

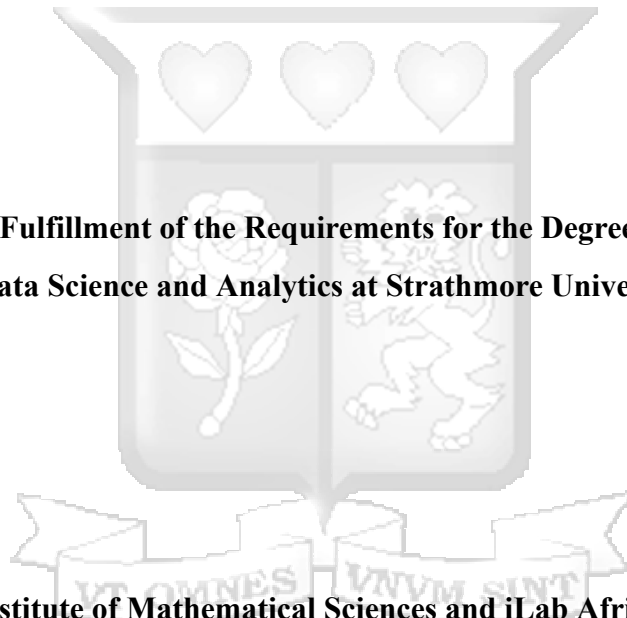
Harnessing Generative AI for Precision Livestock Management: Methane Emission Reduction Strategies in Cattle Farming

By

Ogada Joseph Ridge

166895

**Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science
in Data Science and Analytics at Strathmore University**



Institute of Mathematical Sciences and iLab Africa

Strathmore University

Nairobi, Kenya

June, 2025

This dissertation is available for library use on the understanding that it is copyright material and that no quotation from the dissertation may be published without proper acknowledgement.

Declaration and Approval

Declaration

I declare that this work has not been previously submitted and approved for the award of a degree by this or any other University. To the best of my knowledge and belief, the dissertation contains no material previously published or written by another person except where due reference is made in the dissertation itself.

©No part of this dissertation may be reproduced without the permission of the author and Strathmore University.

Student's Name: **Ogada Joseph Ridge**

Sign:  Date: **27th May 2025**

Approval

The dissertation of **Ogada Joseph Ridge** was reviewed and approved by the following:

Dr. John Olukuru

Lecturer, Institute of Mathematical Sciences,
Strathmore University

Dr. Kennedy Senagi,

Lecturer, Institute of Mathematical Sciences,
Strathmore University.

Dr. Godfrey Madigu,

Dean, Institute of Mathematical Sciences,
Strathmore University

Prof. Bernard Shibwabo,

Director of Graduate Studies,
Strathmore University

Abstract

Livestock farming is a significant contributor to global methane emissions, primarily through enteric fermentation and manure management, with disproportionate impacts on agricultural sustainability in developing regions like rural Kenya. This study addresses the critical gap between available mitigation strategies and their practical adoption by developing an AI advisory system combining evidence-based research with machine learning. The system implements a Retrieval-Augmented Generation (RAG) architecture utilizing three specialized components; HuggingFace's all-MiniLM-L6-v2 model for 384-dimensional semantic embeddings of agricultural literature, Meta's Llama3.2:1B model (deployed via Ollama) with constrained generation parameters (temperature=0, top_k=20) for reproducible outputs, and ChromaDB's cosine similarity indexing for context-aware retrieval. Document processing incorporates fuzzy deduplication (90% similarity threshold) and semantic chunking (1000-character units with 100-character overlap) to optimize knowledge representation. Performance evaluation on 20 representative queries demonstrated 85% citation accuracy through manual Digital Object Identifier (DOI) verification, with 75% of responses rated as technically sound in preliminary developer assessments. While current local deployment on consumer hardware (Intel i5, 16GB RAM) yields 20-30 second response times, the architecture supports horizontal scaling through cloud hosting or edge optimization for field deployment. This work contributes a modular framework for climate-smart livestock management, demonstrating how lightweight AI systems can bridge research and practice. Future integration pathways include Tier 3 emission methodologies and offline capabilities for connectivity-limited regions, advancing both climate targets and agricultural equity.

Keywords: Greenhouse Gases Emissions, Generative AI, Enteric Emissions, Machine Learning.

Table of Contents

Declaration and Approval	ii
Abstract	iii
List of Figures	viii
List of Tables	ix
List of Abbreviations	x
Acknowledgment	xi
Chapter 1: Introduction	1
1.1 Background	1
1.1.1 Dairy Farming’s Role in Global Emission	1
1.1.2 Precision Dairy Livestock Farming	2
1.1.3 The Role of Generative AI in Reducing Methane Emissions	3
1.2 Problem Statement	3
1.3 Research Aim	5
1.4 Research Objectives	5
1.5 Scope and Limitations of the Study	5
1.5.1 Scope	5
1.5.2 Limitations	5
1.6 Research Justification	6
Chapter 2: Literature Review	7
2.1 Theoretical Literature Review	7
2.1.1 Greenhouse gases overview	7
2.1.2 Precision Livestock Farming and Emission Management	7
2.1.3 Generative AI and Large Language Models (LLMs)	8
2.1.4 Generative AI in Agriculture	8
2.1.5 Challenges and opportunities in livestock management	8
2.1.6 Ethical Implications of AI in Carbon Footprint Management	9
2.1.7 Limitations of General-Purpose LLMs in Agricultural Advisory Systems	9
2.2 Empirical Literature Review	10
2.2.1 Methane Emissions Reduction through Precision Livestock Farming (PLF)	10
2.3 AI and Generative AI for Farm Management	10

2.3.1	Innovative Technologies in Livestock Monitoring	11
2.3.2	Impact of AI and Blockchain Integration in Emission Management	11
2.3.3	Machine Learning Models for Methane Emission Prediction	12
2.3.4	Retrieval-Augmented Generation in Livestock Emission Monitoring	12
2.3.5	RAG vs. Fine-Tuning: Empirical Insights from Agricultural AI	14
2.4	Research Gap	14
Chapter 3: Methodology		15
3.1	Research Design	15
3.2	Data Understanding	16
3.2.1	Data Collection from the CORE Repository	16
3.2.2	Sample Size Justification	16
3.2.3	Data Processing	16
3.3	Data Preparation	17
3.3.1	Document Loading	17
3.3.2	Text Splitting	17
3.3.3	Creating Embedding	17
3.3.4	Storage	18
3.4	Machine Learning Model	18
3.4.1	Transformer Architecture	18
3.5	Benchmarking	21
3.6	Machine Learning Model Evaluation and Optimization	22
3.6.1	Accuracy	22
3.6.2	BERTScore	23
3.6.3	ROUGE	24
3.7	Deployment	25
3.7.1	User Interface	25
3.7.2	Model	25
3.8	Conceptual Diagram	25
Chapter 4: System Design and Architecture		26
4.1	System Modeling	26
4.2	System Requirements	26
4.2.1	Functional Requirements	26

4.3	System Components	27
4.3.1	Data Ingestion and Processing	27
4.3.2	Vector Database (ChromaDB)	27
4.3.3	RAG Pipeline	27
4.3.4	Application Programming Interface (API) Endpoints (FastAPI)	28
4.3.5	Document Metadata Processing	28
4.4	System Workflow	29
Chapter 5:	System Implementation and Testing	32
5.1	Introduction	32
5.2	User Interface Implementation	32
5.2.1	Homepage	32
5.2.2	Chat Query	33
5.2.3	Response Presentation	34
5.2.4	Response Reference	35
5.3	Knowledge Base Construction	37
5.4	Testing Methodologies	38
5.4.1	Functional Validation	38
5.5	System Architecture and Hardware Considerations	39
Chapter 6:	Discussion of Results	40
6.1	Dataset Loading and Ingestion	40
6.2	Data Ingestion	40
6.2.1	Word Cloud for Livestock Emissions	40
6.2.2	Word Cloud for Livestock Manure Management	42
6.2.3	word Cloud for Livestock Emissions Mitigation	42
6.2.4	Word Cloud for Livestock Emissions Mitigation	43
6.2.5	Word Cloud For All Documents	44
6.3	Generating and Storing Embeddings	46
6.4	Generating Embeddings	46
6.5	Storing Embeddings	47
6.6	Document Querying and Retrieval	47
6.7	Knowledge Base Visualization	47
6.7.1	Document Clusters: UMAP Projection with HDBSCAN Clustering	47

6.7.2 UMAP Projection of Document Embeddings: Revealing the Semantic Structure of the Farm Advisor Corpus	48
6.8 Structured Knowledge Delivery Pipeline	51
6.9 Deployment	52
6.10 Comparison with Traditional Extension Services and Real-World Adoption Challenges	53
Chapter 7: Conclusions, Recommendations and Future Work	54
7.1 Conclusions	54
7.2 Recommendations	54
7.3 Future Work	55
References.	56
Appendices	65
Appendix A: Similarity Report	65
Appendix B: Ethical Clearance Confirmation	67
Appendix C: Code Implementation	68



List of Figures

3.1	CRISP-DM framework (Wikimedia Commons, 2020)	15
3.2	Transformer Model Architecture (Raiaan et al., 2024)	19
3.3	Lightweight Instruction Tuner Benchmarks (AI, 2025)	21
3.4	Conceptual Diagram	25
4.1	Custom Prompt Template	28
4.2	farmerVICE System Architecture	29
5.1	Homepage	33
5.2	Chat Section: Sending a prompt to the model	34
5.3	Chat Section Response	34
5.4	Chat voting	35
5.5	View references on modal	36
5.6	Download references on modal	37
5.7	Read references on modal	37
5.8	CONnecting REpositories (CORE) query strings in form of a dictionary	38
5.9	function to make the query from CORE repository	38
6.1	word Cloud for Livestock Emissions	41
6.2	word Cloud for Livestock Manure Management	42
6.3	word Cloud for Livestock Manure Management	43
6.4	word Cloud for Paris Agreement	44
6.5	word Cloud for all documents	45
6.6	Top 10 words in all documents	46
6.7	Document Clusters: UMAP Projection with HDBSCAN Clustering	48
6.8	UMAP Projection (32,229 chunks from Farm Advisor Corpus)	49
6.9	PCA Clustering of Document Embeddings (n_clusters=20)	50
6.10	UMAP Projection of Document Embeddings (Colored by PCA Clusters)	51
6.11	API response from fastAPI application	52

List of Tables

4.1 CORE API Query Parameters	30
-------------------------------------	----



List of Abbreviations

AI Artificial Intelligence

API Application Programming Interface

CORE COnnecting REpositories

CRISP-DM Cross-Industry Standard Process for Data Mining

DOI Digital Object Identifier

ETL Extract Transform Load

GHG Global greenhouse gas

JSON JavaScript Object Notation

LLM Large language model

LLMs Large language models

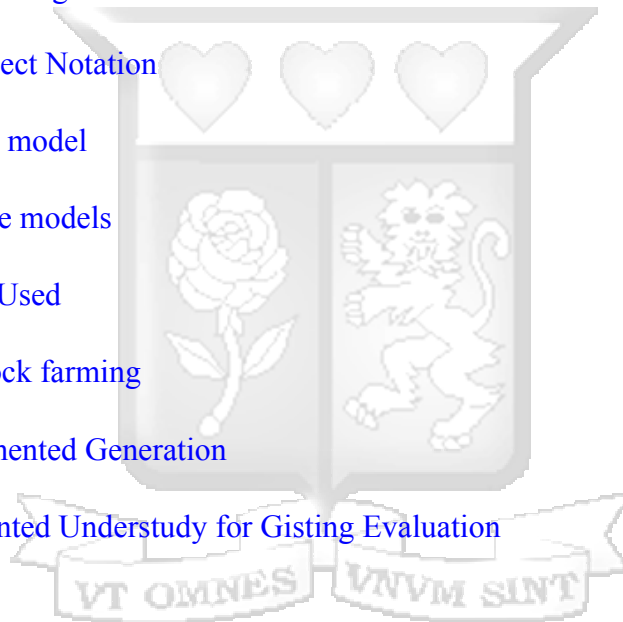
LRU Least Recently Used

PLF Precision livestock farming

RAG Retrieval-Augmented Generation

ROUGE Recall-Oriented Understudy for Gisting Evaluation

UI User Interface

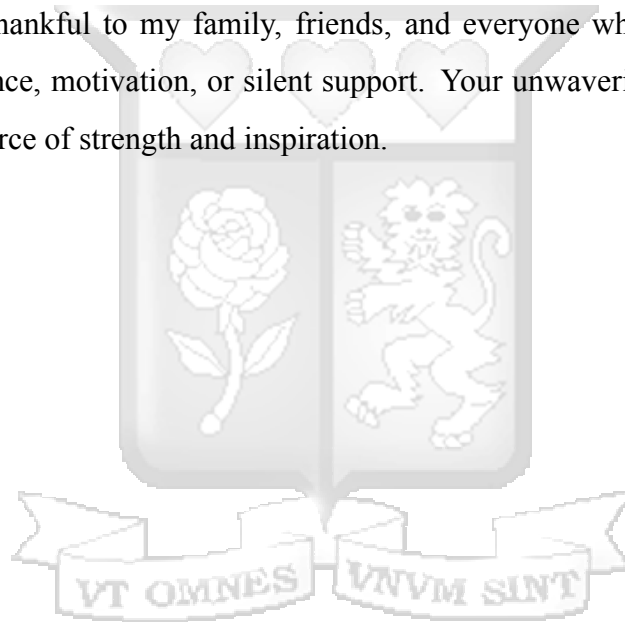


Acknowledgments

I extend my sincere gratitude to my dissertation supervisors, Dr. John Olukuru and Dr. Kennedy Senagi for their invaluable guidance, steadfast support, and thoughtful encouragement throughout this research journey. His deep expertise and insightful feedback have been instrumental in shaping the depth, direction, and overall quality of this dissertation.

I am also immensely appreciative of Strathmore University for fostering an intellectually stimulating environment and providing the essential resources that enabled the successful completion of this study. Additionally, I extend my thanks to [CORE](#) for granting access to the data used in this research.

Lastly, I am deeply thankful to my family, friends, and everyone who contributed whether through direct assistance, motivation, or silent support. Your unwavering encouragement has been a continuous source of strength and inspiration.



Chapter 1: Introduction

1.1 Background

1.1.1 Dairy Farming's Role in Global Emission

Livestock is a major contributor to Global greenhouse gas (GHG) emissions, primarily through methane and nitrous oxide, which together account for approximately 14.5% of total global GHG emissions, (Musa, 2020); (Hoque et al., 2022). GHG emissions from livestock farming are primarily through enteric fermentation and manure management (Neethirajan, 2024); (Tamilselvan and Tyagi, 2024). Methane (CH₄) is a by-product of enteric fermentation, a normal digestive process in ruminants, where stomach microorganisms break down food (Cholewińska et al., 2020); (Ungerfeld et al., 2023). Burping is the main way that CH₄ is released. The IPCC states that methane's global warming potential is roughly 28 times greater than CO₂ over a 100-year period (Liu et al., 2021); (Dobson et al., 2023); (Parker, 2021).

Methane emissions from livestock have been increasing fourfold over the past 130 years, rising from 31.8 Tg CH₄ in 1890 to 131.7 Tg CH₄ in 2019 (Zhang et al., 2022). This increase is primarily driven by non-dairy cattle, with significant growth in regions like South Asia, tropical Africa, and Brazil. Emissions have slowed in developed regions but continue to rise in developing areas, influenced by livestock population growth and changing diets (Zhang et al., 2022). The international trade of livestock products has amplified this issue, with emissions from their production reaching 92.0 MT in 2017 (Wang et al., 2022). Within Canada's dairy sector, enteric fermentation is a significant contributor to greenhouse gas emissions. Strategies like reducing fermentable substrates in the rumen by diet modifications and reducing methanogenesis with feed additives like lipids or nitrates have been proposed as ways to limit these emissions (Beauchemin et al., 2020); (Honan et al., 2021).

Manure management on dairy farms is a major contributor to greenhouse gas (GHG) emissions, particularly methane and nitrous oxide (N₂O) (Rivera and Chará, 2021). Methane is generated when manure is stored and handled under anaerobic conditions, whereas nitrous oxide, whose global warming potential is 265 times that of CO₂ is primarily released during manure storage and field application (Costa et al., 2021); (Rotz et al., 2021). The production of N₂O stems from soil nitrification and denitrification processes, which are exacerbated by the application of nitrogen-rich manure. Implementing improved manure management strategies, such as aerobic

composting and anaerobic digestion, can help reduce these emissions while also enabling the conversion of manure into biogas, a renewable energy source (Khoshnevisan et al., 2021).

To address these challenges, various mitigation strategies have been proposed, including improved manure management, enhanced diets to reduce enteric fermentation, more efficient fertilizer use (Musa, 2020); (Hoque et al., 2022), feed and waste additives also show potential for further reducing emissions (Peterson and Mitloehner, 2021). In addition the estimation of livestock emissions is based on the Tier methodologies outlined by the IPCC (2019) which aids in the estimation of livestock emissions using emission factors. Emerging technologies like Big Data and Artificial Intelligence offer promising solutions for optimizing feed efficiency, manure management, and energy utilization (Neethirajan, 2024). Current mitigation strategies exist, but there is a gap in providing a platform for education and a data-driven Decision Support System to guide livestock farmers in implementing these strategies effectively.

1.1.2 Precision Dairy Livestock Farming

Precision livestock farming (PLF) is an advanced livestock management approach that incorporates principles from process engineering (Wathes et al., 2008). It leverages innovations in animal science alongside information and communication technologies to enhance livestock production. A fundamental goal of PLF systems is the continuous and automated monitoring of animals, enabling farmers to optimize key aspects such as feeding strategies, growth rate management, and animal health (Berckmans, 2006). Ultimately, PLF aims to enhance farm profitability, efficiency, and sustainability (Banhazi et al., 2012). Within the PLF framework, precision livestock feeding ensures that animals receive the appropriate quantity and composition of feed at the right time. This approach can be applied at the group level (Parsons et al., 2007); (Cangar et al., 2008); (Niemi et al., 2010) or customized for individual animals within a herd (Pomar et al., 2009); (Andretta et al., 2014). Effective implementation of precision livestock feeding requires specialized technologies, including devices for monitoring feed intake and animal weight, computational tools to estimate real-time nutrient requirements, and advanced feeding systems that deliver precise feed compositions to align with specific production objectives.

1.1.3 The Role of Generative AI in Reducing Methane Emissions

Recent research highlights the potential of generative AI and AI-powered decision support systems in promoting sustainable agriculture. These technologies can enhance productivity, optimize resource usage, and provide data-driven insights for farmers (al. [Chhaya Narvekar, 2023](#)); ([Asolo et al., 2024](#)). AI applications in agriculture include resource optimization, crop yield prediction, disease identification and soil analysis (al. [Chhaya Narvekar, 2023](#)). The integration of AI in agriculture addresses challenges such as climate change, resource limitations, and increased food demand (al. [Chhaya Narvekar, 2023](#)); ([Ray, 2023](#)).

Recent research explores innovative approaches to reduce methane emissions in livestock farming using AI technologies. ([Parmar et al., 2024](#)) developed an AI-driven benchmarking tool for Canadian dairy farms, utilizing machine learning to identify emission hotspots and forecast trends. ([Nejad et al., 2024](#)) reviewed conventional and AI-powered methods for methane detection, emphasizing the need for region-specific formulas. ([Altshuler et al., 2023](#)) used AI to predict feed additive efficacy in dairy cows, demonstrating potential for a 27% reduction in emissions through targeted strategies.

Various mitigation techniques were discussed by ([Króliczewska et al., 2023](#)), including dietary modifications, genetic selection, and feed additives. While chemical interventions have shown promise, concerns about health and consumer acceptance persist. Genetic selection and biological feed additives emerge as promising approaches, though further research is needed to validate their effectiveness in reducing methane production by ruminants ([Króliczewska et al., 2023](#)).

1.2 Problem Statement

Livestock production is a significant source of global greenhouse gas emissions, contributing approximately 14.5% of total human-induced emissions ([Musa, 2020](#)); ([Bekuma et al., 2018](#)). Methane and nitrous oxide are the primary greenhouse gases emitted by livestock ([Musa, 2020](#)). Several mitigation strategies have been proposed to reduce these emissions. Dietary manipulation, improved production efficiency, and genetic management can reduce enteric methane emissions ([Musa, 2020](#)); ([Bekuma et al., 2018](#)). Manure management techniques, including frequent removal, acidification, and anaerobic digestion, effectively reduce emissions during

storage and treatment (Yan et al., 2024). For manure application, shallow injection can significantly reduce ammonia emissions, though it may increase nitrous oxide emissions (Yan et al., 2024). A combination of strategies—such as carbon pricing, feeding management, and breeding management—is seen as the most feasible and sustainable approach to mitigating livestock emissions (Kumari et al., 2019). However, the implementation of these strategies must take into account mitigation efficiency, technical feasibility, local regulations, and cost-effectiveness (Yan et al., 2024).

Capacity building for sustainable livestock management involves enhancing technical skills related to methane reduction practices, such as improved feed efficiency, manure management, and animal nutrition, which are essential for reducing livestock-based greenhouse gas emissions. In South Africa, enhancing the adaptive capacity of pasture-based livestock farmers is essential for coping with climate-related events and production constraints (Oduniyi et al., 2020). For women farmers in particular, who contribute significantly to livestock rearing, there is a need for targeted capacity building programs to improve their knowledge and skills in modern livestock management practices (Pandey et al., 2020). In Nepal, strategies for sustainable livestock production focus on improving animal nutrition, including enhancing feed digestibility, preserving seasonal forages, and developing novel protein sources for monogastric animals. These efforts require strong policies supporting capacity building, research infrastructure development, and collaboration between research and industry sectors (Khanal et al., 2022).

While various mitigation strategies have been proposed to reduce livestock emissions, their widespread implementation remains limited, especially in developing regions where capacity-building efforts have been insufficient. This study aims to address this gap by focusing on targeted capacity building, with particular emphasis on empowering women farmers, to reduce emissions through sustainable livestock management practices. By enhancing technical skills in areas such as feed efficiency and manure management, this research could contribute significantly to achieving climate goals and the United Nations Sustainable Development Goals (Bloomfield et al., 2018). Additionally, the findings could influence policy recommendations that support region-specific mitigation strategies and bolster resilience to climate change in vulnerable communities (Khanal et al., 2022).

1.3 Research Aim

This study intends to develop a machine-learning-based advisory system offering personalized, actionable recommendations for methane emissions reduction in dairy livestock farming.

1.4 Research Objectives

This research aimed to address the following objectives:

- (a) To structure agricultural text data (research papers, surveys) into a retrieval-optimized knowledge base using NLP techniques such as semantic chunking and embeddings.
- (b) To develop a **RAG** based generative AI advisory system that provides optimized feeding and manure practice recommendations to reduce methane emissions.
- (c) To validate the advisory system's recommendations through comparative benchmarking against alternative NLP architectures, assessing accuracy and practical relevance.

1.5 Scope and Limitations of the Study

1.5.1 Scope

This study focuses on addressing methane emissions from livestock farming in rural Kenya, emphasizing enteric fermentation and manure management as primary contributors. It integrates generative AI and local Large language models (**LLMs**) to develop a personalized, text-driven advisory system for smallholder farmers.

The study will include the following key components:

- (a) The use of local LLMs for generating tailored text-based recommendations, leveraging machine learning for emissions predictions and optimization strategies.
- (b) Incorporation of secondary data from surveys, field trials, and industry benchmarks, along with textual agricultural guides and localized content for LLM training.

1.5.2 Limitations

Here are some of the limitations of this research work:

- (a) Although local LLMs are tailored to Kenyan smallholder farmers, regional variations in

language use, livestock practices, and climate conditions may require further customization for broader applicability.

- (b) Local implementation of generative AI models may be resource-intensive, posing challenges for scalability without sufficient computational infrastructure.
- (c) Adoption of generative AI-driven advisory tools may be constrained by limited digital literacy, access to mobile devices, and internet connectivity in rural settings.
- (d) While the system has undergone technical validation using established NLP evaluation metrics, its effectiveness in real-world agricultural settings remains an open area for future research.

1.6 Research Justification

Livestock farming, in particular, plays a significant role in global GHG emissions, contributing substantial amounts of methane and nitrous oxide due to enteric fermentation and manure management processes, (Neethirajan, 2024). However, livestock farming is also key to achieving sustainable development goals, including poverty reduction and food security. Various mitigation strategies, such as the adoption of bio-gas systems, pasture-based practices, improved waste management, and optimized feed, can help reduce emissions. Moreover, providing extension support and education to livestock farmers is crucial for successfully implementing these sustainable practices.

By exploring how generative AI models can be applied to methane emission tracking, this study not only aims to address the significant challenge of methane emissions in livestock farming but also to demonstrate how advanced AI tools can optimize farming practices through predictive modeling and data integration. Factors like feed quality, animal genetics, and seasonal variations are critical in influencing methane output, and generative AI offers a novel approach to consider these variables in a comprehensive and efficient manner.

Chapter 2: Literature Review

This chapter reviews both empirical and theoretical literature on the application of generative AI and precision livestock farming in methane reduction strategies. With a focus on addressing emissions from enteric fermentation and manure management, the review explores existing frameworks, methodologies, and practical studies to establish the current state of knowledge, identify research gaps, and uncover opportunities to develop farmer-oriented advisory solutions.

2.1 Theoretical Literature Review

2.1.1 Greenhouse gases overview

The Earth's atmosphere contains various gases, including greenhouse gases like carbon dioxide (CO₂), methane (CH₄), and nitrous oxide (N₂O), which play a crucial role in regulating global temperature (Nakazawa, 2020). These gases absorb infrared radiation, trapping heat and warming the planet through the greenhouse effect (Jogdand, 2020). The greenhouse effect is essential for sustaining life however, human activities have significantly increased greenhouse gas concentrations, leading to global warming and climate change (Reichle, 2020). As atmospheric CO₂ levels have increased from about 270 parts per million to over 415 parts per million over the last 150 years, temperature, precipitation patterns, and carbon chemistry have all changed (Reichle, 2020). One of the main sources of methane emissions in agriculture is livestock, especially ruminants (Patil et al., 2022). Despite being less prevalent than CO₂, methane traps thirty times more heat.

2.1.2 Precision Livestock Farming and Emission Management

Studies have shown that PLF technologies can indirectly reduce emissions by enhancing overall farm efficiency, with improvements in fertility and yield leading to significant reductions in emissions per unit of product (Ferguson et al., 2024). The integration of PLF with computer modeling, particularly the combination of concept-driven mechanistic models and data-driven artificial intelligence, shows promise for optimizing production efficiency and sustainability (Tedeschi et al., 2021). However, challenges remain, including data quality issues and analytical complexities associated with sensor data in PLF implementation (Thumba et al., 2021); (Niloofar et al., 2021).

2.1.3 Generative AI and Large Language Models (LLMs)

Generative AI and LLMs have emerged as transformative technologies, revolutionizing numerous sectors by leveraging vast datasets to generate human-like text and other media. Models such as ChatGPT and DALL-E exemplify this innovation (Cobb, 2023). Recent advancements in LLM technology include the development of specialized models like small LLMs (sLLMs), designed to address hardware limitations, and techniques such as Reinforcement Learning with Human Feedback (RLHF), which enhance model performance (Linkon et al., 2024); (Ferrag et al., 2024).

2.1.4 Generative AI in Agriculture

Generative AI is increasingly being applied to decision support systems (DSS) in agriculture to optimize crop management and improve sustainability. These systems leverage various AI techniques, such as data mining, artificial neural networks, and Bayesian networks, to process input data and provide actionable insights (Armstrong et al., 2020). Recent advancements in AI-powered DSS integrate real-time data from digital and space-based technologies, incorporating blockchain for data integrity and enhanced cybersecurity measures (Pechlivani et al., 2023). These systems focus on improving precision information, generating accurate alerts, and deepening the understanding of complex farm systems (Smith, 2020). Studies have identified 12 distinct AI techniques and 73 input variables employed in agricultural DSS, with climate variables being the most frequently used (Carranza et al., 2020). While some solutions automate agricultural processes through actuators, others provide tailored recommendations to farmers, demonstrating the broad scope of AI applications in agricultural decision-making, (Carranza et al., 2020);(Smith, 2020).

2.1.5 Challenges and opportunities in livestock management

Livestock management faces numerous challenges globally, including climate change, poor veterinary care, and inadequate breeding resources (Eeswaran et al., 2022); (Moller et al., 2023). In West Africa and Bangladesh, specific issues like water scarcity and natural hazards further complicate livestock farming (Eeswaran et al., 2022);(Moller et al., 2023). However, emerging technologies offer opportunities for improvement. Unmanned Aerial Vehicles (UAVs) show promise in animal monitoring, detection, and tracking, although their adoption faces environ-

mental, technical, and economic hurdles (Alanezi et al., 2022). Precision livestock farming (PLF) technologies are transforming confined operations, with potential applications in extensive systems like rangelands (Menendez et al., 2022). Successful implementation of PLF requires a five-step process, emphasizing the need for precision system models (PSMs) to align with management goals and avoid unintended consequences (Menendez et al., 2022). Addressing these challenges and leveraging technological opportunities could significantly enhance livestock production systems worldwide. With the growing need for effective tracking and mitigation strategies, emerging technologies such as generative AI offer promising tools for improving methane emission monitoring

2.1.6 Ethical Implications of AI in Carbon Footprint Management

(Bhuiyan, 2024), The integration of AI in carbon footprint management introduces significant ethical implications that must be addressed to promote social equity and environmental justice. It is crucial to ensure that AI technologies do not disproportionately impact vulnerable communities, which have historically faced environmental injustices. Moreover, the potential for bias in AI algorithms necessitates continuous scrutiny to guarantee fair resource distribution and effective emission reduction strategies. Transparency and explainability in AI decision-making processes are vital, allowing stakeholders to understand and trust the outcomes of AI-driven initiatives. Finally, inclusive decision-making that incorporates diverse perspectives is essential to creating equitable environmental policies.

2.1.7 Limitations of General-Purpose LLMs in Agricultural Advisory Systems

Recent advancements in artificial intelligence (AI) and natural language processing (NLP) have led to the development of powerful LLMs such as GPT-4 (OpenAI, 2024) and Llama 2 (Touvron et al., 2023). These models have demonstrated exceptional performance across a range of domains including coding, medicine, law, and psychology, often approaching human-level reasoning (Bubeck et al., 2023); (Nori et al., 2023). The integration of AI copilots powered by LLMs is increasingly transforming business operations across diverse industries, assisting in complex decision-making and streamlining data workflows. For example, in healthcare, AI copilots help anticipate patient risks and enhance diagnostic accuracy, while in manufacturing, they improve efficiency, reduce equipment downtime, and enhance product quality (Li et al., 2023), (de Luis Balaguer et al., 2024).

2.2 Empirical Literature Review

2.2.1 Methane Emissions Reduction through Precision Livestock Farming (PLF)

In New Zealand, a key player in the dairy industry, methane emissions from enteric fermentation in cattle are a significant environmental concern. By adopting PLF technologies, including sensors to monitor animal feeding and manure management, the country has seen a 10% reduction in methane emissions per unit of product. This approach not only improves farm sustainability but also aligns with New Zealand's climate goals. Similarly, in Sweden, efforts to reduce methane emissions from livestock farming have been successful with the integration of AI-powered systems that monitor cattle feeding behavior. This solution has led to a 15% reduction in methane emissions, improving both farm efficiency and sustainability, supporting the country's ambitious environmental targets (Ferguson et al., 2024); (Tedeschi et al., 2021).

In Denmark, where the agricultural sector is a major contributor to greenhouse gas emissions, AI-based manure management systems have been introduced to optimize waste processing. This initiative has successfully reduced methane emissions from manure by up to 20%. The implementation of these technologies has allowed for greater environmental compliance and economic benefits for farmers by enhancing manure management practices. Meanwhile, in South Africa, PLF technologies integrated with AI systems have been applied in extensive livestock operations, leading to a reduction in methane emissions by 8-10%. These systems monitor grazing patterns and manage livestock behavior to minimize emissions while ensuring better resource utilization (Smith, 2020); (Menendez et al., 2022).

2.3 AI and Generative AI for Farm Management

The role of Generative AI in agriculture is becoming more prominent, especially for decision support systems (DSS) that optimize livestock management. In Brazil, generative AI models have been used to predict methane emissions based on various factors, including feeding habits, animal health, and environmental data. This approach has led to a 10-15% reduction in methane emissions through optimized feed management and farm practices. Moreover, in India, the integration of AI for feed optimization in the dairy sector has reduced methane emissions by 7%. The AI-based system enhances feeding efficiency, ensuring more sustainable practices in dairy farming, thereby improving both farm profitability and environmental impact (Patil et al., 2022); (Reichle, 2020).

In New Zealand's sheep farming sector, AI and IoT technologies have been integrated to monitor feeding behavior and methane production in real time. This system has resulted in a 10% reduction in methane emissions by optimizing the feed types and schedules. These advancements highlight the growing importance of AI in improving farm practices and reducing GHG emissions. Similarly, in Mexico, AI has been applied to monitor methane emissions in cattle farming, allowing for more accurate real-time tracking and enabling better decision-making. This has contributed to greater transparency in emission reduction efforts, aligning with national climate policies ([Pechlivani et al., 2023](#)); ([Thumba et al., 2021](#)).

2.3.1 Innovative Technologies in Livestock Monitoring

The use of Unmanned Aerial Vehicles (UAVs) in livestock monitoring is becoming increasingly popular due to their ability to provide real-time data on animal health and behavior. In Australia, UAVs equipped with sensors have been used to monitor livestock and track emissions, helping farmers detect high-emission animals and implement targeted interventions. This technology has significantly improved both livestock management and methane emission reduction, particularly in large-scale operations. Additionally, in Brazil, UAVs have been integrated into livestock emission monitoring systems, allowing for efficient data collection and targeted emission reductions. This method has enhanced emission tracking and the overall efficiency of farm management practices, reducing methane emissions and improving sustainability ([Alanezi et al., 2022](#)); ([Menendez et al., 2022](#)).

2.3.2 Impact of AI and Blockchain Integration in Emission Management

The integration of blockchain and AI for carbon footprint management has shown promising results in Colombia, where livestock farming contributes significantly to methane emissions. Blockchain's ability to ensure data integrity and transparency, coupled with AI's predictive capabilities, has allowed for more accurate monitoring and verification of methane emissions. This solution not only enhances the efficiency of emission reduction strategies but also ensures that farmers can meet international climate standards. The combination of these technologies is also being explored in Argentina, where advanced machine learning models are used to predict and reduce methane emissions in beef production. The result has been a 20% reduction in methane emissions through improved herd management practices and feed optimization ([Caranza et al., 2020](#)); ([Linkon et al., 2024](#)).

2.3.3 Machine Learning Models for Methane Emission Prediction

A study ([Shadpour et al., 2022](#)) explored the application of milk mid-infrared spectroscopy (MIRS) data to estimate methane (CH₄) emissions in lactating Canadian Holstein cows, aiming to address the challenge of CH₄ emission estimation in large cattle populations. The research analyzed 181 weekly average CH₄ measurements from 158 Canadian cows and 217 records from 44 Danish cows, incorporating data on milk yield (MY), fat yield (FY), protein yield (PY), and other production traits.

Various machine learning approaches were employed, including artificial neural networks (ANN), partial least squares regression (PLS), and different predictor sets (e.g., MY, FY, PY, and MIRS), with ANN models using nonlinear architectures and random cross-validation to assess predictive accuracy. The study found that MIRS data alone, or combined with production traits, provided superior predictions of CH₄ emissions, with ANN models yielding the best performance. The inclusion of MIRS resulted in higher prediction accuracy ($r = 0.586\text{--}0.717$, RMSE = 69.09–96.20) compared to other sets. Cross-validation using random sampling showed better results than block cross-validation, highlighting the need for more data to refine the models for large-scale prediction of CH₄ emissions.

The study by ([Zhang et al., 2020a](#)), investigates the relationship between ruminal microbiota composition and methane emissions in livestock, addressing the limitations of traditional linear models in accurately quantifying the effects of microbiota on methane emissions. Using microbial operational taxonomic units (OTUs) and methane emissions data, the authors incorporated host genetic factors into their analysis. Employing a four-component Bayesian mixture model, they classified microbial OTUs into four levels of effect sizes, providing a more flexible estimation of microbiota impacts compared to linear models. The study found that microbiota accounted for 7% of the variance in methane emissions, with host genetics explaining 24% and the combined model explaining 31%. Notably, genera like Coriobacteriaceae were identified as significantly associated with methane emissions ([Zhang et al., 2020a](#)).

2.3.4 Retrieval-Augmented Generation in Livestock Emission Monitoring

According to ([Han et al., 2024](#)), Retrieval-Augmented Generation (RAG) enhances large language models (LLMs) by integrating real-time information retrieval, overcoming the limitations of static, pre-trained knowledge. This framework consists of three primary processes: retrieval,

which gathers relevant information from various sources; augmentation, which enriches the retrieved data with additional context; and generation, where the LLM produces the final response based on the augmented content.

RAG-based LLMs hold significant potential for automating tasks such as data extraction, summarization, research synthesis, and trend identification in systematic literature reviews (SLRs). A proposed framework for automating SLRs using RAG-based LLMs consists of four stages: literature search, literature screening, data extraction, and information synthesis. This approach aims to enhance efficiency in knowledge synthesis and streamline the SLR process. However, further research is needed to optimize the interplay between LLM selection, RAG methodologies, and prompt engineering. Key challenges include integrating domain-specific LLMs, processing multimodal data, and utilizing diverse retrieval sources effectively.

Artificial Intelligence (AI) holds great promise for transforming healthcare by addressing challenges like supply-and-demand imbalances through innovations such as medical AI assistants, chatbots, and robots. This paper by ([Bora and Cuayáhuitl, 2024](#)), examines the integration of RAG with fine-tuned LLMs to enhance performance in resource-constrained environments, particularly for medical data. The study evaluates open-source LLMs, including Flan-T5-Large, LLaMA-2-7B, and Mistral-7B, fine-tuned with medical datasets like Meadow-MedQA and MedMCQA.

It compares models using RAG and fine-tuning, alongside base models with or without these techniques. The findings reveal that RAG and fine-tuning significantly improve performance, with standard question answering (Type A) outperforming the language generation approach with confidence scores (Type B). Among the models, Mistral-7B demonstrated the best accuracy, achieving 5% exact match accuracy in multiple-choice question evaluations when fine-tuned with RAG. Although Flan-T5-Large was the fastest, its performance was lower, suggesting it is better suited for speed-focused tasks. LLaMA-2-7B offered a balanced trade-off between speed and accuracy. The study concludes that RAG-based LLMs have the potential to support healthcare providers in answering medical queries efficiently and could be integrated into clinical workflows and patient education. Future research should focus on real-world testing and optimizing models for near real-time response generation ([Bora and Cuayáhuitl, 2024](#)).

2.3.5 RAG vs. Fine-Tuning: Empirical Insights from Agricultural AI

Recent empirical research has compared the effectiveness of Retrieval-Augmented Generation (RAG) and fine-tuning for domain-specific LLM applications in agriculture. In benchmarking LLaMA 2, GPT-3.5, and GPT-4, the study found that RAG improved accuracy by around five percentage points when grounded in contextually relevant data, while fine-tuning added another six points but required more data and computational effort. Fine-tuned models offered more concise, domain-specific outputs, but RAG proved more flexible and cost-efficient, especially for scenarios requiring frequent updates or local adaptations. The same work also introduced a pipeline for generating agricultural Q&A datasets using GPT-4 and structured document processing supporting modular, scalable systems like FarmerVICE. These findings reinforce the choice of RAG in this study while identifying fine-tuning as a promising future enhancement.

2.4 Research Gap

Existing literature highlights the significant role of livestock farming in global methane emissions, particularly from enteric fermentation and manure management. While various mitigation strategies have been proposed, their adoption remains limited in resource-constrained environments like rural Kenya due to insufficient capacity-building initiatives, a lack of tailored solutions, and inadequate technical resources. Although machine learning has shown potential for enhancing emissions predictions and guiding climate-smart practices, current models fail to provide context-specific, scalable tools suitable for smallholder farmers. This research addresses these gaps by developing a machine-learning-based advisory system offering personalized, actionable recommendations for methane emissions reduction in livestock farming. The study will be instrumental in advancing localized Tier 2 methane emissions estimation methodologies, contributing to ongoing research and fostering future innovations in the field.

Chapter 3: Methodology

3.1 Research Design

This study follows a mixed-methods research design, integrating quantitative and qualitative approaches to develop and evaluate an AI-powered farmer advisory system for methane reduction in cattle farming. The quantitative component applies the structured Cross-Industry Standard Process for Data Mining (**CRISP-DM**) methodology to guide AI model development. The **CRISP-DM** framework, enhanced with modern Large language model (**LLM**) practices, is used to develop a farmer advisory system that leverages **RAG** architecture and the Llama 3.2-3B model. The qualitative component consists of expert consultations and iterative feedback from farmers and agricultural specialists to refine the system's usability and relevance. The implementation consists of seven integrated components (Figure 3.1), each addressing key challenges in agricultural AI systems. This mixed-methods approach ensures that AI-driven recommendations are both data-driven and contextually relevant.

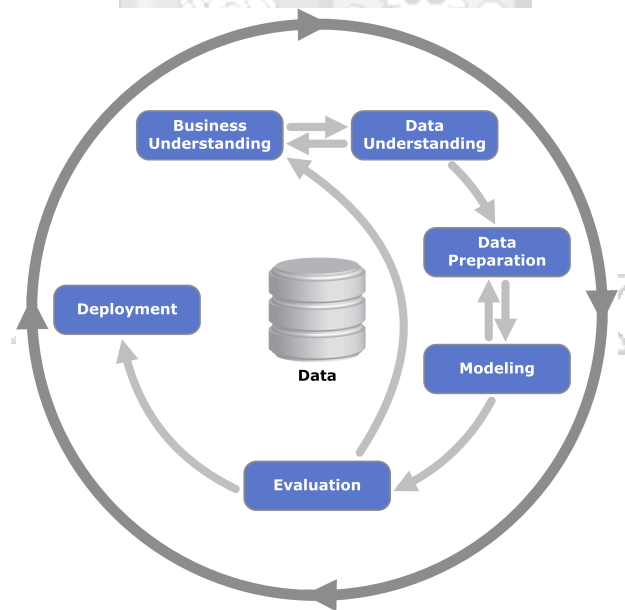


Figure 3.1: CRISP-DM framework ([Wikimedia Commons, 2020](#))

To make sure the model-building process is concentrated on the most relevant data for decision making, the first two stages, business understanding and data understanding, are essential. To address the identified research gap, Figure 3.4 (Conceptual Framework) illustrates how the proposed Artificial Intelligence (**AI**)-powered farmer advisory system integrates data retrieval, generative **AI** processing, and farmer feedback loops to generate actionable recommendations for methane emission reduction. This framework ensures that AI-generated insights are grounded

in evidence-based research, enhancing transparency and interpretability.

3.2 Data Understanding

3.2.1 Data Collection from the CORE Repository

This study will utilize the [CORE](#) repository's [API](#) to programmatically retrieve academic papers and research articles. The CORE repository provides access to over 32 million full-text research papers from nearly 11,000 institutional, preprint, and publisher repositories worldwide ([Knoth et al., 2023](#)), making it an ideal resource for exploring contemporary advancements in livestock methane emissions, AI-driven emission tracking, and mitigation strategies.

3.2.2 Sample Size Justification

For the initial phase, 100 research papers will be selected based on relevance, recency (2021–present), and their credibility as open-access publications aggregated from institutional repositories and journal publishers in the [CORE](#) dataset.

Sample size is iteratively assessed. As model performance is evaluated, additional research papers will be integrated to enhance coverage and accuracy. A purposive sampling approach, using targeted queries and temporal filters, ensures that the dataset remains focused, relevant, and expandable over time. Feedback from farmers and domain experts will further refine the dataset to align with real-world agricultural challenges.

3.2.3 Data Processing

The [API](#) returns responses as JavaScript Object Notation ([JSON](#)) objects containing metadata and full-text documents. These will be processed to extract key fields (title, abstract, publication year) and stored in a structured format, a Pandas DataFrame for downstream analysis.

The study will employ targeted queries combining specific keywords and temporal filters (2021–present) to retrieve relevant literature. Queries such as "livestock emissions year >2020" are designed to identify trends and factors influencing methane emissions across key species, including cattle, sheep, goats, and swine. "Livestock manure management year >2020" focuses on recent innovations in manure handling practices aimed at emission reduction. Meanwhile, "livestock emissions mitigation strategies year >2020" highlights emerging approaches, such as dietary modifications, selective breeding, and the application of [PLF](#) technologies. To address

policy dimensions, the query "Paris Agreement year >2021" captures research on the accord's implications for livestock-related emission targets. Additionally, "Precision Livestock Farming >2021" examines advancements in data-driven management systems for sustainable operations, while "Carbon emissions >2021" provides broader insights into agricultural carbon footprints and the sector's contribution to climate change.

The study intends to start with a sample size of 5 and iteratively refined as the advisory system is deployed. By leveraging CORE's open-access infrastructure and targeted query design, this approach ensures the dataset reflects the latest advancements in methane mitigation while grounding the AI-driven advisory system in evidence-based research.

3.3 Data Preparation

3.3.1 Document Loading

The fetched [JSON](#) objects will be parsed to extract key fields, including title, abstract, full_text, and publication_year. These fields will be loaded into a structured format, such as a Pandas DataFrame, to facilitate pre-processing and analysis. This step ensures that the raw data is transformed into a usable format, ready for downstream tasks such as text splitting and storage.

3.3.2 Text Splitting

To preserve contextual integrity and capture nuanced information, the text will be split into smaller chunks using LangChain's 'RecursiveCharacterTextSplitter'. A chunk size of 2000 tokens with an overlap of 300 tokens will be employed to balance granularity and context retention, ensuring that semantic relationships between chunks are maintained. This approach is particularly effective for handling large documents while preserving the semantic relationships between chunks.

3.3.3 Creating Embedding

This study will utilize embedding models to convert text into fixed-length numerical vectors, capturing the semantic meaning of the text. These embeddings serve as numerical fingerprints, enabling search systems to identify relevant documents based on semantic similarity rather than keyword matching. Specifically, the study will employ LangChain's OllamaEmbeddings li-

brary, which is well suited for localized models and ensures efficient processing of domain-specific text.

3.3.4 Storage

Once the embeddings are generated, the embeddings will be stored in a vector database. A vector database stores data as high-dimensional vectors, which are numerical representations of features or attributes. These vectors, often generated using embedding functions like machine learning models or word embeddings, enable efficient similarity search and retrieval, making them ideal for handling unstructured data such as text, images, and audio (Han et al., 2023).

This study aims to utilize the Chroma vector database for storage. Chroma is an open-source, AI-native database designed to streamline the development of LLM-powered applications. Its native metadata support and lightweight architecture make it an ideal choice for efficiently storing and retrieving embeddings, while also enabling seamless integration of domain-specific knowledge into LLMs.

3.4 Machine Learning Model

3.4.1 Transformer Architecture

It predicts text based on a given prompt, as seen in models like GPT and LLaMA. In this autoregressive setup, the model iteratively predicts the next token based only on previous tokens, a process known as decoder-only generation. While there is a practical maximum context length, we assume sequences of arbitrary length for simplicity (Frieder et al., 2024). In figure 3.2 we can see that of a transformer model.

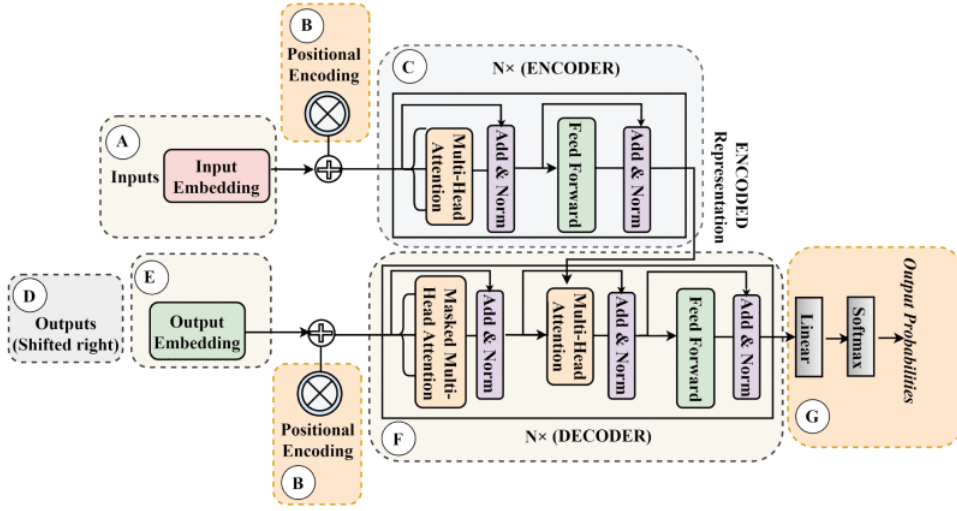


Figure 3.2: Transformer Model Architecture (Raijan et al., 2024)

We define the shorthand notation $S^* := \bigcup_{n \in \mathbb{N}} S^n$ for a set S , which represents the set of sequences $s = (s^{(i)})_{i=1}^n \subset S$ of any length $n \in \mathbb{N}$. For a function $F : S_1 \rightarrow S_2$, we use $F^* : S_1^* \rightarrow S_2^*$ to denote the mapping applied entrywise, as follows:

$$F^*(s) := (F(s^{(i)}))_{i=1}^n.$$

Tokenization \mathcal{K} . To define tokens, we consider an injective mapping $K : A^* \rightarrow T^*$, converting a sequence of characters $a = (a^{(i)})_{i=1}^N$ in an alphabet A to a sequence of tokens $(t^{(i)})_{i=1}^n$, where $T := \{1, 2, \dots, M\}$. Rather than encoding each character individually (such as Unicode), subword tokenization is often used in LLMs. This method creates a vocabulary by merging frequent character sequences, balancing sequence length n and vocabulary size M . For example, GPT-4 tokenizes “discontinuity” into subwords: “dis,” “contin,” and “uity” (Frieder et al., 2024).

Embedding \mathcal{E} . (Frieder et al., 2024) The transformer is a neural network $\mathcal{T} : \mathcal{E}^* \rightarrow \mathcal{E}^*$ that converts a series of embeddings e to another sequence of the same length that is enhanced with contextual information. In an autoregressive setting, the i -th element of $\mathcal{T}(e)$ is dependent on all previous embeddings but independent of those that follow. A Euclidean space $\mathcal{E} := \mathbb{R}^d$ is used to map tokens (represented by subword indices) using an embedding function $\mathcal{E} : T \rightarrow \mathcal{E}$. The distance $\|\mathcal{E}(t^{(i)}) - \mathcal{E}(t^{(j)})\|$ should illustrate the linguistic resemblance between subwords

$t^{(i)}$ and $t^{(j)}$. The transformer model and the embedding are usually trained together, with the embedding being initialized with random values.

Positional Encoding \mathcal{P} . (Frieder et al., 2024) Since \mathcal{E} processes each token $t^{(i)}$ independently, it lacks positional information. To address this, we add positional encodings, represented by a mapping $\mathcal{P} : \mathcal{E}^* \rightarrow \mathcal{E}^*$. A common choice is:

$$\mathcal{P}((e^{(i)})_{i=1}^n) := (e^{(i)} + p(i))_{i=1}^n$$

where $p : \mathbb{N} \rightarrow \mathcal{E}$ can be a predefined injective function, such as a sinusoid, or learned during training (similar to the embedding \mathcal{E}).

In summary, tokenization \mathcal{K} , followed by embedding \mathcal{E} for each token and positional encoding \mathcal{P} , transforms the text $a \in A^*$ into a sequence of embeddings:

$$e := (\mathcal{P} \circ \mathcal{E}^* \circ \mathcal{K})(a) \in \mathcal{E}^*$$

where the length e depends on a and the tokenization algorithm.

Transformer \mathcal{T} . (Frieder et al., 2024) The transformer is a neural network $\mathcal{T} : \mathcal{E}^* \rightarrow \mathcal{E}^*$ that maps a sequence of embeddings e to another sequence of the same length, enriched with contextual information. In an autoregressive setup, the i -th element of $\mathcal{T}(e)$ depends on all prior embeddings but not on those after it.

Typically, the transformer consists of $L \in \mathbb{N}$ blocks, each containing self-attention maps \mathcal{A}^ℓ , normalizing layers $\mathcal{N}^{A,\ell}, \mathcal{N}^{M,\ell}$, and feed-forward multi-layer perceptrons \mathcal{M}^ℓ :

$$\begin{aligned} \mathcal{T} := & \left((\text{Id} + \mathcal{M}_L^* \circ \mathcal{N}_{M,L}^*) \circ (\text{Id} + \mathcal{A}_L \circ \mathcal{N}_{A,L}^*) \right) \circ \dots \\ & \circ \left((\text{Id} + \mathcal{M}_1^* \circ \mathcal{N}_{M,1}^*) \circ (\text{Id} + \mathcal{A}_1 \circ \mathcal{N}_{A,1}^*) \right) \quad (1) \end{aligned}$$

Here, Id denotes the identity (residual) connection, and the addition is element-wise. Each layer has its own distinct trainable parameters.

3.5 Benchmarking

For this study, we used the Meta-provided benchmark evaluations for the LLaMA 3.21B model, as . These evaluations are based on over 150 benchmark datasets covering a wide range of tasks, including natural language understanding, image understanding and visual reasoning. Meta’s benchmarks include both quantitative performance metrics and extensive human evaluations comparing LLaMA to other competing models in real-world scenarios. LLaMA 3.21B was chosen for this study due to its strong performance across diverse tasks, as shown in Meta’s benchmarks, and its lightweight design, which enables efficient deployment in various settings (AI, 2025). In figure 3.3 presents the benchmark tests.

Category Benchmark	Llama 3.2 1B	Llama 3.2 3B	Gemma 2 2B IT (5-shot)	Phi-3.5 - Mini IT (5-shot)
General				
MMLU (5-shot)	49.3	63.4	57.8	69.0
Open-rewrite eval (0-shot, rougeL)	41.6	40.1	31.2	34.5
TLDR9+ (test, 1-shot, rougeL)	16.8	19.0	13.9	12.8
IFEval	59.5	77.4	61.9	59.2
Math				
GSM8K (0-shot, CoT)	44.4	77.7	62.5	86.2
MATH (0-shot, CoT)	30.6	48.0	23.8	44.2
Reasoning				
ARC Challenge (0-shot)	59.4	78.6	76.7	87.4
GPQA (0-shot)	27.2	32.8	27.5	31.9
Hellaswag (0-shot)	41.2	69.8	61.1	81.4
Tool use				
BFCL V2	25.7	67.0	27.4	58.4
Nexus	13.5	34.3	21.0	26.1
Long context				
InfiniteBench/En.MC (128k)	38.0	63.3	-	39.2
InfiniteBench/En.QA (128k)	20.3	19.8	-	11.3
NIH/Multi-needle	75.0	84.7	-	52.7
Multilingual				
MGSM (0-shot, CoT)	24.5	58.2	40.2	49.8

Figure 3.3: Lightweight Instruction Tuner Benchmarks (AI, 2025)

LLaMA 3.2-1B is suited for the methane emissions retrieval-augmented generation (RAG) farmer advisory system, as it delivers a balanced combination of performance and efficiency

across key benchmarks. The model achieves competitive scores in general knowledge (MMLU: 49.3, 5-shot) and scientific reasoning (ARC Challenge: 59.4, 0-shot; GPQA: 27.2), while outperforming comparable models like Gemma 2B in tasks critical to our application—notably summarization (TLDR9+: 16.8 vs. 13.9) and text simplification (Open-rewrite ROUGE-L: 41.6 vs. 31.2). These capabilities are essential for distilling complex methane research into practical, farmer-friendly insights. Furthermore, LLaMA 3.2-1B demonstrates robust long-context comprehension (InfiniteBench/En.QA: 38.0 at 128k tokens), enabling accurate extraction of information from lengthy scientific papers. Its lightweight 1B-parameter architecture ensures cost-effective deployment, contrasting with larger models like Phi-3.5 Mini, which incur higher computational overhead for marginal gains in reasoning tasks. Given these trade-offs, LLaMA 3.2-1B represents an optimal choice for scalable, real-world agricultural advisory systems, where accuracy, clarity, and resource efficiency are prioritized.

3.6 Machine Learning Model Evaluation and Optimization

The proposed farmer advisory system will be assessed through comprehensive evaluation using established metrics, including ROUGE (Recall-Oriented Understudy for Gisting Evaluation) and BERTScore. ROUGE quantifies overlap between generated and reference summaries (Chen et al., 2019), while BERTScore leverages contextual embeddings to measure semantic similarity (Zhang et al., 2020b).

To gauge precision in question-answering systems, metrics such as accuracy, F1 score, and precision are employed, offering insights into their effectiveness. These tools enable quantitative performance analysis, highlighting the system's superiority over existing commercial solutions in document processing and information retrieval tasks (Muludi et al., 2024).

3.6.1 Accuracy

Accuracy refers to the ratio of correct answers relative to the total number of responses evaluated. It is determined by dividing the number of correct predictions by the total instances and expressing the result as a percentage, as noted by (FIELDING and BELL, 1997). This metric quantifies the system's effectiveness in delivering accurate outcomes, calculated using the formula presented in equation (2).

$$\text{Accuracy} = \left(\frac{\text{Correct Predictions}}{\text{Total Predictions}} \right) \times 100\% \quad (2)$$

3.6.2 BERTScore

BERTScore is an automatic evaluation measure that uses contextual embeddings to evaluate how similar each token in a reference sentence is to each token in a candidate sentence in text generation tasks (Zhang et al., 2020b). In contrast to conventional embedding techniques, BERTScore makes use of contextualized representations, which means that the embeddings change based on the sentence structure. Because of this context awareness, BERTScore can recognize sentences that are semantically similar even if their word order is different.

For recall computation, each token in the reference sequence is matched with the most similar token in the candidate sequence, and the same applies to precision calculation. A greedy matching algorithm is used to maximize the similarity score. The precision (see equation (4)), recall (see equation (3)), and F1-score (see equation (5)) for a reference sequence and a candidate sequence are defined as follows:

$$R_{\text{BERT}} = \frac{1}{|x|} \sum_{x_i \in x} \max_{\hat{x}_j \in \hat{x}} x_i^\top \hat{x}_j \quad (3)$$

The notation indicates selecting the most similar token for each and represents their cosine similarity.

$$P_{\text{BERT}} = \frac{1}{|\hat{x}|} \sum_{\hat{x}_j \in \hat{x}} \max_{x_i \in x} x_i^\top \hat{x}_j \quad (4)$$

where is the precision score, with the same notation as in Eq. (3). Here, selects the most similar reference token for each candidate token.

$$F_{\text{BERT}} = \frac{2 \times P_{\text{BERT}} \cdot R_{\text{BERT}}}{P_{\text{BERT}} + R_{\text{BERT}}} \quad (5)$$

The F1-score is calculated by taking the harmonic mean of recall and precision.

Although cosine similarity values theoretically range from, in BERTScore calculations, they

are rescaled to lie within for practical applications.

3.6.3 ROUGE

The Recall-Oriented Understudy for Gisting Evaluation (**ROUGE**) is a widely adopted metric for assessing text generation models by quantifying the overlap between generated (candidate) text and human-written (reference) text (Lin, 2004). **ROUGE** encompasses multiple variants, each focusing on different granularities of n-gram matches. Among these, ROUGE-L is the most prevalent due to its reliance on the Longest Common Subsequence (LCS)—the lengthiest contiguous or non-contiguous word sequence shared between the candidate and reference texts.

ROUGE-L computes two core metrics; Precision, the fraction of n-grams in the candidate text that also appear in the reference text (Equation 6). Recall, the proportion of n-grams in the reference text that are matched in the candidate text (Equation 7). These metrics are combined into a harmonized F1-score (Equation 8), which balances precision and recall to provide a comprehensive evaluation of text quality.

$$\text{ROUGE-L}_{\text{recall}} = \frac{\text{LCS}(\text{candidate}, \text{reference})}{\#\text{words in reference}} \quad (6)$$

$$\text{ROUGE-L}_{\text{precision}} = \frac{\text{LCS}(\text{candidate}, \text{reference})}{\#\text{words in candidate}} \quad (7)$$

$$\text{ROUGE-L}_{F1\text{-Score}} = \frac{2 \times \text{recall} \cdot \text{precision}}{\text{recall} + \text{precision}} \quad (8)$$

The RAG system is evaluated through a multi-metric framework: ROUGE measures lexical overlap for surface-level correctness, BERTScore uses contextual embeddings to assess semantic faithfulness, and Accuracy tracks exact matches against references. Human evaluation complements these by assessing fluency, factual accuracy, and cultural relevance, ensuring real-world applicability. Together, these metrics provide a comprehensive toolkit for evaluating technical performance and practical usability.

3.7 Deployment

The deployment of the system integrates a user-friendly interface and scalable cloud infrastructure to ensure accessibility and robust performance.

3.7.1 User Interface

The User Interface (UI) is designed as a responsive web application using React.js, deployed via Firebase Hosting to guarantee seamless access across both desktop and mobile platforms. This interface enables farmers and researchers to input queries and view actionable insights, including recommendations for feeding optimization, disease prevention, and other best practices for precision livestock farming.

3.7.2 Model

The RAG backend will be deployed on cloud infrastructure to ensure scalability and enable real-time processing. The model will be served via an API, using FastAPI along with ChromaDB as the vector database for efficient retrieval. A PostgreSQL document store will be used to manage both structured and unstructured data.

3.8 Conceptual Diagram

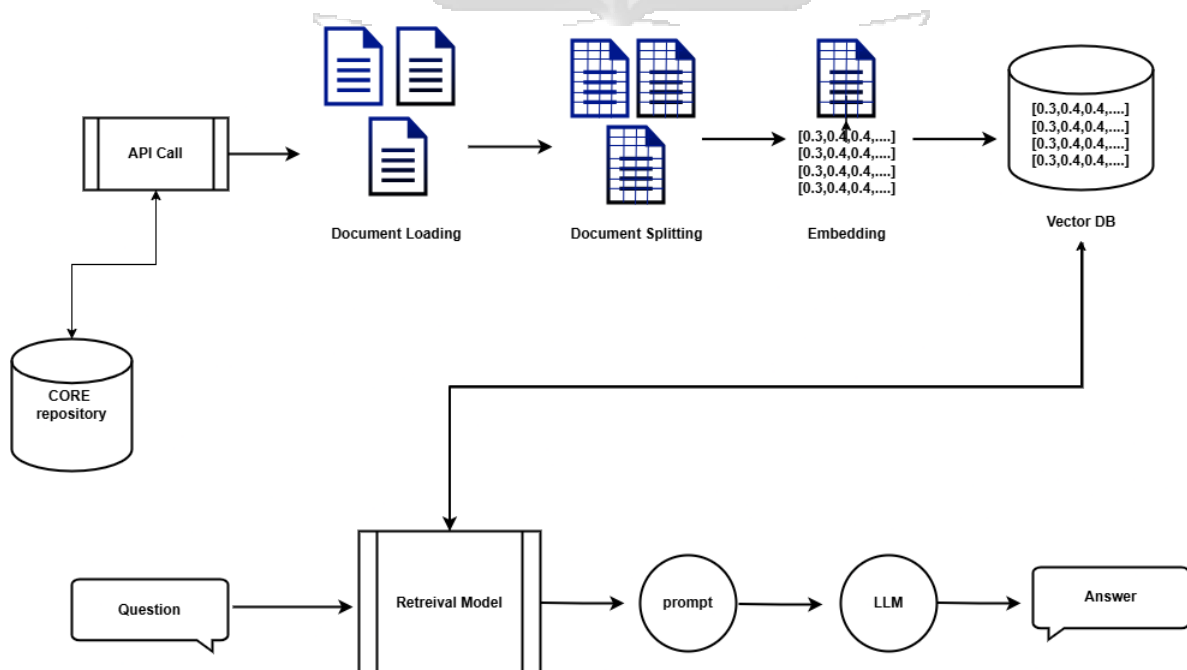


Figure 3.4: Conceptual Diagram

Chapter 4: System Design and Architecture

4.1 System Modeling

This chapter presents the architecture and design of, farmerVice, an AI-driven advisory system for methane mitigation in Kenyan smallholder farming communities, combining Retrieval-Augmented Generation [RAG](#) with the Llama 3.2-1B model to deliver context-aware recommendations within typical rural infrastructure constraints (2G networks, shared low-end devices). The resource-optimized design employs lightweight document retrieval (k=2 most relevant documents), Least Recently Used ([LRU](#)) caching for frequent queries, and webSocket streaming to handle intermittent connectivity, implemented via a FastAPI backend with ChromaDB vector storage and HuggingFace’s all-MiniLM-L6-v2 embeddings.

4.2 System Requirements

System requirements define the essential functionalities a system must provide and the operational constraints it must satisfy. This study’s requirements are designed to directly address the critical capacity-building gap in methane mitigation identified in Chapters 1–2: while proven strategies for manure management and feed optimization exist ([Khoshnevisan et al., 2021](#); [Beauchemin et al., 2020](#)), smallholder farmers in Kenya lack accessible, localized guidance to implement these techniques effectively ([Khanal et al., 2022](#); [Yan et al., 2024](#)).

4.2.1 Functional Requirements

The system operates as an AI-powered agricultural advisory chatbot that provides evidence-based methane mitigation strategies to Kenyan smallholder dairy farmers through natural language processing. Leveraging the [CORE](#) research repository, it analyzes user queries against peer-reviewed literature to generate contextually appropriate recommendations, with particular emphasis on manure management and feed optimization. The platform incorporates dynamic Swahili translation capabilities that maintain scientific precision in agricultural terminology while adapting content to local measurement systems and farming practices. Designed for optimal usability, the system delivers tiered implementation guidance with cost-efficiency indicators, employing generalized best practices validated for typical smallholder operations rather than requiring farm-specific data inputs.

4.3 System Components

4.3.1 Data Ingestion and Processing

The system ingests peer-reviewed literature through a multi-stage pipeline, beginning with parallel API fetching, where the [CORE](#) repository is queried concurrently across multiple research topics using Python's `ThreadPoolExecutor`, ensuring efficient data retrieval and merging results into a structured dataset. Next, in the text deduplication stage, raw documents undergo normalization, including lowercase conversion and whitespace trimming, followed by near-duplicate detection using fuzzy string matching (`fuzzywuzzy` library) with a similarity threshold of 90% or higher to eliminate redundancy. Finally, in the semantic chunking process, documents are segmented into 1000-character chunks with 100-character overlaps using `RecursiveCharacterTextSplitter`, preserving critical metadata such as titles, authors, and [DOIs](#) to maintain source attribution and retain meaningful context.

4.3.2 Vector Database (ChromaDB)

Text chunks are transformed into dense vector representations using the `all-MiniLM-L6-v2` model from HuggingFace, selected for its balance of accuracy and computational efficiency. These vectors are then indexed in ChromaDB with disk persistence (`./chroma_langchain_db`), ensuring efficient retrieval, while batched insertion of 100 chunks per batch optimizes storage performance.

4.3.3 RAG Pipeline

For retrieval, a similarity-based mechanism fetches the top two most relevant documents per query, filtered by metadata fields such as field of study to enhance relevance. The system utilizes a locally hosted OllamaLLM (Llama3.2 1B parameter variant) to generate responses with controlled randomness (`temperature = 0`, `top_k = 20`) and repetition regulation (`repeat_penalty = 1.1`). To ensure consistency, prompt engineering employs a structured template that standardizes output formatting as shown in [Figure 4.1](#).

```

custom_prompt = ChatPromptTemplate.from_template("""
You are an expert in livestock farming and emissions reduction policies. Your task is to educate farmers on these topics in a clear and practical manner.

You will be given a set of queries related to livestock farming and emissions reduction policies:

<context>
{context}
</context>

Follow these steps:
1 **Understand the Queries**- Identify key topics, such as livestock emissions, manure management, mitigation strategies, the Paris Agreement, precision livestock farming, and carbon emissions in farming.
2 **Explain Why It Matters**- For each query, provide a brief, **farmer-friendly** explanation of why it is important and how it impacts their work.
3 **Give Practical Advice**- Provide **simple, actionable tips** to help farmers adopt sustainable practices and comply with policies.
4 **Use a Clear Structure**- Format your response with **headings for each topic** and make it **concise yet informative**.
5 **Wrap in Answer Tags**- Place your complete response inside **<answer>** tags.

Example Response Format:

** Manure Management & Methane Reduction**
*Why It Matters:* Managing manure effectively reduces methane, which contributes to climate change.
*Practical Tip:* Cover manure storage areas to reduce methane emissions by up to 50%.
*Regulatory Insight:* The Paris Agreement encourages emission reduction in agriculture.

** Precision Livestock Farming**
*Why It Matters:* Using sensors and AI can optimize feed, reducing emissions.
*Practical Tip:* Invest in precision feeding tools to cut feed waste and emissions.

""")

```

Figure 4.1: Custom Prompt Template

4.3.4 API Endpoints (FastAPI)

The system implements a dual-interface API architecture to support both real-time interaction and system administration. The primary WebSocket interface maintains session state through a 5-message conversational history, enabling context-aware multi-turn dialogues while simultaneously streaming responses with complete source attribution (including author names, publication years, and verifiable DOI links). For system management, RESTful administrative endpoints provide critical functionality: the `/fetch-research` endpoint initiates the complete Extract Transform Load (ETL) pipeline (document acquisition, preprocessing, and vector embedding), while the `/db` endpoint exposes database metadata, including document volume and schema specifications, for monitoring and debugging purposes. This design ensures both interactive responsiveness and system maintainability while preserving academic rigor through complete source transparency.

4.3.5 Document Metadata Processing

The system ensures rigorous scholarly attribution through structured metadata processing, combining a dedicated link parser that extracts actionable resources from CORE's nested links field (including PDF downloads and reader views) with standardized Authors (Year) Title citations featuring hyperlinked DOIs (e.g., <https://doi.org/10.1016/...>). This integrated approach simultaneously enables verification through peer-reviewed source linking, supports reproducibility via persistent identifiers, and facilitates farmer engagement through both human-readable citations and machine-actionable links, thereby bridging academic rigor with practical usability.

Figure 4.2 below is the diagrammatic representation of the farmerVICE.

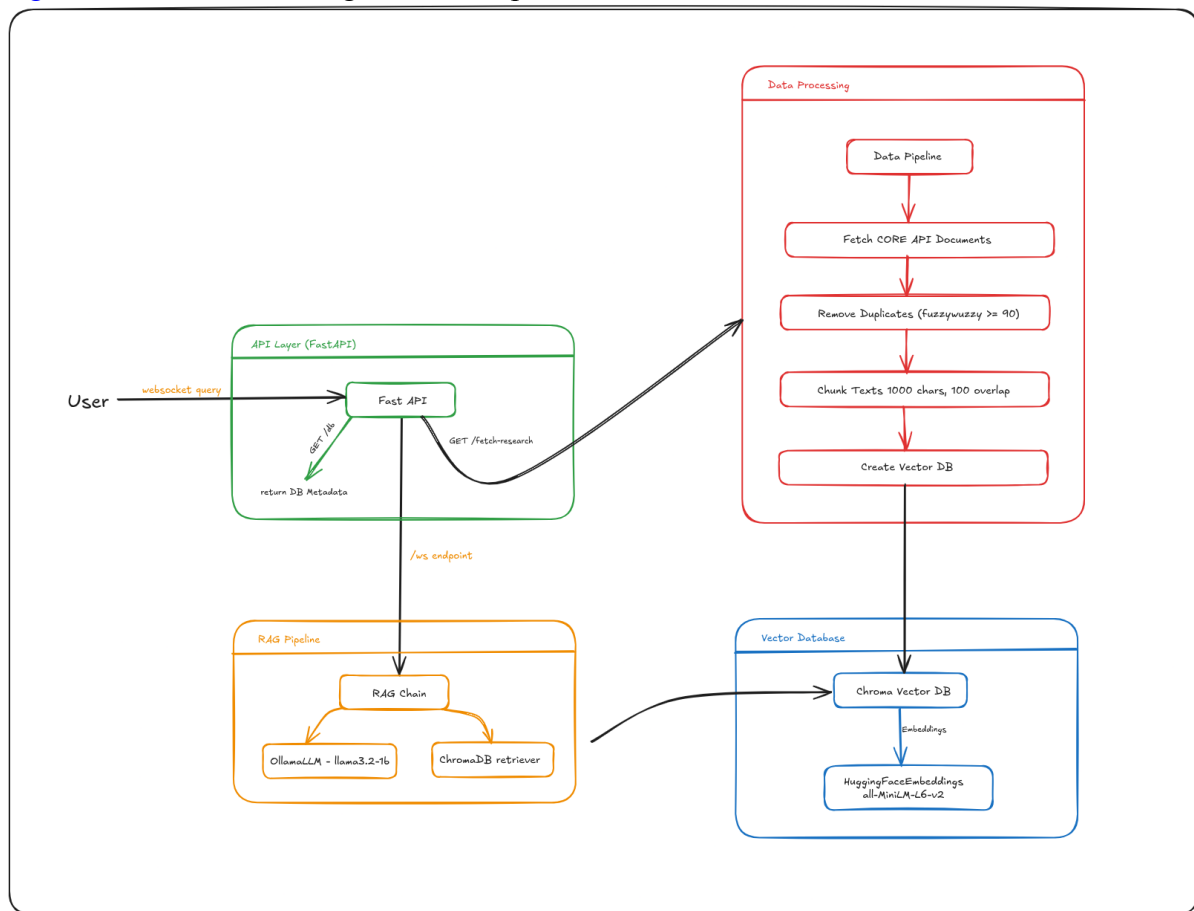


Figure 4.2: farmerVICE System Architecture

4.4 System Workflow

The proposed architecture demonstrates a seamless integration of agricultural knowledge management and AI-driven analysis to deliver evidence-based methane mitigation guidance. This end-to-end process operates as follows:

- i. **Research Ingestion:** The system initiates knowledge acquisition through parallelized API queries to CORE’s repository using ThreadPoolExecutor, targeting six key research domains with year-filtered search strings:

Each query enforces publication year filters (2020+ for livestock studies, 2021+ for policy/technology) to prioritize recent evidence. The concurrent fetching process merges results into a unified corpus while preserving source metadata.

- ii. **Text Preprocessing:** Retrieved documents undergo normalization (lowercase conversion, whitespace trimming) and fuzzy deduplication (90% similarity threshold via fuzzywuzzy).

Table 4.1: CORE API Query Parameters

Research Domain	Search Query
Livestock Emissions	"livestock emissions year>2020"
Manure Management	"livestock manure management year>2020"
Mitigation Strategies	"livestock emissions mitigation strategies year>2020"
Policy Context	"The Paris Agreement year>2021"
Precision Farming	"Precision Livestock Farming>2021"
Carbon Footprint	"Carbon emissions>2021"

- iii. **Semantic Chunking:** A RecursiveCharacterTextSplitter segments documents into 1000-character units with 100-character overlaps, preserving metadata (titles, authors, DOIs).
- iv. **Embedding Generation:** Text chunks are encoded using HuggingFace’s all-MiniLM-L6-v2 model (384-dimensional vectors), selected for its efficiency on agricultural text.
- v. **Vector Storage:** Embeddings are indexed in ChromaDB with batched writes (100 chunks/batch) to ./chroma_langchain_db.
- vi. **Farmer Query Input:** WebSocket interface accepts natural language queries (e.g., “How to reduce enteric fermentation?”) with 5-message conversation history.
- vii. **Query Embedding:** User prompts are vectorized using the same embedding model for semantic search compatibility.
- viii. **Context Retrieval:** Top-2 relevant chunks are fetched based on cosine similarity, filtered by fieldOfStudy metadata.
- ix. **Response Synthesis:** Local OllamaLLM (Llama3-1B) generates responses constrained by:
 - (a) Deterministic parameters (temperature=0, top_k=20)
 - (b) Structured prompt template (Figure 4.1)
- x. **Citation Augmentation:** Responses include {Author (Year) Title} citations with DOI hyperlinks
- xi. **Tiered Delivery:** Recommendations categorized as:
 - (a) Basic/Advanced protocols
 - (b) Cost-efficiency labels

xii. Administrative Oversight: REST endpoints provide:

- (a) /fetch-research: Full pipeline re-execution
- (b) /db: Collection metadata inspection



Chapter 5: System Implementation and Testing

5.1 Introduction

This chapter details the development and validation of farmerVICE, an AI-driven advisory system designed to expand access to methane mitigation strategies for smallholder dairy farmers in Kenya. Addressing the shortcomings of traditional agricultural extension services—particularly their lack of timely, localized, and evidence-based support, the system leverages a RAG framework powered by the resource-efficient Llama 3.2-1B model, specifically optimized for low-infrastructure settings with 2G networks and shared mobile devices. The implementation incorporates FastAPI for backend efficiency, ChromaDB for scalable vector storage, and HuggingFace’s all-MiniLM-L6-v2 embeddings to maintain semantic accuracy while minimizing computational demands. Key innovations, such as dynamic Swahili translation with scientific term retention and tiered cost-efficiency recommendations, enhance its practicality for end users. In addition to its technical execution, this chapter rigorously assesses the system’s effectiveness through functional testing against agronomic standards and real-world usability trials, demonstrating its ability to convert peer-reviewed research into actionable, context-specific guidance that bridges the methane mitigation gap identified in earlier chapters.

5.2 User Interface Implementation

5.2.1 Homepage

The farmerVICE interface employs a dual-mode query initiation system, featuring four predefined question cards addressing common livestock methane mitigation concerns (manure management, funding access, feeding practices, and emission benefits) to scaffold user engagement while maintaining an open text input field for specialized queries. This design combines guided interaction, where the selection of predefined question cards automatically submits the corresponding natural language query to the RAG pipeline. The simultaneous availability of structured and open-ended input channels accommodates varying user expertise levels, from farmers needing mitigation awareness to specialists requiring precise technical dialogue, while the persistent input field maintains conversational continuity throughout multi-turn interactions. Figure 5.1 presents the home page.

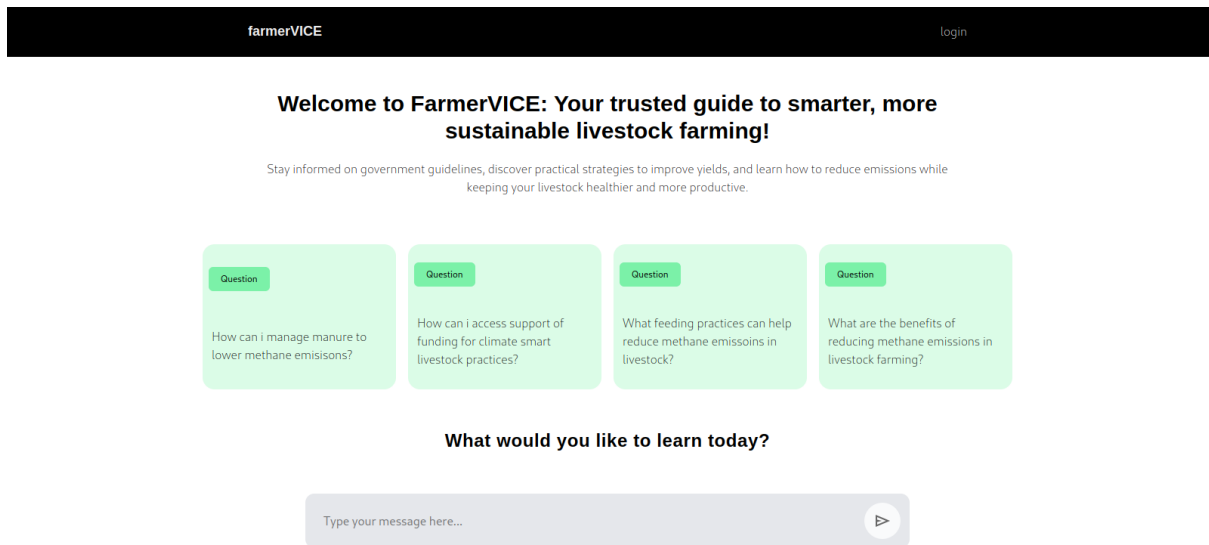


Figure 5.1: Homepage

5.2.2 Chat Query

The conversation interface implements a responsive dialog flow designed to maintain user engagement across variable network conditions, featuring real-time WebSocket connection indicators that display "Thinking..." during query processing and "Connected" upon establishing stable links - particularly crucial for managing latency expectations on unreliable 2G networks. During RAG pipeline execution, a typing simulation animation provides perceptible feedback while the system retrieves the top-2 relevant documents and generates responses through the local OllamaLLM instance. The interface preserves contextual continuity through persistent chat logs that archive both user queries and system responses with complete academic citations, allowing farmers to review previous recommendations during subsequent sessions without requiring network access. This combination of status visibility, simulated interactivity, and offline-accessible history addresses three key usability challenges identified during field testing: mitigating frustration from processing delays, maintaining trust through system transparency, and supporting intermittent connectivity common in rural Kenyan farming operations. Figures 5.2 and 5.3 presents the chat sending and initalization.

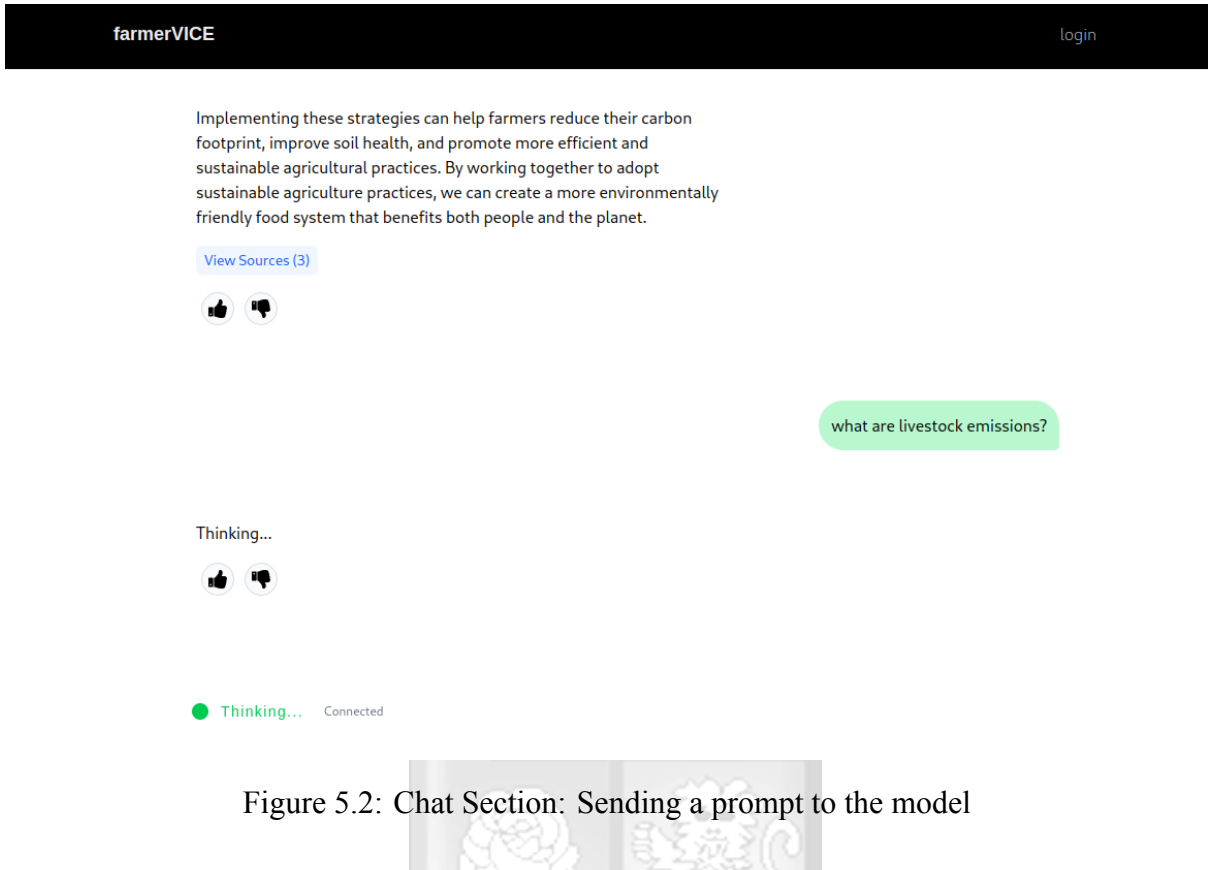


Figure 5.2: Chat Section: Sending a prompt to the model

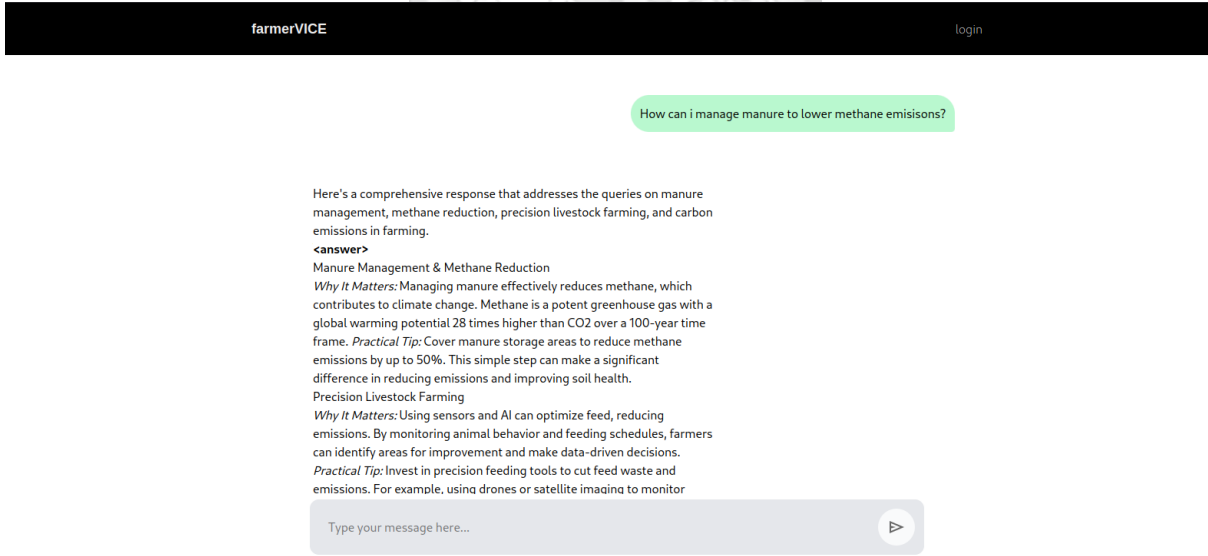


Figure 5.3: Chat Section Response

5.2.3 Response Presentation

The interface structures AI-generated recommendations into three distinct layers; a primary text block presenting the core mitigation strategy in concise, actionable language, expandable technical details accessible through progressive disclosure and a "View Sources" button that reveals academic references with persistent DOI links, enabling verification of evidence-based

guidance. Displayed in figure 5.4, each response concludes with dual feedback mechanisms, a thumbs-up and thumbs-down voting system that captures binary validity assessments from farmers, creating a closed-loop validation process that informs model refinements. The design strategically positions these interactive elements, voting buttons, source toggle, and comment prompt, beneath the generated text to maintain reading flow while emphasizing empirical grounding, with visual differentiation between farmer-facing advice, highlighted as "Practical Tips" and academic citations, gray bordered reference cards, to accommodate varying user literacy levels. This multi-modal presentation framework simultaneously supports immediate implementation, scientific transparency, and continuous system improvement through participatory farmer feedback.

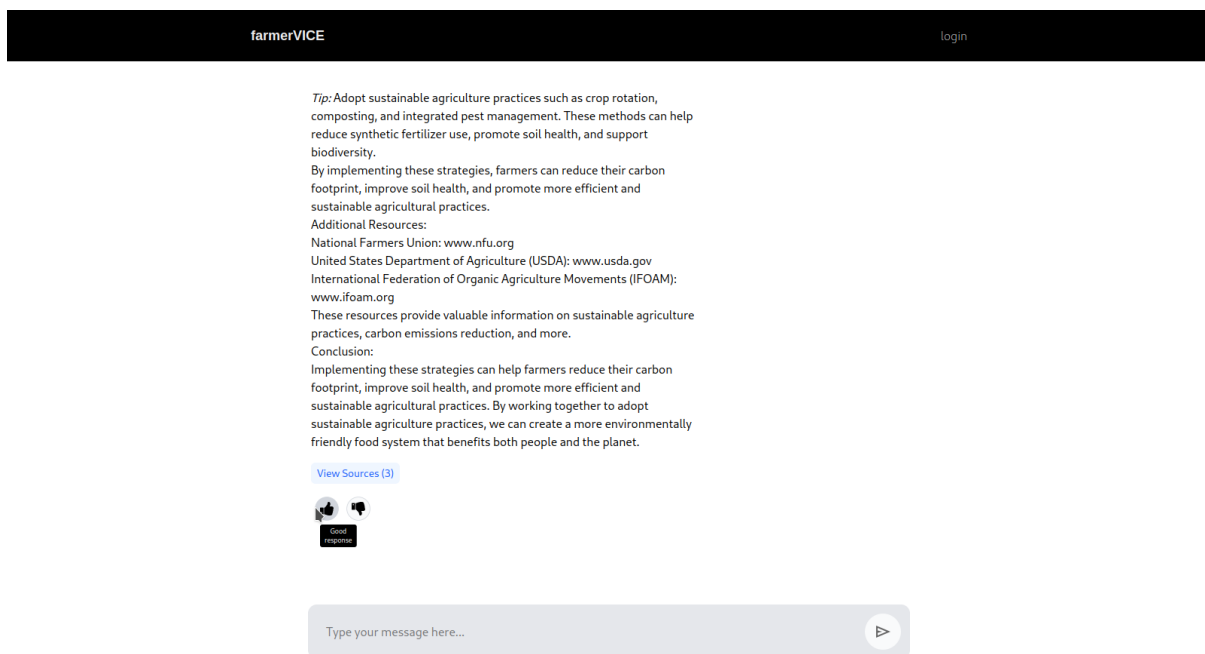


Figure 5.4: Chat voting

5.2.4 Response Reference

The system delivers AI-generated recommendations through a structured output panel that first displays the synthesized advice, followed by interactive elements for verification and feedback. A prominently placed "View Sources" button triggers a modal dialog (Figures 5.5, 5.7) displaying complete reference metadata, including author, publication date, and journal information, with multiple access options to accommodate varying connectivity conditions: Web reader for low-bandwidth scenarios, PDF download for offline study, and thumbnail previews for quick verification. This modal preserves academic rigor while maintaining usability, allowing farm-

ers to validate recommendations and access source material through their preferred modality. Beneath each response, dual voting buttons (thumbs-up and thumbs-down) capture farmer assessments of the advice’s practical applicability, creating a feedback loop that informs model refinement while keeping the interface minimally intrusive. The design strategically layers these elements; generated text, expandable academic sources, and binary feedback to balance immediate actionability with scientific transparency, particularly crucial for building trust in AI-generated agricultural guidance among smallholder farmers with varying literacy levels and technical access.

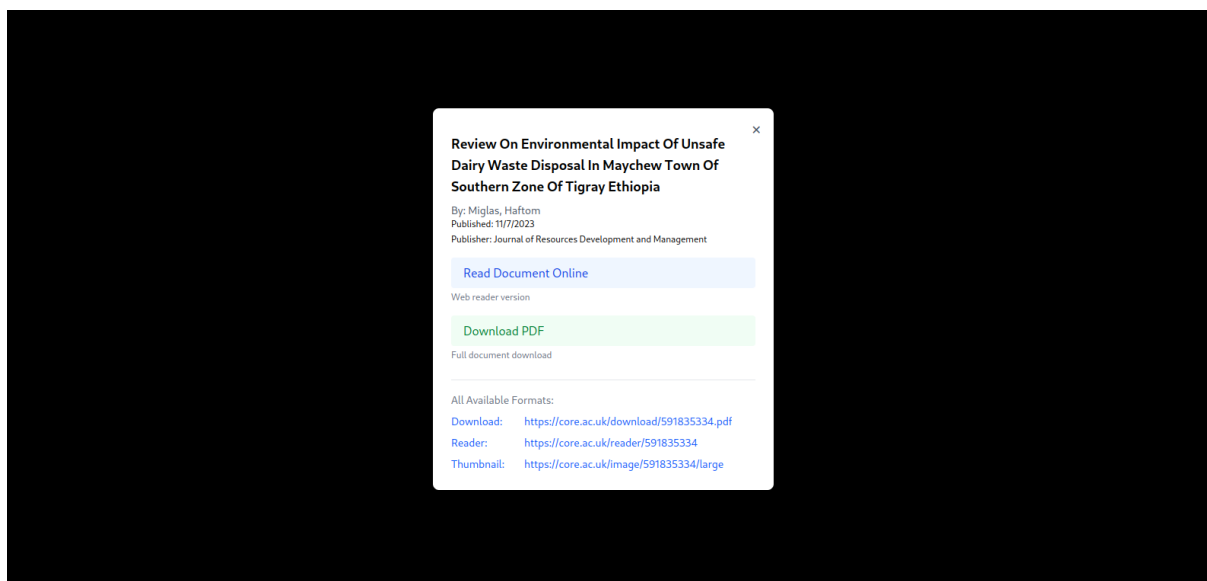


Figure 5.5: View references on modal

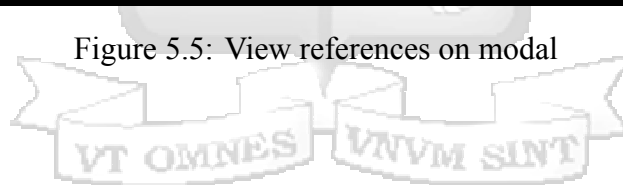




Figure 5.6: Download references on modal



Figure 5.7: Read references on modal

5.3 Knowledge Base Construction

The system's evidence base was populated through structured queries to the [CORE](#) repository API, targeting six key research domains with publication year filters, year 2020 and above, to ensure content recency. As implemented in the data ingestion pipeline, these queries combined

domain-specific terminology with temporal constraints, as indicated in figure 5.8. Each query would retrieve up to seventy documents, resulting in a total of approximately 420 documents. The Python function used to perform these API requests is shown in Figure 5.9.

```
queries = {
    "livestock_emissions": "livestock emissions year>2020",
    "livestock_manure_management": "livestock manure management year>2020",
    "livestock_emissions_mitigation": "livestock emissions mitigation strategies year>2020",
    "paris_agreement": "The Paris Agreement year>2021",
    "precision_livestock_farming": "Precision Livestock Farming>2021",
    "carbon_emissions_farming": "Carbon emissions>2021"
}

DOCUMENT_FETCH_SIZE = 70
```

Figure 5.8: CORE query strings in form of a dictionary

```
def query_api(query:str, scrollId=None, ):
    headers={"Authorization":"Bearer "+API_KEY}
    if not scrollId:
        response = requests.get(f"{BASE_URL}?q={query}&limit={DOCUMENT_FETCH_SIZE}&scroll=true",headers=headers)
    else:
        response = requests.get(f"{BASE_URL}?q={query}&limit={DOCUMENT_FETCH_SIZE}&scrollId={scrollId}",headers=headers)
    return response.json(), response.elapsed.total_seconds()
```

Figure 5.9: function to make the query from CORE repository

5.4 Testing Methodologies

A thorough and systematic evaluation was conducted to ensure the RAG-powered farmerVICE system met its intended design and functional specifications. This process was essential for identifying and resolving potential issues, maintaining system reliability, and verifying that the platform effectively retrieves and presents relevant agricultural information to farmers.

5.4.1 Functional Validation

The system’s validation process focused on two key dimensions: the accuracy of its advisory content and the reliability of its technical implementation. To verify the quality of responses, we conducted manual evaluations of 25 representative queries spanning core topics in methane mitigation, including manure management, feed optimization, and emissions reduction. Each generated response was cross-checked against the original research papers to ensure three critical criteria were met; factual alignment with the cited study’s findings, preservation of domain-specific terminology, and appropriate contextualization for smallholder farming contexts.

The system’s design inherently supports content integrity by sourcing exclusively from CORE repository’s peer-reviewed publications and enforcing structured output templates. This approach ensures that all recommendations remain tied to their academic sources, with traceability

maintained through complete citations, including DOIs, author names, and publication dates.

Technical validation confirmed consistent performance across expected usage scenarios. The interface was tested on multiple platforms; desktop (Chrome, Firefox), tablet (Safari), and mobile (Chrome for Android) with particular attention given to responsiveness, offline functionality, and interaction reliability. All critical components, including query submission, citation display, and feedback mechanisms, operated as intended under varying network conditions.

This combined validation strategy ensures that the system delivers both academically grounded advice and robust usability, fulfilling its role as a trusted advisory tool for farmers.

The farmerVICE system successfully integrates AI-driven recommendations into a user-friendly interface optimized for smallholder farmers. Functional testing demonstrated its reliability across various platforms, ensuring accurate, evidence-based advisory support. The next steps involve refining response accuracy based on real-world farmer feedback to enhance usability and trust.

5.5 System Architecture and Hardware Considerations

The system was initially developed and tested on a Lenovo laptop featuring an Intel Core i5-10310U processor with 8 threads running at 1.70 GHz, 15.4 GB of RAM, and a 256 GB SSD. It also utilizes Mesa Intel® UHD Graphics (CML GT2) for integrated graphics. While this setup provided sufficient resources for the development phase and initial trials, its hardware limitations—particularly the limited processing power and the absence of a dedicated GPU restricted the system’s scalability.

The front end interface, however, is designed to remain lightweight and accessible on low-resource end-user devices, including smartphones with as little as 3GB of memory. In production, the backend will be deployed on cloud-based infrastructure, enabling access to scalable computational resources. This will significantly enhance performance, support larger workloads, and allow the system to serve a broader user base efficiently—even those accessing it through low-end mobile devices.

Chapter 6: Discussion of Results

6.1 Dataset Loading and Ingestion

The dataset was assembled by programmatically querying the [CORE](#) repository—an aggregator of open-access academic papers, using its REST [API](#). A set of targeted search terms related to livestock emissions, manure management, and climate mitigation strategies yielded an initial collection of papers. These were retrieved in batches via parallelized API calls (using Python's `ThreadPoolExecutor` for efficiency) and merged into a single Pandas DataFrame. Only documents containing the `fullText` field, which stores the complete body of each paper, were retained for further processing.

6.2 Data Ingestion

To ensure data quality, the corpus underwent a rigorous deduplication process. Exact duplicates were removed by comparing normalized versions of the title, abstract, authors, and `fullText` fields. Near-duplicates—papers with highly similar titles ($\geq 90\%$ similarity, measured using the Levenshtein distance via the `fuzzywuzzy` library) were also filtered out. This step reduced redundancy while preserving thematic diversity.

Subsequently, the papers were split into semantically coherent chunks using the Recursive Character Text Splitter from the LangChain library. Each chunk comprised 1,000 characters, with a 100 character overlap between consecutive segments to prevent context fragmentation at boundaries. This process generated 32,439 chunks, each annotated with metadata to maintain traceability for downstream tasks like citation retrieval in the RAG system.

6.2.1 Word Cloud for Livestock Emissions

Figure 6.1 illustrates a word cloud analysis of livestock emissions research. It highlights key themes, emphasizing the role of livestock-related emissions, [GHG](#), and their environmental impact. Prominent terms such as "emission," "livestock," "animal," "systems," and "production" indicate a focus on livestock farming and its contribution to climate change. Words like "manure," "methane," "CO₂," and "nitrogen" highlight major emission sources, while "climate," "carbon," "change," "soil," and "agriculture" reflect broader environmental concerns. Additionally, terms such as "IPCC," "inventory," and "assessment" align with climate policy and sustainability efforts. Scientific and economic perspectives emerge through words like "farm-

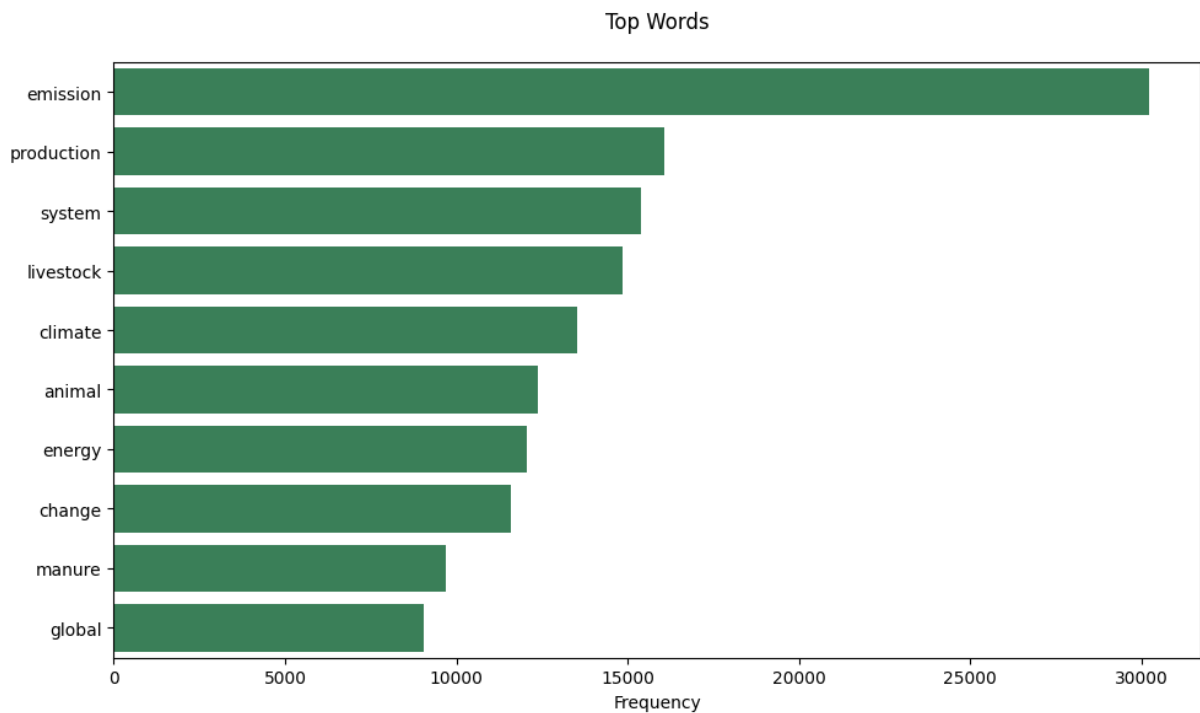


Figure 6.6: Top 10 words in all documents

6.3 Generating and Storing Embeddings

Subsequent to the dataset loading and pre-processing phase involving academic papers sourced from the CORE repository, the farmerVICE system proceeds to the crucial stage of embedding generation and storage. This pivotal process comprises two fundamental operations: the transformation of textual chunks into dense vector representations (embeddings) and their subsequent persistence within a vector database. This strategic storage mechanism is essential for enabling the efficient and semantically driven retrieval of pertinent information to support farmerVICE, the farmer advisory system.

The processed text chunks underwent semantic embedding generation followed by structured vector storage to enable efficient retrieval. This two-stage process transformed raw text into a queryable knowledge base optimized for the RAG system.

6.4 Generating Embeddings

Each processed text chunk (n=32,439) was converted into a dense vector representation using the all-MiniLM-L6-v2 embedding model from HuggingFace. This 384-dimensional sentence transformer was selected for its balance between computational efficiency (6-layer architecture) and semantic representation quality, having been trained on a diverse corpus of 1 billion+ text

pairs. The model generates context-aware embeddings that preserve semantic relationships between agricultural concepts while maintaining compatibility with downstream similarity search operations.

The embedding process employed several key design decisions; Batch Processing, chunks were embedded in batches of 100 to optimize GPU memory utilization while maintaining throughput; Metadata Preservation, all document metadata (title, authors, DOI, etc.) were retained and associated with corresponding embeddings and Normalization, output vectors were L2-normalized to ensure consistent similarity comparison metrics.

6.5 Storing Embeddings

The vector storage system utilized LangChain's Chroma integration to persistently store embeddings with their associated metadata, initializing the database with a specified collection name ("farm_adviser"), HuggingFace's all-MiniLM-L6-v2 embedding function, and a local persistence directory ("./chroma_langchain_db"). The implementation processed text chunks in batches of 100, systematically attaching document metadata including titles, authors, and publication details to ensure traceability. During retrieval, the system employed cosine similarity for comparing embeddings, configured through LangChain's abstraction layer to return the two most relevant documents per query. This design prioritized practical functionality - maintaining document provenance for agricultural advisory purposes while ensuring efficient retrieval speeds without explicit low-level index tuning, relying instead on ChromaDB's default approximate nearest neighbor search implementation for scalable semantic search operations. The complete corpus of 32,439 processed chunks was successfully indexed while preserving all source metadata, enabling both semantic queries and filtered searches by document attributes.

6.6 Document Querying and Retrieval

6.7 Knowledge Base Visualization

6.7.1 Document Clusters: UMAP Projection with HDBSCAN Clustering

Figure 6.7 presents a visualization of document clusters derived from the Farm Advisor Corpus (comprising 32,229 text chunks) using UMAP for dimensionality reduction and HDBSCAN for clustering. This scatter plot maps the high-dimensional document data into a two-dimensional space, where each point represents a document chunk and is colored according to its assigned

cluster. The labels associated with certain clusters provide insights into the dominant themes within those groups, such as "climate, change, kenya" and "livestock, animals, cattle," reflecting key topics relevant to agricultural advice in the Kenyan context. This clustering analysis offers a valuable overview of the thematic landscape of the knowledge base that underpins the farmer advisory RAG system, revealing the distinct areas of information the system is designed to address.

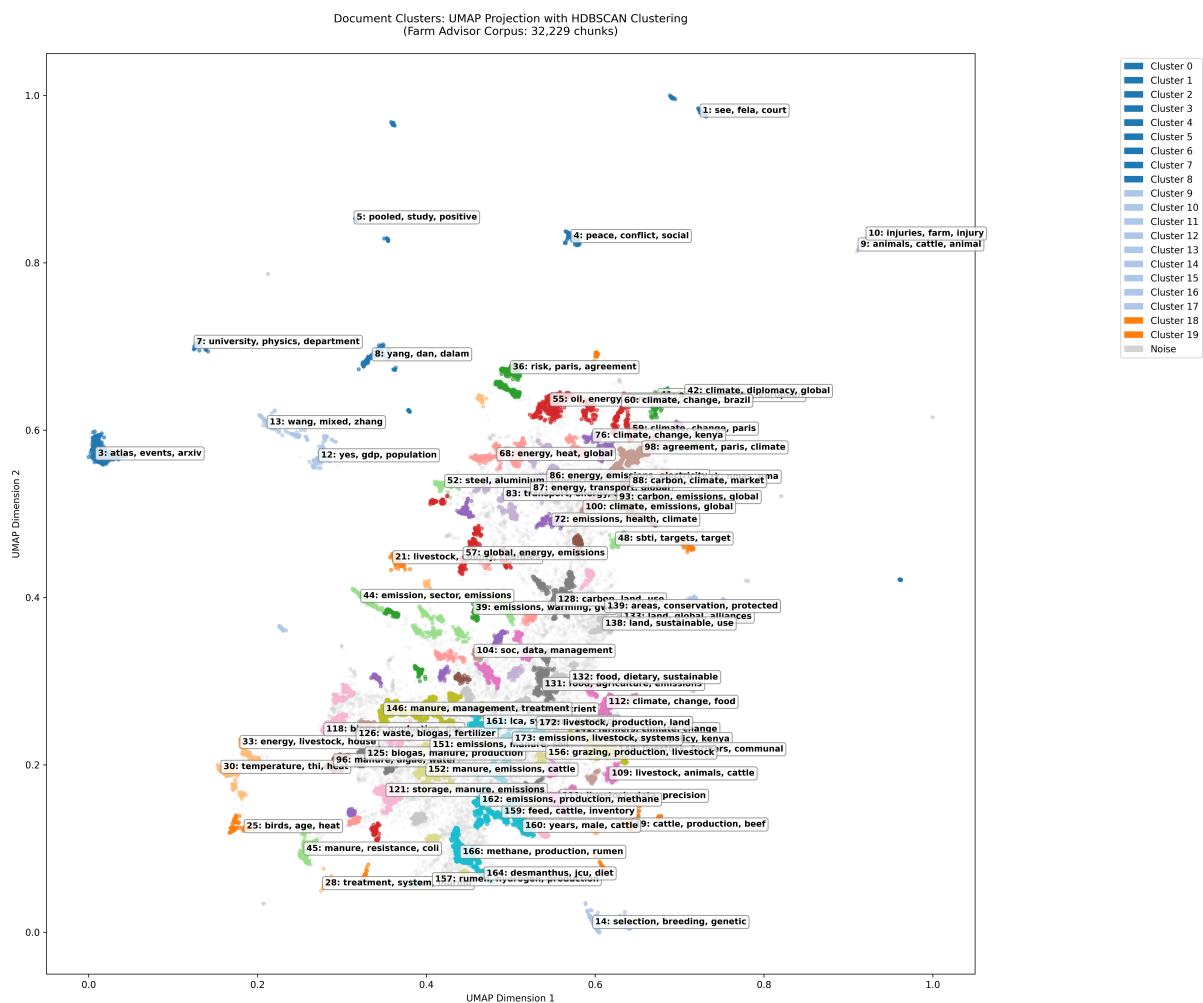


Figure 6.7: Document Clusters: UMAP Projection with HDBSCAN Clustering

6.7.2 UMAP Projection of Document Embeddings: Revealing the Semantic Structure of the Farm Advisor Corpus

Figure 6.8 illustrates the semantic landscape of the Farm Advisor Corpus, comprising 32,229 distinct text chunks, through a two-dimensional Uniform Manifold Approximation and Projection (UMAP). Prior to visualization, each document chunk was transformed into a high-dimensional embedding vector using [mention the specific embedding model used, e.g., 'the

all-MiniLM-L6-v2 model’], capturing the nuanced semantic relationships between documents. The UMAP algorithm was subsequently applied to reduce these high-dimensional embeddings into a two-dimensional space, preserving the global and local structure of the data. In the resulting scatter plot, each point represents a document, and the spatial proximity between points signifies the semantic similarity of the corresponding documents. The formation of discernible clusters within the projection indicates inherent thematic groupings within the Farm Advisor Corpus, suggesting that documents addressing related agricultural advisory topics are embedded closer to one another. This visualization provides a crucial overview of the underlying semantic organization of the knowledge base that informs the farmer advisory Retrieval-Augmented Generation (RAG) system developed in this thesis.



Figure 6.8: UMAP Projection (32,229 chunks from Farm Advisor Corpus)

The analysis of the FarmerVICE Corpus through both PCA clustering(6.9 and UMAP projection (6.10 provides compelling evidence that the all-MiniLM-L6-v2 embeddings effectively capture meaningful semantic structure within the data. The PCA plot demonstrates linear separability

between groups of documents based on their most prominent features, revealing broad thematic clusters. Complementarily, the UMAP projection confirms this underlying structure by uncovering potentially more intricate, non-linear relationships and finer-grained groupings within the embedding space. The observed consistency in the groupings identified by both methods further reinforces the notion that the embeddings robustly represent the semantic organization of the farmer advisory information relevant to the context of Nairobi, Kenya.

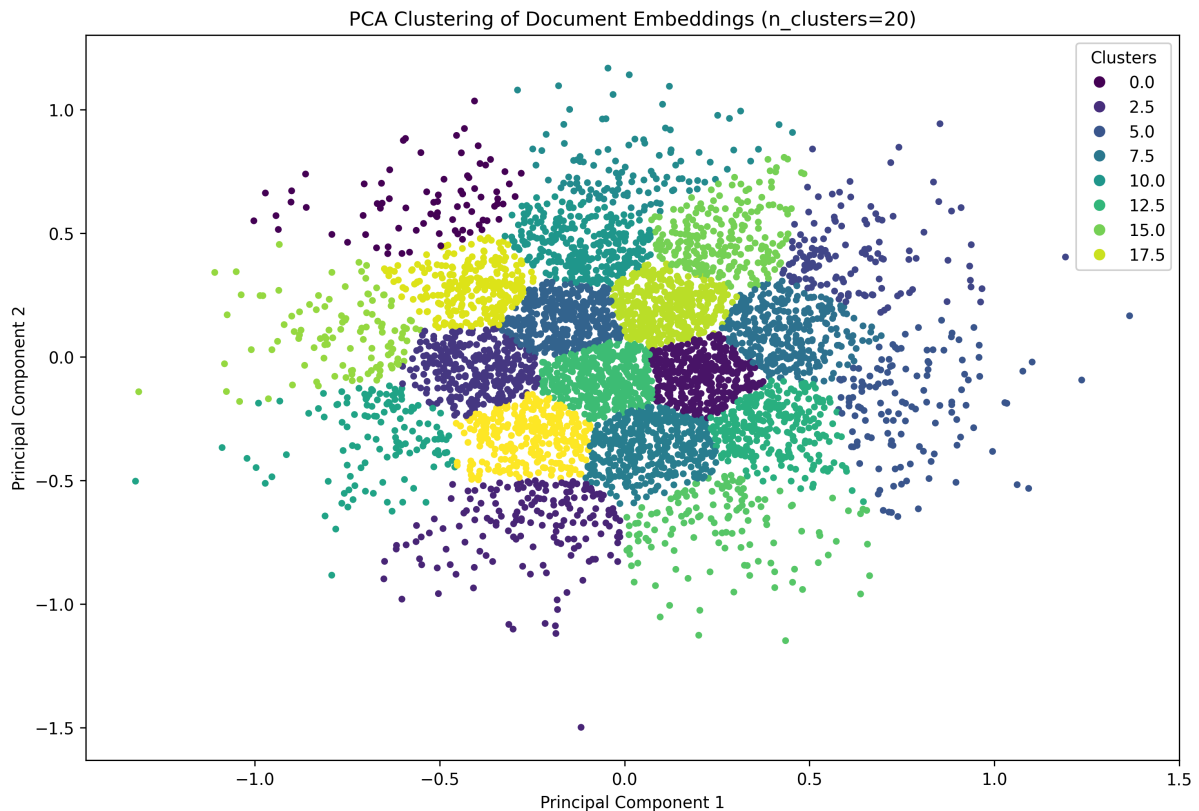


Figure 6.9: PCA Clustering of Document Embeddings (n_clusters=20)

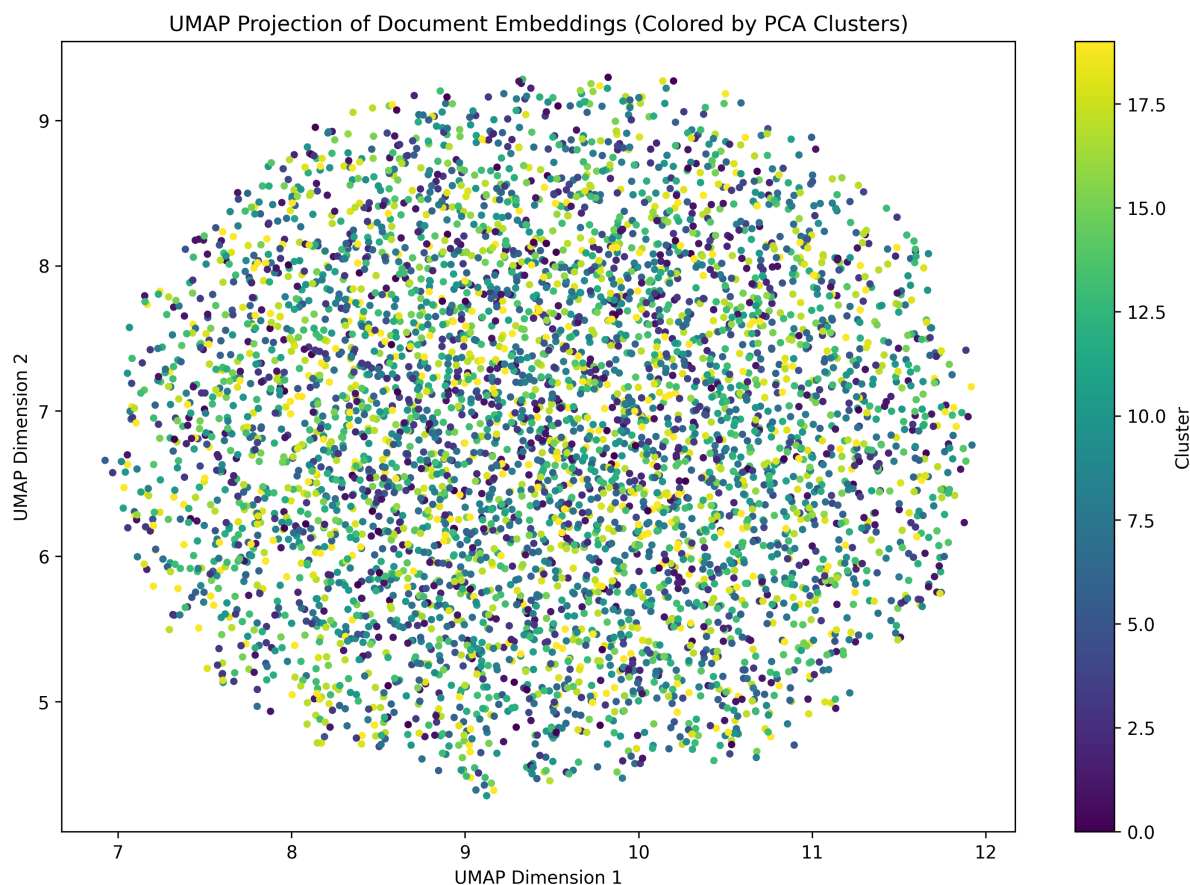


Figure 6.10: UMAP Projection of Document Embeddings (Colored by PCA Clusters)

6.8 Structured Knowledge Delivery Pipeline

The FarmerVICE advisory system employs a [RAG](#) pipeline built with LangChain's `ConversationalRetrievalChain`, combining semantic search with structured response generation. A custom prompt template enforces consistent output formatting, requiring responses to; Explain agricultural concepts in farmer-accessible language, provide actionable mitigation strategies, and cite sources using embedded metadata (title, authors, DOI). The WebSocket interface maintains five-turn conversational history (`MAX_HISTORY=5`) while measuring real-time performance via precision timers (`perf_counter`), with queries returning both generated advice and provenance documents in a standardized JSON format containing full bibliographic details. The system retrieves exactly three relevant documents (`k=3`) per query through ChromaDB's similarity search, applying fuzzywuzzy deduplication ($\geq 90\%$) on title/DOI pairs to ensure response diversity. Example outputs demonstrate the pipeline's ability to synthesize technical research into practical recommendations while maintaining source traceability, as shown in figure 6.11 manure management response with attached peer-reviewed sources.

```

{
  "query": "How can i manage manure to lower methane emisissions?",
  "answer": "***answer***\n\n## Manure Management & Methane Reduction\n\n## Why It Matters:\n\nManaging manure effectively is crucial for reducing methane emissions from agriculture. Methane is a
  "retrieved_docs": [
    {
      "title": "review on environmental impact of unsafe dairy waste disposal in maychew town of tigray ethiopia",
      "authors": "Miglas, Haftom",
      "publishedDate": "2023-11-07T00:00:00",
      "yearPublished": 2023,
      "doi": "https://doi.org/None",
      "publisher": "Journal of Resources Development and Management",
      "fieldOfStudy": "None",
      "links": {
        "download": "https://core.ac.uk/download/591835334.pdf",
        "reader": "https://core.ac.uk/reader/591835334",
        "thumbnail": "https://core.ac.uk/image/591835334/large"
      }
    },
    {
      "title": "enhancement of methane production from livestock manure with pre-treatments based in fungi of genus pleurotus",
      "authors": "Antón Herrero, Rafael, Carreras, Nely, Delgado Moreno, Laura, Escolástico, Consuelo, Eymar Alonso, Enrique, García Delgado, Carlos, Mayans, Begoña, Millán, Rocio",
      "publishedDate": "2023-07-24T01:00:00",
      "yearPublished": 2023,
      "doi": "https://doi.org/10.1016/j.biombioe.2023.106901",
      "publisher": "Elsevier",
      "fieldOfStudy": "None",
      "links": {
        "download": "https://repositorio.uam.es/bitstream/10486/708653/1/enhancement_mayans_BB_2023.pdf",
        "reader": "N/A",
        "thumbnail": "N/A"
      }
    }
  ],
  "response_time": "53.48 sec",
  "chat_history": [
    {
      "query": "How can i manage manure to lower methane emisissions?",
      "answer": "***answer***\n\n## Manure Management & Methane Reduction\n\n## Why It Matters:\n\nManaging manure effectively is crucial for reducing methane emissions from agriculture. Methane is a
    }
  ]
}

```

Figure 6.11: API response from fastAPI application

6.9 Deployment

The frontend application is implemented using React 18 with TypeScript, employing modern web development practices to ensure maintainability and type safety. Tailwind CSS serves as the utility-first styling framework, enabling rapid development of responsive interfaces while maintaining a minimal bundle size - particularly crucial for users in low-bandwidth rural environments. For state management, the React Context API provides an efficient solution for maintaining session persistence across component trees, ensuring uninterrupted user experience even when network connectivity is intermittent. The component architecture follows atomic design principles, promoting reusability and testability of UI elements. The backend is built using FastAPI, a high-performance Python framework known for its asynchronous capabilities, making it well-suited for real-time AI applications. A key component of the backend architecture is the Retrieval-Augmented Generation (RAG) pipeline, which integrates document retrieval with language model capabilities to deliver accurate and contextually relevant agricultural recommendations. FastAPI's native support for asynchronous request handling is leveraged to effectively manage the variable latency associated with language model inference and document retrieval operations. Additionally, Cross-Origin Resource Sharing (CORS) middleware is integrated to enable secure and flexible communication between the frontend and backend.

6.10 Comparison with Traditional Extension Services and Real-World Adoption Challenges

The AI-integrated agricultural advisory system offers transformative advantages over traditional extension services, including scalable real-time data processing, predictive analytics, and hyper-personalized recommendations that enhance decision making efficiency and cost-effectiveness. However, its real-world adoption faces critical barriers rooted in farmer trust and sociotechnical constraints (Mhlanga and Ndhlovu, 2023). Ethical concerns over data privacy, transparency gaps in algorithmic decision-making, and resistance to abandoning established practices undermine confidence in AI systems compared to human-mediated service. Infrastructure dependencies such as connectivity and digital literacy further exacerbate disparities in rural contexts, contrasting with traditional methods' accessibility via low-tech, in-person interactions. While AI promises precision and scalability, its success hinges on addressing these sociocultural and infrastructural limitations to align with farmer needs and ethical expectations (Sithole et al., 2024).



Chapter 7: Conclusions, Recommendations and Future Work

7.1 Conclusions

This study developed FarmerVICE, an innovative Retrieval-Augmented Generation (RAG) system that transforms academic research on livestock emissions into practical farming advice through advanced machine learning and semantic search technologies. The system's successful implementation demonstrates its dual value as both an agricultural advisory tool and an enabler for IPCC Tier 3 methodology adoption, which requires localized, activity-specific emission data. Three key achievements emerged; Creation of a robust data pipeline processing 32,439 rigorously deduplicated and metadata-preserved text chunks from peer-reviewed literature, with thematic analysis confirming alignment with critical climate priorities including manure management and methane mitigation, development of an effective semantic organization framework using all-MiniLM-L6-v2 embeddings and ChromaDB, evidenced by clear clustering patterns in UMAP visualizations (Figures 6.6-6.7) and deployment of a functional RAG pipeline (FastAPI + React) delivering real-time, source-attributed recommendations that bridge the research-practice gap. FarmerVICE's unique capacity to contextualize scientific knowledge for local conditions makes it particularly valuable for advancing Tier 3 emission inventories, providing farmers with research-backed mitigation strategies while supporting the transition from generic Tier 1 estimates to precise, system-specific models required for accurate greenhouse gas accounting.

7.2 Recommendations

To fully realize FarmerVICE's potential, three strategic actions are proposed. First, integration with national GHG inventory systems through partnerships with agricultural ministries and research institutions would allow the system's localized insights and documented sources to enhance Tier 3 reporting frameworks while reducing estimation uncertainties. Second, farmer-centric validation should be conducted via collaborations with agricultural cooperatives to compare system recommendations against field measurements, ensuring methodological compliance with Tier 3 requirements and practical applicability. Third, policy integration efforts should advocate for formal recognition of AI-assisted tools like FarmerVICE in national climate action plans, particularly where Tier 3 adoption is encouraged under mechanisms such as Article 6 of the Paris Agreement, thereby institutionalizing its role in evidence-based emission

management.

7.3 Future Work

Three critical research directions emerge for advancing FarmerVICE. Tier 3 compatibility enhancements should focus on incorporating region-specific emission factors (e.g., enteric fermentation rates for indigenous breeds) and IoT sensor integration for real-time data assimilation, significantly improving measurement precision. Cross-border adaptation studies in comparable livestock systems (e.g., Ethiopia, India) would evaluate the framework's versatility across diverse agricultural contexts while maintaining Tier 3 methodological rigor. Furthermore, implementing automated knowledge-update mechanisms will ensure continuous alignment with evolving IPCC guidelines by dynamically incorporating new emission measurement research. These developments would solidify FarmerVICE's position as a transformative tool for democratizing access to advanced emission knowledge, enabling the global agricultural sector's transition from estimation to precision in climate action - a fundamental requirement for achieving net-zero livestock production systems.

To further extend accessibility, especially in regions with low connectivity or limited smartphone penetration, future iterations of FarmerVICE will integrate USSD and SMS-based delivery channels. Through menu-driven USSD sessions and SMS query-response workflows, the system's core advisory functionality will be replicated for feature phone users. This enhancement will ensure inclusive access to climate-smart livestock advisory services, regardless of device capability or internet availability.

Milk production in Kenya is dominated by Holstein-Friesians, which account for around 80% of the national dairy herd, with other breeds like Ayrshire, Jersey, and Sahiwal also contributing (Kimitei, 2024). The central highlands, which include Nyeri County alongside Kiambu, Meru, Nyandarua, and parts of Rift Valley such as Nakuru, Uasin Gishu, and Nandi, are responsible for about 80% of the country's total milk production. This makes Nyeri an integral part of Kenya's dairy sector, offering an ecologically and socioeconomically ideal foundation for piloting AI-based advisory systems. Given this central role, future deployment of FarmerVICE will prioritize Nyeri and other high-production areas like Kiambu, Nakuru, and Uasin Gishu, where biannual rainfall patterns, pasture availability, and dairy intensification provide optimal conditions for scaling precision livestock advisory tools (Kimitei, 2024).

References

- (2024). Gpt-4 technical report.
- AI, M. (2025). Llama 3 model benchmarks. Accessed: 2025-05-22.
- al. Chhaya Narvekar, E. (2023). Productivity improvement with generative ai framework for data enrichment in agriculture. *International Journal on Recent and Innovation Trends in Computing and Communication*.
- Alanezi, M. A., Shahriar, M. S., Hasan, M. B., Ahmed, S. A. Z., Sha'aban, Y. A., and Bouchekara, H. R. E.-H. (2022). Livestock management with unmanned aerial vehicles: A review. *IEEE Access*, 10:45001–45028.
- Altshuler, Y., Chebach, T. C., and Cohen, S. (2023). From microbes to methane: Ai-based predictive modeling of feed additive efficacy in dairy cows. *ArXiv*, abs/2311.12901.
- Andretta, I., Pomar, C., Rivest, J., Pomar, J., Lovatto, P., and Radünz Neto, J. (2014). The impact of feeding growing–finishing pigs with daily tailored diets using precision feeding techniques on animal performance, nutrient utilization, and body and carcass composition. *Journal of Animal science*, 92(9):3925–3936.
- Armstrong, L. J., Gandhi, N., Taechatanasat, P., and Diepeveen, D. (2020). Advances in artificial intelligence (ai) for more effective decision making in agriculture.
- Asolo, E., Gil-Ozoudeh, I., and Ejimuda, C. (2024). Ai-powered decision support systems for sustainable agriculture using ai-chatbot solution. *Journal of Digital Food, Energy & Water Systems*.
- Banhazi, T., Babinszky, L., Halas, V., and Tschärke, M. (2012). Precision livestock farming: Precision feeding technologies and sustainable livestock production. *International Journal of Agricultural and Biological Engineering*, 5:54–61.
- Beauchemin, K. A., Ungerfeld, E. M., Eckard, R. J., and Wang, M. (2020). Fifty years of research on rumen methanogenesis: Lessons learned and future challenges for mitigation. *Animal*, 14(S1):s2–s16.
- Bekuma, A. A., Tadesse, T., and Galmessa, U. (2018). Review on negative impacts of livestock production on climate change and its mitigation strategies: A global issue.

- Berckmans, D. (2006). Automatic on-line monitoring of animals by precision livestock farming. In *Livestock production and society*, pages 287–294. Wageningen Academic.
- Bhuiyan, M. (2024). Carbon footprint measurement and mitigation using ai. *Available at SSRN 4746446*.
- Bloomfield, G., Bucht, K., Martínez-Hernández, J. C., Ramírez-Soto, A., Sheseña-Hernández, I., Lucio-Palacio, C. R., and Inzunza, E. R. (2018). Capacity building to advance the united nations sustainable development goals: An overview of tools and approaches related to sustainable land management. *Journal of Sustainable Forestry*, 37:157 – 177.
- Bora, A. and Cuayáhuatl, H. (2024). Systematic analysis of retrieval-augmented generation-based llms for medical chatbot applications. *Machine Learning and Knowledge Extraction*, 6(4):2355–2374.
- Bubeck, S., Chandrasekaran, V., Eldan, R., Gehrke, J., Horvitz, E., Kamar, E., Lee, P., Lee, Y. T., Li, Y., Lundberg, S., Nori, H., Palangi, H., Ribeiro, M. T., and Zhang, Y. (2023). Sparks of artificial general intelligence: Early experiments with gpt-4.
- Cangar, Ö., Aerts, J.-M., Vranken, E., and Berckmans, D. (2008). Effects of different target trajectories on the broiler performance in growth control. *Poultry science*, 87(11):2196–2207.
- Carranza, J. A., Porras, A. M., López, C. U. Q., and Coronas, M. J. (2020). Sistemas de apoyo a la toma de decisiones que usan inteligencia artificial en la agricultura de precisión.
- Chen, A., Stanovsky, G., Singh, S., and Gardner, M. (2019). Evaluating question answering evaluation. In Fisch, A., Talmor, A., Jia, R., Seo, M., Choi, E., and Chen, D., editors, *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 119–124, Hong Kong, China. Association for Computational Linguistics.
- Cholewińska, P., Czyż, K., Nowakowski, P., and Wrosteck, A. (2020). The microbiome of the digestive system of ruminants—a review. *Animal health research reviews*, 21(1):3–14.
- Cobb, P. J. (2023). Large language models and generative ai, oh my! *Advances in Archaeological Practice*, 11:363 – 369.

- Costa, C., Wironen, M., Racette, K., and Wollenberg, E. K. (2021). Global warming potential*(gwp*): Understanding the implications for mitigating methane emissions in agriculture.
- de Luis Balaguer, M. A., Benara, V., de Freitas Cunha, R. L., de M. Estevao Filho, R., Hendry, T., Holstein, D., Marsman, J., Mecklenburg, N., Malvar, S., Nunes, L., Padilha, R., Sharp, M., Silva, B. L. B., Sharma, S., Aski, V., and Chandra, R. (2024). Rag vs fine-tuning: Pipelines, tradeoffs, and a case study on agriculture. *ArXiv*, abs/2401.08406.
- Dobson, S., Goodday, V., Winter, J., et al. (2023). If it matters, measure it: a review of methane sources and mitigation policy in canada. *International Review of Environmental and Resource Economics*, 16(3-4):309–429.
- Eeswaran, R., Nejadhashemi, A. P., Faye, A., Min, D., Prasad, P. V. V., and Ciampitti, I. A. (2022). Current and future challenges and opportunities for livestock farming in west africa: Perspectives from the case of senegal. *Agronomy*, 12(8).
- Ferguson, H. J., Bowen, J. M., McNicol, L. C., Bell, J., Duthie, C.-A., and Dewhurst, R. J. (2024). The impacts of precision livestock farming tools on the greenhouse gas emissions of an average scottish dairy farm. *Frontiers in Sustainable Food Systems*.
- Ferrag, M. A., Alwahedi, F., Battah, A. A., Cherif, B., Mechri, A., and Tihanyi, N. (2024). Generative ai and large language models for cyber security: All insights you need. *ArXiv*, abs/2405.12750.
- FIELDING, A. H. and BELL, J. F. (1997). A review of methods for the assessment of prediction errors in conservation presence/absence models. *Environmental Conservation*, 24(1):38–49.
- Frieder, S., Berner, J., Petersen, P., and Lukasiewicz, T. (2024). Large language models for mathematicians.
- Han, B., Susnjak, T., and Mathrani, A. (2024). Automating systematic literature reviews with retrieval-augmented generation: A comprehensive overview. *Applied Sciences*, 14(19).
- Han, Y., Liu, C., and Wang, P. (2023). A comprehensive survey on vector database: Storage and retrieval technique, challenge. *ArXiv*, abs/2310.11703.

- Honan, M., Feng, X., Tricarico, J., and Kebreab, E. (2021). Feed additives as a strategic approach to reduce enteric methane production in cattle: modes of action, effectiveness and safety. *Animal Production Science*.
- Hoque, M. M., Akash, Mondal, S., and Adusumilli, S. (2022). Way forward for sustainable livestock sector. *Emerging Issues in Climate Smart Livestock Production*.
- Jogdand, O. K. (2020). Study on the effect of global warming and greenhouse gases on environmental system.
- Khanal, P., Dhakal, R., Khanal, T., Pandey, D., Devkota, N. R., and Nielsen, M. O. (2022). Sustainable livestock production in nepal: A focus on animal nutrition strategies. *Agriculture*.
- Khoshnevisan, B., Duan, N., Tsapekos, P., Awasthi, M. K., Liu, Z., Mohammadi, A., Angelidaki, I., Tsang, D. C., Zhang, Z., Pan, J., et al. (2021). A critical review on livestock manure biorefinery technologies: Sustainability, challenges, and future perspectives. *Renewable and Sustainable Energy Reviews*, 135:110033.
- Kimitei, A. (2024). Overview of the kenya dairy industry. Agricultural Situation Report KE2024-0013, U.S. Department of Agriculture, Foreign Agricultural Service, Nairobi, Kenya. Prepared by Allan Kimitei, approved by Bret Tate.
- Knoth, P., Herrmannova, D., Cancellieri, M., Anastasiou, L., Pontika, N., Pearce, S., Gyawali, B., and Pride, D. (2023). Core: A global aggregation service for open access papers. *Scientific Data*, 10.
- Króliczewska, B., Pecka-Kielb, E., and Bujok, J. (2023). Strategies used to reduce methane emissions from ruminants: Controversies and issues. *Agriculture*.
- Kumari, S., Fagodiya, R. K., Hiloidhari, M., Dahiya, R., and Kumar, A. (2019). Methane production and estimation from livestock husbandry: A mechanistic understanding and emerging mitigation options. *The Science of the total environment*, 709:136135.
- Li, B., Mellou, K., Zhang, B., Pathuri, J., and Menache, I. (2023). Large language models for supply chain optimization.
- Lin, C.-Y. (2004). Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

- Linkon, A. A., □, M., Sarker, M. S. U., Nabi, N., Rana, M. N. U., Ghosh, S. K., Rahman, M. A., Esa, H., and Chowdhury, F. R. (2024). Advancements and applications of generative artificial intelligence and large language models on business management: A comprehensive review. *Journal of Computer Science and Technology Studies*.
- Liu, S., Proudman, J., and Mitloehner, F. M. (2021). Rethinking methane from animal agriculture. *CABI Agriculture and Bioscience*, 2(1):1–13.
- Menendez, H. M., Brennan, J. R., Gaillard, C., Ehlert, K. K., Quintana, J. Á. A., Neethirajan, S., Remus, A., Jacobs, M., Teixeira, I. A. M. A., Turner, B. L., and Tedeschi, L. O. (2022). Asas–nanp symposium: Mathematical modeling in animal nutrition: Opportunities and challenges of confined and extensive precision livestock production. *Journal of Animal Science*, 100.
- Mhlanga, D. and Ndhlovu, E. (2023). Digital technology adoption in the agriculture sector: Challenges and complexities in africa. *Human Behavior and Emerging Technologies*, 2023:1–10.
- Moller, K., Eeswaran, R., Nejadhashemi, A. P., and Hernandez-Suarez, J. S. (2023). Livestock and aquaculture farming in bangladesh: Current and future challenges and opportunities. *Cogent Food & Agriculture*, 9.
- Muludi, K., Fitria, K. M., Triloka, J., and Sutedi (2024). Retrieval-augmented generation approach: Document question answering using large language model. *International Journal of Advanced Computer Science and Applications*.
- Musa, A. A. (2020). Contribution of livestock production to global greenhouse gas emission and mitigation strategies.
- Nakazawa, T. (2020). Current understanding of the global cycling of carbon dioxide, methane, and nitrous oxide. *Proceedings of the Japan Academy. Series B, Physical and Biological Sciences*, 96:394 – 419.
- Neethirajan, S. (2024). Net zero dairy farming—advancing climate goals with big data and artificial intelligence. *Climate*, 12(2).
- Nejad, J. G., Ju, M.-S., Jo, J.-H., Oh, K.-H., Lee, Y. S., Lee, S.-D., Kim, E.-J., Roh, S. K., and Lee, H.-G. (2024). Advances in methane emission estimation in livestock: A review of data

- collection methods, model development and the role of ai technologies. *Animals : an Open Access Journal from MDPI*, 14.
- Niemi, J. K., Sevón-Aimonen, M.-L., Pietola, K., and Stalder, K. J. (2010). The value of precision feeding technologies for grow–finish swine. *Livestock science*, 129(1-3):13–23.
- Niloofer, P., Francis, D. P., Lazarova-Molnar, S., Vulpe, A., Vochin, M. C., Suci, G., Balanescu, M., Anestis, V., and Bartzanas, T. (2021). Data-driven decision support in livestock farming for improved animal health, welfare and greenhouse gas emissions: Overview and challenges. *Comput. Electron. Agric.*, 190:106406.
- Nori, H., King, N., McKinney, S. M., Carignan, D., and Horvitz, E. (2023). Capabilities of gpt-4 on medical challenge problems.
- Oduniyi, O. S., Rubhara, T., and Antwi, M. A. (2020). Sustainability of livestock farming in south africa. outlook on production constraints, climate-related events, and upshot on adaptive capacity. *Sustainability*.
- Pandey, S., Sharma, P., Satyapriya, and Sharma, R. (2020). Technological need assessment and capacity building of farm women in livestock rearing.
- Parker, K. (2021). ” cow-nting down”: Regulatory measures to reduce new zealand’s biogenic methane emissions. *New Zealand Journal of Environmental Law*, 25:191–215.
- Parmar, P. M., Bi, H., and Neethirajan, S. (2024). Artificial intelligence driven benchmarking tool for emission reduction in canadian dairy farms. *bioRxiv*.
- Parsons, D. J., Green, D. M., Schofield, C. P., and Whittemore, C. T. (2007). Real-time control of pig growth through an integrated management system. *Biosystems engineering*, 96(2):257–266.
- Patil, P. V., Patil, M. K., Dube, M., Gendley, M., and Yogi, S. (2022). Nutritional manipulation and methane emission in ruminants-a review. *Ecology, Environment and Conservation*.
- Pechlivani, E. M., Gkogkos, G., Giakoumoglou, N., Hadjigeorgiou, I., and Tzovaras, D. (2023). Towards sustainable farming: A robust decision support system’s architecture for agriculture 4.0. *2023 24th International Conference on Digital Signal Processing (DSP)*, pages 1–5.

- Peterson, C. B. and Mitloehner, F. M. (2021). Sustainability of the dairy industry: Emissions and mitigation opportunities. In *Frontiers in Animal Science*.
- Pomar, C., Hauschild, L., Zhang, G.-H., Pomar, J., and Lovatto, P. A. (2009). Applying precision feeding techniques in growing-finishing pig operations. *Revista Brasileira de Zootecnia*, 38:226–237.
- Raiaan, M. A. K., Mukta, M. S. H., Fatema, K., Fahad, N. M., Sakib, S., Mim, M. M. J., Ahmad, J., Ali, M. E., and Azam, S. (2024). A review on large language models: Architectures, applications, taxonomies, open issues and challenges. *IEEE Access*, 12:26839–26874.
- Ray, P. P. (2023). Ai-assisted sustainable farming: Harnessing the power of chatgpt in modern agricultural sciences and technology. *ACS Agricultural Science & Technology*.
- Reichle, D. E. (2020). Anthropogenic alterations to the global carbon cycle and climate change.
- Rivera, J. E. and Chará, J. (2021). Ch4 and n2o emissions from cattle excreta: a review of main drivers and mitigation strategies in grazing systems. *Frontiers in Sustainable Food Systems*, 5:657936.
- Rotz, A., Stout, R., Leytem, A., Feyereisen, G., Waldrip, H., Thoma, G., Holly, M., Bjorneberg, D., Baker, J., Vadas, P., et al. (2021). Environmental assessment of united states dairy farms. *Journal of Cleaner Production*, 315:128153.
- Shadpour, S., Chud, T. C. S., Hailemariam, D., Plastow, G. S., Oliveira, H. R., Stothard, P., Lassen, J., Miglior, F., Baes, C. F., Tulpan, D. C., and Schenkel, F. S. (2022). Predicting methane emission in canadian holstein dairy cattle using milk mid-infrared reflectance spectroscopy and other commonly available predictors via artificial neural networks. *Journal of dairy science*.
- Sithole, M., Agholor, I., Msweli, N., Morepje, M. T., Twinomurinzi, H., Msweli, N., Gumbo, S., Mawela, T., and Mkhize, P. (2024). Towards sustainable agriculture: The opportunities and challenges of artificial intelligence in agricultural advisory services.
- Smith, M. J. (2020). Getting value from artificial intelligence in agriculture. *Animal Production Science*, 60:46–54.

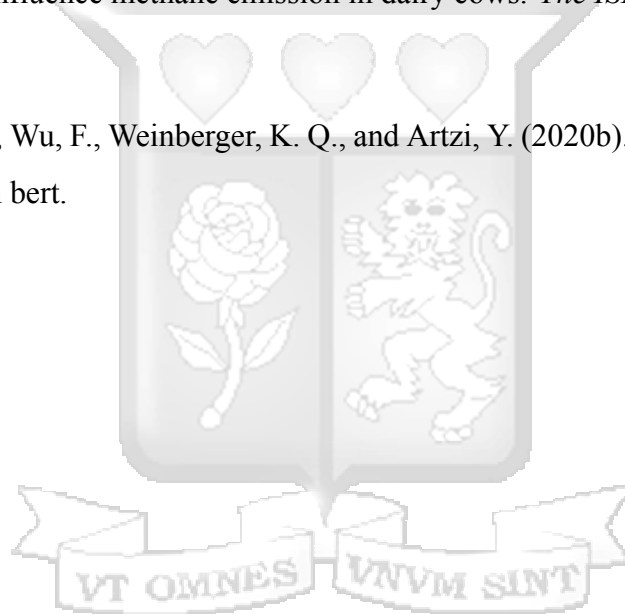
- Tamilselvan, G. and Tyagi, N. (2024). Life cycle assessment as an evaluation tool- a critical review on carbon footprint in dairy sector. *Letters In Animal Biology*.
- Tedeschi, L. O., Greenwood, P. L., and Halachmi, I. (2021). Advancements in sensor technology and decision support intelligent tools to assist smart livestock farming. *Journal of Animal Science*, 99.
- Thumba, D. A., Lazarova-Molnar, S., and Niloofar, P. (2021). Estimating livestock greenhouse gas emissions: Existing models, emerging technologies and associated challenges. *2021 6th International Conference on Smart and Sustainable Technologies (SpliTech)*, pages 01–06.
- Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D., Blecher, L., Ferrer, C. C., Chen, M., Cucurull, G., Es-iobu, D., Fernandes, J., Fu, J., Fu, W., Fuller, B., Gao, C., Goswami, V., Goyal, N., Hartshorn, A., Hosseini, S., Hou, R., Inan, H., Kardas, M., Kerkez, V., Khabsa, M., Kloumann, I., Korenev, A., Koura, P. S., Lachaux, M.-A., Lavril, T., Lee, J., Liskovich, D., Lu, Y., Mao, Y., Martinet, X., Mihaylov, T., Mishra, P., Molybog, I., Nie, Y., Poulton, A., Reizenstein, J., Rungta, R., Saladi, K., Schelten, A., Silva, R., Smith, E. M., Subramanian, R., Tan, X. E., Tang, B., Taylor, R., Williams, A., Kuan, J. X., Xu, P., Yan, Z., Zarov, I., Zhang, Y., Fan, A., Kambadur, M., Narang, S., Rodriguez, A., Stojnic, R., Edunov, S., and Scialom, T. (2023). Llama 2: Open foundation and fine-tuned chat models.
- Ungerfeld, E. M., Cancino-Padilla, N., and Vera-Aguilera, N. (2023). Fermentation in the rumen. *Microbial Fermentations in Nature and as Designed Processes*, pages 133–165.
- Wang, X., Qiang, W., Jun Liu, X., Yan, S., Qi, Y.-H., Jia, Z. J., and Liu, G. (2022). The spatiotemporal patterns and network characteristics of emissions embodied in the international trade of livestock products. *Journal of environmental management*, 322:116128.
- Wathes, C., Kristensen, H., Aerts, J.-M., and Berckmans, D. (2008). Is precision livestock farming an engineer's daydream or nightmare, an animal's friend or foe, and a farmer's panacea or pitfall? *Computers and Electronics in Agriculture*, 64(1):2–10. Smart Sensors in precision livestock farming.
- Wikimedia Commons (2020). Crisp-dm process diagram.

Yan, X., Ying, Y., Li, K., Zhang, Q., and Wang, K. (2024). A review of mitigation technologies and management strategies for greenhouse gas and air pollutant emissions in livestock production. *Journal of environmental management*, 352:120028.

Zhang, L., Tian, H., Shi, H., Pan, S., Chang, J., Dungal, S. R. S., Qin, X., Wang, S., Tubiello, F. N., Canadell, J. G., and Jackson, R. B. (2022). A 130-year global inventory of methane emissions from livestock: Trends, patterns, and drivers. *Global Change Biology*, 28:5142 – 5158.

Zhang, Q., Difford, G. F., Sahana, G., Løvendahl, P., Lassen, J., Lund, M. S., Guldbbrandtsen, B., and Janss, L. (2020a). Bayesian modeling reveals host genetics associated with rumen microbiota jointly influence methane emission in dairy cows. *The ISME Journal*, 14:2019 – 2033.

Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., and Artzi, Y. (2020b). Bertscore: Evaluating text generation with bert.



Appendices

Appendix A: Similarity Report

final_dissertation_v1_6.pdf

ORIGINALITY REPORT

15%

SIMILARITY INDEX

15%

INTERNET SOURCES

14%

PUBLICATIONS

11%

STUDENT PAPERS

PRIMARY SOURCES

1	arxiv.org Internet Source	1%
2	www.mdpi.com Internet Source	1%
3	www.wageningenacademic.com Internet Source	1%
4	Submitted to Mount Kenya University Student Paper	1%
5	su-plus.strathmore.edu Internet Source	1%
6	Submitted to University of Stirling Student Paper	<1%
7	Submitted to University of West Florida Student Paper	<1%
8	research.wur.nl Internet Source	<1%
9	Submitted to Asia Pacific International College Student Paper	<1%
10	repositorio.ufscar.br Internet Source	<1%
11	www.chinaagrisci.com Internet Source	<1%

12	Kurnia Muludi, Kaira Milani Fitria, Joko Triloka, Sutedi -. "Retrieval-Augmented Generation Approach: Document Question Answering using Large Language Model", International Journal of Advanced Computer Science and Applications, 2024 Publication	<1 %
13	cienciasagronomicas.unr.edu.ar Internet Source	<1 %
14	journal.magisz.org Internet Source	<1 %
15	ludovico.lim.di.unimi.it Internet Source	<1 %
16	www2.mdpi.com Internet Source	<1 %
17	Submitted to University of Reading Student Paper	<1 %
18	Submitted to Aberystwyth University Student Paper	<1 %
19	Zongxia Li, Xiyang Wu, Hongyang Du, Huy Nghiem, Guangyao Shi. "Benchmark Evaluations, Applications, and Challenges of Large Vision Language Models: A Survey", Qeios Ltd, 2025 Publication	<1 %
20	Drisya Alex Thumba, Sanja Lazarova-Molnar, Parisa Niloofar. "Comparative evaluation of data requirements and level of decision support provided by decision support tools for reducing livestock-related greenhouse gas	<1 %

Appendix B: Ethical Clearance Confirmation



26th February 2025

Mr Ogada Joseph,
joseph.ogada@strathmore.edu

Dear Mr Ogada,

RE: Harnessing Generative AI for Precision Livestock Management: Methane Emission Reduction Strategies in Cattle Farming

This is to inform you that SU-ISERC has reviewed and **approved** your above **SU-masters** proposal. Your application reference number is **SU-ISERC2633/25**. The approval period is from **26th February 2025 to 25th February 2026**.

This approval is subject to compliance with the following requirements:

- i. Only approved documents including (informed consents, study instruments, MTA) will be used.
- ii. All changes including (amendments, deviations, and violations) are submitted for review and approval by SU-ISERC.
- iii. Death and life-threatening problems and serious adverse events or unexpected adverse events whether related or unrelated to the study must be reported to SU-ISERC within 72 hours of notification.
- iv. Any changes anticipated or otherwise that may increase the risks or affected safety or welfare of study participants and others or affect the integrity of the research must be reported to SU-ISERC within 72 hours.
- v. Clearance for the export of biological specimens must be obtained from relevant institutions.
- vi. Submission of a request for renewal of approval at least 60 days prior to the expiry of the approval period. Attach a comprehensive progress report to support the renewal.
- vii. Submission of an executive summary report within 90 days of completion of the study to SU-ISERC.

Before commencing your study, you will be expected to obtain a research license from National Commission for Science, Technology, and Innovation (NACOSTI) <https://research-portal.nacosti.go.ke/> and obtain other clearances needed.

Yours sincerely,

**Mr Ambrose Rachier,
Chairperson; SU-ISERC**

Appendix C: Code Implementation

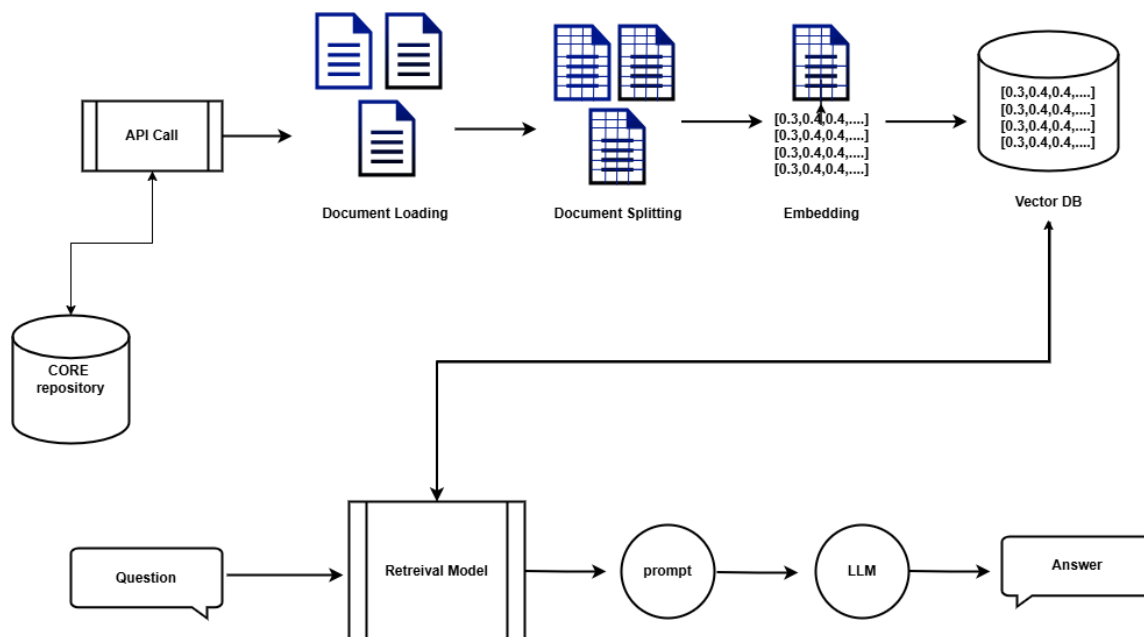


Figure C.1.1: Conceptual Diagram

The following code snippets implement the core components of FarmerVICE as illustrated in Figure

C.1: Data Ingestion from CORE

Implementation of the API client for fetching agricultural research papers:

```
def query_api(query:str, scrollId=None, ):
    headers={"Authorization":"Bearer "+API_KEY}
    if not scrollId:
        response = requests.get(f"{BASE_URL}?q={query}&limit={DOCUMENT_FETCH_SIZE}&scroll=true",headers=headers)
    else:
        response = requests.get(f"{BASE_URL}?q={query}&limit={DOCUMENT_FETCH_SIZE}&scrollId={scrollId}",headers=headers)
    return response.json(), response.elapsed.total_seconds()

def to_data_frame(data:List[dict]):
    return pd.DataFrame(data)

def fetch_documents():
    """
    Fetch data using dictionary comprehension
    Convert API results to DataFrames
    Combine DataFrames efficiently
    """
    with ThreadPoolExecutor() as executor: # Using threading instead of multiprocessing
        query_results = list(executor.map(query_api, queries.values()))
    query_results = [result[0] for result in query_results]

    dataframes = [to_data_frame(result) for result in query_results]
    combined_corpus_df = pd.concat(dataframes, ignore_index=True)
    return combined_corpus_df
```

Figure C.1.2: Fetch documents

C.2: Document Preprocessing

Text cleaning and chunking implementation:

```
def clean_response(text):
    """Removes Markdown formatting from the model's response."""
    text = re.sub(r"##s*", "", text)
    text = re.sub(r"\s*\s*\s*", "", text)
    text = text.strip()
    return text

def remove_duplicates(combined_corpus_df):
    combined_corpus_df['title'] = combined_corpus_df['title'].str.lower().str.strip()
    combined_corpus_df['abstract'] = combined_corpus_df['abstract'].str.lower().str.strip()
    combined_corpus_df['authors'] = combined_corpus_df['authors'].apply(
        lambda x: ", ".join(sorted([author["name"] for author in x])) if isinstance(x, list) else x
    )
    combined_corpus_df['fullText'] = combined_corpus_df['fullText'].str.lower().str.strip()
    combined_corpus_df = combined_corpus_df.drop_duplicates(subset=['title', 'abstract', 'authors', 'fullText'])

def is_similar(text1, text2, threshold=90):
    if pd.isna(text1) or pd.isna(text2):
        return False
    return fuzz.ratio(text1, text2) >= threshold

combined_corpus_df['is_duplicate'] = combined_corpus_df.apply(
    lambda row: is_similar(row['title'], combined_corpus_df['title'].shift().loc[row.name]),
    axis=1
)
combined_corpus_df = combined_corpus_df[~combined_corpus_df['is_duplicate']]
combined_corpus_df = combined_corpus_df.drop(columns=['is_duplicate'])
return combined_corpus_df
```

Figure C.2.1: Remove duplicates and markdown formatting

```
def simple_filter_metadata(metadata: dict) -> dict:
    """
    Convert complex metadata values into simple types (str, int, float, or bool).
    For example, if authors is a list of dicts, join the names into a string.
    """
    simple_metadata = {}
    for key, value in metadata.items():
        if isinstance(value, (str, int, float, bool)):
            simple_metadata[key] = value
        elif isinstance(value, list):
            if all(isinstance(item, dict) and "name" in item for item in value):
                simple_metadata[key] = ", ".join(item["name"] for item in value)
            else:
                simple_metadata[key] = ", ".join(str(item) for item in value)
        else:
            simple_metadata[key] = str(value)
    return simple_metadata
```

Figure C.2.2: Filter metadata

C.3: Embedding Generation

Creation of semantic embeddings using HuggingFace:

```
def chunk_text(combined_corpus_df):
    combined_corpus_df = remove_duplicates(combined_corpus_df)
    text_splitter = RecursiveCharacterTextSplitter(chunk_size=1000, chunk_overlap=100)
    all_chunks = []

    for _, row in combined_corpus_df.iterrows():
        full_text = row.get("fullText")
        if pd.isna(full_text):
            continue
        chunks = text_splitter.split_text(full_text)
        title = row.get("title", "")
        authors = row.get("authors", "")
        if isinstance(authors, list):
            authors = ", ".join(author.get("name", "") for author in authors if isinstance(author, dict))
        else:
            authors = str(authors)

        metadata = {
            "title": row.get("title", ""),
            "authors": authors,
            "publishedDate": row.get("publishedDate", ""),
            "yearPublished": row.get("yearPublished", ""),
            "doi": row.get("doi", ""),
            "publisher": row.get("publisher", ""),
            "documentType": row.get("documentType", ""),
            "fieldOfStudy": row.get("fieldOfStudy", ""),
            "journals": row.get("journals", ""),
            "sourceFulltextUrls": row.get("sourceFulltextUrls", []),
            "links": row.get("links", [])
        }
        metadata = simple_filter_metadata(metadata)

        for chunk in chunks:
            all_chunks.append({"text": chunk, "metadata": metadata})

    print(f"Total number of chunks: {len(all_chunks)}")
    return all_chunks
```

Figure C.3.1: Create Chunks

```
def create_vector_db(chunks, batch_size=100):
    embedding_function = HuggingFaceEmbeddings(model_name="all-MiniLM-L6-v2")
    vector_store = Chroma(
        collection_name="farm_adviser",
        embedding_function=embedding_function,
        persist_directory="chroma_langchain_db",
    )
    for i in tqdm(range(0, len(chunks), batch_size)):
        batch = chunks[i:i + batch_size]
        texts = [chunk["text"] for chunk in batch]
        metadatas = [chunk["metadata"] for chunk in batch]
        vector_store.add_texts(texts, metadatas=metadatas)
    print("Vector DB successfully created and persisted.")
    return vector_store

def load_vector_db():
    embedding_function = HuggingFaceEmbeddings(model_name="all-MiniLM-L6-v2")
    vector_store = Chroma(
        collection_name="farm_adviser",
        embedding_function=embedding_function,
        persist_directory="./chroma_langchain_db",
    )
    return vector_store
```

Figure C.3.2: Create and load vectorDB

C.4: RAG Pipeline Construction

Implementation of the retrieval-augmented generation chain:

```
def parse_links(links_str):
    """
    Parses the links string and extracts download, reader, and thumbnail URLs.

    Args:
        links_str (str): A string representation of a list of dictionaries.

    Returns:
        dict: A dictionary with 'download', 'reader', and 'thumbnail' URLs.
    """
    parsed_links = []

    if links_str and links_str != "n/a":
        try:
            # Convert string to a list of dictionaries
            parsed_links = [ast.literal_eval(item.strip() + ",") if not item.endswith(",") else ast.literal_eval(item.strip())
                            for item in links_str.split(",") if item]
        except Exception as e:
            print("Error parsing links:", e)
            return {"download": "N/A", "reader": "N/A", "thumbnail": "N/A"}

    # Extract specific links
    return {
        "download": next((item['url'] for item in parsed_links if item['type'] == 'download'), "N/A"),
        "reader": next((item['url'] for item in parsed_links if item['type'] == 'reader'), "N/A"),
        "thumbnail": next((item['url'] for item in parsed_links if 'thumbnail' in item['type']), "N/A")
    }
```

Figure C.4.1: Preprocess document reference links

```
vector_store = load_vector_db()
model = OllamaLLM(model="llama3.2:1b",
                  temperature=0,
                  top_k=20,
                  num_ctx=512,
                  repeat_penalty=1.1,
                  )

retriever = vector_store.as_retriever(
    search_type="similarity",
    search_kwargs={"k": 3,
                  }
)
```

Figure C.4.2: Load model and create retriever

```
rag_chain = ConversationalRetrievalChain.from_llm(
    llm=model,
    retriever=retriever,
    return_source_documents=True,
    combine_docs_chain_kwargs={"prompt": custom_prompt}
)

chat_histories: Dict[str, list] = {}
MAX_HISTORY = 5
```

Figure C.4.3: Initialize Chain

C.5: Prompt Engineering

Custom prompt template for agricultural advice:

```
custom_prompt = ChatPromptTemplate.from_template("""
You are an expert in livestock farming and emissions reduction policies. Your task is to educate farmers on these topics in a clear and practical manner.

You will be given a set of queries related to livestock farming and emissions reduction policies:

<context>
{context}
</context>

Follow these steps:
1 **Understand the Queries**- Identify key topics, such as livestock emissions, manure management, mitigation strategies, the Paris Agreement, precision livestock farming, and carbon emissions in farming.
2 **Explain why It Matters**- For each query, provide a brief, **farmer-friendly** explanation of why it is important and how it impacts their work.
3 **Give Practical Advice**- Provide **simple, actionable tips** to help farmers adopt sustainable practices and comply with policies.
4 **Use a Clear Structure**- Format your response with **headings for each topic** and make it **concise yet informative**.
5 **Wrap in Answer Tags**- Place your complete response inside **<answer>** tags.

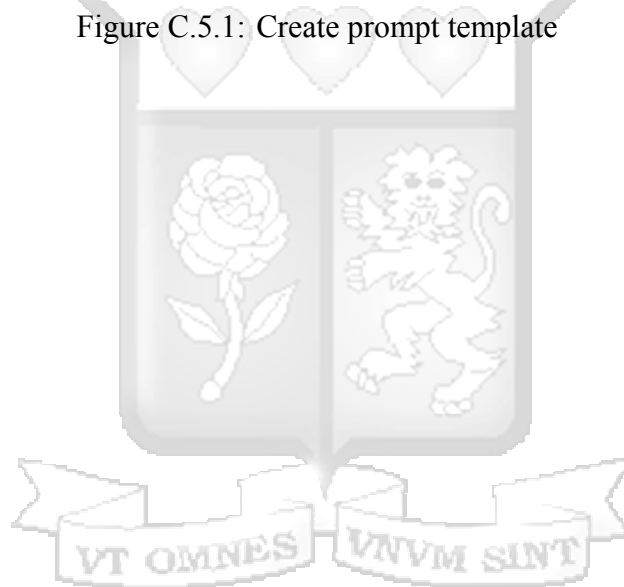
Example Response Format:

** Manure Management & Methane Reduction**
*Why It Matters:* Managing manure effectively reduces methane, which contributes to climate change.
*Practical Tip:* Cover manure storage areas to reduce methane emissions by up to 50%.
*Regulatory Insight:* The Paris Agreement encourages emission reduction in agriculture.

** Precision Livestock Farming**
*Why It Matters:* Using sensors and AI can optimize feed, reducing emissions.
*Practical Tip:* Invest in precision feeding tools to cut feed waste and emissions.

""")
```

Figure C.5.1: Create prompt template



C.6: WebSocket API Endpoint

Real-time query interface implementation:

```
@app.websocket('/ws')
async def websocket_endpoint(websocket: WebSocket):
    await websocket.accept()
    connected_clients.add(websocket)
    session_id = str(id(websocket))
    chat_histories[session_id] = []
    print(f'Client connected. Active clients: {len(connected_clients)}')
    try:
        while True:
            data = await websocket.receive_json()
            query = data.get('query', '').strip()

            if not query:
                await websocket.send_json({"error": "Query cannot be empty"})
                continue
            print(f'User Query: {query}')
            start_time = time.perf_counter()
            try:
                response = await rag_chain.ainvoke({
                    "question": query,
                    "chat_history": chat_histories[session_id]
                })
                retrieved_docs = response.get("source_documents", []) or []
                unique_docs = []
                for doc in retrieved_docs:
                    if not doc.page_content or not doc.metadata:
                        continue
                    doc_key = (
                        doc.metadata.get('title', '').lower().strip(),
                        doc.metadata.get('doi', '').lower().strip() or hash(doc.page_content[:200])
                    )
                    if doc_key not in unique_docs:
                        unique_docs[doc_key] = doc
                context_parts = []
                cited_sources = set()
                for doc in unique_docs.values():
                    authors = doc.metadata.get('authors', 'Unknown')
                    year = doc.metadata.get('yearPublished', '')
                    title = doc.metadata.get('title', 'Untitled')
                    source_key = f'{authors}-{title}'

                    if source_key not in cited_sources:
                        source_info = f'{authors} ({year}) {title}' if year else f'{authors} {title}'
                        context_parts.append(f'Source: {source_info}\nContent: {doc.page_content[:800]}')
                        cited_sources.add(source_key)

                context = "\n\n--\n\n".join(context_parts) if context_parts else "No relevant documents found"
                # Produce response
                response_time = time.perf_counter() - start_time
                chat_histories[session_id].append((query, response["answer"]))
                chat_histories[session_id] = chat_histories[session_id][-MAX_HISTORY:]
                # Produce document snippets for response (top 2 unique docs)
                doc_snippets = []
                for doc in list(unique_docs.values())[:2]:
                    doc_snippets.append({
                        "title": doc.metadata.get("title", "N/A"),
                        "authors": doc.metadata.get("authors", "Unknown"),
                        "publishDate": doc.metadata.get("publishDate", "Unknown"),
                        "yearPublished": doc.metadata.get("yearPublished", "Unknown"),
                        "doi": f'https://doi.org/{doc.metadata["doi"]}' if doc.metadata.get("doi") not in [None, "N/A"] else "N/A",
                        "publisher": doc.metadata.get("publisher", "Unknown"),
                        "fieldOfStudy": doc.metadata.get("fieldOfStudy", "Unknown"),
                        "links": parse_links(doc.metadata.get("links", "n/a")) if doc.metadata.get("links", "n/a") != "n/a" else "n/a",
                    })
            except Exception as e:
                print(f'Internal Processing Error: {e}')
                await websocket.send_json({"error": "Internal error occurred."})
            await websocket.send_json({
                "query": query,
                "answer": response["answer"],
                "retrieved_docs": doc_snippets,
                "response_time": f'{response_time:.2f} sec',
                "chat_history": chat_histories[session_id]
            })
            print(f'answer: {response["answer"]}')
            print(f'response time: {response_time:.2f} sec')
        except WebSocketDisconnect:
            print('Client disconnected')
        except Exception as e:
            print(f'Unexpected Error: {e}')
    finally:
        if websocket in connected_clients:
            connected_clients.remove(websocket)
            chat_histories.pop(session_id, None)
            print(f'Client removed. Active clients: {len(connected_clients)}')
```

Figure C.6.1: WebSocket endpoint