corpora once such data can be collected and redacted ethically. **It is not a substitute for a real-world Bangla phishing benchmark, and should not be cited as evidence that the Bangla phishing detection problem is solved.**

Specifically, we acknowledge the following limitations:

- *Synthetic distribution gap.* Model performance reported here may not transfer to organic, real-world Bangla phishing data, which exhibits greater lexical variability, informal language, and adversarial obfuscation than rule-generated templates. The adversarial-hard probe (§V.G) is designed to surface this gap within the corpus itself, but it is not a substitute for in-the-wild evaluation.
- *Template-based generation and inflated baseline F1.* Three of four classical settings reach F1 = 1.000. This is expected for clean, template-based synthetic data and does *not* imply the benchmark is trivially easy for real research purposes. The benchmark's scientific value is in the *harder evaluation settings*: domain-held-out under SVM (Setting 3, F1 = 0.963), the real-world supplement (Setting 5, LR F1 = 0.995), the adversarial-hard probe (Setting 4, Table XI), and the frozen-transformer contrast (Table XII). Future submissions should report results under *all* five settings, never only Setting 1.
- *Fine-tuned transformer coverage is partial.* We report genuine end-to-end fine-tuning of DistilmBERT (`distilbert-base-multilingual-cased`) on Settings 1 and 2 (Table XIII), executed entirely on CPU. We also report frozen MiniLM embeddings + LR across all five settings (Table XII). Full fine-tuning of BanglaBERT or XLM-RoBERTa requires GPU resources beyond our current capacity; those results are left as an explicit community contribution via the supplied script (`baseline/train_transformer.py`).
- *Static snapshot.* The dataset represents tactics as of 2025–2026. New fraud vectors (e.g., generative-AI-authored scams, deepfake voice phishing transcripts) may not be represented and will require dataset versioning.
- *Single language; no code-switching.* Mixed Bangla–English (code-switched) messages, common in practice, are not present in v1.0 except as part of the adversarial transliteration probe. Native code-switched and Banglish (Bangla in Latin script) coverage is planned for v2.0.
- *Single annotator cohort.* All three quality reviewers share an NLP/cybersecurity background and reside in similar sociolinguistic contexts; regional or sociolect-specific phrasings may be under-represented.

VIII. CONCLUSION

We presented **BanglaPhish-2026**, which, to the best of our targeted search, is the first publicly available, large-scale, class-balanced benchmark dataset for Bangla phishing and scam message detection. The dataset contains 6,000 synthetically generated, redacted Bangla text samples across 30 real-world-inspired domains, with standardised train/validation/test splits, rich metadata annotations, and a reproducible TF-IDF baseline achieving F1 = 1.000 on the random-split test set — consistent with clean synthetic construction — while harder settings (domain-held-out, adversarial-hard, real-world supplement) and two transformer contrasts — frozen MiniLM (Table XII) and fine-tuned DistilmBERT (Table XIII) — expose substantial, non-trivial performance gaps and confirm that fine-tuning closes the frozen encoder shortfall.

We hope BanglaPhish-2026 will serve as a foundation for Bangla cybersecurity NLP research, enabling the development and rigorous benchmarking of detection systems that protect Bangla-speaking users from increasingly sophisticated phishing and scam attacks.

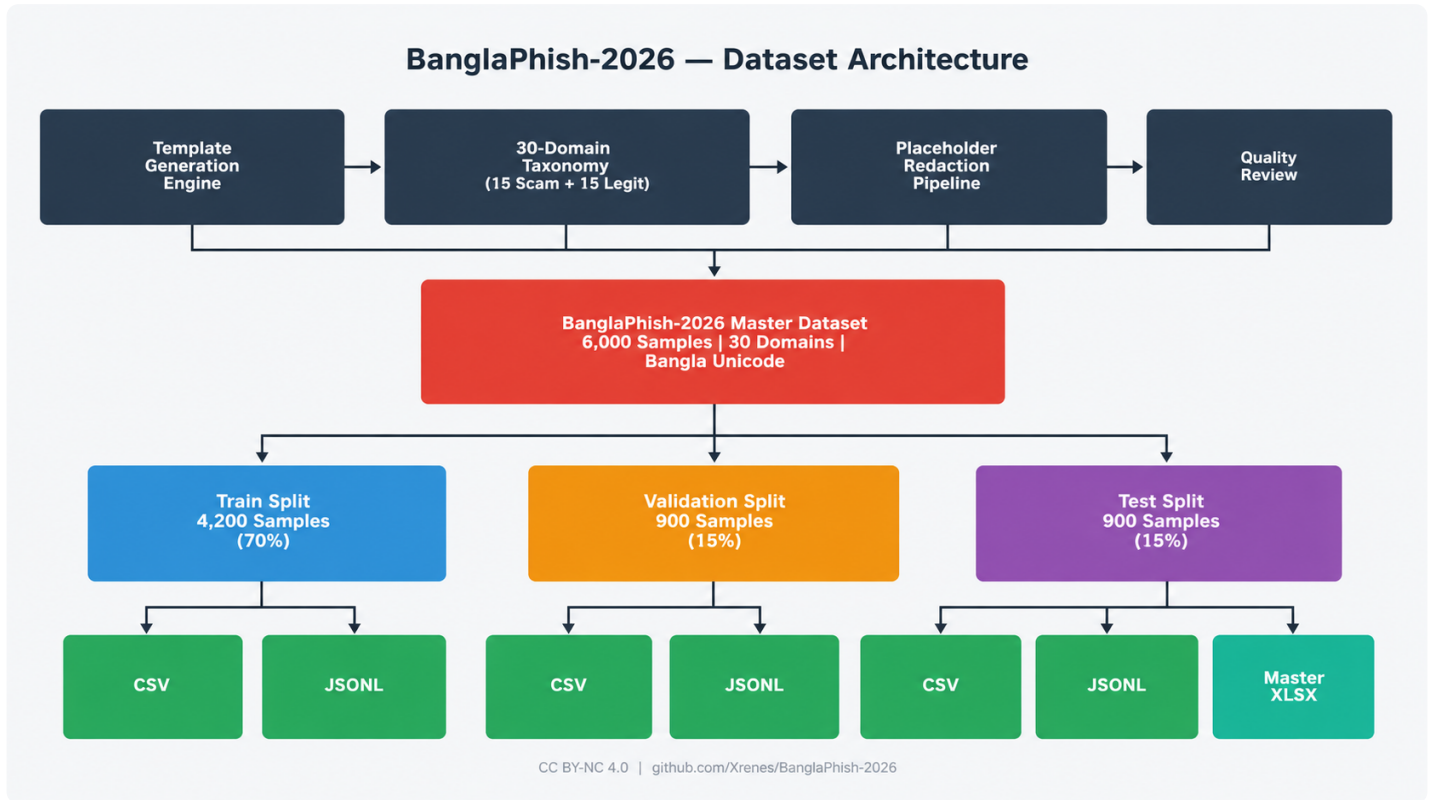


Fig. 8. End-to-end dataset construction pipeline: template generation engine → 30-domain taxonomy → placeholder redaction → quality review → 6,000-sample master dataset → stratified 70/15/15 train/validation/test splits → CSV/JSONL/XLSX distribution formats.

The dataset, baseline code (classical and transformer fine-tuning scripts), real-world supplement, and full documentation are available at:

github.com/Xrenes/BanglaPhish-2026

REFERENCES

- [1] J. Ma, L. Saul, S. Savage, and G. Voelker, "Beyond blacklists: Learning to detect malicious web sites from suspicious URLs," *KDD*, 2009.
- [2] H. Le, Q. Pham, D. Sahoo, and S. Hoi, "URLNet: Learning a URL representation with deep learning for malicious URL detection," *arXiv:1802.03162*, 2018.
- [3] R. Mohammad, F. Thabtah, and L. McCluskey, "Predicting phishing websites based on self-structuring neural network," *Neural Comput. Appl.*, 25(2):443–458, 2014.
- [4] OpenDNS, "PhishTank: A collaborative clearing house for phishing data," 2006. [Online] phishtank.com
- [5] APWG, *Phishing Activity Trends Report*, 2024. [Online] apwg.org
- [6] T. Almeida and J. Hidalgo, "Contributions to the study of SMS spam filtering," *DOCENG*, 2011.
- [7] Md. Hossain et al., "Bangla text classification using deep learning," *IJACSA*, 11(3), 2020.
- [8] J. Walonoski et al., "Synthea: An approach for generating synthetic patients," *JAMIA*, 25(3):230–238, 2018.
- [9] D. Hendrycks et al., "CUAD: An expert-annotated NLP dataset for legal contract review," *arXiv:2103.06268*, 2021.
- [10] L. Williams and R. Licata, "Using synthetic data for cybersecurity research," *Workshop on Cyber Security*, 2018.
- [11] R. Cialdini, *Influence: The Psychology of Persuasion*. Harper Business, 2006.
- [12] R. Karim et al., "BengaliNLP: A Bangla NLP toolkit," *arXiv:1912.00701*, 2019.
- [13] A. Bhattacharjee et al., "BanglaBERT: Language model pretraining for low-resource language understanding in Bangla," *Findings of NAACL*, 2022.
- [14] Md. Rahman et al., "Sentiment analysis of Bangla social media posts," *arXiv:2101.00210*, 2021.

- [15] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," *NAACL-HLT*, pp. 4171–4186, 2019.
- [16] A. Conneau et al., "Unsupervised cross-lingual representation learning at scale," *ACL*, pp. 8440–8451, 2020.
- [17] A. Vaswani et al., "Attention is all you need," *NeurIPS* 30, 2017.
- [18] T. Brown et al., "Language models are few-shot learners," *NeurIPS* 33, 2020.
- [19] I. Fette, N. Sadeh, and A. Tomasic, "Learning to detect phishing emails," *WWW*, pp. 649–656, 2007.
- [20] A. Basit et al., "A comprehensive survey of AI-enabled phishing attacks detection techniques," *Telecommunication Systems*, 76(1):139–154, 2021.
- [21] D. Sahoo, C. Liu, and S. Hoi, "Malicious URL detection using machine learning: A survey," *arXiv:1701.07179*, 2017.
- [22] P. Maneriker et al., "URLTran: Improving phishing URL detection using transformers," *MILCOM*, 2021.
- [23] T. Sarkar et al., "A comprehensive study on Bangla language models," *arXiv:2212.09660*, 2022.
- [24] F. Islam, Md. Bhuiyan, and M. Arefin, "Bangla NLP: A comprehensive survey," *IEEE Access*, 9:38999–39032, 2021.
- [25] N. Kshetri, "Cybercrime and cybersecurity in Africa," *J. Global IT Mgmt.*, 22(2):77–81, 2019.
- [26] World Bank, *Digital Financial Services and Mobile Money*. World Bank Group, 2023.
- [27] J. Yoo and K.-S. Shin, "Detecting social engineering attacks based on NLP," *IEEE Access*, 10:70225–70238, 2022.
- [28] P. Bountakas, K. Koutroumpouchos, and C. Xenakis, "HelpNets: A privacy-aware phishing email detection system," *WiseML*, 2022.
- [29] I. Ahmad et al., "Fake news detection in Bangla language using machine learning," *ICCA*, 2020.
- [30] A. Sinha and G. Shrivastava, "Identifying smishing and spam SMS using NLP," *Procedia Comput. Sci.*, 189:261–269, 2021.
- [31] N. Vishwamitra et al., "Towards automated content-based photo privacy control," *CODASPY*, 2021.
- [32] Y. Yao et al., "Phishing detection with large language models: A zero-shot evaluation," *arXiv:2305.14171*, 2023.

CITATION

```
@dataset{banglaphish_2026_v1,  
title    = {BanglaPhish-2026: A Synthetic Bangla Phishing  
            and Scam Detection Benchmark},  
author   = {Sayed Ifti Ahmed and Mst. Khadiza Akter Sammi},  
year     = {2026},  
version  = {1.0},  
license  = {CC BY-NC 4.0},  
url      = {https://github.com/Xrenes/BanglaPhish-2026}  
}
```