Large-Scale Multilingual SMS Fraud Detection For Telecom Networks

Orlando Amaral Cejas and **Yuejun Guo** and **Qiang Tang**
{orlando.amaral-cejas,yuejun.guo,qiang.tang}@list.lu
Luxembourg Institute of Science and Technology, 4362 Luxembourg

Abstract

Short Message Service is a fundamental communication channel in modern telecom networks, yet its ubiquity continues to be exploited for large-scale attacks. Recent advances in multilingual embeddings and large language models have shown strong performance on general text classification tasks. However, their effectiveness and efficiency for multilingual SMS fraud detection under constraints of real telecom environments, remain underexplored. In particular, existing studies largely focus on monolingual datasets, cloud-based inference, or overlook the constraints imposed by real telecom environments. In this work, we investigate large-scale multilingual SMS classification for HAM, SPAM, and SMISHING detection using embedding models. We construct and curate a proprietary multilingual SMS dataset and conduct a systematic evaluation of four different multilingual embedding models. Using the obtained dataset, we fine-tune the models, demonstrating that domain-adapted embeddings significantly improve SMS classification across several languages. Overall, this study addresses the gap between embedding-centric NLP research and real-world telecom requirements, providing empirical and practical insights for effective and efficient deployment of multilingual SMS fraud detection systems. We open source all non-proprietary material at: https://figshare.com/s/1df3ca8d08a4eb4d2712.

1 Introduction

Short Message Service (SMS) remains a core communication channel in global telecom networks, supporting a wide range of services including authentication, notifications, and person-to-person messaging. Despite the proliferation of internet-based messaging platforms, SMS continues to be widely used due to its universality, low latency, and independence from data connectivity. This ubiquity, however, has made SMS the recurrent target of malicious activities, particularly unsolicited messages (SPAM) and SMISHING (SMS-based PHISHING). These aim at deceiving users into disclosing sensitive information or performing fraudulent actions. Recent industry reports and academic studies highlight a steady increase in SMS-based fraud, with attackers adopting increasingly sophisticated social engineering techniques to evade detection and exploit user trust (Almeida et al., 2011; Chiew et al., 2018).

A defining characteristic of modern SMS fraud is its multilingual and cross-regional nature. Telecom operators routinely handle traffic spanning dozens of languages, dialects, and writing styles, often within the same geographical region. Fraud campaigns frequently leverage code-switching, transliteration, and obfuscation to bypass traditional filters, significantly complicating automated detection (Ramanujam et al., 2022). While early SMS SPAM detection systems relied on rule-based approaches or classical machine learning with handcrafted features, these methods struggle to generalize when facing evolving attack patterns and linguistic diversity (Timko and Rahman, 2023).

Recent advances in deep learning and large language models (LLMs) have demonstrated strong performance in text classification tasks, including SPAM and PHISHING detection (Devlin et al., 2019; Conneau et al., 2020). However, most state-of-the-art solutions assume cloud-based inference or monolingual scenarios. Such assumptions are often incompatible with regulated telecom environments, where strict requirements on data privacy, sovereignty, and customer confidentiality limit the feasibility of off-premise processing. In addition, cloud-centric solutions introduce latency, operational costs, and scalability concerns that are misaligned with SMS throughput requirements.

These constraints motivate the need for local SMS classification, where models are deployed on-premise or at the network edge within the operator's infrastructure. Local processing enables com-

Proceedings of the Second International Conference on Natural Language Processing and Artificial Intelligence for Cyber Security (NLPAICS'2026)
11–12 June, 2026.

pliance with data locality regulations, reduces inference latency, and provides predictable operational costs. However, the local deployment of models has intrinsic challenges. SMSs are extremely short, noisy, and informal, limiting the amount of contextual information available for classification. Moreover, real-world telecom datasets are highly imbalanced, with legitimate messages (HAM) vastly outnumbering SPAM and SMISHING instances, making reliable threat detection particularly difficult (Munoz and Islam, 2025). Finally, local deployment imposes strict limits on model size, memory footprint, and inference speed, constraining the choice of embedding models.

In this work, we address these challenges by focusing on large-scale multilingual SMS fraud detection for telecom networks, with an emphasis on efficiency and effectiveness. Our overall goal is to obtain and evaluate a local SMS classification pipeline capable of accurately distinguishing between HAM, SPAM, and SMISHING messages across multiple languages, while meeting the intrinsic operational constraints of telecom systems. Therefore, the main contributions of this work are:

1. Creation and curation of DS4, a proprietary large-scale multilingual SMS dataset derived from telecom traffic that reflects real-world language diversity and SMS characteristics.

2. Benchmarking of three open source SPAM detectors and four embedding models, finetuned for SPAM detection (using DS4), in public and proprietary datasets.

3. Benchmarking of four embedding models, finetuned for SPAM and SMISHING detection (using DS4), in public and proprietary datasets.

4. Impact analysis of personally identifiable information (PII) and component-level finetuning on SPAM and SMISHING detection performance.

2 Background and Related Work

In this section we position our work within existing literature and expose the most common gaps.

2.1 SPAM Detection

Early research on SMS SPAM detection largely adopted rule-based solutions and feature-engineering approaches, inspired by email SPAM filtering. These methods rely on manually crafted rules, keyword blacklists, and shallow textual features such as text length, token frequency, and presence of URLs or phone numbers. While effective in constrained settings, rule-based systems were shown to be brittle and difficult to maintain under rapidly evolving SPAM strategies (Chiew et al., 2018).

To address these limitations, supervised machine learning techniques became the dominant paradigm. Almeida et al. introduced one of the first publicly studied SMS SPAM datasets and evaluated classifiers such as Naïve Bayes and Support Vector Machines (SVMs), demonstrating substantial gains over rule-based filtering (Almeida et al., 2011). Subsequent work explored more expressive feature representations and classifiers, including Random Forests (RF) and ensemble methods, improving robustness against lexical variation (Gómez Hidalgo et al., 2006).

With the rise of deep learning, neural architectures such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) were applied to SMS and short-text classification, enabling automatic feature learning from raw text (Zhang and Wallace, 2017; Le-Hong and Le, 2018). Although these models improved detection accuracy, they are often trained and evaluated on small, monolingual datasets, limiting their applicability to real-world telecom environments.

2.2 SMISHING Detection

SMISHING represents a more targeted and harmful class of SMS abuse compared to generic SPAM. While SPAM messages typically focus on unsolicited advertising, SMISHING messages are designed to deceive users into revealing credentials, financial information, or installing malicious software. As a result, SMISHING detection requires deeper semantic understanding and contextual awareness. Academic studies have shown that SMISHING messages differ from SPAM in both linguistic structure and intent (Joo et al., 2017; Seo et al., 2024).

In-depth analysis of large-scale SMISHING campaigns highlights the use of urgency cues, impersonation of trusted entities, and adaptive message templates. Detection approaches have therefore incorporated not only lexical features, but also semantic representations and behavioral indicators, such as sender patterns and campaign-level sim-

ilarities (Marchal et al., 2014). However, many SMISHING-focused studies rely on auxiliary metadata that may not always be available or usable due to privacy constraints in telecom systems, reinforcing the importance of content-based classification.

2.3 Multilingual Text Classification

Modern telecom networks operate in inherently multilingual environments, where SMS traffic spans multiple languages, scripts, and writing conventions. This has motivated the adoption of multilingual representation learning, particularly transformer-based models pre-trained on large multilingual corpora. Models such as multilingual BERT (Dufter and Schütze, 2020; Kulshreshtha et al., 2020) and XLM-RoBERTa (Conneau et al., 2020) have demonstrated strong cross-lingual transfer capabilities across a wide range of NLP tasks. To improve efficiency and applicability to sentence-level tasks, several multilingual sentence embedding models have been proposed, enabling compact semantic representations useful for subsequent classification steps (Reimers and Gurevych, 2019).

Despite these advances, multilingual SMS classification remains challenging due to text brevity, informal language, code-switching, and the presence of obfuscation techniques commonly used by attackers. Prior work has shown that short-text classification exacerbates data sparsity issues and limits the effectiveness of contextual modeling, particularly in low-resource languages. These challenges are insufficiently addressed in existing studies, which often assume monolingual or high-resource settings (Lee and Dernoncourt, 2016).

2.4 LLMs for Text Classification

More recently, LLMs have emerged as a powerful tool for text classification, either through fine-tuning or prompt-based inference. Fine-tuned transformer models consistently outperform classical architectures on supervised classification tasks, including SPAM and PHISHING detection (Yang et al., 2019). Prompt-based approaches, while attractive due to their flexibility and minimal training requirements, have been shown to suffer from instability, sensitivity to prompt design, and inconsistent performance across domains (Brown et al., 2020).

In security-sensitive applications such as SMS classification, prompt-based inference present additional concerns. Its reliance on cloud-based inference, lack of determinism, and high computational cost limit their suitability for production deployment in telecom environments. Recent studies have highlighted that fine-tuned, smaller-scale models often provide better trade-off between accuracy and efficiency than larger-scale, general-purpose models when used locally (Sanh et al., 2019).

2.5 Telecom-Specific Constraints

Unlike many other domains, telecom systems are subject to strict operational and regulatory constraints. SMS traffic frequently contains personal or sensitive information, imposing requirements on data privacy, retention, and local processing. From a systems perspective, SMS classification must operate at high throughput with low latency, often under tight memory and compute budgets. Models deployed in such environments must therefore balance detection performance against resource consumption and operational stability (Edozie et al., 2025). Most of the existing work do not explicitly account for these constraints, which limits the practical applicability of many proposed approaches.

2.6 Gap Analysis

The reviewed literature reveals several key gaps that motivate this work. First, there is a lack of large-scale multilingual SMS datasets derived from realistic telecom settings, with most studies relying on small or monolingual corpora. Second, while LLMs have shown capabilities for handling text classification, there is limited research on the feasibility of local LLM-based SMS classification, explicitly balancing accuracy, latency, and resource usage. Finally, existing studies provide insufficient analysis of how different finetuning choices affect SMS classification performance, particularly in highly imbalanced HAM–SPAM–SMISHING scenarios. This paper addresses these gaps by presenting a large-scale multilingual SMS classification study grounded in telecom constraints, resulting in domain-adapted embedding models ready to be used in production.

3 Study Design

This work is explicitly guided by constraints of real telecom environments, where scalability, effectiveness, and efficiency are critical. Under these operational requirements, inference must remain deterministic, low-latency, and executable on on-premise hardware with no GPU infrastructure available. Therefore, LLMs are considered a non-viable option. Their computational cost, memory foot-

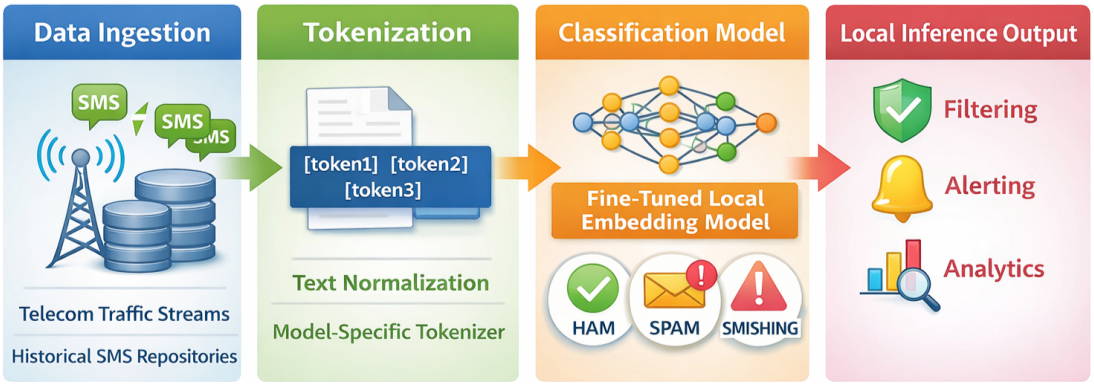

Figure 1: Fraud Detection Pipeline

print, and deployment complexity are not aligned with carrier-grade throughput and cost constraints. Instead, we focus on compact multilingual embedding models, which provide a substantially better effectiveness–efficiency trade-off for these specific business needs.

3.1 Problem Definition

We formulate fraud detection as a multi-class text classification task, where each incoming SMS is assigned to one of the following mutually exclusive classes HAM, SPAM, or SMISHING. HAM messages correspond to legitimate user or service communications, SPAM messages include unsolicited or bulk advertising content, and SMISHING messages represent malicious attempts to deceive users into revealing sensitive information or performing fraudulent actions. The objective is to learn a function that maps an SMS to its corresponding class label, while maintaining high accuracy for the minority threat classes, particularly SMISHING.

Thus, with our work we aim to answer the following research questions:

- RQ1 - How well do finetuned embedding models perform compared to open source SPAM detectors?
- RQ2 - How well do embedding models perform when finetuned for SPAM and SMISHING detection?
- RQ3 - How does the inclusion of PII and finetuning different combinations of components impact detection performance?

The research questions encompassed by this study allow us to evaluate the capability of the implemented pipeline to detect SPAM and SMISHING threats in real-world telecom data.

3.2 Study Overview

Figure 1 illustrates the end-to-end fraud detection pipeline encompassing four stages. First, SMSs are collected from different sources. Then, each message is tokenized and passed to a finetuned embedding model that predicts one of the target classes. Finally, the predicted labels are made available to the subsequent parts of the telecom infrastructure for filtering, alerting, and analytics. Notice that even when our study considers all four stages, it mainly focuses on stages 2 and 3.

3.3 Dataset Preparation

For our experiments, we employed three open source datasets (DS1 - DS3) and one proprietary dataset (DS4). As seen in Table 1, in total, the open source datasets contain $\approx$ 17,000 labeled SMSs in English. The datasets DS1 and DS2 have two labels (i.e., HAM and SPAM), while DS3 has three (i.e., HAM, SPAM, and SMISHING).

Dataset ID	DS1	DS2	DS3	Total
Number of SMS	5573	5574	5971	17118

Table 1: Open source datasets details.

DS4 is derived from real-world telecom SMS traffic, and it is obtained in the Data Ingestion stage (Figure 1). The data was collected in accordance with internal policies and applicable data protection regulations. Messages originate from multiple regions and reflect realistic traffic patterns observed in operational networks. The dataset includes messages in multiple languages, covering both high-resource and low-resource settings. A summary of the language distribution is provided in Table 2. Some of the languages encompassed by the category *Other* are Arabic, Chinese, and Spanish.

SMS Language	Number of Messages
English	≈ 36K
French	≈ 13K
German	≈ 11K
Italian	≈ 5K
Hindi	≈ 2K
Other	≈ 19K
All	≈ 86K

Table 2: Languages encompassed by DS4.

In DS4, labels were assigned according to clear operational criteria, HAM are legitimate messages, SPAM are unsolicited or promotional messages without direct malicious intent, and SMISHING are messages designed to deceive users through impersonation, urgency cues, or fraudulent requests. DS4 contains 70423 HAM, 7244 SPAM, and 8339 SMISHING messages. Annotation was performed by telecom expert ensuring consistency with telecom security practices, and ambiguous cases were analyzed by at least three of them. The experts involved in the dataset curation recognized the challenging nature of tis task, proven by long sessions held to separate SPAM from SMISHING messages. This is due to the need of considering the intention of the sender. This is identified as a highly nuanced detail and not a straightforward learning task. Additionally, a remark was made regarding inconsistencies (mislabeling) that could yet exist in the dataset. These details could influence negatively in the performance of the inference stage. Furthermore, to evaluate the influence of PII in the solution performance, we kept two versions of DS4. In one of them, all PII was removed or irreversibly anonymized prior to the SMS tokenization.

3.4 Selection of Embedding Models

Embedding models are an essential component of the pipeline, as they constitute the core of the text classification task. Model selection was guided by the following criteria: positioning in the MTEB leaderboard for text classification tasks, multilingual coverage, embedding dimensionality, latency, and memory footprint. Based on the mentioned criteria, we consider as candidates the four embeddings showed in Table 3. Notice that the tokenization of the text is performed using the tokenizer associated with each of the candidates to maintain consistency in all tasks (e.g., training, inference).

4 Experimental Setup and Results

In this section we explain in details the performed experiments. We cover the used datasets and splits performed (i.e., train, validation, and test), as well as leveraged embedding models. Notice that all experiments were executed tenfold. Each time, we generate new train, validation, and test sets and made sure all considered labels are present in every set. In our experiments, we compute three averaged metrics (i.e., precision, recall, and accuracy), based on correctly and misclassified messages. A message is correctly classified when the predicted label matches the original label in the dataset. Conversely, when the predicted label does not match the label in the dataset, the message is considered misclassified. We conducted all the experiments on a laptop with Windows 11 Enterprise OS, 32 GB of RAM, an Intel Core i7-1370P 1.90 GHz CPU, and a 128 MB Intel Iris Xe graphics card.

4.1 Model Benchmarking for SPAM Detection

In this experiment we compare the performance of four embedding models (i.e., M1, M2, M3, M4), once finetuned, against three Hugging Face-hosted SPAM detectors (i.e., SD1,SD2, SD3). No further finetuning was performed over the SPAM detectors, their current hosted versions were used.

4.1.1 Setup

We employ the anonymized version of DS4 to finetune the four candidates in Table 3 for text classification. This allows the models to adapt to domain-specific language patterns. The models are finetuned to directly optimize multi-class prediction performance, ensuring the required stable and deterministic behavior. The training objective is defined using a categorical cross-entropy loss only over two target classes (i.e., HAM, SPAM). This loss formulation directly reflects the multi-class nature of the task but does not make yet the distinction between unsolicited and malicious messages. For this, we adapted the datasets DS3 and DS4 to encompass only labels of the two target classes. This means that we removed all the samples labeled as SMISHING, leaving DS3 with 4844 HAM and 489 SPAM messages and DS4 with 70423 HAM and 7244 SPAM messages.

Finetuning is performed in a supervised setting using the 80% of DS4, namely $Training$, and the finetuned versions are tested over the entire DS1, DS2, DS3, and the 20% of DS4, namely $Testing$.

ID	Model Name	Languages Supported	Embedding Dimensions	Millions of Parameters
M1	lealla-base	+100	192	$\approx$ 100
M2	multilingual-e5-small	+100	384	$\approx$ 100
M3	multilingual-e5-base	+100	768	$\approx$ 300
M4	multilingual-e5-large	+100	1024	$\approx$ 600

Table 3: List of embedding models.

We ensured a dataset partitioning that preserved the original class distribution. This ensured a real representation of minority classes across training, validation, and testing subsets. For instance, the original class distribution of DS4 is $\approx$90,67% HAM and $\approx$9,33% SPAM, while the class distribution in $Testing$ is $\approx$90,70% HAM and $\approx$9,30% SPAM. This evidences the stratified split performed.

Hyperparameters are selected to balance convergence stability and computational efficiency. A small learning rate of $2e-4$ is employed to avoid forgetting the pretrained representations, while an eight batch training is used to fit within memory constraints. Training is conducted over the pooler layer for 10 epochs and checkpoints are saved at each epoch. The checkpoint achieving the best validation performance is selected for final evaluation over the mentioned datasets. All four models are fine-tuned under identical hyperparameter to ensure a fair comparison across architectures. This fine-tuning setup reflects operational constraints typical of telecom environments, prioritizing stable optimization, reproducibility, and predictable inference behavior over aggressive model adaptation.

4.1.2 Results

Table 4 shows the accuracy achieved for all used datasets, as well as the time needed for classifying the $\approx$ 15,500 SMSs in $Testing$. The execution time is proportional to the amount of messages contained in each dataset (see Table 1). Notice that we do not report on metrics such as precision and recall. This is because the goal of this experiment is to make a preliminary selection of possible candidates for the next experiment (SPAM and SMISHING detection).

As shown in Table 4, the results obtained for the public datasets are similar for all the Hugging Face-hosted SPAM detectors and finetuned embedding models. However, the performance remains stable in $Testing$ just for the finetuned models. This reveals the limited capabilities of the detectors to identify SPAM messages in real data. Additionally, the results suggest that all four candidates are able to accurately discern between HAM and SPAM messages in both public and real data. Trying to narrow down the candidate selection, we consider a requirement gathered from telecom experts. The processing time per message must be under $35e-3$ seconds to be deemed as real time (i.e., feasible deployment). This means that, for instance, the processing time for the 15,500 SMSs in $Testing$ should be around 9 minutes. Given their performance, all four finetuned models are possible candidates for SPAM and SMISHING detection.

Detector / Model	DS1 (%)	DS2 (%)	DS3 (%)	$Testing$ (%)	Time (min)
SD1	97.50	97.58	95.46	80.74	$\approx$ 1.5
SD2	98.76	98.70	97.88	81.10	$\approx$ 2.5
SD3	99.32	99.26	98.30	82.58	$\approx$ 8.0
M1	95.35	95.10	94.77	93.68	$\approx$ 7.2
M2	94.26	96.93	95.18	93.63	$\approx$ 7.0
M3	98.80	98.64	97.99	95.37	$\approx$ 7.5
M4	96.91	94.10	95.14	94.17	$\approx$ 7.8

Table 4: HAM-NOT HAM detection & time.

The answer to RQ1 is that all finetuned models perform better than available SPAM detectors.

4.2 Model Benchmarking for SPAM and SMISHING Detection

Here we aim at finding out which one out of the preselected candidates performs best not when discerning between SPAM and SMISHING messages.

4.2.1 Setup

As commented before in Section 2, SMISHING represents a more dangerous class of SMS. These messages aim at, for instance, deceiving users into revealing sensitive personal information. Trying to address this fact highlighted by telecom experts, class weighting is applied during finetuning. We assign higher importance to this class without altering the underlying data distribution. For this experiment, we also finetune the candidates in Ta-

ble 3 to directly optimize multi-class prediction performance. However, this time we try to capture the subtle semantic distinctions between SPAM and SMISHING. Thus, no labels were removed from DS3 or DS4. The training objective is defined using a categorical cross-entropy loss over three target classes (i.e., HAM, SPAM, and SMISHING). This loss formulation encourages clear separation between benign, unsolicited, and malicious messages. Here, we perform the same DS4 split, 80% for $Training$ and 20% for $Testing$ preserving the original class distribution. In this case, while the original class distribution of DS4 is ≈81,88% HAM, ≈8,42% SPAM, and ≈9,70% SMISHING, the class distribution in $Testing$ is ≈81,92% HAM, ≈8,40% SPAM, and ≈9,68% SMISHING.

4.2.2 Results

To better understand models behavior, we analyze the results for both, overall performance and performance for each target class (i.e., HAM, SPAM, and SMISHING). Moreover, since we expect to gain insights of a multilingual classification performance, we base our evaluation on DS4, which is the only dataset encompassing multiple languages.

Model	Pre (%)	Rec (%)	Acc (%)	F1 (%)
M1	**86.25**	**75.00**	**93.00**	**80.23**
M2	83.63	65.61	89.89	73.53
M3	84.64	64.70	89.78	73.34
M4	85.38	63.17	90.01	72.61

Table 5: SMS Classification in $Testing$.

Table 5 exhibits the global metrics computed for all four models when classifying the ≈ 16,000 SMSs in $Testing$. As revealed, while precision and accuracy are similar for all models, M1 has significantly better recall and F1 measure.

Model	Pre (%)	Rec (%)	Acc (%)	F1 (%)
M1	**96.30**	**97.43**	**94.83**	**96.86**
M2	95.91	97.17	94.29	96.54
M3	95.31	97.21	94.27	96.25
M4	95.44	97.11	94.22	96.27

Table 6: HAM Detection.

Tables 6, 7, and 8 display the per-class metrics computed for all four models when classifying the ≈ 16,000 SMSs in $Testing$ as HAM, SPAM, or SMISHING. The results reveal that while not far from the other models, M1 consistently presents the best detection metrics even when it has the smallest embedding dimensions (Table 3). We attribute the significant difference between the recall for SPAM and SMISHING detection, across all evaluated finetuned models, to stronger discriminative semantic patterns in SMISHING messages than generic SPAM. This suggests that SMISHING messages exhibit stronger and more consistent semantic indicators than generic SPAM messages.

Model	Pre (%)	Rec (%)	Acc (%)	F1 (%)
M1	**72.32**	**55.75**	**94.49**	**62.96**
M2	70.14	54.17	94.21	61.13
M3	71.73	54.32	94.36	61.82
M4	70.69	54.77	92.64	61.72

Table 7: SPAM Detection.

Model	Pre (%)	Rec (%)	Acc (%)	F1 (%)
M1	**79.89**	**87.81**	**96.68**	**83.66**
M2	76.78	86.64	93.65	81.41
M3	78.33	86.48	94.20	82.20
M4	77.88	87.42	95.61	82.37

Table 8: SMISHING Detection.

Unlike conventional SPAM, which encompasses a broad and heterogeneous range of promotional or advertising content that frequently overlaps with legitimate HAM communications, SMISHING messages typically contain highly discriminative phishing-related patterns such as credential requests, urgent calls to action, impersonation attempts, and malicious URLs. These characteristics create clearer semantic boundaries that transformer-based models can learn more effectively. The consistency of the results across all four models further indicates that the observed performance gap is primarily driven by intrinsic dataset and class-structure properties rather than by model architecture differences.

The answer to RQ2 is that all finetuned models perform well for the identification of HAM and SMISHING messages. Conversely, their performance fall short when identifying SPAM messages. Overall, M1 shows the best performance, having similar execution time (i.e., between 7 and 8 minutes) and higher accuracy for SPAM and SMISHING identification. Due to its performance and aiming at answering RQ3, M1 is further explored

in our third and last experiment.

4.3 Influence of Finetuning Different Components of M1 Structure

In this experiment, we finetune different combinations of the components in M1 structure (i.e., embedding layer, encoder, pooler layer). Since we aim at analyzing the influence of PII over this task, we employ both versions of DS4 (i.e., anonymized, non-anonymized). In all cases we address the problem as a multi-class text classification task, considering HAM, SPAM, and SMISHING messages.

4.3.1 Setup

To find out about the influence of the different components of the model M1 in the finetuning task, we explore all other possible combination: I) only embedding layer, II) only encoder, III) embedding layer + encoder, IV) embedding layer + pooler layer, V) encoder + pooler layer, VI) embedding layer + encoder + pooler layer. Furthermore, we perform the finetuning employing both version of DS4. This means that we ended having 12 additional versions of the finetuned model, which we evaluated over each of the respective $Testing$ subsets (i.e., anonymized and non-anonymized). Due to the poor results obtained for the majority of the combinations, we report only one of them (i.e., embeddings + pooler layer). The behavior observed in the rest of the combinations suggests that finetuning only the embedding layer perturbs the input representation space while keeping the encoder fixed, which breaks the alignment the encoder expects. Similarly, finetuning the encoder destabilizes the pretrained semantic representations, particularly under strong class imbalance. As a result, the model converges toward the dominant class (HAM), leading to a collapse of the decision boundary and failure to detect SPAM and SMISHING messages.

4.3.2 Results

Tables 9 and 10 report only on combination IV for both, the anonymized (A) and non-anonymized (N-A) versions of $Testing$. The results of the remaining five combinations are not included here due to the poor performance achieved during the identification task. The results obtained for both versions of DS4 are not far apart. This suggests that the presence of PII does not have a significant impact on the SMS classification task. This means that the telecom organization can decide whether to anonymized the text before its processing.

Metric	Pre (%)	Rec (%)	Acc (%)	F1 (%)
HAM	96.72	98.39	95.95	97.55
SPAM	82.05	63.86	95.79	71.82
SMISHING	91.34	95.57	98.69	93.41

Table 9: Anonymous Finetuning Results.

Metric	Pre (%)	Rec (%)	Acc (%)	F1 (%)
HAM	96.34	97.39	94.85	96.86
SPAM	71.73	55.21	94.46	62.40
SMISHING	80.26	88.47	96.66	84.16

Table 10: Non-anonymous Finetuning Results.

We believe that the better results obtained with the anonymized data may be attributed to the removal of noisy PII elements (e.g., phone numbers or identifiers), which typically carry high lexical variability but limited semantic information for SMS classification and can hinder model generalization. By replacing such tokens with consistent placeholders, anonymization reduces vocabulary sparsity and prevents the model from overfitting to dataset-specific identifiers. This encourages the classifier to focus on more meaningful linguistic and semantic patterns associated with HAM, SPAM, and SMISHING messages, ultimately leading to slightly improved and more robust classification performance. Notice that after finetuning the embeddings, the processing time for $Testing$ increased by $\approx$ 1 minute, which takes it to a total of $\approx$ 8.5 minutes.

The answer to RQ3 is twofold. First, by finetuning the embedding and pooler layers, M1 achieves its best performance in identifying SPAM and SMISHING messages in real data. Second, although it exists, the impact of PII in the identification performance is substantial only for the SPAM and SMISHING classes.

5 Discussion

Overall, our findings confirm that both embedding selection and finetuning details play a decisive role in balancing detection performance under constraints of real telecom environments.

5.1 Important Outcomes

Across all experiments interesting outcomes, related to representational expressiveness, computa-

tional cost, and inference performance, are visible. First, the use of larger embeddings does not imply higher classification performance. This is evidenced in RQ2, where M1, the model with the smallest embedding dimensions achieves the best performance. Second, bigger models do not always entail an increased latency during the text classification task. This is reflected in RQ1 and RQ2, where M4, despite being six times larger, takes a very similar time to the rest of the models to classify 16,000 SMS. Third, our study demonstrate that more compact embedding models can exhibit better inference performance when properly finetuned on domain-specific data. This indicates that domain adaptation can compensate for reduced model dimensions and initial capabilities. This can be seen in RQ1, where M1-M4 once finetuned using proprietary data, show better performance than available models (SD1-SD3), finetuned using public data.

5.2 Telecom Systems Implications

Our solution is required to operate under the previously mentioned strict carrier-grade constraints linked to the existing telecom infrastructure. From a practical perspective, our results suggest that state-of-the-art multilingual SMS classification does not necessarily require cloud-based LLMs. Instead, locally deployable models, when carefully selected and finetuned, can meet predefined requirements. The performance observed across multiple languages indicates that a single multilingual model can replace language-specific pipelines, simplifying system architecture and maintenance.

6 Conclusion

Our work suggests that effective multilingual SMS classification depends less on model scale and more on domain adaptation and efficiency-aware design, enabling robust, scalable, and privacy-preserving protection against evolving SPAM and SMISHING threats. The preliminary results obtained evidence that embedding models finetuned with real data are capable of performing better than available models finetuned using public data. This performance improvement is highlighted when trying to identify HAM, SPAM, and SMISHING messages in real data. Additionally, the results suggest a possible use of M1 for large-scale telecom SMS classification. This compact multilingual sentence-level embedding model provides a favorable balance between classification performance, memory footprint, and inference latency. This emphasizes that model selection should be driven not only by raw accuracy but also by throughput requirements, operational stability, and ease of integration into existing infrastructure.

Acknowledgements

This research was conducted in close collaboration with telecom experts, whose insights were instrumental in shaping this work.

Limitations

Despite the encouraging results, several limitations should be acknowledged. First, the dataset, while large and multilingual, is derived from a specific telecom context and may not fully capture regional variations in fraud tactics or linguistic patterns. Second, we do not analyze the performance across individual languages. Therefore, while the models are trained and evaluated on multilingual data, the results do not explicitly quantify potential performance variations across languages. Consequently, potential variations in detection accuracy across individual languages remain an open question for future work. This is translated in a possible lower performance for low-resource languages. Hence, the continued need for careful monitoring and potential data augmentation strategies in underrepresented regions. Third, the experimental evaluation focuses on content-based classification and does not incorporate traffic-level or behavioral signals that could further enhance SPAM and SMISHING detection. Moreover, the study evaluates models under static conditions and does not address concept drift, which is a known challenge in adversarial domains such as fraud detection. Finally, while local inference constraints are considered in terms of latency and memory, broader system-level factors, such as integration with existing filtering pipelines and real-time feedback loops, are outside of this work scope.

Ethics Statement

The author(s) have not employed any Generative AI tools to write the paper.

References

Tiago A Almeida, José María G Hidalgo, and Akebo Yamakami. 2011. Contributions to the study of sms spam filtering: new collection and results. In *Proceedings of the 11th ACM symposium on Document engineering*, pages 259–262.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Kang Leng Chiew, Kelvin Sheng Chek Yong, and Choon Lin Tan. 2018. A survey of phishing attacks: Their types, vectors and technical approaches. *Expert Systems with Applications*, 106:1–20.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 8440–8451.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the Conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 4171–4186.

Philipp Dufter and Hinrich Schütze. 2020. Identifying elements essential for bert's multilinguality. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 4423–4437.

Enerst Edozie, Aliyu Nuhu Shuaibu, Bashir Olaniyi Sadiq, and Ukagwu Kelechi John. 2025. Artificial intelligence advances in anomaly detection for telecom networks. *Artificial Intelligence Review*, 58(4):100.

José María Gómez Hidalgo, Guillermo Cajigas Bringas, Enrique Puertas Sánz, and Francisco Carrero García. 2006. Content based sms spam filtering. In *Proceedings of the ACM symposium on Document engineering*, pages 107–114.

Jae Woong Joo, Seo Yeon Moon, Saurabh Singh, and Jong Hyuk Park. 2017. S-detector: an enhanced security model for detecting smishing attack for mobile computing. *Telecommunication Systems*, 66(1):29–38.

Saurabh Kulshreshtha, Jose Luis Redondo Garcia, and Ching Yun Chang. 2020. Cross-lingual alignment methods for multilingual bert: A comparative study. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 933–942.

Phuong Le-Hong and Anh-Cuong Le. 2018. A comparative study of neural network models for sentence classification. In *2018 5th NAFOSTED Conference on Information and Computer Science (NICS)*, pages 360–365.

Ji-Young Lee and Franck Dernoncourt. 2016. Sequential short-text classification with recurrent and convolutional neural networks. In *Proceedings of the 2016 conference of the north american chapter of the association for computational linguistics: human language technologies*, pages 515–520.

Samuel Marchal, Jérôme François, Radu State, and Thomas Engel. 2014. Phishstorm: Detecting phishing with streaming analytics. *IEEE Transactions on Network and Service Management*, 11(4):458–471.

Miriam L Munoz and Muhammad F Islam. 2025. Deep learning approaches for multi-class classification of phishing text messages. *Journal of Cybersecurity and Privacy*, 5(4):102.

E. Ramanujam, K. Shankar, and Arpit Sharma. 2022. Multi-lingual spam sms detection using a hybrid deep learning technique. In *2022 IEEE Silchar Subsection Conference (SILCON)*, pages 1–6.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Jae Woo Seo, Jong Sung Lee, Hyunwoo Kim, Joonghwan Lee, Seongwon Han, Jungil Cho, and Choong-Hoon Lee. 2024. On-device smishing classifier resistant to text evasion attack. *IEEE Access*, 12:4762–4779.

Daniel Timko and Muhammad Lutfor Rahman. 2023. Commercial anti-smishing tools and their comparative effectiveness against modern threats. In *Proceedings of the 16th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, pages 1–12.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

Ye Zhang and Byron C Wallace. 2017. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing*, pages 253–263.