IDS 705 Final Report

Behavioral Cues and Adversarial Robustness in Phishing Email Detection

Team Members:
Ailina Aniwan
Jay Liu
Eric Ortega Rodriguez
Tursunai Turumbekova

Team Identifier:
12 Capybara

Abstract

Phishing emails remain a major cybersecurity threat, often exploiting linguistic ambiguity and psychological manipulation to bypass traditional filters. This project evaluates four dimensions of phishing detection using supervised learning: model architecture, text preprocessing, adversarial robustness, and behavioral feature engineering. We compare traditional models (Naïve Bayes, Logistic Regression, XGBoost) with deep learning (BERT), and investigate how structured metadata and deception cues improve classification. Our results show that BERT offers stable high performance with minimal tuning, while XGBoost with additional features achieves the highest overall accuracy. We also find that even subtle improvements, such as a reduction in false negatives, are meaningful in high-risk domains. Finally, our result indicates that integrating psychological deception features can enhance detection in lightweight models and offer an interpretable, effective supplement to content-based approaches.

Introduction

Phishing is one of the most widespread and damaging forms of cyberattack, often the first step in a chain of events that leads to data breaches, financial loss, and identity theft. Research suggests that over 90% of cyberattacks begin with a phishing email (Deloitte, n.d.), which shows the urgency of developing more reliable detection systems. The financial stakes are substantial: the average cost of a phishing-related breach now exceeds $4.91 million, according to IBM’s 2023 Cost of a Data Breach Report (IBM, 2023). These attacks often impersonate trusted organizations and exploit psychological tactics such as fear, urgency, and authority to deceive victims (Proofpoint, 2023).

Over the past decade, phishing detection methods have evolved from static rule-based filters and blacklists to more dynamic, machine learning-based approaches. Leading platforms such as Gmail have recently deployed large language models (LLMs) to strengthen their spam and scam defenses. Google’s latest LLM, trained on phishing, malware, and spam examples, now blocks 20% more spam and processes 1,000 times more user-reported spam per day compared to earlier systems (Wen, 2024). However, attackers are increasingly leveraging generative AI to craft convincing phishing messages, creating new challenges for detection systems that rely solely on content or structural heuristics.

Saha et al. (2023) underscore the importance of robustness to adversarial examples—phishing messages intentionally designed to evade classifiers. Similarly, Microsoft’s *2023 Digital Defense Report* notes that approximately 90% of phishing attacks exploit social engineering, targeting behavioral vulnerabilities (Microsoft Threat Intelligence, 2023). Because users often respond reflexively to these tactics, technical safeguards alone may be insufficient. These findings suggest that incorporating psychological insights could enhance existing models’ ability to detect more nuanced phishing strategies.

To address these challenges, we design a four-part experimental framework: (1) comparing baseline machine learning models with BERT to identify tradeoffs in performance and interpretability; (2) evaluating how different preprocessing pipelines affect traditional models; (3) testing BERT’s robustness to adversarial noise via synthetic perturbations; and (4) investigating whether engineered behavioral cues, such as urgency or fear, can enhance detection in lightweight classifiers. We conduct these experiments using a realistic email corpus sourced from the 2007 TREC Public Spam Corpus, recently cleaned and re-released by Champa et al. (2024b). This dataset contains over 50,000 labeled emails, annotated for phishing versus legitimate status, and provides a rich foundation for exploring both textual and metadata-driven phishing detection strategies. Together, our experiments move beyond accuracy metrics to examine stability, interpretability, and practical resilience in phishing detection systems.

Background

Phishing detection has long been a focus of cybersecurity research, with early efforts applying traditional supervised learning methods to email content and metadata. Fette et al. (2007) stated that Random Forests could effectively identify phishing emails using structural features extracted from email headers. Similarly, Cormack and Lynam (2007) benchmarked multiple classical classifiers, including Naïve Bayes and Support Vector Machines (SVMs), on the TREC Public Spam Corpus. More recently, Champa et al. (2024b) revisited the TREC 2007 dataset, using Naïve Bayes and SVMs to establish classical baselines and release a cleaned version of the corpus for academic use.

In addition to these foundational studies, recent research has expanded phishing detection using both deep learning and behavioral approaches. Otieno et al. (2023) demonstrated that fine-tuning BERT models on email data can significantly improve phishing classification accuracy with minimal feature engineering. Aslam et al. (2024) further scaled phishing detection to large datasets, achieving over 98% accuracy by applying lightweight fine-tuning techniques to transformer architectures. To enhance model robustness, other studies have applied adversarial training techniques, such as using GAN-generated phishing examples, resulting in models that are substantially harder to fool (Ramesh et al., 2023). Complementing model-based advances, behavioral studies have deepened understanding of psychological vulnerabilities exploited in phishing. Akdemir and Yenal (2021) conducted a content analysis of COVID-19 phishing emails, finding that emotional appeals like fear and urgency were dominant deception tactics. Similarly, Butavicius et al. (2022) showed that time pressure and cognitive load reduce user detection of phishing cues. Together, these recent studies illustrate that both machine learning advancements and an awareness of human psychological vulnerabilities are critical to building resilient phishing detection systems.

While these approaches remain widely used, they rely on shallow lexical and syntactic features that may miss deeper semantic signals. Modern deep learning models such as Long Short-Term Memory (LSTM) networks and transformer-based architectures like BERT have shown strong performance in general-purpose NLP tasks but remain underexplored in phishing-specific applications. Few studies benchmark these advanced models against traditional baselines on realistic datasets like TREC. This leaves open questions about their robustness, efficiency, and practicality in real-world scenarios. Our goal is to fill this gap by systematically comparing traditional and deep learning models under controlled experimental settings.

Parallel research has looked at how behavioral and psychological cues appear in phishing emails and contribute to their effectiveness. Verma and Das (2017) showed that deception cues, such as urgency, fear, and authority, frequently appear in phishing messages and can improve classification performance when modeled explicitly. Abdelhamid et al. (2014) extended this insight to social media phishing, while Bountakas and Xenakis (2022) proposed hybrid ensemble models that integrate behavioral and structural email features. More recently, Schmitt and Flechais (2024) pointed out the growing importance of combining syntactic patterns with social engineering tactics, especially in the context of increasingly realistic phishing attacks powered by generative AI. These studies inspire our fourth experiment, where we engineer deception-aware features to test whether incorporating behavioral cues can improve phishing detection beyond text classification alone.

Taken together, these works motivate our multi-pronged evaluation: benchmarking modern NLP models on a phishing corpus, assessing the impact of preprocessing pipelines, evaluating adversarial robustness, and introducing behavioral feature engineering as a path to more resilient phishing detection.

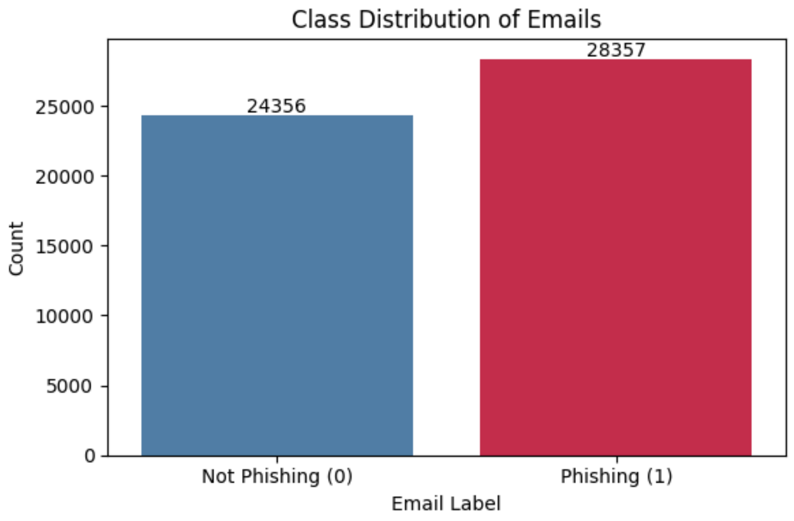
Data

We used a publicly available email corpus sourced from the 2007 TREC Public Spam Corpus (Champa et al., 2024b), hosted on Zenodo. The dataset contains 52,713 labeled emails, each annotated as phishing (label = 1) or not phishing (label = 0). For each email, the dataset includes metadata such as sender, receiver, subject line, body text, number of URLs, and timestamp information. This rich structure allowed us to experiment with both pure-text and hybrid modeling approaches.

However, it is important to note that phishing techniques have evolved significantly since 2007. For example, modern phishing attacks increasingly incorporate AI-generated text, sophisticated spear-phishing tactics, and advanced evasion strategies. These attacks have become notably more personalized and targeted toward individual victims, making detection more challenging (IBM X-Force Threat Intelligence Index, 2024).

The dataset offers a realistic view of email traffic and phishing strategies observed in the wild, including varied writing styles, spam tactics, and impersonation patterns. As shown in Figure 1, the dataset is moderately imbalanced, with phishing emails comprising about 54% of all samples. This class distribution points to the need for high recall in phishing detection, as even a small number of false negatives could result in real-world security risks.

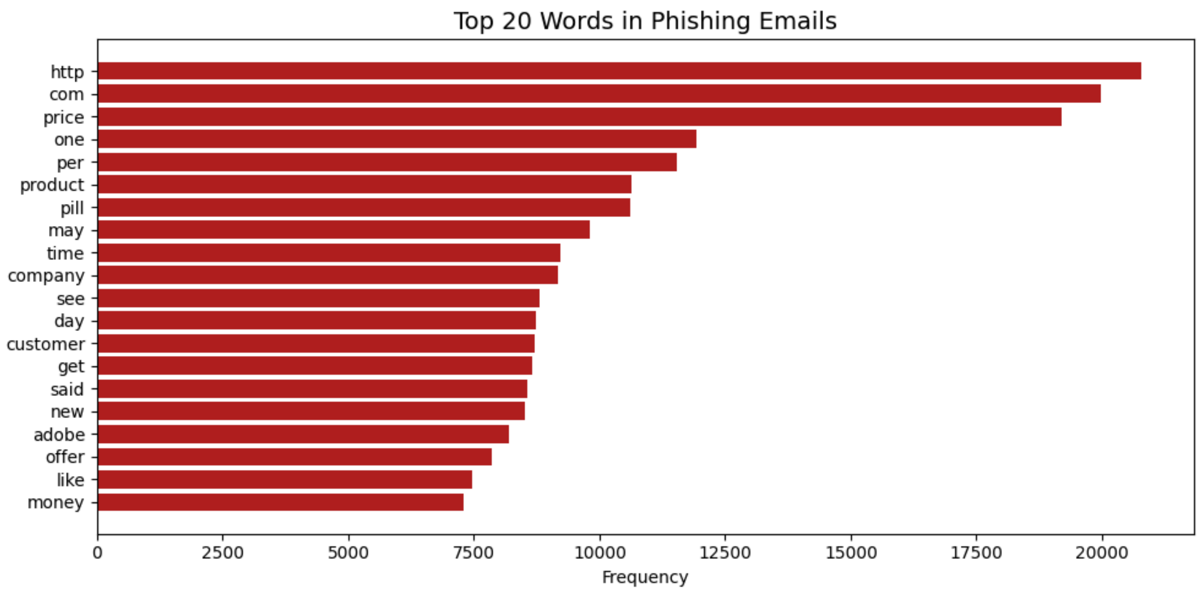
Figure 1. Class Distribution of Phishing vs. Non-Phishing Emails



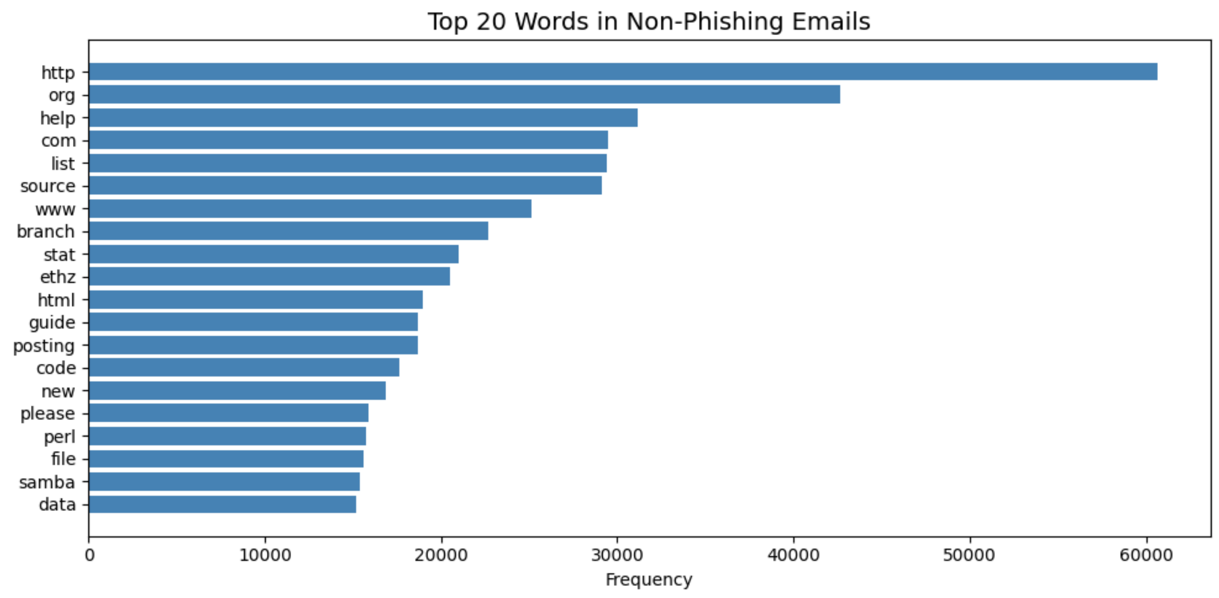
To develop our models, we created multiple input variants that differed in their degree of text preprocessing, from raw concatenations of subject and body to fully normalized versions with structural cleaning. These variants allowed us to systematically examine how text quality influences classification performance, as detailed in later experiments.

Beyond overall class distribution, we also explored lexical patterns that distinguish phishing from non-phishing emails. Figure 2 presents the top 20 most frequent terms within each class. As shown in Figure 2a, phishing emails are dominated by marketing and transactional vocabulary such as *"price"*, *"pill"*, *"offer"*, and *"money"*. These terms are commonly associated with scams or unsolicited promotions designed to manipulate recipients into clicking malicious links or making purchases. Figure 2b shows that non-phishing emails more frequently include technical, collaborative, or organizational language, such as *"help"*, *"guide"*, *"source"*, and *"data"*, indicative of routine professional or academic exchanges. These content-level contrasts validate our use of textual features in model training and confirm that phishing content exhibits recognizable linguistic patterns.

Figure 2. Top 20 Most Frequent Words in Phishing and Non-Phishing Emails.



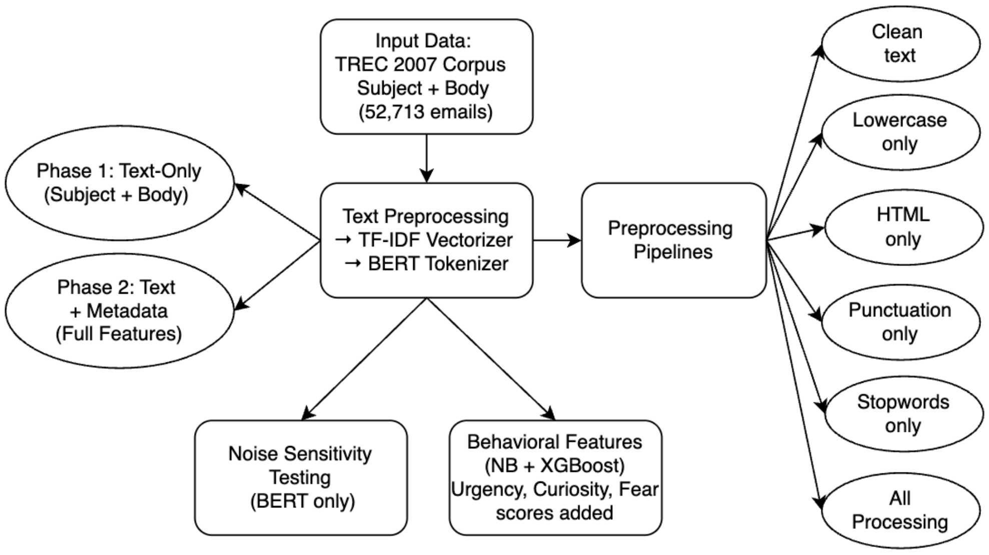
(a) Phishing emails.



(b) Non-phishing emails.

Experiments

Figure 3. Overview of Experimental Design and Modeling Pathways


Input Data: TREC 2007 Corpus Subject + Body (52,713 emails)
Phase 1: Text-Only (Subject + Body)
Phase 2: Text + Metadata (Full Features)
Text Preprocessing → TF-IDF Vectorizer → BERT Tokenizer
Preprocessing Pipelines
Clean text
Lowercase only
HTML only
Punctuation only
Stopwords only
All Processing
Noise Sensitivity Testing (BERT only)
Behavioral Features (NB + XGBoost) Urgency, Curiosity, Fear scores added

Experiment 1: Model Comparison and Feature Augmentation

This experiment aimed to identify the most effective supervised learning approach for phishing email detection. We evaluated four models: Naïve Bayes, Logistic Regression, XGBoost, and BERT, under two sequential conditions: (1) trained on clean email text only (subject + body), and (2) trained on text plus engineered metadata features. This two-phase design allowed us to isolate the impact of raw linguistic content from structural cues, and compare interpretable models with deep language architectures.

Phase 1: Clean Text Only
In the first phase, all models were trained using TF-IDF features extracted from the clean text of each email. The goal was to evaluate how well each classifier could perform using only linguistic content, without structural metadata.

Phase 2: Augmenting with Metadata
In the second phase, we extended the input to include structural features commonly found in phishing messages: sender and receiver (TF-IDF encoded), and a binary indicator for the presence of URLs. These features were concatenated with the TF-IDF vectorized text and used to train Logistic Regression and XGBoost.
BERT was excluded from this phase due to its architectural constraint: it does not natively accept tabular inputs alongside tokenized text without custom model engineering. Integrating structured inputs into BERT would require a multimodal architecture such as TabTransformer or early/late fusion designs—beyond the scope of this project.

Experiment 2: Data Processing Pipelines Comparison

In Experiment 2, we explored whether standard text preprocessing techniques, such as lowercasing, punctuation splitting, and HTML tag removal, would improve phishing email detection performance. While these steps are generally effective in traditional natural language processing tasks, phishing emails present unique challenges: they often contain deliberate noise, irregular formatting, or deceptive structures designed to evade automated systems. It was therefore uncertain whether conventional preprocessing would enhance classification performance or inadvertently remove meaningful linguistic signals.

To disentangle these effects, we tested each preprocessing step individually, in addition to a combined pipeline. This allowed us to isolate the marginal impact of each transformation and assess whether certain steps were genuinely helpful when applied to phishing data, or whether some transformations might prove specifically beneficial compared to others. By evaluating both independent and combined approaches, we aimed to determine whether additional preprocessing consistently improved performance or risked degrading key signals relevant to phishing detection.

Experimental Setup

We specifically tested three supervised learning models: Logistic Regression, Naïve Bayes, and XGBoost. Each model was evaluated under two major settings: (1) using text features alone, and (2) using text features combined with additional metadata features (such as sender domain, receiver domain, and URL presence). This dual approach helped to examine how preprocessing effects differ depending on the richness of available input features.

For the textual inputs, we constructed six different preprocessing pipelines applied to the raw email content, each modifying the data as follows:

- **clean_text**: Raw combined subject and body text without any transformation or cleaning beyond basic extraction.
- **lowercase_only**: Converted all text to lowercase without further modification.
- **html_only**: Removed HTML tags but retained all other elements of the original text.
- **punct_only**: Split punctuation into separate tokens to capture potential patterns related to phishing attempts.
- **stopwords_only**: Removed standard stopwords from the text to focus on potentially more meaningful tokens.
- **all_preprocessing**: Applied lowercasing, HTML tag removal, punctuation splitting, and whitespace normalization in combination. Stopwords were intentionally retained to preserve informative urgency-related tokens (e.g., "now," "verify," "your").

All text variants were vectorized using TF-IDF with a cap of 1,000 features. For models incorporating metadata, structured fields such as sender, receiver, and URL presence were separately vectorized (where appropriate) and concatenated with the text features.

The data was split into 80% training and 20% testing subsets using stratified sampling to preserve the original class distribution. Across all models and pipelines, performance was evaluated using accuracy, precision, recall, F1-score, and confusion matrices. Given the context of phishing detection, special attention was paid to recall for the phishing class, where missed detections could have particularly severe consequences.

In total, the experimental design yielded twelve configurations for each model type: six preprocessing pipelines evaluated under text-only conditions and six evaluated with full metadata inclusion.

Experiment 3: Evaluating BERT’s Robustness to Adversarial Noise

Background and Motivation

Phishing emails are intentionally altered to bypass detection systems using techniques such as homoglyph alterations, typographic errors, or even including misleading formatting. While humans can often recognize these manipulations with context, machine learning models, particularly those that rely on text features, are vulnerable to such attacks. Prior studies, such as Ramesh et al. (2023), highlight how transformer-based models such as BERT suffer. While Ramesh et al. (2023) primarily focus on synonym substitution and word swapping, our work focuses on character-level adversarial attacks, including homoglyph changes, typographical errors, and grammatical perturbations.

These tactics are also increasingly being seen in real-world attacks. According to the IBM X-Force Threat Intelligence Index 2024, phishing has remained one of the most common attacks and is a technique that has exploited many automated detection systems. This experiment highlights how a range of noise levels can bypass and deceive models into misclassifying adversarial attack content.

For example, BERT's performance drops when exposed to adversarial noise. Given BERT’s strong baseline performance from earlier experiments, our team implemented a test to evaluate BERT’s robustness under real-world conditions, especially one where a user might experience an adversarial attack. The types of noise we explore in this experiment align with the manipulation mentioned above.

More specifically, homoglyph substitutions, typographic distortions, and grammatical errors are considered adversarial tactics because they subtly alter the text in a way that preserves human readability while deceiving automated models. These conditions exploit the gap between human contextual understanding and the literal pattern recognition of language models, which makes them effective for phishing attempts.

Experimental Design

To simulate adversarial attacks, we introduced textual noise into the clean email dataset by randomly injecting homoglyphs, grammatical errors, and character alterations to a defined percentage of each email's text. We applied this noise at four levels: 5%, 10%, 15%, and 30%.

Each model was trained using the same 80/20 train-test split, with consistent hyperparameters across runs. This ensures that the observed performance differences are attributed directly to the adversarial noise levels. In the end, we evaluated models based on accuracy, precision, recall, and F1-score for the phishing class.

To capture more nuanced trends, we also expanded the adversarial noise levels from 0% to 50% in 5% increments. This granularity allowed us to assess BERT's sensitivity to a wider range of perturbation levels and better simulate real-world adversarial conditions.

Table 1. Visual examples of adversarial noise injections across increasing noise levels.

Noise Level	**Perturbed Sentence**
0%	*Your account has been suspended. Click here to verify your identity.*
5%	*Your accoUnt has Ьeen suspended Click here to verify your identity*
10%	*Your acCOunt has Ьeen $uspended CIlck here to verify your identiтy*
15%	*Your acCOunT has Ьeen $Uspended Click here to verify Your identiтy*
30%	*Your aCCouNt has ЬeEN $uspenԁed Click here тO vErify your identiтy*
50%	*YouR aCCoUnт has ЬEen $u$pEnԁeD Click heRe To verify your ideNтITy*

Integration into Workflow

The implementation was executed through a series of notebooks using a standardized training and evaluation pipeline. Each model's performance was logged and summarized in a visualization notebook that consolidates the key metrics and enables side-by-side (across noise levels). In addition to this, our team also created a realistic simulation of inbox behavior, visualizing how a user receiving 100 emails per day might be affected, tracking correct classifications, false positives, and false negatives at each noise level.

To ensure reliable evaluation, we averaged performance metrics across three randomized runs per noise level. This helped stabilize outlier effects and produced more robust estimates of false positives, false negatives, and accuracy. We also simulated realistic user-facing consequences by modeling how a user receiving 100 emails daily would be impacted under each noise level, using counts of true and false classifications.

Experiment 4: Social Engineering Detection via Behavioral Feature Engineering

Phishing attacks rely not only on malicious links or attachments but also on psychological manipulation. Emails that invoke urgency, fear, or authority are often designed to pressure recipients into taking unsafe actions. This experiment evaluates whether detecting manipulative psychological tactics improves phishing classification accuracy. We aim to quantify social engineering traits—specifically urgency, fear, and curiosity—and incorporate them as additional features in machine learning models.

Our approach draws inspiration from recent work on behavioral threat detection in natural language processing. Notably, SOCIALBERT, developed by Abobor and Josyula (2023), fine-tuned a DistilBERT model to detect 13 social engineering tactics, later narrowed to 6 dominant tactics (e.g., Pretexting, Baiting, Authority, and Urgency) in SMS messages. SOCIALBERT achieved a 97.55% accuracy, outperforming baseline transformer and Naive Bayes models. While SOCIALBERT provides a powerful framework for labeling deception traits, we did not fine-tune SOCIALBERT directly on our phishing email dataset. Rather, following their insights, we adapted the concept of behavioral deception feature extraction tailored to email-specific phishing scenarios. A future extension of this work could involve applying SOCIALBERT to emails and analyzing which psychological tactics are easier or harder for models to detect. In parallel, Shahriar et al. (2023) proposed psychological trait scoring for phishing detection and demonstrated the predictive value of urgency, fear, and desire-based language patterns.

To model psychological manipulation patterns, we constructed a custom prompt dictionary representing three prominent social engineering tactics: urgency, fear, and curiosity. These categories were selected based on prior research (Shahriar et al., 2023) and consistent behavioral patterns identified in phishing emails. For each category, we manually crafted five representative phishing-style prompts based on real-world phishing examples and cybersecurity reports. These include phrases such as: “Act immediately or your account will be closed”, “Click here to uncover a hidden opportunity” etc. (See appendix for more examples) Using Sentence-BERT (SBERT), we computed cosine similarity between these prompts and each email’s clean text. Both the maximum similarity and the average similarity across multiple prompt variations were used to generate deception scores for each trait. These deception features were then scaled using MinMaxScaler and appended to TF-IDF matrices (for Naive Bayes) or to sparse feature sets containing metadata (for XGBoost). We assessed the impact of these features by comparing models trained with and without them.

Results

To assess model performance across different conditions, we evaluated each experiment using consistent validation and scoring methods. All models were trained using an 80/20 stratified train-test split, with performance measured by accuracy, precision, recall, and F1-score. We also examined confusion matrices and false negative counts, which are especially critical in phishing detection. The following subsections summarize key outcomes from each experiment and where models succeeded, struggled, and how specific design choices affected results.

Experiment 1 Results: Model Comparison and Feature Augmentation

Phase 1 (Clean Text Only): As shown in Table 1, BERT achieved the highest overall performance (accuracy: 99.68%, F1-score: 1.00), with XGBoost and Logistic Regression close behind. Naïve Bayes underperformed relative to the other models, struggling particularly with legitimate emails containing spam-like phrases. Confusion matrix analysis revealed that BERT made only 34 total misclassifications (22 false positives and 12 false negatives), while XGBoost and Logistic Regression made 73 and 150, respectively. These results showed the advantage of deep pre-trained models in capturing contextual nuances from unstructured text. However, XGBoost's strong results, achieved with significantly lower training cost, make it a viable alternative in resource-constrained settings.

Phase 2 (Clean Text + Metadata): When structural features (sender, receiver, and URL presence) were added to the traditional models, performance improved for both Logistic Regression and XGBoost. Logistic Regression reached 99.66% accuracy and an F1-score of 1.00. XGBoost achieved 99.75% accuracy even surpassing BERT's performance despite relying on simpler modeling techniques. These gains demonstrate that structured metadata can compensate for the lack of deep semantic modeling, especially when combined with a flexible learner like XGBoost. Because BERT cannot natively accept these features without custom architecture, it was excluded from this phase.

The key takeaway from this experiment is that BERT excels in pure-text classification tasks, where it achieves near-perfect performance without requiring extensive feature engineering. Its performance was stable across multiple runs, even when trained with minimal tuning (only two epochs). This makes BERT particularly attractive in contexts where high performance is needed from raw text alone, or where structured metadata is unavailable.

In contrast, XGBoost, with the addition of metadata features, was able to slightly surpass BERT's accuracy. However, this performance came from model-specific tuning and engineered inputs. While effective, this approach may be more sensitive to data quality and feature availability. Logistic Regression similarly benefited from feature augmentation but did not match BERT's baseline performance. These findings suggest that while XGBoost is the best overall performer when full input is available, BERT remains the most stable and versatile model across varied conditions. As a result, we selected BERT as our benchmark deep learning model in the subsequent experiments.

Table 2. Model Performance Across Input Conditions (Clean Text vs. Text + Metadata)

Model	**Input Features**	**Accuracy**	**F1-Score**
Naïve Bayes	clean_text	95.40%	0.95
Logistic Regression	clean_text	98.55%	0.99
XGBoost	clean_text	99.31%	0.99
BERT	clean_text	**99.68%**	**1.00**
Logistic Regression	Clean_text + metadata	99.66%	1.00
XGBoost	Clean_text + metadata	**99.75%**	**1.00**

Recent research shows that achieving very high accuracy rates (95%–99%) is typical for phishing detection models evaluated on benchmark datasets. For example, Otieno et al. (2023) fine-tuned BERT models for phishing classification and reported accuracy above 98%, while Aslam et al. (2024) achieved similar performance by optimizing lightweight transformer models. Extremely high accuracy is particularly important in phishing detection because the cost of false negatives, failing to detect a phishing email, can result in severe security breaches (Abobor & Josyula, 2023). However, it is important to recognize that datasets like the TREC 2007 corpus reflect phishing strategies from more than a decade ago, when attacks were generally less sophisticated than today's AI-assisted phishing tactics (Schmitt & Flechais, 2024). Thus, while the results on this dataset are consistent with the literature, they may somewhat overestimate expected performance against modern phishing threats.

Experiment 2 Results: Data Processing Pipelines Comparison

Across all models, the most meaningful performance gains came from the inclusion of metadata, while the impact of text preprocessing alone was more model-specific. The effectiveness of individual preprocessing steps varied, with some transformations improving accuracy or reducing critical errors, while others provided little to no benefit.

Naïve Bayes showed the greatest sensitivity to textual preprocessing. Removing stopwords improved accuracy from 95.40% to 95.76%, and F1-score from 0.95 to 0.96. This variant also reduced false positives from 306 to 254, though false negatives increased slightly from 179 to 193. These changes suggest that eliminating non-informative words helped Naïve Bayes better isolate relevant features for phishing detection. Other preprocessing steps, such as lowercasing or punctuation splitting, had minimal or slightly negative effects. Full preprocessing did not outperform stopword removal alone, which suggests that aggressive normalization may reduce model effectiveness by discarding useful cues.

Logistic Regression was largely robust to textual transformations under the text-only condition. Accuracy remained stable around 98.55% across all pipelines, with no meaningful differences. However, when metadata was included, performance improved consistently, reaching as high as 99.68% accuracy. While this increase over the text-only condition may seem small, the corresponding confusion matrices show reduced false negatives, from 52 in the text-only clean version to just 9 in the full-featured model. This reduction is particularly valuable in phishing detection, where even a single missed attack could result in real-world harm. The addition of metadata appears to compensate for the lack of textual preprocessing and allows the model to identify phishing signals based on sender identity, URL presence, and recipient context.

XGBoost achieved the highest overall performance. Under the text-only setting, accuracy improved slightly with preprocessing (from 99.30% to 99.35%), but the effect was minimal. The model consistently maintained near-perfect classification across all text pipelines, which confirms that XGBoost is highly resilient to input noise and does not depend heavily on preprocessing. The most meaningful gains occurred when metadata was introduced. All full-feature variants surpassed 99.72% accuracy, with the best-performing pipeline (all preprocessing and full features) reaching 99.78%. This configuration reduced false negatives to 11 and false positives to 12 and made it the best overall model in the study. These improvements, while numerically small, are practically significant in the phishing context. For example, a 0.1% accuracy gain translates to over 100 additional correctly classified emails in a dataset of this size, and thousands when deployed at scale.

Overall, these results show that textual preprocessing can provide modest benefits, especially for simpler models like Naïve Bayes, but that the inclusion of structured metadata plays a far more critical role in reducing classification errors. Models that incorporate sender, receiver, and URL information consistently outperformed their text-only counterparts, and this pattern held across all algorithms.

Table 3. Top Model Performances Across Preprocessing Pipelines and Input Settings.

Model	**Input Setting**	**Best Preprocessing**	**Accuracy**	**Recall**	**FN**	**FP**
Naïve Bayes	Text only	stopwords_only	95.76%	0.96	193	254
Logistic Regression	Text only	punct_only	98.58%	0.99	51	99
Logistic Regression	Full features	all_preprocessing	99.68%	1.00	10	24
XGBoost	Text only	all_preprocessing	99.36%	0.99	17	51
XGBoost	Full features	all_preprocessing	99.78%	1.00	11	12

Note 1: Even small differences in accuracy (e.g., 0.1%) correspond to dozens of corrected predictions in this dataset. In phishing detection, minimizing false negatives is particularly important to reduce real-world risk.

Note 2: FN = False Negatives, FP = False Positives (from confusion matrix).
Full confusion matrices for all tested configurations are provided in the Appendix.

Experiment 3 Results: Evaluating BERT's Robustness to Adversarial Noise

After running our models for comparison, we observed a noticeable decline in BERT's robustness at the 10% noise level. As illustrated in Figure 3, the F1-score for phishing detection dropped significantly from 0.973 to 0.800, while overall accuracy fell to 74.5%. This dip in performance was most striking in the precision metric, which plummeted to 0.694, indicating that many safe emails were incorrectly flagged as phishing. Since precision represents the proportion of predicted phishing emails that are actually phishing, this drop reflects an increase in false positives, which is a critical issue in real-world deployment.

To contextualize this impact, we can simulate how such errors would affect a typical user, "John," who receives approximately 100 emails per day, with 30% being phishing. As shown in Figure 4, under 10% adversarial noise, BERT misclassified 12 legitimate emails as phishing (false positives) and missed 2 phishing emails (false negatives). This level of misclassification could lead to frustration, reduced trust in the filtering system, and even the ignoring of legitimate security alerts due to "alert fatigue."

Interestingly, at the 15% noise level, BERT's performance rebounded. The F1-score climbed back to 0.973, and accuracy returned to 97%, suggesting a partial recovery, possibly due to the model overfitting to more pronounced patterns introduced by the added noise. However, this improvement was not sustained; at 30% noise, performance declined again, with the F1-score dropping to 0.888. Although false negatives (missed phishing emails) remained consistently low across all levels of noise, the cost came in the form of increasing false positives, which pose a different kind of risk, namely, reduced efficiency and missed opportunities due to incorrectly blocked legitimate communication.

These findings mirror prior work by Ramesh et al. (2023), which demonstrated that deep learning models, particularly those based on transformers, are susceptible to even modest adversarial perturbations. Our results reinforce this vulnerability in a phishing detection context, suggesting that character-level noise, such as typos, homoglyphs, or other obfuscations, is sufficient to significantly disrupt performance. Thus, we recommend that future production-ready models incorporate robust preprocessing pipelines, adversarial training, or architecture-level improvements to sustain reliability under noisy conditions.

Figure 4. Impact of Adversarial Noise on BERT Classification Metrics. F1-score, accuracy, precision, and recall degrade sharply at 10% noise.

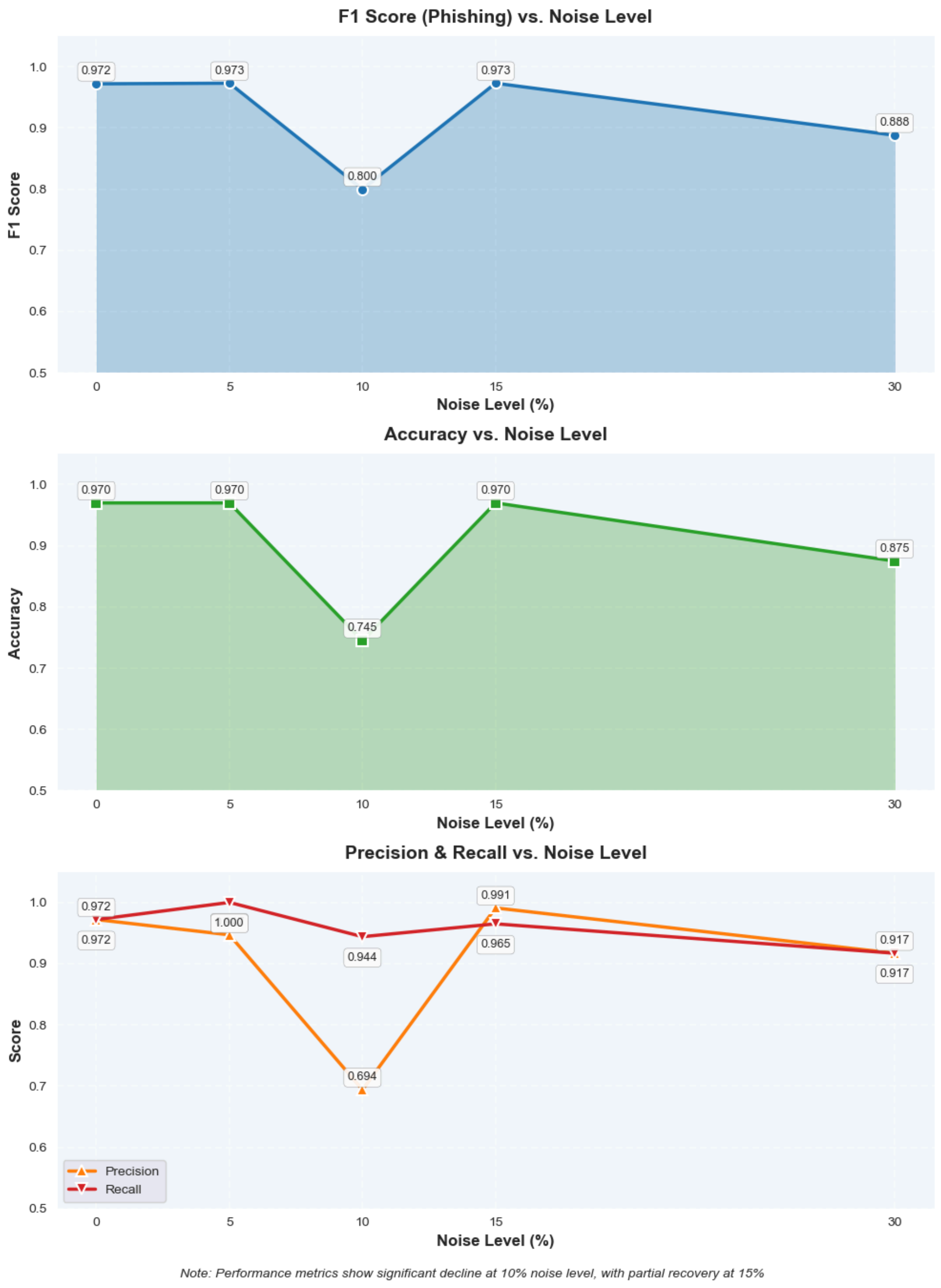
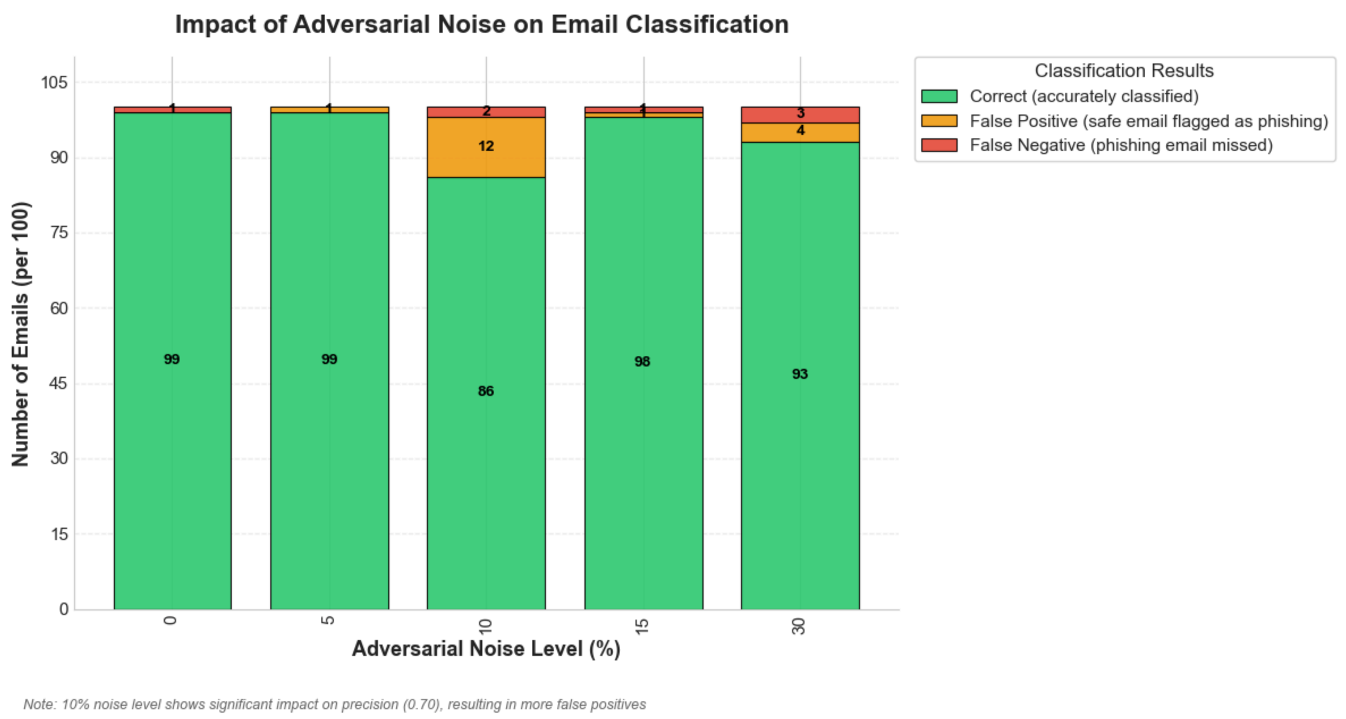


Figure 5. Simulated Email Classification Outcomes Across Noise Levels.

Classification performance with 30% phishing emails in daily traffic (100 emails)



We also refined our simulation of inbox outcomes by averaging results across three randomized runs per noise level. As shown in Figure 7, this averaging yielded more stable estimates and helped mitigate anomalies from individual runs. For instance, at 10% noise, the model misclassified an average of 12 legitimate emails as phishing. By 50% noise, false positive rates stabilized, suggesting the model began flagging most messages as suspicious. Together, Figures 6 and 7 illustrate both the statistical variance and the user-facing implications of deploying models under adversarial conditions.

Figure 6. Average BERT Classification Performance Across Noise Levels (0%–50%).

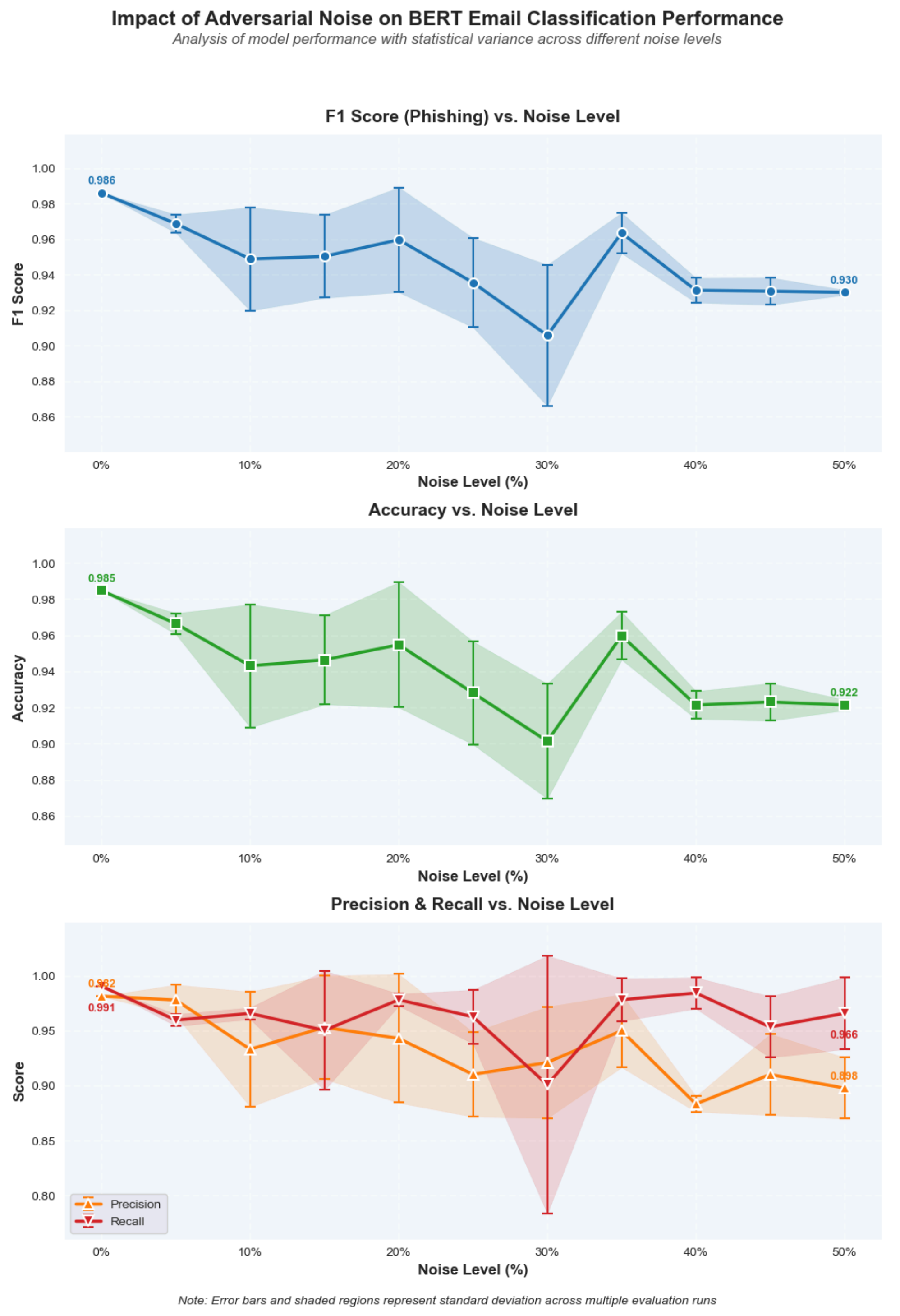
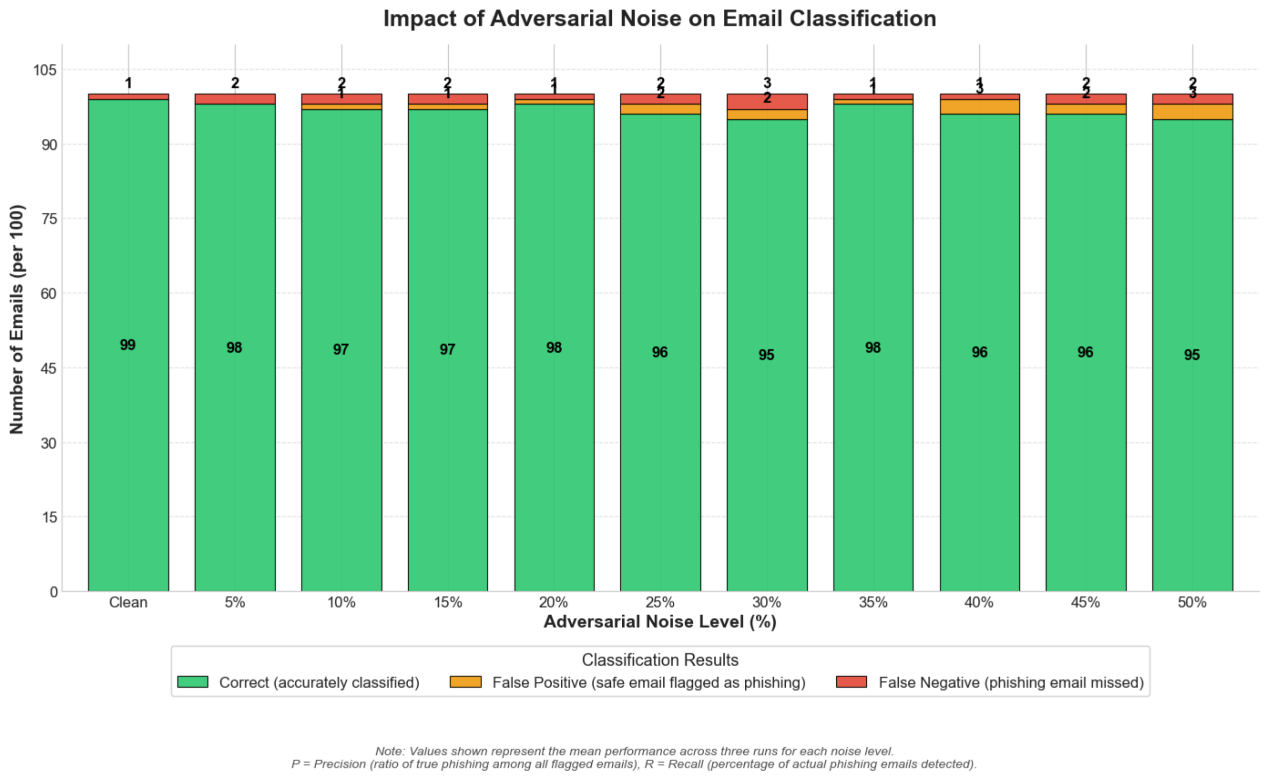


Figure 7. Simulated Inbox Outcomes Under Adversarial Noise (100 Emails per Day).



While averaged performance may appear stable, Figures 6 and 7 reveal that variance and misclassification risks persist, particularly in false positives at moderate noise levels. This underscores the importance of evaluating statistical consistency and user-facing consequences when deploying models in security-critical applications like phishing detection. As phishing remains a leading initial attack vector, as highlighted in the 2024 IBM X-Force Threat Intelligence Index, further work should explore a broader range of adversarial tactics, including synonym substitution, sentence reordering, and grammar distortions, to better simulate evolving threats.

Experiment 4 Results: Social Engineering Detection via Behavioral Feature Engineering

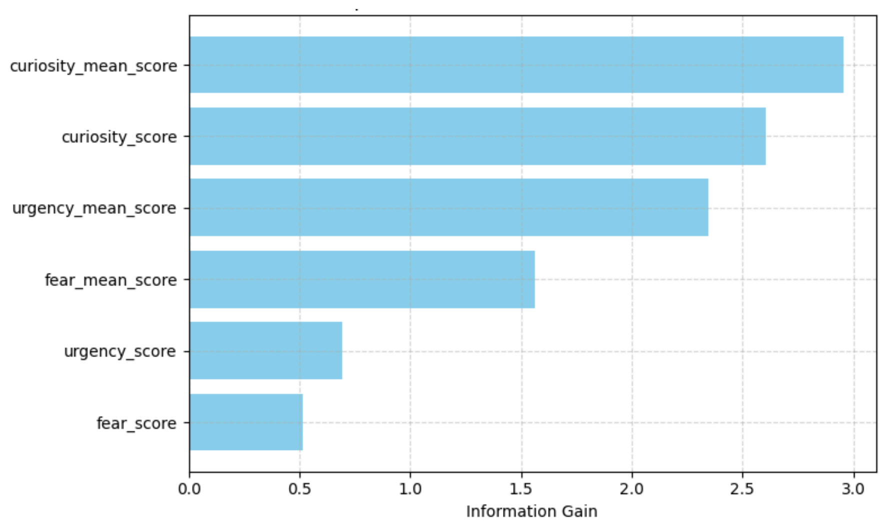
Across models, adding deception features consistently improved phishing detection performance. For Naive Bayes, accuracy increased from 95.39% to 95.96%, accompanied by a 13% reduction in false negatives. For XGBoost, which already achieved a strong baseline accuracy of 99.75%, incorporating deception features reduced false negatives by 33% and slightly decreased false positives, leading to a modest but measurable accuracy gain from 99.75% to 99.80%. These improvements, though subtle in absolute terms, are significant in high-stakes contexts like phishing detection, where even a few false negatives can lead to major security breaches.

Table 4. Model Performances Before and After Adding Deception Features.

Model	**Accuracy**	**Precision**	**Recall**	**FP**	**FN**
Naive Bayes Baseline	95.39%	0.95	0.95	306	179
Naive Bayes + Deception	95.96%	0.96	0.97	270	155
XGBoost Baseline	99.75%	1.00	1.00	14	12
XGBoost + Deception	99.80%	1.00	1.00	13	8

Figure 8 illustrates the relative contribution of each deception feature in the XGBoost model using information gain. Features derived from curiosity (both max and mean similarity scores) ranked highest, followed by urgency and fear. This reinforces psychological literature that highlights curiosity-driven exploration and urgency as key persuasive elements in phishing.

Figure 8. Relative contribution of deception features in XGBoost model using information gain



While our best-performing text classifier was a fine-tuned BERT model, integrating structured deception features (e.g., social engineering scores) directly into Hugging Face's BertForSequenceClassification is nontrivial. This architecture accepts only tokenized input fields (input_ids, attention_mask, and labels) and cannot directly accept additional structured numerical features. Integrating deception features would require custom PyTorch modeling—modifying the forward pass and model head to fuse structured and unstructured inputs. Due to these constraints and our project scope, we opted for a hybrid design. We used Sentence-BERT embeddings to compare deception prompt vectors with email vectors, extract psychological traits, and feed these into interpretable models like Naive Bayes and XGBoost.

This experiment demonstrates the value of psychological trait scoring in phishing detection. We validate our approach against *SOCIALBERT* and the work of Shahriar et al., showing that a hybrid of semantic and behavioral feature engineering significantly improves detection performance across traditional and ensemble models.

These findings emphasize the importance of integrating cognitive deception analysis with traditional ML pipelines for security applications. Future directions could include leveraging large-scale annotated deception datasets, refining prompt quality, and building BERT-based classifiers that can accept hybrid structured inputs.

Ethical Considerations

Our project raises several important ethical considerations across fairness, privacy, societal impact, and model transparency. First, we recognize the potential for bias in our dataset, which primarily includes English-language emails from the TREC 2007 Public Spam Corpus. This limits the model's generalizability and may result in reduced accuracy when applied to phishing content from other languages, cultural contexts, or more recent attack strategies. Additionally, our use of behavioral features, such as urgency or fear cues, could inadvertently flag legitimate communications that express emotion or urgency, disproportionately impacting certain user groups depending on communication norms.

From a privacy standpoint, while our dataset is anonymized, phishing detection systems are typically deployed in real-time environments where they analyze personal or organizational email content. This raises concerns around surveillance, consent, and data governance. Any real-world implementation of such models should therefore include strong encryption, auditability, and clear data retention policies to mitigate these risks.

There is also potential for negative societal impact if the technology is misuse, for example, if malicious actors reverse-engineer behavioral models to create phishing emails that evade detection. To reduce this risk, we focus on transparency and responsible disclosure, and we recommend ongoing model adaptation and adversarial robustness testing to stay ahead of evolving threats.

Finally, explainability is critical in cybersecurity applications. Stakeholders need to understand why an email is flagged as phishing, especially in enterprise and legal settings. We addressed this by favoring interpretable models like Logistic Regression and XGBoost, and by visualizing the contribution of behavioral deception features. These choices allow for meaningful insight into the model's decision-making process, which is essential for building trust and ensuring accountability.

In summary, while our system improves phishing detection performance, its ethical deployment must balance accuracy with fairness, privacy, transparency, and resilience to misuse.

Conclusions

As phishing attacks grow more sophisticated and widespread, defending against them calls for detection systems that combine high accuracy with robustness, interpretability, and adaptability. In this project, we explored how different modeling approaches, preprocessing strategies, noise conditions, and behavioral features affect the performance of phishing email classifiers. Through a series of four experiments, we evaluated a diverse set of models, including Naïve Bayes, Logistic Regression, XGBoost, and BERT, using the TREC 2007 public email corpus.

There are several key takeaways from our findings. First, while BERT delivered the strongest performance in text-only settings, XGBoost outperformed all other models when paired with structured metadata, which demonstrates the power of hybrid feature sets that include sender identity, URL presence, and recipient fields, especially in traditional models that benefit from structured inputs.

Second, we found that textual preprocessing had mixed effects. For simpler models like Naïve Bayes, removing stopwords improved accuracy and significantly reduced false positives, whereas transformations like lowercasing and HTML tag removal provided limited or even negative returns. In contrast, XGBoost and Logistic Regression were largely robust to these preprocessing choices, particularly when metadata was included. These results show that standard NLP preprocessing should not be assumed to help in adversarial contexts like phishing detection, where irregular formatting can carry predictive signals.

Third, even high-performing models like BERT were shown to be vulnerable to modest character-level adversarial noise. A 10% perturbation led to a sharp drop in precision and overall accuracy, which pointed to the need for robustness testing before deployment. Simulations further illustrated how such degradation could lead to alert fatigue and misclassification in real-world inboxes.

Finally, our behavioral feature engineering experiment showed that deception-aware cues like quantifying urgency, curiosity, and fear, can meaningfully improve phishing detection. Adding these features to Naïve Bayes and XGBoost reduced false negatives and boosted accuracy supported findings from psychological research on social engineering. Though incorporating these features into deep models like BERT remains technically challenging, they proved to be a valuable complement to traditional classifiers.

These experiment results suggest that effective phishing detection requires a multifaceted approach: strong models alone are not enough. Success depends on the careful integration of structural metadata, resilience to adversarial attacks, and signals grounded in behavioral psychology. Looking forward, future work could explore multimodal architectures that jointly encode metadata and text, adversarial training methods to enhance robustness, and larger deception-labeled corpora to enrich behavioral feature extraction. The intersection of machine learning and cognitive cues holds significant potential for advancing email security at scale.

Roles

Jay Liu led Experiment 1, responsible for training all four initial models (Naïve Bayes, Logistic Regression, XGBoost, and BERT) and building the foundational modeling pipeline used in later experiments. His work provided a strong baseline for subsequent experiments. Jay also conducted performance evaluations and comparative analysis between models with and without metadata features.

Ailina Aniwan conducted Experiment 2, which evaluated the effects of different text preprocessing pipelines. She implemented and tested six distinct preprocessing strategies across multiple models, analyzed model sensitivity to each pipeline, and incorporated full-feature extensions. She was in charge of project coordination, report editing, and overall quality control.

Eric Ortega Rodriguez led Experiment 3, which evaluated BERT's vulnerability to character-level adversarial noise. He implemented the synthetic noise generation process, trained BERT under various perturbation levels, and analyzed how noise impacted classification accuracy and precision. He also designed and visualized the inbox simulation to demonstrate how false positives and false negatives could affect users in practical settings.

Tursunai Turumbekova was responsible for Experiment 4, which integrated behavioral features derived from social engineering cues. She collected and cleaned the dataset used across all experiments, curated the deception prompt dictionary, and implemented SBERT-based feature extraction. She also engineered hybrid models that fused psychological trait scores with TF-IDF and metadata inputs to evaluate their impact on classification performance.

This project was a collaborative effort from start to finish. All team members actively participated in project planning, troubleshooting, and shared decision-making throughout. While each person led distinct components of the work, the project's success relied on close coordination and collective problem-solving across experimentation, analysis, and documentation.

References

Abdelhamid, N., Ayesh, A., & Thabtah, F. (2014). Phishing detection based on hybrid intelligent model. *Expert Systems with Applications, 63*, 321–332. https://doi.org/10.1016/j.eswa.2016.07.037

Abobor, M., & Josyula, D. P. (2023). SOCIALBERT: A transformer-based model used for detection of social engineering characteristics. In *2023 International Conference on Computational Science and Computational Intelligence (CSCI)* (pp. 174–178). IEEE. https://doi.org/10.1109/CSCI62032.2023.00033

Akdemir, A., & Yenal, S. (2021). Content analysis of COVID-19 themed phishing emails. Security Journal, 34(4), 433–449. https://doi.org/10.1057/s41284-021-00294-3

Aslam, M., Riaz, T., & Shafait, F. (2024). Lightweight Fine-tuning of BERT for Large-Scale Phishing Detection. Proceedings of the 32nd International Conference on Computational Linguistics.

Bountakas, A., & Xenakis, C. (2022). HELPHED: A hybrid ensemble learning model for phishing email detection. *Computers & Security, 112*, 102514. https://doi.org/10.1016/j.cose.2021.102514

Butavicius, M., Parsons, K., & Ferguson, L. (2022). The impact of time pressure and deception cues on phishing email detection. Journal of Cybersecurity, 8(1), 1–14. https://doi.org/10.1093/cybsec/tyac003

Champa, A. I., Zibran, M. F., & Rahayu, W. (2024a). Why phishing emails escape detection: A closer look at the failure points. In *2024 12th International Symposium on Digital Forensics and Security (ISDFS)*. IEEE. https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=10527344

Champa, A. I., Zibran, M. F., & Rahayu, W. (2024b). Curated datasets and feature analysis for phishing email detection with machine learning. In *2024 3rd IEEE International Conference on Computing and Machine Intelligence (ICMI)*.

https://www2.cose.isu.edu/~minhazzibran/resources/MyPapers/Champa_ICMI2024_Published.pdf

Cormack, G. V., & Lynam, T. (2007). Online supervised spam filter evaluation. *ACM Transactions on Information Systems, 25*(3), 1–35. https://doi.org/10.1145/1247715.1247717

Deloitte. (n.d.). 91% of all cyber attacks begin with a phishing email to an unexpected victim. *Deloitte Malaysia*. Retrieved March 20, 2025, from https://www2.deloitte.com/my/en/pages/risk/articles/91-percent-of-all-cyber-attacks-begin-with-a-phishing-email-to-an-unexpected-victim.html

Fette, I., Sadeh, N., & Tomasic, A. (2007). Learning to detect phishing emails. In *Proceedings of the 16th International Conference on World Wide Web* (pp. 649–656). https://doi.org/10.1145/1242572.1242660

IBM Security. (2024). X-Force Threat Intelligence Index 2024. IBM Corporation. https://www.ibm.com/reports/threat-intelligence

Microsoft Threat Intelligence. *Microsoft Digital Defense Report: Building and Improving Cyber Resilience*. Oct. 2023.

Otieno, F., Owuor, P., & Njagi, K. (2023). Transformer-Based Email Classification for Phishing Detection. International Journal of Computer Applications, 186(3), 1–8. https://doi.org/10.5120/ijca2023922667

Ramesh, S., Zhang, C., & Andreas, J. (2023). Feature-level adversarial robustness in natural language processing. *Transactions of the Association for Computational Linguistics, 11*, 788–805. https://aclanthology.org/2023.tacl-1.47/

Saha, P., Patra, S., & Dey, N. (2023). Hybrid CNN–GRU based deep phishing email detection using embedded linguistic features. *Journal of King Saud University – Computer and Information Sciences, 35*(6), 3725–3736. https://www.preprints.org/manuscript/202202.0024/v1

Schmitt, D., & Flechais, I. (2024). *Generative AI and phishing: Leveraging behavioral cues to identify AI-generated attacks*. Computers & Security, 140, 103355. https://doi.org/10.1016/j.cose.2024.103355

Shahriar, S., Mukherjee, A., & Gnawali, O. (2023). Improving phishing detection via psychological trait scoring. *arXiv preprint*. arXiv:2305.13263. https://arxiv.org/abs/2305.13263

Verma, R., & Das, A. (2017). What's in a URL? Using URLs to detect malicious websites. *International Journal of Information Security and Privacy, 11*(3), 1–14. https://dl.acm.org/doi/10.1145/3041008.3041016

Wen, Andy. "Email scams surge over the holiday — here's how Gmail keeps you safe." *Google Blog*, 18 Dec. 2024. https://blog.google/products/gmail/gmail-holidays-2024-spam-scam/

Appendix

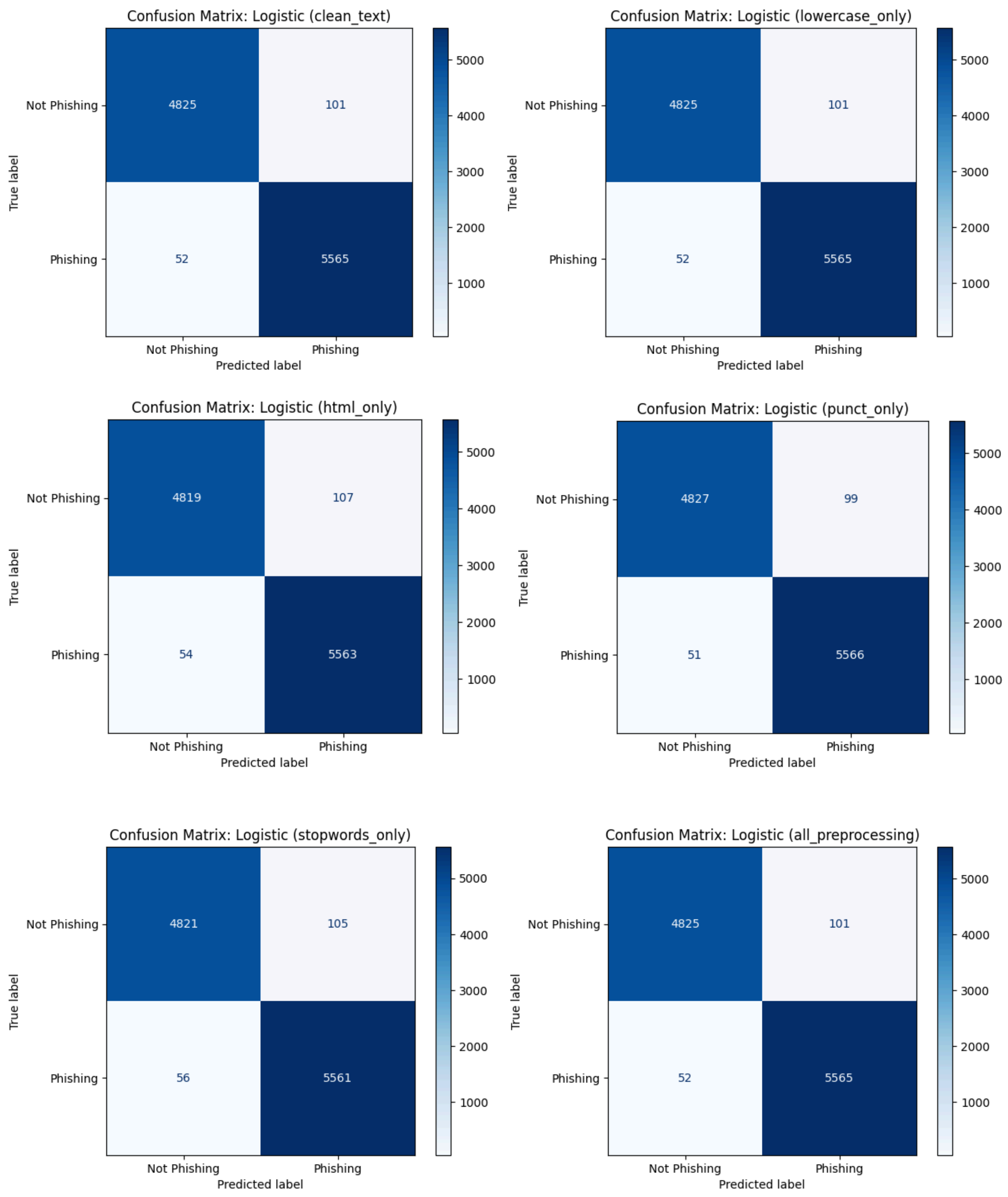


Figure A1. Confusion matrices for Logistic Regression (text-only pipelines)

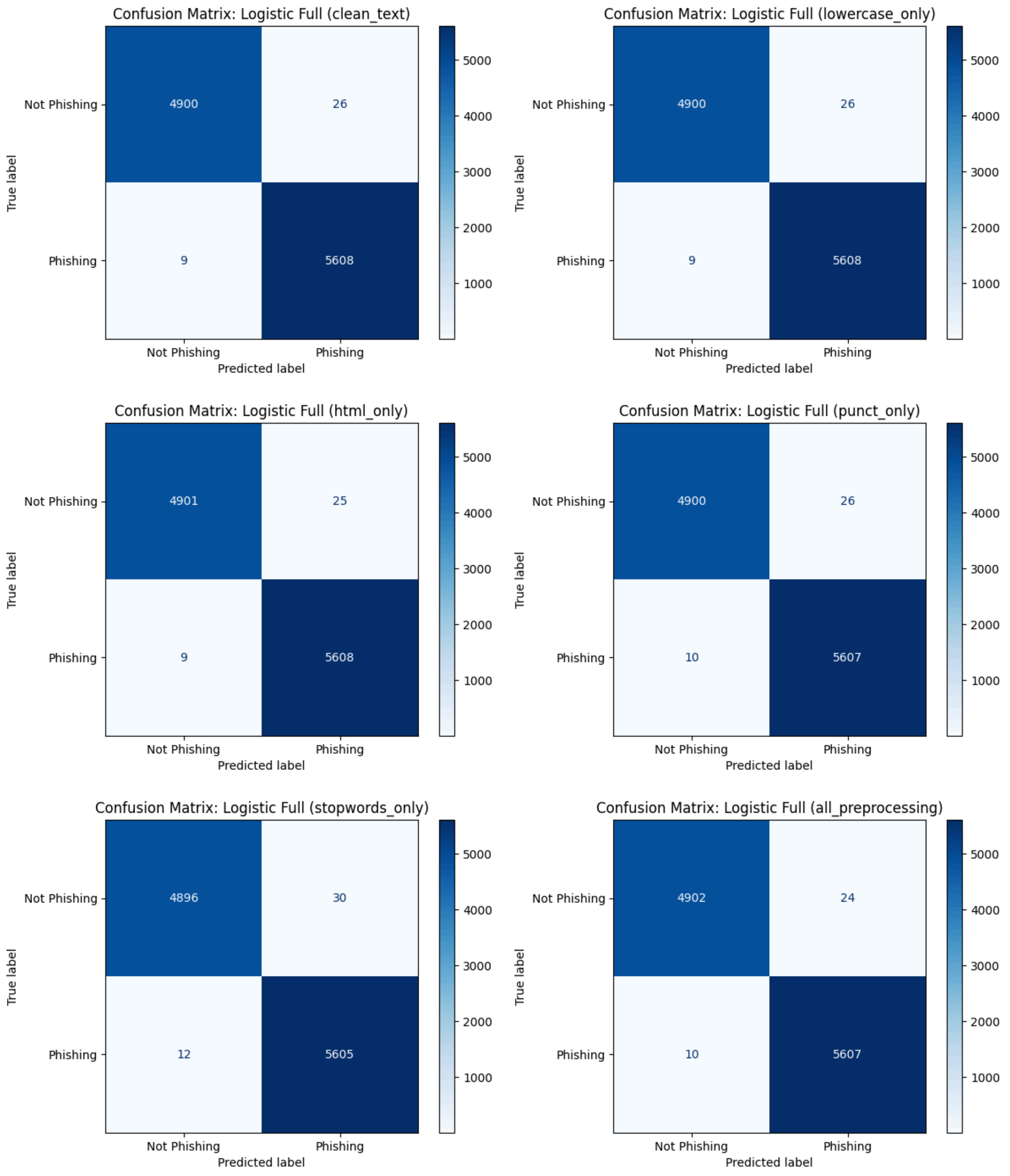


Figure A2. Confusion matrices for Logistic Regression (full-feature pipelines)

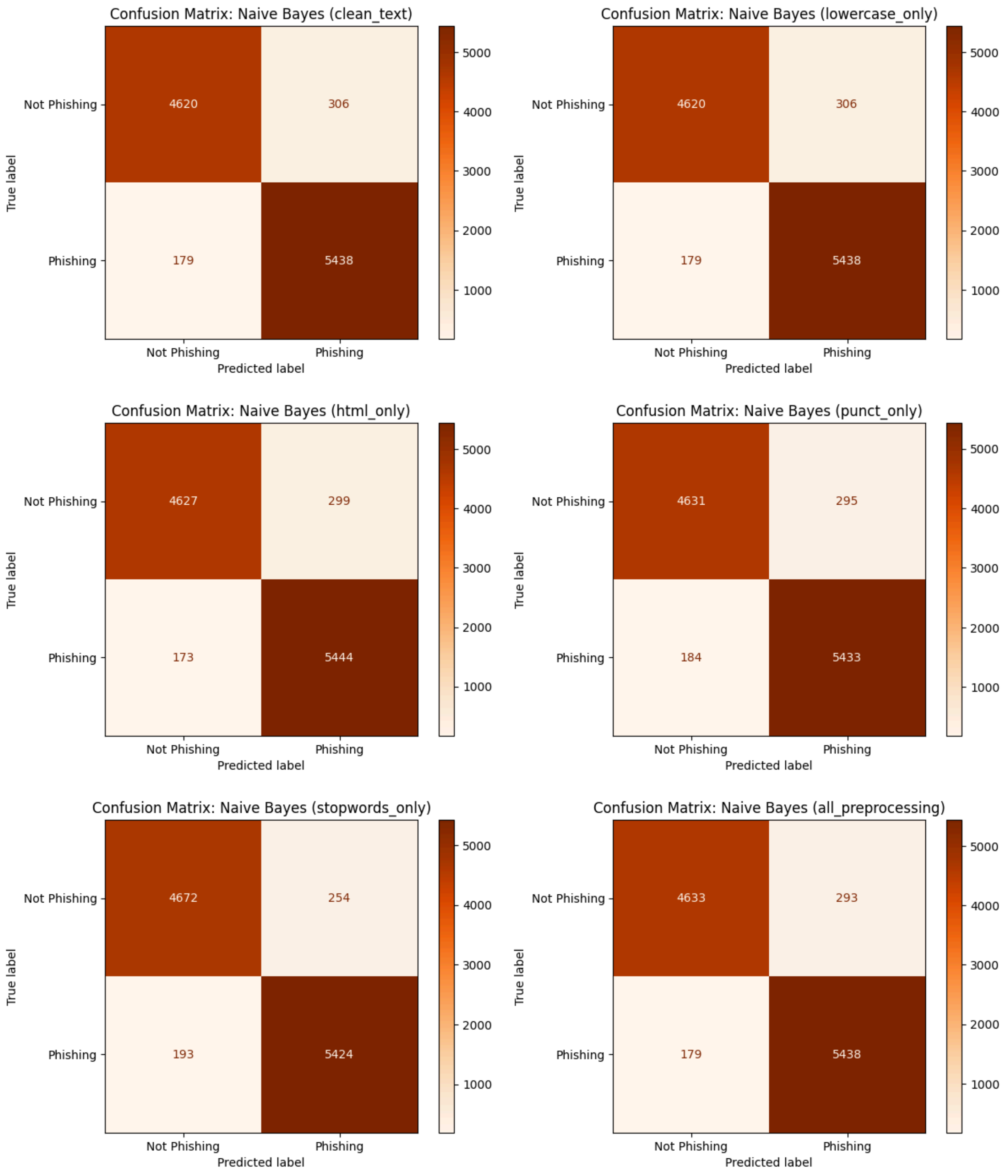


Figure A3. Confusion matrices for Naïve Bayes (text-only pipelines)

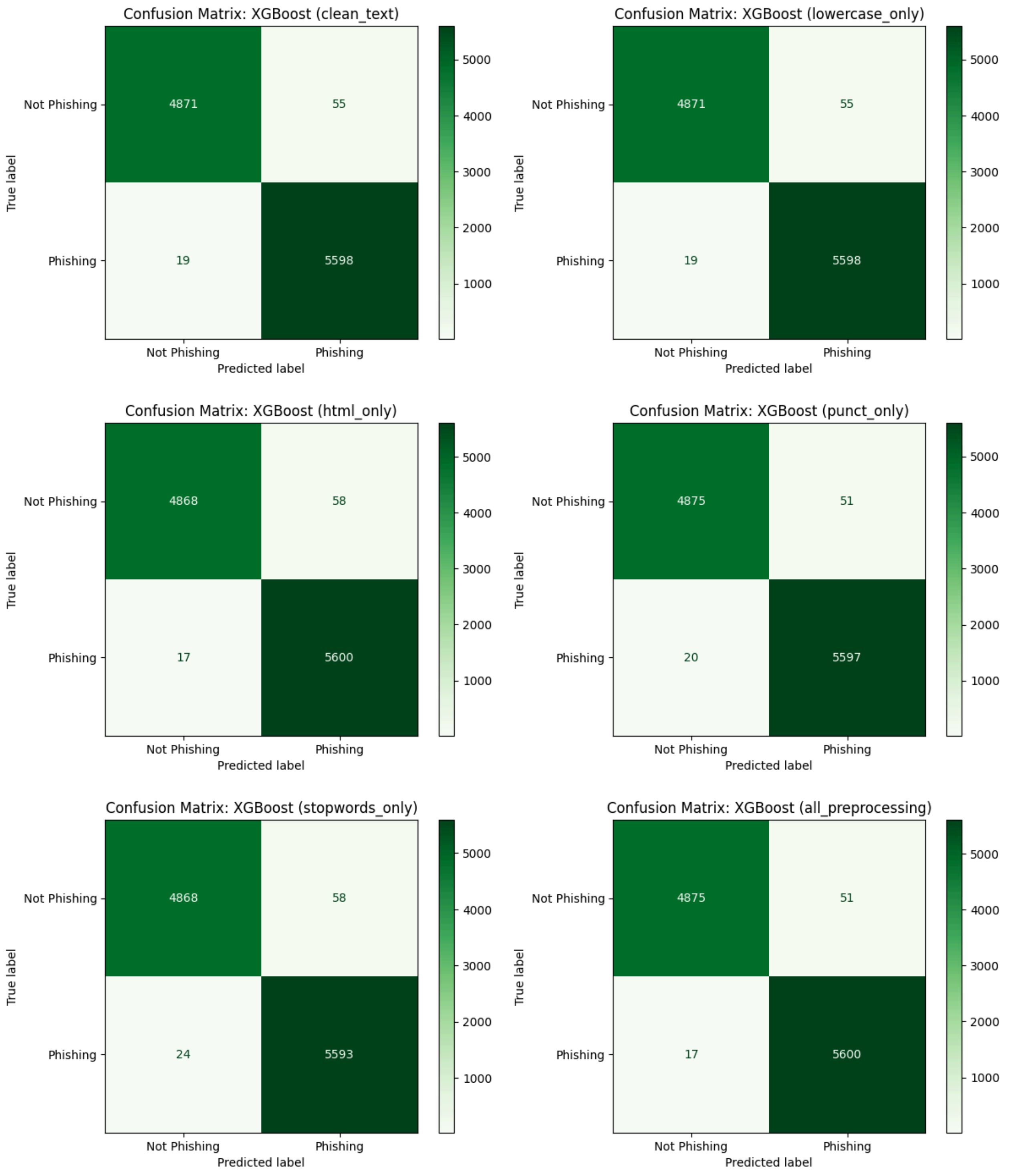


Figure A4. Confusion matrices for XGBoost (text-only pipelines)

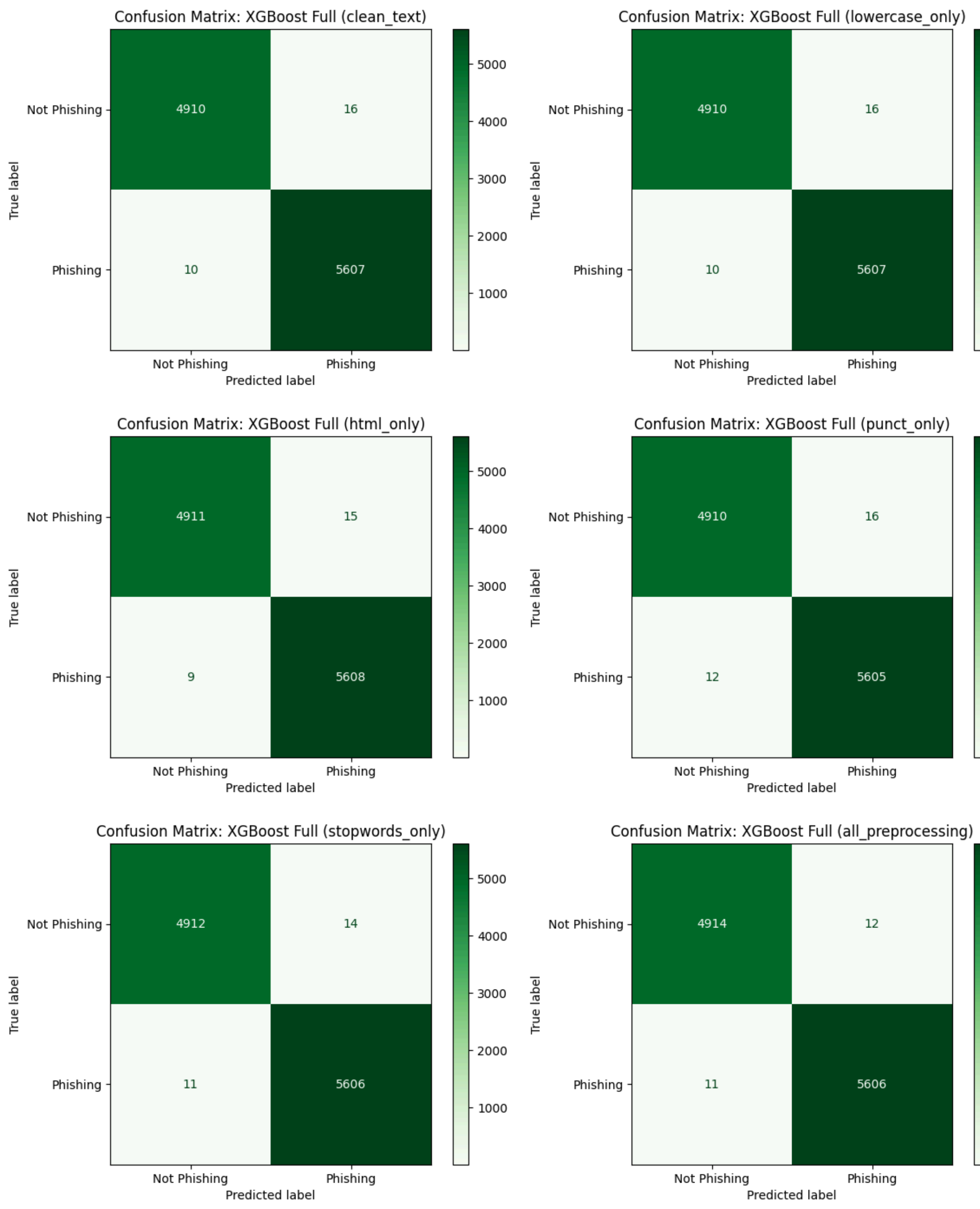


Figure A5. Confusion matrices for XGBoost (full-feature pipelines)

Experiment 4:

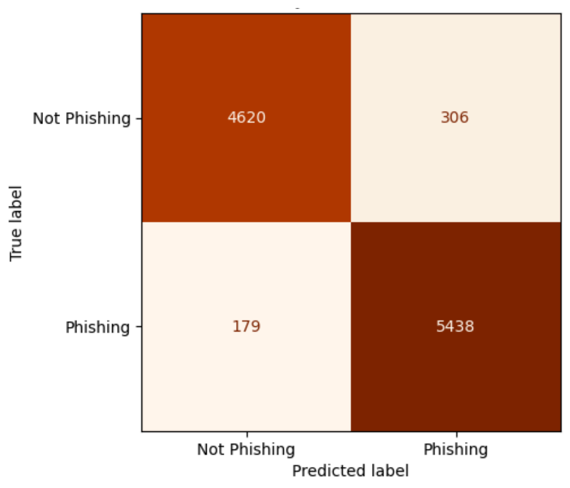


Figure A6. XGboost Confusion Matrix

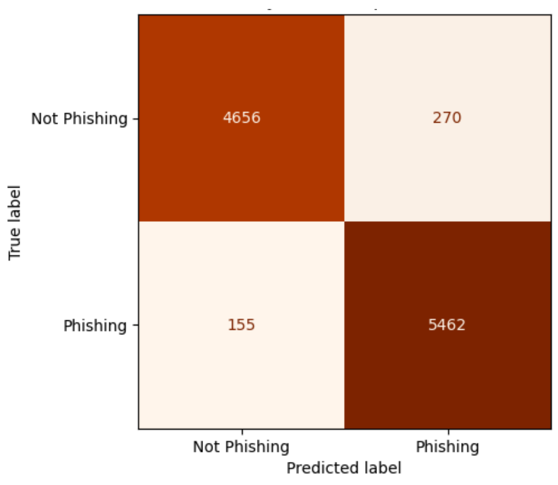


Figure A7. XGboost + Deception Features

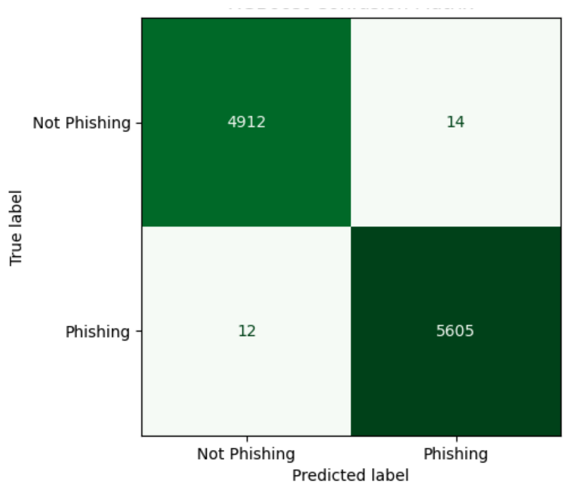


Figure A8. Naive Bayes Confusion Matrix

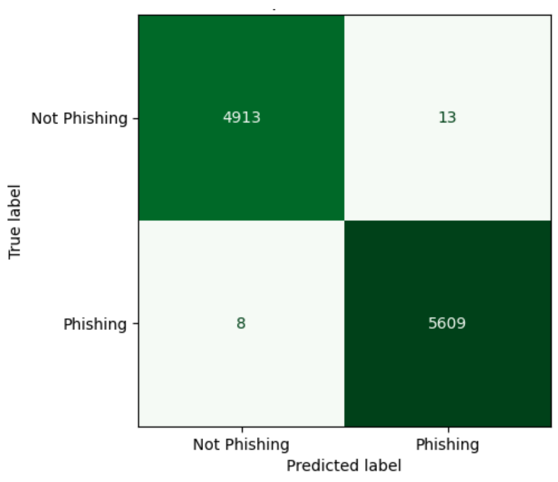


Figure A9. Naive Bayes + Deception Features

Table A1. Sample Prompts for Social Engineering Feature Extraction

Category	Example Prompt 1	Example Prompt 2
Urgency	Act immediately or your account will be closed.	Immediate action is required on your account.
Fear	Your account has been suspended due to suspicious activity.	Warning: your personal information is at risk.
Curiosity	Click here to uncover a hidden opportunity.	Don’t miss this once-in-a-lifetime chance.