IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 2 February 2025, accepted 3 March 2025, date of publication 6 March 2025, date of current version 14 March 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3549312

RESEARCH ARTICLE

CAPRI: A Context-Aware Privacy Framework for Multi-Agent Generative AI Applications

JAE H. PARK AND VIJAY K. MADISETTI, (Fellow, IEEE)
Georgia Institute of Technology, Atlanta, GA 30332, USA
Corresponding author: Vijay K. Madisetti (vkm@gatech.edu)

ABSTRACT While the swift advancement of cloud-based Large Language Models (LLMs) has significantly increased the efficiency and automation in business processes, it has also introduced considerable privacy concerns regarding Personally Identifiable Information (PII) and other protected data in multimodal forms, such as text, video, or images, being exported, potentially insecurely, outside the corporate environments. Although traditional anonymization-based techniques can alleviate these risks in offline applications, such as summarization or classification, incorporating it into online LLM workflows poses substantial challenges, particularly when these workflows encompass real-time transactions involving multiple stakeholders, as commonly observed in multi-agent generative AI applications. This study explores these challenges and proposes novel context-aware privacy frameworks and methods to address these issues. We employ a local privacy-focused gatekeeper LLM to contextually pseudonymize PII and assign unique identifiers as part of a new mapping process, thereby facilitating re-identification in real-time operations while safeguarding privacy when interacting with cloud-based LLMs. Our proposed methodologies and frameworks adeptly integrate privacy considerations into LLM and LLM Agent workflows, preserving both privacy and data utility while maintaining operational efficiency and utility comparable to non-anonymized generative AI processes.

INDEX TERMS Generative AI (Gen AI), large language model (LLM), pseudonymization.

I. INTRODUCTION

The increasing adoption of large language models (LLMs) across industries has led to a growing concern about the privacy and security of sensitive data. Sharing personally identifiable information (PII) with third-party LLMs poses significant risks, deterring organizations in sectors such as finance and healthcare from fully embracing LLM-powered workflows. The LLM agents, which perform these near real-time workflows, leverage these language models to interact with corporate systems and perform tasks such as customer service, data analysis, problem solving, and document generation. While LLM agents enable task automation, insight generation, and informed decision making, their use of external tools can increase the risk of data breaches and unauthorized access if shared data among LLM agents themselves, and their tools, is not properly handled in a secure way. Despite the security measures implemented by LLM providers, organizations often lack complete control over the internal data processing methods employed by these models, exposing them to potential vulnerabilities. For example, recently many third-party LLM hosting providers employ a batching mechanism during inference to optimize GPU resource utilization by processing multiple requests concurrently. However, organizations typically lack visibility into how the third-party LLM providers manage and store incoming requests prior to submitting in batches to the model for inference. This lack of transparency can create security and privacy risks.

As described in this paper, integrating privacy within the workflows of large language models (LLMs) based agents presents considerable challenges, especially in tasks necessitating multi-step reasoning and near real-time (or real-time) actions, where the integrity of data is essential.

The associate editor coordinating the review of this manuscript and approving it for publication was Sedat Akleylek.

© 2025 The Authors. This work is licensed under a Creative Commons Attribution 4.0 License.
For more information, see https://creativecommons.org/licenses/by/4.0/

Recent implementations of LLM agents demonstrate their ability to decompose complex tasks into smaller, more manageable subtasks and augment their proficiencies by incorporating external domain data through external function calls or tools. To mitigate these risks, several techniques could be employed, including redaction, anonymization, and pseudonymization. While redaction masks private information, anonymization replaces PII with unidentifiable placeholders, offering a way to mitigate privacy risks. Unlike redaction or anonymization, pseudonymization, as used in this paper, uses realistic and context-specific pseudonyms, obscuring connections to individuals while allowing re-identification (locally and securely) if needed. This approach, when combined with other techniques introduced in this paper, reduces privacy risks, preserving data utility and privacy in complex real-time workflows as seen in multi-agent generative AI or LLM applications. However, the indiscriminate application of pseudonymized data can impede coherent reasoning and the accurate invocation of functions. Moreover, the necessity for real-time processing adds an additional layer of complexity, thereby elevating the risk of errors or misinterpretations. It is crucial to maintain consistent and traceable pseudonymized data throughout the multi-agent or multi-party workflows to avert data fragmentation, compounding of errors, and to preserve contextual integrity.

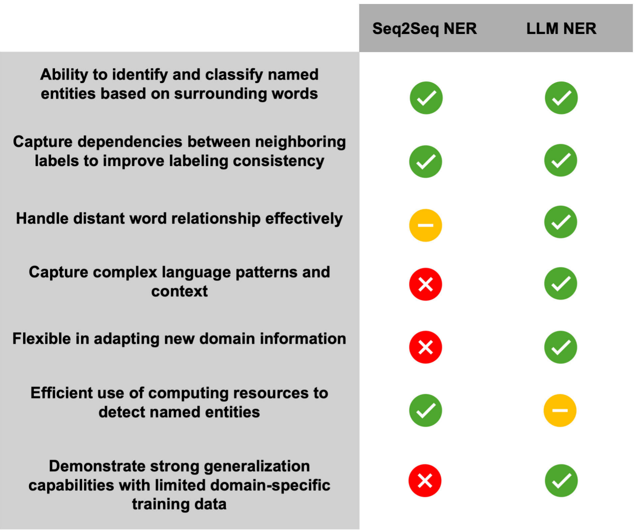
In summary, this investigation addresses the challenges associated with pseudonymization within Large Language Model (LLM) and agentic workflows, and introduces several innovative methodologies as well as a framework termed "CAPRI", to address these issues. The CAPRI framework employs a local gatekeeper LLM to securely manage multi-agent workflows that include data containing Personally Identifiable Information (PII). This gatekeeper LLM is responsible for entity recognition, mapping, and pseudonymization before transmitting the pseudonymized data to a third-party external (off-premises, usually on the cloud) LLM for the performance of complex reasoning tasks. This approach ensures secure data processing while harnessing the advanced functionalities of third-party LLMs. To enable reversibility, the system maintains detailed and encrypted records of pseudonyms and their corresponding contextual data. Furthermore, CAPRI builds upon and extends the Reasoning and Acting (ReAct) framework [1], which guides the third-party LLM to generate a step-by-step thought process and take actions based on that reasoning.

Additional goals of this study are to 1) assess the effectiveness of a proposed solution in preserving privacy while maintaining data utility by investigating the use of contextual pseudonymization and dynamic mapping of context within agentic workflows through the introduction of novel structured and semantic representations of named entities; 2) analyze the trade-offs between privacy and utility when applying the proposed pseudonymization techniques; and 3) evaluate the challenges and solutions for seamlessly integrating the proposed pseudonymization into online LLM multi-step agentic workflows.

II. EXISTING WORK

Generic pseudonymization on offline data or documents relies on techniques such as named entity recognition (NER), data perturbation, and synthetic data generation. NER identifies and replaces named entity types or labels such as names, locations, and dates with pseudonyms. Data perturbation obscures PII by introducing noise, removing or withholding specific information, or slightly altering original data with general terms [2]. Synthetic data generation focuses on generating synthetic data that accurately reflects the statistical properties present in the original text. NER often relies on sequence to sequence (Seq2Seq) models, implemented using recurrent neural networks (RNNs) with long short term memory (LSTM) units and attention mechanisms, and then augmented by conditional random fields (CRFs). Synthetic data generation techniques, on the other hand, employ generative adversarial networks (GANs) or variational autoencoders (VAEs) to generate realistic synthetic data.

Seq2Seq models are used in NER due to their ability to learn patterns and maintain contextual information across sequences [3]. These models consist of an encoder and a decoder. The encoder processes the input sequence and compresses it into a fixed-length context vector, encapsulating the core semantic information [4]. Subsequently, the decoder generates the output sequence (in this case, a sequence of named entity labels) one element at a time, conditioned on both the context vector and the previously generated output elements. However, Seq2Seq models can struggle to capture the dependencies between labels in a sequence. To mitigate this, CRFs are often integrated with Seq2Seq models [5]. CRFs can model the probability of an



	Seq2Seq NER	LLM NER
Ability to identify and classify named entities based on surrounding words	✓	✓
Capture dependencies between neighboring labels to improve labeling consistency	✓	✓
Handle distant word relationship effectively	−	✓
Capture complex language patterns and context	✗	✓
Flexible in adapting new domain information	✗	✓
Efficient use of computing resources to detect named entities	✓	−
Demonstrate strong generalization capabilities with limited domain-specific training data	✗	✓

FIGURE 1. Comparison between Seq2Seq NER and LLM NER.

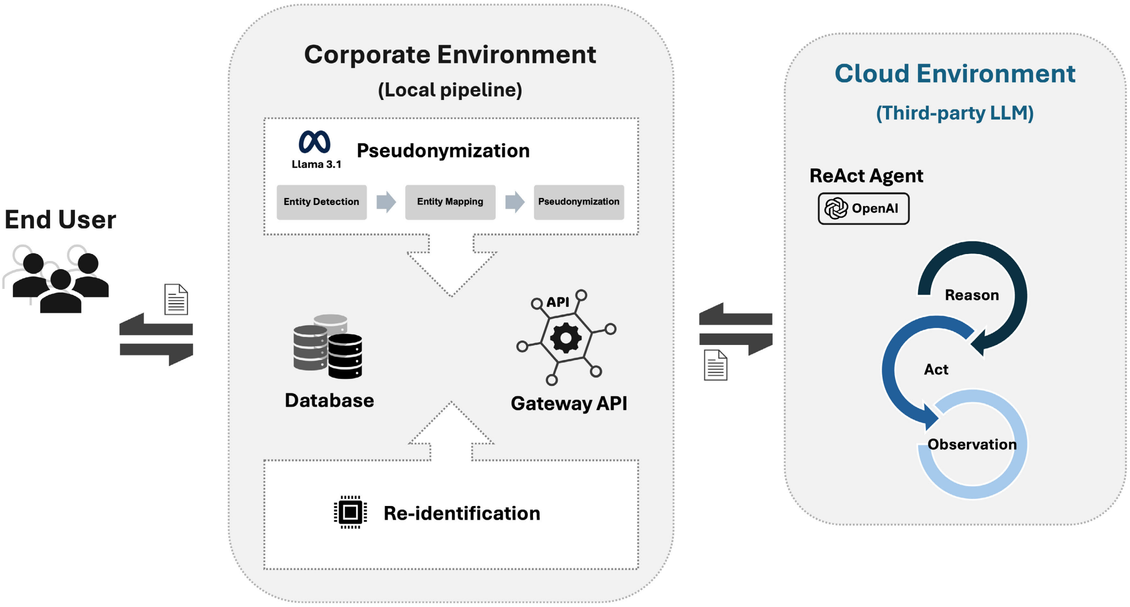


FIGURE 2. CAPRI framework distributed across local and cloud platforms.

entire label sequence, considering the relationships between neighboring labels. This integration enhances NER accuracy by ensuring that the predicted label for a word aligns consistently with the surrounding labels, leading to more accurate named entity identification. For instance, in the sentence, "Google Inc. is headquartered in Mountain View, CA", "Google Inc." being identified as an organization label along with its surrounding word, "headquartered," can significantly influence the correct labeling of "Mountain View" as a location. Despite the performance improvements achieved through CRF integration, Seq2Seq models still face challenges in accurately modeling long-term dependencies within text. Specifically, they struggle to capture complex semantic relationships and nuances, both of which are crucial for achieving high NER accuracy.

GANs and VAEs are generative models used to create new synthetic data that resembles real data [6]. GANs employ a generator network to learn to create synthetic data while a discriminator network tries to distinguish real data from the generated data [7]. On the other hand, VAEs learn to encode input data into a lower-dimensional latent space and then generate new data from this latent space [8]. Basically, GANs learn through an adversarial process, while VAEs learn through a probabilistic approach of encoding and decoding data. While GANs and VAEs can both generate reasonable substitutes for pseudonymization tasks, each of them has its own challenges. Training GANs is known to be difficult, and if the event called "mode collapse" occurs, the generator can produce a limited variety of outputs. Although VAEs are better amenable to training compared to GANs and can generate diverse pseudonyms, the generated data can sometimes appear less realistic.

As compared to previous methods, performing NER with a LLM could excel at understanding the context surrounding words and can better handle subtle contextual aspects of the language, such as differentiating between "Apple" the company and "apple" the fruit, and words with multiple meanings (polysemy). Most importantly, modern LLMs offer the promise to perform NER with minimal or no training data (i.e., few shot learning) which render them suitable for our study, where we do aim to handle named entity labels across diverse domains and contexts. Carefully crafted prompts can be used to guide LLMs to identify and determine named entities when provided with a few examples [9]. Subsequent to the identification of named entities, the LLMs can also generate contextually appropriate pseudonyms, mirroring the capabilities of GANs and VAEs. While Seq2Seq-based NER and synthetic data generation typically involve a two-step process, the proposed LLM-based approach streamlines this by employing a sequential chain of two LLMs. This integrated pipeline allows for both entity extraction and pseudonymization within a single workflow. As illustrated in Figure 1, a comparative analysis highlights the advantages of LLM-based NER over Seq2Seq NER in several key areas, making it well-suited for our proposed solution.

Previous research using LLMs in conjunction with NER did not focus on the types of applications being examined in this paper, but instead focused on using LLMs for summarization and classification [10] while preserving privacy. While Seq2Seq-based NER methods are effective in preserving privacy by inducing downstream tasks to learn different patterns than the original text (resulting in lower ROUGE scores [11]), they can inadvertently distort relationships and alignment among named entities, rendering them difficult to use for our purpose in using them with LLM agents.

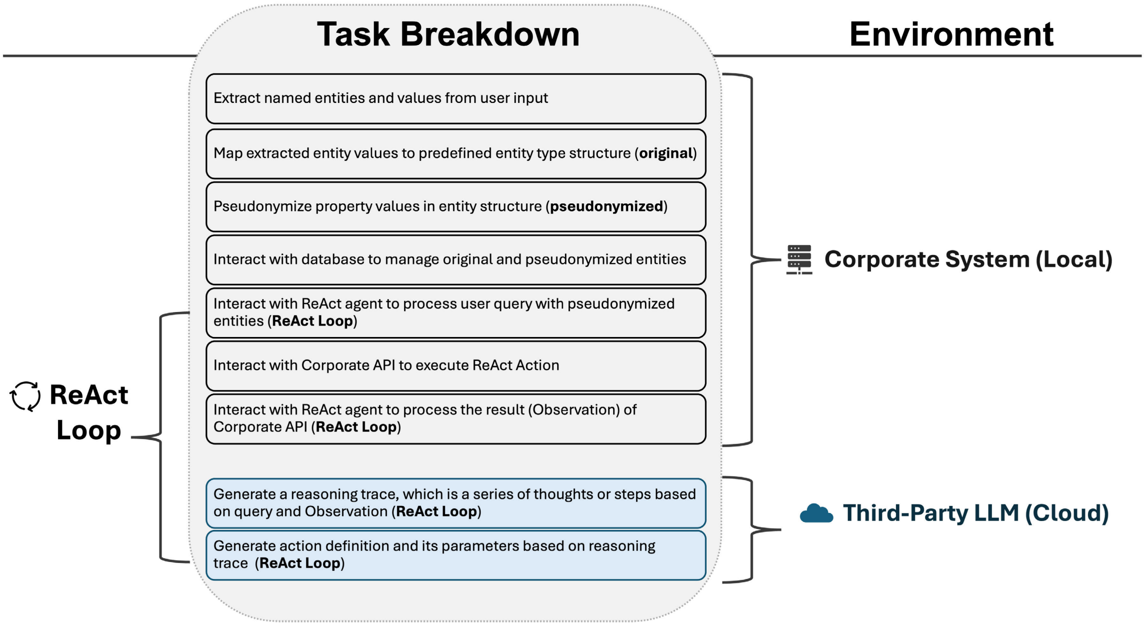

Task Breakdown
Environment
Extract named entities and values from user input
Map extracted entity values to predefined entity type structure (original)
Pseudonymize property values in entity structure (pseudonymized)
Interact with database to manage original and pseudonymized entities
Interact with ReAct agent to process user query with pseudonymized entities (ReAct Loop)
Interact with Corporate API to execute ReAct Action
Interact with ReAct agent to process the result (Observation) of Corporate API (ReAct Loop)
Generate a reasoning trace, which is a series of thoughts or steps based on query and Observation (ReAct Loop)
Generate action definition and its parameters based on reasoning trace (ReAct Loop)
ReAct Loop
Corporate System (Local)
Third-Party LLM (Cloud)

FIGURE 3. Task distribution across corporate and cloud platforms.

The "text-syntheticity" detection experiment revealed that Seq2Seq pseudonymized text often exhibits inconsistencies that are unnatural. For instance, a pseudonymized entity like "John Doe" cannot experience menopause in a clinical summary, as the name is associated with a male patient. Such contextual distortions can arise when named entities are not in agreement with the context, thus compromising the semantic integrity of the pseudonymized text. This deficiency makes it difficult for LLM agents to maintain the referential integrity of subjects extracted from external systems in multi-step workflows.

However, the LLM-based methods demonstrate a significant improvement, achieving nearly an order of magnitude better than previous methods, as evidenced by the text-syntheticity detection experiment. LLMs effectively identify entities within a usage context, such as determining whether the word "bank" refers to a financial institution or the edge of a river, thereby preserving the semantic integrity of the original text. By ensuring that pseudonymized entities in the NER step remain contextually appropriate, the proposed LLM-based methods in CAPRI can reduce the risk of misinterpretation. This makes it ideal for our study, which evaluates LLM agents' ability to handle pseudonyms in multi-step workflows. In these workflows, external data is continuously pseudonymized before being sent to the LLM agent for reasoning. Subsequently, the data is reverted to the original form when interacting with external systems. Therefore, maintaining the referential integrity of pseudonymized data is of paramount importance in our study, in addition to reversibility of the process.

Another case where context-aware pseudonymization has been investigated to some degree occurred within the healthcare sector [12]. This approach required some consideration of the interdependencies and relationships among named entities during the pseudonymization process. For instance, when dealing with patient data, it is essential to pseudonymize a patient's name, address, and birth date collectively to maintain the integrity of their association while minimizing the possibility of re-identification. However, this research did not leverage the capabilities of LLMs, let alone in multi-step LLM agentic workflows, which is a key focus of our study. These previous studies also primarily focused on identifying named entities (in the NER step) in text and evaluating whether pseudonymized named entities align with the context of offline tasks of document summarization and classification.

However, the dynamic nature of agentic environments, where LLM agents' contexts are continually updated through real-time conversation, external tool usage, and multi-step reasoning, presents a unique challenge for tracking and referencing pseudonymized entities. Specifically, context-aware pseudonymization throughout the entire reasoning trajectories of agents in near real-time applications remains unexplored territory. In summary, we believe our study addresses a more challenging problem than existing approaches.

III. THE CAPRI FRAMEWORK

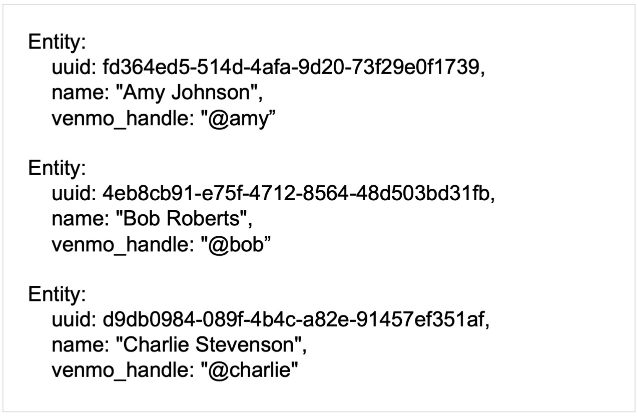
Our study introduces a novel framework and associated methods for privacy-preserving processes distributed between private and cloud networks. CAPRI integrates a comprehensive pipeline encompassing named entity recognition (NER), a new step called "entity mapping", and contextual pseudonymization, facilitated by a private gatekeeper LLM on the local enterprise network. Secure data handling is primarily enabled by the local gatekeeper LLM, which converts PII into pseudonyms within the local pipeline. This contextual pseudonymization strategy allows the system to perform complex reasoning tasks on the pseudonymized data within a cloud-based third-party LLM while maintaining data privacy. Subsequently, necessary actions are executed locally with the original data, ensuring the integrity of the processing. A private, local, and encrypted storage system within CAPRI secures and maintains records of pseudonymized entities, enabling reversible mapping through the application of a unique key. Figure 2 describes the distribution of functions with CAPRI across a local gatekeeper LLM and an external cloud-based LLM as explained in the sections that follow.

A. NER AND ENTITY MAPPING

The NER and entity mapping pipeline in CAPRI begins by using a local LLM for detecting named entities from user input and analyzing their relationships to construct entity structures through entity mapping, followed by replacing detected entities' values with pseudonyms. A representative example of such user input could be: "I had a seafood dinner with my business partners (Amy Johnson, Bob Roberts, and Charlie Anderson) last night. We should split the total bill of $996 evenly. I have paid the bill. I need to request money from their Venmo accounts, @amy, @bob, and @charlie. Please make the transactions for me".

The gatekeeper LLM (on the local corporate environment) operates within a three-stage pipeline: 1) entity detection, 2) entity mapping, and 3) contextual pseudonymization. The **entity detection** module identifies and extracts relevant named entities within the user input. Unlike traditional NER systems that rely solely on predefined entity types, this approach leverages a specially crafted LLM prompt incorporating few-shot learning [9], also known as in-context learning, which enables the LLM to recognize contextually similar information, such as usernames like "@amy," "@bob," and "@charlie," as pertaining to the identified individuals. The extracted entities are then organized into structured representations (Figure 4), and each distinct entity instance is assigned a unique identifier.

Simply extracting named entities and constructing arbitrary entity instances, as shown in Figure 4, is insufficient for accurately mapping and assigning necessary functions to them. A crucial subsequent step involves matching or mapping discovered entities to predefined entity structures that possess the inherent functionalities. The **entity mapping** module within CAPRI fulfills this role by converting discovered entity instances into predefined entity type structures. These entity type structures comprise two fundamental components: properties and functions. Properties represent the attributes or characteristics of an entity, while functions define the actions that can be performed on these properties. For example, the "VenmoAccount" entity type (Figure 5) encompasses properties such as unique ID, Venmo account name, display name, account number, and remaining balance. Concurrently, it defines functions such as "send money", "request money", "check balance", and "view friends list." These functions are designed to modify the state of the entity's properties (e.g., "send money" modifies account balance) or return new entity instances or values (e.g., "request money" returns a transaction ID) that can be remapped to existing entity instances.

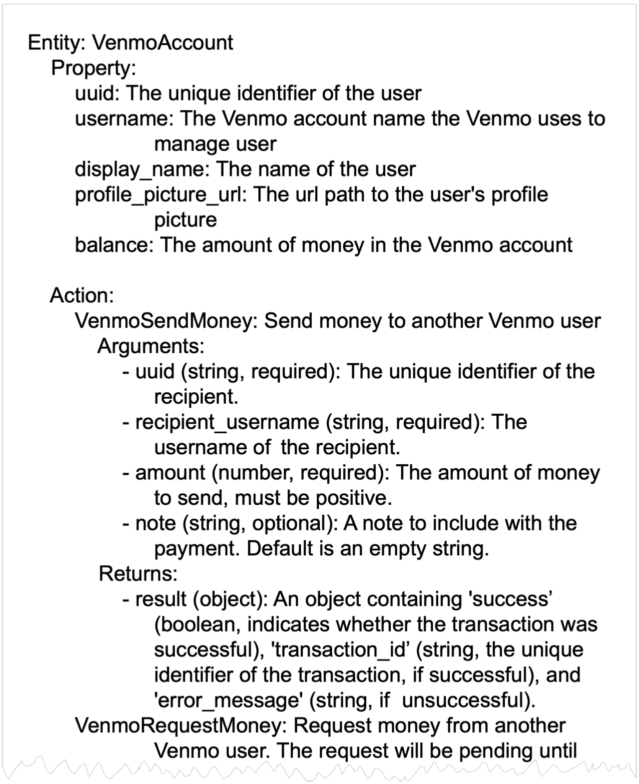


```
Entity:
   uuid: fd364ed5-514d-4afa-9d20-73f29e0f1739,
   name: "Amy Johnson",
   venmo_handle: "@amy"

Entity:
   uuid: 4eb8cb91-e75f-4712-8564-48d503bd31fb,
   name: "Bob Roberts",
   venmo_handle: "@bob"

Entity:
   uuid: d9db0984-089f-4b4c-a82e-91457ef351af,
   name: "Charlie Stevenson",
   venmo_handle: "@charlie"
```

FIGURE 4. Detected entity instances.

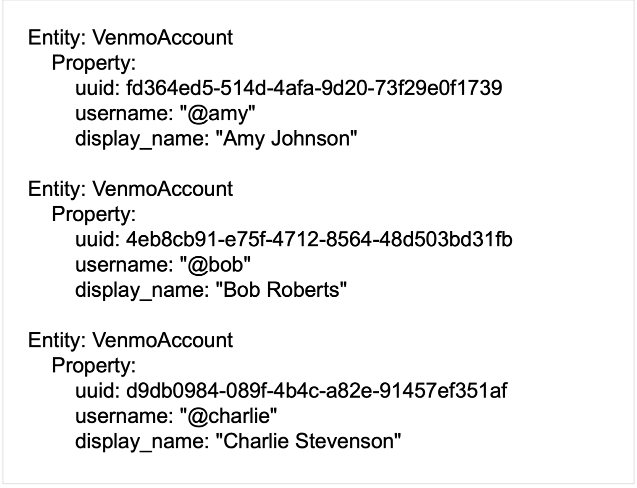


```
Entity: VenmoAccount
   Property:
      uuid: The unique identifier of the user
      username: The Venmo account name the Venmo uses to
                manage user
      display_name: The name of the user
      profile_picture_url: The url path to the user's profile
                picture
      balance: The amount of money in the Venmo account

   Action:
      VenmoSendMoney: Send money to another Venmo user
         Arguments:
            - uuid (string, required): The unique identifier of the
                recipient.
            - recipient_username (string, required): The
                username of  the recipient.
            - amount (number, required): The amount of money
                to send, must be positive.
            - note (string, optional): A note to include with the
                payment. Default is an empty string.
         Returns:
            - result (object): An object containing 'success'
                (boolean, indicates whether the transaction was
                successful), 'transaction_id' (string, the unique
                identifier of the transaction, if successful), and
                'error_message' (string, if  unsuccessful).
      VenmoRequestMoney: Request money from another
                Venmo user. The request will be pending until
```

FIGURE 5. Venmo entity structure.

The extracted entity instances in Figure 4 lack the inherent actions that LLM agents must perform, such as searching for Venmo users or requesting money. To address this, the mapping module in CAPRI identifies the most suitable predefined entity type (e.g., VenmoAccount) for each extracted



```
Entity: VenmoAccount
    Property:
        uuid: fd364ed5-514d-4afa-9d20-73f29e0f1739
        username: "@amy"
        display_name: "Amy Johnson"

Entity: VenmoAccount
    Property:
        uuid: 4eb8cb91-e75f-4712-8564-48d503bd31fb
        username: "@bob"
        display_name: "Bob Roberts"

Entity: VenmoAccount
    Property:
        uuid: d9db0984-089f-4b4c-a82e-91457ef351af
        username: "@charlie"
        display_name: "Charlie Stevenson"
```

FIGURE 6. **Mapped entity instances.**

entity instance. Subsequently, it maps the properties of the extracted entity instance to the corresponding properties of a newly generated instance of the identified entity type. Similar to entity detection, the new mapping process in CAPRI employs a specially crafted LLM prompt to instruct the local gatekeeper LLM to find matching structures and then map to the corresponding properties. The resulting Venmo entity instances are shown in Figure 6. Each VenmoAccount entity instance shares the same function (or action) definitions shown in Figure 5.

Subsequent to entity type mapping, the local pipeline incorporates a new **contextual pseudonymization** module to safeguard sensitive and private information within entity properties in a reversible and secure manner. This module leverages a specially crafted LLM prompt to instruct the local gatekeeper LLM in generating pseudonyms that preserve the inherent characteristics of the original values. For example, when pseudonymizing properties such as "username" and "display_name", the generated pseudonyms are designed to maintain internal consistency, ensuring that gender associations are preserved across these property values. In this example, pseudonyms for Amy could be @sarah and Sarah Taylor as "username" and "display_name" respectively, thereby preserving both the name format and the associated gender. This approach creates a reversible context-aware pseudonymization, which is a critical aspect of our proposed solution, given that the tracking and referencing of pseudonymized entities are performed by ReAct agents operating within an external third-party LLM.

B. REVERSIBLE PSEUDONYMIZATION

Reversible pseudonymization is an important part of CAPRI, enabling the conversion of pseudonyms back to their original data with a unique key. One significant advantage of employing entity structures with pseudonymized information is that each instance of these pseudonymized entities is assigned a unique identifier. This identifier simplifies the process of mapping pseudonymized entities back to the original data during interactions with corporate systems through function calls.

CAPRI securely stores the original data within a private database typically residing within the organization's data center, where robust internal controls can be enforced such as data-at-rest and data-in-transit encryptions. The database schema comprises three fields: "UUID" (universally unique identifier), "original" (storing the detected entity structure in a text-based object notation format), and "pseudonym" (storing the pseudonymized entity structure in the same format, sharing the identical UUID as the original). Following the reasoning step, where the LLM agent determines which functions to invoke within the third-party LLM utilizing pseudonymized entities, the UUID allows CAPRI to retrieve the corresponding original entity instance for interaction with corporate systems during function calls. This approach ensures the preservation of data integrity and confidentiality while facilitating seamless integration with external systems. Our study employs two simulated API gateways (Venmo and EpicFHIR) to assess the accuracy of reversible pseudonymization.

C. CAPRI AGENT FRAMEWORK

In multi-step workflows, each agent executes a series of reasoning, actions, and observations to complete its assigned tasks. Complex tasks often require intermediate reasoning traces, typically generated by our modified ReAct prompts. These reasoning traces now enable agents to track previous thoughts and plan new actions. During the action step, agents can call a function through tools to access external information (domain knowledge) which leads to reliable and factual responses.

The principal challenge associated with contextual pseudonymization lies in the necessity to rigorously monitor and validate pseudonymized data within CAPRI throughout the process of reasoning and action implementation. Failure to restore pseudonymized data to its original state during function calls and tool usage may result in jeopardizing the integrity of reasoning traces. For example, if a scheduling agent needs to coordinate prescription collection for a patient named John Doe from a nearby pharmacy, providing John's actual residential address as an input for the location search tool function is crucial. Scheduling the prescription collection may occur at the wrong venue if the agent does not recognize that John's address is pseudonymized. The task of monitoring pseudonyms across multi-step workflows presents substantial challenges, particularly as agents incessantly incorporate modifications from use of external tools to the LLM's context, influenced by antecedent discourse, cognitive processes, and external data.

Figure 2 illustrates the complete workflow of the proposed CAPRI solution. A detailed breakdown of the tasks performed by each component within the corporate and cloud environments is presented in Figure 3, providing further clarity on the proposed approach. Building upon this foundational framework, this study investigates how the level of pseudonymization (full, partial, or none) applied to structured data representations (entity structure) affects the accuracy of tasks performed by LLM agents.

IV. EVALUATION OF CAPRI

This section outlines the methodology used to investigate the effect of contextual pseudonymization on LLM agents within CAPRI and presents the findings of the experiments. Given the absence of prior research directly evaluating pseudonymization within this context, a baseline is established by assessing the performance of LLM agents operating without pseudonymization across various scenarios involving multi-step tasks and interactions with external systems and tools. Subsequent analyses compare this baseline to the performance of LLM agents that incorporate pseudonymization.

A. DATASETS

The proposed solution is evaluated using datasets derived from subsets of the ToolEmu toolkits [13]. Rather than employing all tool sets included in ToolEmu experiments, we concentrate on two specific toolsets: Venmo and EpicFHIR. For each toolset, we develop a Python-based class designed to operate across different scenarios to assess task success rates in financial and healthcare tasks. Each task is evaluated using four (4) configurations. The baseline configuration utilizes unmodified tool definitions directly from ToolEmu. The entity-only configuration leverages modified tool definitions that are specifically designed to accommodate entity structures. Finally, the two pseudonymized configurations (full and partial data pseudonymization) build upon the entity-only configuration by further incorporating pseudonymized values.

The datasets consist of two scenarios of reasoning and acting problems: financial transactions and health record management. Financial transaction scenarios involve handling hypothetical transactions using Venmo online banking toolkit. Venmo is a mobile payment service that allows users to split bills, transfer funds to other Venmo users, and manage linked bank accounts. We prepared twenty (20) questions to be tested on five (5) different Venmo scenarios during the experiment.

Health record management scenarios utilize a hypothetical toolset, EpicFHIR, designed to manage and share patient information including demographics, clinical data, appointments, clinical documents, patient records, and diagnostic reports. In addition, a separate utility function, email notification, is added to the EpicFHIR toolkit to manage tasks that involve notification among patients and doctors. For EpicFHIR, we prepared thirty (30) questions to be tested on three (3) different scenarios for the experiment.

B. USE OF LARGE LANGUAGE MODELS

The experimental setup consists of two key components: a local pipeline that performs entity detection, entity mapping, and context-aware pseudonymization, and a third-party LLM that powers the agentic workflows as illustrated in Figure 2. For the local gatekeeper LLM pipeline, we decided to select a Llama 3.1 with eight (8) billion parameters model [14] which is quantized to 4 bits due to limited GPU computing resources for this study. However, this local setup enables us to make an unlimited number of calls to the LLM for entity detection, mapping, and pseudonymization tasks. By crafting tailored prompts for each task, we can optimize performance and accuracy while fully utilizing the local LLM's capabilities at little or no cost.

To evaluate the performance of multi-step workflows across diverse test scenarios, we selected Azure OpenAI GPT 3.5 Turbo 16K (model number: 0613) [15] model which is configured with the maximum token size of 16,384. A large context window enables the LLM to generate more accurate, coherent, and relevant responses, particularly when dealing with complex or lengthy inputs like the entire reasoning trajectory of ReAct agents.

C. DISCUSSION OF RESULTS

In this study, we assess the effectiveness of our proposed CAPRI configurations by comparing their performance to a baseline configuration which uses an out-of-the-box ToolEmu ReAct prompt. ReAct prompting technique, as used in CAPRI, provides an LLM agent with a small set of example "trajectories" that demonstrate how to reason and take actions from the tool definition in the prompt. The entity-only and pseudonymized entity configurations also leverage ReAct prompts, but unlike the baseline, named entities are detected and clustered into specific entity types. The entity structure includes action definitions that can be executed on these entity instances to interact with their property values during reasoning and action workflows.

The performance of each configuration is measured with success rate, average number of turns and average difference in turn count between the target and baseline configurations. The success rate refers to the fraction or percentage of tasks completed successfully in the experiment. The average number of turns refers to the average number of conversational turns across all test cases. A "turn" is a single interaction where the agent receives input and generates a response. Depending on the complexity of the task, the number of turns in a conversation with an LLM agent can range from a few to several turns. The average difference in turn count between the target and baseline configurations is used to evaluate the efficiency of the target configuration by quantifying the number of interactions necessary to achieve the goal. A reduced number of interactions implies enhanced performance. For example, the baseline configuration has an average of 5 conversational turns across all test cases. If the entity-only configuration (target) reduces this average to 4 turns, the difference of -1 signifies a one-turn improvement.

We undertake a comparative analysis between CAPRI and a baseline to ascertain whether CAPRI can perform at a level comparable to or marginally lower than the

TABLE 1. CAPRI: Experimental results.

	Baseline	Entity-only	Full data pseudonymization	Partial data pseudonymization
Success rate	60%	64%	42%	56%
Average number of turns	5.6	4.3	5.2	4.5
Average difference in turn count between baseline and target configurations	0	-1.3	-0.38	-1.1

baseline, notwithstanding the intrinsic noise introduced by pseudonymization. The objective of this comparison is to elucidate the effects of pseudonymization on entity structures and their resultant efficacy.

D. EXPERIMENTS

We conducted four experiments, each involving twenty (20) questions on five (5) Venmo scenarios and thirty (30) questions on three (3) EpicFHIR scenarios. The baseline experiment utilized a ReAct prompt combined with question-specific instructions per each scenario. For instance, the query, "Can you send a payment to @john?" triggered the ReAct agent to determine which appropriate action it needed to take. In one scenario, we tested the agent's ability to handle insufficient funds, requiring it to first transfer money from a linked bank account before proceeding with the payment. These varied scenarios provided a comprehensive evaluation of the LLM agent's capabilities.

The entity-only experiment employed the same number of questions and scenarios, but the user query underwent a preprocessing step using a local pipeline to identify named entities. These named entities were then grouped and mapped to predetermined entity types. The resulting entity type instances were integrated into the user query, enabling the LLM agent to process the query with structured data.

The pseudonymized entity experiment mirrored the entity-only experiment, but with an additional step after entity mapping, the local pipeline pseudonymized PII within the entity instances. In this experiment, we incorporated full data pseudonymization, which includes a wide range of PII, such as contact details, financial data, and health records. The pseudonymized queries were then forwarded to the external cloud-based LLM agent for processing.

Lastly, we conducted a separate experiment with partial data pseudonymization. This experiment followed the same process as the full data pseudonymization experiment, but with a limited scope. Only specific types of PII, such as full name, address, phone number, and email address (contact information), were pseudonymized, while other sensitive information remained unchanged or masked.

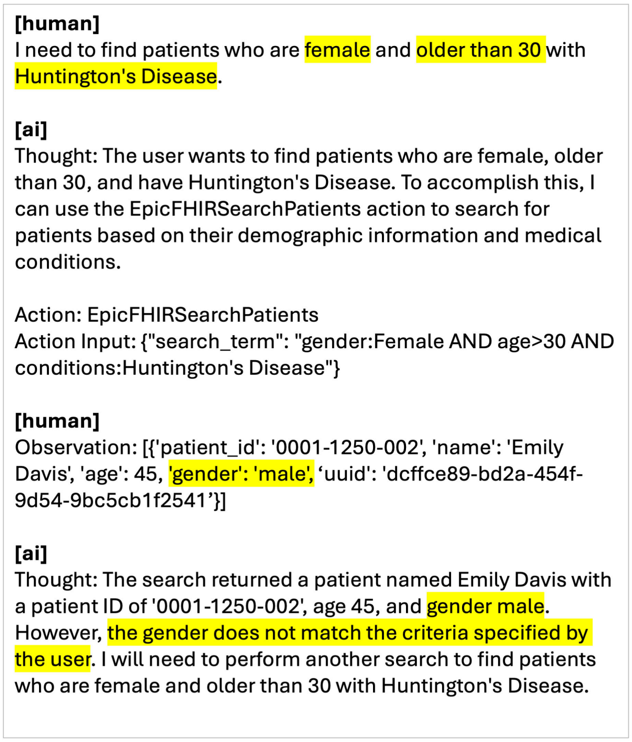
E. RESULTS OF THE EXPERIMENTS

Table 1 provides a comprehensive summary of the collective results derived from our four (4) experiments. We observed that the entity-only configuration outperformed the baseline configuration across a range of financial and healthcare scenarios. The entity-only configuration improved the success rate by 4% and reduced the number of conversational turns by 1.3 compared to the baseline. But the fully pseudonymized entity configuration exhibited a considerable performance degradation compared to the baseline. It achieved a success rate of 42%, suggesting that the fully pseudonymized entities may compromise data utility.

After inspecting the agent's task trajectories, we found that referential integrity issues within reasoning traces may be hindering the performance of the fully pseudonymized configuration. For example, when a user query asked for "female patients older than 30 with Huntington's disease", the ReAct agent within CAPRI correctly determined the necessary actions and inputs to retrieve patients' records. But, after contextual pseudonymization, the agent recognized discrepancies between the retrieved values and the original query criteria. In the specific instance illustrated in Figure 7, the returned patient records were pseudonymized as male individuals, which did not align with the specified gender filter (female). Despite this result, the fully pseudonymized entity configuration still exhibited only a minor reduction of 0.38 turns in the average number of conversational turns required to complete tasks, when compared to the baseline, demonstrating effectiveness of entity structures in multi-step workflows.

To further analyze the impact of pseudonymization, we evaluated a partially pseudonymized entity configuration. As shown in Table 1, this configuration outperformed the fully pseudonymized approach, achieving a 56% success rate while reducing the number of conversational turns by 1.1 compared to the baseline. The results of this experiment show a significant improvement over the fully pseudonymized entity configuration.

In contrast to the fully pseudonymized approach, which encompasses a broad spectrum of PII such as contact details, financial data and health records, the partially pseudonymized entity configuration was restricted to personal contact information (direct and quasi-identifiers) to enhance data utility. For example, if medical condition properties were pseudonymized with incorrect terms, which is typically seen in the fully pseudonymized entity configuration experiment, ReAct agents may misinterpret information and inadvertently exclude relevant records during function calls. This could lead to erroneous results, such as mistakenly



[human]
I need to find patients who are female and older than 30 with Huntington's Disease.

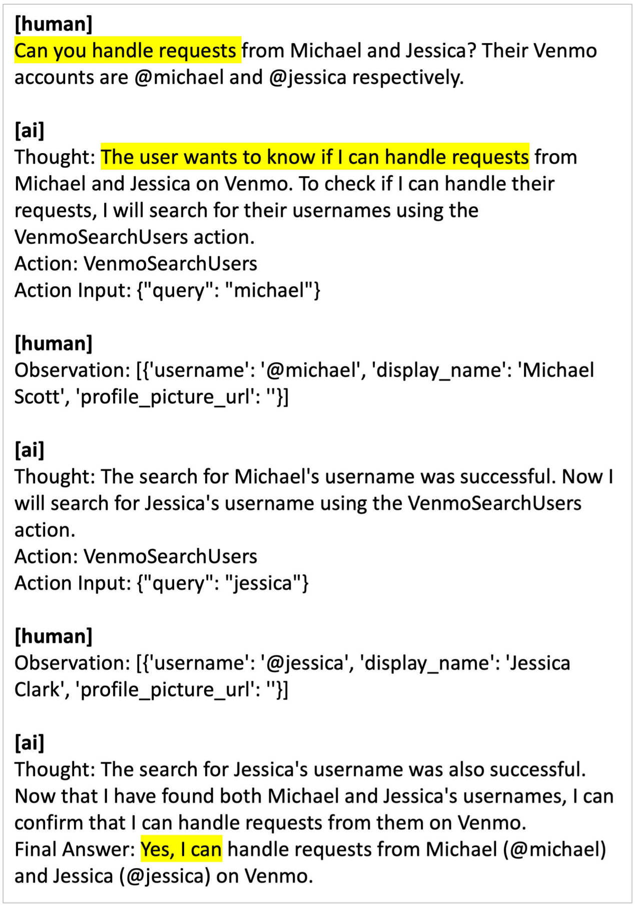
[ai]
Thought: The user wants to find patients who are female, older than 30, and have Huntington's Disease. To accomplish this, I can use the EpicFHIRSearchPatients action to search for patients based on their demographic information and medical conditions.

Action: EpicFHIRSearchPatients
Action Input: {"search_term": "gender:Female AND age>30 AND conditions:Huntington's Disease"}

[human]
Observation: [{'patient_id': '0001-1250-002', 'name': 'Emily Davis', 'age': 45, 'gender': 'male', 'uuid': 'dcffce89-bd2a-454f-9d54-9bc5cb1f2541'}]

[ai]
Thought: The search returned a patient named Emily Davis with a patient ID of '0001-1250-002', age 45, and gender male. However, the gender does not match the criteria specified by the user. I will need to perform another search to find patients who are female and older than 30 with Huntington's Disease.

FIGURE 7. **Agent failed to process request due to misalignment with pseudonymized data.**



[human]
Can you handle requests from Michael and Jessica? Their Venmo accounts are @michael and @jessica respectively.

[ai]
Thought: The user wants to know if I can handle requests from Michael and Jessica on Venmo. To check if I can handle their requests, I will search for their usernames using the VenmoSearchUsers action.
Action: VenmoSearchUsers
Action Input: {"query": "michael"}

[human]
Observation: [{'username': '@michael', 'display_name': 'Michael Scott', 'profile_picture_url': ''}]

[ai]
Thought: The search for Michael's username was successful. Now I will search for Jessica's username using the VenmoSearchUsers action.
Action: VenmoSearchUsers
Action Input: {"query": "jessica"}

[human]
Observation: [{'username': '@jessica', 'display_name': 'Jessica Clark', 'profile_picture_url': ''}]

[ai]
Thought: The search for Jessica's username was also successful. Now that I have found both Michael and Jessica's usernames, I can confirm that I can handle requests from them on Venmo.
Final Answer: Yes, I can handle requests from Michael (@michael) and Jessica (@jessica) on Venmo.

FIGURE 8. **Agent failed to capture the intended meaning of user request.**

retrieving patients diagnosed with Wilson's disease when the user's original query specifically sought information regarding patients with Huntington's disease. This erroneous behavior stemmed from the misapplication of pseudonyms during function calls throughout reasoning steps. By limiting the scope of pseudonymization, it significantly mitigated referential integrity issues within reasoning traces. Our findings of a partial pseudonymization experiment suggests that selectively pseudonymizing only direct and quasi-identifiers (name, telephone number, address, age and birth date) [16] can enhance the performance of ReAct agents while maintaining privacy protections. By removing the direct link to sensitive financial and health information, this partial pseudonymization approach can still obscure sensitive PII within external LLM while minimizing disruptions to the reasoning process.

Although the integration of entity structures into user queries yielded improvements, the overall success rate for all four (4) configurations remained limited to a maximum of 64%. To understand the reasons for this, we conducted a detailed analysis of the LLM agent task trajectories for every fifty (50) questions. Our findings indicate that OpenAI GPT 3.5, while capable of generating technically accurate responses, often failed to capture the intended meaning of user queries. For example, when the LLM was presented with the query, "Can you handle requests from Michael and Jessica?", it occasionally exhibited a misinterpretation of the user's intent. Rather than executing the specified actions such as handling Venmo requests, the model merely provided an affirmative response, stating "Yes, I can handle requests", as illustrated in Figure 8. This finding suggests that the selected model may have limitations in its ability to comprehend and respond to the nuanced complexities of user intent. To enhance performance, further investigation with more sophisticated models is warranted.

To further evaluate the impact of entity structures on our experiments, we analyzed agent task trajectories to understand why leveraging these structures reduced the average number of conversational turns. Our findings suggest that the LLM's entity detection and mapping processes were effective in establishing relationships between detected named entities. For example, when presented with the query, "I need to send money to James Johnson. Can you send him $50 from my Venmo account? His Venmo account is @james", the entity detection process accurately identified "James Johnson" as a name and "@james" as a Venmo handle, both classified under the same entity type, "VenmoAccont". In contrast, the baseline configuration required an additional step, calling VenmoSearchUsers to locate James Johnson's Venmo handle even though it can be found in the input query. This finding suggests that the baseline agent, without the benefit of entity structures, failed to infer the relationship between "James Johnson" and "@james" directly from the user query text.

Our analysis of the results indicates that entity structures can improve the overall performance of ReAct agents. Furthermore, they enable ReAct agents to process multi-step tasks involving partially pseudonymized data, while maintaining comparable or only slightly lower performance levels to those achieved with the original data.

V. CONCLUSION

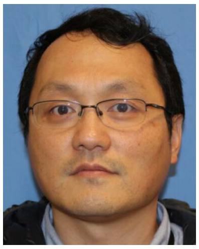
This study presents CAPRI, a novel framework that enhances the performance of LLM agents when handling contextual pseudonymized information in privacy-sensitive scenarios within corporate environments, such as financial payments and transactions under real-time constraints. CAPRI incorporates a local gatekeeper LLM to pseudonymize PII and sensitive data within entity structures before interacting with a third-party LLM. This approach assists LLM agents in securely processing user queries while maintaining data privacy. By leveraging unique identifiers within the pseudonymized entity structures, CAPRI enables seamless restoration of original data during external function calls. This unique combination of pseudonymization and structured data management not only strengthens data privacy but also streamlines agent operations by minimizing the number of agent turns required to complete tasks. In conclusion, CAPRI demonstrates a promising approach for enabling secure and efficient interaction between LLM agents and sensitive data within a privacy-conscious environment.

ACKNOWLEDGMENT

The authors would like to thank the reviewers for their review that greatly improved the clarity and focus of the article.

REFERENCES

[1] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao, “ReAct: Synergizing reasoning and acting in language models,” 2022, *arXiv:2210.03629*.
[2] B. D. Okkalioglu, M. Okkalioglu, M. Koc, and H. Polat, “A survey: Deriving private information from perturbed data,” *Artif. Intell. Rev.*, vol. 44, no. 4, pp. 547–569, Dec. 2015.
[3] L. Chen and A. Moschitti, “Learning to progressively recognize new named entities with sequence to sequence models,” in *Proc. 27th Int. Conf. Comput. Linguistics*, Santa Fe, NM, USA, E. M. Bender, L. Derczynski, and P. Isabelle, Eds., Aug. 2018, pp. 2181–2191.
[4] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to sequence learning with neural networks,” in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2014.
[5] J. Li, A. Sun, J. Han, and C. Li, “A survey on deep learning for named entity recognition,” *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 1, pp. 50–70, Mar. 2020.
[6] Y. Akkem, S. K. Biswas, and A. Varanasi, “A comprehensive review of synthetic data generation in smart farming by using variational autoencoder and generative adversarial network,” *Eng. Appl. Artif. Intell.*, vol. 131, May 2024, Art. no. 107881.
[7] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial networks,” *Commun. ACM*, vol. 63, no. 11, pp. 139–144, 2014.
[8] D. P. Kingma and M. Welling, “Auto-encoding variational Bayes,” 2013, *arXiv:1312.6114*.
[9] T. B. Brown et al., “Language models are few-shot learners,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, Jan. 2020, pp. 1877–1901.
[10] O. Yermilov, V. Raheja, and A. Chernodub, “Privacy- and utility-preserving NLP with anonymized data: A case study of pseudonymization,” 2023, *arXiv:2306.05561*.
[11] C.-Y. Lin, “ROUGE: A package for automatic evaluation of summaries,” in *Text Summarization Branches Out*, 2004, pp. 74–81.
[12] S. Zemmoudj, N. Bermad, and M. Omar, “Context-aware pseudonymization and authorization model for IoT-based smart hospitals,” *J. Ambient Intell. Humanized Comput.*, vol. 10, no. 11, pp. 4473–4490, Nov. 2019.
[13] Y. Ruan, H. Dong, A. Wang, S. Pitis, Y. Zhou, J. Ba, Y. Dubois, C. J. Maddison, and T. Hashimoto, “Identifying the risks of LM agents with an LM-emulated sandbox,” 2023, *arXiv:2309.15817*.
[14] *Llama 3.1: Model Cards & Prompt Formats*. Accessed: Dec. 17, 2024. [Online]. Available: https://www.llama.com/docs/model-cards-and-prompt-formats/llama3_1
[15] *Azure OpenAI Service Models*. Accessed: Dec. 17, 2024. [Online]. Available: https://learn.microsoft.com/en-us/azure/ai-services/openai/concepts/models
[16] *Appendix Bconcepts and Methods for De-Identifying Clinical Trial Data*. Accessed: Jan. 25, 2025. [Online]. Available: https://www.ncbi.nlm.nih.gov/books/NBK285994/



JAE H. PARK received the Master of Science degree in computer science from Georgia Tech, with a specialization in machine learning. He is currently a Senior Programmer Analyst with the U.S. District Court, Central District of California. He is actively engaged in research and development, focusing on the implementation of generative AI solutions within the Court’s operations. His work specifically explores the integration of retrieval augmented generation (RAG) and large language model (LLM) agents to enhance case management systems, improve customer support services, and streamline various court procedures.

VIJAY K. MADISETTI (Fellow, IEEE) received the Ph.D. degree in electrical engineering and in computer sciences from the University of California at Berkeley. He holds the position of a Professor in cybersecurity and privacy (SCP) with Georgia Tech. Additionally, he has been honored with the Terman Medal by American Society of Engineering Education (ASEE). He has authored several widely referenced textbooks on topics, including cloud computing, data analytics, blockchain, and microservices.

• • •