aveni

FinLLM Case Study

Aveni Detect Vulnerability Detection

aveni

Table of contents

Introduction

Overview

The primary method of communication between financial advisers and their clients is through telephone or video calls. These meetings, usually lasting around 60 minutes, produce lengthy call transcripts. During client calls or online interactions indicators of vulnerability such as references to ill health, unemployment, or financial hardship may emerge. Early identification of these situations is critical for establishing a meaningful client-advisor relationship and enables timely and appropriate guidance or interventions. In AI assisted chatbot systems, this can function as a form of automated triage to detect and prioritise vulnerable clients for escalation to human customer representatives.

Accurate extraction of this information from transcripts is a key feature of Aveni Detect, but currently relies on custom-built classifiers and general-purpose GPT models provided by OpenAI. Our goal is to implement a version of FinLLM fine-tuned for detecting the presence and location of vulnerabilities mentioned in long call transcripts, and aligned to the UK financial sector.

FinLLM-7B-Instruct-IE-Q rivals much larger models and outperforms those of a similar size, thanks to an improved SFT data mix that includes synthetic use-case data, as well as data for finance-specific instruction following, information extraction, and tabular QA. It simplifies vulnerability detection by processing entire call transcripts in a single step and surpasses Aveni's existing vulnerability detection model. Its smaller size lowers costs compared to external API-based models, and has enabled more LLM-powered processes within the Aveni Detect pipeline. This in turn reduces adviser risk & compliance review time by 30–50% while improving compliance monitoring, decision-making, and adviser risk management.

Existing Set-up -AveniDetect Vulnerability Detection

The vulnerability detection pipeline in Aveni Detect uses both a RoBERTa-based detection model and external API-based models provided to identify features in an adviser-client call transcript. This involves splitting the transcript into smaller utterance windows before performing the vulnerability detection and classification on each window.

Key Challenges

Some of the key challenges are:

Challenge	Description
Length	The length of the transcripts (around an hour each, consisting of approximately 40,000 tokens) requires the FinLLM model and deployment to be able to handle long-context inputs.
Vulnerability Definition	The definition of what should be classified as a vulnerability depends on context and is subjective, with large variance between tenants and across users between tenants.
Instruction Following	Due to the variance and subjective nature of the task, the instructions provided to the model need to be sufficiently detailed, providing descriptions and examples of each type of vulnerability. This requires FinLLM to understand long and detailed instructions.
Deployment	Some tenants frequently perform batch uploads where they upload hundreds of calls in a short period of time. Any deployment of FinLLM must be able to handle this load.

FinLLM Workflow

The goal of this use-case is to provide a fine-tuned FinLLM model which will act as a drop-in replacement for the RoBERTa vulnerability detection model. In addition, the fine-tuned FinLLM model will:

- Ingest the entire call transcript, rather than splitting it into windows of utterances as we believe this will provide the model with a wider context which would improve performance on vulnerability detection.
- Replace RoBERTa vulnerability detection classifier.
- Output both:
  a. If the call contains a vulnerability or not
  b. The location of the vulnerability within the transcript, if any was detected

Supervised Fine-Tuning

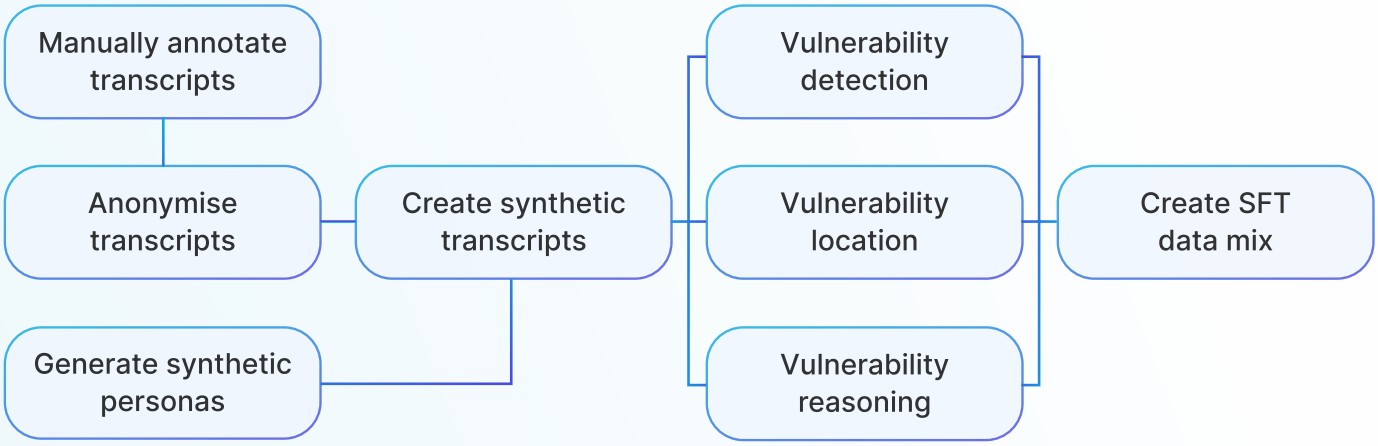
For improving FinLLM on this task, we used supervised fine-tuning with expanded data mixes (compared to FinLLM-7B-Instruct-Tab-Q) that reflect the tasks required by the use-case, specifically:

- Increasing the amount of **finance data** by taking existing, publicly available instruction following datasets and keeping any that are tagged as being in the "finance domain" by our proprietary classifier.
- Increasing the amount of **tabular question-answering datasets** using publicly available datasets.
- Adding publicly available datasets which involve **information extraction** from multi-turn conversations, i.e. multi-turn conversation input to structured (JSON) output.
- Adding **synthetic task-aligned information extraction** from multi-turn conversation datasets that have been generated using seed data from real calls which have been manually annotated and anonymised in-house.

Synthetic Data Generation

We relied on synthetic data generation to create more data for this use-case, using the following pipeline:

Synthetic Data Generation Pipeline



Evaluation Approach

We evaluate the use-case specific version of FinLLM using two datasets:

- **Anonymised Dataset:** Consists of approximately 100 calls across two tenants that have been manually anonymised and annotated. All of these contain a vulnerability. **Human-in-the-Loop (HITL) Dataset**: Consists of approximately 500 calls that have been manually annotated for the existence of vulnerability. These contain a mixture of calls containing a vulnerability and those that do not. No data from these calls was used in the creation of synthetic data for this use-case.

For both datasets, we evaluate **hit rate**. This is the percentage of time that the model's predicted location of vulnerability within a transcript is within five utterances of the manually labelled location of vulnerability. We use a +/- 5 utterance span as this reflects the 11 utterance window used by the current RoBERTa-based models and also captures the fact that conversations concerning the client's vulnerability status may span multiple utterances.

For the anonymised dataset, we evaluate **recall** (true positive rate). As all examples in this dataset only contain calls which contain vulnerabilities, the model should ideally predict that each one contains a vulnerability.

For the HITL dataset, as some calls do not contain a vulnerability, we measure the commonly used **precision, recall, and f1-score** metrics.

Performance & Gains

Anonymised Dataset Results

Table 1: Anonymised Dataset Results

Model	Hit Rate	Recall
gpt-4o	0.35	0.91
gpt-4o-mini	0.46	1.0
Qwen2.5 7B Instruct	0.27	0.68
FinLLM-7B-Instruct-Tab-Q	0.18	0.90
FinLLM-7B-Instruct-IE-Q	0.40	0.95

Table 1 shows our FinLLM-7B-Instruct-IE-Q model outperforms OpenAI's GPT gpt-4o despite having significantly fewer parameters. FinLLM-7B-Instruct-IE-Q significantly outperforms Qwen2.5 7B Instruct and FinLLM-7B-Instruct-Tab-Q, both of which have the same number of parameters as it, and is only outperformed by gpt-4o-mini. FinLLM-7B-Instruct-IE-Q also outperforms Magistral, gpt-4.1, gpt-5, and gpt-5-mini (results in Appendix) all of which have significantly more parameters than it.

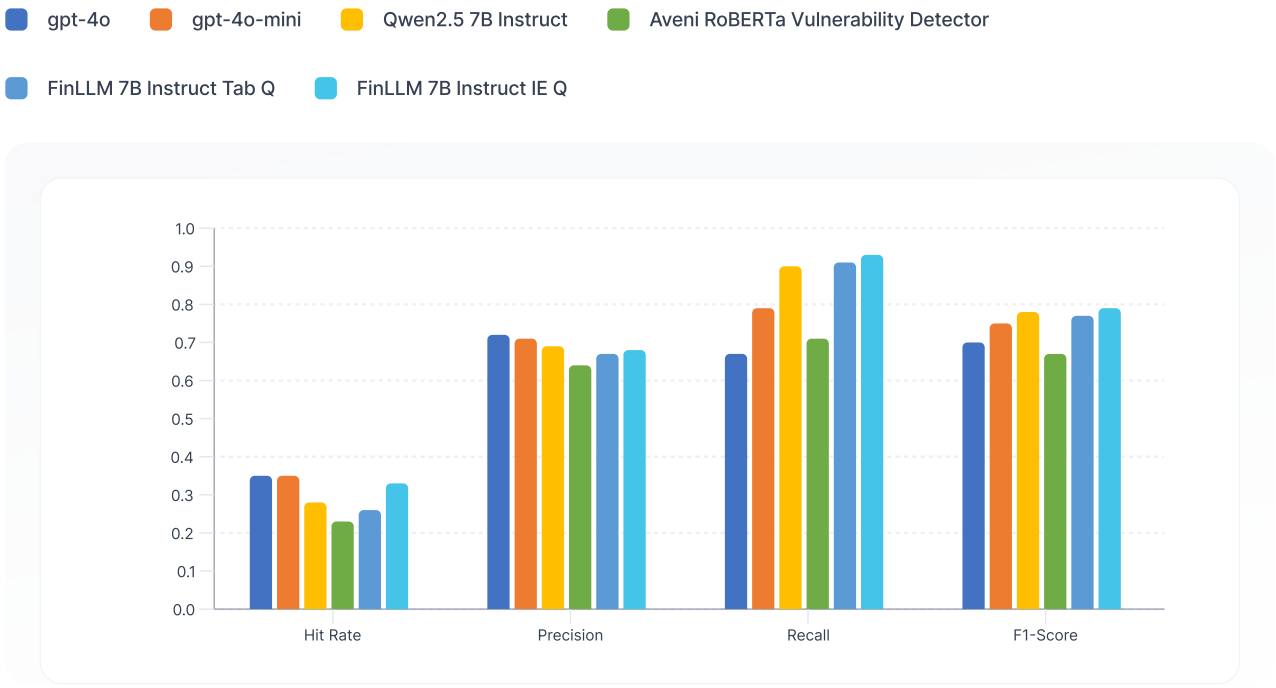
Human in the Loop (HITL) Dataset Results

Table 2: HITL Dataset Results

Model	Hit Rate	Precision	Recall	F1-Score
gpt-4o	0.35	0.72	0.67	0.70
gpt-4o-mini	0.35	0.71	0.79	0.79
Qwen2.5 7B Instruct	0.28	0.69	0.90	0.78
Aveni RoBERTa Vulnerability Detector	0.23	0.64	0.71	0.67
FinLLM-7B-Instruct-Tab-Q	0.26	0.67	0.91	0.77
FinLLM-7B-Instruct-IE-Q	0.33	0.68	0.93	0.79

The results on the HITL dataset, shown in **Table 2,** shows that FinLLM-7B-Instruct-IE-Q has a competitive hit rate with OpenAI's GPT models (which have considerably more parameters), and significantly outperforms Qwen2.5 7B Instruct, Aveni's RoBERTa based vulnerability detector, and FinLLM-7B-Instruct-Tab-Q. In terms of vulnerability classification, FinLLM-7B-Instruct-IE-Q obtains the highest F1-score out of all models. See Appendix for full results.

Vulnerability Detection - HITL Manual Annotation Results



Key Takeaways

We set out to demonstrate how through synthetic data creation and fine-tuning, FinLLM can compete and outperform leading external API-based LLMs on a core financial task. We validated this through real-world application in Aveni Vulnerability Detection, resulting in improved compliance adherence and monitoring, better decision making and reduced risk for advisers.

1. FinLLM-7B-Instruct-IE-Q is competitive with significantly larger models and significantly outperforms similar sized models. This is due to an improved training mix which contains additional use-case specific synthetic data, information extraction data, finance-specific instruction following data, and tabular question-answering data.

2. We simplified the process of detecting vulnerabilities within a call as this can now be done in a single step by passing the entire transcript to FinLLM.

3. We've increased the number of processes in the Aveni Detect pipeline powered by LLMs, proving our commitment to state-of-the-art research by shifting away from traditional classifier-based methods.

Appendix

Anonymised Dataset Results

Table A1: Anonymised Dataset Results (Full model list)

Model	Hit Rate	Recall
gpt-4o	0.35	0.91
gpt-4o-mini	0.46	1.0
gpt-4.1	0.31	0.79
gpt-4.1-mini	0.40	0.95
gpt-5	0.39	0.91
gpt-5-mini	0.36	0.93
Mistral Small 3.1 24B Instruct	0.43	0.99
Magistral Small 2507	0.39	0.87
Qwen2.5 7B Instruct	0.27	0.68
FinLLM-7B-Instruct-Tab-Q	0.18	0.90
FinLLM-7B-Instruct-IE-Q	0.40	0.95

HITL Dataset Results

Table A2: HITL Dataset Results (Full model list)

Model	Hit Rate	Precision	Recall	F1-Score
gpt-4o	0.35	0.72	0.67	0.70
gpt-4o-mini	0.35	0.71	0.79	0.75
gpt-4.1	0.34	0.74	0.64	0.69
gpt-4.1-mini	0.37	0.73	0.68	0.70
gpt-5	0.33	0.73	0.67	0.70
gpt-5-mini	0.37	0.73	0.62	0.67
Qwen2.5 7B Instruct	0.28	0.69	0.90	0.78
Aveni RoBERTa Vulnerability Detector	0.23	0.64	0.71	0.67
FinLLM-7B-Instruct-Tab-Q	0.26	0.67	0.91	0.77
FinLLM-7B-Instruct-IE-Q	0.33	0.68	0.93	0.79