Agentic AI-Powered Field Sales Co-Pilot

# 1. Abstract

Field sales operations in domains such as FMCG and pharmaceuticals often suffer from inefficient route planning, generic product pitching, and lack of structured end-of-day reporting. This paper proposes an open-source, agentic AI-based system—a Field Sales Co-Pilot—that guides sales representative through daily operations using synthetic data, modular AI agents, and local language models. The system incorporates beat planning, product recommendation, personalized sales messaging, and automated daily summaries. We use a hybrid architecture combining simulation-driven data pipelines, a LangGraph-style planner, and open-source LLMs orchestrated using modular agents. The proposed framework aims to improve daily planning precision, personalization in sales interaction, and reporting effectiveness.

# 2. Introduction

Sales representatives face growing pressure to perform highly personalized, data-driven customer interactions across large territories with minimal technological support. Traditional CRM and ERP systems fail to adapt dynamically to on-ground realities such as traffic, localized product preference, or rep fatigue. AI-based solutions have emerged but are often monolithic, hard to customize, and dependent on enterprise data.

This project introduces a modular, agentic system—**Field Sales Co-Pilot**—that uses local language models and open-source tooling to assist field reps in day-to-day planning, personalized product pitches, and reflection-based summaries. To ensure feasibility in a data-restricted academic setting, we generate a realistic synthetic sales environment using simulation logic.

# 3. Literature Survey

The foundation for this project is built upon several key papers that advance the field of agentic AI and sales automation.  
  
1. **Neural Orchestration for Multi-Agent Systems (2025)** – Agrawal and Nargund introduced a neural orchestrator capable of intelligently selecting agents based on task needs and historical context. This concept is crucial for dynamically activating various sales agents in real-time.  
(<https://www.researchgate.net/publication/Neural_Orchestration_for_Multi-Agent_Systems>)  
  
2. **Orchestrating Agents and Data for Enterprise (2025)** – Kandogan et al. presented a Compound AI architecture that separates concerns into agent registries, data managers, planners, and orchestrators. This blueprint is reflected in our modular, layered system design.  
(<https://arxiv.org/abs/2404.02883>)  
  
3. **SalesRLAgent: Real-Time Sales Conversion Optimization (2025)** – Nandakishor et al. proposed a reinforcement learning-based agent that improves conversion during live sales interactions. This supports our pitch and feedback agent logic.  
(<https://arxiv.org/abs/2403.18383>)  
  
4. **Towards Personalized Conversational Sales Agents (2025)** – Kim et al. discussed profiling customers in dialogue to enable proactive recommendations. This aligns with the pitch and personalization aspects of our co-pilot.  
(<https://arxiv.org/abs/2403.13930>)  
  
5. **Multi-Agent Simulation of Retail Management Practices (2009)** – Siebers et al. demonstrated how agents can be used to simulate store operations and test retail policies. This informs our simulation logic for synthetic visits and sales behavior.  
(<https://link.springer.com/article/10.1007/s10257-009-0119-0>)  
  
6. **An Agent-Based Simulation of In-Store Customer Experiences (2007)** – This study explored modeling shopper behavior using agents, which supports our end-to-end simulation environment and sales flow.  
(<https://www.sciencedirect.com/science/article/abs/pii/S1568494606001555>)

# 4. Architecture

The architecture consists of a UI layer (Streamlit) for user interaction, a planning layer (LangGraph) for managing task flows, and modular agents. These agents include:

* Beat Planning Agent for optimized store visits.
* Recommendation Agent for personalized product suggestions.
* Pitch Generator Agent for context-aware sales messages.
* Summary Agent for generating daily logs.

Synthetic data is generated using rule-based simulation logic, providing a realistic foundation for training and evaluation without relying on sensitive business data.

# 5. Data Model

* **retailers**:

| **Column Name** | **Data Type** | **Description** |
| --- | --- | --- |
| Retailer\_ID | TEXT | Unique ID (Primary Key) |
| Name | TEXT | Retailer name |
| City | TEXT | City of operation (Bengaluru) |
| Channel | TEXT | Retailer channel type |
| Latitude | REAL | Geo-coordinate (lat) |
| Longitude | REAL | Geo-coordinate (long) |

Locality info(prime location)

Sales rep assigned

* **products:**

| **Column Name** | **Data Type** | **Description** |
| --- | --- | --- |
| **Product\_ID** | **TEXT** | **Unique product ID (Primary Key)** |
| **Product\_Name** | **TEXT** | **Name of the product** |
| **Category** | **TEXT** | **Product category** |
| **Price** | **REAL** | **Unit price** |
| **Pack\_Size** | **TEXT** | **Packaging size** |

* **visits**:

| **Column Name** | **Data Type** | **Description** |
| --- | --- | --- |
| Visit\_ID | TEXT | Unique visit ID (Primary Key) |
| Retailer\_ID | TEXT | Foreign key referencing retailers |
| Date | TEXT | Date of the visit |
| Products\_Suggested | TEXT | List of suggested product IDs |
| Feedback | TEXT | Feedback or notes from visit |
| Order\_Placed | INTEGER | 1 if order was placed, else 0 |

Sales rep id

* **sales**:

| **Column Name** | **Data Type** | **Description** |
| --- | --- | --- |
| Invoice\_ID | TEXT | Unique invoice ID |
| Visit\_ID | TEXT | Foreign key referencing visits |
| Retailer\_ID | TEXT | Foreign key referencing retailers |
| Product\_ID | TEXT | Foreign key referencing products |
| Quantity | INTEGER | Units sold |
| Date | TEXT | Date of sale |
| Total\_Amount | REAL | Total sale amount (Quantity × Price) |

Sales rep master

# 6. Implementation Plan

* Generate synthetic sales, retailer, product, and visit data using simulation rules.
* Develop and evaluate product recommendation models using Classification models/Recommender engine
* Implement Beat Planning using clustering and geolocation data.
* Implement large language models to create sales pitch and summary agents.
* Orchestrate all agents using LangGraph and build the frontend with Streamlit with a chatbot feature for Sales Officers.
* Evaluate using metrics like Precision, Recall, and qualitative feedback on pitch summaries.