Using LLM-Generated Data to Create a Roman Urdu Scam Call Detector

Sameed Irfan[1], Aswad Sheeraz[2], Muhammad Hasnain[3]

[1]Department of Computer Science, Bloomfield Hall College, Multan, Pakistan
[2,3]Department of Computer Science, Beaconhouse College Program, Multan, Pakistan

Abstract: The issue of scam calls is on the rise, with losses expected to exceed $1 trillion globally in 2024. While easily incorporating Machine Learning has been effective in countering scam calls, the dominant models continue to suffer from glaring insufficiencies. Most models can only detect monolingual scam calls, and LLM-based solutions, though they can be multilingual, are impractical due to the resources they require. Furthermore, scam call tactics are constantly changing; hence, many models can become outdated. To address these challenges, this paper proposes a structure where a model is trained on LLM-generated data, allowing for a multilingual and easy-to-update dataset. To test the accuracy of these models, a small dataset of human-written scam and non-scam call dialogues was used. This model was trained on synthetic data and tested on real-world scam calls data, achieving, on average, over 90% accuracy and f1_score.

Keywords: LLM, Scam Call Detection, Machine Learning, Training models with Synthetic Data, Urdu Scam Call Detector
Email: khawajahasnain666@gmail.com

1. Introduction

Scam calls are fraudulent calls that deceive people, and most of the time, they trick the person into sending them money. These can be especially problematic for senior citizens and uneducated people, especially in villages. The scammers also target people in dire need of money to exploit their current circumstances. These calls have certain patterns that can be identified using Machine Learning; as a result, a high level of accuracy has been reached in detecting these calls. The main problem is that these solutions aren't practical. As the solutions relying on models trained by the manual collection of data would require these datasets to be constantly updated. Also, actual recordings of scam calls are rarely available. On the other hand, the solutions relying on LLM could only run on web servers in a more practical application. This would mean that a constant internet connection on the client side would be required.

Previous works mostly include detecting spam calls in English, and many tools such as Truecaller help in achieving this, but our main focus is on preventing local scams, especially for underprivileged people, and to make a scam call dataset to even help agencies.

2. Related Work

Scam and spam detection in low-resource languages like Roman Urdu has gained importance due to the increasing frequency of fraudulent communication. Early studies using traditional

machine learning models such as Naïve Bayes, SVM, and Extremely Randomized Trees achieved strong results but were limited by the manual collection of training data [1], [2]–[4]. Recent work has expanded detection methods to include both voice and text, improving the identification of phone scams through conversational and acoustic cues [3]. To overcome data scarcity, researchers have started generating synthetic datasets using large language models (LLMs), which have shown high accuracy and strong generalization in scam detection tasks [2], [4], [5]. The spam filtering capabilities of technology have been greatly improved with the advent of multilingual transformer models [6]. In light of such advancements, the current study attempts to use the LLM-generated Roman Urdu data along with lightweight classifiers to develop a novel scam call detection system that is efficient, precise, and highly scalable.

3. Data Collection

Innovative language models, including ChatGPT, helped in producing a bulk fake dataset with which the model was trained. These models were tasked with creating realistic epitomes of telephone dialogues, which were tagged as either a scam or a non-scam, following given prompts and samples of scam and non-scam transcripts. Each of these transcripts was designed as a realistic conversation and saved in CSV format with appropriate labels. To ensure variety and exhaust all potential deep structures, we optimized deep-search plus agentic prompting and subsequently engineered over twenty thousand mix-scam and non-scam dialogues spanning various phishing, impersonation, and financial frauds, and genuine conversations like lottery scams.

Table 01: Preview of the LLM-generated Dataset

Data	Label
Mera bank ke masla hua wa aap aapna account number bhj sakta hu please	scam
deye gaye number par rabta kerke ainaam jete	scam
Baba kidher hain?	non-scam
Hn yaar kesa ha	non-scam
Ap ka lucky draw nikla hai apna account number da da for transfer.	scam

Testing, however, began with real-world data. A family and friends' spreadsheet with real call transcripts was filled out and shared. Each participant classified the calls as "scam" or "non-scam" based on their personal judgment. This process resulted in more than 100 genuine samples, which provided a realistic baseline with which to assess the model. Unlike the synthetic training data, these transcripts portrayed the messy, human aspects of conversations, i.e., pauses, common phrases, and informal words, which scams use to feel more realistic and believable.

The separation of synthetic data for training and authentic data for testing ensured that the model was trained on a large, controlled, and synthetic dataset, and its performance was evaluated on real and local authentic scams.

4. Methodology

4.1 Pre-processing

One of the crucial, yet often neglected, aspects of constructing any language-related detection systems is pre-processing. Unlike the formal domain datasets, scam calls are, by their nature, disorganized. They typically possess informal vocabulary, switch between languages, have the presence of pauses and fillers, and attempt to confuse and/or pressure the targeted victim. The system will end up 'understanding' only noise rather than 'understanding' meaning if such data is input into the model with no regard to cleaning and encoding it. This is the reason why our work focuses on developing a 'hardcore' pre-processing system that can automate the cleaning of both the multilingual setting and the peculiarities of scam speech.

To structure this set of texts into a more useful format, the data was processed under a TF-IDF (Term Frequency–Inverse Document Frequency) vectorizer. TF-IDF considers a word's frequency in a given transcript and its uniqueness in the whole corpus. This means generic terms like "hello" and "thank you," which would be ubiquitous and non-informative, would not be given as much prominence in the learning process of the model, while more indicative terms of scams, such as "transfer," "verify," and "bank account," were given more weight. This resulted in a sparse matrix with a high number of dimensions, where each call transcript formed a row and each word feature was a column with an assigned weight indicating its significance.

Each transcript was subsequently assigned a label in which "1" signified a scam and "0" denoted a non-scam call. This binary classification allowed the models to learn from actionable and differentiated ground-truth signals. Unlike other studies, which rely on human annotations, this study uses LLM-generated synthetic data to broaden the range of different scam scripts. This approach helps to address the shortage of multilingual scam call datasets

and opens a new area of research focused on the real, human-generated calls' generalization potential of models trained on AI-augmented datasets.

4.2 Models

Following the pre-processing stage, the next important step in building the multilingual scam call detector was the selection and training of machine learning models. Given our focus on building a compact, highly effective system capable of functioning in multilingual environments and on resource-poor devices, we preferred classical machine learning models over bulky neural networks. This was driven by two primary research considerations. First, there is a need for interpretability and low computational cost. Second, there is a need to establish a benchmark on how well traditional models perform on a dataset enriched with LLM-generated synthetic scam transcripts.

We compared four distinct machine learning algorithms, each with unique strengths in the treatment of the text data represented in TF-IDF. These models were trained on a mixed dataset comprising human-annotated and LLM-generated transcripts and were then evaluated on a holdout human-only set to test for real-world generalization.

a) Support Vector Classification (SVC)

From a historical perspective, SVMs, especially with linear kernels, have been able to classify text with remarkable accuracy due to their capabilities in hyperplane class separation in high dimensions. For this problem, we trained a classifier to maximize the margin between the “scam” and “non-scam” calls in the TF-IDF feature space. SVC is a classifier very useful for sparse, high-dimensional feature sets, like text, which is why it serves as a strong baseline in this research. Also, we were able to evaluate the SVC scalability with multilingual data since the scam keyword in various languages usually results in non-linear SVC decision boundaries, which is a strong suit of SVC.

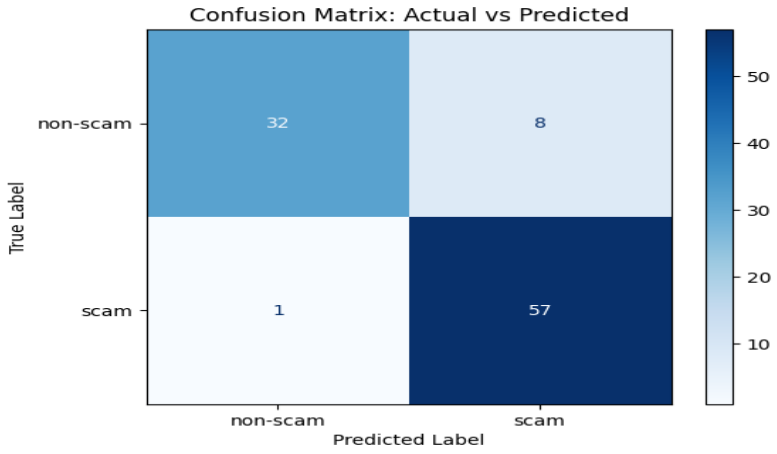

Figure 01: Confusion Matrix for SVC

b) Logistic Regression

Although quite a basic model, Logistic Regression is still quite powerful and an interpretable model in its own right for classification problems. It predicts the likelihood of a given call being a scam by considering the weighted contributions of TFIDF features and estimating probabilities of scam classification. Logistic Regression's strength is in its efficiency in processing and predicting large-scale data while still being interpretable—for every scam-related vocabulary, there is a corresponding weight to the feature. This model has proved quite useful for multilingual transcripts in scam-detection by the relevance of phrases like "premio", "INAAM", "verify account", and other phrases. The fact that the model is capable of explaining its predictions is important in establishing confidence in the system, especially in the real world, where the model ultimately helps users to identify calls being flagged as fraudulent.

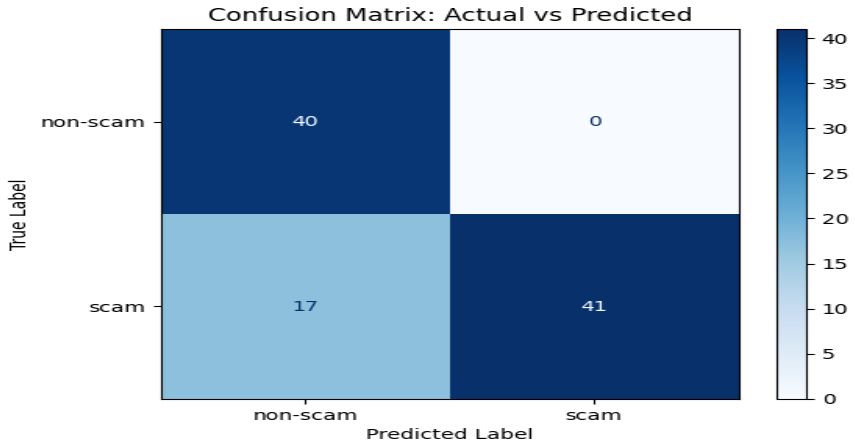

Figure 02: Confusion Matrix for Logistic Regression

c) Naive Bayes

The Naive Bayes classifier, and in particular the Multinomial type, is "tailor-made" for text categorization, particularly in cases where word frequencies are the primary distinguishing characteristics. The independence of features assumption, for all its naivety, tends to work surprisingly well in practice for natural languages. For our dataset, Naive Bayes worked particularly well in pattern detection, where particular scam words recurred in large volumes in the calls. Also, due to the effectiveness of Naive Bayes in terms of computational resources and memory, it has promising capabilities for deployment in real time on low-power systems. This is very suitable for the "light" detector aim of this study.

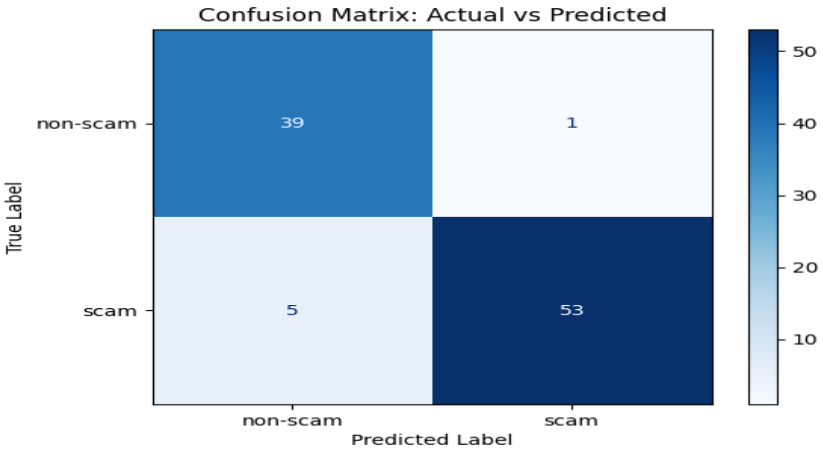

Figure 03: Confusion Matrix for Naive Bayes

d) Extra Trees Classifier

The Extra Trees Classifier implements the ensemble learning approach, where many uncorrelated decision trees are built and the outputs of the trees are combined to improve classification results. As opposed to methods like Logistic Regression or SVC, Extra Trees is capable of capturing intricate interactions between features without the need for underlying linearity. When applied to scam detection, the model was able to treat multi-word scam phrases such as “urgent transfer required” or “government grant approved” as collective signals instead of isolated words. Extra Trees is more computationally expensive than Naïve Bayes; it is still quite efficient regarding deep learning architectures and gives an initial understanding of the extent to which tree-based ensembles can exploit the diversity of real and LLM-generated transcripts.

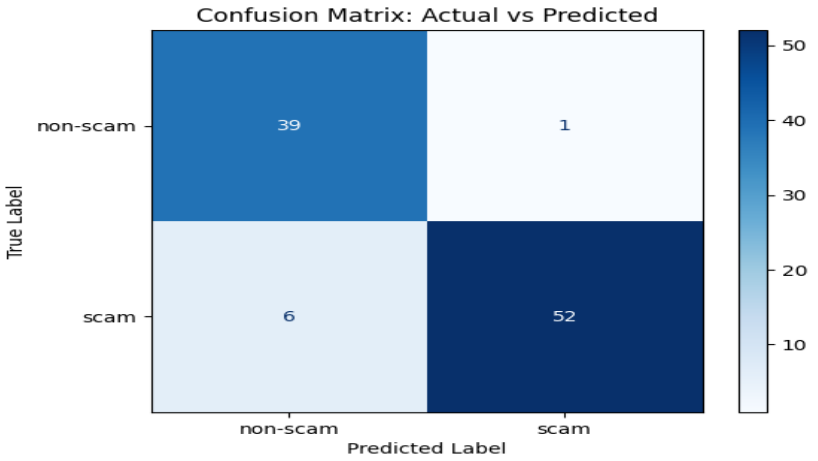

Figure 04: Confusion Matrix for Extra Trees Classifier

5. Comparative Analysis

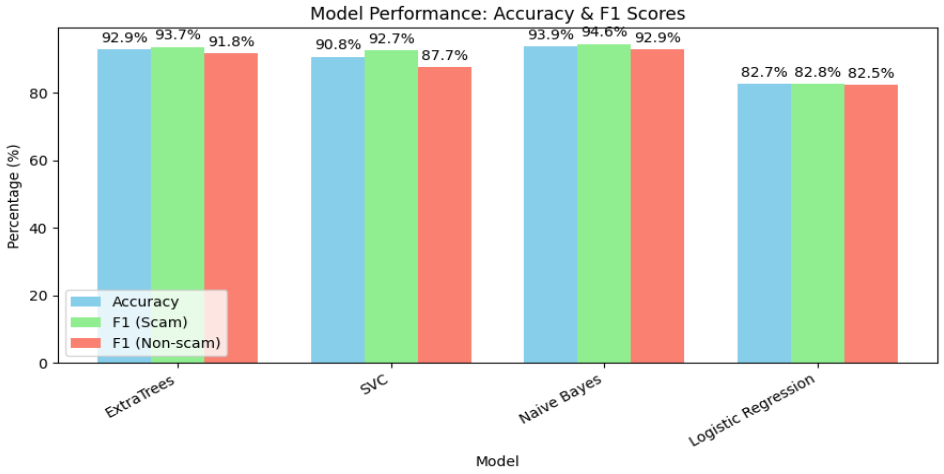

Figure 05 Accuracies and F1_scores (For Scam and Non-scam calls) for all 4 models

Naive Bayes proved to be the best performing mode, covering the highest accuracy (93.9%) and F1-scores for both scam (94.6%) and non-scam (92.9%) calls. For the rest of the models, Extra Trees and SVC performed reasonably well, but Logistic Regression achieved the least accuracy. In the end, regardless of the Roman Urdu data, Naive Bayes remains the best, most accurate, and most efficient for identifying scam and non-scam calls.

6. Future Work

Although our current system demonstrates that a lightweight multilingual scam detector is possible, there remains considerable potential for development. One notable opportunity is branching out to underrepresented languages in our dataset, such as Punjabi, Balochi, and Siraiki. Because scammers often switch to these languages to increase rapport, building detection models for these languages will significantly increase inclusiveness and support for the most at-risk populations.

We also aim to build a basic web application that allows users to paste or upload call transcripts to check in real time whether a call is a scam or not. Such a tool may provide a novel means for users and the public, in general, to access readily available resources while allowing us to gather data to refine the model.

In the far future, our aim is complete integration of the system with telecom providers. Envision a world in which every phone call comes with the inclusion of scam detection: unverifiable activity could easily be flagged even before a user answers the call. Research of such nature in collaboration with telecom providers could shift from the realm of academia to the real world, protecting millions of users in their daily interactions.

References

[1]. Ayaz, M., Nizamani, S., & Chandio, A. Detection of Roman Urdu fraud or spam SMS in Pakistan using machine learning. *International Journal of Computing and Digital Systems*, *15*(1), 1053-1061, (2024).

[2]. Geurts, P., Ernst, D., & Wehenkel, L. Extremely Randomized Trees. Machine Learning, 63(1), 3–42, (2006).

[3]. Cortes, C., & Vapnik, V. Support Vector Networks. Machine Learning, 20(3), 273–297, (1995).

[4]. Rennie, J. D. M., Shih, L., Teevan, J., & Karger, D. R. Tackling the Poor Assumptions of Naïve Bayes Text Classifiers. Proceedings of ICML 2003, 616–623, (2003).

[5]. Lazer, R., Pucci, D., & Thompson, E. Optimizing Fraud Detection Models with Synthetic Data: Advancements and Challenges. International Journal of Artificial Intelligence and Data Science, 4(2), 55–70, (2025).

[6]. Almeida, T. A., Hidalgo, J. M. G., & Yamakami, A. Contributions to the Study of SMS Spam Filtering: New Collection and Results. Proceedings of the 11th ACM Symposium on Applied Computing (SAC 2011), 1310–1317, (2011).