Artifact Abstract: Making Acoustic Side-Channel Attacks on Noisy Keyboards Viable with LLM-Assisted Spectrograms' "Typo" Correction

Seyyed Ali Ayati[1], Jin Hyun Park[1], Yichen Cai[2], and Marcus Botacin[1]

[1]Texas A&M University
[2]University of Toronto
{ali.a, jinhyun.park, botacin}@tamu.edu, yichen@cs.toronto.edu

May 2025

Abstract

This artifact accompanies the paper *Making Acoustic Side-Channel Attacks on Noisy Keyboards Viable with LLM-Assisted Spectrograms' "Typo" Correction*. The paper introduces a transformer-based pipeline that combines Vision Transformers (VTs) for spectrogram-based keystroke classification and Large Language Models (LLMs) for context-aware correction of misclassifications under noisy conditions.

The artifact contains:

- Preprocessed Phone and Zoom keystroke audio datasets augmented with varying levels of Gaussian noise.
- Implementations of Vision Transformer models (e.g., Swin, ViT) and a baseline CoAtNet model for keystroke classification.
- Large Language Models, including fine-tuned Llama-3.2-3B and GPT-4o, used for correcting noisy predictions via few-shot prompting and Low-Rank Adaptation (LoRA).
- Scripts and instructions for reproducing the full evaluation pipeline, including audio preprocessing, spectrogram generation, classification, noise simulation, and LLM-based correction.

Artifact Evaluation Expectations and Badges

We submit this artifact for the following badges:

- **Artifact Available:** The complete source code, pretrained models, and datasets are publicly accessible. [1]
- **Artifact Functional:** All components of the pipeline are fully executable. Users can preprocess audio data, classify keystrokes, simulate noise, and apply LLM-based correction using our scripts and documentation.
  - **Verifying Functional Requirements (CoAtNet models):** To verify the functionality of our CoAtNet models, related to "Table 4" in the paper, navigate to `ours/phone/phone.ipynb`. This notebook loads the Phone dataset, runs the CoAtNet model, and saves the results in `ours/phone/confusion_matrix` and `ours/phone/predictions-phone.csv`. Please note that the cells under the "training" section should not be executed unless you intend to train the model from scratch. For repeating this with the Zoom dataset, follow the same procedure using `ours/zoom/zoom.ipynb`.

[1] Dataset and code at `https://doi.org/10.5281/zenodo.15634005`, fine-tuned model weights at `https://doi.org/10.57967/hf/5152` and `https://doi.org/10.57967/hf/5151`.

  - **Verifying Functional Requirements (CoAtNet models on noisy dataset):** To demonstrate the performance drop of CoAtNet under noisy conditions, related to “Table 6 first column”, use `ours/phone/phone_llm.ipynb`. Ensure that `GENERATE_FT_DATASET` is set to `False`. This notebook loads the Phone dataset, adds noise with predefined noise factors (as in Table 2), and saves the outputs in a CSV file prefixed with “noise_” followed by the noise factor. Repeat this process for all noise factors. For the Zoom dataset, use `ours/zoom/zoom_llm.ipynb` and follow the same procedure.
  - **Verifying Functional Requirements (LLMs):** To illustrate the performance improvement of CoAtNet on noisy datasets when augmented with LLMs, related to “Table 6 llm columns”, execute `ours/phone/llm_auto.py`. Provide an LLM model ID (e.g., “meta-llama/Llama-3.2-1B-Instruct”) and your Hugging Face token to download the model. Repeat the same procedure on `ours/zoom/llm_auto.py` for the Zoom dataset.
  - **Verifying Functional Requirements (Fine-tuning LLM):** To generate datasets for fine-tuning LLMs, repeat the steps described in “Artifact functional CoAtNet models but on noisy dataset”, but set `GENERATE_FT_DATASET` to `True`. Subsequently, follow the instructions in the `README` file within the `ours/finetune` directory.

- **Results Reproduced:**
  - Our CoAtNet and Vision Transformer models achieve classification accuracies of up to 100% (Phone) and 98.9% (Zoom) under clean conditions.
  - Under high noise, baseline BLEU scores drop to 0.021 (Phone) and 0.018 (Zoom), but rise to 0.623 and 0.489, respectively, after correction with fine-tuned LLMs.
  - BLEU, METEOR, and ROUGE-L metrics match those reported in the paper when the pipeline is re-executed.
  - To reproduce the results in the paper, please refer to the following:
    - **Table 4 (CoAtNet and Vision Transformer Accuracy):** The classification accuracies for clean conditions are generated by running `ours/phone/phone.ipynb` and `ours/zoom/zoom.ipynb` respectively. The outputs relevant to Table 4 can be found in the generated confusion matrices and prediction files.
    - **Table 6 (Baseline and LLM-corrected BLEU scores):** The baseline BLEU scores under high noise and the improved scores after LLM correction are obtained by running the experiments detailed in the ”Verifying Functional Requirements (CoAtNet models on noisy dataset)” and ”Verifying Functional Requirements (LLMs)” sections above. The relevant metrics will be outputted to the console or saved in the specified output files.
    - **BLEU, METEOR, and ROUGE-L metrics:** These metrics, which are presented in the paper, can be reproduced by executing the full evaluation pipeline described in the artifact’s documentation. The relevant scripts and notebooks for this are located within the `ours/` directory and specifically within `results` sub-folder for LLM-based correction, which calculate these metrics.

These reproducibility guarantees are enabled by the public availability of our trained models and data, the automation of the evaluation pipeline, and alignment of output metrics with those presented in the publication.