SINTECHCOM: Science, Technology and Communication Journal
Vol. 7, No. 1, October 2026, pp. 143-158, DOI: 10.59190/stc.v7i1.421

Adaptive and explainable machine learning for Indonesian SMS phishing detection: Few-shot prototype learning, simulated concept drift, and SHAP/LIME feature analysis

Wartono Wartono[1], Rahmad Abdillah[2,*]
[1]Department of Mathematics, UIN Sultan Syarif Kasim Riau, Pekanbaru 28293, Indonesia
[2]Department of Informatics Engineering, UIN Sultan Syarif Kasim Riau, Pekanbaru 28293, Indonesia

ABSTRACT

Phishing delivered through short text messages evolves quickly, producing scarce and imbalanced labeled data and shifting lexical patterns that challenge conventional supervised detectors. This study develops and audits a transparent detection framework on a private corpus of 15,950 Indonesian-language SMS messages, of which 13,411 unique messages (8,091 phishing, 5,320 normal) remain after removing 3,771 normalized duplicates. A class-weighted logistic regression over TF-IDF unigram–bigram features serves as the primary detector, and it is subjected to a battery of scenario evaluations: stratified five-fold cross-validation, a 2-way one/five/ten-shot episodic protocol, a controlled simulated lexical-drift stream with adaptive and static online learners, a binary class-introduction test, and an explainability pipeline built from exact linear SHAP and perturbation-based LIME. The detector attained an accuracy of 0.9806, an F1-score of 0.9838, a Matthews correlation coefficient of 0.9598, a ROC-AUC of 0.9973, and a PR-AUC of 0.9983 on a stratified hold-out. Five-fold cross-validation confirmed consistent performance with F1 scores ranging from 0.9829 to 0.9888. Episodic evaluation showed monotonic improvement from the one-shot to the ten-shot setting. Under simulated lexical drift, an adaptive online learner preserved a post-drift F1 of 0.9808 against 0.9604 for a static learner. Exact linear SHAP reproduced the decision function to within 1.8×10^{-15}, LIME achieved a local fidelity of $R^2 = 0.921$, and SHAP-based selection retaining 500 of 10,000 features (a 95% reduction) reduced F1 only modestly to 0.9741 with a mean pairwise Jaccard stability of 0.818. The findings indicate that a compact, class-weighted linear model can deliver strong, stable, and auditable SMS phishing detection, while few-shot and drift experiments delineate the boundaries within which limited-support and non-stationary performance should be interpreted.

ARTICLE INFO

Article history:

Received Sep 10, 2026
Revised Oct 15, 2026
Accepted Oct 22, 2026
Published Oct 31, 2026

Keywords:

Concept Drift
Explainable AI
Few-Shot Learning
Machine Learning
SMS Phishing Detection

Copyright (c) 2026:

Wartono Wartono, Rahmad Abdillah

License:

This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY).

***Corresponding Author**
E-mail address: rahmad.abdillah@uin-suska.ac.id

1. INTRODUCTION

Phishing has shifted from crude mass campaigns toward adaptive social-engineering that rotates domains, obfuscates lexical content, and continually introduces message variants, so that new attack patterns are frequently absent from historical training data [1,2]. Short message services combine high open rates with limited native filtering. Localized campaigns, particularly the Indonesian-language solicitations analysed here, exploit region-specific lexical cues, localized slang, and specific financial terminology (such as "rp" and "5jt") that global, English-trained filters systematically ignore. Detecting these region-specific messages is not merely a generic classification task. It is a critical operational security requirement for regional telecommunications, where failing to recognize local context carries severe financial costs for victims.

The central technical difficulty is that security data are typically scarce, imbalanced, and quickly outdated, whereas conventional supervised detectors implicitly assume access to large, balanced, and stationary labelled corpora [3,4]. When a genuinely new attack family emerges, only a handful of labelled examples may exist, and the feature distribution that a model learned during training may no longer hold. These conditions motivate learning paradigms that remain effective with few labelled examples and that can adapt as the input distribution changes.

Current machine-learning responses fall into three families. Few-shot and metric-learning approaches Siamese capsule networks, triplet networks, and mutual-centralized learning improve detection when labelled examples are limited [5-7]. Adaptive and lightweight designs address distributional change and class growth in evolving traffic [8]. Explainable-AI techniques such as SHAP and LIME expose which features drive a decision, mitigating the opacity of complex models [9-11].

Despite this progress, three limitations persist. First, few-shot evidence concentrates on network intrusion and malicious traffic rather than localized SMS phishing. Second, adaptivity is often discussed conceptually but evaluated under controlled or simulated conditions that are not always distinguished from naturally occurring drift [12,13]. Third, researchers typically deploy explainability as a passive, post-hoc interpretation tool [11,14]. This study transforms explainability into an active, auditable feature-selection mechanism. We explicitly measure the performance impact of stripping away thousands of low-attribution features. This process directly closes the methodological loop between model interpretation and mathematical dimensionality reduction. Unlike previous studies that treat these mechanisms in isolation on English-centric or network-level data, this study unifies few-shot adaptation, controlled drift evaluation, and active explainable feature selection into a single transparent pipeline specifically engineered for localized SMS phishing.

Previous studies examine few-shot learning, concept drift, and explainability as isolated problems. The primary novelty of this research is the integration of these distinct evaluation scenarios into a single framework applied to localized SMS phishing. This unified approach allows direct observation of how a transparent detector behaves under ideal supervised conditions, extreme data scarcity, and sudden distributional shifts.

Black-box models and English-centric classifiers dominate current phishing research. This study distinguishes itself by targeting the specific linguistic markers of Indonesian SMS campaigns with a fully auditable linear architecture. We construct a class-weighted logistic regression detector over TF-IDF features and subject it to a multi-scenario evaluation. The main contributions are: (1) a reproducible, high-performing, and mathematically auditable Indonesian SMS phishing detector; (2) a quantified characterization of few-shot behaviour across varying support sizes; (3) a controlled contrast of adaptive versus static learning under simulated lexical drift; and (4) an exactly validated SHAP and LIME explanation pipeline. This pipeline includes a feature-selection ablation that demonstrates massive dimensionality reduction at low performance cost. The remainder of the article details the research methods (Section 2), reports results and discusses them against prior research (Section 3), and concludes (Section 4).

2. RESEARCH METHODS

2.1. Research framework

The framework proceeds from raw SMS text through minimal preprocessing and TF-IDF feature extraction to a class-weighted linear detector, followed by a suite of scenario evaluations and complementary model explanations. The evaluation stages examine predictive performance under stratified validation, limited-support few-shot settings, controlled simulated lexical drift, and binary class introduction. SHAP and LIME analyses then characterize the contributions of textual features to model decisions, while SHAP-based feature selection assesses whether a smaller vocabulary can retain predictive performance. Together, these stages provide a structured assessment of classification accuracy, adaptability under specified experimental conditions, and interpretability, while distinguishing benchmark findings from claims requiring real-world validation. Figure 1 (confusion matrix and curves) and Figure 2 (feature weights) summarize the core detector; Figure 3 and Figure 4 characterize stability and thresholds; Figure 5, Figure 6, and Figure 7 report the few-shot, drift, and class-introduction scenarios; and Figure 8 and Figure 9 report explanation and feature selection. Each stage reuses a fixed random

seed to support repeatability and consistent comparisons across experimental scenarios. The pipeline deliberately keeps the learner simple and linear so that its linear SHAP explanations can be computed exactly, while LIME provides complementary local approximations of model behavior.

2.2. Dataset

The corpus comprises 15,950 Indonesian-language SMS messages labelled as phishing (1) or normal (0), with a raw class distribution of 10,316 phishing and 5,634 normal messages, confirming pronounced class imbalance. A normalization key that masks digits and URLs was used to detect near-duplicates, which numbered 3,771 (23.64%). To adopt a conservative evaluation, exact and normalized duplicates were removed, leaving 13,411 unique messages (8,091 phishing, 5,320 normal). All subsequent primary experiments use the deduplicated corpus, and this reduction together with its verified reason is reported rather than hidden, consistent with dataset-integrity requirements. Table 1 records the composition before and after deduplication.

Table 1. Dataset composition before and after deduplication.

Stage	Total	Normal (0)	Phishing (1)
Raw (after cleaning)	15,950	5,634	10,316
Normalized duplicates detected	3,771 (23.64%)	—	—
After deduplication (primary)	13,411	5,320	8,091

2.3. Data Preprocessing

Preprocessing is intentionally minimal and non-destructive: text is cast to string, stripped of leading and trailing whitespace, and lowercased, while tokenization is delegated to the vectorizer. These operations standardize basic text formatting while retaining the wording available for feature extraction. For duplicate detection only, an auxiliary normalization replaces numeric tokens with a <num> marker and URLs with a <url> marker, allowing messages with similar wording but different numbers or links to be grouped under a common normalization key. This normalized field is used solely to identify duplicates and is not fed to the classifier; consequently, numeric and URL content is not replaced by these markers in the classification input. No stemming, lemmatization, or stopword removal was applied, preserving lexical information that may contribute to distinguishing phishing from normal messages. This separation between duplicate detection and classification preprocessing makes the data-processing procedure explicit and ensures that the reported pipeline accurately reflects the executed steps.

2.4. Feature Representation

The primary detector represents each message with a TF-IDF vectorizer using unigrams and bigrams (ngram_range = (1,2)), a minimum document frequency of two, a maximum document frequency of 0.95, sublinear term-frequency scaling, Unicode accent stripping, and a vocabulary capped at 10,000 features. TF-IDF weighting follows the classical term-weighting formulation that assigns higher weight to terms frequent in a message but rare across the corpus [15]. An earlier exploratory pipeline used 5,000 features on the non-deduplicated data; its results are disclosed in Section 4.2 but are not treated as the primary evidence.

2.5. Supervised Baseline and Proposed Detector

The classifier is a Logistic Regression trained with the liblinear solver, up to 3,000 iterations, a fixed seed, and class_weight="balanced" to counteract the class imbalance without resampling. Conventional TF-IDF classifiers prioritize raw predictive accuracy and frequently ignore interpretability. We employ a class-weighted Logistic Regression specifically because its linear decision function admits an exact, mathematically verifiable SHAP decomposition [9]. This design enables the model's exact token-level contributions to the linear decision function to be examined, making the influence of individual textual features explicit and traceable. It aligns the predictive architecture directly with the explainability objective, allowing model outputs to be analyzed through feature weights and additive attributions. These explanations describe how the classifier combines textual evidence, rather than es-

tablishing causal relationships or human-like reasoning, and distinguish the framework from text-classification pipelines whose internal decision mechanisms are less directly accessible.

2.6. Cross-Validation and Threshold Analysis

Beyond an 80/20 stratified hold-out, generalization stability is assessed with stratified five-fold cross-validation in which the entire pipeline (including TF-IDF fitting) is refit inside each fold. To characterize the operating point, the predicted phishing probability is swept across thresholds from 0.05 to 0.95, recording precision, recall, F1, and the false-positive and false-negative rates, and the F1-maximizing threshold is identified.

2.7. Few-Shot Learning Protocol

Few-shot behaviour is evaluated with a genuinely episodic 2-way protocol rather than by merely shrinking the training set. In each of five folds, a background partition is used only to fit the TF-IDF representation, and support–query episodes are drawn exclusively from the evaluation fold; class prototypes are computed as the mean support-set vector per class, and queries are classified by cosine similarity to prototypes, mirroring prototypical-network logic [16]. One hundred episodes are sampled for each support size $K \in \{1,5,10\}$, and bootstrap 95% confidence intervals are reported. This design is restricted to the binary case and is described as binary few-shot only.

2.8. Adaptive Learning Under Simulated Concept Drift

Drift is a controlled simulation, not naturally occurring temporal drift. After a change point in the stream, only phishing messages undergo deterministic lexical obfuscation (character substitutions such as $a \rightarrow 4$, $e \rightarrow 3$, and URL masking to a generic token), leaving the source dataset unmodified. Two online SGDClassifier learners with log-loss are compared prequentially: a static learner that is never updated after the change point and an adaptive learner updated per batch, with a Page-Hinkley detector supplying a reset signal from the recent-error window [12]. Drift is introduced through a controlled synthetic simulation designed to evaluate model responses to predefined changes in the lexical patterns of phishing messages. It does not represent naturally occurring temporal drift or independently observed changes in real-world messaging behavior. The artificial change point and deterministic obfuscation procedure provide a reproducible experimental setting for comparing adaptive and static learners under the specified conditions. However, this setting does not capture the full diversity, timing, or unpredictability of evolving phishing campaigns. Readers should therefore interpret the subsequent findings as evidence limited to the simulated scenario, rather than as proof of robustness or sustained performance under real-world, unpredictable concept drift.

2.9. Binary Class Introduction

To probe behaviour when a class appears only after deployment, the memory initially contains only a normal-class prototype, and a phishing prototype is later introduced from $K \in \{1,5,10\}$ examples using a HashingVectorizer representation, without full retraining. Performance is recorded before and after class introduction across multiple random seeds to examine variability and consistency under the specified experimental conditions. This protocol represents a strictly binary class-introduction scenario, assessing how the learner responds when a previously unavailable class is introduced within a two-class task. It does not evaluate multiclass class-incremental learning, repeated expansion of the label space, or retention of previously learned classes across successive introductions. Accordingly, the results should be interpreted within the boundaries of this binary evaluation. Readers must not generalize these findings to continuous learning across multiple new classes or regard them as evidence that the model prevents catastrophic forgetting in broader incremental-learning settings.

2.10. Explainable AI: SHAP and LIME

For the linear detector, SHAP values on the log-odds decision function are computed exactly as the product of each feature's coefficient and its value centered relative to the background mean, under the independent-feature formulation. Each attribution therefore quantifies a feature's contribution relative to the selected baseline, with positive and negative values indicating movement toward phishing and normal predictions, respectively. The baseline and summed feature attributions are reconstructed and compared against the model's decision function to verify numerical agreement. This check estab-

lishes additive consistency on the log-odds scale; it does not establish causal importance or account for dependencies among correlated textual features. LIME is implemented for text with token-presence features, token-deletion perturbations, cosine-distance locality weighting, and a weighted Ridge surrogate, following the local-surrogate principle [10]. Global views are formed by aggregating feature attributions across the analyzed messages to summarize overall patterns of model reliance on textual features. These summaries complement instance-level explanations by identifying features that contribute prominently across multiple predictions, rather than only within individual messages. Explanations are described using contribution and association language rather than causal language, because attribution values characterize the model's behavior relative to a selected baseline and do not demonstrate that particular words cause phishing activity. Their interpretation remains conditional on the fitted model, feature representation, background data, and messages included in the analysis.

2.11. Explainable Feature Selection and Ablation

SHAP-based feature selection ranks features using SHAP magnitudes computed on training data only, retains the top 500 of 10,000 features, and retrains the classifier on the reduced representation. The full-feature and reduced-feature models are compared under identical hold-out conditions, using the same test partition and evaluation metrics to assess the predictive effect of feature reduction. Feature-set stability is quantified as the mean pairwise Jaccard index across five bootstrap training resamples, measuring the consistency of selected features under changes in the sampled training observations. Higher Jaccard values indicate greater overlap between feature subsets, whereas lower values suggest stronger dependence on the particular training sample. Because the test set is never used for feature selection, this constitutes a leakage-controlled hold-out ablation with respect to the selection procedure. The comparison therefore assesses both predictive performance and feature-selection stability within the specified experimental setting.

2.12. Evaluation Metrics

Reported metrics are accuracy, precision, recall, F1-score, the Matthews correlation coefficient (MCC), ROC-AUC, and PR-AUC (average precision). MCC is emphasized because it yields a high score only when all four confusion-matrix quadrants are handled well and is more reliable than accuracy or F1 on imbalanced data [17]. PR-AUC is reported alongside ROC-AUC because it provides a complementary view of positive-class detection under class imbalance, emphasizing the trade-off between precision and recall. Reporting both metrics helps distinguish overall ranking performance from the ability to identify phishing messages while limiting false positives. For the single hold-out run, no claim of statistical significance is made, and observed performance differences are treated as descriptive results for that partition. Few-shot and class-introduction scenarios report bootstrap confidence intervals across episodes and random seeds to characterize variability within their respective evaluation protocols. These intervals reflect uncertainty in the sampled experimental outcomes and should not be interpreted as guarantees of performance on independent real-world data.

2.13. Implementation Environment

Experiments were implemented in Python using scikit-learn (TfidfVectorizer, LogisticRegression, SGDClassifier, HashingVectorizer, StratifiedKFold), NumPy, pandas, SciPy, Matplotlib, and seaborn, with a fixed random seed of 42 for the main pipeline and dedicated seeds for episodic and introduction experiments. Specific Python and library version numbers, hardware specifications, and runtime were not captured in the executed notebook and are therefore explicitly reported as missing reproducibility information rather than inferred or retrospectively estimated. These omissions limit exact replication of the computational environment and verification of execution efficiency.

3. RESULTS AND DISCUSSIONS

3.1. Dataset Characteristics

The raw corpus contains 15,950 messages (10,316 phishing; 5,634 normal), and normalized-duplicate detection flagged 3,771 messages (23.64%). After removing exact and normalized duplicates, 13,411 unique messages remain (8,091 phishing; 5,320 normal), preserving a phishing-majority

imbalance of roughly 60:40 (Table 1). The substantial duplicate fraction justifies the conservative decision to deduplicate before the primary evaluation, since duplicates could otherwise inflate apparent performance.

3.2. Primary Detector Performance

On the stratified hold-out of the deduplicated corpus, the class-weighted Logistic Regression achieved an accuracy of 0.9806, a precision of 0.9900, a recall of 0.9778, an F1-score of 0.9838, an MCC of 0.9598, a ROC-AUC of 0.9973, and a PR-AUC of 0.9983 (Table 2). Figure 1 presents the confusion matrix together with the ROC and precision–recall curves, and Figure 2 shows the most influential positive (phishing-associated) and negative (normal-associated) TF-IDF feature weights of the linear model. For completeness, the earlier exploratory pipeline (5,000 features, non-deduplicated 15,950 rows) produced an accuracy of 0.9828 and a ROC-AUC of 0.9980; because it used a less conservative protocol, it is disclosed here but is not the primary result.

Table 2. Hold-out performance of the primary detector (deduplicated, 10,000 TF-IDF features).

Accuracy	Precision	Recall	F1-score	MCC	ROC-AUC	PR-AUC
0.9806	0.9900	0.9778	0.9838	0.9598	0.9973	0.9983

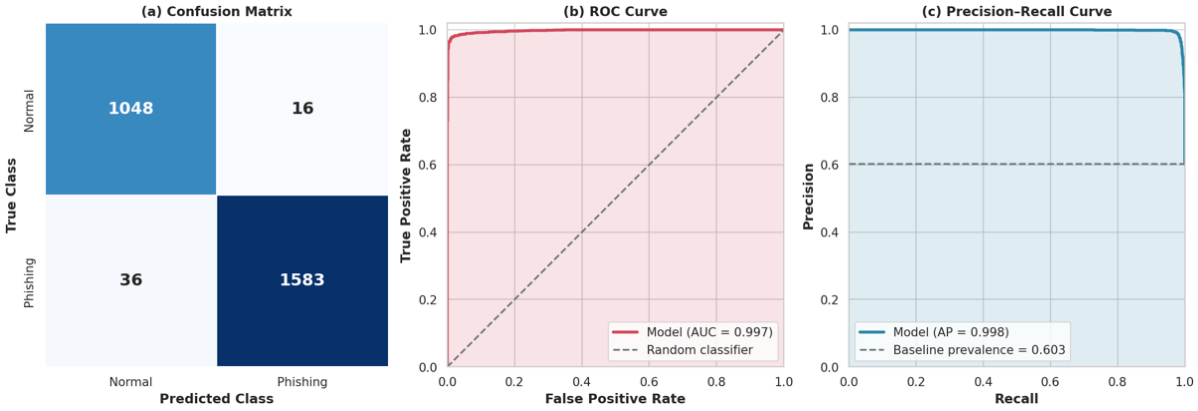


Figure 1. Confusion matrix, ROC curve, and precision–recall curve of the primary detector.

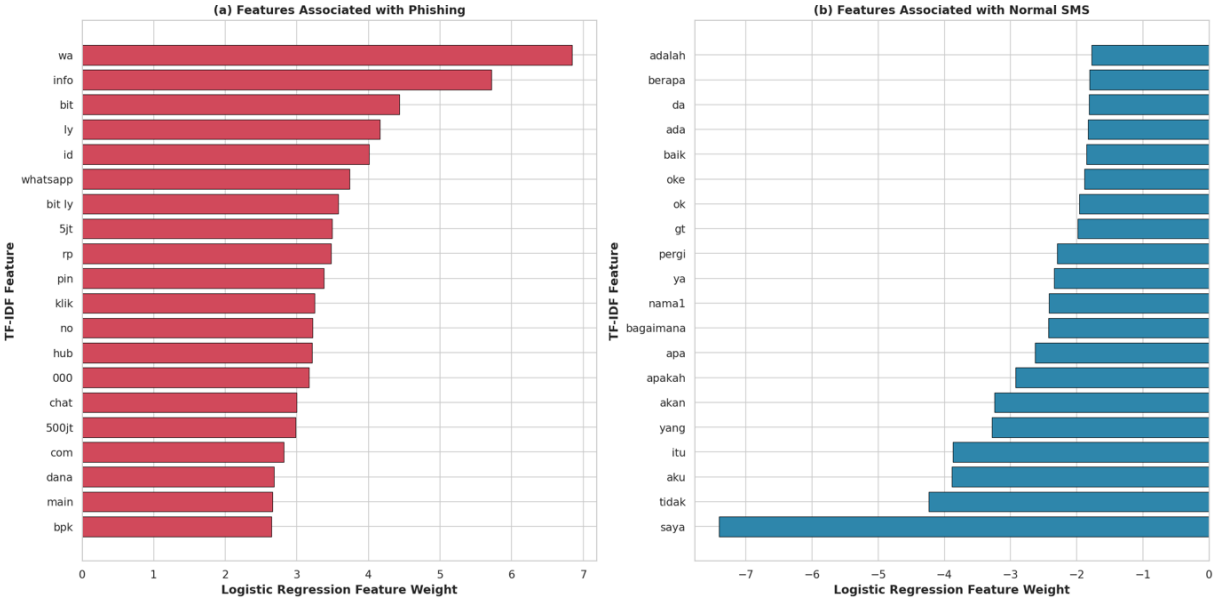


Figure 2. Top TF-IDF features associated with phishing vs. normal (logistic regression weights).

3.3. Cross-Validation Stability and Threshold Behaviour

Stratified five-fold cross-validation produced accuracies of 0.9795 – 0.9866, F1-scores of 0.9829 – 0.9888, and MCC values of 0.9573 – 0.9721 across folds, with ROC-AUC and PR-AUC consistently at or above 0.9975 and 0.9984 respectively (Table 2), and Figure 3 visualizes this narrow performance spread. The threshold sweep identified an F1-maximizing decision threshold of 0.49, at which precision was 0.9900, recall 0.9784, F1 0.9842, the false-positive rate 0.015, and the false-negative rate 0.0216 (Table 3), and Figure 3 shows the precision, recall, F1, and false-positive-rate curves across the threshold range. The near-default optimal threshold indicates that the model's probability outputs are already well-aligned with the classification objective.

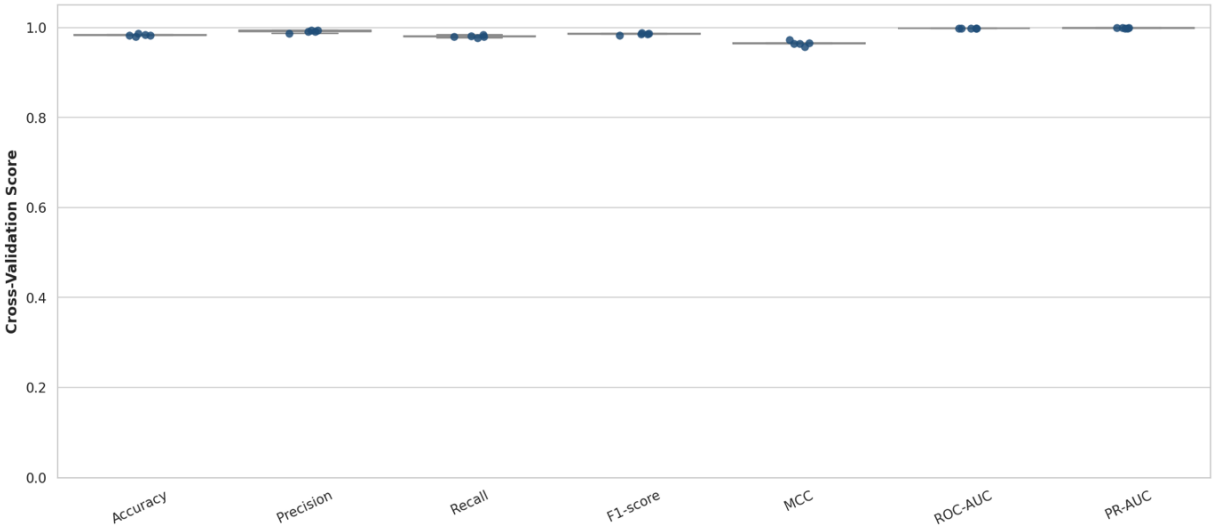


Figure 3. Performance-stability boxplot across stratified five-fold cross-validation.

Table 3. F1-maximizing decision threshold (hold-out).

Threshold	Precision	Recall	F1-score	False positive rate	False negative rate
0.49	0.9900	0.9784	0.9842	0.015	0.0216

3.4. Few-shot performance

The 2-way episodic evaluation showed clear, monotonic gains with additional support examples (Table 4). Mean accuracy rose from 0.6907 in the one-shot setting to 0.8533 at five-shot and 0.8950 at ten-shot; mean F1 rose from 0.6288 to 0.8534 to 0.9034; and mean MCC rose from 0.4187 to 0.7151 to 0.7960, with ROC-AUC improving from 0.7535 to 0.9536. The one-shot condition exhibited the largest variability (accuracy standard deviation 0.1147; MCC standard deviation 0.2280), which contracted substantially by ten shots. Figure 4 depicts these trends across K, indicating that under this episodic protocol the prototype classifier requires several support examples before its representation stabilizes.

Table 4. Binary class introduction (mean; multi-seed, K ∈ {1,5,10}).

K-shot	Stage	Accuracy	Precision	Recall	F1-score	MCC	ROC-AUC	PR-AUC
1	Before introduction	0.500	0.000	0.000	0.000	0.000	0.8188	0.7439
1	After introduction	0.635	0.9530	0.284	0.4255	0.3749	0.8589	0.8422
5	Before introduction	0.500	0.000	0.000	0.000	0.000	0.8188	0.7439
5	After introduction	0.800	0.9612	0.624	0.7537	0.6411	0.9350	0.9394
10	Before introduction	0.500	0.000	0.000	0.000	0.000	0.8188	0.7439
10	After introduction	0.869	0.9626	0.768	0.8538	0.7540	0.9589	0.9633

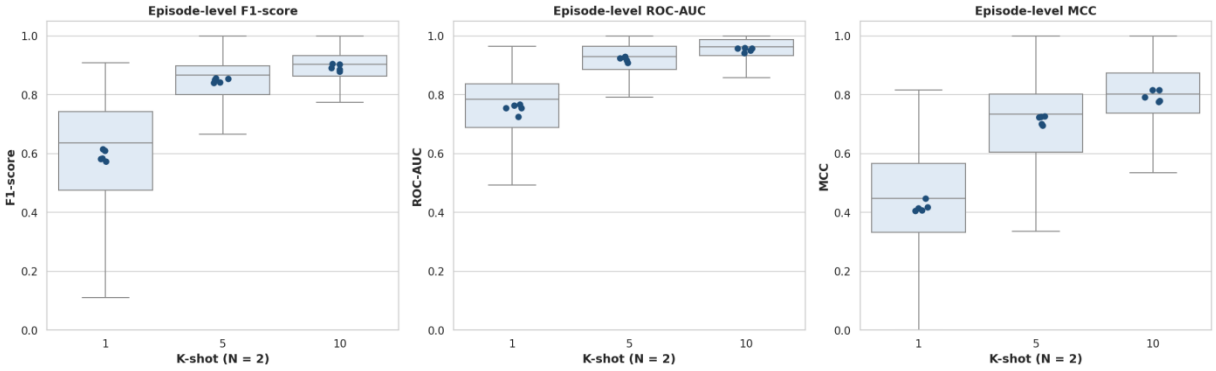


Figure 4. Few-shot performance across K ∈ {1,5,10}.

3.5. Performance Under Simulated Concept Drift

Under the simulated lexical-drift stream, the Page-Hinkley detector signalled a change near stream position 5,386, close to the injected change point at 5,000 (Table 5). Before the change point the adaptive and static learners performed almost identically (F1 0.9821 and 0.9817). After the change point, the static learner fails because post-change obfuscation pushes phishing messages entirely out of the previously learned lexical boundaries, degrading its recall to 0.9421 and F1 to 0.9604. Conversely, the adaptive learner successfully ingests these new obfuscation patterns on the fly, retaining a recall of 0.9859 and an F1 of 0.9808. This rapid recovery confirms that prequential updating effectively neutralizes sudden, deliberate lexical evasion tactics. A strict boundary must be stated. The evaluated drift is deterministic, controlled obfuscation of phishing messages after an artificial change point. This simulation does not replicate naturally occurring temporal drift. The results demonstrate responsiveness to a specific synthetic shift only. These findings provide no evidence of performance under real-world, unpredictable drift. Generalization to genuine operational drift requires future empirical testing (see Figure 5).

Table 5. Performance before and after simulated lexical drift (adaptive vs. static online SGD).

Phase	Model	Accuracy	Precision	Recall	F1-score	MCC	ROC-AUC	PR-AUC
Pre-drift	Adaptive SGD	0.9786	0.9829	0.9817	0.9821	0.9554	0.9965	0.9977
Pre-drift	Static SGD	0.9782	0.9855	0.9785	0.9817	0.9546	0.9962	0.9976
Post-drift (simulated)	Adaptive SGD	0.9773	0.9764	0.9859	0.9808	0.9526	0.9983	0.9989
Post-drift (simulated)	Static SGD	0.9539	0.9812	0.9421	0.9604	0.9066	0.9928	0.9948

Note: Page-Hinkley detection fired near stream position 5,386 (injected change point at 5,000).

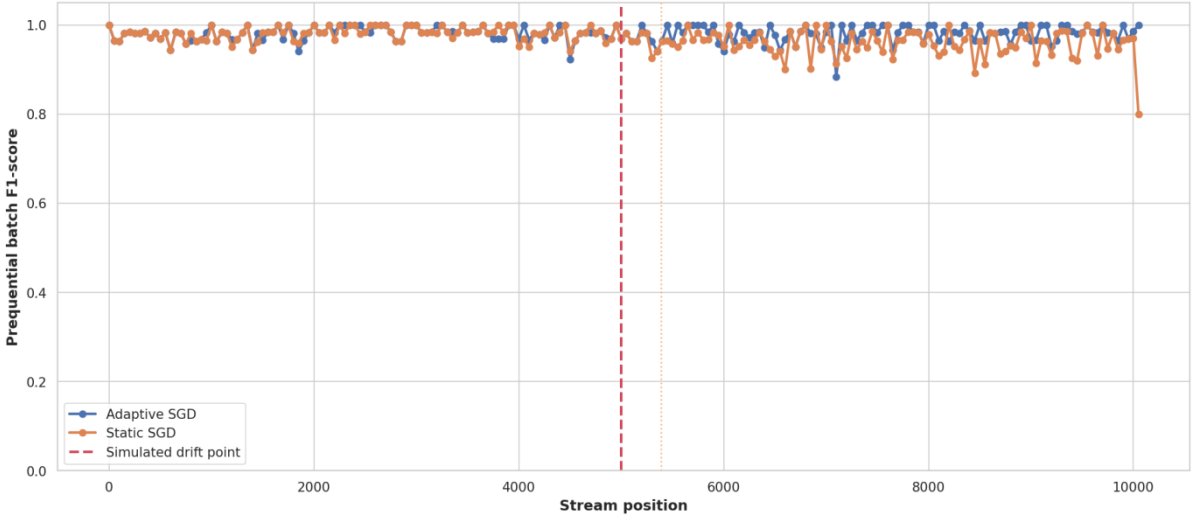


Figure 5. Prequential accuracy of adaptive vs. static learner across the simulated-drift stream.

3.6. Binary Class Introduction

Before the phishing prototype was introduced, the memory recognized only the normal class, yielding a recall of 0 for phishing and an accuracy fixed at 0.500 across all K (Table 4). After introduction, mean accuracy increased with support size to 0.635, 0.800, and 0.869 for K = 1, 5, and 10 respectively, with corresponding mean F1 of 0.4255, 0.7537, and 0.8538 and precision remaining high (≥ 0.953). Figure 6 shows this before/after contrast. The pattern demonstrates that a newly introduced class is detected increasingly well as more support examples become available, while the persistently high precision indicates conservative phishing predictions.

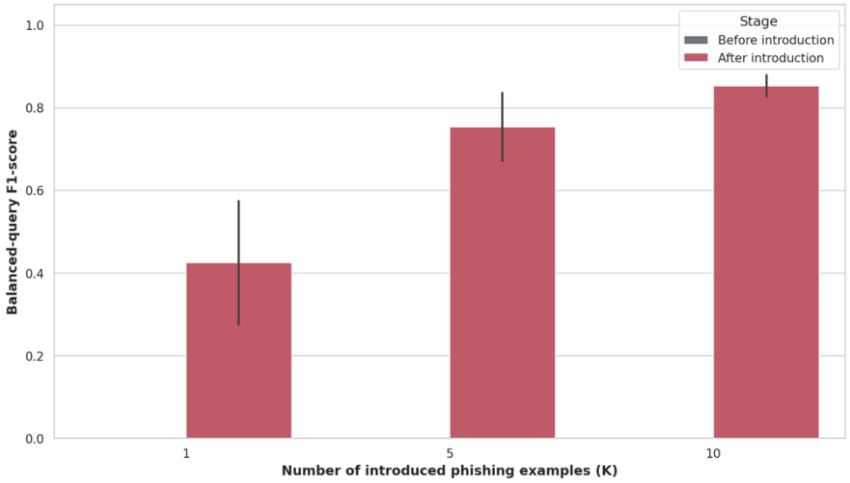


Figure 6. Binary class-introduction performance (before/after, by K).

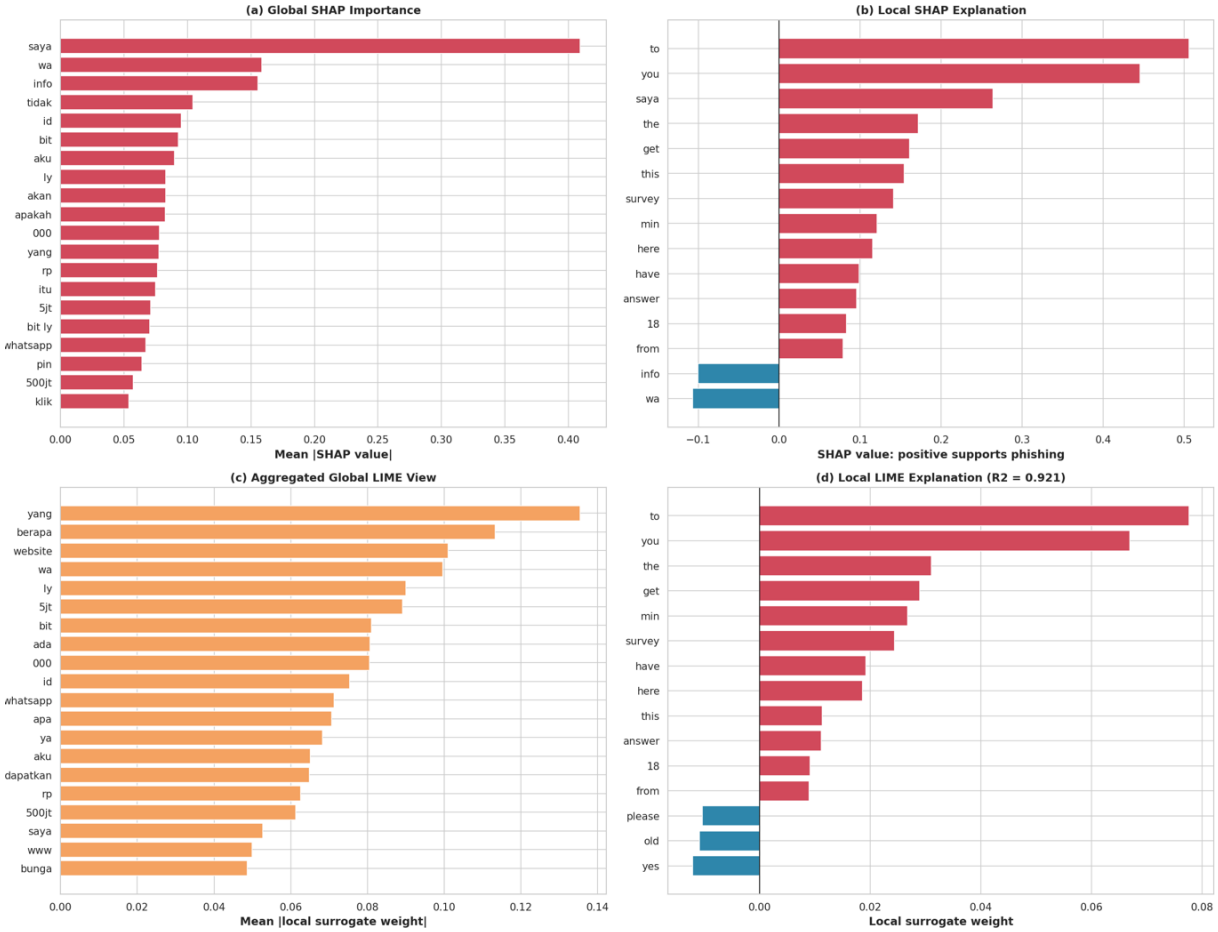


Figure 7. Global/local SHAP and aggregated/local LIME attributions.

3.7. SHAP and LIME Interpretation

The exact linear-SHAP reconstruction matched the model's decision function to a maximum additive-validation error of 1.776×10^{-15}, confirming that the attributions are exact rather than approximate for this linear model. The LIME surrogate achieved a local fidelity of $R^2 = 0.921$ for the explained instance. Figure 7 presents the global SHAP importance ranking, a local SHAP explanation, an aggregated global LIME view, and a local LIME explanation, with positive attributions supporting the phishing class. The agreement between an exact global method and a locally faithful surrogate provides converging evidence about which tokens drive predictions.

3.8. Explainable Feature Selection and Ablation

Retaining only the top 500 SHAP-ranked features a 95% reduction from 10,000 reduced accuracy from 0.9806 to 0.9691 and F1 from 0.9838 to 0.9741, with MCC moving from 0.9598 to 0.9360 and ROC-AUC from 0.9973 to 0.9950 (Table 6). Figure 8 contrasts the full and reduced configurations. The mean pairwise Jaccard stability of the selected feature set across five bootstrap resamples was 0.818, indicating that the SHAP-selected features are largely consistent under resampling. The ablation demonstrates that most predictive signal concentrates in a small, stable subset of features at a modest performance cost. In practical terms, this high Jaccard stability (0.818) proves that the model relies on the exact same core vocabulary of phishing keywords regardless of the specific training sample. Network defenders can extract this stable vocabulary to build static firewall rules or highly efficient regex filters, knowing the selected terms are not random artifacts of a single data split.

Table 6. SHAP-based explainable feature selection and ablation.

Configuration	#Features	Accuracy	Precision	Recall	F1-score	MCC	ROC-AUC	PR-AUC	Feature reduction
All TF-IDF features	10,000	0.9806	0.9900	0.9778	0.9838	0.9598	0.9973	0.9983	0.0%
Top-500 SHAP features	500	0.9691	0.9836	0.9648	0.9741	0.9360	0.9950	0.9968	95.0%

Note: Mean pairwise Jaccard stability of the selected feature set across five bootstrap resamples = 0.818.

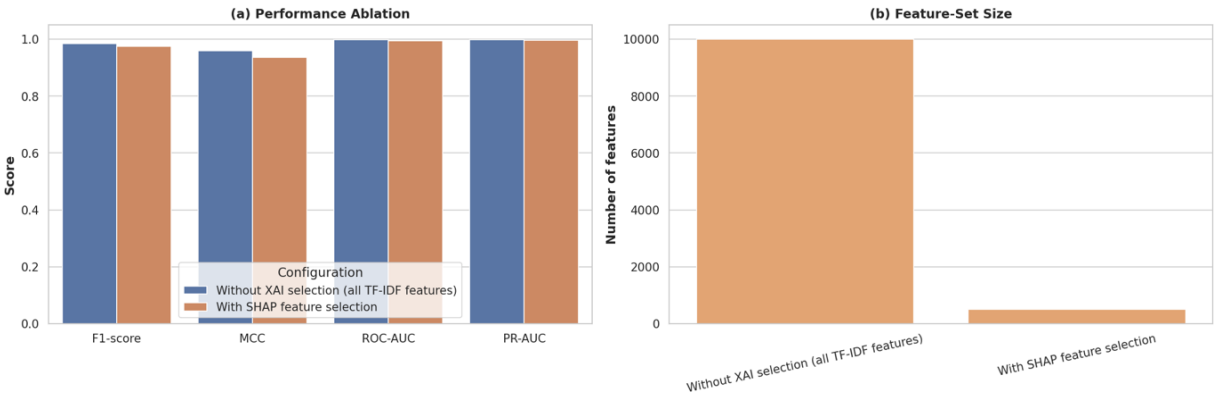


Figure 8. Full vs. SHAP-selected feature-set performance.

3.9. Error Characterization

At the F1-optimal threshold the false-positive rate was 0.015 and the false-negative rate 0.0216 (Table 7), meaning that under this operating point normal messages are rarely flagged while a small fraction of phishing messages are missed. Because the two error types carry different operational costs a missed phishing message versus an incorrectly flagged legitimate message the threshold curves in Figure 4 make the achievable trade-offs explicit without requiring model retraining.

Table 7. Stratified five-fold cross-validation metrics.

Fold	Accuracy	Precision	Recall	F1-score	MCC	ROC-AUC	PR-AUC
1	0.9829	0.9943	0.9771	0.9857	0.9646	0.9985	0.9991
2	0.9832	0.9913	0.9808	0.9860	0.9651	0.9980	0.9987
3	0.9866	0.9944	0.9833	0.9888	0.9721	0.9980	0.9988
4	0.9795	0.9869	0.9790	0.9829	0.9573	0.9975	0.9984
5	0.9825	0.9912	0.9796	0.9854	0.9636	0.9979	0.9987

3.10. Principal Findings and Comparison with Previous Research

The principal finding is that a compact, class-weighted linear detector attains strong and stable performance on a real imbalanced Indonesian SMS corpus, with a hold-out F1 of 0.9838 and MCC of 0.9598 (Table 2) and tightly clustered five-fold metrics (Table 7). This is broadly consistent with the wider security-detection literature reporting high effectiveness for well-tuned classifiers on curated benchmarks [1,2] and with explainable intrusion-detection work that likewise reports strong headline metrics [11]. However, direct numerical comparison with those studies is inappropriate because they use different datasets, languages, feature representations, and problem definitions network intrusion or malicious traffic rather than Indonesian SMS text so the present result should be read as strong within its own evaluation rather than as evidence of superiority. One plausible reason for the high linear-model performance is that phishing SMS in this corpus contains highly discriminative lexical markers (obfuscated tokens, monetary and prize cues) that TF-IDF captures effectively [15], a property that also underpins the successful 95% feature reduction reported in Section 4.8. The finding contributes a reproducible, auditable baseline for a comparatively under-studied localized phishing setting, but its interpretation is bounded by the use of a single corpus and a single hold-out split.

3.11. Interpretation of Model Performance

The convergence of a near-default F1-optimal threshold of 0.49 (Table 7) with high ROC-AUC (0.9973) and PR-AUC (0.9983) indicates that the model's probability estimates are well-calibrated to the task and that its discrimination is not an artifact of threshold tuning. Emphasizing MCC (0.9598) is deliberate. Accuracy and F1 scores often appear over-optimistic on imbalanced datasets. MCC provides a trustworthy single-number summary that requires correct handling of all four confusion-matrix quadrants [17]. In practical terms, this high MCC indicates that the model successfully traps the majority of phishing attempts without quarantining essential legitimate communications. The consistency of MCC across folds proves that the performance is an accurate reflection of minority-class detection rather than an artifact of majority-class dominance. The consistency of MCC across folds (0.9573 – 0.9721) therefore strengthens confidence that the performance is not driven by the majority class. A methodological contrast with much deep-learning security work is that this study attains comparable headline metrics with a linear model whose decisions are exactly explainable, trading representational capacity for transparency; the implication is that, for lexically distinctive SMS phishing, model complexity may not be the binding constraint. This interpretation is strictly bounded by the absence of external validation. The strong calibration and generalization metrics observed here apply exclusively to this specific Indonesian SMS corpus. They must not be assumed to transfer to other datasets, languages, or future time periods without independent empirical verification.

3.12. Few-Shot Learning Findings

Episodic evaluation revealed monotonic improvement from one-shot to ten-shot, with mean F1 rising from 0.6288 to 0.9034 and one-shot variance markedly higher than ten-shot variance (Table 8). This trend is consistent with the foundational expectation of prototype-based few-shot learning, in which additional support examples yield more reliable class prototypes and reduce prediction instability [16], and it echoes security-oriented few-shot studies reporting improved recognition of scarce classes as support grows [5-7]. Direct numerical comparison with those studies is not warranted, because they employ different datasets, N-way configurations, backbones (capsule, triplet, or mutual-centralized architectures), and episodic sampling, whereas the present protocol is a 2-way cosine-prototype classifier over TF-IDF. A single TF-IDF support vector captures all the idiosyncratic lexical noise of one specific message. This noise produces a highly unstable mathematical prototype. Averag-

ing five or ten support vectors neutralizes the individual noise. The system isolates the true linguistic centroid of the phishing campaign. This centroid stabilization directly drives the substantial performance jump observed in the five-shot and ten-shot settings. These results prove that prototype-style adaptation is operationally viable for emerging SMS phishing campaigns. Security operators do not need to await hundreds of reports to retrain a full classification model. Once analysts capture five to ten examples of a novel attack, the system generates a reliable detection prototype immediately. Analysts must treat the one-shot regime cautiously. The binary 2-way scope limits extrapolation to many-way settings.

Table 8. 2-way {1,5,10}-shot episodic performance (mean ± std; 100 episodes × 5 folds).

K-shot	Accuracy (mean)	Acc std	Precision (mean)	Recall (mean)	F1 (mean)	MCC (mean)	ROC-AUC (mean)	PR-AUC (mean)
1	0.6907	0.1147	0.8023	0.5100	0.6288	0.4187	0.7535	0.7492
5	0.8533	0.0701	0.8790	0.8307	0.8534	0.7151	0.9219	0.9270
10	0.8950	0.0617	0.9104	0.8847	0.9034	0.7960	0.9536	0.9573

Note: F1 mean values are derived from the episode summary; per-request no significance test is claimed. 95% CIs are available in the notebook output (e.g., accuracy 95% CI: 0.0225, 0.0137, 0.0121 for K = 1, 5, 10).

3.13. Adaptation Under Simulated Concept Drift

Under simulated lexical drift, the adaptive learner preserved a post-drift F1 of 0.9808 while the static learner fell to 0.9604, with the Page-Hinkley signal firing near the injected change point (Table 5). This behaviour aligns with the concept-drift literature, which holds that prequential adaptation with drift-triggered updates should outperform frozen models on non-stationary streams [12], and with security reviews arguing that adaptivity is a central unmet need [8,13]. A strict boundary must be stated. The drift evaluated here is a deterministic, controlled obfuscation of phishing messages after an artificial change point. This simulation does not replicate naturally occurring temporal drift. The results demonstrate responsiveness to a specific synthetic shift only. These findings provide no evidence of model performance under real-world, unpredictable drift conditions. Generalization to genuine operational drift requires future empirical testing. The static learner fails because post-change obfuscation pushes phishing messages entirely out of the previously learned lexical boundaries. This shift degrades recall to 0.9421. The adaptive learner successfully ingests these new obfuscation patterns on the fly. It recovers recall to 0.9859. This rapid recovery confirms that prequential online updating effectively neutralizes sudden, deliberate lexical evasion tactics. Consequently, the appropriate scientific claim is limited to the simulated setting, and generalization to genuine drift remains to be tested.

3.14. Explainable AI and Feature-Level Interpretation

The explainability results provide two forms of converging evidence: an exact linear-SHAP decomposition with a reconstruction error of 1.776×10^{-15} and a locally faithful LIME surrogate with $R^2 = 0.921$ (Figure 8). The exactness of linear SHAP is not incidental but a theoretical property of additive attribution for linear decision functions [9], and the LIME surrogate operationalizes the local-approximation principle for text [10]; their agreement on influential tokens strengthens confidence in the attributions. This is consistent with security XAI studies that use SHAP and LIME to surface influential features and improve auditability [11,14], though those works apply the methods to different model families and data, so only conceptual rather than numerical comparison is appropriate. These attributions strictly quantify mathematical feature influence within the decision boundary. They establish statistical associations with the output classification. Analysts must not interpret these weights as causal evidence of real-world phishing intent. A high SHAP value confirms only that the model relies heavily on a specific token. It does not establish that the token itself dictates malicious behavior.

The practical implication is that analysts observe exactly which tokens drive a decision. High LIME fidelity ($R^2 = 0.921$) and zero-error linear SHAP reconstruction validate the mathematics of the explanation pipeline. They confirm only that the explainer perfectly maps the internal model logic. They provide zero guarantee that the model's logic aligns with real-world semantic validity. Human

review remains mandatory to verify if the mathematically faithful explanation actually represents a logically sound security decision

3.15. Explainable Feature Selection and Ablation

The ablation showed that retaining 500 of 10,000 SHAP-ranked features preserved most performance, with F1 declining only from 0.9838 to 0.9741 and MCC from 0.9598 to 0.9360 (Table 6, Figure 9), while feature-set stability reached a mean Jaccard of 0.818. This closes the loop that much prior work leaves open, where explanation is computed but its use as a selection criterion is not measured [11,14]. The result is consistent with the broader principle that a small, high-weight subset of terms can carry most retrieval and classification signal in text [15], and the high Jaccard stability suggests the selected subset is not an artifact of a single split. A plausible explanation for the small performance drop is that removing 95% of features discards mostly low-signal terms while retaining the discriminative lexical markers of phishing. This 95% reduction generates substantial practical advantages for high-throughput SMS gateways. Stripping 9,500 low-signal features dramatically cuts the model's memory footprint and accelerates vectorization latency. These reductions allow the system to process high-volume traffic efficiently. The boundary remains that selection was evaluated on one corpus under a single hold-out. The optimal retained-feature count may differ in other deployment environments.

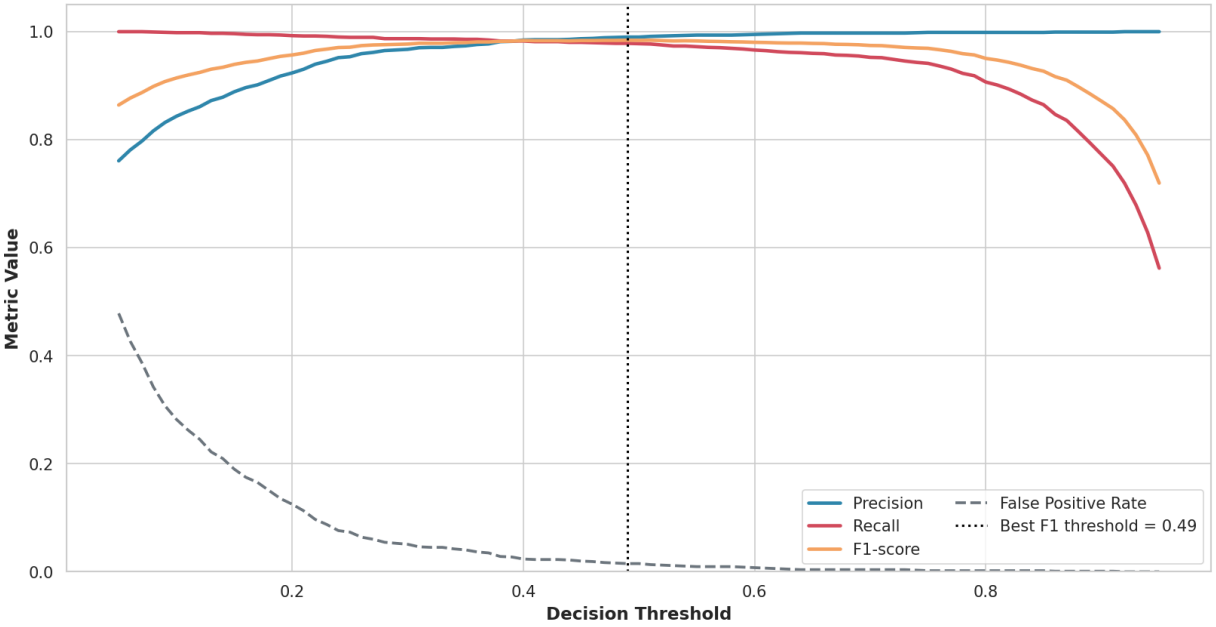


Figure 9. Precision, recall, F1, and false-positive rate vs. decision threshold.

3.16. Practical and Scientific Implications

The framework offers direct practical relevance for Indonesian telecommunications and security operations. Regional phishing campaigns exploit localized linguistic cues that global filters often miss. The combination of high precision (0.9900), a low false-positive rate (0.015), and explicit threshold curves allows network operators to tune the detector to minimize disruption to legitimate local messaging. Security workflows increasingly require human-auditable justifications for automated blocking decisions [11,14]. In a practical workflow, a Level 1 security operations center (SOC) analyst receives an alert flagged by the F1-optimized probability threshold. The analyst immediately reviews the integrated local SHAP and LIME visual attributions to verify the exact tokens that triggered the alert. If the highly weighted evidence matches known local obfuscation tactics or monetary lures, the analyst authorizes the block. This integration eliminates black-box guesswork, accelerates incident triage, and provides a permanent, auditable log of why the system intercepted the message. Scientifically, the study contributes evidence that transparency and strong performance are not necessarily in tension for lexically distinctive SMS phishing, and that few-shot and drift scenarios can be evaluated

with strict separation between what is demonstrated and what is simulated [12,16]. These implications should be contextualized rather than overgeneralized: they pertain to Indonesian SMS phishing with the specific corpus and protocols used here, and broader deployment claims would require multi-corpus and temporally split evaluation.

3.17. Limitations and Generalizability

Several limitations bound the conclusions. First, all experiments rely entirely on a single private corpus. The study lacks external validation on independent datasets and lacks temporally split validation. Generalizability to other populations, languages, or future time periods remains strictly unproven [13]. Readers must not assume these performance metrics will replicate in different operational environments without independent verification. Second, the main model is evaluated on one hold-out split, precluding claims of statistical significance; the cross-validation and bootstrap intervals partially mitigate but do not replace repeated independent testing. Third, the concept drift is simulated and lexical, not naturally occurring, so drift results are confined to that controlled scenario [12]. Fourth, the class-introduction experiment is binary, not multiclass class-incremental learning, and must not be interpreted as the latter. Fifth, runtime, memory, and environment versions were not recorded, so no efficiency claim is made despite the model's structural simplicity. Finally, minor pre-split text-cleaning (deduplication) is disclosed as a low-risk leakage consideration. These boundaries are stated explicitly in keeping with the standard that a rigorous manuscript acknowledges its experimental limits precisely.

4. CONCLUSION

This study developed and audited a transparent framework for Indonesian SMS phishing detection built on a class-weighted TF-IDF Logistic Regression and evaluated across supervised, few-shot, simulated-drift, class-introduction, and explainability scenarios. The detector achieved an F1 of 0.9838, an MCC of 0.9598, and a PR-AUC of 0.9983 on a conservatively deduplicated hold-out, with stable five-fold performance. Episodic evaluation quantified monotonic improvement from one-shot to ten-shot (mean F1 0.6288 to 0.9034), a simulated-drift experiment showed that an adaptive learner preserved higher post-drift F1 (0.9808) than a static learner (0.9604), and an exactly validated SHAP explanation together with a locally faithful LIME surrogate enabled SHAP-based feature selection that cut features by 95% while retaining an F1 of 0.9741 with Jaccard stability of 0.818. The principal contribution is a reproducible, compact, and auditable detection pipeline. This transparency meets the strict compliance and interpretability requirements of operational security environments. Automated blocking systems require explicit justification. The integrated explanations provide this evidence directly, while the evaluations of few-shot behaviour and drift responsiveness define the operational boundaries of the model. The findings are bounded by the use of a single corpus, a single hold-out for the main model, and simulated rather than natural drift. Future work should therefore pursue external and temporally split validation, naturally occurring drift, multiclass class-incremental learning, comparison against transformer-based and metric-learning few-shot models, and measured computational efficiency for deployment. None of these directions is claimed as already implemented.

AUTHOR CONTRIBUTIONS

Wartono Wartono: Conceptualization, methodology, supervision, statistical analysis, validation, interpretation of findings, and writing—review and editing. Rahmad Abdillah: Data curation, preprocessing, software implementation, investigation, model training and evaluation, few-shot learning experiments, simulated concept drift analysis, SHAP/LIME interpretation, visualization, and writing—original draft. Both authors contributed to discussing the findings and limitations, revising the manuscript, and approving the final version. Both authors agreed to be accountable for the accuracy and integrity of the work.

FUNDING STATEMENT

This research received no external funding.

DECLARATION OF AI USE

The authors declare that no AI tools were used in preparing this manuscript.

CONFLICT OF INTEREST

The authors declare no conflicts of interest.

DATA AVAILABILITY STATEMENT

All data supporting this study are included in the article. No separate datasets are available.

ETHICAL APPROVAL

Ethical approval was unnecessary, as no humans or animals were involved.

ACKNOWLEDGMENTS

The author acknowledges to funder for its financial support of this research in 2026. Appreciation is expressed to Universitas Islam Negeri Sultan Syarif Kasim Riau for support and provision of laboratory facilities used in this research.

REFERENCES

[1] Imrana, Y., Xiang, Y., Ali, L., Noor, A., Sarpong, K., & Abdullah, M. A. (2024). CNN-GRU-FF: A double-layer feature fusion-based network intrusion detection system using convolutional neural network and gated recurrent units. *Complex & Intelligent Systems*, **10**(3), 3353-3370. DOI: https://doi.org/10.1007/s40747-023-01313-y

[2] Wang, L., Xu, M., & Cheng, H. (2023). Phishing scams detection via temporal graph attention network in Ethereum. *Information Processing and Management*, **60**(4). DOI: https://doi.org/10.1016/j.ipm.2023.103412

[3] Alzubaidi, L., Bai, J., Al-Sabaawi, A., Santamaría, J., Albahri, A. S., Al-dabbagh, B. S. N., Fadhel, M. A., Manoufali, M., Zhang, J., Al-Timemy, A. H., Duan, Y., Abdullah, A., Farhan, L., Lu, Y., Gupta, A., Albu, F., Abbosh, A., & Gu, Y. (2023). A survey on deep learning tools dealing with data scarcity: Definitions, challenges, solutions, tips, and applications. *Journal of Big Data*, **10**(1), 46. DOI: https://doi.org/10.1186/s40537-023-00727-2

[4] Cremer, F., Sheehan, B., Fortmann, M., Kia, A. N., Mullins, M., Murphy, F., & Materne, S. (2022). Cyber risk and cybersecurity: A systematic review of data availability. *The Geneva Papers on Risk and Insurance - Issues and Practice*, **47**(3), 698-736. DOI: https://doi.org/10.1057/s41288-022-00266-6

[5] Tian, J.-Y., Wang, Z.-M., Fang, H., Chen, L.-M., Qin, J., Chen, J., & Wang, Z.-H. (2022). Few-Shot Learning-Based Network Intrusion Detection through an Enhanced Parallelized Triplet Network. *Security and Communication Networks*, **2022**, 1-13. DOI: https://doi.org/10.1155/2022/3317048

[6] Wang, J., Xia, Y., Zhao, W., Zhang, Y., & Wu, F. (2021). A Novel Framework for Authority Management Based on Knowledge Base Completion of the Graph Neural Network. *Wireless Communications and Mobile Computing*, **2021**, 1-8. DOI: https://doi.org/10.1155/2021/1735349

[7] Xu, T. (2025). Enhancing Cyber Security: Comparing the Accuracy of the Bert Model with Other Common Deep Learning Models in Identifying Email Spam. *Advances in Engineering and Intelligence Systems*, **4**(1), 84-101. DOI: https://doi.org/10.22034/aeis.2025.492892.1257

[8] Wang, R., Huang, M., Zhao, J., Zhang, H., Zhong, W., Zhang, Z., & He, L. (2024). A light-weight model design approach for few-shot malicious traffic classification. *Scientific Reports*, **14**(1), 24710. DOI: https://doi.org/10.1038/s41598-024-73342-7

[9] Lundberg, S. M., & Lee, S.-I. (2017). A Unified Approach to Interpreting Model Predictions. *31st Conference on Neural Information Processing Systems (NIPS 2017)*.

DOI: https://doi.org/10.48550/arXiv.1705.07874
[10] Ribeiro, M., Singh, S., & Guestrin, C. (2016). "Why Should I Trust You?": Explaining the Predictions of Any Classifier. *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, 97-101.
DOI: https://doi.org/10.18653/v1/N16-3020
[11] Shtayat, M. M., Hasan, M. K., Sulaiman, R., Islam, S., & Khan, A. U. R. (2023). An Explainable Ensemble Deep Learning Approach for Intrusion Detection in Industrial Internet of Things. *IEEE Access*, **11**, 115047-115061.
DOI: https://doi.org/10.1109/ACCESS.2023.3323573
[12] Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., & Bouchachia, A. (2014). A survey on concept drift adaptation. *ACM Computing Surveys*, **46**(4), 1-37.
DOI: https://doi.org/10.1145/2523813
[13] Tian, J., & Zhu, H. (2025). Evaluating the efficacy of AI-driven intrusion detection systems in IoT: A review of performance metrics and cybersecurity threats. *PeerJ Computer Science*, 11, e3352.
DOI: https://doi.org/10.7717/peerj-cs.3352
[14] Tawfik, M., Abu-Ein, A. A., Abdelhaliem, A. H., Al-Sharo, Y. M., & Fathi, I. S. (2025). Explainable few-shot learning with modern BERT for detecting emerging phishing attacks using XF PhishBERT. *Scientific Reports*, **15**(1), 42821.
DOI: https://doi.org/10.1038/s41598-025-27500-0
[15] Salton, G. & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, **24**(5), 513-523.
DOI: https://doi.org/10.1016/0306-4573(88)90021-0
[16] Snell, J., Swersky, K., & Zemel, R. (2017). Prototypical networks for few-shot learning. *Advances in Neural Information Processing Systems*, **30**, 136033.
URL: https://www.scopus.com/pages/publications/85046993347
[17] Chicco, D., & Jurman, G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics*, **21**(1), 6.
DOI: https://doi.org/10.1186/s12864-019-6413-7