Navigating Enterprise Constraints: Building a Hybrid Multi-Modal Mobile Intelligent App

Ketan Malempati, Akshat Gupta, Kaikai Liu
Computer Engineering Department
San José State University (SJSU)
San José, CA, USA
Email: {ketan.malempati, akshat.gupta, kaikai.liu}@sjsu.edu

2024 IEEE Intelligent Mobile Computing (MobileCloud) | 979-8-3315-3985-6/24/$31.00 ©2024 IEEE | DOI: 10.1109/MobileCloud62079.2024.00008

***Abstract*—The promise of artificial intelligence (AI) has prompted various industries to explore how it can enhance their businesses and improve customer experiences. However, the practical deployment of AI services faces challenges due to internal infrastructure limitations, data policies, and network restrictions. In this paper, we present our current project—a solution designed to empower mental health centers with AI tools while adhering to stringent data and network constraints. Our approach involves deploying AI services and data within internal enterprise networks, while the customer-facing mobile app resides in the public cloud. We introduce a custom AI agent system that monitors requests from the public cloud, enforces data access policies, schedules GPU computing resources, and performs model inference. All computations and raw data storage remain confined within the enterprise network. Additionally, we develop a comprehensive multi-modal AI services app, encompassing semantic search, data indexing, document and summarization, question-answering, and translation services for multilingual users. Administrative users have the ability to correct AI-generated data and contribute to continuous model refinement. Our solution serves as a blueprint for enterprises facing similar restrictions.**

***Index Terms*—AI, mobile AI services, Healthcare, private deployment, hybrid cloud**

I. Introduction

The integration of artificial intelligence (AI) into business processes has become a strategic imperative for organizations across various sectors [1]–[4]. As organizations seek different AI solutions to support their customers, mental health centers stand out as beneficiaries of AI-powered tools. These tools enhance patient care, streamline administrative tasks, and improve overall efficiency. For instance, an AI-powered semantic search engine can help healthcare practitioners organize internal confidential documents, records, videos, web pages, and links. It provides search results that align with immediate needs while adhering to access policy requirements. Additionally, AI-powered translation services facilitate efficient communication with multilingual customers.

However, deploying AI services within the constraints of an enterprise system and network presents unique challenges. Uploading internal confidential documents to public AI services, such as ChatGPT and AI translation services, is strictly prohibited to prevent data leakage. Storing internal data in public clouds that support AI services requires multiple rounds of evaluation and permission processes. Unfortunately, most institutions are not open to any particular cloud AI services.

Leveraging the concept of edge computing—bringing computing services to the data—offers a potential solution. Some enterprises choose to deploy AI models within their internal servers. However, publishing results to customers outside the enterprise network remains challenging. Opening a dedicated network port for customer access to internal servers poses cybersecurity risks and lacks the scalability and content delivery network (CDN) capabilities offered by public clouds [5].

The adoption of AI in mental health centers is essential for improving patient outcomes and operational efficiency. However, traditional deployment methods face limitations related to data privacy, network access, and scalability. Our proposed hybrid mobile cloud solution aims to address these challenges effectively.

In this paper, we propose hybrid approach that combines the strengths of both public and private cloud services. We develop a mobile app hosted on a scalable computing service platform provided by the public cloud infrastructure. The app serves as the customer-facing interface, allowing users to interact with AI-powered features. Our custom AI agent system monitors requests originating from the public cloud. Requests are stored in a real-time database within the cloud, eliminating the need for additional network ports to the enterprise network. The agent performs the following tasks: 1) Data Access Policy Check: Ensures compliance with data access policies. 2) Request Distribution: Routes requests to internal AI inference servers. 3) GPU Resource Scheduling: Allocates additional GPU computing resources as needed. 4) Model Inference: Executes specific AI models in response to user requests.

All computations and raw data storage remain confined within the enterprise network. Inference results pass through the data access policy and return to the public cloud database. Customers are notified of the available data results through the mobile app. Our solution will enable mental health centers to harness the power of AI while adhering to data security and network restrictions.

II. System Design

As shown in Fig. 1, our solution comprises two main components: 1) **Custom AI Agent System**: Manages AI

This work is in part funded by National Science Foundation under the grant number 2148353.

979-8-3315-3985-6/24/$31.00 ©2024 IEEE

DOI 10.1109/MobileCloud62079.2024.00008

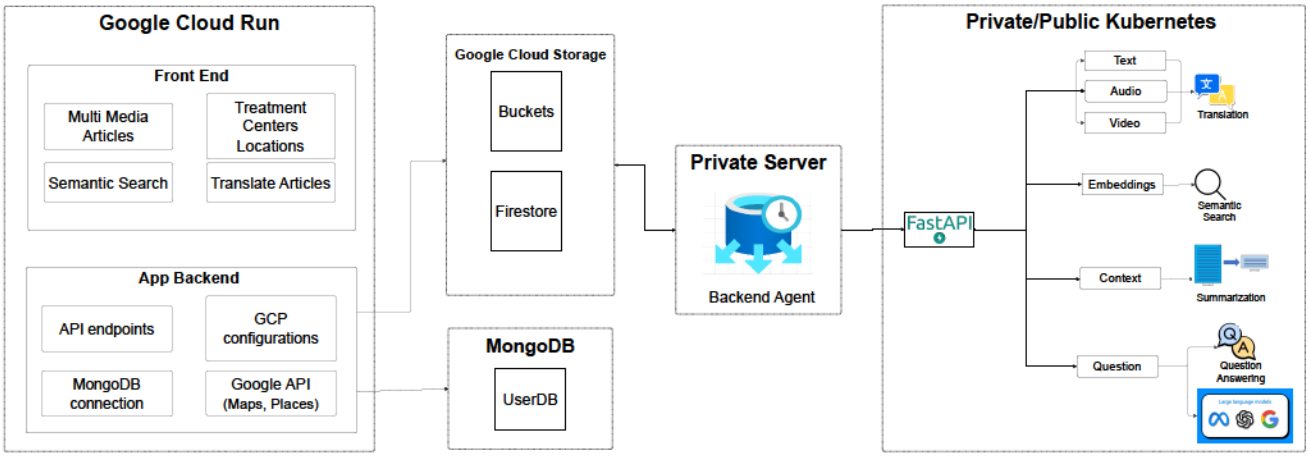

Fig. 1: System diagram.

services and data reside within dedicated internal AI machines. No additional network ports are opened for external communication. Raw data storage remains confined within the enterprise network. 2) **Customer-Facing Mobile App (Public Cloud)**: The frontend mobile app, accessible to users, is deployed in the public cloud. Users interact with the app to access AI-powered features.

A. Constraints and Challenges

Internal Infrastructure. Mental health centers often operate within tightly controlled internal networks. These networks prioritize security, limiting external access to sensitive data. Deploying AI services within such an environment requires careful consideration of infrastructure compatibility and resource allocation.

Data Policy. Compliance with data protection regulations is paramount in healthcare settings. Mental health centers must adhere to strict data access policies, ensuring patient confidentiality and privacy. Any AI solution must align with these policies while providing meaningful insights.

Network Restrictions. Network restrictions prevent direct communication between internal systems and external cloud services. Opening additional network ports for AI deployment is often not feasible due to security concerns.

III. Multi-Modal AI Services App

Our comprehensive mobile app offers the following features:

- **Semantic Search**: Enables users to search for relevant information within the mental health center's resources.
- **Data Upload and Indexing**: Allows users to upload relevant data securely. Indexes uploaded data for efficient retrieval.
- **Document and Video Summarization**: Summarizes lengthy documents and videos for quick review.
- **Question-Answering**: Provides answers to user queries based on available data.
- **Translation Services**: Supports multilingual users by offering translation capabilities.

Fig. 2 illustrates the components of our AI services. The following subsections will provide detailed descriptions of the major components.

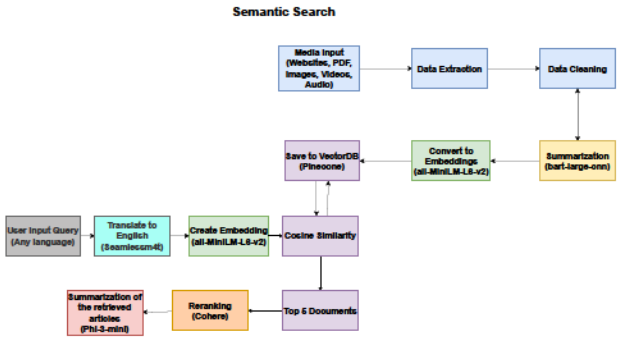

Fig. 2: Components for AI services.

A. Leveraging Multi-Media Data: Data Extraction and Question Generation

In our system, we initiate the process with data extraction, focusing on multi-media data types such as text files, PDFs, websites, videos, audio, and YouTube videos. Users have the flexibility to upload multiple files or multiple URLs simultaneously. Upon submission, the data extraction process commences, tailored to the specific media type:

1) **Text Extraction for Websites and PDFs:** For websites and PDFs, we extract relevant text content. This step ensures that textual information from web pages and documents is readily available for further analysis.
2) **Transcription and Translation for Audio and Video:** When dealing with audio and video files, we perform transcription. Additionally, we translate the transcribed content to English. This enables cross-lingual accessibility and ensures that the extracted information is universally understandable.
3) **Data Storage in Google Cloud Buckets:** All uploaded files and URLs are securely downloaded and stored in

Google Cloud storage buckets. This centralized storage ensures efficient data management and retrieval.

4) **Question Generation:** Once data extraction is complete, we move to question generation. We employ three distinct models for this task:
   - **potsawee-t5-large-generation-squad-QuestionAnswer:** Fine-tuned on the SQuAD dataset [6], this T5 model takes the context/passage as input and generates questions followed by answers.
   - **allenai-t5-small-squad2-question-generation:** Built based on the SQuAD 2.0 dataset, this model specializes in question generation.
   - **mrm8488-t5-base-finetuned-question-generation-ap:** Google's T5, fine-tuned on SQuAD v1.1, generates questions by prepending the answer to the context.

B. Question Answering on Uploaded Files: Leveraging Multiple Models

Next, we perform question answering on both sample questions and user-generated queries related to the files they upload. Our approach involves utilizing four distinct models:

- `deepset-roberta-base-squad2`: This model is based on RoBERTa-base and has been fine-tuned using the SQuAD2.0 dataset. It is trained on question-answer pairs, including unanswerable questions, specifically for the task of Question Answering.
- `distilbert-base-uncased-distilled-squad`: DistilBERT, a smaller and faster variant of BERT, is trained by distilling knowledge from BERT base. It retains over 95% of BERT's performance while having 40% fewer parameters [7]. The model is fine-tuned on SQuAD v1.1 using knowledge distillation.
- `google-flan-t5-large`: FLAN-T5 combines two components: a network and a language model. FLAN (Finetuned LAnguage Net) is paired with T5, a language model developed by Google in 2020. This model improves the effectiveness of zero-shot learning compared to the original T5 model.

C. Document Summarization and Semantic Search for Large Files

Given the substantial size of the content within files, we employ a summarization model to distill the essential information. Specifically, we summarize documents—whether in PDF format, from websites, or transcribed text—using the Facebook BART (Bidirectional and Auto-Regressive Transformers) model with a large CNN (Convolutional Neural Network) variant. BART, originally designed for text generation tasks, combines bidirectional encoders (similar to BERT [8]) with autoregressive decoders (akin to GPT). It achieves effectiveness in tasks such as summarization, translation, text classification, and question answering1.

After this summarization step, we integrate the processed data into our knowledge database. To facilitate efficient search through these files, we employ semantic search, which converts all textual content into embeddings. These embeddings, representing the semantic meaning of the text, allow us to group similar documents together. Specifically, we utilize the `krlvi-sentence-t5-base-nlpl-code-search-net` model, which converts text into 768 dimensional dense vector space embeddings. Along with metadata, these embeddings are stored in Pinecone, a powerful vector search engine.

When a user enters a search query, it is transformed into embeddings and sent to Pinecone. The system then retrieves the top 5 matching documents based on semantic similarity.

We have successfully implemented a chatbot using large language models (LLMs) such as llama2 and mistralAI. To enhance modularity and flexibility, we utilize a vector database called FIASS for storing all the documents. Additionally, we employ langchain to establish connections between FIASS and LLMs. This modular approach allows seamless swapping of vector databases and LLMs as needed.

The LLMs chatbot serves as a valuable resource for users seeking deeper insights into the documents they upload. Users can interact with the chatbot, ask questions, and receive informative answers related to the uploaded content.

D. Multimodal Translation Module: Enhancing Language Accessibility

Our system incorporates a versatile translation module that caters to various media types. Users have three options: text, audio, and video. Here's how it works:

1) **Audio and Video Transcription:**
   - When users select audio or video files, we employ the **Whisper** model for transcription.
   - The transcribed content is then converted to English, ensuring uniformity across different media formats.
2) **Language Translation:**
   - Users can further enhance accessibility by translating the text.
   - We utilize the **SeamlessMT4** model, a massive multilingual multimodal machine translation system that supports approximately 100 languages.
   - This seamless integration allows users to explore content in their preferred language, regardless of the original format.

By combining transcription and translation capabilities, our system promotes cross-lingual understanding and facilitates meaningful interactions with health-related content.

E. Admin User Contributions

Administrative users play a crucial role in maintaining data quality. Admins can correct any inaccuracies in AI-generated data. Ensures reliable information for users. Admins can add new labels to enhance model training. Contributes to continuous model improvement.

IV. Custom AI Agent System (Private Cloud)

We develop a custom AI agent system responsible for the following tasks:

- **Request Monitoring**: Monitors incoming requests from the public cloud. Ensures compliance with data access policies.
- **Resource Scheduling**: Dynamically allocates GPU computing resources based on demand. Optimizes resource utilization for model inference.
- **Model Inference**: Executes specific AI models in response to user requests. Provides real-time insights to users.

The entire system is encapsulated within a FastAPI framework [9], [10], which ensures rapid communication with underlying models and facilitates quick responses. Deployed within Docker containers, our system guarantees scalability, enabling seamless handling of large volumes of multimedia data.

There were two architectural approaches employed in this project. Initially, we followed the old architecture, where all modules and models resided together in a single location. However, we transitioned to a new architecture, wherein each module has its dedicated FastAPI module containing the associated models and code. This newer architecture significantly enhances communication efficiency, resulting in quicker responses.

The connection between the frontend and backend components relies on Google Firestore. The backend continuously listens to Firestore documents. When a new request is initiated, the backend agent processes it and invokes the corresponding FastAPI. Subsequently, the model processes the request and sends the response back to the agent. Finally, the agent updates the Firestore with the generated response. In scenarios where multiple requests target the same module, Docker replicates the module to handle concurrent processing.

V. Evaluation

A. Mobile App Functions

As shown in Fig. 3 and Fig. 4, the treatment center services page of our tool empowers users to discover health resources in their vicinity. Users can enter a specific location in the provided entry box or drop a pin directly on the map interface. Additionally, two dropdown menus allow users to filter health centers by "Facility Type" (e.g., Mental Health, Substance Use) and "Community Type" (e.g., Asian Americans). Search results can be displayed in either a list view or a map view as shown in Fig. 5. Our goal is to provide a user-friendly experience while respecting privacy and data security.

As shown in Fig. 6, the **Search Resources** page serves as a gateway to a wealth of health-related knowledge articles.

- **Tag-Based Filtering** Users have the flexibility to filter resources based on multiple tags, including Identifiers, Type, Category, Community, Topic, and Language. By selecting relevant tags, users narrow down their search to precisely match their needs.

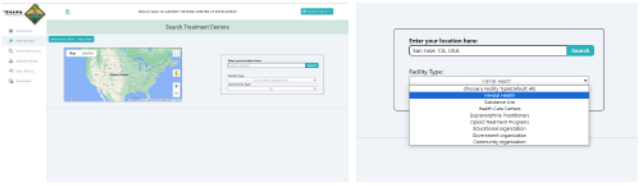

Fig. 3: Search treatment center.

Fig. 4: Search treatment center results.

- **Semantic Search Capability** Our system incorporates a powerful semantic search feature. Users can simply enter text into the search box, triggering a backend API call. The API ranks articles using a sophisticated semantic search model, ensuring accurate and context-aware results.

As shown in Fig. 7 Resource Details Page: When users click the "More Info" button on a resource card, they gain access to detailed information. The resource details page includes:

- **Tags Cloud:** Displaying applicable filters related to the resource. Users can suggest changes via the "Edit Tags" button, contributing to ongoing model improvement. Tag data is stored separately as a JSON schema, facilitating future model training.
- **Summary:** A concise text summary generated by an AI model.
- **Original Context:** Information scraped from the resource's original website, article, or PDF.
- **Original Resource:** An embedded iframe showcasing the original resource website.
- **Frequently Asked Questions:** Relevant questions posed by other users, along with AI-generated answers.
- **Question/Answer Interaction** Users can actively engage with the resource. The question/answer feature allows users to ask specific questions related to the article.

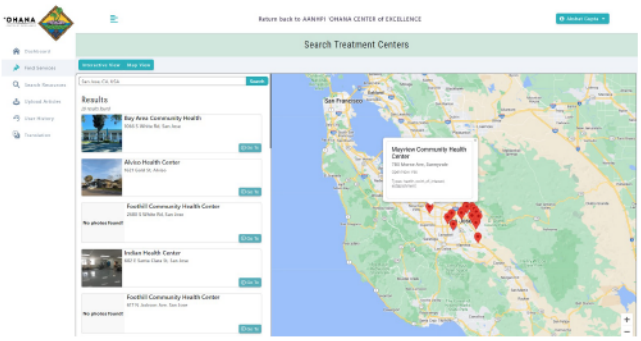

Fig. 5: Map view of all treatment center results.

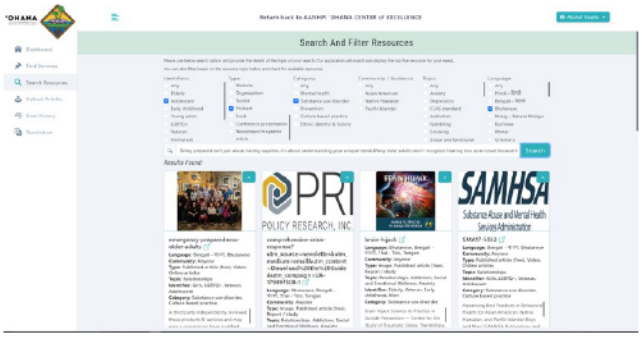

Fig. 6: Search resources list.

Fig. 7: The detailed view for one search result.

Backend models promptly provide answers, creating a dynamic and informative interaction.

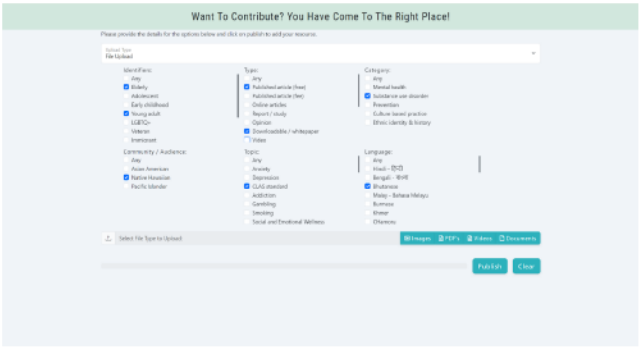

Fig. 8: The upload resources view.

As shown in Fig. 9, our web app introduces the Upload Resource feature, enabling users within the community to contribute health-related resources. Whether it's a website link, a PDF document, or any other media file, users can share valuable content that benefits others. The flexibility to submit websites, PDFs, or other media files ensures inclusivity and diverse content.

As soon as a resource is uploaded to our Google Firestore database, our system springs into action. Information extraction algorithms analyze the resource, capturing essential details. A concise summary is generated, distilling the resource's key points. The summarized resource is seamlessly integrated into our resources file. Users can access it via the Search Resources page. Whether they're seeking information on mental health, nutrition, fitness, or specific medical conditions, our community-contributed resources provide valuable insights.

Fig. 9: The upload resources view.

B. Performance Evaluation

We evaluate the performance of our implemented services. Figure 10 presents examples of question-and-answer tasks of three varying lengths: small, medium, and large. We also compared with FastAPI option with direct PyTorch inference approach in terms of speed. Figure 11 and Figure 12 shows the performance of video and audio transcription tasks across three different sizes.

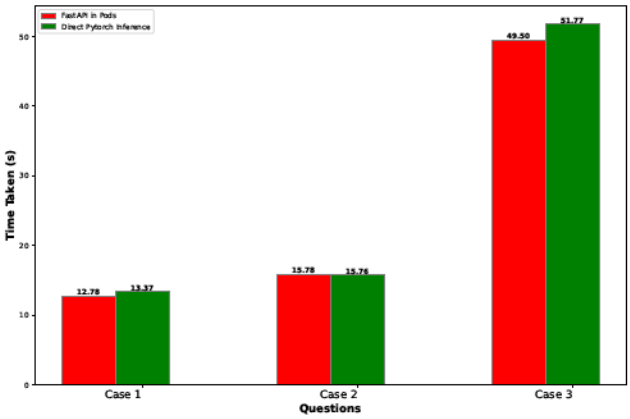

Fig. 10: Question and answering performance comparison.

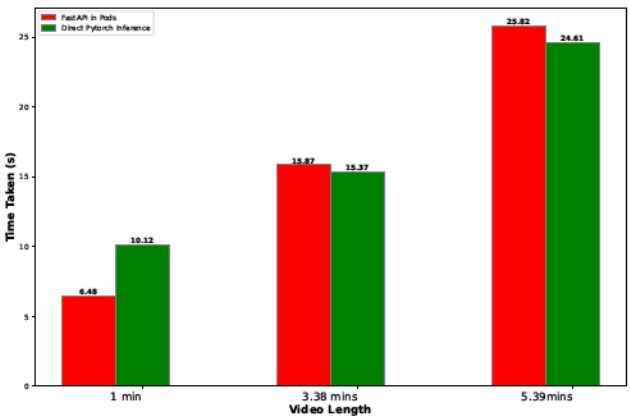

Fig. 11: Audio Transcription performance comparison.

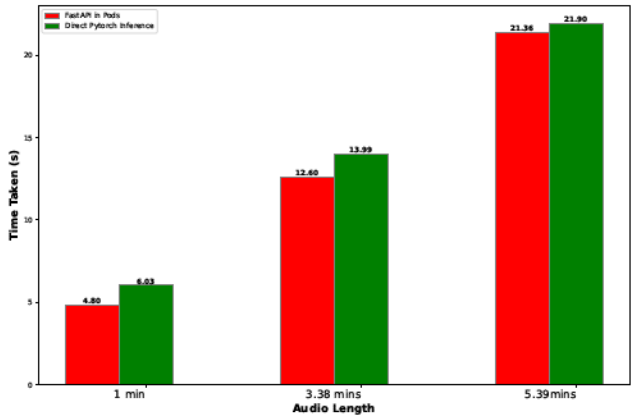

Fig. 12: Audio Transcription performance comparison.

Figures 13 and 14 display the performance evaluation of two translation tasks across three varying text lengths.

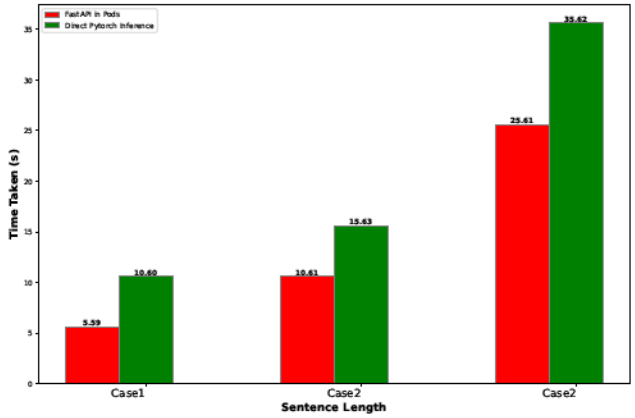

Fig. 13: Translation performance comparison.

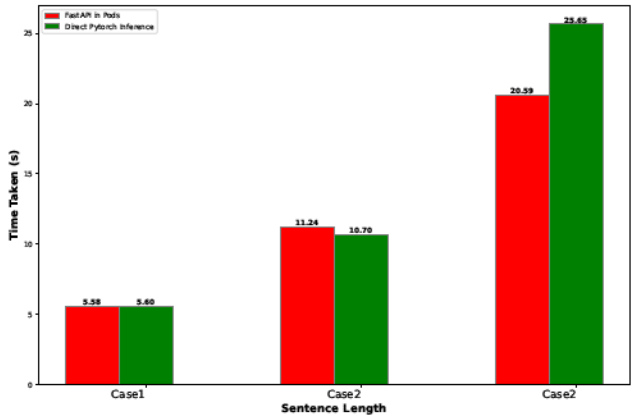

Fig. 14: Translation performance comparison.

C. Semantic search comparison

In the traditional search approach, full text search relies on exact word matching. While effective for finding names of people or places, it may struggle with long sentences. It splits the input into words and matches them individually. Semantic search represents a newer paradigm. It converts sentences into embeddings (dense vector representations) and performs similarity-based searches. By capturing semantic meaning, it provides more accurate results.

In our testing dataset, we have a total of 164 documents, which encompass various media types such as websites, PDFs, and videos. We evaluated five different vector search methods: Google Vertex AI, Scann, Faiss, and ChromaDB. Table. I shows the comparison of these vector search databases and their characteristics. For all these models, we utilized the embedding model **krlvi/sentence-t5-base-nlpl-code-search-net**, which provides 768-dimensional embeddings. Analyzing the results, we observed that both Faiss and ChromaDB performed exceptionally well, yielding identical outcomes. However, Pinecone also delivered promising results and surfaced some intriguing documents.

In addition to traditional metrics, such as precision and recall, there are other evaluation measures that provide deeper insights into search performance.

Mean Reciprocal Rank (MRR): MRR assesses the quality of ranked search results. It calculates the average reciprocal rank of the first relevant document retrieved by the search system. A higher MRR indicates better performance. The equation is

$$MRR = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{\text{rank}_i}$$

, where N is the total number of queries, rank_i represents the position of the first relevant document for query i.

Top-N Accuracy: This metric evaluates whether the correct answer appears within the top N search results. It's particularly useful for recommendation systems and question-answering tasks. For example, if $N = 5$, we check whether the correct answer is among the top 5 retrieved documents.

Precision at K: Precision at K measures the proportion of relevant documents among the top K retrieved results. It helps assess the precision of the search system within a limited result set. The formula is

$$\text{Precision@K} = \frac{\text{Number of relevant documents in top K}}{K}$$

Table. II shows the comparison of different vector search engine using various metrics.

VI. Conclusions

Practical AI deployment encounters hurdles related to internal infrastructure, data governance, and network limitations. In this paper, we present our innovative solution—a tailored framework designed to empower mental health centers with AI capabilities while adhering to stringent data and network constraints. Our approach involves a strategic separation of AI services and data: while the customer-facing mobile app resides in the public cloud, AI services and sensitive data remain within the secure confines of internal enterprise networks. Based on these, we have developed a robust and

Model	Time Taken	Cost	Setup
VertexAI	100-200ms	$15 per day (even if not used)	Manual work, difficult to automate using Python
Scann	100-150ms	Free	n4-standard-2 (CPUs: 2, Memory: 8GB)
Faiss	100-150ms	Free	n4-standard-2 (CPUs: 2, Memory: 8GB)
ChromaDB	100-150ms	Free	n4-standard-2 (CPUs: 2, Memory: 8GB)
Pinecone	100-150ms	Free (up to 2GB storage)	Easy to follow documentation
Redis	-	Free	n4-standard-2 (CPUs: 2, Memory: 8GB, $69.184071/month)

TABLE I: Comparison of different vector search database and their characteristics.

Model	MRR	Precision@3	Top 3 accuracy
Vertex AI	0	0	0
Scann	0.09	0	0
Faiss	0.4167	1	0.6
ChromaDB	0.4167	1	0.6
Pinecone	0.15	0.33	0.2

TABLE II: Comparison of different vector search engine using various metrics.

modular system for health knowledge discovery. Leveraging large language models (LLMs), multimodal data processing, and semantic search, our platform empowers users to explore health-related content efficiently. The seamless integration of summarization, translation, and question answering ensures that users can access valuable insights across various media formats. As we continue to enhance and refine our system, we remain committed to bridging the gap between health information and those who seek it.

REFERENCES

[1] D. So, W. Mańke, H. Liu, Z. Dai, N. Shazeer, and Q. V. Le, "Searching for efficient transformers for language modeling," *Advances in neural information processing systems*, vol. 34, pp. 6010–6022, 2021.

[2] P. Esmaeilzadeh, "Use of ai-based tools for healthcare purposes: a survey study from consumers' perspectives," *BMC medical informatics and decision making*, vol. 20, pp. 1–19, 2020.

[3] J. Morley, C. C. Machado, C. Burr, J. Cowls, I. Joshi, M. Taddeo, and L. Floridi, "The ethics of ai in health care: a mapping review," *Social Science & Medicine*, vol. 260, p. 113172, 2020.

[4] W. Tan, Y. Zhang, and K. Liu, "Intelligent emergency notification mobile service via multi-task bert models," in *2023 11th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud)*. IEEE, 2023, pp. 51–58.

[5] I. Bermudez, S. Traverso, M. Mellia, and M. Munafo, "Exploring the cloud from passive measurements: The amazon aws case," in *2013 Proceedings IEEE INFOCOM*. IEEE, 2013, pp. 230–234.

[6] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "Squad: 100,000+ questions for machine comprehension of text," *arXiv preprint arXiv:1606.05250*, 2016.

[7] R. Silva Barbon and A. T. Akabane, "Towards transfer learning techniques—bert, distilbert, bertimbau, and distilbertimbau for automatic text classification from different languages: a case study," *Sensors*, vol. 22, no. 21, p. 8184, 2022.

[8] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[9] O. Neumann, M. Schilling, M. Reischl, and R. Mikut, "Easymlserve: Easy deployment of rest machine learning services," in *PROCEEDINGS 32. WORKSHOP COMPUTATIONAL INTELLIGENCE*, vol. 1, 2022, p. 11.

[10] S. Horchidan, E. Kritharakis, V. Kalavri, and P. Carbone, "Evaluating model serving strategies over streaming data," in *Proceedings of the Sixth Workshop on Data Management for End-To-End Machine Learning*, 2022, pp. 1–5.