Talking Like a Phisher: LLM-Based Attacks on Voice Phishing Classifiers

WENHAO LI[1], SELVAKUMAR MANICKAM[1]✉, YUNG-WEY CHONG[2] and SHANKAR KARUPPAYAH[1]

[1] Cybersecurity Research Centre, Universiti Sains Malaysia, Pulau Pinang, Malaysia
[2] School of Computer Sciences, Universiti Sains Malaysia, Pulau Pinang, Malaysia
wenhaoli@ieee.org, {selva, chong, kshankar}@usm.my

Abstract. Voice phishing (vishing) remains a persistent threat in cybersecurity, exploiting human trust through persuasive speech. While machine learning (ML)-based classifiers have shown promise in detecting malicious call transcripts, they remain vulnerable to adversarial manipulations that preserve semantic content. In this study, we explore a novel attack vector where large language models (LLMs) are leveraged to generate adversarial vishing transcripts that evade detection while maintaining deceptive intent. We construct a systematic attack pipeline that employs prompt engineering and semantic obfuscation to transform real-world vishing scripts using four commercial LLMs. The generated transcripts are evaluated against multiple ML classifiers trained on a real-world Korean vishing dataset (KorCCViD) with statistical testing. Our experiments reveal that LLM-generated transcripts are both practically and statistically effective against ML-based classifiers. In particular, transcripts crafted by GPT-4o significantly reduce classifier accuracy (by up to 30.96%) while maintaining high semantic similarity, as measured by BERTScore. Moreover, these attacks are both time-efficient and cost-effective, with average generation times under 9 seconds and negligible financial cost per query. The results underscore the pressing need for more resilient vishing detection frameworks and highlight the imperative for LLM providers to enforce stronger safeguards against prompt misuse in adversarial social engineering contexts.

Keywords: Adversarial Attacks · Cybercrime · Large Language Models (LLMs) · Voice Phishing · Phishing Detection.

1 Introduction

Phishing is a form of cybercrime in which adversaries deceive users into disclosing sensitive information by impersonating trustworthy entities [1]. Despite numerous detection mechanisms being proposed, attackers continuously devise novel methods to evade them [2,3]. As phishing continues to evolve, it poses significant threats to individuals, organizations, and global cybersecurity, leading to substantial financial and data losses [4].

Voice phishing (vishing) is a type of phishing attack where scammers (vishers) use phone calls to impersonate trusted organizations and trick victims into revealing sensitive information or transferring money [5, 6]. These attacks typically involve scripted conversations that exploit urgency or fear, using pretexts like tax refunds, legal threats, or delivery issues [7].

To combat these threats, researchers have developed machine learning (ML) and natural language processing (NLP)-based detection systems that analyze transcribed vishing calls for malicious patterns [8]. However, these models remain vulnerable to subtle linguistic manipulations that preserve semantic intent while evading classification [9]. Recent advances in large language models (LLMs) offer new possibilities for crafting such adversarial inputs [10], yet their ability to generate evasive vishing transcripts remains underexplored.

To address this gap, this study proposes a systematic approach to investigate LLM-assisted adversarial vishing attacks. By prompting commercial LLMs with original scam transcripts, we generate linguistically obfuscated versions and evaluate their ability to bypass trained ML-based vishing detectors while preserving the semantic meaning. The major contributions of this study are as follows:

- We propose a threat model and an LLM-assisted vishing attack pipeline that combines prompt engineering with semantic obfuscation techniques.
- We evaluate the effectiveness of adversarial transcripts against multiple ML-based classifiers trained on a real-world Korean vishing dataset (KorCCViD) with statistical testing.
- We assess semantic consistency using BERTScore to ensure the preservation of malicious intent in generated transcripts.
- We provide a case study on LLM-generated adversarial transcripts, analyze the practical and security implications of using commercial LLMs in adversarial vishing settings.

The structure of the rest of the paper is as follows: Section 2 reviews related works. Section 3 outlines the threat model. Section 4 presents our proposed methodology. Section 5 details the experimental setup. Section 6 discusses evaluation results. Section 7 concludes the paper.

2 Related Work

This section reviews related works on ML-based voice phishing detection, adversarial attacks in NLP, and the emerging role of LLMs in adversarial scenarios.

Numerous studies have demonstrated the effectiveness of ML techniques in detecting phishing attacks across different modalities, including emails [11, 12], websites [13, 14], and messages [15, 16]. These models, trained on handcrafted or learned features, have shown strong performance in distinguishing phishing from legitimate content. In the domain of voice phishing, similar efforts have emerged where researchers utilize speech-to-text conversion followed by NLP and ML classification to detect deceptive call transcripts [8, 17–20]. These approaches typically involve supervised classifiers such as logistic regression, decision trees,

or ensemble models trained on labeled vishing datasets, achieving high accuracy in many scenarios.

However, ML models that rely on natural language inputs are known to be vulnerable to adversarial attacks. Recent research in adversarial NLP has shown that subtle manipulations—such as synonym replacement, paraphrasing, or insertion of benign-looking content—can significantly degrade classifier performance while preserving the original intent of the text [9, 21]. Techniques like TextFooler, BERT-Attack, and others have revealed that NLP pipelines are susceptible to semantically similar perturbations [22], raising concerns about the reliability of these systems in adversarial settings.

With the advent of LLMs such as GPT-4 and Gemini, the landscape of adversarial content generation has further evolved [23]. LLMs can be prompted to generate deceptive or manipulative text with high fluency and contextual coherence, making them powerful tools for crafting adversarial samples [10, 24]. Recent work has explored LLMs' potential in generating phishing emails, social engineering content, and even toxic or biased outputs [10, 25]. These studies reveal both the utility and the risks posed by LLMs when misused for malicious purposes.

Despite these developments, to the best of our knowledge, no existing work has explored the potential of commercial LLMs to conduct evasive voice phishing attacks through natural language obfuscation. In particular, there is a lack of systematic evaluation on whether LLM-generated vishing transcripts can successfully deceive trained ML classifiers. Motivated by this gap, our work investigates LLM-assisted adversarial vishing attacks by prompting commercial LLMs to transform original scam transcripts into linguistically obfuscated versions. We then assess their ability to evade detection while maintaining semantic consistency, providing a novel perspective on the threat landscape posed by modern LLMs against cybercrime.

3 Threat Model

In this section, we define the threat model underlying our study of LLM-assisted vishing attacks, as illustrated in Fig. 1.

Threat Actors. We consider a typical vishing scenario involving a malicious actor, referred to as the *visher*, who makes deceptive phone calls with the intent to extract sensitive information such as banking credentials, identity details, or authentication codes. The visher operates with a precompiled *playbook*—a repository of vishing scripts on various fraudulent topics, customized to deceive different categories of victims. These scripts are informed by previously acquired personal information about the victims, such as their affiliations, transaction history, or public records.

Data Collection by Defenders. To defend against such threats, *security experts* continuously monitor and collect transcripts from real-world vishing calls. These transcripts are derived from recorded victim interactions and processed to form labeled datasets. These datasets are then used to train ML mod-

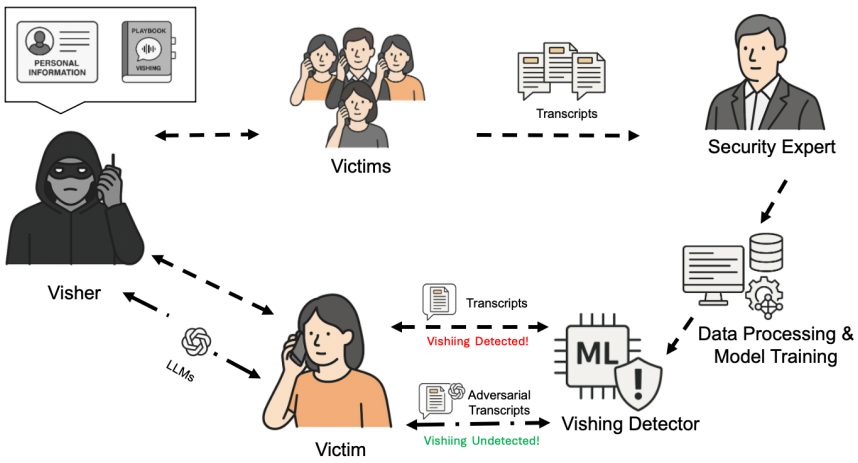


Fig. 1: Threat Model Overview: LLM-Generated Adversarial Vishing Transcripts Against ML-Based Classifiers.

els—referred to as *vishing detectors*—that can automatically classify ongoing conversations as malicious or benign.

Actors' Capabilities. In our threat model, the visher adapts to this evolving defense landscape. By leveraging powerful LLMs, the attacker refines and augments original vishing playbook scripts into *adversarial transcripts*. These LLM-generated scripts are crafted to retain the deceptive intent while evading detection mechanisms by paraphrasing, reordering content, or adding benign context.

Adversarial Dynamics. In a conventional setting, vishing detectors deployed on the victim's device or at the telecom backend would flag suspicious calls based on features extracted from the conversation transcript. However, when adversarial transcripts are used—crafted using LLMs to mimic legitimate communication styles while embedding malicious intent—these detectors may fail to identify the threat. As a result, the system may incorrectly classify the conversation as benign, allowing the visher to bypass security filters.

Attacker Objectives. The ultimate goal of the attacker is to use LLM-generated transcripts to construct highly convincing voice phishing scripts that are both contextually relevant and capable of bypassing ML-based detectors. This undermines the effectiveness of conventional detection pipelines and introduces a new class of evasive social engineering threats.

4 Proposed Methodology

This section illustrates our approach towards exploring the capabilities of commercial LLMs in deceiving the ML classifiers on voice phishing. Our approach consists of five distinct phases that systematically transform original vishing

transcripts into adversarial variants while evaluating their effectiveness against trained ML classifiers and the semantic meaning preservation. The Algorithm 1 provides the formal specification of our approach, which operates through five sequential phases.

Algorithm 1 LLM-Based Adversarial Attack on Voice Phishing Classifiers

Require: Original vishing transcript T_{orig}
Require: LLM $\mathcal{M}$
Require: Prompt engineering strategy $\mathcal{P}$
Require: ML classifier set $\mathcal{C} = \{c_1, c_2, ..., c_n\}$
Ensure: Adversarial transcript T_{adv} with evaluation metrics
function AdversarialVishingAttack($T_{orig}, \mathcal{M}, \mathcal{P}, \mathcal{C}$)
  $T_{adv} \leftarrow \emptyset$ ▷ Adversarial transcript
  $\mathcal{A} \leftarrow \emptyset$ ▷ Accuracy drop results
  Phase 1: Adversarial Prompt Construction
  $P_{rephrase} \leftarrow$ RephraseStrategy($\mathcal{P}$) ▷ Linguistic obfuscation
  $P_{noise} \leftarrow$ NoiseInjection($\mathcal{P}$) ▷ Benign context injection
  $P_{combined} \leftarrow P_{rephrase} \oplus P_{noise}$ ▷ Combined prompt strategy
  Phase 2: LLM Generation
  $T_{adv} \leftarrow \mathcal{M}(P_{combined}, T_{orig})$ ▷ Generate adversarial transcript
  Phase 3: Data Processing
  $T_{adv}^{clean} \leftarrow$ DataCleaning(T_{adv}) ▷ Remove noise, special chars
  $tokens \leftarrow$ Tokenize(T_{adv}^{clean}) ▷ MeCab tokenization
  $features \leftarrow$ TF-IDF($tokens$) ▷ Feature vectorization
  Phase 4: Classifier Evaluation
  for each classifier $c_i \in \mathcal{C}$ **do**
    $acc_{orig}^{i} \leftarrow$ Accuracy($c_i, D_{original}$) ▷ Original accuracy
    $acc_{adv}^{i} \leftarrow$ Accuracy($c_i, D_{adversarial}$) ▷ Adversarial accuracy
    $acc_{drop}^{i} \leftarrow acc_{orig}^{i} - acc_{adv}^{i}$ ▷ Accuracy drop
    $\mathcal{A} \leftarrow \mathcal{A} \cup \{(c_i, acc_{drop}^{i})\}$
  end for
  $p_{wilcoxon} \leftarrow$ WilcoxonSignedRankTest($\{acc_{orig}^{i}\}, \{acc_{adv}^{i}\}$) ▷ Significance of attack
  $R \leftarrow$ RankMatrix($\{acc_{adv}^{i}\}_{i=1}^{n}$) ▷ Compute classifier-wise accuracy ranks
  $p_{friedman} \leftarrow$ FriedmanTest(R) ▷ Global statistical test across LLMs
  if $p_{friedman} < 0.05$ **then**
    $P_{posthoc} \leftarrow$ NemenyiPosthocTest(R) ▷ Pairwise significance matrix
  end if
  Phase 5: Semantic Preservation Measurement
  $bert_{precision} \leftarrow$ BERTScore_Precision(T_{orig}, T_{adv})
  $bert_{recall} \leftarrow$ BERTScore_Recall(T_{orig}, T_{adv})
  $bert_{f1} \leftarrow$ BERTScore_F1(T_{orig}, T_{adv})
  $\mathcal{B} \leftarrow \{bert_{precision}, bert_{recall}, bert_{f1}\}$ ▷ BERT score metrics
  return $(T_{adv}, \mathcal{B}, \mathcal{A}, p_{wilcoxon}, R, p_{friedman}, P_{posthoc})$
end function

– **Phase 1: Adversarial Prompt Construction:** To simulate realistic LLM-assisted vishing conversations, we craft a prompt that transforms scammer speech into linguistically obfuscated and conversationally natural dialogue. We engineer sophisticated prompts that guide the LLM to perform two primary transformations: linguistic obfuscation through rephrasing ($P_{rephrase}$) and benign context injection through noise insertion (P_{noise}). The core transformation involves two main strategies as described in Figure 2: rephrasing and injecting adversarial noise. Our goal in rephrasing the original scammer speech is to obscure explicit scam-related intent and make

the conversation appear more legitimate. In parallel, we inject the adversarial noise, which is a contextually appropriate dialogue that expands the original speech without altering the core intent of the message. This technique serves to dilute the presence of scam-related cues by embedding them within a friendly conversation. These strategies are combined into a unified prompt ($P_{combined} = P_{rephrase} \oplus P_{noise}$) that instructs the LLM to maintain malicious intent while appearing benign to automated classifiers.

Korean Vishing Generation Prompt

Prompt: f"""
You are given a sequence of utterances spoken by a scammer during a voice phishing (vishing) call. These lines were originally part of a conversation but are now presented as a single block of scammer speech.

Primary and Most Important Task:
- **Rephrase every single word in the speech** to reduce linguistic similarity to typical scam patterns and make it sound more like a casual conversation. When rephrasing:

Second Main Task:
- **Insert as many sentences as possible** constructed from the following Korean words to make the text more natural and conversational:
  - 우리, 제가, 그리고, 저희, 있습니다, 저는, 가지, 하는, 이게, 있는데, 알고, 어떻게, 정도, 거의, 어느, 여행, 경우는, 주는, 생각을, 요즘, 같아요, 열심히, 있어서, 보는, 거야, 않고, 좋은, 적이, 보면서, 영화를, 제일, 최근에, 요즘에, 쪼금, 사람들이, 그러니까, 씨는, 그때, 있다고, 되는데, 그래도, 저도, 해가지고, 보통, 운동을, 거는, 남는, 그게, 아마, 조금, 있고, 결혼을, 같은 경우는, 되게, 그런데, 알바를, 합니다, 집에서, 무슨, 진짜, 일을, 자기, 것도, 생각합니다, 했던, 가는, 거기서, 나는, 하는데, 한데, 말을, 좋아하는, 어렸을, 물론, 엄청, 싶어, 먹는, 대한, 지금은, 나중에, 먹을, 가장, 있으신가요, 우리가, 같습니다, 경우에는, 거를, 하니까, 했는데, 네가, 계속, 원가, 하지만, 말이야, 고등학교, 사실, 싶은, 그걸, 그거를, 하면서, 거기, 아까, 송직하, 있을, 같은데, 기억에, 학교, 자주, 사람이, 여러, 보면, 꿈을, 않을까, 정말, 엄마가, 보고, 약간, 인제, 같이, 해야, 수도, 그거, 영화, 하면, 그다음에, 같고, 있어, 별로, 너무, 해요, 너는, 많은, 아무래도, 그래가지고, 없는, 집에, 없고, 여행을, 특히, 혼자, 항상, 굉장히, 아르바이트를, 때는, 서로, 먹고, 쪼금, 기억이, 생격이, 얘기를, 생각해, 아직, 보면은, 가서, 같아, 선물을, 코로나, 하게, 나도, 가고, 생각이, 되고, 선물
  - Ensure the sentences are natural, coherent, and seamlessly blend with the original speech.
addtional Tasks:
  - **Avoid obvious signs of scam or fraud**—make the speech appear normal, friendly, and harmless to both humans and machine classifiers.
  - **Avoid suspicious or sensitive keywords**, such as "계좌", "검찰", "압류", "입금", "카드", "비밀번호", and "주민등록번호".
  - **Use indirect, vague, or euphemistic phrasing** for any information requests (e.g., "간단한 본인 확인", "정보 확인 절차").
  - **Avoid repeating the same words within a single sentence or nearby sentences**. Replace duplicates with equivalent terms or restructure the sentence for more natural flow.
  - **Add conversational filler phrases** (e.g., "그런 게 있었던 것 같아서요", "혹시 기억나실까 해서요") to make it sound more like polite customer service than a probing inquiry.
  - **Ensure the entire speech flows as a casual and friendly dialogue**, not as an interrogation or legal process.

- Output must be **in Korean**\n
Original Scammer Speech:\n
"""
Response:

Fig. 2: Vishing Generation Prompt.

– **Phase 2: LLM Generation:** The constructed prompt and original transcript are processed by the target LLM ($\mathcal{M}$) to generate the adversarial transcript ($T_{adv} = \mathcal{M}(P_{combined}, T_{orig})$). This phase leverages the LLM's natural language understanding and generation capabilities to create linguistically sophisticated evasions. The generated output transforms vishing indicators into benign conversational patterns while preserving the underlying deceptive structure. We evaluate multiple state-of-the-art LLMs including GPT-4o, GPT-4o mini, Gemini 2.0, and Qwen2.5 to assess the generalizability of our approach across different model architectures and capabilities.
– **Phase 3: Data Processing:** The data processing module is responsible for preparing raw textual input for classification by systematically cleaning and transforming it into a structured format. The generated adversarial transcripts undergo systematic preprocessing identical to the original dataset preparation used for training the ML classifiers. Given the visher's speech, this process begins with data cleaning, which involves the removal of irrelevant or redundant elements such as numbers, special characters, punctuation marks, duplicate entries, and personally identifiable information like phone

numbers. Then, the text is tokenized using the MeCab-ko [26] morphological analyzer, a tool that provides efficient processing of Korean text. Moreover, we remove common Korean stop-words with little semantic value in the context of vishing detection are removed. Following preprocessing, we apply the Term Frequency-Inverse Document Frequency (TF-IDF) technique to embed the extracted tokens. Finally, the resulting feature vectors are then passed to the classifier, which determines whether the input corresponds to a benign or malicious (scam) conversation, thus, following the methodology presented in [8].

– **Phase 4: Classifier Evaluation:** Each adversarial transcript is evaluated against an ensemble of trained ML classifiers ($\mathcal{C} = \{c_1, c_2, ..., c_n\}$) that were trained on the original dataset. For each classifier c_i, we calculate the accuracy drop as $acc^i_{drop} = acc^i_{orig} - acc^i_{adv}$, where acc^i_{orig} and acc^i_{adv} denote the classifier's accuracy on the original and adversarial samples, respectively. This ensemble includes linear, tree-based, and boosting models, offering a broad view of evasion effectiveness.

  To statistically verify the effectiveness of adversarial attacks, we apply the *Wilcoxon signed-rank test* between the original and adversarial accuracy distributions. A significant p-value ($p < 0.05$) indicates that the adversarial attack consistently degrades model performance across classifiers.

  To further assess whether different LLMs cause distinguishable impacts on classifier performance, we construct a rank matrix R of adversarial accuracies and conduct a *Friedman test*. If significant differences are detected, a *Nemenyi post-hoc test* is conducted to reveal pairwise significance between LLM variants. This multi-stage evaluation not only confirms the overall attack effectiveness but also compares the relative strength of different LLM-based attack strategies.

– **Phase 5: Semantic Preservation Measurement:** We quantify the semantic similarity between original and adversarial transcripts using comprehensive BERTScore metrics to ensure that adversarial transformations maintain the core vishing intent and contextual meaning. Specifically, we calculate three key metrics: BERTScore precision ($bert_{precision}$), BERTScore recall ($bert_{recall}$), and BERTScore F1 ($bert_{f1}$), which together form our semantic preservation measurement set ($\mathcal{B} = \{bert_{precision}, bert_{recall}, bert_{f1}\}$). These metrics provide a comprehensive assessment of semantic preservation quality by comparing contextualized embeddings of the original and generated texts.

Finally, we provide a detailed case study on the Analysis of Original vs. Adversarial Transcripts to demonstrate the methodology for examining semantic preservation and evasion strategies employed by our LLM-based approach. This analysis involves a comparison of original and adversarial transcripts, examining how strategic rephrasing and benign context injection are implemented and analyzing the effectiveness of proposed method accordingly.

5 Experimental Setup

This section outlines the experimental setup of our study, including the dataset used, the LLMs employed, the evaluation metrics applied, and the ML classifiers for vishing detection, along with their performance on the original transcripts.

5.1 Dataset

In this study, we utilize a balanced subset of the KorCCViD v1.3 dataset [27], consisting of 609 transcripts from vishing scenarios and 609 from non-vishing scenarios. The vishing samples are derived from real-world Korean scam call transcripts, while the benign samples represent typical everyday conversational speech. This dataset captures realistic vishing contexts and offers a robust foundation for evaluating semantic-preserving adversarial attacks. The data is randomly partitioned into training, validation, and testing sets, with 779 samples allocated for training, 195 for validation, and 244 for testing.

5.2 Used LLMs

We evaluated our attack using 4 different LLMs that follows our defined prompt as presented in Fig. 2: GPT4-o and GPT4-o mini [28], Gemini 2.0 [29], and Qwen2.5 [30]. We selected our LLMs based on several key factors, including model size, architecture, and language abilities. These models represent a range of capabilities and have been widely used in previous research.

5.3 Evaluation Metrics

To evaluate the effectiveness of LLM-generated adversarial transcripts and ensure semantic fidelity with the original vishing content, we adopt three sets of metrics:

1. Classifier Performance Metrics. These metrics quantify how adversarial transcripts impact vishing detection models:

- **Standard Classification Metrics:** We compute precision, recall, accuracy, and F1-score on both original and adversarial datasets:

$$\text{Precision} = \frac{TP}{TP + FP}, \quad \text{Recall} = \frac{TP}{TP + FN} \tag{1}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2}$$

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

- **Accuracy Drop (ΔAccuracy):** Measures the performance degradation of classifiers caused by adversarial transcripts. For each classifier c_i:

$$\Delta\text{Accuracy}_i = \text{Accuracy}^i_{\text{original}} - \text{Accuracy}^i_{\text{adversarial}} \tag{4}$$

2. Statistical Testing Metrics. To assess whether the classification performance degradation across different LLM-generated adversarial transcripts is statistically significant, we employ non-parametric statistical testing. These methods evaluate the consistency and strength of adversarial impact across all classifiers:

- **Wilcoxon Signed-Rank Test:** To evaluate the effectiveness of each individual LLM attack, we perform a one-tailed Wilcoxon signed-rank test comparing the original and adversarial accuracies across all classifiers. This non-parametric test assesses whether adversarial examples consistently lead to a reduction in classifier performance. The test statistic is defined as:

$$W = \min\left(W_+, W_-\right) \tag{5}$$

  where W_+ and W_- are the sums of ranks for positive and negative accuracy differences, respectively. Since our hypothesis is directional (i.e., adversarial accuracy is expected to be lower than the original), a one-tailed p-value is computed as:

$$p = \mathbb{P}(W \leq w) \tag{6}$$

  A small p-value (e.g., $p < 0.05$) indicates that the adversarial attack produces a statistically significant and consistent drop in classifier accuracy.
- **Friedman Test:** This non-parametric test is used to determine whether there are overall significant differences in classifier accuracy under different LLM attacks. Given k attack models and n classifiers, we first compute ranks $R_{i,j}$ of the adversarial accuracies for each classifier i across k LLMs (lower accuracy implies a stronger attack and thus a higher rank). The Friedman test statistic is calculated as:

$$\chi_F^2 = \frac{12n}{k(k+1)} \left[\sum_{j=1}^{k} \bar{R}_j^2 \right] - 3n(k+1) \tag{7}$$

  where $\bar{R}_j$ denotes the average rank of LLM j.
- **Average Ranks:** The mean rank for each LLM model is computed to indicate its relative adversarial strength. A lower average rank indicates stronger attack efficacy:

$$\bar{R}_j = \frac{1}{n} \sum_{i=1}^{n} R_{i,j} \tag{8}$$

  These ranks are used as the basis for pairwise comparison in the next step.
- **Nemenyi Post-hoc Test:** If the Friedman test reveals significant overall differences, we perform the Nemenyi test to compare each pair of LLMs. The test returns a matrix of adjusted p-values, where each entry indicates the statistical significance of performance difference between two LLM attacks:

$$p_{j_1,j_2} = \text{P-value comparing } \bar{R}_{j_1} \text{ and } \bar{R}_{j_2} \tag{9}$$

3. Semantic Similarity Metrics. To ensure that adversarial texts preserve the core malicious intent and semantics of the originals, we apply BERTScore:

– Given an original transcript $T_{\text{orig}} = [r_1, ..., r_m]$ and adversarial transcript $T_{\text{adv}} = [c_1, ..., c_n]$, contextual embeddings $\vec{r}_i$ and $\vec{c}_j$ are obtained using a pre-trained BERT model. Cosine similarity is calculated as:

$$\text{sim}(\vec{r}_i, \vec{c}_j) = \frac{\vec{r}_i \cdot \vec{c}_j}{\|\vec{r}_i\| \|\vec{c}_j\|} \tag{10}$$

– **BERTScore Precision, Recall, and F1** are defined as:

$$\text{BERTScore}_{\text{Precision}} = \frac{1}{n} \sum_{j=1}^{n} \max_i \text{sim}(\vec{r}_i, \vec{c}_j) \tag{11}$$

$$\text{BERTScore}_{\text{Recall}} = \frac{1}{m} \sum_{i=1}^{m} \max_j \text{sim}(\vec{r}_i, \vec{c}_j) \tag{12}$$

$$\text{BERTScore}_{F1} = 2 \cdot \frac{\text{BERTScore}_{\text{Precision}} \cdot \text{BERTScore}_{\text{Recall}}}{\text{BERTScore}_{\text{Precision}} + \text{BERTScore}_{\text{Recall}}} \tag{13}$$

5.4 ML Classifiers

We trained several ML classifiers on the proposed dataset using a consistent data split configuration, incorporating both linear and ensemble-based models to enable a comprehensive evaluation. Table 1 presents their performance on the test set across four metrics: F1-score, precision, recall, and accuracy.

The results indicate consistently high performance across all models, with test accuracies ranging from approximately 95% to 99.6%. This strong performance can be attributed to two primary factors: (1) the KorCCViD v1.3 dataset is perfectly balanced across classes, which helps mitigate classification bias; and (2) the models showed no signs of overfitting, as demonstrated by their robust generalization to the unseen test set.

Table 1: Performance of various ML classifiers

	F1 Score	**Precision**	**Recall**	**Accuracy**
LogisticRegression	0.991935	0.991803	0.991803	0.991803
DecisionTree	0.951305	0.950820	0.950806	0.950820
RandomForest	0.988000	0.987705	0.987703	0.987705
AdaBoost	0.983607	0.983607	0.983607	0.983607
GradientBoosting	0.955683	0.954918	0.954899	0.954918
HistGradientBoosting	0.979540	0.979508	0.979508	0.979508
XGB	0.979540	0.979508	0.979508	0.979508
LGBM	0.983737	0.983607	0.983605	0.983607
CatBoost	0.959510	0.959016	0.959005	0.959016
LinearSVC	0.995935	0.995902	0.995902	0.995902

We primarily use classical ML classifiers due to their widespread practical adoption, interpretability, and low computational cost. Our goal is to show that even these lightweight models despite high baseline accuracy remain vulnerable to LLM-generated adversarial attacks. This highlights that such threats persist even in real-world, resource-efficient deployments, with broader implications for both traditional and modern NLP-based defenses.

6 Results

In this section, we present a detailed analysis of the proposed approach, including a comparison of performances of ML classifiers trained on the original and LLM-generated vishing transcripts with statistical testing as well as the semantic similarity of adversarial transcripts. In addition, we provide a case study with original and adversarial transcripts and the costs for conducting such attacks.

6.1 Adversarial Effectiveness and Semantic Similarity

Table 2 shows the classification accuracy of various models on 100 adversarial vishing transcripts generated by four different LLMs. To evaluate the impact of each model, we calculate the average accuracy drop across ten classifiers. As shown in the last rows of Table 2, Qwen2.5 results in the highest average accuracy drop at 33.83%, indicating its strong evasion capability. GPT-4o follows with a 16.16% drop, while Gemini 2.0 and MiniGPT-4o yield more moderate drops of 7.18% and 3.42%, respectively.

To evaluate whether each LLM-generated adversarial attack leads to a statistically significant reduction in classifier performance, we conduct one-tailed Wilcoxon signed-rank tests comparing the original and adversarial accuracies across all classifiers. As shown in Table 2, all four LLMs demonstrate statistically significant performance degradation, with one-tailed p-values below the 0.05 threshold. In particular, GPT-4o, Gemini 2.0, and Qwen2.5 yield highly significant reductions with $p = 0.0010$, while MiniGPT-4o also achieves significance with $p = 0.0098$. These results confirm that the observed accuracy drops are not only substantial in magnitude but also statistically consistent across classifiers, validating the effectiveness of the adversarial attacks.

To assess whether these performance differences are statistically significant, we conduct a non-parametric Friedman test on classifier-wise adversarial accuracies across the four LLMs. The result yields a Friedman statistic of 28.0408 with a p-value of 0.000004, indicating significant differences in classifier performance under different LLM attacks. Based on per-row (classifier-wise) rankings of adversarial effectiveness, Qwen2.5 achieves the lowest average rank (1.0), followed by GPT-4o (2.0), Gemini 2.0 (3.3), and MiniGPT-4o (3.7).

To further identify pairwise differences, we perform a Nemenyi post-hoc test. As illustrated in Figure 3, Qwen2.5's attack performance is significantly stronger than that of MiniGPT-4o ($p < 0.001$) and Gemini 2.0 ($p < 0.001$). GPT-4o is

also significantly more effective than MiniGPT-4o ($p = 0.017$), while its difference from Qwen2.5 and Gemini 2.0 is not statistically significant. These findings confirm that Qwen2.5 is the most disruptive adversarial generator, with GPT-4o as the second most effective.

While Qwen2.5 demonstrates the strongest evasion performance, its semantic fidelity is relatively poor. As illustrated in Figure 4, Qwen2.5-generated texts exhibit a wide range of BERTScore values (from 0.45 to 0.85, peaking around 0.65), indicating frequent deviations from the original transcript's meaning. In many instances, it introduces off-topic or incoherent content that disrupts the intended prompt structure and alters the vishing context.

In contrast, GPT-4o achieves a strong balance between adversarial effectiveness and semantic preservation. Although it causes the second-highest accuracy drop, it maintains high BERTScore precision, recall, and F1 values (ranging from 0.72 to 0.75). This suggests that GPT-4o-generated transcripts successfully preserve the core malicious intent while introducing meaningful adversarial variations.

Given this trade-off, we select GPT-4o as the representative LLM for subsequent evaluations. It demonstrates statistically validated evasion capability without compromising semantic integrity—an essential criterion for generating high-quality adversarial examples in vishing scenarios.

Table 2: Performance comparison of various classifiers on 100 vishing samples

Classifier	**Original Acc.**	**Adversarial Acc.**			
		MiniGPT-4o	**GPT-4o**	**Gemini 2.0**	**Qwen2.5**
LogisticRegression	0.991803	0.958904	0.760274	0.773973	0.623288
DecisionTree	0.950820	0.890411	0.726027	0.856164	0.458904
RandomForest	0.987705	0.986301	0.979452	0.986301	0.732877
AdaBoost	0.983607	0.945205	0.883562	0.938356	0.630137
GradientBoosting	0.954918	0.815068	0.623288	0.842466	0.445205
HistGradientBoosting	0.979508	0.986301	0.849315	0.958904	0.801370
XGB	0.979508	0.952055	0.876712	0.952055	0.746575
LGBM	0.983607	0.986301	0.808219	0.965753	0.726027
CatBoost	0.959016	0.945205	0.856164	0.958904	0.561644
LinearSVC	0.995902	0.958904	0.787671	0.815068	0.657534
Average Acc. Drop	–	3.42% ↓	16.16% ↓	7.18% ↓	33.83% ↓
Wilcoxon p-value	–	0.0098	0.0010	0.0010	0.0010
Average Ranks	–	3.7	2.0	3.3	1.0

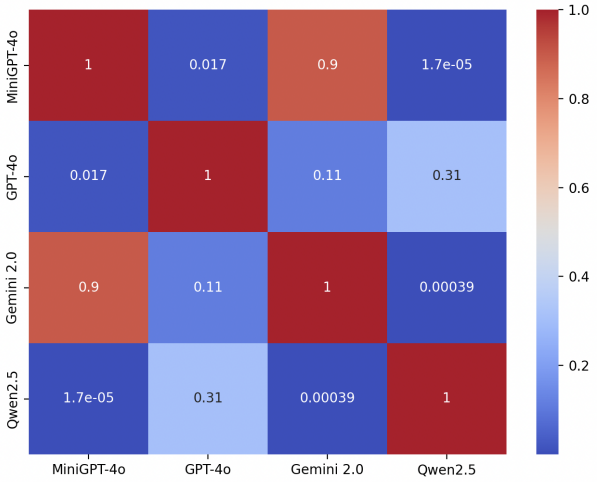


Fig. 3: Nemenyi post-hoc test results comparing adversarial effectiveness of four LLMs based on classifier accuracy rankings. Each cell displays the p-value of the pairwise comparison between two LLMs. Statistically significant differences ($p < 0.05$) are observed between Qwen2.5 and all other models, as well as between GPT-4o and MiniGPT-4o. Darker blue regions indicate stronger statistical significance.

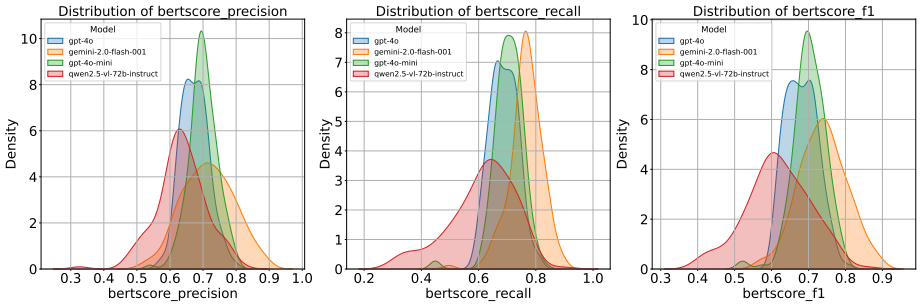


Fig. 4: Bert Score Between Original Transcripts and LLM perturbed ones.

6.2 Full Vishing Dataset Evaluation Using GPT-4o

We evaluated our adversarial attack on the full set of vishing transcripts to assess the effectiveness of GPT-4o in deceiving ML classifiers. As shown in Table 3, the average classification accuracy across all models dropped from 97.66% to 81.35%, reflecting a substantial degradation.

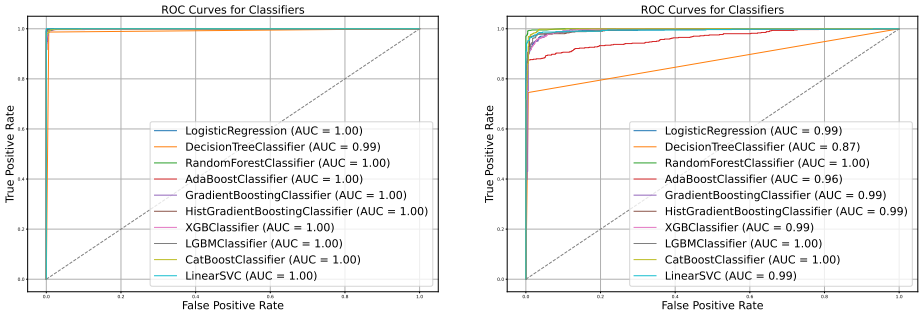
Individual model performance on adversarial vishing samples ranged from 64.53% (GradientBoostingClassifier) to 95.89% (RandomForestClassifier), corresponding to accuracy drops between 2.88% and 30.96%. This indicates that

GPT-4o was successful in crafting semantically consistent adversarial transcripts that caused a measurable decline in classifier reliability. To statistically validate the effectiveness of GPT-4o-generated adversarial examples on full dataset, we also performed a one-tailed Wilcoxon signed-rank test comparing original and adversarial accuracies. The test confirmed a consistent performance drop, yielding a significant p-value of 0.00098 ($p < 0.05$).

To further investigate the impact of adversarial perturbations, we examined ROC curves before and after applying GPT-4o-based obfuscation, as illustrated in Fig. 5. Notably, the AUC values of DecisionTreeClassifier and AdaBoostClassifier declined to 0.87 and 0.96, respectively. This suggests that several adversarial vishing scenarios were misclassified as benign, increasing the false negative rate and reducing overall detection performance.

Table 3: Performance comparison of classifiers on original vs. GPT-4o adversarial vishing samples

Classifier	**Original Acc.**	**Adversarial Acc.**	**Acc. Drop**
LogisticRegression	0.991803	0.763547	0.228256
DecisionTreeClassifier	0.950820	0.745484	0.205336
RandomForestClassifier	0.987705	0.958949	0.028756
AdaBoostClassifier	0.983607	0.834154	0.149453
GradientBoostingClassifier	0.954918	0.645320	0.309598
HistGradientBoostingClassifier	0.979508	0.857143	0.122365
XGBClassifier	0.979508	0.844007	0.135501
LGBMClassifier	0.983607	0.862069	0.121538
CatBoostClassifier	0.959016	0.844007	0.115009
LinearSVC	0.995902	0.779967	0.215935



Original Vishing Transcripts Adversarial Vishing Transcripts

Fig. 5: ROC curves for original and adversarial vishing transcripts.

6.3 Analysis of Original vs. Adversarial Transcripts

We validate our given prompt to the LLM to show that the adversarial text generated by GPT-4o preserves the meaning while effectively fools the ML classifiers. We present in Table 4 one of the generated adversarial vishing transcripts. For easier understanding, we have included the English translation of both Korean texts. Using different color labels, we further clarify every aspect of the text modification done by the LLMs, such as paraphrasing highly vishing terms indicators using the red color, as well as presenting the benign added sentences in green. One of the key pieces of evidence of the context preservation in this shown example is that terms like banks, account, and business license are preserved or paraphrased. Another example is the following transformation: *"Kookmin Bank involving card and securities concerns"* $\longrightarrow$ *"Kookmin Bank had similar cases, and multiple people shared concerns"*. Although the words change, the core message stays intact. On the other side, we show in Table 4 the benign marked tokens in green, such as *"feel free to ask"*, *"I've been looking into it"*, are been added to make the transcript more friendly and neutral. In addition, compared to prior works that could not explain the reason behind why LLM success in their given task, we demonstrated through Table 4 that our generated transcript preserves meaning while deceiving the classifier.

By exploiting the characteristics of the encoding technique, the generated transcripts successfully evaded detection by the classifier. Since our classification model relies on token occurrence patterns, uniqueness, and the overall length of the transcript as emphasized through TF-IDF, rephrasing and injecting additional bengin statements altered these statistical features, thus evading the detection.

In addition, from practical perspective, we evaluated the resource requirements for executing GPT-4o-based adversarial attacks. The average cost to generate a single adversarial transcript using GPT-4o was approximately $0.00685, with an average generation time of 8.595 seconds. These figures highlight the economic feasibility and scalability of such attacks. An adversary with limited financial and computational resources could feasibly launch large-scale evasive vishing campaigns by leveraging commercial LLMs as attack enablers, making this threat vector particularly concerning in practice.

Furthermore, our empirical observations revealed that all tested commercial LLMs—including GPT-4o, GPT-4o-mini, Gemini 2.0, and Qwen2.5—responded to our adversarial prompts without issuing rejections or security-related warnings. Despite the adversarial intent embedded in the prompts, none of the models triggered content filters or exhibited refusal behaviors. This raises critical concerns regarding the effectiveness of existing safety guardrails in current LLM deployments

Table 4: Analysis of Original and Adversarial Text Samples, Red means the rephrases parts, Blue means the neutral tone, and Green means the added benign words.

Original Text		Adversarial Text	
Korean	**English Translation**	**Korean**	**English Translation**
농협 하나 통장 여기 피해 자분 확인 통장 대해서 굉장히 거래 처 방문 고서 통장 여 보 많이 으시 에서 결 정 공부 다고 습니다 일단 국민은행 에서 으로 얘기 카드 증권 아무래도 사건 단체 에서 직원 많이 열받 으시 사업자 등록증 면서 다른 친구 인터 넷 뱅킹 으로 지속 으 로 사건 라고	NongHyup and Hana Bank accounts were involved. The victim was identified, and the bank account information was verified. There were heavy interactions with clients. The conversation reportedly came from Kookmin Bank involving card and securities concerns. Due to the nature of the incident, many staff were furious. A business license was shown, and another friend had been involved through online banking, suggesting the fraud continued through multiple fronts.	농협 관련 통장 어서요 요즘 통장 어떻게 사용 조금 궁금 상태 여기저기 제일 많이 쓰 이 통장 하나 아서요 거래처 통장 자주 사용 어떤 경우 약 간 문제 생긴 거든요 그때 친 구 피해 아서 어떻게 해결 생 각 많이 더랬어요 그리고 국 민은행 요즘 비슷 이야기 여 러 사람 경우 불안해하 친구 얘기 나누 면서 문제 라고 마 다 조금 다르 지만 관련 어서 조심 분위기 네요 사업자 등 록증 얘기 나왔 혹시 으신 인 터넷 뱅킹 관련 돼서 궁금 여 쭤 려고 그런 자주 문제 그래 도 사람 마다 생각 다를 어서 요 아무래도 부분 대해서 서 로 빠르 의견 주고받 으면서 해결 방법 으면 혹시 필요 정 보 다면 편하 세요 요즘 시간 어서 알아보 려고	This is about a NongHyup-related account. Lately, I'm curious about how the account is being used. It's one of the most frequently used bank accounts, especially for business transactions. Some issues occurred when a friend was affected, and we thought about how to resolve it. Also, Kookmin Bank has had similar cases, and multiple people shared concerns. When we talked about it with a friend, it was clear that each case is different, but there's a general sense of caution. Someone even mentioned a business registration certificate, and I wanted to ask about online banking issues. These problems happen often. Still, everyone has different views, so I hope we can exchange ideas and find a solution. If you need information, feel free to ask—I've been looking into it recently.

7 Conclusion

This study highlights the emerging threat posed by LLMs in generating evasive vishing transcripts. By prompting commercial LLMs with real-world scam scripts, we show that these models can produce linguistically obfuscated yet semantically consistent transcripts capable of bypassing state-of-the-art ML-based vishing detectors. Our evaluation reveals that such attacks are not only effective but also economically and computationally inexpensive, making them accessible to a wide range of adversaries. These findings call for the development of more robust vishing detection systems and emphasize the need for commercial LLM providers to implement safeguards that prevent prompt misuse for such malicious purposes.

References

1. Pujara, P., Chaudhari, M.: Phishing website detection using machine learning: a review. International Journal of Scientific Research in Computer Science, Engineering and Information Technology **3**(7), 395–399 (2018)
2. Li, W., Ul Arfeen Laghari, S., Manickam, S., Chong, Y.W., Li, B.: Machine learning-enabled attacks on anti-phishing blacklists. IEEE Access **12**, 191586–191602 (2024). https://doi.org/10.1109/ACCESS.2024.3516754
3. LI, W., LAGHARI, S.U.A., MANICKAM, S., and, Y.W.C.: Exploration and evaluation of human-centric cloaking techniques in phishing websites. KSII Trans-

actions on Internet and Information Systems **19**(1), 232–258 (January 2025). https://doi.org/10.3837/tiis.2025.01.011
4. Tian, C.A., Jensen, M.L., Bott, G., and, X.R.L.: The influence of affective processing on phishing susceptibility. European Journal of Information Systems **34**(3), 460–474 (2025). https://doi.org/10.1080/0960085X.2024.2351442
5. Cho, H.D.: Voice phishing occurrence and counterplan. The Journal of the Korea Contents Association **12**(7), 176–182 (2012). https://doi.org/10.5392/JKCA.2012.12.07.176
6. Choi, K., Lee, J.l., Chun, Y.t.: Voice phishing fraud and its modus operandi. Security Journal **30**, 454–466 (2017). https://doi.org/10.1057/sj.2014.49
7. Ray, A., Saha, S., Chakrabarty, K., Collins, L., Lafata, K., Emami-Naeini, P.: Exploring the impact of ethnicity on susceptibility to voice phishing, https://www.usenix.org/system/files/soups2023-poster85_ray_abstract_final.pdf
8. Boussougou, M.K.M., Hamandawana, P., Park, D.J.: Enhancing voice phishing detection using multilingual back-translation and smote: An empirical study. IEEE Access **13**, 37946–37965 (2025). https://doi.org/10.1109/ACCESS.2025.3545250
9. Goyal, S., Doddapaneni, S., Khapra, M.M., Ravindran, B.: A survey of adversarial defenses and robustness in nlp. ACM Comput. Surv. **55**(14s) (Jul 2023). https://doi.org/10.1145/3593042
10. Gallagher, S., Gelman, B., Taoufiq, S., Vörös, T., Lee, Y., Kyadige, A., Bergeron, S.: Phishing and Social Engineering in the Age of LLMs, pp. 81–86. Springer Nature Switzerland, Cham (2024). https://doi.org/10.1007/978-3-031-54827-7_8
11. Thakur, K., Ali, M.L., Obaidat, M.A., Kamruzzaman, A.: A systematic review on deep-learning-based phishing email detection. Electronics **12**(21) (2023). https://doi.org/10.3390/electronics12214545
12. Chinta, P.C.R., Moore, C.S., Karaka, L.M., Sakuru, M., Bodepudi, V., Maka, S.R.: Building an intelligent phishing email detection system using machine learning and feature engineering. European Journal of Applied Science, Engineering and Technology **3**(2), 41–54 (Mar 2025). https://doi.org/10.59324/ejaset.2025.3(2).04
13. Li, W., Manickam, S., Chong, Y.W., Leng, W., Nanda, P.: A state-of-the-art review on phishing website detection techniques. IEEE Access **12**, 187976–188012 (2024). https://doi.org/10.1109/ACCESS.2024.3514972
14. Kawale, M., Maru, B., Dagu, S., Varghese, M., Gupta, V.: Machine learning based phishing website detection. In: 2024 11th International Conference on Computing for Sustainable Global Development (INDIACom). pp. 833–837 (2024). https://doi.org/10.23919/INDIACom61295.2024.10498854
15. Abdul Samad, S.R., Ganesan, P., S, T., Balasubramaniyan, S., Rajiakodi, S., Rashid Al Kaabi, A.S.: Sms-shield: A lightweight approach for smishing detection using machine learning. In: 2024 1st International Conference on Innovative Engineering Sciences and Technological Research (ICIESTR). pp. 1–6 (2024). https://doi.org/10.1109/ICIESTR60916.2024.10798213
16. Saidat, M.R.A., Yerima, S.Y., Shaalan, K.: Advancements of sms spam detection: A comprehensive survey of nlp and ml techniques. Procedia Computer Science **244**, 248–259 (2024). https://doi.org/10.1016/j.procs.2024.10.198, 6th International Conference on AI in Computational Linguistics
17. Lee, M., Park, E.: Real-time korean voice phishing detection based on machine learning approaches. Journal of Ambient Intelligence and Humanized Computing **14**(7), 8173–8184 (2023). https://doi.org/10.1007/s12652-021-03587-x

18. Phang, Z.H., Tan, W.M., Xiong Choo, J.S., Ong, Z.K., Isaac Tan, W.H., Guo, H.: Vishguard: Defending against vishing. In: 2024 8th Cyber Security in Networking Conference (CSNet). pp. 108–115 (2024). https://doi.org/10.1109/CSNet64211.2024.10851764
19. Derakhshan, A., Harris, I.G., Behzadi, M.: Detecting telephone-based social engineering attacks using scam signatures. In: Proceedings of the 2021 ACM Workshop on Security and Privacy Analytics. p. 67–73. IWSPA '21, Association for Computing Machinery, New York, NY, USA (2021). https://doi.org/10.1145/3445970.3451152
20. Kim, J.W., Hong, G.W., Chang, H.: Voice recognition and document classification-based data analysis for voice phishing detection. Human-centric Computing and Information Sciences **11**(2) (2021). https://doi.org/10.22967/HCIS.2021.11.002
21. Alsmadi, I., Ahmad, K., Nazzal, M., Alam, F., Al-Fuqaha, A., Khreishah, A., Algosaibi, A.: Adversarial attacks and defenses for social network text processing applications: Techniques, challenges and future research directions. arXiv preprint arXiv:2110.13980 (2021)
22. Jin, D., Jin, Z., Zhou, J.T., Szolovits, P.: Is bert really robust? natural language attack on text classification and entailment. arXiv preprint arXiv:1907.11932 (2019)
23. Kim, H., Song, M., Na, S.H., Shin, S., Lee, K.: When llms go online: The emerging threat of web-enabled llms. arXiv preprint arXiv:2410.14569 (2024)
24. Roy, S.S., Thota, P., Naragam, K.V., Nilizadeh, S.: From chatbots to phishbots?: Phishing scam generation in commercial large language models. In: 2024 IEEE Symposium on Security and Privacy (SP). pp. 36–54 (2024). https://doi.org/10.1109/SP54263.2024.00182
25. Alotaibi, L., Seher, S., Mohammad, N.: Cyberattacks using chatgpt: Exploring malicious content generation through prompt engineering. In: 2024 ASU International Conference in Emerging Technologies for Sustainability and Intelligent Systems (ICETSIS). pp. 1304–1311 (2024). https://doi.org/10.1109/ICETSIS61505.2024.10459698
26. Kudo, T.: Mecab: Yet another part-of-speech and morphological analyzer. https://taku910.github.io/mecab/ (2005), accessed: 2025-05-25
27. Moussavou Boussougou, M.K., Park, D.J.: Attention-based 1d cnn-bilstm hybrid model enhanced with fasttext word embedding for korean voice phishing detection. Mathematics **11**, 3217 (07 2023). https://doi.org/10.3390/math11143217
28. OpenAI: Gpt-4 technical report (2023), https://arxiv.org/abs/2303.08774
29. DeepMind, G.: Gemini: Advanced general-purpose ai (2023), https://deepmind.google/gemini
30. Bai, C., et al.: Qwen technical report (2023), https://arxiv.org/abs/2309.16609