# Tree-Based Detection of Malicious SMS Using Natural Language Processing Features and Metadata

MGSC401 - Final Project

Kaibo Zhang

Email: kaibo.zhang@mail.mcgill.ca

April 7, 2025

**Abstract**

This project develops a supervised classification framework for SMS phishing using a dataset of 930 real-world text messages enriched with text and metadata features. By combining natural language processing (NLP) techniques, such as token normalization, stemming, and bag-of-words encoding, with behavioral indicators like domain age, sender type, and temporal metadata, the study captures both linguistic and structural cues common in phishing attempts. A logistic regression model was first implemented for baseline comparison but exhibited limited performance due to sparsity and collinearity. In contrast, a tuned XGBoost model delivered substantially improved and balanced predictive performance across accuracy, recall, and F1 score and achieved the highest AUC, indicating strong probability calibration and ranking reliability. Model interpretation and feature importance analysis reveal that malicious messages often use newer domains and simulate urgent alerts in text, while subtle smishing attempts may evade detection due to tone or topic. The findings inform strategic defenses from real-time domain and sender validation to user education on less obvious phishing content.

## 1 Introduction

We evaluated classification models on the Smishtank Smishing Dataset [1], which contains 930 real-world SMS phishing messages labeled on a scale from 0 (benign) to 7 (highly malicious). Each message includes rich features such as text content, sender identifiers, embedded URLs, and metadata from VirusTotal and WHOIS. To support binary classification, the target variable was transformed such that all nonzero levels were labeled as malicious. Message texts were stemmed and vectorized using bag-of-words (BoW) encoding, while categorical fields like `Brand` were grouped into broader entity types. Timestamp variables were decomposed into components such as year, month, and minute. A baseline Logistic Regression model was constructed to establish a benchmark, followed by the development of an optimized XGBoost classifier. Performance was evaluated using 5-fold cross-validation and a separate test set, confirming substantial improvements over the baseline. This project demonstrates how structured preprocessing and model tuning can meaningfully enhance smishing detection and inform more effective detection strategies for mobile security applications.

## 2 Data Cleaning and Preprocessing

To standardize SMS content and retain smishing-relevant features, messages were lowercased and normalized by replacing emails, URLs, phone numbers, and brand names with standardized tokens (e.g., `<email>`, `<url>`, `<company>`), and informal shorthand like "u" was expanded. Spelling correction using the `hunspell` dictionary addressed typos and adversarial variants. After tokenization, stopwords were removed, and stemming was applied to unify word forms (e.g., "confirmed", "confirmation" → "confirm"). The text was encoded using a Bag-of-Words (BoW) representation, chosen over other semantic embedding methods like Word2Vec or BERT for its transparency and interpretability in short-form SMS data. We also engineered features from timestamps and brand metadata to surface behavioral patterns. Time fields (e.g., `Received Time`, `Domain Creation Date`) were decomposed into year, month, and hour. Missing values were set to -1 for structural consistency. Brand names were cleaned and grouped into six broader categories to reduce sparsity and highlight phishing strategies tied to familiar entities. These steps generated transparent and informative features for downstream classification tasks.

## 3 Exploratory Data Analysis

### 3.1 Dataset Description

The dataset includes 930 SMS messages labeled with a binary `Malicious` indicator, with 43% marked as malicious and 57% as benign. This balance reflects realistic user-submitted content, where both clearly harmful and ambiguous messages are reported. As shown in Figure 1, 69% of messages were sent from phone numbers—commonly used to mimic trusted contacts—while 21% came from email-to-text sources, likely to bypass spam filters. Brand mentions centered on well-known consumer and financial institutions like Amazon and Bank of America, consistent with phishing campaigns that exploit brand familiarity. Messages most often referenced account alerts or delivery issues, designed to generate urgency. Additionally, over 60% of messages contained deceptive URLs, indicating attempts to evade blacklist detection. These patterns reflect the use of social engineering and spoofed infrastructure typical of smishing attacks.

### 3.2 Breakdown of Categorical Levels by Malicious Label

Figure 2 reinforces the observation from data description. Malicious messages are more concentrated within finance-related brand clusters and disproportionately rely on sender types such as phone numbers and email-to-text conversions, while benign messages show more dispersion. The presence of a URL is common in both classes, but malicious samples more frequently involve deceptive structures such as random domains or obfuscated subdomains. Missing domain creation or update dates are also more prevalent among smishing messages, highlighting the use of disposable or untraceable infrastructure. Message category distributions reveal that while account alerts and advertisements appear across both classes, malicious texts are more likely to exploit delivery and finance-related themes—common vectors for impersonation and data harvesting.

### 3.3 Distribution of Continuous Variables by Malicious Label

Figure 3 illustrates the distributions of continuous features related to SMS timing and domain metadata, stratified by message label. Temporal variables such as `Received Hour` and `Received Minute` display relatively even distributions across both malicious and non-malicious messages. Notably, malicious texts show a slight concentration during working hours, potentially reflecting attacker attempts to mimic legitimate institutional contact times. Similarly, `Received Month` and `Received Day` lack strong seasonality patterns but are consistent with opportunistic message delivery strategies observed in phishing campaigns. This pattern suggests that attackers do not rely on specific calendar events but rather distribute messages broadly across time to maximize exposure and avoid detection tied to predictable schedules. Domain-related features (`Domain Created/Updated`) highlight notable artifacts. A significant portion of domain metadata is missing or defaulted to invalid years (e.g., pre-2000), particularly in malicious samples. This aligns with common tactics where attackers register disposable domains or spoof metadata to avoid traceability. As domain age is often correlated with credibility, the presence of recently created or non-resolvable domains in phishing attempts supports its utility as a risk signal. The asymmetries are driven by deliberate obfuscation and synthetic data generation, suggesting that integrating domain age with indicators of metadata validity may improve model robustness against evasive attack behaviors.

### 3.4 Token Distribution and Lexical Patterns

Figure 4 shows that malicious messages tend to be longer, both in token and character count, with greater variability. This pattern likely reflects attempts to imitate legitimate communication through added detail, such as instructions or embedded links. These trends are further supported by word frequency analysis (Table 2). Details such as company names, clickable links, and polite or directive terms like “please” and “inform” appear frequently in both malicious and benign messages. The substantial overlap in vocabulary highlights the inherent challenge of smishing detection, where malicious messages are intentionally crafted to resemble benign ones. As a result, relying on individual word usage is insufficient for accurate classification. This overlap limits the capability of simple models with linear decision boundaries, which struggle to capture the nonlinear, context-dependent nature of language. Consistent with linguistic theory, meaning often emerges through word interactions rather than isolated terms. This emphasizes the need for modeling with richer structural and sequential patterns.

## 4 Methodology

### 4.1 Logistic Regression Baseline

We trained a logistic regression model as a baseline using all available features and an 80-20 train-test split. Chosen for its interpretability and effectiveness in binary classification, the model achieved only slightly better than random performance, with moderate precision and notably low recall and F1 scores even on the training set (Tables 3). This suggests a conservative classifier that flags fewer messages as malicious, potentially missing subtle smishing cues, as expected from EDA. The trade-off is problematic in security contexts where false negatives

carry high risk. Diagnostic analysis revealed instability caused by high dimensionality from bag-of-words encoding, near-zero variance features, and multicollinearity (Table 5). These results highlight the limitations of linear models in capturing the complex interactions inherent in deceptive text.

### 4.2 Modeling with XGBoost

To address the limitations of the baseline logistic regression, we turned to XGBoost, a gradient-boosted decision tree method designed to handle sparse data, variable interactions, and high-dimensional feature sets more effectively. XGBoost incorporates automatic feature selection through its use of decision trees, which naturally ignore uninformative or redundant predictors. This provides an essential advantage, given the sparsity introduced by the BoW encoding. In contrast to ensemble methods like bagging that aggregate parallel models to reduce variance, boosting builds models sequentially, with each new tree trained to correct the errors made by the previous ones. Malicious messages often closely resemble legitimate ones, and the signal-to-noise ratio is low. Boosting enables the model to adapt to hard-to-classify examples by refining decision boundaries incrementally rather than diluting them through averaging. As shown in Tables 3 and 4, XGBoost trained with default hyperparameters significantly outperformed the baseline across all evaluation metrics, particularly recall and F1 score. However, there remains room for improvement, as performance gains plateaued beyond a certain depth and complexity, suggesting potential benefits from additional fine-tuning efforts for regularization.

### 4.3 Model Fine-Tuning and Hyperparameter Optimization

#### 4.3.1 Tuning Strategies Around Default Parameters

In smishing detection, missing a positive label is costly, as failing to catch a malicious message could expose users to financial or personal risk. While accuracy and precision are valuable, recall ensures that the model captures the largest number of true malicious cases, even at the cost of misclassifying some benign messages. In high-risk domains like mobile security, minimizing false negatives is often more important than minimizing false positives, making recall a suitable objective for tuning the XGBoost model.

The first tuning grid (Table 6) was designed to explore a controlled range around XGBoost's default settings, focusing on parameters most relevant to modeling high-dimensional and sparse data. The tuned model selected a total of 100 boosting rounds (i.e., 100 trees), which provided sufficient capacity to learn meaningful patterns without allowing the model to overfit to noise of rare token patterns. Each tree was limited to three layers of splits, indicating a preference for shallow trees that limit overfitting while capturing essential word interactions. The model performs a split only when the improvement in the loss function exceeds 5, imposing a meaningful threshold that suppresses weak, data-specific splits and encourages simpler, more robust decision rules. In addition, tree creations considered only half of the available features, thereby improving generalization and reducing reliance on spurious tokens. Full-row sampling (subsample = 1) was retained to preserve context from each message during training. A minimum child weight of 1 allowed the model to split on terms as long as the resulting node contained

at least one instance, helping it detect subtle, low-frequency patterns common in smishing content. As shown in Table 3, this combination of parameter values yielded strong, balanced performance across all evaluation metrics, supporting its suitability for this text-based classification task.

#### 4.3.2 Balancing Recall and Reliability

Building on insights from the first grid, we narrowed the parameter space around high-performing values to further control overfitting (Table 7). This second grid led to a model with the highest recall on the test set, but as seen in Tables 3 and 4, it underperformed in precision and F1 score compared to V1. This indicates that while V2 is more aggressive in capturing malicious messages, it sacrifices predictive reliability by generating more false positives. This tradeoff is also visualized in Figure 5, where ROC curves compare the true positive rate (sensitivity) against the false positive rate across thresholds. At the same true positive rate, the model V2 tends to classify more benign messages as malicious. The Area Under the Curve (AUC) quantifies the model's ability to rank positive instances higher than negative ones. XGBoost V1 achieved the highest AUC (0.846), indicating superior discriminative ability and calibration as a probabilistic classifier. In contrast, V2's AUC (0.824) was lower, confirming that despite higher recall, its overall ranking performance was less optimal. Taken together, these results highlight the importance of not optimizing a single metric in isolation and support the selection of V1 for its balanced and reliable performance.

## 5 Results

### 5.1 Model Performance

As shown in Table 3 and Table 4, XGBoost V1 offers balanced performance across accuracy, precision, recall, and F1 score—an indicator of strong generalization rather than overfitting to training patterns. This consistency is especially important for smishing detection, where both missed threats and false alarms carry serious consequences. V1 reliably identifies malicious SMS while minimizing false positives, achieving a high F1 score that reflects strength in both precision and recall. Its ability to perform well across multiple metrics allows it to capture both obvious and subtle phishing patterns. The ROC curve in Figure 5 further underscores this model's strength from a probabilistic standpoint. V1 maintains a higher true positive rate with fewer false positives across thresholds, and its superior AUC reflects stronger class separation. This superior rank-ordering capability means one can set a decision threshold for V1 with greater confidence. For instance, if a deployment demands higher recall to catch more attacks, V1 can achieve that with relatively smaller sacrifices in precision and vice versa. In a real-world smishing defense system, this translates to more dependable threshold tuning and risk scoring. Security teams can trust V1's predicted probabilities as meaningful risk scores, allowing them to adjust the system's sensitivity to the desired level of caution without the model becoming erratic or missing threats. In summary, XGBoost V1's balanced performance across metrics and its superior AUC-driven discrimination make it the most robust choice for deployment in a domain where both false negatives and false positives carry significant consequences.

### 5.2 Understanding Malicious Text Characteristics

Figure 6 provides concrete insight into the linguistic and structural traits that distinguish smishing messages. Words like "link," "please," and "incomplete" consistently appear as high-gain features, reflecting the persuasive tone and urgency attackers often use to drive user engagement. Temporal features such as message hour and domain update year also rank highly, indicating that attackers often operate outside regular business hours and use freshly registered domains to bypass detection. Notably, terms like "account," "tax," and "inform" tend to appear alongside impersonated brand categories (e.g., government or nonprofit institutions), emphasizing attackers' tendency to exploit institutional trust in gaining victims' trust, coinciding with insights gained during EDA. Compared to more recall-optimized models, the more reliable V1 model emphasizes structured patterns and known phishing indicators rather than idiosyncratic tokens, providing a stable and interpretable foundation for policy and defense design. For instance, V1 prioritizes "Domain Created Year", a feature tied to the longevity and legitimacy of a sender's web infrastructure. Malicious domains are often newly registered, making this a reliable fraud signal. In contrast, V2 emphasizes "Received Year", which is more indicative of submission timing than message content. While this may help optimize recall on recent attack patterns, it risks capturing dataset-specific temporal biases, limiting generalizability to future or unseen threats.

The decision paths (Figure 7) of the first few trees further reveal attacker behavior. For instance, messages that reference account-related content and use random domain URLs are assigned higher malicious probabilities, reinforcing well-documented tactics like account spoofing and redirection via disposable domains. Likewise, messages sent at off-hours and referencing domains created recently are flagged more aggressively, suggesting that malicious actors target windows when users are less alert and monitoring infrastructure is lighter. From a managerial perspective, these insights highlight specific rules or monitoring triggers that can be implemented within spam filters or enterprise firewalls, such as blocking SMS from suspicious domains registered within the last year or flagging off-hour bulk messages with urgent content.

### 5.3 Predictions and Managerial Insights

To interpret the XGBoost model's behavior and gain qualitative insights into its decision logic, we examined selected predictions from the test set (Table 8), including both correct and incorrect classifications across malicious and benign classes. The first correctly predicted benign message involves a customer service alert originating from a known institution and contains typical service-oriented language without abnormal urgency. The model assigned a low probability of maliciousness (0.31), suggesting confidence in classifying legitimate transactional messages when structural cues are non-threatening and links are absent. Similarly, the correctly identified malicious message features a deceptive delivery prompt and an embedded link. The probability was well above the decision threshold (0.5), demonstrating the model's ability to recognize urgent directives and unresolved action requests, especially when paired with delivery-related keywords. In contrast, one false positive message was flagged as malicious with high confidence (0.54) despite being benign. This message referenced a prepaid debit card and

included financial institution language. The model's overreaction to these financial keywords suggests that the presence of transactional and account-related terminology—even when legitimate—may push probabilities upward. While this is operationally safer than missing a true threat, it may lead to false alarms that reduce user trust. For managers, this highlights the need to fine-tune alert thresholds and possibly augment model outputs with contextual metadata, such as known sender registrars or prior message frequency, before triggering automated actions.

On the other hand, the false negative case is more concerning. The message contained an encouraging tone and a call to complete an "application" process but lacked strong phishing indicators like financial urgency. The model assigned a low malicious probability (0.35), resulting in a missed detection. This suggests that subtle smishing tactics that avoid traditional trigger terms while mimicking routine onboarding or promotional language remain difficult for even well-calibrated models to detect. From a public awareness standpoint, this calls for updated educational messaging that warns users about positive-tone smishing—texts that sound helpful or congratulatory but still request action. To improve robustness, organizations should consider incorporating domain knowledge into the feature space, such as sender reputation scoring or click-through history, to balance recall and precision based on risk level. Additionally, dynamic thresholding tied to message type (e.g., finance vs. marketing) could allow for greater flexibility in classification without overloading users with false positives. In operational deployment, human-in-the-loop feedback mechanisms may also help the model learn from edge cases where text tone or style deviates from known malicious archetypes. Ultimately, while the model performs reliably on high-risk patterns, continued refinement and contextual layering are essential for capturing the evolving and often ambiguous nature of social engineering threats.

## 6 Discussion and Conclusion

This study introduced an interpretable, feature-rich framework for detecting malicious SMS by combining NLP with metadata and tree-based modeling. While logistic regression served as a baseline, it struggled with high-dimensional, sparse input. In contrast, the tuned XGBoost model offered strong generalization, achieving high recall, balanced precision, and superior AUC, making it effective for both classification and probabilistic risk scoring. Key managerial insights emerged from the model's behavior and feature importance. Indicators such as newly registered domains, urgent financial cues, and delivery language were common in malicious messages, reinforcing the need for real-time detection mechanisms and targeted employee or user education. Misclassified examples showed that smishing often imitates legitimate language, suggesting training should highlight subtle cues like onboarding prompts or overly polite tone. The ability to fine-tune decision thresholds allows organizations to adapt detection strictness to their risk context, while XGBoost's interpretability supports transparent policy development such as flagging messages from suspicious senders or domains. Integrating such models into layered security systems, coupled with adaptive feedback loops, can enhance both detection reliability and user trust. Overall, the findings support practical, data-driven strategies to combat evolving smishing threats.

# 7 Appendix

| MainText | Not Stemmed | Stemmed |
|---|---|---|
| Costco: Daniel, the code 42003 printed on your receipt from 10 came in 2nd in our Airpods draw: f2gpy.info/RzNKEws Zve | company denial the code number printed on your receipt from number came in 2nd in our air pods draw URL eve | compani denial code number print receipt number came 2nd air pod draw URL eve |
| wel01.us/r/rest05 WELLS FARGO(CS):Profile locked because of unusual activities, kindly restore. Reply STOP to unsubscribe | URL company cs profile locked because of unusual activities kindly restore reply company to unsubscribe | URL compani cs profil lock unusu activ kind restor repli compani unsubscrib |

Table 1: Examples of NLP-Processed Smishing Messages

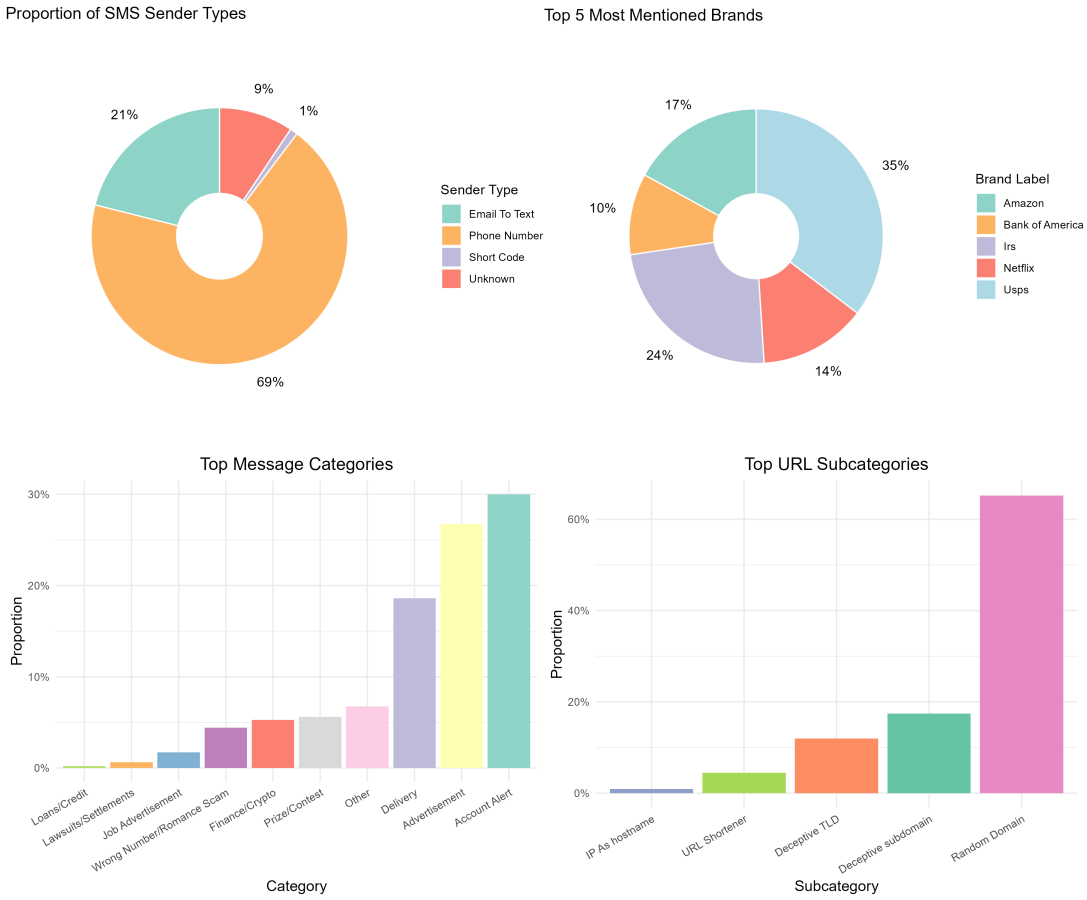


Figure 1: Matrix Plot of Metadata Distributions in Smishing Messages


Figure 2: Matrix Plot of Categorical Variable Distributions by Message Type

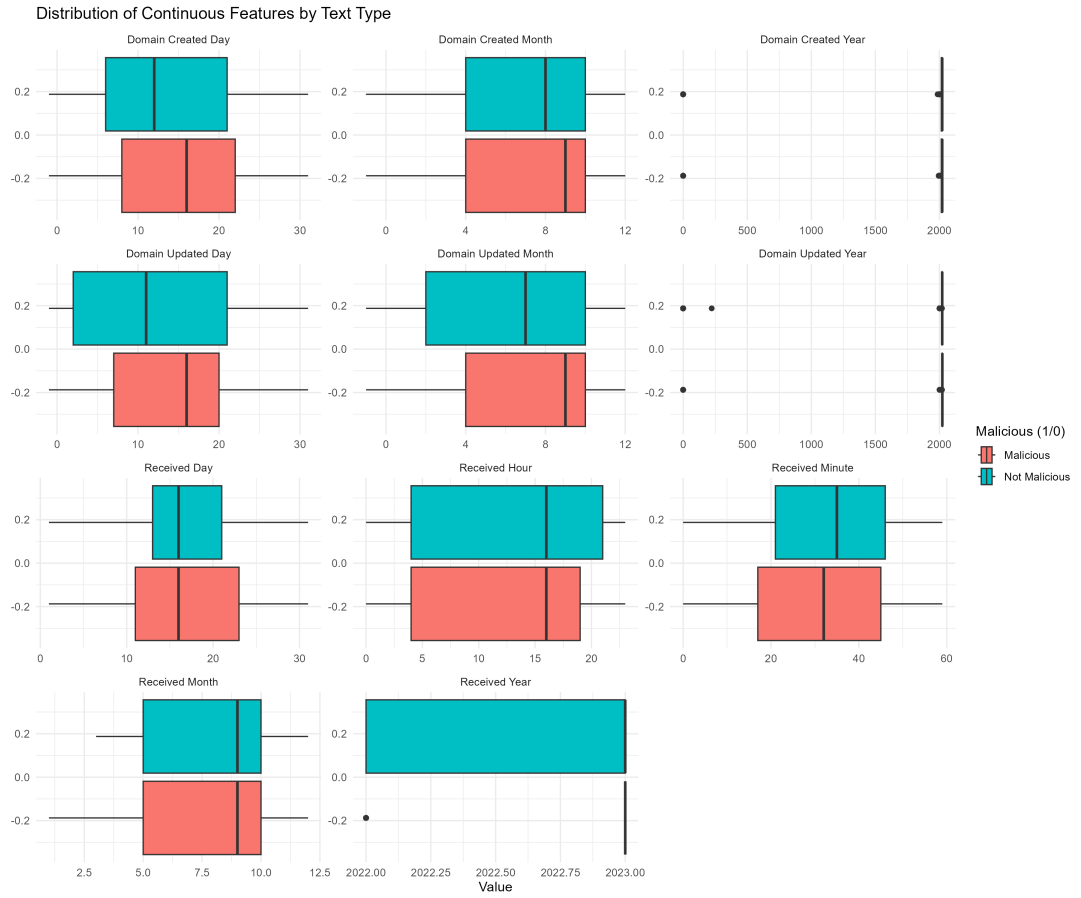


Figure 3: Matrix Plot of Continuous Variable Distributions by Message Type

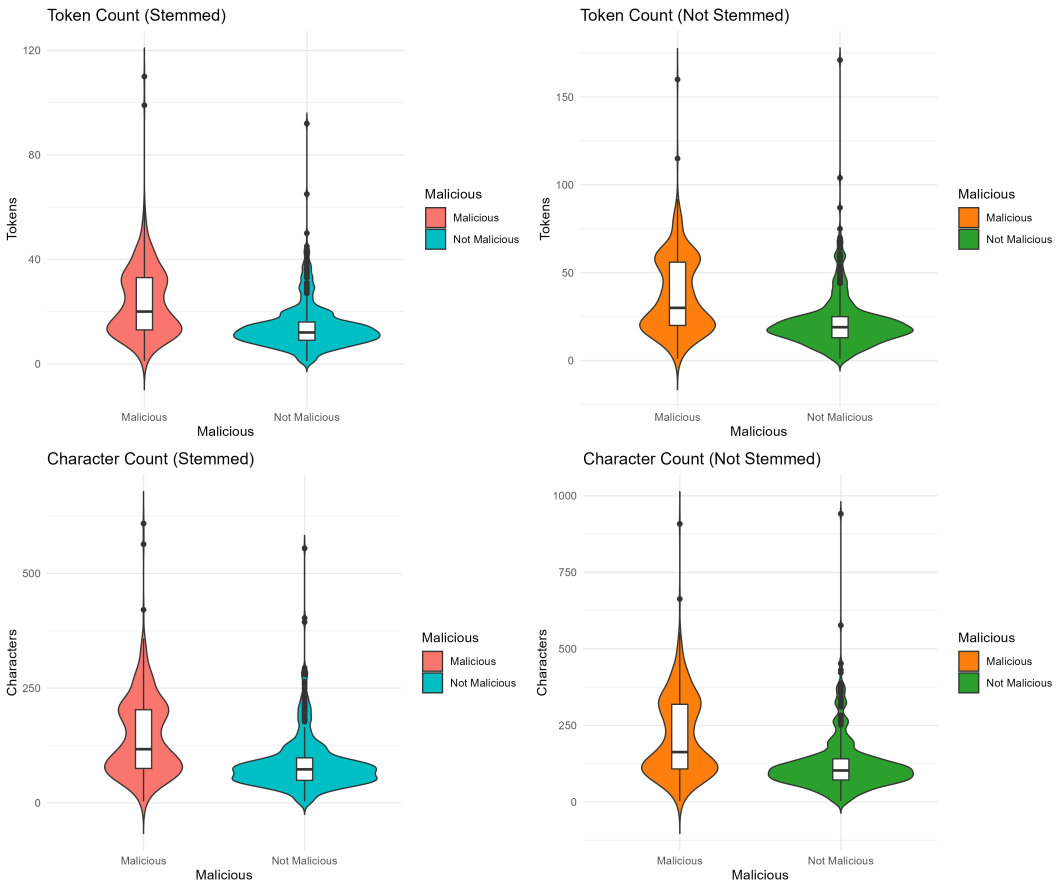


Figure 4: Matrix Plot of Token and Charcater Length Distributions by Message Type

| Rank | Top Malicious Word | Count | Top Non-Malicious Word | Count |
|---|---|---|---|---|
| 1 | number | 492 | url | 524 |
| 2 | compani | 404 | number | 448 |
| 3 | url | 384 | compani | 359 |
| 4 | link | 325 | account | 104 |
| 5 | pleas | 294 | pleas | 93 |
| 6 | inform | 242 | link | 91 |

Table 2: Top Word Occurrences in Malicious and Non-Malicious Messages

| Model | Accuracy | Precision | Recall | F1 Score |
|---|---|---|---|---|
| Logistic Regression (Baseline) | 0.564 | 0.614 | 0.488 | 0.539 |
| XGBoost (Default Parameters) | 0.736 | 0.620 | 0.723 | 0.665 |
| XGBoost (Tuned, Grid Search v1) | 0.737 | 0.522 | 0.797 | 0.628 |
| XGBoost (Tuned, Grid Search v2) | 0.728 | 0.462 | 0.829 | 0.592 |

Table 3: Cross Validation (k=5) Performance on the Training Set

| Model | Accuracy | Precision | Recall | F1 Score |
|---|---|---|---|---|
| Logistic Regression (Baseline) | 0.497 | 0.557 | 0.431 | 0.486 |
| XGBoost (Default Parameters) | 0.751 | 0.595 | 0.771 | 0.671 |
| XGBoost (Tuned, Grid Search v1) | 0.773 | 0.557 | 0.863 | 0.677 |
| XGBoost (Tuned, Grid Search v2) | 0.751 | 0.519 | 0.837 | 0.641 |

Table 4: Test Set Performance Summary

| Variable | Frequency Ratio | Percent of Uniqueness | Is Variance Near Zero |
|---|---|---|---|
| april | 0 | 0.1342282 | TRUE |
| assum | 0 | 0.1342282 | TRUE |
| detox | 0 | 0.1342282 | TRUE |
| japan | 0 | 0.1342282 | TRUE |
| ownership | 0 | 0.1342282 | TRUE |
| unsuccess | 0 | 0.1342282 | TRUE |
| cashback | 0 | 0.1342282 | TRUE |

Table 5: Example of Variables with Zero Variance in the Training Set

| Parameter | Values Explored |
|---|---|
| Number of Boosting Rounds (nrounds) | 100, 200 |
| Maximum Tree Depth (max_depth) | 3, 6 |
| Learning Rate (eta) | 0.1, 0.3 |
| Minimum Loss Reduction (gamma) | 0, 1, 5 |
| Column Subsampling per Tree (colsample_bytree) | 0.3, 0.5, 0.7 |
| Minimum Child Weight (min_child_weight) | 1, 5, 10 |
| Row Subsampling (subsample) | 0.7, 1.0 |

Table 6: Grid V1 used for XGBoost Hyperparameter Tuning

| Parameter | Values Explored |
|---|---|
| Number of Boosting Rounds (nrounds) | 100, 150, 200 |
| Maximum Tree Depth (max_depth) | 2, 3, 4 |
| Learning Rate (eta) | 0.05, 0.1, 0.15 |
| Minimum Loss Reduction (gamma) | 3, 5, 7 |
| Column Subsampling per Tree (colsample_bytree) | 0.4, 0.5, 0.6 |
| Minimum Child Weight (min_child_weight) | 1, 2, 3 |
| Row Subsampling (subsample) | 0.9, 1.0 |

Table 7: Refined Grid V2 for XGBoost Hyperparameter Tuning

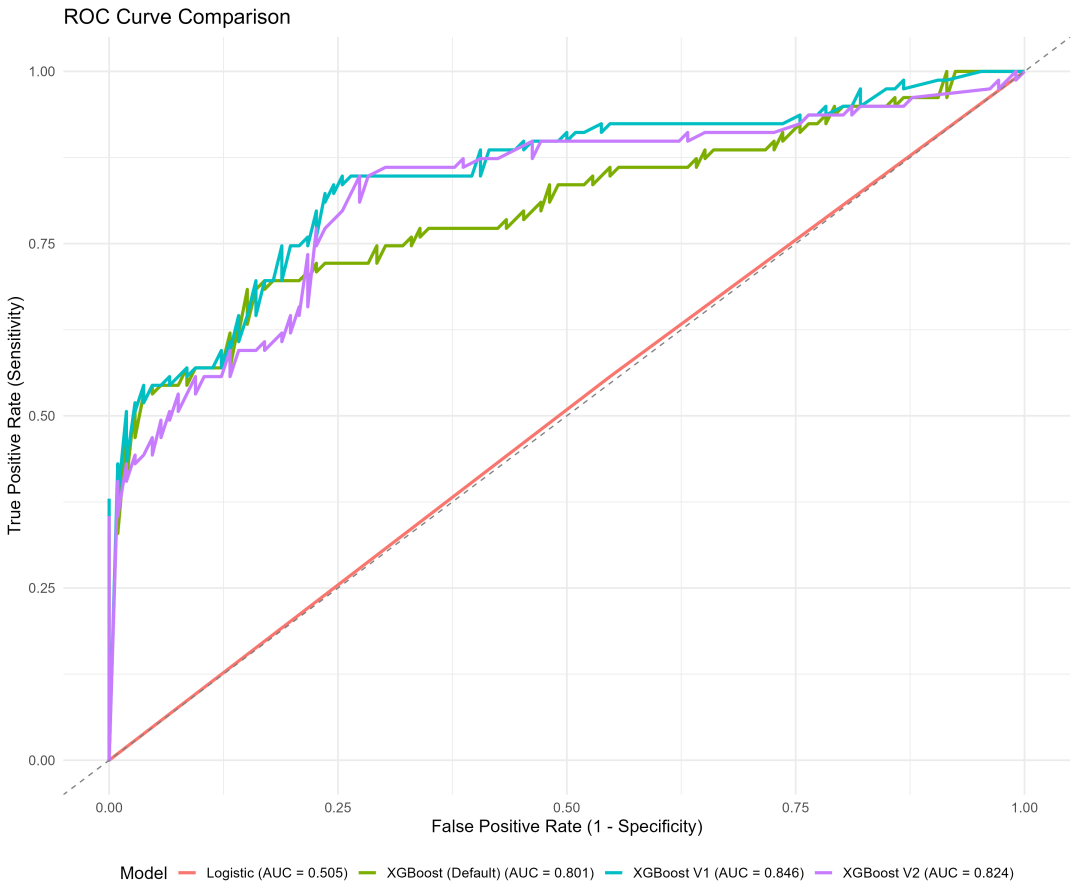


Figure 5: ROC curves Comparing the Probability Calibration of All Models

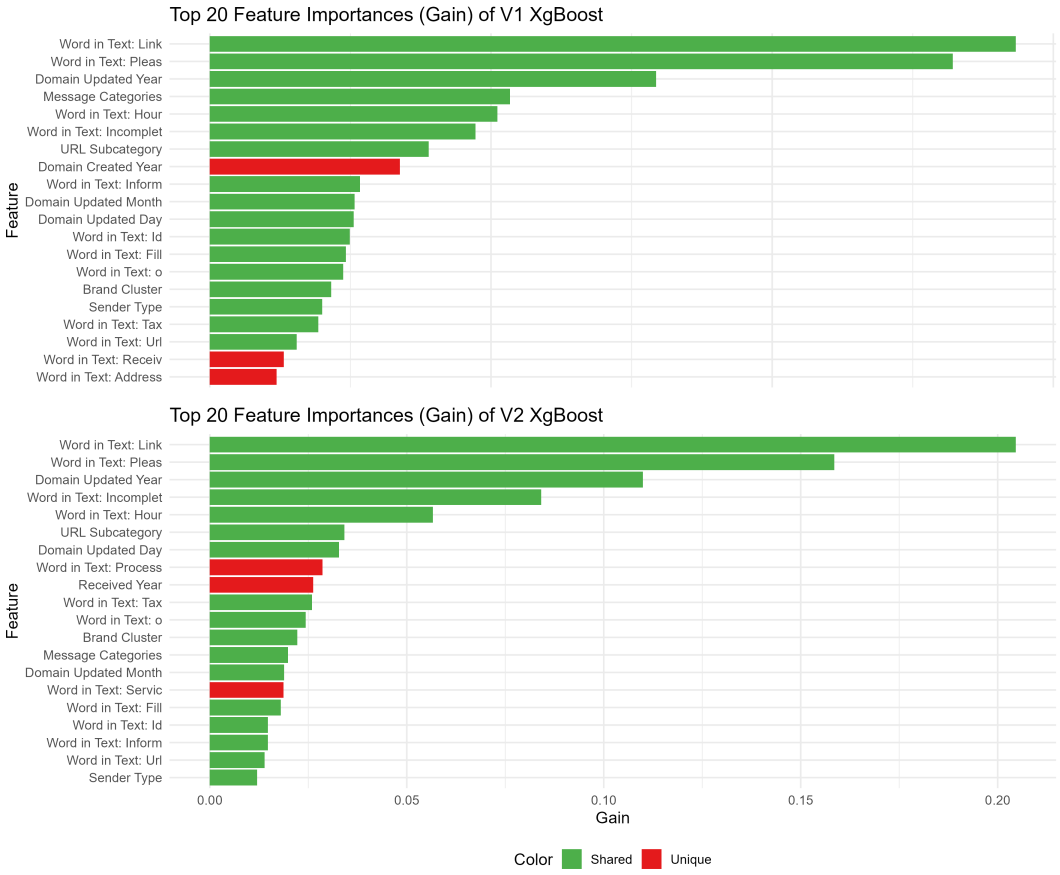


Figure 6: Feature Importance of XGBoost Model V1 and V2

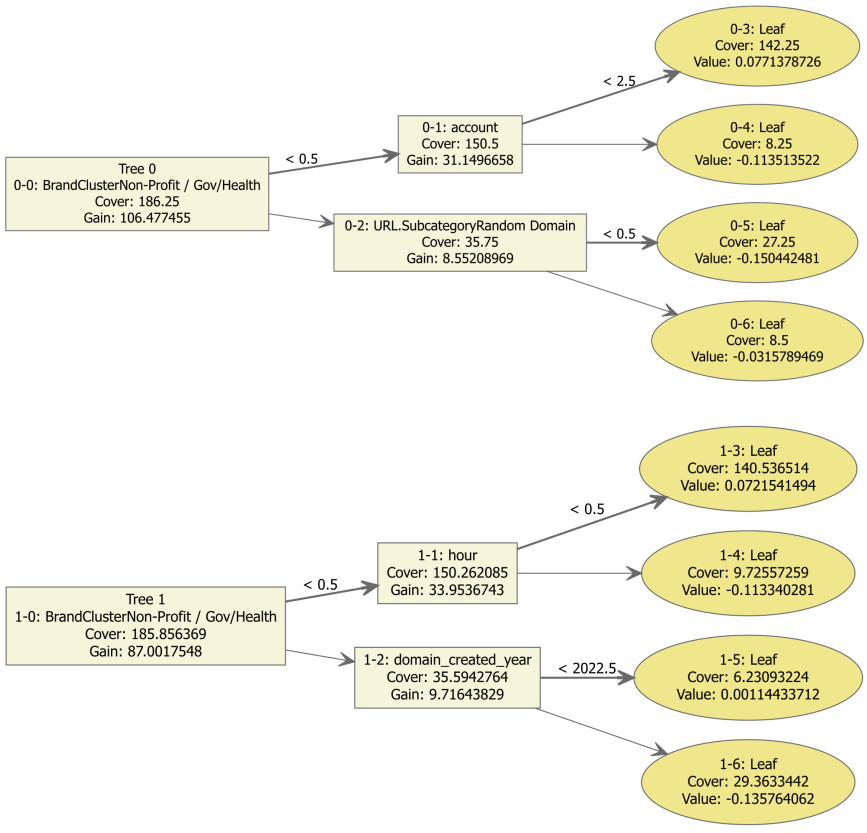


Figure 7: Up: First Tree Extracted from XGBoost V1. Down: Second Tree Extracted from XGBoost V1.

| Original Text | Actual | Predicted | Probability | Time Received | Domain Creation | Domain Update |
|---|---|---|---|---|---|---|
| URL company cs profile locked because of unusual activities kindly restore reply company to unsubscribe | 0 | 0 | 0.3058 | 04/02/2022, 03:03:00 | 08/30/2021 | 08/30/2021 |
| you have filled in an incorrect delivery address please update your address to the post office URL | 1 | 1 | 0.6344 | 09/21/2022, 05:11:23 | 08/04/2022 | 09/10/2022 |
| company company alert services your company prepaid company debit card number has been locked and under review due to unusual activities on the account click the secured website link provided below to review your correct card information to reactivate and reopen your card now URL | 0 | 1 | 0.5411 | 12/20/2022, 07:07:37 | 08/19/2022 | 09/30/2022 |
| amazing news your application has been accepted finish the process URL | 1 | 0 | 0.3451 | 04/03/2022, 15:31:07 | 11/09/2021 | 11/09/2021 |

Table 8: Examples of Correct and Incorrect Predictions with Metadata and Model Probabilities

## References

[1] Daniel Timko and Muhammad Lutfor Rahman. Smishing dataset i: Phishing sms dataset from smishtank.com. In *Proceedings of the Fourteenth ACM Conference on Data and Application Security and Privacy*, pages 289–294. ACM, 2024.