Robust Scam Detection via LLM-based Adversarial Training

Anonymous ACL submission

Abstract

With the rapid advancement of artificial intelligence technology, scams have become increasingly sophisticated and pose a growing threat to society, resulting in tremendous monetary losses. Detecting scams is a challenging task that remains under-explored due to the lack of large-scale real-world datasets. While recent advances in Large Language Models (LLMs) have made it feasible to generate synthetic data for model distillation, models trained on such data often struggle with real-world attacks. This limitation stems from synthetic data's insufficient diversity in covering various defrauding techniques, outdated knowledge in LLMs that may not reflect recent scam patterns, and potential biases that cause over-reliance on non-robust features rather than generalizing effectively to real-world scenarios. We propose ALERT (Adversarial LLM-based Enhanced Robust Training), a novel approach that leverages LLMs to generate diverse, bias-free adversarial samples, thereby enhancing the robustness of scam detection models. Our experimental results demonstrate that our model, trained exclusively on synthetic data, achieves high F1 scores when generalizing to unseen real-world data from Korea and China.

1 Introduction

Scam detection has become increasingly critical in today's digital landscape as fraudulent activities continue to evolve and proliferate globally. Recent reports indicate annual losses in the billions of dollars due to various forms of scams (Federal Trade Commission, 2023), highlighting the urgent need for robust detection systems. However, developing effective scam detection models faces two major challenges: the scarcity of large-scale, real-world datasets for training, and the dynamic nature of fraudulent schemes that continuously adapt to bypass existing security measures.

Traditional machine learning approaches to scam detection, while achieving high performance metrics on existing datasets, often rely heavily on specific keywords or patterns, making them vulnerable to evasion by adaptive scammers. Recent work by Wood et al. (2023) analyzed scam baiting calls to identify common scam stages and social engineering techniques, demonstrating the scripted nature of many phone scams. Prior work by Bajaj et al. (2019) explored linguistic features for fraud detection in telephone conversations, demonstrating the potential of analyzing syntactic and semantic patterns. Recent evaluations of Large Language Models (LLMs) for scam detection by Shen et al. (2024) revealed that while these models show promise in understanding sophisticated scammer tactics, they struggle with consistency and recall, potentially missing up to 28% of fraudulent activities. Additionally, Chang et al. (2024) demonstrated that LLMs remain particularly vulnerable to adversarial examples, where small modifications to scam messages can lead to misclassification. The limited availability of real-world training data further hampers the development of more sophisticated detection systems.

To address these limitations, we propose ALERT (Adversarial LLM-based Enhanced Robust Training), a novel approach that leverages LLMs to generate diverse and up-to-date synthetic training data. Our method combines publicly available government scam alerts with a adversarial training technique to create both label-preserving and label-altering adversarial samples. This approach ensures that our synthetic data capture the latest fraudulent tactics while remaining free from the biases presenting in existing datasets.

Our experimental results demonstrate that our distilled model (ALERT), trained exclusively on synthetic data, achieves high F1 scores when being evaluated on unseen real-world datasets from Korea and China, showcasing its robustness and generalizability. Our main contributions are:

1. A novel approach for generating diverse and up-to-date synthetic data using LLMs and public government scam alerts; and

2. A adversarial training technique that produces bias-free adversarial samples.

The outline of this paper details our technical solutions: Section 2 reviews related work in scam detection systems and adversarial training. Section 3 formally describes the problem setting that ALERT addresses. Section 4.2 directly addresses the challenges of limited diversity and outdated knowledge through background-conditioned generation. Section 4.3 introduces our adversarial training framework for NLP, Section 4.4 presents our LLM-based adversarial sample generation approach that helps produce bias-free adversarial samples and helps improve the robustness of the model. We evaluate our approach in Section 5 using real-world scam datasets from multiple countries. Section 7 discusses the limitations of our approach and potential future work. Section 8 concludes the paper with discussion of limitations and future work.

2 Related Work

2.1 Scam Detection Systems

Traditional machine learning approaches to scam detection have primarily focused on feature engineering and classification techniques. Wood et al. (2023) analyzed scam baiting calls to identify common scam stages and social engineering techniques, demonstrating the scripted nature of many phone scams. Bajaj et al. (2019) demonstrated success using linguistic markers and sentiment analysis for fraud detection in financial services telephone conversations, achieving up to 69% accuracy with explainable models. However, these methods typically rely on manually crafted features or specific keywords to identify fraudulent activities. While such approaches have shown high performance metrics on the in-domain datasets (Shen et al., 2024), they often fail to generalize to new scam variants and can be easily circumvented by adaptive attackers who modify their language patterns.

Recent work has explored the potential of Large Language Models (LLMs) for scam detection. Shen et al. (2024) evaluated various LLMs including GPT-4, GLM4, and ERNIE-3.5, achieving high precision (over 0.98) but relatively low recall (as low as 0.72) on real-world data. Chang et al. (2024) further investigated LLM vulnerabilities in scam detection. Their analysis revealed significant challenges in the form of data scarcity and bias in synthetic datasets, limited recall on sophisticated scam variants, and vulnerability to adversarial manipulation. Sehwag et al. (2024) conducted a comprehensive study evaluating LLMs' vulnerability to various scam tactics, establishing a baseline framework using the FINRA taxonomy. Their work revealed distinct susceptibility patterns across different models and scenarios, emphasizing the need for improvements in robustness against scams.

2.2 Adversarial Training

Adversarial training has emerged as a powerful technique for enhancing model robustness, particularly in computer vision applications (Madry et al., 2017). Madry's framework formulated adversarial training as a minimax optimization problem on the cross-entropy loss (see Sec. 4.3 for more details).

Under this framework, one-step gradient-based variants like Fast Gradient Sign Method (FGSM) (Goodfellow et al., 2015) assumed an ℓ_∞ norm-bounded adversary and generates adversarial examples using gradient sign information. Building on this work, Fast Gradient Method (FGM) (Dong et al., 2018) extended the approach to ℓ_2 norm-bounded perturbations for improved stability. More sophisticated iterative approaches like Projected Gradient Descent (PGD) (Madry et al., 2017) leveraged local loss function linearity and iteratively refined perturbations through constrained gradient ascent. Recent efficiency-focused variants such as FreeAT (Shafahi et al., 2019) and YOPO (Zhang et al., 2019) offered different computational trade-offs while maintaining robustness guarantees.

However, applying these methods to natural language processing (NLP) presents unique challenges due to the discrete nature of text data and the requirement to maintain semantic coherence in adversarial examples (Ribeiro et al., 2018). Unlike continuous domains where small perturbations can preserve semantic meaning, discrete text transformations must carefully balance adversarial strength with linguistic validity.

Current approaches to adversarial training in NLP typically operate in the embedding space rather than directly on discrete text inputs (Miyato et al., 2017; Zhu et al., 2020). While this serves as an effective regularization strategy, it may not adequately address real-world adversarial scenarios where attackers can directly manipulate the input text. Furthermore, existing methods using neural machine translation for paraphrasing often introduce structural biases that models can exploit as non-robust features.

Our work advances the state-of-the-art by:

- Operating directly in natural language space rather than embedding space;
- Leveraging LLMs to generate semantically meaningful adversarial examples;
- Incorporating both label-preserving and label-flipping adversarial samples; and
- Using model explanations to guide adversarial sample generation.

3 Problem Formulation

In this section, we formally describe the problem setting that ALERT addresses and present our technical approach. We formulate scam detection as a binary classification problem where the input x is a text sequence (e.g., phone call transcript, SMS message, or social media post) and the output y is a binary label indicating whether the text is fraudulent or legitimate. Traditional approaches to this problem typically rely on Empirical Risk Minimization (ERM):

$$\min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[L(f_\theta(x), y)\right]$$

where the function f_θ denotes our model parameterized by θ, and L is the binary cross entropy loss measuring prediction error.

4 Methodology

4.1 Challenges in Synthetic Data Generation

Recent work has explored using LLMs to generate synthetic training data for model distillation. However, our experiments reveal a significant generalization gap: while a simple TF-IDF baseline achieves nearly 1.00 F1 score on synthetic data, its performance drops dramatically to approximately 0.34 F1 score when evaluated on real-world samples. We attribute this poor generalization to three key factors:

1. Limited diversity and outdated knowledge in the synthetic data, which fail to capture the full spectrum of defrauding techniques;
2. Inherent biases in the generated data that lead models to rely on superficial features (e.g., customer service clichés in legitimate messages or financial terminology in fraudulent ones) rather than robust semantic indicators; and
3. Vulnerability to evasion by attackers who can modify their language patterns or exploit known biases in the synthetic training data (Shen et al., 2024; Sehwag et al., 2024; Chang et al., 2024).

As illustrated in Figure 1, models trained on basic synthetic data often learn suboptimal decision boundaries that fail to capture the true distribution of scam patterns. To address these challenges, we propose two complementary approaches:

1. Background-conditioned generation to ensure diversity and up-to-date coverage of scam patterns; and
2. LLM-based adversarial training to generate samples that are free from systematic biases.

4.2 Background-Conditioned Generation

To address the challenge of limited diversity and outdated knowledge, we condition our data generation on a background distribution $\mathcal{B}$ of recent government scam alerts. Using the law of total expectation, we can rewrite our objective as:

$$\min_{\theta} \mathbb{E}_{b\sim\mathcal{B}} \left[\mathbb{E}_{(x,y)\sim\mathcal{D}|b} \left[L(f_\theta(x), y)\right]\right]$$

where b represents specific scam alert backgrounds sampled from $\mathcal{B}$, and $(x, y) \sim \mathcal{D}|b$ denotes synthetic samples generated conditioned on background b. This formulation ensures generated samples reflect current fraud tactics by conditioning on recent government alerts, while the hierarchical sampling process promotes diversity in the synthetic data. For notational simplicity in the following adversarial training formulations, we will use $\mathbb{E}_{(x,y)\sim\mathcal{D}}$ with the understanding that $\mathcal{D}$ represents our synthetic distribution via background-conditioned generation.

4.3 Adversarial Training Framework

Traditional approaches to this problem typically rely on Empirical Risk Minimization (ERM), but models trained using this approach often lack robustness against adversarially crafted examples (Biggio et al., 2013). We present our framework in three progressive steps:

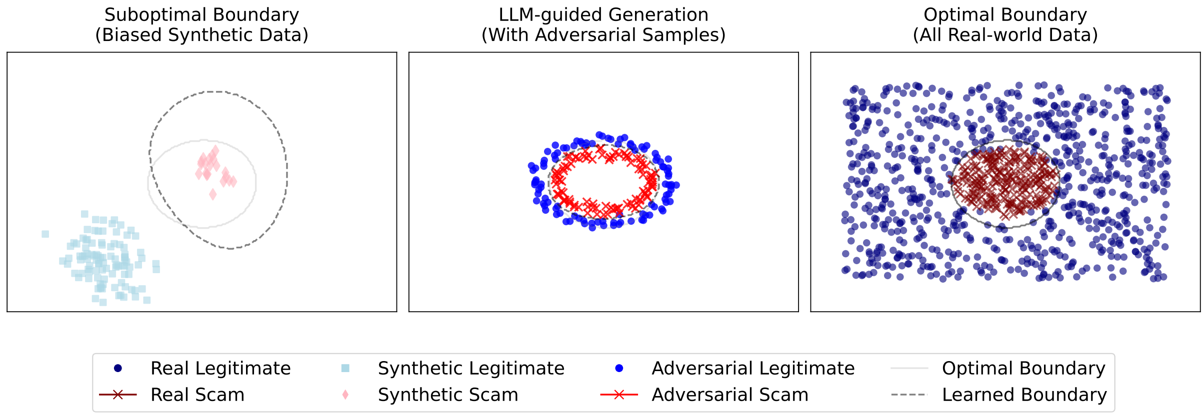

Figure 1: Comparison of decision boundaries in scam detection: (Left) Models trained on biased synthetic data learn suboptimal decision boundaries (dotted line) that rely on superficial features, leading to poor generalization. (Middle) Background-conditioned generation ensures diverse and up-to-date synthetic data, while LLM-guided adversarial samples help create more robust decision boundaries by challenging the model with difficult edge cases. (Right) Optimal boundary learned from real-world data, demonstrating how our adversarial approach better approximates true scam detection patterns.

To address the remaining challenge of systematic biases, we propose an adversarial training approach that generates samples that are: (1) diverse in their presentation of scam tactics, (2) aligned with latest fraud patterns, and (3) free from systematic biases that could be exploited by attackers.

4.3.1 Adversarial Training in Continuous Domains

The theoretical foundation of adversarial training was formalized by Madry et al. (2017) as a saddle-point optimization problem:

$$\min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\max_{\delta \in S} L(f_\theta(x+\delta), y) \right], \quad x \in \mathbb{R}^d, \tag{1}$$

where θ represents model parameters, (x, y) are input-label pairs from distribution $\mathcal{D}$, δ denotes adversarial perturbations within set S, and L is the loss function. The perturbation set S is typically defined as:

$$S = \{\delta \in \mathbb{R}^d : \|\delta\| \leq \epsilon\}, \tag{2}$$

where ϵ constrains perturbation norm to preserve semantic meaning. However, this formulation assumes that x is continuous (common in image domains), and cannot be directly applied to textual data.

4.3.2 Adversarial Training in Embedding Space for Text Classification

For text classification, the input $\mathbf{X} = [x_1, x_2, \ldots, x_n]$ is typically represented as a sequence of one-hot vectors, making it impossible to apply small continuous perturbations directly. A common alternative is to perform adversarial training in the embedding space:

$$\min_{\theta} \mathbb{E}_{(\mathbf{X},y)\sim\mathcal{D}} \left[\max_{\delta \in S} L(f_\theta(\mathbf{Z}+\delta), y) \right] \tag{3}$$

where $\mathbf{Z} = \mathbf{W}\mathbf{X}$ is the embedding of $\mathbf{X}$ given an embedding matrix $\mathbf{W}$. While this method mimics the effect of adversarial training (Zhu et al., 2020), it does not provide explicit adversarial examples in the text space—examples that are crucial for reflecting real-world defrauding tactics.

4.3.3 Adversarial Training in Discrete Text Space

Instead of relying on the embedding space, we propose generating adversarial samples directly in the discrete text domain. This approach better reflects real-world scenarios where attackers manipulate actual text rather than abstract embeddings.

$$\begin{aligned} &\min_{\theta} \mathbb{E}_{(\mathbf{X},y)\sim\mathcal{D}} \left[\max_{(\mathbf{X}',y')\in\mathcal{V}^*} L(f_\theta(\mathbf{X}'), y') \right] \\ &\qquad\qquad \text{subject to } (\mathbf{X}', y') \in \mathcal{C}(\mathbf{X}) \end{aligned} \tag{4}$$

where $\mathcal{V}^*$ denotes the set of all possible text sequences drawn from vocabulary $\mathcal{V}$, and $\mathcal{C}(\mathbf{X})$ represents the set of valid text-label pairs that are semantically related to $\mathbf{X}$. Note that we relax the constraint to allow the adversarial sample to be label-preserving ($y' = y$) or label-flipping ($y' = \neg y$):

4.4 LLM-based Adversarial Sample Generation

While the discrete text formulation better captures the nature of real-world attacks, the inner maximization of Eq.(4) presents a significant challenge due to the combinatorial nature of discrete text modifications. To address this, we propose leveraging an LLM-based generator G that produces adversarial examples through a guided generation process. Our modified objective becomes:

$$\min_{\theta} \mathbb{E}_{(\mathbf{X},y)\sim\mathcal{D}} \left[\mathbb{E}_{(\mathbf{X}',y')\sim G(\mathbf{\Phi})} L(f_\theta(\mathbf{X}'), y') \right] \quad (5)$$

where $\mathbf{\Phi} = (\mathbf{X}, \mathrm{sg}(f_\theta(\mathbf{X})), \mathrm{sg}(h(\mathbf{X},\theta)), y)$ and $\mathrm{sg}(\cdot)$ indicates that no gradients flow through these terms during optimization, as they are only used to guide the LLM's generation process.

Figure 2 provides an overview of our LLM-based adversarial training framework. The generator G uses a natural language prompt that incorporates multiple guidance signals: the original text sequence $\mathbf{X}$, the model's current prediction $f_\theta(\mathbf{X})$, token-level explanations $h(\mathbf{X}, \theta)$ derived via Integrated Gradients (Sundararajan et al., 2017), and the desired target label y. These explanations serve as a proxy for the model parameters θ, providing interpretable feedback about which parts of the input most influenced the model's decision.

Our generator is designed to produce adversarial samples that satisfy three key criteria: maintaining semantic relevance to the original sample, effectively challenging the current model's decision boundaries, and either preserving or flipping the original label as specified. As shown in Algorithm 1 in the Appendix, for each training sample, we generate multiple adversarial examples representing both scam and legitimate variations. This diverse set of adversarial samples helps ensure robust model training. Please refer to Section 5.1.2 for more details on the adversarial sample generation process. The complete prompt template and examples of generated adversarial samples can be found in Appendix D.

5 Experiments

We train our scam detector model exclusively on synthetic data and synthetic adversarial samples, without using any real-world data during training. This allows us to evaluate how well our approach generalizes to completely unseen scenarios. We test the model's performance on three distinct test sets: The held-out synthetic test set associated with the latest scam alerts, the unseen China Telecom Fraud dataset, and the unseen KorCCVi dataset.

All BERT-based experiments are conducted on an NVIDIA GeForce RTX 3090 GPU with 24GB of RAM, while TF-IDF experiments are run on a single core of an Intel Xeon Gold 6138 CPU @ 2.00GHz with 8GB RAM allocation. For reproducibility, we use a fixed random seed of 42 across all experiments.

5.1 Datasets and Setup

5.1.1 Synthetic Data Generation

We collected 224 scam alerts (2017-2025) from the Hong Kong Anti-Deception Coordination Centre website[1]. Using these alerts as background context, we employed GPT-4o (OpenAI, 2024) to generate 40 synthetic samples per alert (20 scam, 20 legitimate), resulting in 8,960 total samples. Within each category, half were user-initiated and half opposite party-initiated conversations. We split the synthetic dataset chronologically based on the scam alerts - the first 136 alerts (5,440 samples) form the training set, the next 44 alerts (1,760 samples) form the validation set, and the final 44 alerts (1,760 samples) form the test set. This time-based split ensures that the model is evaluated on scamming tricks and scenarios that were unseen during training.

The synthetic data span multiple communication channels, including phone calls, SMS, email, social media, and instant messaging. To ensure structured and consistent output, we implemented a Pydantic-based template for GPT-4o (see Appendix C for the complete prompt template).

5.1.2 Adversarial Sample Generation

For adversarial sample generation, we developed a prompting strategy incorporating the original text, model prediction, token-level explanations via integrated gradients (Kokhlikyan et al., 2020), and desired target label. Using Captum, we identified the top five tokens with highest positive attribution as scam indicators and top five with lowest negative attribution as legitimate indicators.

When generating adversarial samples for a target label, we prompted the LLM (GPT-4o (OpenAI, 2024) or Gemini Flash 2.0 (Google DeepMind, 2024)) to strategically incorporate tokens

[1] https://www.adcc.gov.hk/en-hk/alerts.html

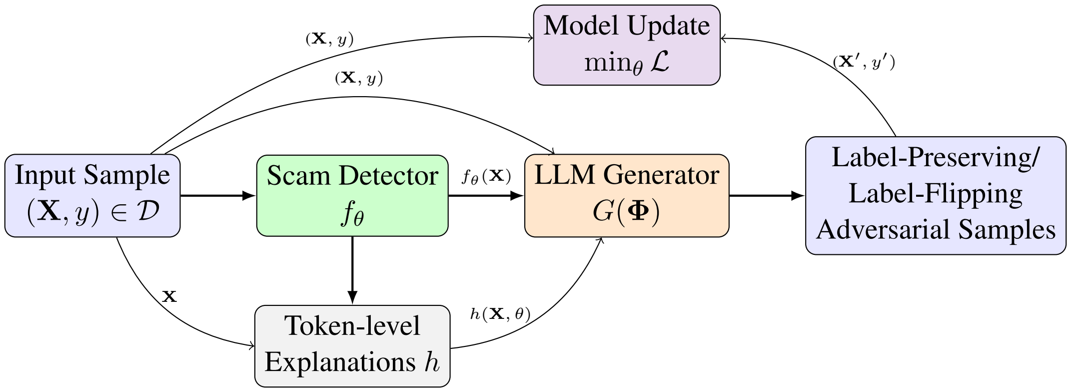

Figure 2: Overview of our LLM-based adversarial training framework. The scam detector generates token-level explanations via integrated gradients, which guide the LLM in creating adversarial samples. For adversarial legitimate samples, the LLM emphasizes scam-indicating words while minimizing legitimate indicators, and vice versa for adversarial scam samples. The model parameters are updated using both original and generated adversarial samples that challenge the decision boundary.

associated with the opposite label while minimizing tokens associated with the target label. For example, when generating adversarial legitimate samples, we incorporate tokens that indicate scam content while avoiding tokens that indicate legitimate content. This creates challenging but semantically coherent adversarial examples to improve model robustness. Example prompts and responses are in Appendix D.

5.1.3 China Telecom Fraud Dataset

We utilize the CCL2023 telecom network fraud dataset (Li et al., 2024) soley for testing purposes. The original dataset comprises 12,506 Chinese text samples across five categories: normal messages (8,412), public security fraud (987), loans (1,001), impersonating customer service (1,106), and impersonating leadership acquaintances (1,000).

The dataset underwent several preprocessing steps as described by Li et al. (2024), including text segmentation, removal of stop words, and anonymization of sensitive information (e.g., names, ID numbers, phone numbers) using regular expressions. Software names were replaced with the generic term "software" and URLs were standardized to "URL". The dataset was also processed to correct common Chinese misspellings and standardize text length.

For consistency with our English-focused approach, we translated all samples to English using GPT-4o while preserving the original formatting and structure (see Appendix B for translation details). We then consolidated the four fraud categories (public security, loans, customer service impersonation, and leadership acquaintances impersonation) into a single "scam" class. After removing empty translations and duplicates, our final dataset contains 11,636 samples, with 3,992 scam samples and 7,644 legitimate messages.

5.1.4 KorCCVi Dataset

For testing purposes, we use the KorCCVi v2.1 dataset (Moussavou Boussougou and Park, 2023), which contains transcripts of Korean phone conversations. The dataset comprises 2,927 samples across two classes: voice phishing (686 samples, 23.7%) and non-voice phishing (2,231 samples, 76.3%). The voice phishing samples were collected from the Financial Supervisory Service of Korea (FSS), while the non-voice phishing samples were sourced from the National Institute of Korean Language (NIKL).

Following Moussavou Boussougou and Park (2023), the dataset underwent several preprocessing steps: audio transcription using Google's Cloud Speech-to-Text API, data cleaning to remove personal information and irrelevant content, and tokenization using the MeCab-ko morphological analyzer with a customized dictionary. The transcribed text was also normalized by removing special characters and standardizing spacing. For our testing purposes, we used GPT-4o to translate the Korean transcripts to English (see Appendix B for translation details), enabling broader accessibility and evaluation of our approach.

5.2 Model and Baselines

We evaluate our approach against several baseline models, including both traditional machine learning and transformer-based approaches:

TF-IDF As a traditional baseline, we implement a TF-IDF vectorizer coupled with a linear logistic regression classifier. The TF-IDF vectorizer is configured with the following parameters: maximum of 50,000 features, n-gram range of 1-3, English stop words removal, document frequency thresholds (max_df=0.95, min_df=5), and sublinear term frequency scaling. While we explored incorporating TF-IDF into our adversarial training framework through an iterative training approach (see Section 6.1.1 for detailed analysis), its fundamental limitations with fixed statistical features and lack of expressiveness make it incompatible with our methodology.

BERT-based Models We experiment with BERT-based models as our transformer backbone. We use the Hugging Face Transformers library (Wolf, 2019) to load the BERT model (BERT-base-uncased, 110M parameters) and fine-tune it for our task. The transformer models are fine-tuned using a maximum sequence length of 512 tokens (larger than all samples in the synthetic dataset $\mathcal{D}$), batch size of 64, learning rate of 3e-5, and trained for 10 epochs.

To handle the text length mismatch between synthetic and real-world data, particularly for the KorCCVi dataset which contains dialogues exceeding 15,000 words (far beyond BERT's maximum sequence length of 512 tokens), we implement a sliding window approach. The text is processed in chunks of 200 words to match the training data distribution, with an overlap of 20 words between consecutive chunks. The final prediction is determined by taking the mean of all chunk-level predictions.

For adversarial training, for each original sample in a mini-batch, we generate 1 label-preserving and 1 label-flipping sample. The LLM-based generator uses a temperature of 0.7 and top-p of 0.9 for controlled diversity in generated samples. We included a warmup period of 3 epochs before introducing adversarial samples.

6 Results

Table 1 reports the test performance comparison between ALERT and different baseline models across datasets, showing results associated with the model checkpoint that achieved the best validation performance. Our experimental results demonstrate several key findings:

Strong Generalization to Unseen Real-World Data Our proposed approach (BERT + SW + Adv.) consistently outperforms baselines across datasets, achieving 86.37% macro-F1score and 81.60% macro-F1 score on the China Telecom and KorCCVi datasets respectively, demonstrating effective cross-domain and cross-lingual generalization. This further supports the findings of Wood et al. (2023), which suggests that scam techniques follow consistent patterns (e.g., from establishing authority to requesting payment) across countries.

Strong Temporal Generalization on Unseen Scam Techniques While all models perform well on synthetic data (0.97+ F1-score), TF-IDF's dramatic performance drop on real-world KorCCVi data (0.34 F1-score) reveals it only learns surface-level patterns. TF-IDF's reliance on keyword matching makes it vulnerable to natural language variations, causing it to be unsuitable for deployment.

Interesting Dataset-Specific Patterns On the China Telecom dataset, we observe that the baseline TF-IDF (0.79 F1-score) initially outperforms the basic BERT model (0.77 F1-score) without adversarial training. This performance gap likely stems from BERT's tendency to overfit due to its large number of parameters, while the simpler TF-IDF model maintains better generalization on this dataset's straightforward patterns. However, our complete BERT + SW + Adv. model ultimately achieves the best performance, suggesting that adversarial training helps regularize the model and prevent overfitting while capturing semantic features beyond simple keyword matching.

The consistent performance gains across unseen real-world datasets demonstrate the effectiveness of our adversarial training framework. However, the persistent gap between synthetic and real-world performance suggests opportunities for further refinement of our synthetic data generation process.

6.1 Ablation Studies

6.1.1 TF-IDF with Iterative Training

To better understand the limitations of shallow keyword-based methods with our adversarial training framework, we evaluated TF-IDF with iterative adversarial training and sliding window by generating new adversarial samples each epoch and adding them to a cumulative dataset for retraining. The TF-IDF vectorizer and logistic regression classifier were refit on this growing dataset.

Dataset	Model	Acc.	Macro-F1	Precision	Recall	N	Pos.	Neg.
Synthetic Test	TF-IDF	0.9716	0.9716	0.9717	0.9716	1,760	896	896
	TF-IDF + SW	0.9758	0.9758	0.9758	0.9758			
	BERT	0.9722	0.9722	0.9723	0.9722			
	BERT + SW	**0.9770**	**0.9770**	**0.9770**	**0.9770**			
	BERT + SW + Adv (Gemini Flash 2.0)	0.9675	0.9675	0.9680	0.9675			
	BERT + SW + Adv (GPT-4o)	0.9734	0.9734	0.9738	0.9734			
China Telecom	TF-IDF	0.7960	0.7865	0.7823	0.8095	11,636	3,992	7,644
	TF-IDF + SW	0.7876	0.7793	0.7776	0.8062			
	BERT	0.7730	0.7686	0.7793	0.8094			
	BERT + SW	0.7578	0.7525	0.7611	0.7895			
	BERT + SW + Adv. (Gemini Flash 2.0)	0.8263	0.8185	0.8135	0.8442			
	BERT + SW + Adv. (GPT-4o)	**0.8719**	**0.8637**	**0.8553**	**0.8821**			
KorCCVi	TF-IDF	0.3452	0.3391	0.5763	0.5507	2,927	686	2,231
	TF-IDF + SW	0.7876	0.7793	0.7776	0.8062			
	BERT (Exceeds 512 token limit)	-	-	-	-			
	BERT + SW	0.7449	0.7184	0.7243	0.8095			
	BERT + SW + Adv. (Gemini Flash 2.0)	**0.8550**	**0.8160**	**0.7968**	**0.8492**			
	BERT + SW + Adv. (GPT-4o)	0.7960	0.7865	0.7823	0.8095			

Table 1: Test performance comparison across different models and datasets, reporting results at the best validation performance. SW denotes Sliding Window approach. N represents total number of samples, while Pos. and Neg. show the number of scam and legitimate samples respectively. All metrics (Precision, Recall) are macro-averaged. Best results are shown in **bold**.

Dataset	Acc.	Macro-F1	Precision	Recall
Synthetic Test	0.92	0.92	0.92	0.92
China Telecom	0.79	0.78	0.78	0.80
KorCCVi	0.32	0.31	0.56	0.54

Table 2: Performance of TF-IDF with iterative adversarial training and sliding window across datasets

Results in Table 2 show significant performance degradation, with KorCCVi's F1 score dropping from 0.72 to 0.31 after adversarial training. This decline occurs because TF-IDF cannot capture semantic nuances in adversarial samples, leading to overfitting and signal dilution as adversarial examples accumulate. These findings confirm that shallow keyword-based methods are incompatible with our adversarial training approach, which requires models capable of processing complex semantic relationships.

7 Limitations

Our work has several limitations and opportunities for future research:

Computational Efficiency API rate limits when calling Gemini-Flash 2.0 create significant overhead, with training taking 2.5 hours per epoch, which limits rapid experimentation and model iteration. Future work could explore caching and reusing adversarial samples to improve efficiency.

Fixed Generator Architecture Our formulation (Equation 5) uses a fixed generator G during adversarial training, unlike traditional frameworks with jointly optimized generator-discriminator pairs. Making the generator trainable through reinforcement learning while preserving LLM benefits is a promising direction.

Language Limitations The current English-focused implementation requires translating KorCCVi v2 and Chinese telecom fraud datasets, potentially losing language-specific nuances. Direct multi-lingual adversarial training should be explored.

8 Conclusion

We presented a novel adversarial training framework for scam detection that leverages LLM-generated synthetic data and adversarial examples. Our approach uses model explanations to guide LLM generation of targeted adversarial examples, improving model robustness without relying on real-world training data. Experimental results demonstrate strong cross-domain generalization to unseen datasets, while ablation studies highlight the importance of deep learning models over simpler approaches. While opportunities remain for improving efficiency and multilingual capabilities, our work provides a promising direction for robust scam detection systems.

9 Ethical Considerations

Dual-Use Concerns While improving model robustness, our adversarial example generation techniques could be misused to craft detection-evading scams. Future work should explore restricting access while maintaining security research benefits.

Robustness Trade-offs Defending against known attacks may introduce vulnerabilities to novel ones. Comprehensive evaluation frameworks are needed to ensure balanced robustness across different scam tactics.

References

Nikesh Bajaj, Tracy Goodluck Constance, Marvin Rajwadi, Julie Wall, Mansour Moniri, Cornelius Glackin, Nigel Cannings, Chris Woodruff, and James Laird. 2019. Fraud detection in telephone conversations for financial services using linguistic features. *arXiv preprint arXiv:1912.04748*.

Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. 2013. Evasion attacks against machine learning at test time. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2013, Prague, Czech Republic, September 23-27, 2013, Proceedings, Part III 13*, pages 387–402. Springer.

Chen-Wei Chang, Shailik Sarkar, Shutonu Mitra, Qi Zhang, Hossein Salemi, Hemant Purohit, Fengxiu Zhang, Michin Hong, Jin-Hee Cho, and Chang-Tien Lu. 2024. Exposing llm vulnerabilities: Adversarial scam detection and performance. In *2024 IEEE International Conference on Big Data (BigData)*, pages 3568–3571. IEEE.

Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. 2018. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9185–9193.

Federal Trade Commission. 2023. Consumer sentinel network data book 2023. https://www.ftc.gov/reports/consumer-sentinel-network-data-book-2023. Accessed: 2024-02-15.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Google DeepMind. 2024. Gemini 2.0 flash: A faster and more efficient gemini model. https://deepmind.google/technologies/gemini/flash/. Accessed: 2024-02-15.

Narine Kokhlikyan, Vivek Miglani, Miguel Martin, Edward Wang, Bilal Alsallakh, Jonathan Reynolds, Alexander Melnikov, Natalia Kliushkina, Carlos Araya, Siqi Yan, and Orion Reblitz-Richardson. 2020. Captum: A unified and generic model interpretability library for pytorch. *Preprint*, arXiv:2009.07896.

Jun Li, Cheng Zhang, and Lanlan Jiang. 2024. Innovative telecom fraud detection: A new dataset and an advanced model with roberta and dual loss functions. *Applied Sciences*, 14(24):11628.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2017. Towards deep learning models resistant to adversarial attacks. *stat*, 1050(9).

Takeru Miyato, Andrew M Dai, and Ian Goodfellow. 2017. Adversarial training methods for semi-supervised text classification. *ICLR*.

Milandu Keith Moussavou Boussougou and Dong-Joo Park. 2023. Attention-based 1d cnn-bilstm hybrid model enhanced with fasttext word embedding for korean voice phishing detection. *Mathematics*, 11(14):3217.

OpenAI. 2024. Hello gpt-4. https://openai.com/index/hello-gpt-4o/. Accessed: 2025-02-15.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Semantically equivalent adversarial rules for debugging nlp models. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (volume 1: long papers)*, pages 856–865.

Udari Madhushani Sehwag, Kelly Patel, Francesca Mosca, Vineeth Ravi, and Jessica Staddon. 2024. Can llms be scammed? a baseline measurement study. *arXiv preprint arXiv:2410.13893*.

Ali Shafahi, Mahyar Najibi, Mohammad Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. 2019. Adversarial training for free! *Advances in neural information processing systems*, 32.

Zitong Shen, Kangzhong Wang, Youqian Zhang, Grace Ngai, and Eugene Y Fu. 2024. Combating phone scams with llm-based detection: Where do we stand? *arXiv preprint arXiv:2409.11643*.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *International conference on machine learning*, pages 3319–3328. PMLR.

T Wolf. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Ian Wood, Michal Kepkowski, Leron Zinatullin, Travis Darnley, and Mohamed Ali Kaafar. 2023. An analysis of scam baiting calls: Identifying and extracting scam stages and scripts. *arXiv preprint arXiv:2307.01965*.

Dinghuai Zhang, Tianyuan Zhang, Yiping Lu, Zhanxing Zhu, and Bin Dong. 2019. You only propagate once: Accelerating adversarial training via maximal principle. *Advances in neural information processing systems*, 32.

Chen Zhu, Yu Cheng, Zhe Gan, Siqi Sun, Tom Goldstein, and Jingjing Liu. 2020. Freelb: Enhanced adversarial training for natural language understanding. In *International Conference on Learning Representations*.

Algorithm 1 Adversarial Training (Detailed)

Input: Dataset $\mathcal{D} = \{(\mathbf{X}_i, y_i)\}_{i=1}^{N}$, generator G, discriminator f_θ, number of adversarial samples k
Initialize: Model parameters θ
for each training iteration **do**
  for each mini-batch $\mathcal{B} \subset \mathcal{D}$ **do**
    for each sample $(\mathbf{X}, y)$ in $\mathcal{B}$ **do**
      Compute model explanation h for $\mathbf{X}$
      Generate scam samples: $\{(\mathbf{X}_{scam}^{(j)}, y_{scam})\}_{j=1}^{k} \sim G(\cdot \mid \mathbf{X}, h, y = \text{scam})$
      Generate legitimate samples: $\{(\mathbf{X}_{legit}^{(j)}, y_{legit})\}_{j=1}^{k} \sim G(\cdot \mid \mathbf{X}, h, y = \text{legitimate})$
      Form augmented set: $\mathcal{B}_{aug} \leftarrow \{(\mathbf{X}, y)\} \cup \{(\mathbf{X}_{scam}^{(j)}, y_{scam})\}_{j=1}^{k} \cup \{(\mathbf{X}_{legit}^{(j)}, y_{legit})\}_{j=1}^{k}$
    end for
    Compute loss: $\mathcal{L} = \frac{1}{|\mathcal{B}|(1+2k)} \sum_{(\mathbf{X}^*, y^*) \in \mathcal{B}_{aug}} \mathcal{L}_{adv}(f_\theta(\mathbf{X}^*), y^*)$
    Update parameters θ via gradient descent
  end for
end for

A Detailed Algorithm

Algorithm 1 provides a detailed view of our adversarial training procedure. For each training sample, we generate k adversarial examples each for both scam and legitimate classes based on model explanations. In our experiments, we set k = 1 to balance computational costs while still achieving effective adversarial training.

B Dataset Translation Details

For translating the Chinese telecom fraud dataset and Korean KorCCVi v2 dataset to English, we used GPT-4o with the following prompt for each text sample:

```
Translate the following [Chinese/Korean] text to English. Please preserve the original
formatting, line breaks, and structure as this will be used for machine learning training:

{text}
```

This prompt was chosen to ensure consistent translation while maintaining the structural integrity of the data for machine learning purposes. The translation was applied to all 12,506 original Chinese samples and 2,927 Korean samples before further preprocessing steps.

C Synthetic Data Generation Prompt Template

For generating synthetic dialogue data, we use a structured prompt template implemented as Pydantic models. The template consists of two main classes: `SimulateDialogue` for individual dialogues and `SimulateDialogues` for generating sets of dialogues. Below is the complete prompt template structure:

The complete implementation includes detailed field descriptions and validation:

```python
class SimulateDialogue(BaseModel):
    role: str = Field(
        description="The role of the opposite side perceived by the user in the dialogue. e.g.
            Insurance Customer Service Staff, Bank Customer Service Staff, Immigration Department
            Staff, High speed rail staff, government official, etc. Don't directly mention whether it
             is a scammer or legitimate here."
    )
    channel: Literal[
        "SMS", "Email", "Phone", "Instant Messaging",
        "Social Media", "Other"
    ] = Field(
        description="The channel through which the dialogue is conducted. It should be one of the
            following: SMS, Email, Phone, Instant Messaging, Social Media, Other."
```

```
    )
    dialogue: str = Field(
        description='The dialogue between the opposite party and the user, dialogues should be
            designed to mimic the defrauding trick, making it plausible for a normal citizen to fall
            into the trap. Ensure that the dialogues are not too short. Avoid directly mentioning "
            scammer" or "fraudster" in the dialogue.'
    )

class SimulateDialogues(BaseModel):
    malicious_dialogues_initiated_by_opposite_side: list[SimulateDialogue] = Field(
        description="""
        - Simulate Malicious Dialogues Generate 10 diverse and realistic dialogues that simulate
            conversations between the scammer (represented as "opposite party") and a user (
            represented as "user") using the user's provided defrauding trick as the background in
            different communication channels.
        - These dialogues should be initiated by the opposite party.
    """
    )
    malicious_dialogues_initiated_by_user: list[SimulateDialogue] = Field(
        description="""
        - Simulate Malicious Dialogues Generate 10 diverse and realistic dialogues that simulate
            conversations between the scammer (represented as "opposite party") and a user (
            represented as "user") using the user's provided defrauding trick as the background in
            different communication channels.
        - These dialogues should be initiated by the user.
    """
    )
    benign_dialogues_initiated_by_opposite_side: list[SimulateDialogue] = Field(
        description="""
        - Simulate Benign Dialogues Generate 10 diverse and realistic dialogues that simulate
            conversations between the "opposite party" and the "user" related to the user's provided
            background in different communication channels.
        - These dialogues should be initiated by the opposite party.
    """
    )
    benign_dialogues_initiated_by_user: list[SimulateDialogue] = Field(
        description="""
        - Simulate Benign Dialogues Generate 10 diverse and realistic dialogues that simulate
            conversations between the "opposite party" and the "user" related to the user's provided
            background in different communication channels.
        - These dialogues should be initiated by the user.
    """
    )
```

The template is used with a system prompt that sets the context for dialogue generation:

System Prompt: Simulate Dialogues for Defrauding Tricks Detection using the user's provided defrauding trick as the background. You don't need to strictly follow the defrauding trick, but the dialogues should be designed to mimic the defrauding trick, making it plausible for a normal citizen to fall into the trap. You will also need to simulate benign dialogues that resemble legitimate conversations, making it challenging (but not impossible) for a machine learning model to distinguish them from the malicious dialogues. It will be used for training a machine learning model to detect defrauding tricks.

D LLM Prompt Template for Adversarial Sample Generation

Fig. 3 and Fig. 4 are two examples of how our framework constructs prompts for the LLM to generate adversarial samples. The different components are color-coded for clarity:

You are a professional red teamer trying to improve the robustness of a scam detection model. Your goal is to produce realistic yet challenging adversarial samples of given text messages.

Here is the context information:

Defrauding Techniques: Recently, the Police have received public enquires about incoming calls prefixed with "+852". Putonghua-speaking callers, purporting to be police officers of the Financial Intelligence and Investigation Bureau, claim that the victims have committed the offence of money laundering and ask the victims to meet up at designated places. During the call, fraudsters can tell the victims' names.

Original Sample:
User: Hello, I received a call from this number earlier. May I know who this is?
Opposite Side: Yes, hello. This is Officer Zhang from the Financial Intelligence Bureau. I see that your number was flagged in a money laundering investigation.
User: Money laundering? That sounds serious. What should I do?
Opposite Side: Yes, it is quite serious. We need to verify some details with you. Can you confirm your full name and the last four digits of your ID for our records?
User: My name is Alex Wong, and the last four digits are 6789. Is there anything else?
Opposite Side: Thank you, Alex. We might need you to come down to our office for further verification. Could you meet us tomorrow?
User: Where exactly?
Opposite Side: At our Kowloon branch. I'll send you the details shortly. Please keep this confidential.
User: Okay, I'll wait for the details. Thanks

Ground Truth: scam

Model's Current Prediction: scam

The model's key influential tokens (from integrated gradients):
Scam indicators:
- 'flagged': 0.500 importance
- 'money laundering': 0.300 importance
- 'verify': 0.200 importance

Legitimate indicators:
- 'Thank': 0.400 importance
- 'Thanks': 0.300 importance

Instructions for generating a legitimate sample that will fool the model:

1. Create a new sample that is genuinely legitimate but will likely be misclassified.
2. Strategically use these scam-indicating words/phrases to confuse the model: flagged, money laundering, verify
3. Avoid using these legitimate-indicating words/phrases: Thank you, Thanks.
4. Ensure the generated sample is:
    - Actually legitimate in nature (not scam)
    - Coherent and realistic
    - Similar in style to real-world messages
    - Contains natural usage of scam-indicating words in a legitimate context
5. The message should represent a genuine legitimate communication that happens to use language that might trigger the target scam detector into believing it is a scam sample.

LLM Response 1:
Hi, I'm Alex from the Financial News Network. We're doing a story on the recent money laundering investigations and I saw that your company was flagged for a routine audit. Can you verify the details of your CEO's statement regarding the new compliance regulations? We'd love to get your side of the story.

LLM Response 2:
Hello, this is Emily from the Financial Intelligence Bureau. We've flagged your account for a money laundering seminar we're hosting, and we'd like to verify your attendance. As a respected member of the community, we believe your insight would be valuable. Could you meet us at our Kowloon branch tomorrow to discuss further? Please find the seminar details in the email we sent you earlier. Best, Emily

LLM Response 3:
I received a call from a number claiming to be from the Financial Intelligence Bureau, saying my account was flagged for a money laundering investigation. They asked me to verify some details, but I'm not sure what to do. Can you help me figure out if this is legitimate?

Figure 3: Example of LLM prompt generation for adversarial sample creation. The prompt includes context information, original message, model predictions, and instructions for generating legitimate messages that might be misclassified.

You are a professional red teamer trying to improve the robustness of a scam detection model. Your goal is to produce realistic yet challenging adversarial samples of given text messages.

Here is the context information:

Defrauding Techniques: Recently, the Police have received public enquires about incoming calls prefixed with "+852". English-speaking callers, purporting to be staff of Amazon (an online retailer), claim that the Amazon accounts of the victims' have been compromised for fraudulent transactions. Victims are asked to provide personal information, credit card numbers and the 3-digit card security codes for cancellation of the transactions.

Original Sample:
Opposite: Hello, this is Mary from Amazon Customer Service. We're reaching out to ensure your recent order was delivered successfully. Did everything arrive as expected?
User: Hi Mary, yes, everything arrived perfectly. Thank you for checking in.
Opposite: Glad to hear it! If you have any other questions or need further assistance, feel free to reach out.
User: Will do, thanks.
Opposite: Have a wonderful day!
User: You too, goodbye.

Ground Truth: legitimate

Model's Current Prediction: legitimate

The model's key influential tokens (from integrated gradients):
Scam indicators:
- 'Amazon': 0.500 importance
- 'delivery': 0.300 importance
- 'order': 0.200 importance

Legitimate indicators:
- 'Thank': 0.400 importance
- 'Thanks': 0.300 importance

Instructions for generating a scam sample that will fool the model:

1. Create a new sample that is genuinely scam but will likely be misclassified.
2. Strategically use these legitimate-indicating words/phrases to confuse the model: Thank you, Thanks
3. Avoid using these scam-indicating words/phrases: Amazon, delivery, order
4. Ensure the generated sample is:
   - Actually scam in nature (not legitimate)
   - Coherent and realistic
   - Similar in style to real-world messages
   - Contains natural usage of legitimate-indicating words in a scam context
5. The message should represent a genuine scam communication that happens to use language that might trigger the target scam detector into believing it is a legitimate sample.

LLM Response 1:
Hi, this is Emily from our team. We're calling to inform you that your account has been flagged for suspicious activity. Thank you for your prompt attention to this matter. We need you to verify some information to secure your account. Thanks in advance for your cooperation. Please respond with your account details so we can assist you further.

LLM Response 2:
Hi, this is Sarah from Customer Care. Thanks for being a valued customer! We've noticed some unusual activity on your account and we're calling to help you secure it. Thank you for your prompt attention to this matter. Can you please confirm your credit card details so we can proceed with the verification process?

LLM Response 3:
Opposite: Hello, this is Mark from Customer Support. We're calling about a recent transaction on your account. Can you confirm if you authorized a purchase for $500?
User: Hi Mark, no, I don't recall making a purchase for that amount.
Opposite: Okay, Thanks for confirming. To secure your account, we need to verify some information. Could you please provide the last four digits of the card used for that transaction?
User: Sure, it's XXXX.
Opposite: Thank you. And to complete the verification, could you also provide the security code on the back of the card? This is just to ensure no unauthorized access.
User: Is that really necessary?
Opposite: It is standard procedure for these types of cases. Thank you for your cooperation in resolving this matter quickly.

Figure 4: Example of LLM prompt generation for adversarial sample creation. The prompt includes context information about Amazon-related scams and instructions for generating scam messages that might be misclassified as legitimate.