Research Proposal

Table of Contents

Protecting PII Data in LLM's by using Encoding/ introducing noise at client end

Abstract

This paper investigates privacy-preserving methods for client data protection in Large Language Model (LLM) inference. We examine existing approaches including the Split-N-Denoise (SnD) framework (Mai et al., 2024) cryptographic techniques such as homomorphic encryption(T. Chen et al., 2022) , and various perturbation methods to analyze how they balance utility and privacy, with particular focus on local differential privacy (LDP) applications for text data.

This paper proposes the **CleanSplit Model (CSM)**, a novel privacy-preserving inference framework that deploys only a lightweight token representation layer on the client side. The primary objective of CSM is to protect user privacy during LLM inference without requiring modifications to existing model parameters, thereby optimizing the critical trade-off between privacy preservation and computational utility.

The expected outcome is to demonstrate that our proposed approach significantly reduces computational overhead for removing personally identifiable information (PII) at the client side—where sensitive data resides—while maintaining LLM performance efficiency and result quality.

1. Background

The growing concern over data privacy in Large Language Models (LLMs) is reflected in recent governmental and regulatory interventions, including the EU's AI Act, the US Executive Order on Safe, Secure, and Trustworthy AI, and established regulations such as **GDPR and HIPAA**. These frameworks emphasize the necessity for AI systems to incorporate robust privacy-preserving mechanisms, making the development of practical and efficient solutions for local privacy protection in LLM inference paramount.

As per latest HIPPA report, healthcare-data-breach-report healthcare organizations experienced a 16.67% monthly increase in security incidents, while the volume of individuals affected by protected health information breaches or unauthorized disclosures surged by 302.71% compared to the previous month. Monthly, tens of millions of individuals' PII are compromised, with one survey from June 2025 noting over 7.6 million affected just that month. As per report(data-breach-statistics), cybercrime's worldwide economic impact is expected to hit $10.5 trillion by 2025, with costs escalating at an annual growth rate of 15 percent..

Current privacy-preserving approaches face significant limitations. **Cryptographic methods** such as **secure multi-party computation and homomorphic encryption (HE)** incur substantial computational overhead, making them impractical for real-time applications. Conversely, perturbation approaches often struggle to balance utility and privacy, frequently resulting in reduced model accuracy and performance degradation.

Another Architecture **Split-N-Denoise (SnD)** framework addresses some of these limitations by employing differential privacy algorithms that add mathematical noise to protect sensitive information. However, this approach still presents critical vulnerabilities: the added noise can inadvertently reveal personally identifiable information (PII) to LLM servers, and the framework lacks Named Entity Recognition (NER) capabilities, leading to potential PII leakage. These shortcomings are particularly concerning in high-stakes domains such as healthcare and finance, where complete PII removal is essential for regulatory compliance and user trust.

This research aims to fill the critical gap in privacy-preserving LLM inference by proposing a novel framework **CleanSplit Model (CSM)** that addresses the limitations of existing approaches. By focusing on applications in educational, healthcare, and financial sectors—where PII protection is

paramount—**this study seeks to enhance security measures by adding NER replacements at client side while maintaining computational efficiency and model utility**.

2. Literature Review

Intuitive approaches, such as anonymizing sensitive terms before LLM input, prove insufficient for comprehensive privacy protection. These methods fail to conceal other linguistic elements vital for semantic interpretation, thereby compromising both privacy and the effectiveness of tasks requiring precise semantic understanding. Additionally, existing denoising techniques face fundamental limitations when deployed server-side, as the server lacks knowledge of injected noise levels, creating an inherent conflict with privacy protection objectives.

Differential Privacy

Researchers including (Yang et al., 2022a) (Wei et al., 2020) (Yang et al., 2022b)(Yu et al., 2022) (Kerrigan et al., 2020) presented a generalized gradient perturbation split learning framework with demonstrable differential privacy assurances. Their approach ensures differential privacy protection extends to both training parameters and communication exchanges in distributed computing scenarios. In this algorithm, Stronger privacy (lower epsilon value) requires adding more noise, which can significantly reduce the accuracy and usefulness of the data. Also, Differential privacy works best with large datasets. When applied to small or sparse datasets, the added noise can distort results to the point where meaningful analysis becomes impossible. So this approach was not very suitable for chatbot applications.

Homographic Encryption Protection Analysis

Research by (Gilbert & Gilbert, 2024), (Bae et al., 2025), (Liu & Liu, 2023) advocated for homomorphic encryption (HE) as a data privacy protection method. This cryptographic approach enables direct computational operations on encrypted information without requiring prior decryption. Throughout the entire processing cycle, data remains encrypted, thereby minimizing exposure vulnerabilities. Major limitation with HE is it is extremely resource-intensive, Operations on encrypted data can be up to 1,000 times slower than plaintext computations. This makes real-time processing of PII (e.g., in healthcare or finance) impractical without specialized hardware or optimization.

Hide and Seek (HaS)

(Y. Chen et al., 2023) introduced an innovative framework designed for safeguarding prompt privacy within large language models. The HaS system incorporates dual core methodologies: concealing private entities through anonymization processes and retrieving private entities via de-anonymization techniques. Specifically, the HaS approach ensures privacy preservation by concealing sensitive entities within prompts using specialized anonymization components. The generative scheme employed a Bloomz model trained on data annotated by GPT-4, which makes the client side chatbots heavy as well as the required performance is not achieved.

Split-N-Denoise Framework Analysis

(Mai et al., 2024) (Vepakomma et al., 2018) (Singh et al., 2019)proposed a novel privacy-preserving inference framework that introduces a client-side denoising model. This approach leverages knowledge of raw inputs and noise levels to enhance embedding utility after noise injection. However, the framework exhibits critical limitations in PII protection due to its reliance on Differential Privacy (DP) techniques.

While DP effectively protects statistics or summaries derived from user data, it fails to provide adequate protection for individual data points. The mathematical noise added through DP can still allow inference of personally identifiable information, particularly when dealing with sensitive textual data in high-stakes domains. This limitation is especially problematic in healthcare and financial applications where complete PII removal is mandated by regulatory requirements.

Research Gap and Contribution

The literature reveals a significant gap in privacy-preserving methods that can effectively protect individual PII data while maintaining semantic context and LLM performance. Current approaches either sacrifice utility for privacy or fail to provide adequate protection for sensitive individual information.

This study addresses these limitations by proposing an enhanced architecture **CleanSplit Model (CSM)** that integrates Named Entity Recognition (NER) techniques to identify and replace PII data with contextually appropriate synthetic data. This approach aims to preserve conversational context while ensuring complete PII protection, making it particularly suitable for chatbot applications in healthcare and financial domains where customer privacy is paramount and regulatory compliance is essential.

3. Research Questions (If any)

This study addresses three fundamental research questions that are critical to developing an effective privacy-preserving LLM inference framework:

RQ1: Client-Side NER Implementation Efficiency

How can Named Entity Recognition (NER) objects be effectively extracted on the client side while minimizing computational overhead and performance impact in real-time chatbot applications? This question examines the trade-offs between deploying lightweight NER frameworks versus client-side LLM-based extraction methods, with particular focus on processing speed, resource consumption, and accuracy in identifying personally identifiable information.

RQ2: Impact of NER-Based PII Replacement on Model Performance

To what extent does the replacement of identified PII entities with synthetic alternatives affect the accuracy and quality of LLM outputs? This investigation will quantify the performance degradation, if any, when original sensitive data is substituted with contextually appropriate fake data, and identify optimal replacement strategies that maintain semantic coherence.

RQ3: Privacy-Context Balance in Conversational AI

How can conversational context be preserved while ensuring complete privacy protection in multi-turn dialogues with LLMs? This question explores methods for maintaining semantic continuity and contextual understanding across conversation sessions without compromising the anonymization of sensitive information, particularly in domains requiring strict privacy compliance such as healthcare and financial services.

These research questions collectively address the core challenge of developing a practical privacy-preserving framework that maintains both computational efficiency and conversational quality while ensuring robust protection of sensitive user data.

4. Aim and Objectives

Aim:

The intent of this paper is to develop a novel architectural framework for efficiently protecting personally identifiable information (PII) in Large Language Model inference while maintaining computational efficiency, semantic context, and model performance in privacy-sensitive applications.

Objectives:

Objective 1: Comprehensive Analysis of Existing Privacy-Preserving Methods

Systematically review current privacy-preserving techniques for LLM inference, including cryptographic approaches and differential privacy frameworks. Identify limitations, computational overhead, and privacy vulnerabilities, focusing on applicability to real-time chatbot applications in healthcare and financial domains.

Objective 2: Development of Client-Side PII Detection and Replacement Framework

Design and implement client-side Named Entity Recognition (NER) methodology for identifying and replacing PII data with contextually appropriate synthetic alternatives while preserving semantic meaning and conversational flow before server transmission.

Objective 3: Architecture Design and Performance Optimization

Propose a lightweight, scalable architectural pattern integrating PII protection mechanisms without compromising LLM inference efficiency. Optimize for client-side deployment, minimizing computational overhead while maintaining high accuracy in detection and performance.

5. Significance of the Study

Data breach costs hit a record peak of $4.88 million in 2024, representing a 10% rise from the previous year. Approximately 46% of security incidents compromised customer personally identifiable information, encompassing tax identification numbers, email addresses, telephone numbers, and residential addresses. In 2024, ~275 million healthcare records were breached in the U.S (2024-healthcare-data-breach-report).

Financial institutions face an average data breach cost of roughly $6.08 million per incident when personally identifiable information is compromised in 2025. Regulatory penalties for non-compliance may total up to 4% of worldwide annual revenue under GDPR, with other frameworks imposing thousands of dollars per individual violation. Approximately 50% of U.S. citizens have suffered personally identifiable information breaches within the past five years, as billions of data records face annual exposure. These financial calculations exclude difficult-to-measure consequences such as psychological trauma, credit rating deterioration, and missed opportunities stemming from PII misuse.

Proposed Architecture that could enhance the protection of PII data from client side chatbots whereby providing a more reliable and efficient way of preserving PII data. This would enhance the security and trust of LLM chatbots particularly amongst financial/ healthcare and education industry where protecting your PII data is utmost important and is mandated by regulators. This contributes in potentially reducing frauds by not leaking PII data over internet.

6. Scope of the Study

Propose Architecture only covers PII data(like Contact details, Demographic information, Personal identifiers etc) protection and efficiency but doesn't cover other type of secret data like

- PHI (Protected Health Information) - Medical conditions, prescriptions, diagnoses.
- PIB (Personally Identifiable Behaviour)- Browsing history, app usage, search queries.
- Secrets or Credentials- API keys, passwords, access tokens, SSH keys.
- Financial Information - Bank balances, investment data, transaction details.
- Intellectual Property (IP) - Source code, internal documentation, trade secrets.
- National Security or Classified Data - Defence-related terms, classified project names.

7. Research Methodology

This study employs a **mixed-methods research approach** that combines both quantitative and qualitative methodologies to comprehensively investigate.

Qualitative Component:

This component facilitates an in-depth examination of architectural design decisions and their influence on chatbot behaviour, offering insights into design rationales, trade-offs, and their downstream effects on system performance and user interaction.

Quantitative Component:

This component enables the precise measurement of performance metrics, including response accuracy, latency, computational efficiency, and the effectiveness of personally identifiable information (PII) removal.

It also supports benchmarking existing PII protection methods using standardized leakage metrics to assess and compare their robustness.

Calculating response accuracy for an LLM (Large Language Model) depends on the task type (e.g., classification, question answering, summarization)

- **For Classification or Structured Outputs**

If the LLM is generating answers from a fixed set (e.g., yes/no, A/B/C, intent labels, sentiment):

Accuracy = Number of correct predictions/ Total number of predictions.

- **For Extractive or Factual Question Answering**

When the LLM answers questions based on facts (e.g., "Who is the president of X?"):

Use **Exact Match (EM) and/or F1 score**.

Exact Match: Output exactly matches the ground truth answer.

F1 Score: Harmonic mean of precision and recall over overlapping tokens between the prediction and reference.

You can compute:

EM Accuracy = % of predictions that are an exact match.

F1 Score (avg) = Mean token overlap across samples.

- **For Safety or PII-Related Tasks**

If you define a correct response as "no PII leaked", then:

Response Accuracy = Number of responses with no PII leakage / Total responses

You can use PII detectors (e.g., Presidio, spaCy, LlamaGuard) to flag whether output contains PII.

Step-by-Step Guide to Measuring PII Removal Effectiveness

1. Define What Counts as PII

Common categories: names, phone numbers, emails, addresses, SSNs, IPs, usernames, etc.

Choose a taxonomy (e.g., NIST, GDPR, HIPAA) to standardize labels.

---

2. Choose or Build a PII Detection Tool

Use off-the-shelf detectors like:

- Microsoft Presidio
- spaCy + regex/custom NER
- AWS Comprehend / Google DLP
- Custom LLM-based classifier

These tools will tag any detected PII in both the original and sanitized outputs.

---

3. Calculate Key Metrics

✅ A. Precision, Recall, F1 (on PII detection)

Compare predicted removed PII to the actual PII present in ground truth (if you have labeled data):

True Positives (TP): Correctly removed PII

False Positives (FP): Non-PII mistakenly removed

False Negatives (FN): PII not removed

Formulas:

Recall = True Positives / (True Positives + False Negatives)

Precision = True Positives / (True Positives + False Positives)

F1 Score is defined as the harmonic mean between precision and recall measures.

This requires ground truth annotations (PII spans in text).

✅ B. Leakage Rate (if no ground truth annotations)

Measure how much PII is left after sanitization:

PII Leakage Rate = PII Entities Detected After Removal / PII Entities Detected Before Removal

Then:

PII Removal Effectiveness = 1−Leakage Rate

Example:

- 10 entities detected in original text
- 2 still found after sanitization → Effectiveness = 1 - (2 / 10) = 80%

✅ C. PII False Positive Rate

To measure over-sanitization (e.g., removing non-PII words):

False Positive Rate = non-PII words incorrectly removed/ non-PII words total

Optional but important if utility is a concern.

---

4. Optional: Utility Preservation Score

To measure how much the text's meaning or readability is preserved after removal (important in chatbots):

BLEU / ROUGE / BERTScore between original and sanitized non-PII segments

Human rating: "Is this text still understandable/useful?"

Cosine Similarity-Based Comparison

Comparing the outputs of LLM for same input but using various architectures like Masking, no masking, replacement of PII data or measuring **semantic drift**- how much the meaning or content of the output changes based on the PII protection technique applied.. For this we would be leveraging something like cosine similarity to compare the results.

Steps:

1. **Choose an Embedding Model**

You'll need a sentence embedding model to convert LLM responses to vector form:
Recommended models:

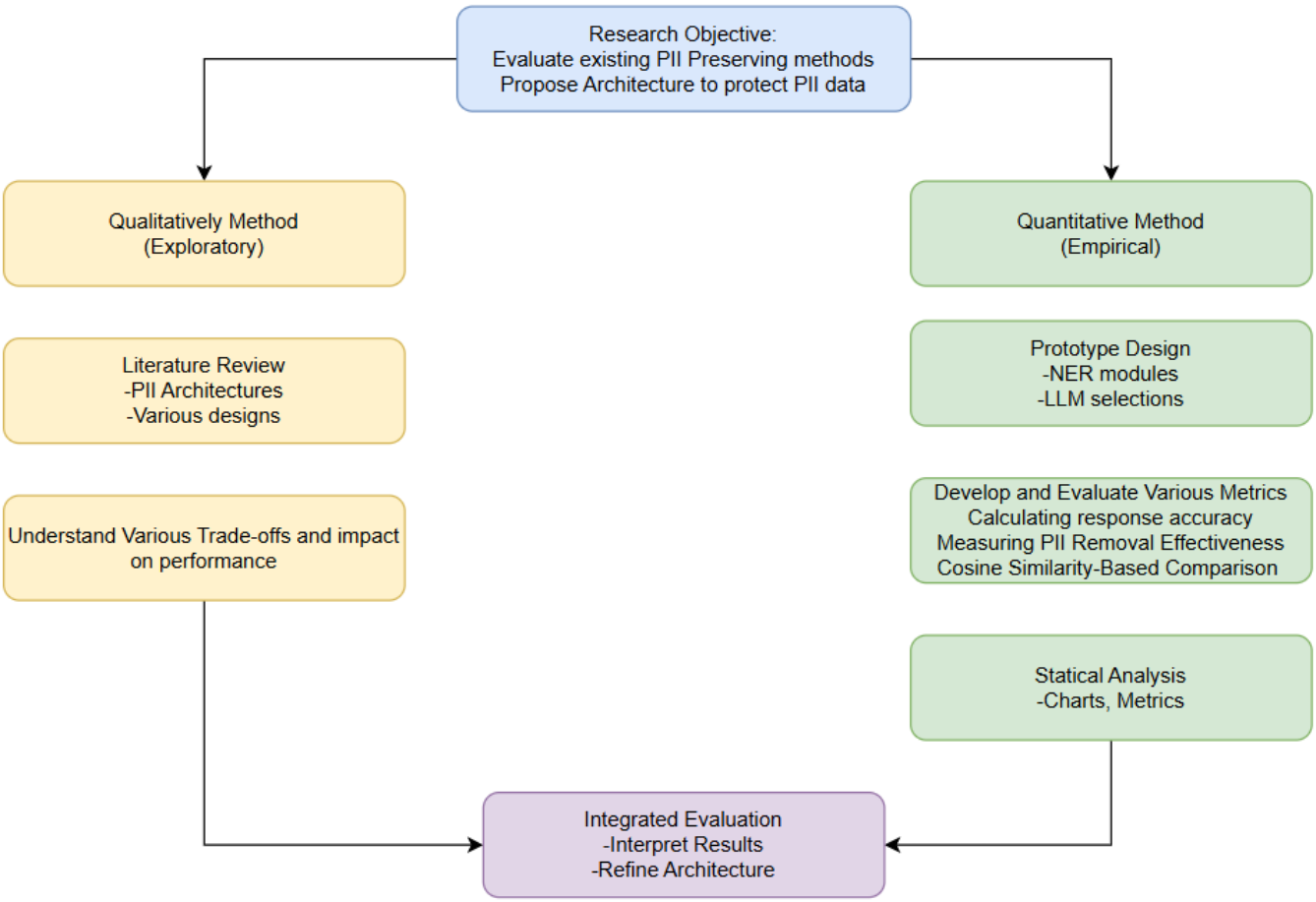
- Sentence-BERT (e.g., all-MiniLM-L6-v2)
- OpenAI's text-embedding-ada-002 (for larger scale)
- Cohere or Google Universal Sentence Encoder

2. **Compute Cosine Similarity Between Outputs**
   Compare embeddings of outputs using cosine similarity
3. **Interpretation**
   Example Table

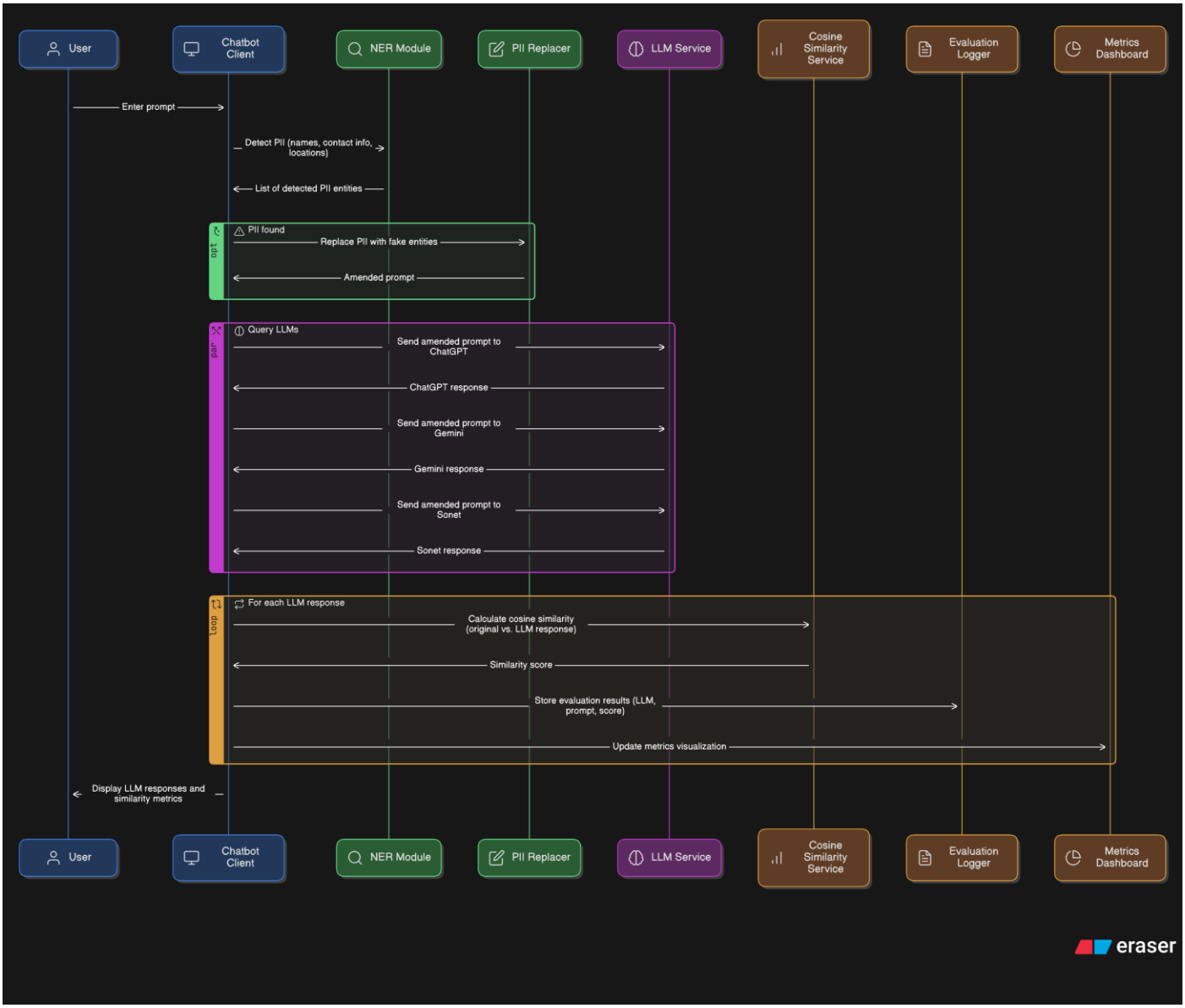
Pair Compared	Cosine Similarity	Interpretation
Raw vs Masked	High (~0.90+)	Semantic meaning is largely preserved
Raw vs Replaced	Lower (0.80–0.90)	Slight semantic shift due to synthetic data
Masked vs Replaced	Can vary	Shows how close different strategies are to each other

You can compute average similarities across a test set for robust evaluation.

Below Diagram shows the Research methodology.



Below Diagram shows what steps will be performed and sample evaluations to be performed.



8. Requirements Resources

Software/Tools:

Category	Tools
Programming	Python, Jupyter, or Colab
ML/LLM Frameworks	OLLAMA, LangChain, OpenAI API, spaCy
PII Detection	Presidio (Microsoft) or Regex-based custom matchers
Metrics & Privacy Testing	PrivacyMeter
Visualization	Matplotlib, Seaborn, or Plotly

Category	Tools
Backend Services	Docker, FastAPI

Hardware: Good spec machine with at least 16 GB of RAM and access to LLM models like Chat GPT, Sonet , OLLAMA

Data Sources: PII | External Dataset , The data collection incorporates comprehensive samples featuring personally identifiable information (PII) including individual names, communication information, and residential locations.

9. Research Plan

Milestone 1: Research Proposal (4 weeks)

Week 1:

- Background
- Problem Statement
- Literature review
- Hypothesis formulation
- Clarifications of doubts with Thesis Supervisor

Week 2:

- Finalize research questions
- Aims and Objectives
- Significance and Scope of Study
- Clarifications of doubts with Thesis Supervisor

Week 3:

- Research Methodology
- Requirement Resources
- Define evaluation metrics: privacy protection, response quality, latency
- Clarifications of doubts with Thesis Supervisor

Week 4:

- Title of the Research/Abstract
- Research plan
- References

- Create a Final Research Proposal template
- Get Signed-off from Thesis Supervisor
- Submit the Research Proposal

Risk:

- Limited access to recent studies.
- Dedicated time for research each day.

Contingency:

- Use interlibrary loans and online databases to access necessary resources.
- Put extra efforts in weekends

Milestone 2: Mid-Thesis (Interim Report) Submission (6 weeks)

Week 5:

- Do more Literature review
- Design & Data collection and preprocessing

Week 6:

- Decide on NER model (e.g., fine-tune spaCy or transformer-based NER)
- Define categories of PII to target (e.g., names, addresses, emails)
- Set up machine and softwares on the laptop
- Perform Pre-processing of source data
- Clarifications of doubts with Thesis Supervisor

Week 7:

- Plan architecture integration of NER & fake data injection
- Create Various Architecture Diagrams

Week 8:

- Code Set up and getting subscriptions of various LLM's
- Experimentation with some of the Key Architecture components
- Clarifications of doubts with Thesis Supervisor

Week 9:

- Initial peak into Experimental results

- Writing Interim report
- Review Interim report with Supervisor

Week 10:

- Refine Interim report and absorb comments from Thesis Supervisor
- Get Signed-off from Thesis Supervisor
- Submit Interim Report

Risk:

- Delays in data acquisition.
- Understanding some of the python libraries.
- Machine set up issues

Contingency:

- Have backup data sources and extend the data collection period if needed.
- Take some courses on python libraries.
- Take help from fellow colleagues in code setup.

Milestone 3: Final Thesis & Video Presentation (8 weeks)

Week 11:

- Do more Literature review
- Data analysis and interpretation

Week 12:

- Train or fine-tune the NER model
- Impact statement
- Refining the Experimental results
- Clarifications of doubts with Thesis Supervisor and absorb previous comments from supervisor

Week 13:

- Refining the evaluation metrics
- Looking at the evaluation metrics, propose best Architecture

Week 14:

- Writing and dissemination of findings

- Clarifications of doubts with Thesis Supervisor and absorb previous comments from supervisor

Week 15:

- Writing and dissemination of findings

Week 16:

- Discussion and Future Work
- Writing and dissemination of findings
- Clarifications of doubts with Thesis Supervisor and absorb previous comments from supervisor

Week 17:

- Review Interim report with Supervisor
- Prepare Video presentation

Week 18:

- Refine Final report and absorb comments from Thesis Supervisor
- Get Signed-off from Thesis Supervisor
- Submit Final Report and Video presentation

Risk:

- Software or hardware malfunctions.
- Delays in writing or peer review.

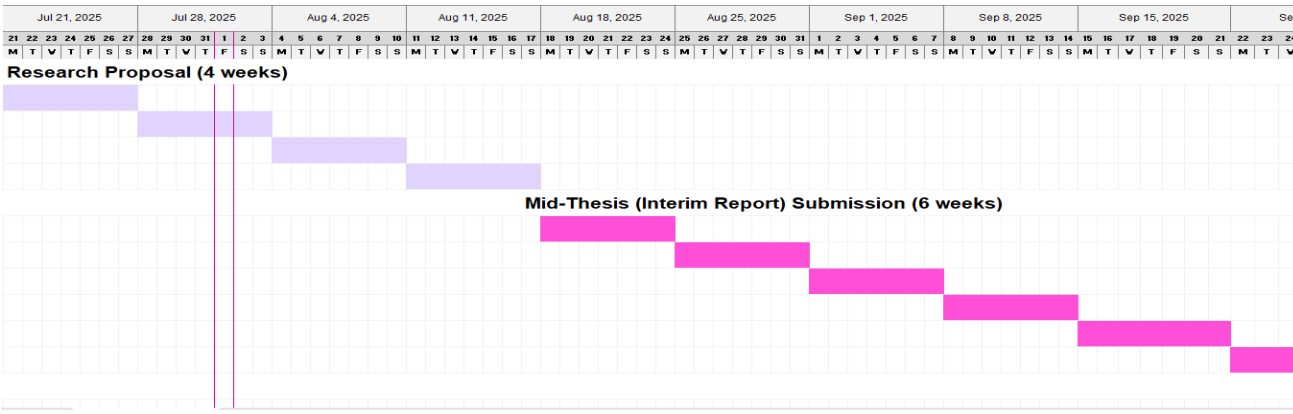
Contingency:

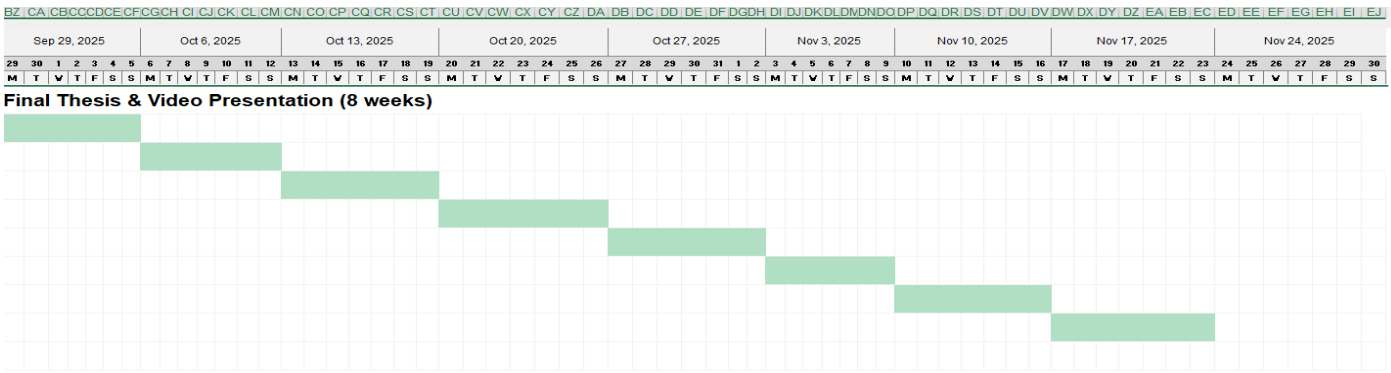
- Regularly back up data and have alternative analysis tools available.
- Allocate extra time for revisions and seek early feedback from peers.

Below is the Table showing above Milestone, Tasks and dates

TASK	PROGRESS	START	END
Research Proposal (4 weeks)			
Background, Problem Statement, Literature review, Hypothesis formulation, Clarifications	100%	7/21/25	7/27/25
Finalize research questions, Aims and Objectives, Significance, Scope, Clarifications	100%	7/28/25	8/3/25
Research Methodology, Resources, Evaluation metrics, Clarifications	100%	8/4/25	8/10/25
Title, Research plan, References, Proposal template, Supervisor sign-off, Submit	100%	8/11/25	8/17/25
Mid-Thesis (Interim Report) Submission (6 weeks)			
Literature review, Design & Data collection	0%	8/18/25	8/24/25
NER model decision, Define PII categories, Setup, Pre-processing	0%	8/25/25	8/31/25
Plan architecture integration, Create diagrams	0%	9/1/25	9/7/25
Code setup, LLM subscriptions, Key component experiments	0%	9/8/25	9/14/25
Experimental results, Write Interim report, Supervisor review	0%	9/15/25	9/21/25
Refine and submit Interim report	0%	9/22/25	9/28/25
Final Thesis & Video Presentation (8 weeks)			
Literature review, Data analysis and interpretation	0%	9/29/25	10/5/25
Train/fine-tune NER model, Impact statement, Refine results, Supervisor clarifications	0%	10/6/25	10/12/25
Refine evaluation metrics, Propose best architecture	0%	10/13/25	10/19/25
Write and disseminate findings, Supervisor clarifications	0%	10/20/25	10/26/25
Write and disseminate findings	0%	10/27/25	11/2/25
Discussion, Future work, Writing, Supervisor clarifications	0%	11/3/25	11/9/25
Review final report, Prepare video	0%	11/10/25	11/16/25
Refine final report, Supervisor sign-off, Submit final report and video	0%	11/17/25	11/23/25

Gannt Chart showcasing the dates to be followed.





References

Bae, Y., Kim, M., Lee, J., Kim, S., Kim, J., Choi, Y., & Mireshghallah, N. (2025). *Privacy-Preserving LLM Interaction with Socratic Chain-of-Thought Reasoning and Homomorphically Encrypted Vector Databases*.

Chen, T., Bao, H., Huang, S., Dong, L., Jiao, B., Jiang, D., Zhou, H., Li, J., & Wei, F. (2022). *THE-X: Privacy-Preserving Transformer Inference with Homomorphic Encryption*.

Chen, Y., Li, T., Liu, H., & Yu, Y. (2023). *Hide and Seek (HaS): A Lightweight Framework for Prompt Privacy Protection*.

Gilbert, C., & Gilbert, M. (2024). The Effectiveness of Homomorphic Encryption in Protecting Data Privacy. *International Journal of Research Publication and Reviews*, *5*, 3235–3256. https://doi.org/10.2139/ssrn.5259722

Kerrigan, G., Slack, D., & Tuyls, J. (2020). *Differentially Private Language Models Benefit from Public Pre-training*.

Liu, X., & Liu, Z. (2023). *LLMs Can Understand Encrypted Prompt: Towards Privacy-Computing Friendly Transformers*.

Mai, P., Yan, R., Huang, Z., Yang, Y., & Pang, Y. (2024). *Split-and-Denoise: Protect large language model inference with local differential privacy*.

Singh, A., Vepakomma, P., Gupta, O., & Raskar, R. (2019). *Detailed comparison of communication efficiency of split learning and federated learning*.

Vepakomma, P., Gupta, O., Swedish, T., & Raskar, R. (2018). *Split learning for health: Distributed deep learning without sharing raw patient data*.

Wei, K., Li, J., Ding, M., Ma, C., Yang, H. H., Farokhi, F., Jin, S., Quek, T. Q. S., & Vincent Poor, H. (2020). Federated Learning With Differential Privacy: Algorithms and Performance Analysis. *IEEE Transactions on Information Forensics and Security*, *15*, 3454–3469. https://doi.org/10.1109/TIFS.2020.2988575

Yang, X., Sun, J., Yao, Y., Xie, J., & Wang, C. (2022a). *Differentially Private Label Protection in Split Learning*.

Yang, X., Sun, J., Yao, Y., Xie, J., & Wang, C. (2022b). *Differentially Private Label Protection in Split Learning*.

Yu, D., Naik, S., Backurs, A., Gopi, S., Inan, H. A., Kamath, G., Kulkarni, J., Lee, Y. T., Manoel, A., Wutschitz, L., Yekhanin, S., & Zhang, H. (2022). *Differentially Private Fine-tuning of Language Models*.