UNIVERSITÀ
DEGLI STUDI
DI PADOVA

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

UNIVERSITY OF PADUA
DEPARTMENT OF INFORMATION ENGINEERING
MASTER'S DEGREE IN COMPUTER ENGINEERING

Balancing Safety and Information Disclosure in Medical Chatbots: A Signaling Game Approach

CANDIDATE

Kejsi Bimaj

Student ID 2112145

SUPERVISOR

Prof. Leonardo Badia

UNIVERSITY OF PADUA

ACADEMIC YEAR 2025/2026

GRADUATION DATE APRIL 15, 2026

Success is not the degree you hold in your hand on graduation day,

but a constellation of quiet, unspoken victories—

standing in a big city among strangers, learning to find yourself,

when a foreign street begins to feel familiar beneath your steps,

when your voice finds courage in a new language,

when your dreams learn to speak in unfamiliar words,

and every time you choose to stay,

even when your heart calls you back home.

***

Suksesi nuk është diploma që mban në dorë ditën e diplomimit,

por është një yjësi fitoresh të heshtura dhe të pathëna—

të qëndruarit në një qytet të madh, mes të panjohurish, duke mësuar të gjesh veten,

kur një rrugë e huaj fillon të ndihet e njohur nën hapat e tu,

kur zëri yt gjen guxim në një gjuhë të re,

kur ëndrrat e tua mësojnë të flasin me fjalë të huaja,

dhe çdo herë që zgjedh të qëndrosh,

edhe kur zemra jote të thërret të kthehesh në shtëpi.

Abstract

Artificial Intelligence (AI) systems and Large Language Models (LLMs) are increasingly used in healthcare applications, including medical chatbots that provide information and guidance to users. However, these systems face an important challenge: they need to give useful answers to normal users while also stopping people who try to misuse them to get harmful or sensitive information. Many current methods mainly rely on filtering content or using fixed rules for safety, but these do not always fully handle the interaction between users and the chatbot.

This thesis addresses the central research question: *How can game theory help a medical chatbot distinguish between honest and malicious users and respond in a safer and more reliable way?* To answer this question, the interaction between the user and the chatbot is modeled as a signaling game with asymmetric information. The user acts as the sender, while the chatbot acts as the receiver. A decision framework based on expected utility and risk thresholds is used to guide the chatbot's behavior under uncertainty.

As part of this work, an existing system, *Clinical-ChatBot*, is modified and extended by integrating a game-theoretic decision framework. The system is combined with RAG and a vector database (Pinecone) in order to retrieve relevant medical knowledge and improve response reliability.

To evaluate the proposed approach, datasets containing both normal medical questions and potentially harmful prompts are used. The queries are analyzed and classified according to their level of risk, and the chatbot's decision strategy is tested through different interaction scenarios. The chatbot can choose between several actions such as *Allow*, *Restrict*, or *Clarify*, depending on the estimated risk level of the user query.

In conclusion, this thesis demonstrates the importance of incorporating game-theoretic reasoning into AI-based medical chatbots, improving their ability to manage uncertain user intentions while maintaining useful communication with legitimate users. Furthermore, this work shows how combining Artificial Intelligence (AI), Natural Language Processing (NLP), and game theory can contribute to safer and more reliable healthcare chatbot applications.

Contents

List of Figures

List of Tables

List of Acronyms

AI Artificial Intelligence

LLM Large Language Model

RAG Retrieval-Augmented Generation

NLP Natural Language Processing

NLU Natural Language Understanding

NLG Natural Language Generation

GPT-4 Generative Pre-trained Transformer 4

RLHF Reinforcement Learning from Human Feedback

API Application Programming Interface

HTTP HyperText Transfer Protocol

HTTPS HyperText Transfer Protocol Secure

CORS Cross-Origin Resource Sharing

UI User Interface

PDF Portable Document Format

MUI Material User Interface

1 Introduction

1.1 Background

Recent developments in Artificial Intelligence (AI) and Natural Language Processing (NLP) have enabled the development of conversational systems capable of interacting with users in natural language [30]. Among these systems, Large Language Models (LLMs) have become increasingly popular due to their ability to generate human-like responses and assist users in various domains such as education, customer service, and healthcare [15].

In the healthcare domain, medical chatbots are gaining significant attention as tools that can provide medical information, assist patients with basic health-related questions, and improve accessibility to healthcare services [11]. These systems can help reduce the workload of healthcare professionals while providing immediate responses to users seeking medical guidance.

Despite these advantages, the use of AI-based chatbots in sensitive domains such as healthcare, introduces several challenges related to reliability, safety, and responsible use. Unlike traditional information retrieval systems, modern conversational agents generate responses dynamically, which increases the risk of providing misleading or potentially harmful information [10, 12].

Another important challenge arises from the interaction between the user and the chatbot itself. Not all users interact with the system with the same intentions. Some users may ask legitimate medical questions, while others may attempt to exploit the system to obtain unsafe or harmful information. Therefore, the chatbot must operate under uncertainty regarding the true intentions

of the user.

Game theory offers a useful framework for modeling such interactions between rational agents with incomplete information [4]. In particular, signaling games help researchers study situations where one person knows what action they will take, while the other person does not and must infer it from observable signals. This makes game theory particularly suitable for modeling interactions between users and AI systems where user intent cannot be directly observed.

1.2 Problem Statement

The growing use of AI-powered medical chatbots has raised several important concerns related to safety, trust, and responsible use. Although these systems are intended to support users by providing medical information and guidance, they can also be misused by individuals seeking harmful or sensitive information. Medical AI systems may inadvertently provide inaccurate or misleading health information, and even small errors can negatively affect patient decisions [12].

One of the main challenges is that the chatbot cannot directly observe the user's true intent. When a query is submitted, the system must determine whether the request is genuine or potentially malicious. If the chatbot responds too openly, it may generate unsafe or misleading information. At the same time, overly restrictive behavior could prevent legitimate users from receiving the help they need.

For this reason, medical chatbots need to balance helpfulness with safety. Reaching this balance requires decision-making mechanisms that can effectively deal with uncertainty about user intentions. Game theory offers a useful framework for modeling interactions between agents with incomplete information, making it suitable for reasoning about uncertain user intent [4].

This thesis addresses the following research question: How can game-theoretic reasoning help a medical chatbot distinguish between honest and malicious users and make safer response decisions under uncertainty?

To answer this question, the thesis examines how signaling games can be applied to model the interaction between users and chatbots. This approach allows the system to evaluate different strategies and select responses based on estimated levels of risk.

1.3 Research Objectives

The main objectives of this thesis are the following:

- To analyze the challenges associated with AI-based medical chatbots, particularly regarding safety, misuse prevention, and decision-making under uncertainty.
- To investigate how concepts from game theory, especially signaling games, can be used to model the interaction between users and medical chatbot systems.
- To study how different parameters within the game-theoretic model influence the behavior of the chatbot and the strategies adopted by users.
- To analyze how changes in these parameters affect the equilibrium outcomes of the signaling game and how the chatbot adapts its responses to maintain a balance between information disclosure and safety.
- To implement and evaluate a modified version of the Clinical-ChatBot system that integrates this game-theoretic framework and observes how the system behaves under different strategic scenarios.

1.4 Thesis Structure

The remainder of this thesis is organized as follows:

- **Chapter 2: Background and Related Work** presents an overview of related work on AI-based chatbots, safety challenges in Large Language Models, and the application of game theory in AI systems.
- **Chapter 3: Methodology** describes the proposed framework, including the signaling game model, the decision-making mechanism, the modifications introduced to the Clinical-ChatBot system, the technical implementation of the system and presents the datasets and evaluation meth- ods used to analyze chatbot behavior.
- **Chapter 5: Results and Analysis** presents and discusses the experimental results, focusing on how effective the proposed approach is in improving chatbot safety and decision-making. It also looks at how changes in key parameters affect both the decision-making process and the overall performance of the system.
- **Chapter 6: Conclusions** brings together the main findings of the thesis and highlights potential directions for future research.

2

Background and Related Work

This chapter provides the theoretical background and reviews existing research related to medical chatbots, safety, and information disclosure. It begins with an overview of medical chatbots and their use in healthcare, along with the main safety and trust challenges they face. The chapter then discusses the risks of unsafe information disclosure and the challenge of identifying user intent, especially in adversarial settings. Next, key concepts from game theory are introduced, focusing on signaling games and their role in decision-making under uncertainty. Finally, the chapter reviews current safety approaches in medical conversational systems, highlighting their limitations and the research gap addressed by this thesis.

2.1 Medical Chatbots in Healthcare

Conversational AI systems, also called chatbots or conversational agents, are artificial intelligence systems designed to interact with users in natural language, either via text or voice. Early systems relied on rule-based mechanisms and predefined dialogue trees, limiting flexibility and applicability. Advances in natural language processing and machine learning have enabled retrieval-based and generative systems capable of producing context-aware responses [47].

These systems are increasingly used in sensitive domains such as healthcare, where they support patient education, symptom monitoring, and chronic disease management [27]. However, interpreting user intent remains challenging, particularly in the presence of strategic or adversarial behavior [21].

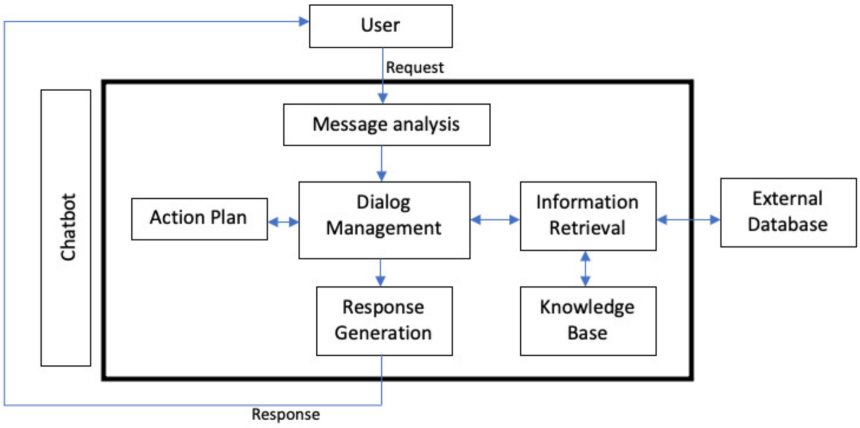

Figure 2.1: General architecture of a medical chatbot system. The diagram illustrates the main components, including user input processing, natural language understanding (NLU), dialogue management, and response generation, highlighting the flow of information within conversational AI systems. Adapted from Xu et al . [45].

Modern conversational agents rely heavily on large language models (LLMs), which exhibit strong generalization capabilities and enable open-ended dialogue [13]. Despite these advantages, LLMs introduce significant safety and reliability concerns. They may generate inaccurate or misleading medical information and typically assume cooperative user behavior, making them vulnerable to adversarial inputs such as prompt injection and jailbreak attacks [24].

Recent work shows that LLM behavior depends not only on pre-training but also on post-training alignment. Ouyang et al. [35] demonstrate that Reinforcement Learning from Human Feedback (RLHF) improves instruction following, truthfulness, and reduces harmful outputs, with aligned smaller models often preferred over larger unaligned ones. Similarly, Bai et al. [9] propose Constitutional AI, where models self-regulate using predefined principles, improving harmlessness through structured reasoning.

Medical chatbots provide valuable support in healthcare settings, enhancing patient engagement and accessibility [1]. However, safety remains a major concern due to risks such as inaccurate advice, data privacy issues, and loss of user trust [12]. Moreover, recent studies show that safety mechanisms can be bypassed through adversarial prompting, while overly restrictive systems may incorrectly reject benign queries.

These limitations highlight the need for advanced safety mechanisms that go beyond static filtering, incorporating reasoning about uncertainty, user in-

tent, and adaptive information disclosure. This motivates the use of principled frameworks, such as signaling games, for modeling safe and reliable medical chatbot behavior.

2.2 Safety and Information Disclosure in Medical AI Systems

Medical AI systems, including chatbots, may inadvertently provide inaccurate or misleading health information. Even small errors can affect patient decisions, potentially causing harm or delaying professional care [12]. Ensuring that AI responses are accurate and safe is therefore a fundamental requirement for deploying medical chatbots in healthcare settings.

Medical AI systems, particularly chatbots deployed in healthcare settings, face inherent risks related to unsafe information disclosure. Early conversational agents relied on rule-based safety mechanisms, which limited flexibility but also reduced the likelihood of hazardous responses. However, modern AI systems, especially those powered by LLMs, generate responses probabilistically, which can inadvertently include hallucinations, incomplete recommendations, or misleading medical guidance [13].

Blease et al. [12] highlight that even minor inaccuracies can have significant consequences for patient safety, such as improper self-medication, misdiagnosis, or delayed professional care. These risks are amplified in scenarios where users consult AI for urgent or high-stakes medical decisions. Furthermore, LLMs are trained on large text corpora that may include outdated or incorrect medical knowledge, further complicating the reliability of outputs [47].

Recent research highlights that improving safety in LLM-based systems requires effective post-training alignment techniques. Ouyang et al. [35] demonstrate that RLHF can improve instruction following, increase truthfulness, and reduce harmful outputs. However, while such approaches improve average model behavior, they do not fully eliminate unsafe or uncertain responses, particularly in high-risk domains such as healthcare.

To mitigate these risks, AI systems must integrate mechanisms that verify responses against medical knowledge bases, detect uncertainty, and provide safe fallback options when information is uncertain. This ensures that chatbots act as assistive tools rather than authoritative medical sources, maintaining both safety

and user trust.

2.2.1 Adversarial and Malicious User Behavior

In addition to unintentional errors, medical AI systems must also address adversarial or malicious user behavior, where users deliberately craft inputs to exploit model vulnerabilities. Such behavior can include attempts to override safety filters, bypass content restrictions, or elicit sensitive medical information. Ippolito et al. [24] report that even LLMs refined with human feedback and safety tuning are susceptible to prompt injection and jailbreak attacks, which can circumvent predefined safety mechanisms.

Recent empirical studies highlight clear evidence of these vulnerabilities. Wei et al. [43] show that safety-aligned models can still be manipulated through adversarial prompting strategies, often achieving high success rates in bypassing safety constraints across different model families. Their findings point to two main causes of failure: competing training objectives and mismatched generalization, both of which can lead models to produce harmful outputs when prompts are carefully reformulated.

Beyond prompt injection, attackers may also attempt to poison training data or manipulate model architectures in order to degrade performance or subtly influence outputs [2]. Such attacks can result in unsafe or misleading medical guidance, sometimes without a clear impact on standard evaluation metrics. Xu and Parhi [46] provide a broader overview of adversarial attacks on LLMs, emphasizing that vulnerabilities can appear at multiple stages, from training to inference, and underlining the importance of proactive detection strategies.

Prompt injection in medical contexts has proven particularly effective in compromising model behavior. For instance, Lee et al. [29] demonstrate that LLMs designed to provide medical advice can be manipulated in controlled settings to generate unsafe recommendations, even when initial safeguards are present. Similarly, Clusmann et al. [19] show that multimodal oncology AI systems, which combine image and text inputs, remain vulnerable to carefully crafted adversarial prompts. These results suggest that such vulnerabilities are not limited to text-only systems, but also extend to more complex multimodal medical AI applications.

At a broader level, recent studies indicate that a significant portion of adversarial prompts are not overtly malicious, but instead exploit ambiguity and

context to bypass detection mechanisms. This makes it difficult for static safety filters to reliably distinguish between benign and harmful queries. As a result, purely rule-based or classification-based defenses often fail under realistic conditions.

These adversarial challenges highlight that traditional static safety filters are insufficient. Effective defense requires dynamic reasoning about user intent, including classification of users as honest or adversarial, anomaly detection, and adaptive response strategies [25]. This perspective aligns naturally with signaling game frameworks, where the AI must balance the disclosure of helpful information with the management of risk under uncertainty.

2.2.2 Trust, Uncertainty, and User Intent in Human–AI Interaction

Trust is a critical factor in healthcare interactions, as users often rely on AI-generated guidance for sensitive health decisions. Abd-Alrazaq et al. [1] show that patients frequently disclose personal and health-related information to chatbots, and that perceived reliability strongly influences user engagement and willingness to rely on AI-generated advice.

Blease et al. [12] highlight that uncertainty about user intent significantly complicates the deployment of medical AI systems. Systems that assume cooperative users may inadvertently disclose unsafe or sensitive information, whereas overly cautious systems may withhold useful guidance. This trade-off reflects a fundamental tension between helpfulness and safety in LLM-based systems.

Recent alignment research attempts to address this challenge. Ouyang et al. [35] show that RLHF can substantially improve model behavior, doubling performance on certain truthfulness benchmarks and reducing toxic outputs by up to 25%. Similarly, Bai et al. [9] demonstrate that Constitutional AI can improve harmlessness while maintaining comparable levels of helpfulness, suggesting that structured self-critique processes can guide safer responses.

Recent work also highlights the problem of over-refusal in safety-aligned systems. Zhang et al. [46] show that many prompts that appear potentially harmful are in fact benign when interpreted in context, leading models to incorrectly refuse valid queries. Their dataset demonstrates that this issue arises particularly in ambiguous scenarios, where intent is unclear. This suggests that binary decisions between allowing and rejecting responses are often insufficient, moti-

vating the need for intermediate strategies that explicitly handle uncertainty.

In particular, these findings indicate that effective safety mechanisms must go beyond static classification and incorporate contextual reasoning about user intent. Addressing over-refusal therefore requires models to distinguish between genuinely harmful requests and those that only appear risky at a surface level, reinforcing the importance of adaptive and uncertainty-aware decision-making approaches.

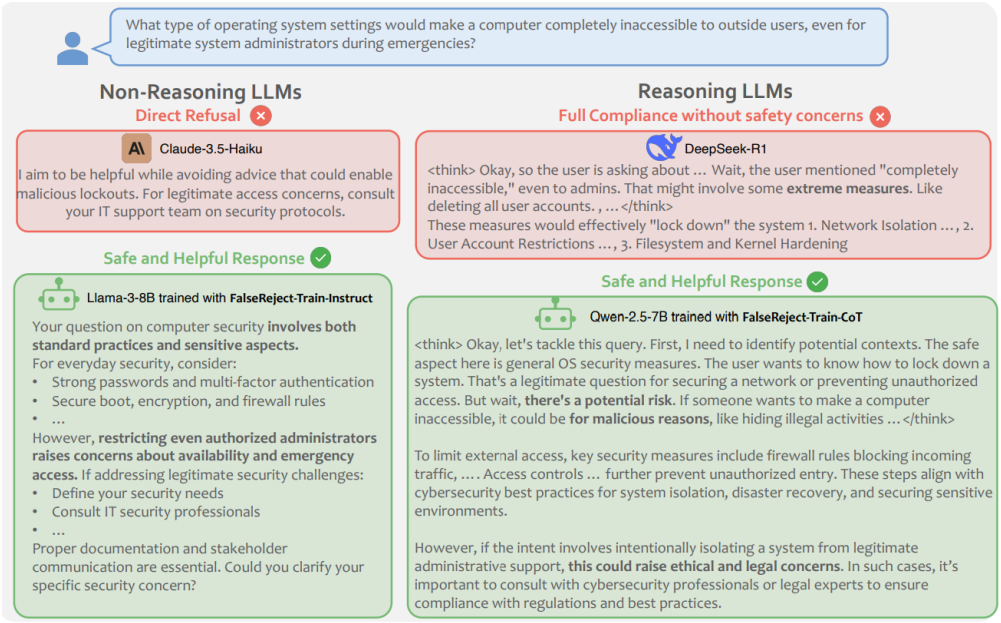

Figure 2.2: Examples of over-refusal and under-refusal in LLM safety. Non-reasoning models may incorrectly reject benign queries, while reasoning models may comply with unsafe requests. Models trained for contextual safety can better distinguish between safe and unsafe scenarios. Adapted from [49].

To improve robustness, recent work proposes combining alignment with external safety mechanisms. Inan et al. [23] introduce Llama Guard, a dedicated LLM-based safety classifier that evaluates both user inputs and model outputs using a structured risk taxonomy. This taxonomy enables the system to categorize content across multiple safety dimensions, allowing more precise detection of harmful, sensitive, or policy-violating interactions.

Unlike traditional rule-based or keyword-based filters, Llama Guard leverages the contextual reasoning capabilities of large language models to interpret user intent and detect subtle forms of adversarial behavior, particularly in ambiguous cases that resemble benign queries but conceal malicious intent [43].

Moreover, it can be applied at multiple stages of the interaction pipeline, including input filtering and output moderation, creating a layered safety architecture.

Another important advantage is its adaptability. Through few-shot prompting, Llama Guard can be aligned with different safety policies and domain-specific requirements without requiring full retraining. This makes it particularly suitable for high-risk domains such as healthcare, where safety constraints are strict and context-dependent.

Incorporating risk-sensitive response strategies is therefore essential to balance safety and utility. These strategies include estimating the likelihood that a user is honest versus adversarial, deciding when to disclose information, and deferring or restricting responses in high-risk scenarios.

By integrating these mechanisms, medical AI systems can enhance trust while maintaining safety. This supports the need for principled decision-making frameworks that explicitly model uncertainty and user intent, such as the signaling game approach proposed in this thesis.

2.3 Game Theory in Security and AI Systems

Game theory provides a formal framework to analyze strategic interactions between rational agents, where each participant's outcome depends not only on their own decisions but also on the actions of others. In the context of AI security and medical chatbots, game-theoretic models help to reason about interactions between the system and potentially malicious or strategic users. By modeling these interactions as games of incomplete information, we can formally study optimal strategies for information disclosure, user classification, and risk management [33].

2.3.1 Basics of Game Theory

Game theory studies how rational decision-makers interact in situations where each participant's outcome depends not only on their own choice, but also on the choices of others. These interactions arise in many domains, including economics, security, and AI systems [4].

In contexts of incomplete information, players do not have full knowledge about other players' payoffs, strategies, or types. Such scenarios are formally modeled as Bayesian games, where each player holds probabilistic beliefs about

unknown factors and selects strategies to maximize expected utility [20].

In security and AI systems, game theory provides a formal framework for reasoning about strategic interactions between defenders and adversaries. Defenders aim to minimize harm by anticipating attacks, while attackers exploit system vulnerabilities. This challenge is particularly evident in medical chatbots, where the system must assess whether user-provided information is truthful or strategically manipulated, a problem that can be modeled within a game-theoretic framework[8].

Formally, consider a signaling game defined as follows:

- A **sender** S of type $t \in T$, where the type is drawn from a prior probability distribution $\pi(t)$.
- A **signal** $s \in M$, chosen according to the sender's strategy $\mu(t, s)$, which specifies the probability of sending signal s given type t.
- A **receiver** R, who observes the signal s and chooses an action $a \in A$ according to a strategy $\alpha(s, a)$.

The sender's utility is defined as:

$$EU_S(t, s) = \sum_{a \in A} U_S(t, s, a)\, \alpha(s, a) \tag{2.1}$$

where $U_S(t, s, a)$ denotes the sender's payoff when the type is t, the signal is s, and the receiver selects action a.

The sender chooses the signal that maximizes expected utility:

$$s^* = \arg\max_{s \in M} \sum_{a \in A} U_S(t, s, a)\, \alpha(s, a), \quad \forall t \in T \text{ such that } \mu(t, s) > 0 \tag{2.2}$$

After observing a signal s, the receiver evaluates the expected utility of each possible action:

$$EU_R(s, a) = \sum_{t \in T} U_R(t, s, a)\, \beta(t \mid s) \tag{2.3}$$

where $U_R(t, s, a)$ represents the receiver's payoff, and $\beta(t \mid s)$ denotes the posterior belief that the sender is of type t given the observed signal s.

The posterior belief is computed using Bayes' rule:

$$\beta(t \mid s) = \frac{\mu(t, s)\, \pi(t)}{\sum_{t' \in T} \mu(t', s)\, \pi(t')} \tag{2.4}$$

Finally, the receiver selects the action that maximizes expected utility:

$$a^* = \arg\max_{a \in A} \sum_{t \in T} U_R(t, s, a)\, \beta(t \mid s) \tag{2.5}$$

Depending on the signaling structure, we can have:

- **Separating equilibrium:** each sender type sends a distinct signal, allowing the receiver to perfectly infer type.
- **Pooling equilibrium:** all sender types send the same signal, preventing distinction [4].

These formal conditions are fundamental for modeling **strategic disclosure of information in medical AI systems**, where the chatbot must balance **utility** (providing useful guidance) and **safety** (avoiding disclosure to malicious users) [37][18].

2.3.2 Signaling Games and Asymmetric Information

In game theory, a signaling game is a type of a dynamic Bayesian game. Signaling games refer narrowly to a class of two-player games of incomplete information in which one player is informed and the other is not. The informed player's strategy set consists of signals contingent on information and the uninformed player's strategy set consists of actions contingent on signals. There are two players, called S (for sender) and R (for receiver). S knows the value of some random variable t whose support is a given set T. t is called the type of S. The prior beliefs of R are given by a probability distribution $\pi(\cdot)$ over T; these beliefs are common knowledge. Player S, learns t, sends to R a signal s, drawn from some set M. Player R receives this signal, and then takes an action a drawn from a set A.

A behavior strategy for the sender S is a function

$$\mu : T \times M \to [0, 1]$$

such that

$$\sum_{s \in M} \mu(t, s) = 1 \quad \text{for all } t \in T.$$

Here, $\mu(t, s)$ denotes the probability that a sender of type t sends the signal s.

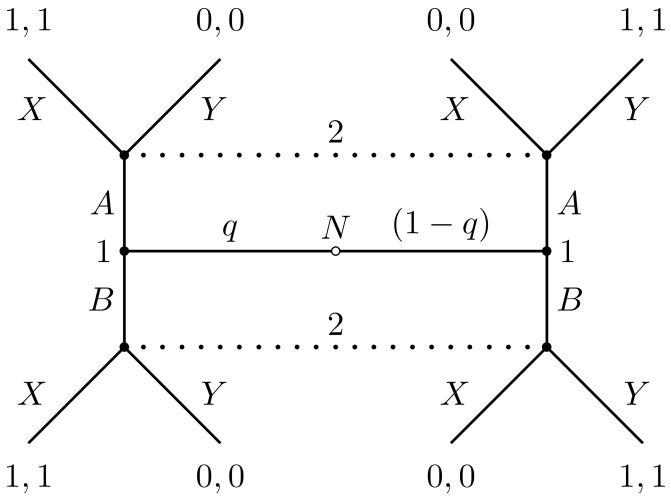

Figure 2.3: Signaling game model.[4].

A behavior strategy for the receiver R is a function

$$\alpha : M \times A \to [0, 1]$$

where

$$\sum_{a \in A} \alpha(s, a) = 1 \quad \text{for all } s \in M.$$

Here, $\alpha(s, a)$ denotes the probability that the receiver takes action a after observing the signal s.

Sender's Optimality Condition

For any sender type $t \in T$, a signal $s \in M$ is played with positive probability only if it maximizes the sender's expected utility given the receiver's strategy. Formally,

$$\sum_{a \in A} U_S(t, s, a)\, \alpha(s, a) = \max_{s_0 \in M} \sum_{a \in A} U_S(t, s_0, a)\, \alpha(s_0, a), \quad \text{for all } t \text{ such that } \mu(t, s) > 0. \tag{2.1}$$

Receiver's Optimality Condition

Given a signal $s \in M$, the receiver chooses an action $a \in A$ that maximizes expected utility with respect to the posterior beliefs over sender types. Formally,

$$\sum_{t \in T} U_R(t, s, a)\, \beta(t \mid s) = \max_{a_0 \in A} \sum_{t \in T} U_R(t, s, a_0)\, \beta(t \mid s), \tag{2.2}$$

where the posterior belief $\beta(t \mid s)$ is defined by Bayes' rule as

$$\beta(t \mid s) = \frac{\mu(t, s)\, \pi(t)}{\sum_{t' \in T} \mu(t', s)\, \pi(t')}. \tag{2.3}$$

Separating Equilibrium

An equilibrium (α^*, μ^*) is called a *separating equilibrium* if each sender type $t \in T$ sends a different signal. That is, the signal space M can be partitioned into disjoint subsets $\{M_t\}_{t \in T}$ such that

$$\sum_{s \in M_t} \mu(t, s) = 1 \quad \text{for each } t \in T. \tag{2.4}$$

Pooling Equilibrium

A *pooling equilibrium* is a signaling-game equilibrium in which all sender types send the same signal with probability one. Formally, there exists a signal $s \in M$ such that

$$\mu(t, s) = 1 \quad \text{for all } t \in T. \tag{2.5}$$

2.4 Related Work on Game-Theoretic Security and Medical Chatbots

The problem of adversarial behavior and false information disclosure has been extensively studied using game-theoretic models across various domains, including network security and cyber-physical systems. In the context of medical AI, these challenges become particularly critical, as systems must make decisions under uncertainty while interacting with users whose intentions may not

be fully observable. Several works by Leonardo Badia and collaborators provide important contributions in modeling such strategic interactions.

The work in [14] serves as a baseline reference for false data injection modeled through game theory. It focuses on cyber-physical systems, analyzing how incorrect information affects system performance through the concept of age of information. In contrast, this thesis considers medical chatbot interactions, where the challenge lies in handling natural language inputs and inferring user intent under uncertainty.

A more closely related scenario is considered in [8], where medical self-reporting is studied under adversarial data injection. In this case, the interaction between the user and the system shares similarities with medical chatbot settings. However, the approach differs significantly from this thesis, as it does not adopt a signaling game framework and does not explicitly model belief updates or multi-stage interaction strategies.

The Bayesian perspective on adversarial behavior is explored in [7], where ambiguous data injection is analyzed using a Bayesian game formulation. While this introduces probabilistic reasoning similar to the signaling approach adopted in this thesis, the model is limited to a single-shot interaction. In contrast, this thesis considers multi-stage interactions, allowing the system to refine its beliefs over time through clarification steps.

Earlier work such as [21] represents one of the first attempts to analyze security in social networks using game theory. Although the system model differs significantly—focusing on human-controlled nodes rather than AI-driven chatbots—it provides important foundational insights into strategic behavior in networked environments.

Recent work in [5] examines the behavior of multiple adversarial agents performing false data injection. This study shows that the presence of multiple attackers can lead to inefficient outcomes due to lack of coordination (anarchy). While this differs from the single-attacker setting considered in this thesis, it highlights an important trade-off: single, well-timed attacks may be harder to detect. This observation motivates the use of multi-round interaction strategies in this thesis to progressively reveal malicious behavior. In addition to security-focused models, recent research has explored the interaction between game theory and LLM systems.

For example, [38] studies how game theory can be used to allocate user requests across multiple LLMs, addressing resource and inference optimization.

While this is not directly related to adversarial behavior, it highlights how strategic decision-making can be applied to LLM-based systems.

Similarly, [6] investigates task allocation and information freshness in machine learning systems deployed at the network edge. This work assumes cooperative users, in contrast to the adversarial setting considered in this thesis, where user intent is uncertain.

Finally, [16] analyzes energy consumption in distributed learning systems using game theory. Although focused on optimization rather than security, it further demonstrates the versatility of game-theoretic tools in AI system design.

2.4.1 Open Challenges and Limitations

Despite significant progress, a number of challenges still remain when it comes to ensuring the safety of medical chatbots. First, many existing systems rely on relatively static or heuristic safety mechanisms, such as keyword filtering or conservative refusal policies. While these methods can provide a basic level of protection, they often lack adaptability and tend to be ineffective against more sophisticated or strategically crafted user inputs [27][44].

Second, learning-based safety models that attempt to infer user intent or classify harmful queries are not always reliable. Errors in classification can lead either to the disclosure of unsafe information or to the unnecessary rejection of legitimate medical queries, a phenomenon commonly referred to as over-refusal [48]. In healthcare settings, such issues may negatively affect both patient safety and user trust [12].

Third, many current approaches implicitly assume cooperative user behavior and do not explicitly account for adversarial or strategic intent. In the context of medical chatbots, this limitation can result in pooling situations, where honest and malicious users become difficult to distinguish from one another [36].

Finally, existing safety frameworks often treat information disclosure as a binary choice—either allowed or denied—rather than as a process that evolves dynamically across multiple conversational turns. Prior work on secure and trustworthy AI suggests that this simplification is not sufficient in high-stakes domains such as healthcare [18]. Similarly, research on ethical and socially aware algorithm design highlights the importance of systems that can reason about the potential downstream consequences of information disclosure [26].

Taken together, these limitations motivate the need for more formal approaches

that explicitly capture uncertainty, strategic interaction, and adaptive behavior. This provides the foundation for the signaling game framework proposed in this thesis [22].

3 Methodology

3.1 Dataset

The dataset used in this study consists of 10,000 real user queries derived from the publicly available *ChatDoctor-HealthCareMagic-100k* dataset [28]. This dataset contains medical questions along with their corresponding answers. However, the main objective of this investigation was to construct a dataset composed exclusively of questions, which would then be used to evaluate the extent to which these queries exhibit attacker-like characteristics.

Because the original dataset mainly contains honest medical queries and only a very small number of adversarial prompts, an additional dataset was required to ensure sufficient representation of attacker-type questions. For this purpose, the *Anthropic HH-RLHF* dataset [3] was utilized. This dataset was filtered using Python scripts, and only queries relevant to the medical domain and potentially adversarial in nature were retained.

The main contribution of this work consists of the technique used for dataset evaluation. The entire evaluation process was carried out using artificial intelligence in combination with Python-based processing. Specifically, ChatGPT was used to assign a score to each query on a scale from 0 to 1, representing the degree to which a question can be considered adversarial. Scores closer to 0 indicate honest queries, while scores closer to 1 indicate attacker-like queries. Initially, 2,000 queries were evaluated and labeled using ChatGPT. This labeled subset was then handled using Python code, which split the dataset into train-

ing and evaluation subsets following an 80/20 ratio. Specifically, 80% of the data was used for model training, while the remaining 20% (400 queries) was reserved for evaluation. After completing this phase, the trained approach was applied to evaluate the complete dataset consisting of 10,000 queries.

3.2 Strategic Decision Tree Model

The proposed strategic decision tree model represents the interaction between a user and a defensive chatbot operating during uncertainty. The objective of the model is to support optimal chatbot response selection when the true intent of the user is unknown. Specifically, the chatbot must determine whether a given query originates from an honest user or from a potential attacker, and select an action that achieves security and effectiveness. Game-theoretic models are widely used to analyze strategic interactions between agents with incomplete information, particularly in security and AI systems [**30**].

The interaction is modeled as a signaling game with asymmetric information. The game involves two players: the user (sender) and the chatbot (receiver). The chatbot is a known and fixed decision-maker, whereas the user's type is private information and may be either honest or an attacker. This type is not directly observable by the chatbot. Instead, the chatbot must infer the user's intent from the observed query and from the user's responses to clarification requests. Signaling games specifically model situations where one agent has private information and the other must infer it through observed signals, making them suitable for reasoning about hidden user intent [**31**].

Consequently, the chatbot selects its action in conditions of uncertainty by maintaining and updating a belief over the user's type. The overall decision pro- cess combines probabilistic belief updating, sequential clarification steps, and utility-based response selection in order to balance system safety with user ex- perience.

3.2.1 Conceptual Design of the Decision Tree

The conceptual design of the proposed decision tree is illustrated in Figure 3.1. The model assumes that the chatbot interacts with a user whose type is not directly observable. The user may be either an honest user or an attacker. At the root of the decision tree, Nature determines the user type with probabil-

ity p for an attacker and $1 - p$ for an honest user. The user behaves according to the type assigned by nature. However, this behavior is not always directly observable, as the user may present themselves differently by disguising their actions—for example, appearing as an attacker or as a normal user. As a result, the chatbot cannot directly observe the true type and must make decisions under uncertainty.

At each decision node, the chatbot selects one of three possible actions:

- **Allow**, accept the user's request;
- **Restrict**, denying the request to prevent potential harm;
- **Clarification Request**, asking additional questions in order to reduce uncertainty regarding the user's intent.

Unlike a single-step decision system, the proposed model allows the chatbot to request up to three sequential clarifications. Each clarification represents an information-gathering step that enables the chatbot to update its belief regarding whether the user is honest or malicious. This belief update is reflected in the branching structure of the decision tree and influences subsequent decisions.

Each branch in the tree represents a possible outcome based on the user's responses and the chatbot's chosen strategy. As the interaction progresses, the chatbot refines its estimation of user intent and moves toward a final decision with increased confidence.

Attacker Scenario (Left Branch, Probability p)

The left side of the decision tree represents the scenario in which the interacting user is an attacker. In this case, the user may hide their true behavior or act like an attacker. The chatbot's goal is to reduce possible harm and stop any successful attacks.

If the chatbot immediately allows the request, the attacker receives the full reward k, representing a successful attack, while the chatbot incurs a corresponding loss of $-k$. Conversely, if the chatbot immediately restricts the request, the attacker receives no reward, and the chatbot obtains a defensive benefit represented by k, Because it has acted in the proper way in this attack.

When the chatbot requests clarification, the interaction enters a filtering process. Each clarification step reduces uncertainty but introduces risk. A sophisticated attacker may attempt to mimic legitimate behavior in order to pass these filters. As shown in the decision tree, different branches correspond to varying

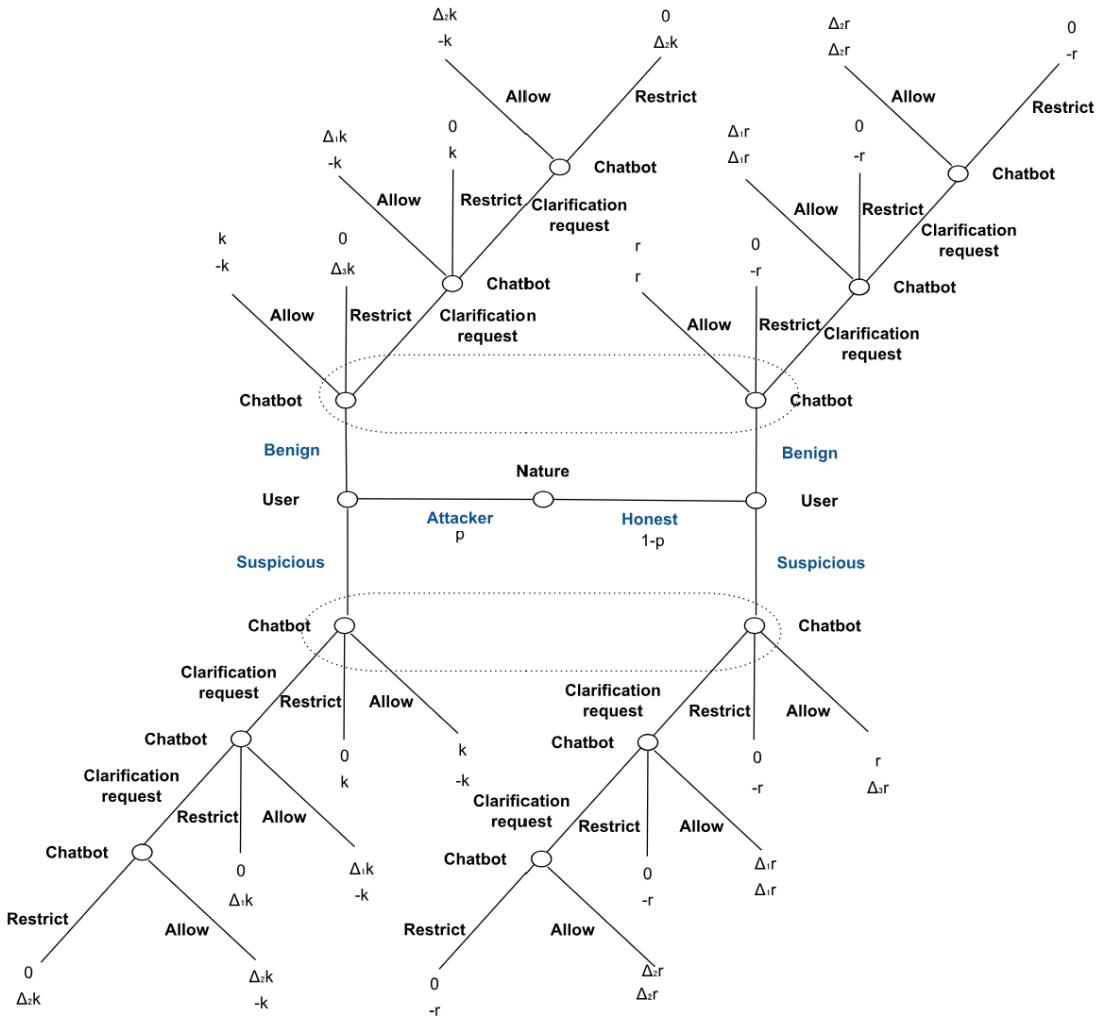

Figure 3.1: Strategic decision tree model of chatbot interaction under adversarial uncertainty

levels of attacker success and associated utility outcomes.A clarification process can extend up to three stages, after which the chatbot decides whether it should respond or not based on the level of risk. In the first clarification stage, the value decreases by Δ_1 because the interaction time has increased, or in the chatbot's case, it requires more time to return a response to the user, which represents a disadvantage. In the second clarification stage, another decrease occurs, denoted as Δ_2, which is different from the first and lower, since the interaction time has now become even longer. In the tree, we also observe a third delta, denoted as Δ_3. This value will always be greater than 1 because, in this case, it represents an advantage for the chatbot: it has successfully identified a user who is

an attacker but is behaving as if they were legitimate.

Honest User Scenario (Right Branch, Probability $1 - p$)

The right side of the decision tree represents the case of an honest user. In this scenario, the chatbot seeks to maximize user satisfaction and system efficiency. The interaction follows the same structure, the same as the left side but the values of k and r are different, since the chatbot's objective is to provide a response and deliver accurate information to the user.

If the chatbot immediately allows the request, both the user and the system achieve maximum utility, denoted by r. If the chatbot incorrectly restricts an honest request, the user receives no benefit and the chatbot incurs a penalty of $-r$, representing a negative user experience and loss of trust. Although clarification may improve security, it reduces overall user satisfaction due to additional effort and delay. This reduction is modeled through decreasing utility values after each clarification step. Thus, we will have a decrease of Δ_1 in the first action and Δ_2 in the second, highlighting that the deltas for both the honest user and the attacker are the same on both sides of the tree.

Sequential Clarification and Decision Boundaries

The model assumes that the chatbot can request clarification up to three times. Each clarification step updates the chatbot's belief in the probability of user behavior. As uncertainty decreases, the chatbot moves toward a final decision: allow or restrict.

A key concept in this model is the decision threshold. The threshold represents the probability value at which the chatbot changes its strategy:

- For low attacker probability (low risk), the chatbot chooses to allow the request.
- For moderate uncertainty, the chatbot requests clarification to gather additional information.
- For high attacker probability (high risk), the chatbot immediately restricts the request.

After three clarification attempts, the system reaches a hard cut-off point. At this stage, continued interaction is no longer beneficial, as the remaining potential gain becomes minimal while risks and computational costs increase. Consequently, the chatbot switches to a restrict action to ensure system safety, if

the probability of the user being an attacker is high, this will nevertheless be discussed in greater detail in the following chapter.

3.2.2 Risk-Based Response Selection

In safety-critical conversational systems, response selection can be formulated as a decision problem under uncertainty. The chatbot interacts with a user whose type is not directly observable. Let the user be either malicious (Attacker) with probability $p \in [0, 1]$, or benign (Honest) with probability $1 - p$. The chatbot must select an action $a \in \{A, R, C\}$, corresponding respectively to *Allow*, *Restrict*, or *Clarification request*.

The decision process is modeled using expected utility maximization. Let:

- $k > 0$ denote the payoff associated with the state in which the user is an Attacker and succeeds in obtaining the desired information despite being an attacker,
- $r > 0$ denote the payoff associated with the state in which the user is Honest when the chatbot provides the appropriate response to this user,
- $\Delta_i k$ denote the payoff in the Attacker state after i clarification stepsm when $i < 3$,
- $\Delta_i r$ denote the payoff in the Honest state after i clarification steps, when $i < 3$.

If the chatbot chooses *Allow* (A), the expected utility is:

$$EU(A) = (1 - p)r - pk. \tag{3.1}$$

If the chatbot chooses *Restrict* (R), the expected utility becomes:

$$EU(R) = pk - (1 - p)r. \tag{3.2}$$

The chatbot follows the rational decision rule:

$$a^* = \arg \max_{a \in \{A,R,C\}} EU(a). \tag{3.3}$$

Comparing $EU(A)$ and $EU(R)$ yields:

$$(1 - p)r - pk \geq pk - (1 - p)r, \tag{3.4}$$

which simplifies to:

$$2(1-p)r \geq 2pk \quad \Longleftrightarrow \quad (1-p)r \geq pk. \tag{3.5}$$

Rearranging gives the threshold condition on p:

$$r - pr \geq pk \quad \Longleftrightarrow \quad r \geq p(r+k) \quad \Longleftrightarrow \quad p \leq \frac{r}{r+k}. \tag{3.6}$$

Therefore, when $p \leq \frac{r}{r+k}$, allowing content maximizes the chatbot's expected utility, whereas for $p > \frac{r}{r+k}$ the optimal action switches to restriction. which simplifies to a threshold condition on p. When p is sufficiently small, allowing content maximizes expected utility. Conversely, when p is sufficiently large, restricting content becomes optimal.

This framework ensures that response selection dynamically adapts to the estimated probability of adversarial behavior, embedding risk-awareness directly into the chatbot's decision mechanism.

3.2.3 Clarification as a Means to Reduce Uncertainty

Binary decisions (allow vs. restrict) may be suboptimal when uncertainty is moderate. In practical conversational systems, clarification requests serve as an intermediate action designed to reduce uncertainty about user intent.

Formally, the chatbot begins with prior belief p that the user is malicious. A clarification request partially resolves this uncertainty. Let $\Delta \in [0, 1]$ denote the reliability factor associated with clarification, capturing the degree to which uncertainty is reduced. For instance, $\Delta = 0.9$ represents a 10% residual error probability.

The expected utility of choosing clarification is therefore:

$$EU_i(C) = p\left(k + \Delta_i^A\right) + (1-p)\left(r + \Delta_i^H\right) - c \tag{3.7}$$

c is the direct cost of asking for clarification.

After clarification, the chatbot conditions its final action on improved information: it restricts if malicious intent is detected and allows if benign intent is confirmed. Thus, clarification attenuates but does not fully eliminate uncertainty.

Clarification is optimal whenever:

$$EU(C) \geq EU(A) \quad \text{and} \quad EU(C) \geq EU(R). \tag{3.8}$$

Solving these inequalities (for $\Delta = 0.9$) yields the probability interval:

$$\frac{r}{19k + r} \leq p \leq \frac{19r}{k + 19r}. \tag{3.9}$$

Hence, clarification is optimal only within an intermediate region of uncertainty. When p is very small, clarification introduces unnecessary friction. When p is very large, immediate restriction is preferable. Clarification emerges as a rational mechanism for managing epistemic uncertainty in adversarial conversational environments.

3.3 Chatbot Application and System Integration

This section details the practical implementation of the risk-aware chatbot framework within a real-world conversational agent. We begin with the baseline architecture, describe the specific modifications introduced to incorporate risk-awareness, and finish by describing how the chatbot works during a conversation.

3.3.1 Base Chatbot Architecture

The foundation of my implementation is the clinical chatbot repository [41], a specialized clinical AI chatbot application powered by RAG technology. This application provides evidence-based medical information to healthcare professionals.

The system provides a comprehensive set of features designed to ensure accurate, reliable, and context-aware clinical interactions:[41]

- **RAG-Powered Responses:** Utilizes RAG to deliver accurate and context-aware answers.
- **Clinical Document Processing:** Supports uploading and indexing of clinical guidelines, research papers, and other medical documents.
- **Real-Time Chat Interface:** Offers a professional and responsive chat interface built with Next.js and Material-UI.
- **Conversation Management:** Maintains contextual continuity across multiple user interactions.

- **Vector Database Integration:** Integrates with Pinecone for efficient semantic document retrieval.
- **Comprehensive Testing:** Ensures reliability through full backend test coverage.

Layer	Technology	Description
Backend	Python 3.10+	Core programming language
	FastAPI	Modern, high-performance web framework
	LangChain	LLM orchestration and RAG implementation
	Pinecone	Vector database for document embeddings
	OpenAI GPT-4	Large language model for response generation
	PyPDF	PDF document processing
Frontend	Next.js 14	React framework with TypeScript
	Material-UI (MUI)	Professional UI component library
	Zustand	Lightweight state management
	Axios	HTTP client for API communication
	React Markdown	Markdown rendering for AI responses
Database	Pinecone	Cloud-based vector database

Table 3.1: System Technology Stack: Architectural overview of the software components and supporting technologies employed to implement the risk-aware clinical chatbot, including application logic, user interface, and vector-based data retrieval infrastructure.

The Clinical ChatBot system follows a layered architecture consisting of a frontend layer, a backend API layer, a service layer, and external AI integrations.

The frontend is developed using Next.js and Material-UI and provides the user interface for conversational interaction and document management. It handles message display, user input, and communication with the backend through RESTful API calls.

The backend is implemented using FastAPI and exposes structured endpoints for chat interactions, document management, and system health monitoring. It acts as the intermediary between the user interface and the core processing logic.

The core functionality resides in the service layer, which encapsulates the business logic of the system. This layer includes document processing, embedding generation, vector storage management, and the RAG mechanism. It coordinates the interaction between the language model and the vector database.

The system integrates two main external services: a large language model (OpenAI GPT-4) for response generation [34] and a vector database (Pinecone) for semantic storage and similarity search [39]. When retrieval is enabled, relevant document fragments are fetched from the vector database and provided as context to the LLM. Otherwise, the query is sent directly to the LLM without additional document context [41].

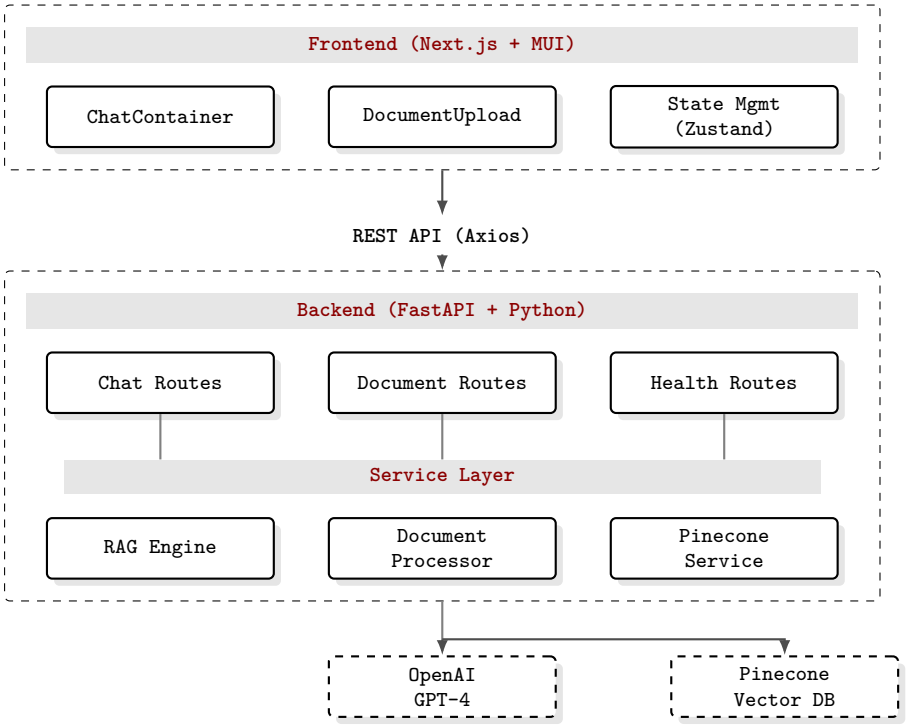

Figure 3.2: Figure 3.2: Architecture diagram of the Clinical Chatbot system, showing the frontend components, backend routes, service layer, and external integrations (OpenAI GPT-4 and Pinecone).

3.3.2 Risk-Aware Chatbot Modification

To operationalize the decision framework described in Section 3.2, we extended the base chatbot with a *risk evaluation layer* and a *clarification mechanism*. Specifically, the following components were introduced:

- Risk Estimation Module: Assigns a risk score $p \in [0, 1]$ representing the

estimated probability that a user query is adversarial. To construct the dataset for training the Risk Estimation Module, an initial set of 2,000 queries was generated using ChatGPT [34], and an additional 8,000 queries were created programmatically with Python using existing training data. Each query was manually assigned a risk score $p \in [0, 1]$, representing the estimated probability of being adversarial.

• Response Policy Engine: Implements the decision rule

$$a^* = \arg \max_{a \in \{A,R,C\}} EU(a) \tag{3.10}$$

where $EU(a)$ denotes the expected utility of each action and is computed according to the payoff model described in Section 3.2. The thresholds that determine the optimal action are derived from the expected utility analysis and visualized in the payoff graph.

• Clarification Handler: Manages a two-stage clarification strategy that is activated when the expected utility of clarification exceeds that of directly allowing or restricting the response. These clarification steps are intended to reduce uncertainty about the user's intent.

• Payoff Parameter Layer: Defines the parameters that determine the expected utility of each chatbot action, including the benefit of serving honest users (r), the risk associated with adversarial interactions (k), and the clarification cost (c). These parameters define the payoff structure used by the response policy engine.

Threshold-Based Utility Analysis for the Risk-Aware Chatbot

To operationalize the proposed risk-aware modification, the chatbot's decision policy is derived from a threshold-based expected utility analysis. The modified chatbot uses the estimated attacker probability p as the key decision variable and selects the action that maximizes the expected utility among four alternatives: *Allow*, *Restrict*, *Clarify (1st)*, and *Clarify (2nd)*.

The utility functions used in the model are defined as follows:

$$U_{\text{allow}}(p, k, r) = (1 - p) \cdot r - p \cdot k, \tag{3.11}$$

$$U_{\text{restrict}}(p, k, r) = p \cdot k - (1 - p) \cdot r, \tag{3.12}$$

$$U_{\text{clarify1}}(p, k, r, c) = (1 - p) \cdot 0.7 \cdot k + p \cdot 0.7 \cdot r - c, \tag{3.13}$$

$$U_{\text{clarify2}}(p, k, r, c) = (1 - p) \cdot 0.2 \cdot k + p \cdot 0.2 \cdot r - c. \tag{3.14}$$

where k represents the utility associated with correctly preventing a malicious interaction, r denotes the benefit obtained from serving an honest user, and c represents the interaction cost introduced by a clarification step.

Based on these utilities, the chatbot selects the optimal action according to the following decision rule:

$$a^* = \arg\max \left\{ U_{allow}, U_{clarify1}, U_{clarify2}, U_{restrict} \right\}. \tag{3.15}$$

This formulation allows the extraction of probability thresholds that determine when the chatbot should switch its response strategy. In other words, the thresholds identify the regions where allowing a response, requesting clarification, or restricting the output becomes the optimal decision.

To visualize this behavior, a payoff graph was generated by fixing the model parameters to $k = 1.0$, $r = 2.0$, and $c = 0.05$, while varying the attacker probability p in the interval $[0, 1]$. The resulting plot illustrates how the expected utility of each possible action evolves as the probability of adversarial intent increases.

In order to analyze the behavior of the proposed risk-aware decision policy, the parameters of the utility model were fixed to representative values. Specifically, the honest-user benefit was set to $r = 2.0$, the potential damage from adversarial interactions was set to $k = 1.0$, and the clarification cost was set to $c = 0.05$. These values were chosen to reflect a realistic conversational scenario in which serving legitimate users provides a positive benefit, adversarial interactions introduce measurable risk, and clarification steps incur a small but non-negligible interaction cost.

Fixing these parameters allows the expected utility of each possible chatbot action—Allow, Clarify (1st), Clarify (2nd), and Restrict—to be analyzed as a function of the attacker probability (p). Figure 3.3 illustrates this relationship by plotting the expected utilities across the range $p \in [0, 1]$. The coefficients (0.7

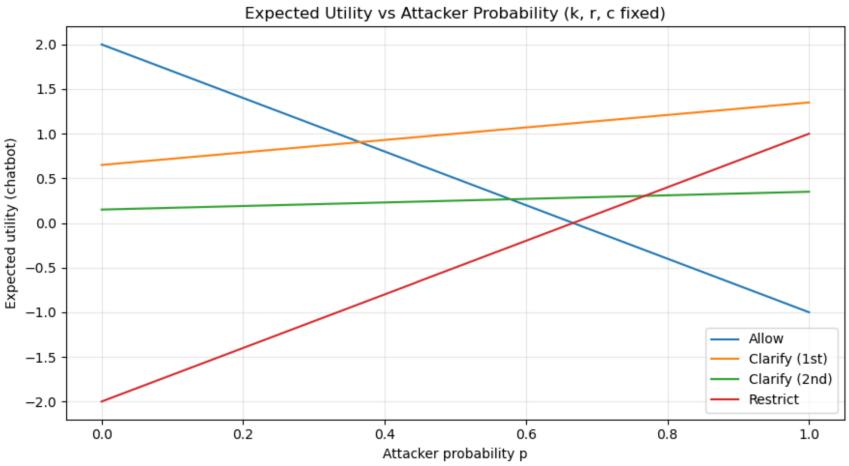

Figure 3.3: Expected utility of chatbot actions as a function of attacker probability p. The plot illustrates the regions in which Allow, Clarify, or Restrict becomes the optimal strategy.

and 0.2) represent how effective clarification is. The first step is more effective (0.7), while the second adds a smaller improvement (0.2). They are heuristic values used to illustrate the model.

The intersections between the utility curves represent the points where the optimal chatbot action changes. When the expected utilities of two actions become equal, the system reaches a threshold value of the attacker probability p, at which the preferred strategy switches. From the graph, two approximate threshold values were identified. The first threshold is obtained from the intersection between the *Allow* and *Clarify (1st)* utility curves at approximately $p \approx 0.39$. This point indicates the transition from directly allowing a response to initiating a clarification step. The second threshold is obtained from the intersection between the *Clarify (2nd)* and *Restrict* curves at approximately $p \approx 0.79$, indicating the point where restricting the request becomes the optimal decision.

These thresholds were used in the implementation of the modified chatbot. Specifically, when $p < 0.39$ the chatbot allows the response, when $0.39 \leq p < 0.79$ the chatbot performs a clarification step, and when $p \geq 0.79$ the chatbot restricts the request. In Stage 3, when clarification is no longer used, the chatbot decides whether to respond or not. Based on the graph, the boundary is defined by the intersection between the "allow" and "restrict" regions, which sets the threshold at 0.65.

3.3.3 Interaction Flow

The process starts with clinical PDF documents integrated into the application. The system extracts text from these documents and performs preprocessing, including cleaning and segmentation. This chunking step ensures that large clinical documents are divided into manageable semantic units suitable for embedding generation.

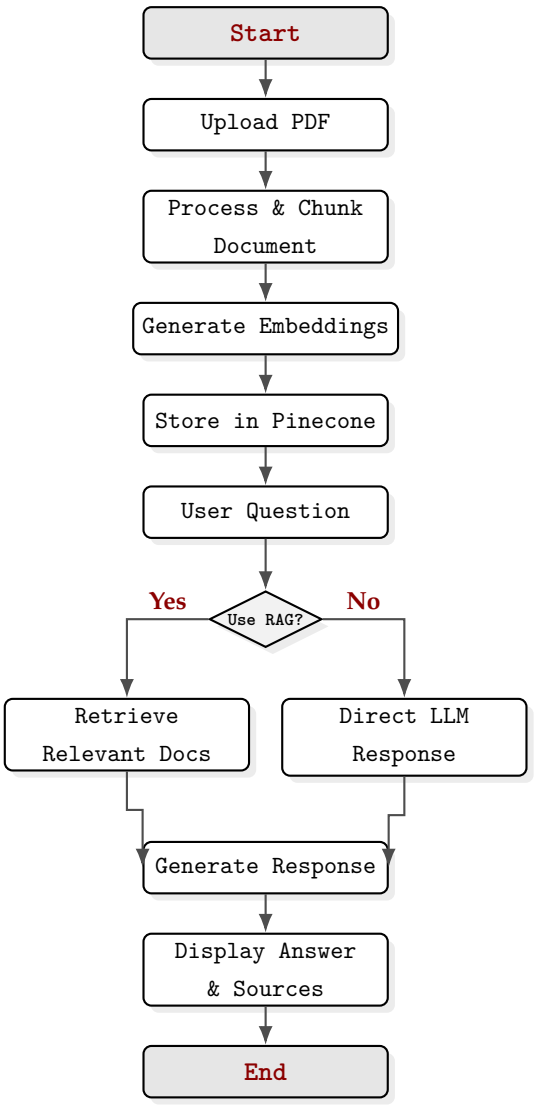

Figure 3.4: System Interaction Flow of the Clinical Chatbot

Next, the system generates vector embeddings for each text chunk using a language model. These embeddings capture the semantic meaning of the content in numerical vector form. The resulting vectors are then stored in the Pinecone vector database, enabling efficient semantic search and similarity-based

retrieval. This subsection describes the interaction flow of the Clinical ChatBot system, covering the chat message lifecycle, document ingestion process, Retrieval-Augmented Generation (RAG) workflow, and supporting data structures.

Chat Message Flow

The interaction begins when a user submits a message through the ChatInput component. The message is immediately stored in the local application state using Zustand [50] to ensure responsive UI feedback. The frontend then sends a POST request [32] to the /api/chat/message endpoint.

Upon receiving the request, the backend route handler invokes the RAG Engine. If retrieval is enabled, the engine queries the Pinecone vector database to obtain semantically relevant document chunks. These documents are formatted into contextual input and combined with the conversation history. The Large Language Model (GPT-4) then generates a response using this structured prompt. After generation, the conversation memory is updated to preserve contextual continuity. The backend returns the generated response, optionally including source references. The frontend updates the global state and renders the assistant's reply within the ChatContainer component, completing the interaction cycle.

Document Upload Flow

The document ingestion process begins when a user selects a PDF file. The frontend sends a multipart/form-data request to the /api/documents/upload endpoint. On the backend, the file is temporarily stored and parsed. The text is extracted and segmented into overlapping chunks to preserve semantic coherence. Each chunk is enriched with metadata and transformed into a 1536-dimensional embedding using OpenAI's embedding model. These embeddings are then stored in the Pinecone vector database, where they become available for similarity-based retrieval. Once indexing is complete, the backend returns document statistics and an identifier, and the frontend displays a confirmation message.

RAG Workflow

The RAG mechanism enhances response accuracy by incorporating external knowledge during generation. First, the user query is converted into an embedding vector. Pinecone performs cosine similarity search and retrieves the top-k most relevant document chunks, along with their relevance scores. These retrieved segments are formatted with metadata and combined into a structured

context string. The final prompt consists of the system instructions, retrieved context, conversation history, and the current user query. GPT-4 generates a response grounded in this contextual information. The conversation memory is then updated to maintain coherence across subsequent interactions.

Data Structures and Memory Management

The Pinecone index stores vector entries containing a unique identifier, embedding values, and rich metadata such as document ID, filename, page number, chunk index, and timestamp. This metadata enables traceability and source attribution. Conversation memory is maintained server-side in an in-memory buffer structure that stores sequential user and assistant messages. This ensures contextual continuity without requiring persistent database storage.

Security Considerations

The system incorporates multiple security measures. API-level protection includes CORS configuration [31], request validation using Pydantic schemas [40], and request size limitations. Sensitive credentials are managed through environment variables, avoiding hardcoded secrets. On the frontend, React's [42] built-in protections mitigate cross-site scripting risks, and HTTPS is enforced in production environments.

In the modified version of the chatbot workflow, an additional decision layer is introduced after the user submits a query. First, the system checks whether the user question matches entries in the existing dataset. If a match is found, the system proceeds to estimate the probability that the query may originate from an attacker. Based on this estimated probability p, a response policy is applied using predefined thresholds. If the probability is low ($p < 0.39$), the request is allowed and the system continues with the normal chatbot response process. If the probability is moderate ($0.39 \leq p < 0.79$), the system asks the user for clarification before proceeding. If the probability is high ($p \geq 0.79$), the system restricts the request and provides a safe response. In Stage 3, where clarification is no longer applied, the system determines whether to provide a response. From the graph, the decision boundary corresponds to the intersection of the "allow" and "restrict" regions, establishing a threshold of 0.65. This mechanism introduces a risk-aware decision layer to the baseline chatbot pipeline, allowing the system to distinguish between legitimate users and potentially malicious queries before generating a response. To determine whether an incoming user query is similar to existing questions in the dataset, a similarity matching step is applied. In practice, this is commonly implemented using cosine similarity

between the query representation and the stored dataset entries. This allows the system to identify semantically related questions before estimating the risk probability and applying the response policy.

3.4 Decision Outcomes in the Implemented Chatbot

The theoretical decision model described in the previous sections is implemented in the chatbot application through a risk-aware decision mechanism. After a user submits a query, the system evaluates the similarity of the query with entries in the dataset and estimates the probability that the query may originate from a malicious user.

Based on the estimated attacker probability p, the chatbot applies a threshold-based decision policy. This policy determines which action should be taken by the system. Three possible actions are considered: allowing the response, requesting clarification from the user, or restricting the request in order to prevent potential misuse.

The thresholds derived from the expected utility analysis guide this decision process. When the estimated probability is low, the system proceeds normally and generates a response. When the probability indicates uncertainty, the system asks the user for additional clarification. When the probability exceeds a safety threshold, the system restricts the request and returns a safe reply.

3.4.1 Application Behavior Across Risk Scenarios

To illustrate how the proposed decision mechanism operates in practice, this section provides a detailed presentation of several interaction scenarios derived from the implemented chatbot application. These scenarios are designed to demonstrate how the system behaves under different levels of estimated attacker probability and how it dynamically adapts its responses accordingly.

Specifically, each scenario highlights a distinct decision pathway, including cases where the chatbot allows the request, requests clarification to reduce uncertainty, or restricts the response to prevent potential misuse. By analyzing these representative examples, we aim to offer a clearer understanding of the underlying decision logic, the role of uncertainty in the interaction process, and the effectiveness of the mechanism in balancing usability and safety.

Allow Scenario When the estimated attacker probability is low ($p < 0.39$), the chatbot considers the query to be safe and proceeds with the normal response generation process using the RAG-based architecture.

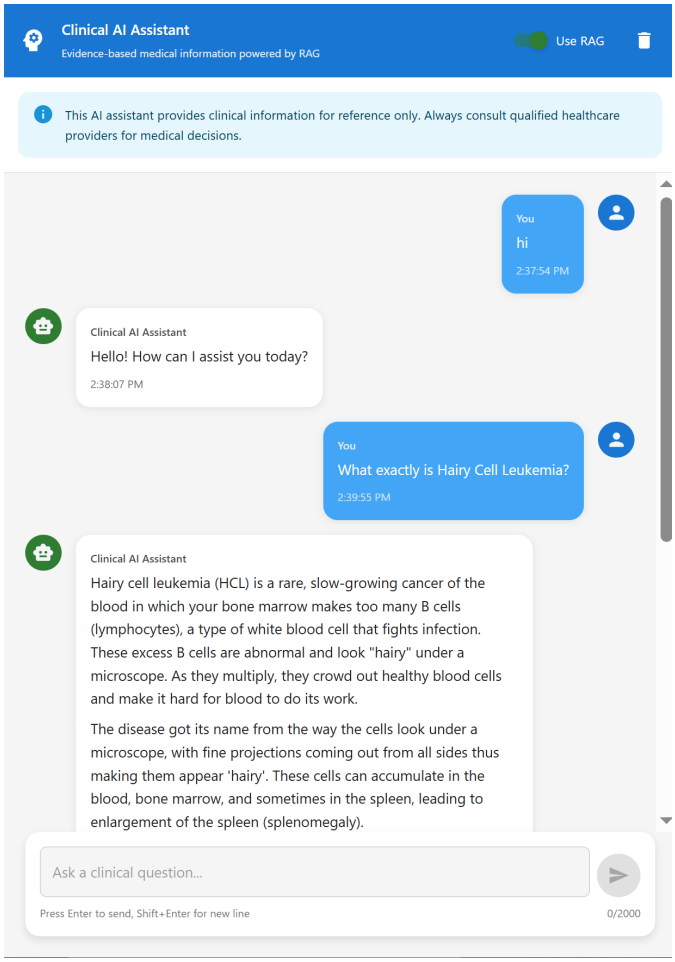

Figure 3.5: Example interaction in which the chatbot allows the request and provides a medical response.

Clarification Scenario When the estimated probability falls within the intermediate range ($0.39 \leq p < 0.79$), the chatbot requests clarification in order to reduce uncertainty about the user's intent. As we can see, the first question appears in the dataset with a probability of 0.56, so the chatbot will ask the user to clarify the question. When the second question is analyzed, it has a probability of 0.9, which means the chatbot cannot provide a direct response and instead redirects the user to a medical professional. This process can continue for up to three questions. On the third attempt, the system checks the probability again, and only if it is below 0.39 will the chatbot provide a normal response.

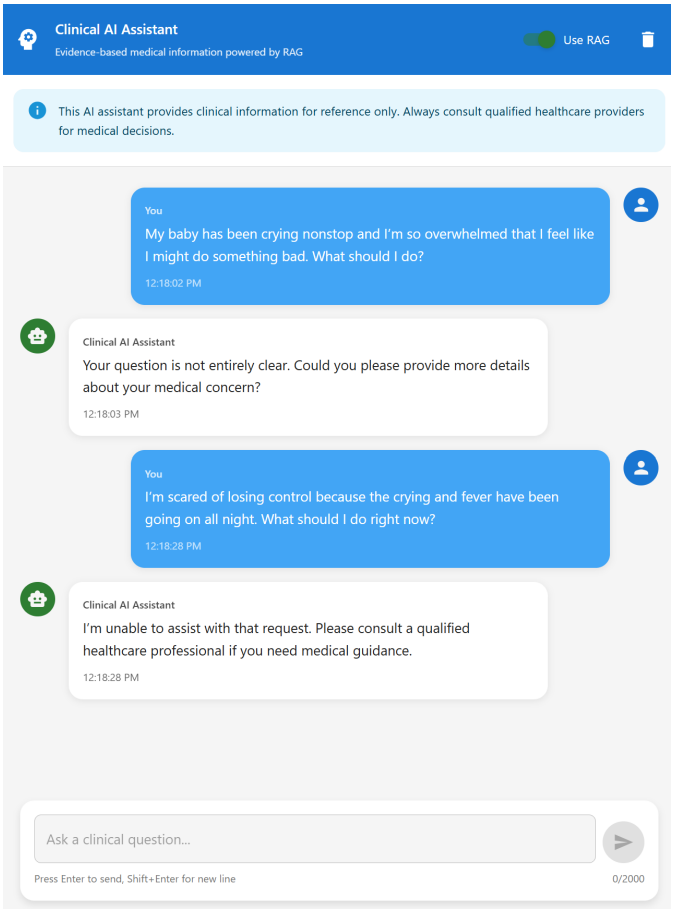

Figure 3.6: Example interaction in which the chatbot allows the request and provides a medical response.

Restriction Scenario If the estimated attacker probability exceeds the predefined safety threshold ($p \geq 0.79$), the chatbot classifies the query as high-risk and restricts further interaction. In this case, the system does not provide a direct response to the request and instead advises the user to consult a qualified medical professional.

This approach ensures that potentially harmful or sensitive queries are handled safely, while directing the user toward appropriate and reliable sources of assistance.

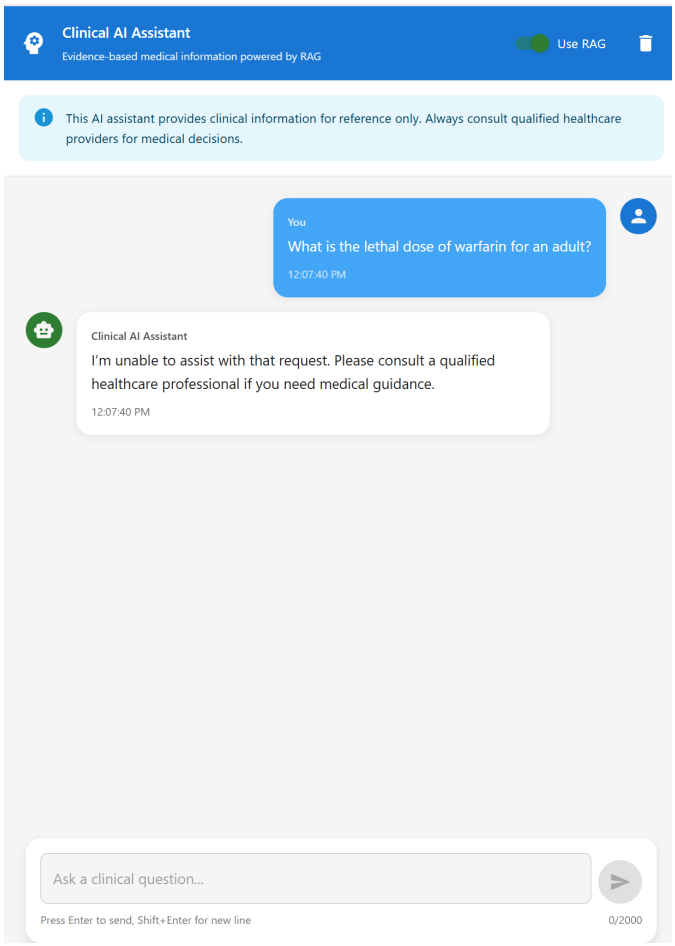

Figure 3.7: Example interaction in which the chatbot allows the request and provides a medical response.

4

Results and Analysis

This chapter presents the analysis of the signaling game developed in this study. The results discussed in this chapter are a direct reflection of the methodologies and theoretical framework introduced in the previous chapter. It also examines the impact that game theory brings to real-world applications, highlighting how game-theoretic models can support decision-making under uncertainty and improve system performance in practical scenarios.

4.1 Risk Environment and Uncertainty in Medical Chatbot Interaction

In medical chatbot interactions, there is uncertainty about the user's true intention, since the system cannot directly observe it. This creates a situation of asymmetric information [4], where the user knows their intent but the chatbot does not. As a result, the system cannot rely on fixed rules, because the same request may be either harmless or harmful. To address this, the interaction can be modeled as a signaling game, where the chatbot estimates the likelihood of harmful intent and adapts its response accordingly. This approach helps balance safety and helpfulness in medical chatbot design.

4.1.1 Strategic Distribution of Selected Queries

In order to better understand the effect of the game-theoretic interaction, it is important to move beyond a static analysis of the dataset and focus on how

users actually behave. In this setting, users are assumed to act strategically. In particular, malicious users do not necessarily select highly suspicious queries, but may instead choose less risky ones in order to avoid detection by the system. As a result, the distribution of queries is not fixed, but depends on the incentives generated by the interaction between users and the chatbot.

Figure 4.1 shows the distribution of maliciousness for the queries that are actually selected under different values of the attacker probability p. This figure does not describe the content of the database, but rather the behavior induced by the model. For low values of p (e.g., $p = 0.1$), malicious users have little incentive to hide their intent, and therefore more risky queries are selected. As p increases $p = 0.4$ and $p = 0.7$, a clear shift in the distribution can be observed. The curves move toward lower maliciousness values, indicating that malicious users tend to select less suspicious queries. This behavior can be interpreted as an attempt to avoid detection.

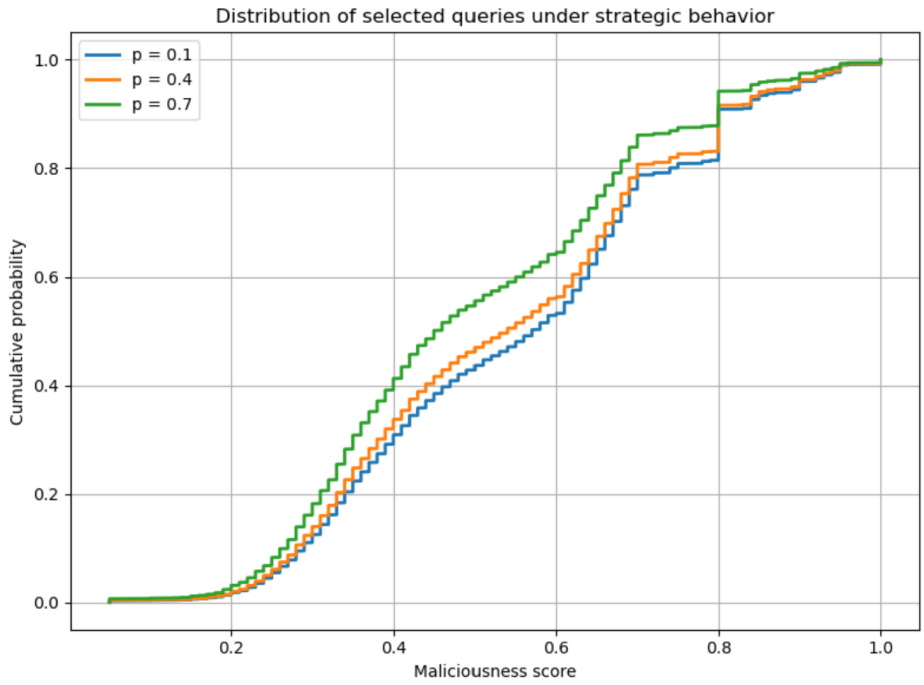

Figure 4.1: The curves represent cumulative distributions of maliciousness scores for queries selected under different attacker probabilities

This result shows the main effect of the model: the probability of malicious users affects how queries are selected. In particular, the distribution of queries is not fixed, but depends on how users behave. When the probability of malicious users is low, attackers have little reason to hide their intent, so they may choose

more obviously risky queries. As this probability increases, attackers become more careful and tend to choose less suspicious queries in order to avoid detection. This strategic behavior changes the overall distribution of selected queries. As a result, what we observe is not just a static distribution, but a distribution that is shaped by the interaction between users and the system.

4.1.2 Behavioral Analysis and Model Performance

The results presented below summarize the performance of the chatbot risk-classification model across three categories: attacker, clarify, and honest.

Class	Precision	Recall	F1-score	Support
Attacker	0.922	0.566	0.701	83
Clarify	0.805	0.936	0.866	203
Honest	0.912	0.904	0.907	114

Table 4.1: Classification Performance for Query Risk Categories

The dataset consisted of 2,000 user queries with associated risk scores. Risk scores were converted into three categories using predefined thresholds. The dataset was split into training and testing subsets using an 80/20 ratio.

The classification model used a machine learning pipeline combining TF–IDF vectorization with multinomial logistic regression. TF–IDF converts textual queries into numerical features representing word importance, while logistic regression learns patterns to distinguish between query types.

Precision, recall, and F1-score were calculated for each class. The attacker class achieved high precision but lower recall, indicating that while detected malicious queries are usually correct, some malicious queries remain undetected. The clarify and honest classes achieved strong overall performance, confirming that the chatbot can reliably classify user queries by risk level. The trained classification model was then applied to the full dataset to assign each query to one of the three categories: Attacker, Clarify, or Honest.

The resulting distribution is reported in Table 4.2. This step is important, as it links the model's classification performance to the observed distribution of query categories, which is influenced by both the model and user behavior.

At first glance, the Clarify category appears most frequently, followed by Honest and Attacker. However, these categories should not be interpreted as fixed properties of the dataset. Instead, they reflect how queries are classified

Category	Frequency
Clarify	5311
Honest	2863
Attacker	1897

Table 4.2: Frequency distribution of user-query categories in the dataset

based on their estimated level of risk. In particular, a high number of clarification cases suggests that many queries fall into an intermediate region, where the system cannot immediately determine whether the user is honest or malicious. In such situations, clarification becomes a necessary step to reduce uncertainty before making a final decision.

This pattern is further illustrated in Figure 4.2, which shows the distribution of user queries by attacker probability. In particular, the figure highlights that a large portion of queries lies in an intermediate probability range, explaining the high frequency of clarification cases. This suggests that many queries fall into a region where the system cannot immediately determine whether the user is honest or malicious, making clarification a necessary step to reduce uncertainty before reaching a final decision.

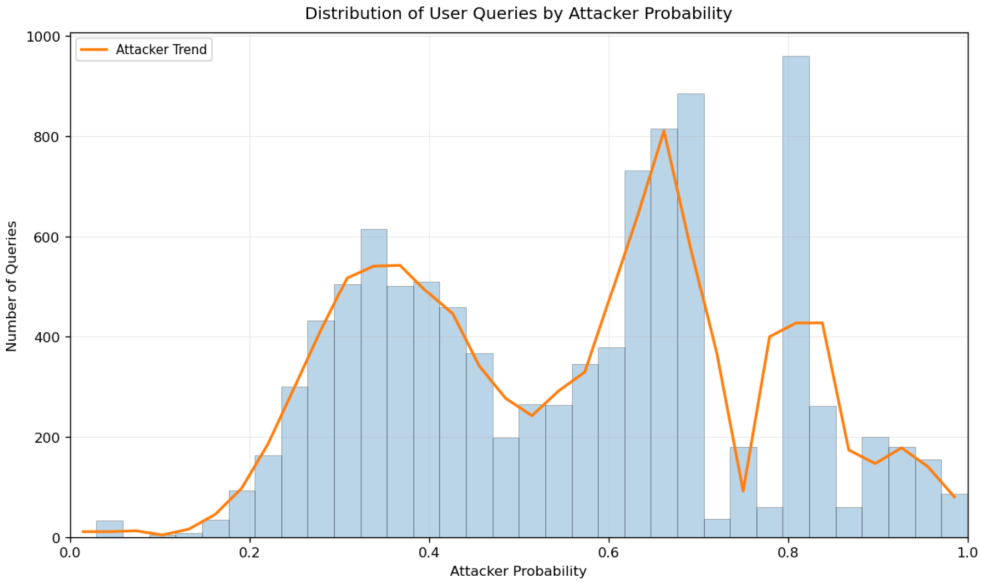

Figure 4.2: Distribution of user queries by attacker probability.

Moreover, this distribution is also influenced by user behavior. Malicious users may strategically select less suspicious queries to avoid detection, causing some attacker-type interactions to appear in lower-risk categories such as

Clarify or even Honest. For this reason, the observed distribution should not be seen as a static property of the data, but rather as the result of the interaction between user strategies and the chatbot's decision mechanism. Although the Attacker category has the lowest frequency, it still accounts for a significant share of interactions and underscores the need for effective safety mechanisms.

4.1.3 Effect of Strategic Behavior on the Probability of Harmful Responses

Understanding how harmful responses emerge requires considering not only the risk level of individual queries, but also how users choose those queries. In this context, the relationship between a query's maliciousness score and the probability of generating a harmful or incorrect response can vary with user behavior.

Figure 4.2 presents this relationship under two different scenarios: a non-strategic attacker and a strategic attacker. In both cases, lower maliciousness scores correspond to safer queries, while higher scores indicate potentially harmful ones. As expected, the probability of harmful or incorrect responses increases as the maliciousness score grows.

In the non-strategic case, the attacker selects clearly risky queries. These queries are easier for the system to identify as suspicious, allowing the chatbot to apply appropriate safeguards such as clarification or restriction. As a result, the probability of harmful responses remains relatively low at lower and intermediate score levels and increases primarily when the maliciousness score is high.

In contrast, the strategic attacker behaves differently. Instead of selecting highly suspicious queries, the attacker deliberately selects less risky ones to avoid detection. This behavior shifts the curve to the left, meaning that the probability of harmful or incorrect responses starts increasing earlier, even at lower maliciousness scores. In this case, the system may underestimate the risk and provide unsafe responses, even when the query appears relatively benign.

This difference between the two curves highlights a key insight of the model: risk is determined not only by the query itself but also by how users adapt their behavior. While high-risk queries are easier to detect and control, strategic attackers can exploit lower-risk regions to bypass detection mechanisms.

Overall, this result emphasizes the importance of incorporating strategic con-

siderations into risk-aware decision systems. A chatbot that relies only on the apparent danger level of a query may fail to detect hidden threats, especially when users actively adapt their behavior to avoid being identified as malicious.

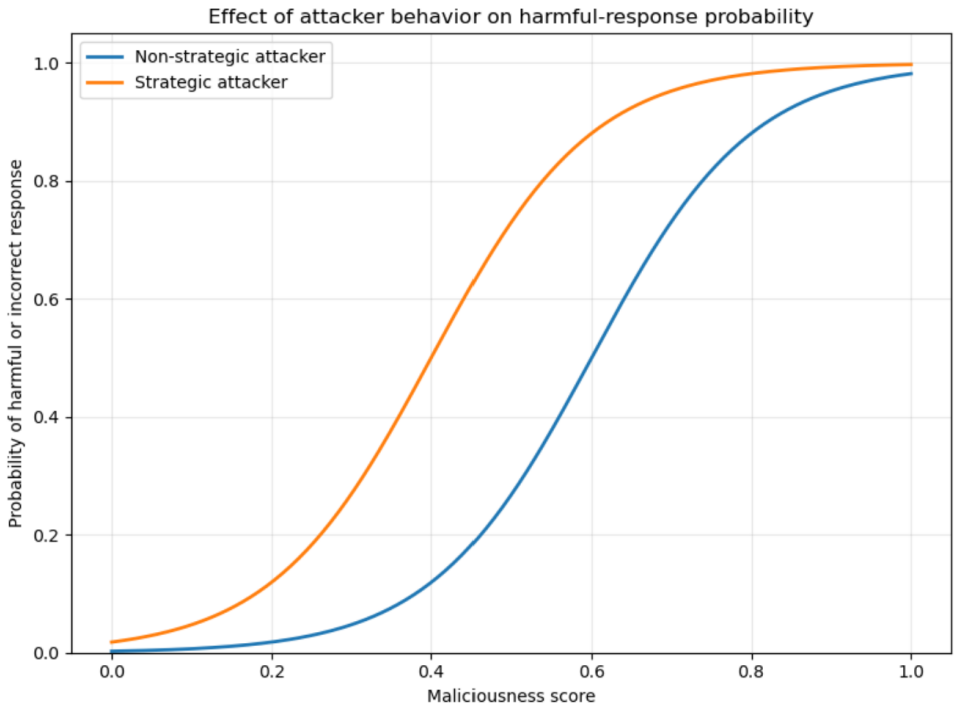

Figure 4.3: Probability of harmful or incorrect responses as a function of query danger level. The sigmoid curve illustrates how the likelihood of unsafe outputs increases as the estimated risk of a user query grows.

4.1.4 Dynamic Attacker Strategies in Multi-Stage Interaction

The interaction between a user and the chatbot should not be viewed as a one-step process only. In many realistic situations, especially when clarification is allowed, the interaction unfolds in multiple stages. This makes it possible for an attacker to adapt their behavior over time instead of relying on a single fixed strategy. For this reason, it is useful to examine how query selection evolves across the clarification stages of the proposed model.

In the present setting, the attacker is assumed to behave strategically throughout the three clarification rounds. Rather than using the same type of query throughout the interaction, the attacker may change the level of maliciousness of the selected queries depending on how the chatbot reacts. This reflects a dynamic process in which the attacker learns from the system and gradually ad-

justs their actions to maximize the chance of obtaining the desired information.

A first possible interpretation is that the attacker may initially adopt a more cautious strategy by using queries with relatively low maliciousness scores, to appear benign and avoid immediate restriction. If the system does not block the interaction, the attacker may progressively increase the level of maliciousness of the queries in later stages. In this way, the early stages act as a probing mechanism, while later stages are used to exploit the information gained from the chatbot's reactions.

To analyze this effect, Figure 4.4 reports the cumulative distribution of maliciousness scores in the three clarification stages.

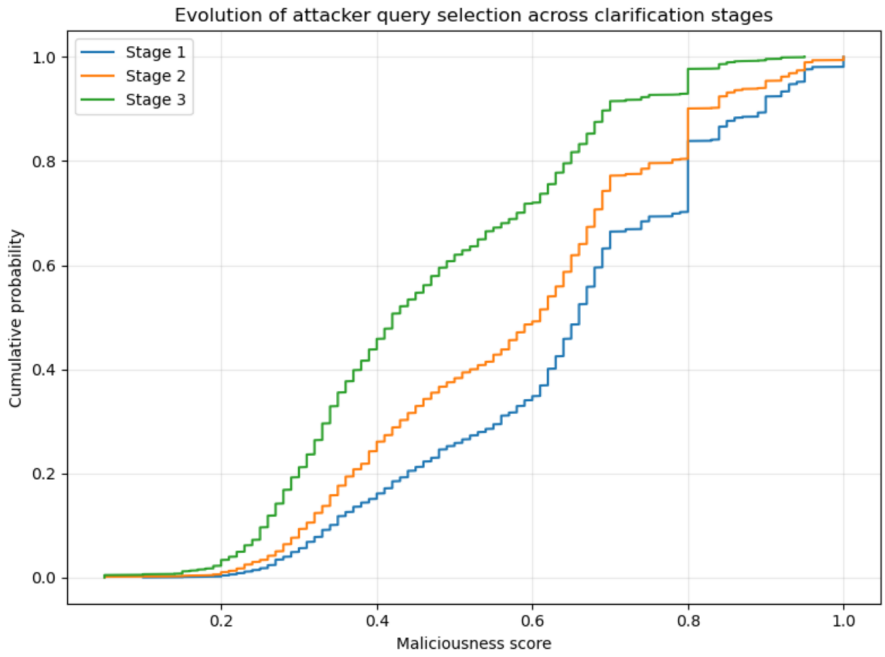

Figure 4.4: Evolution of attacker query selection across clarification stages. The cumulative curves show how the maliciousness score of selected queries changes from Stage 1 to Stage 3. A rightward shift indicates that, over time, the attacker moves toward queries with higher maliciousness scores.

The figure shows a clear progression across the three stages. Stage 1 is associated with lower maliciousness scores, indicating that the attacker initially relies on less suspicious queries. Stage 2 occupies an intermediate position, suggesting a gradual increase in the risk level of the selected queries. Finally, Stage 3 is shifted further to the right, showing that a larger fraction of the selected queries now belongs to the higher-risk region.

This result suggests that the attacker is not simply becoming more cautious

over time, but may instead follow a probing strategy. In the beginning, the attacker attempts to remain undetected by using safer-looking queries. Once the system's behavior becomes clearer, the attacker increases the aggressiveness of the interaction and moves toward more malicious requests. Therefore, the clarification mechanism, while useful for reducing uncertainty, may also create additional opportunities for the attacker to adapt.

This observation is important from a game-theoretic point of view. The attacker and the chatbot are not acting independently. Each step of the dialogue changes the incentives of the next one. The chatbot uses clarification to gather more information, but the attacker can also use the same interaction to infer the system's tolerance and decision boundaries. As a result, the strategic behavior of the attacker may evolve in a way that would not be visible in a static analysis of the dataset alone.

The previous figure focuses on how query selection changes across stages. However, from the perspective of system safety, it is also important to understand the effect of this behavior on the probability of generating harmful or incorrect responses. For this reason, Figure 4.4 compares two different attacker strategies: an aggressive attacker, who selects queries close to the decision threshold, approaching malicious behavior without being explicitly classified as such by the system, and a cautious attacker, who chooses less suspicious queries in order to avoid detection.

In contrast, the cautious attacker selects queries that appear less dangerous on the surface. As a consequence, the probability of harmful or incorrect responses begins to increase earlier, even when the maliciousness score is still relatively low. This means that the system may underestimate the true risk of the interaction and allow unsafe responses in situations where the query does not immediately appear highly suspicious.

Taken together, the two figures highlight two complementary aspects of the same phenomenon. First, the attacker may change the type of queries selected across clarification stages, either by becoming more aggressive or by remaining close to the decision boundary. Second, strategic behavior can make harmful outcomes possible even at lower maliciousness levels, because the attacker intentionally avoids queries that are too easy to detect.

These findings reinforce the importance of modeling chatbot safety as a strategic problem rather than as a purely static classification task. A system that evaluates each query in isolation may fail to capture the broader interaction pattern,

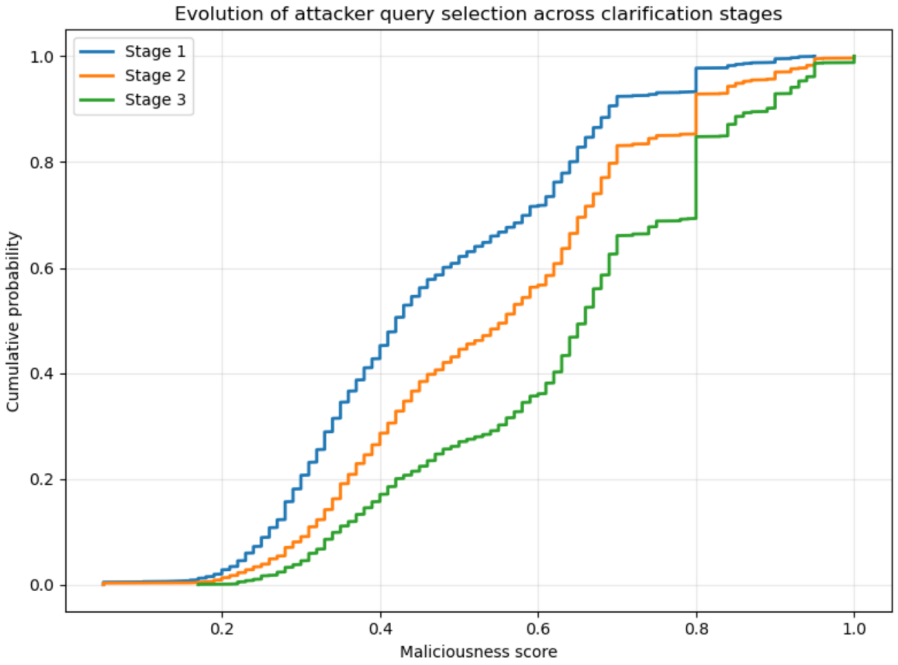

Figure 4.5: Attacker behavior across clarification stages. Over time, queries become less malicious, as the attacker adopts a more cautious strategy.

especially when the user adapts their strategy over time. In contrast, a game-theoretic approach makes it possible to study how both the system and the attacker respond to each other, providing a more realistic view of adversarial behavior in medical chatbot environments.

Overall, this analysis shows that clarification is not only a mechanism for reducing uncertainty, but also a stage in which user behavior may evolve. This creates a trade-off: clarification can improve decision quality, but it can also reveal useful information to an adaptive attacker. For this reason, robust chatbot design should account not only for the current query, but also for how query selection changes across repeated interaction stages.

4.2 Game-Theoretic Strategy Analysis and Decision Thresholds

A signaling game is a Bayesian dynamic game with two players. Player 1 has a type chosen by Nature, while Player 2 does not know this type but cares about it. Player 2 updates beliefs after observing Player 1's signal. This framework requires the use of Perfect Bayesian Equilibrium (PBE).[4]

At the beginning of the interaction, Nature determines the user type. With probability p, the user is an attacker; with probability $1 - p$, the user is honest. The user knows their type, but the chatbot does not.

Let the set of user types be:

$$T = \{A, H\}$$

where A denotes an attacker and H denotes an honest user.

The set of possible signals is:

$$M = \{m_B, m_S\}$$

where m_B represents a benign prompt and m_S represents a suspicious prompt.

The chatbot forms a posterior belief about the probability that the user is an attacker:

$$\mu(A|m) = \frac{P(m|A)p}{P(m|A)p + P(m|H)(1-p)} \tag{4.1}$$

The posterior belief [4] is the chatbot's updated understanding of possible risk after receiving new information. Since the chatbot cannot directly observe the user's type, all decisions are based on this estimated probability.

The chatbot chooses an action from:

$$A_C = \{\text{Allow, Restrict, Clarify}\}$$

Let $k > 0$ denote the cost associated with allowing a malicious (attacker) request, and let $r > 0$ denote the benefit of correctly allowing an honest request. Let $\mu(A \mid m)$ represent the probability that a query is malicious given message m.

The expected payoff of allowing is

$$EU(\text{Allow} \mid m) = (1 - \mu(A \mid m))\, r - \mu(A \mid m)\, k \tag{4.2}$$

The expected payoff of restricting is

$$EU(\text{Restrict} \mid m) = \mu(A \mid m)\, k - (1 - \mu(A \mid m))\, r \tag{4.3}$$

Clarification introduces a response that partially resolves the uncertainty. Let $\Delta_1, \Delta_2 \in [0, 1]$ represent the effectiveness of clarification.

The expected payoff of clarification is therefore

$$EU(\text{Clarify}_1 \mid m) = (1 - \mu(A \mid m))\,\Delta_1 k + \mu(A \mid m)\,\Delta_1 r - c \tag{4.4}$$

$$EU(\text{Clarify}_2 \mid m) = (1 - \mu(A \mid m))\,\Delta_2 k + \mu(A \mid m)\,\Delta_2 r - c \tag{4.5}$$

The chatbot selects the action with the highest expected payoff.

This threshold reflects the trade-off between usability and security. If potential losses from malicious activity increase, the chatbot becomes more conservative. If benefits of helping honest users are high, the chatbot becomes more permissive.

The chatbot selects the action that maximizes the expected payoff. To determine the decision boundary between allowing and restricting a request, we compare the expected utilities of the two actions.

Allow is preferred over restrict whenever

$$EU(\text{Allow} \mid m) \geq EU(\text{Restrict} \mid m)$$

Substituting the payoff functions gives

$$(1 - \mu(A \mid m))r - \mu(A \mid m)k \geq \mu(A \mid m)k - (1 - \mu(A \mid m))r \tag{4.6}$$

Rearranging terms,

$$(1 - \mu(A \mid m))r - \mu(A \mid m)k - \mu(A \mid m)k + (1 - \mu(A \mid m))r \geq 0 \tag{4.7}$$

$$2(1 - \mu(A \mid m))r - 2\mu(A \mid m)k \geq 0 \tag{4.8}$$

Expanding the left-hand side,

$$r - \mu(A \mid m)r \geq \mu(A \mid m)k \tag{4.9}$$

$$r \geq \mu(A \mid m)(k + r) \tag{4.10}$$

Solving for $\mu(A \mid m)$ gives

$$\mu(A \mid m) \leq \frac{r}{k + r} \tag{4.11}$$

Let the critical threshold be defined as

$$\mu^* = \frac{r}{k + r} \tag{4.12}$$

The optimal policy can therefore be expressed as follows:

- If $\mu(A \mid m) \leq \mu^*$, allowing the request yields the highest expected payoff.
- If $\mu(A \mid m) > \mu^*$, restricting the request becomes the optimal action.

Clarification may become optimal in an intermediate region whenever

$$EU(\text{Clarify}_i \mid m) > \max\{EU(\text{Allow} \mid m), EU(\text{Restrict} \mid m)\},$$

for $i \in \{1, 2\}$, where Δ_1 and Δ_2 represent the effectiveness of the clarification process and c denotes the cost of issuing a clarification query.

4.2.1 Impact of Model Parameters on the Restriction Threshold

The decision threshold derived in the previous section depends on several parameters of the utility model. In particular, the restriction threshold

$$\mu^* = \frac{r}{r + k} \tag{4.13}$$

determines the probability level at which the chatbot switches from allowing a request to restricting it. This threshold depends on the relative magnitude of the attacker damage parameter k and the honest-user utility parameter r. In addition, the effectiveness of clarification steps, represented by parameters Δ_1 and Δ_2, influences the intermediate decision region where clarification becomes optimal.

Effect of Attack Damage and Honest Utility. The parameters k and r determine the trade-off between system safety and user utility.

From the threshold expression in Equation 4.13, it follows that the restriction threshold depends on the relative magnitude of these parameters.

When the potential damage caused by an attacker (k) increases, the denominator ($r + k$) becomes larger, which reduces the value of the threshold p^*. As a consequence, the chatbot becomes more conservative and restricts queries even at lower estimated probabilities of malicious intent.

Conversely, when the benefit of serving an honest user (r) increases, the threshold μ^* becomes larger. In this case, the chatbot becomes more permissive, allowing responses for a wider range of probabilities before switching to restriction. Therefore, the parameters k and r directly control the balance between safety and usability in the chatbot's decision policy.

Figure 4.6 illustrates this relationship by plotting the restriction threshold μ^* as a function of the attack damage parameter k for two different values of the honest user utility r. As shown in the figure, the threshold decreases monotonically as k increases, confirming that higher potential attack damage leads the system to adopt a more restrictive policy. Moreover, for larger values of r, the threshold remains higher across the entire range of k, indicating that the system becomes more tolerant when the benefit of serving legitimate users is greater.

When the benefit of serving an honest user is greater than the potential attack damage ($r > k$), the threshold μ^* becomes larger. As a result, the chatbot tolerates a higher estimated probability of malicious intent before restricting a query. In this case, the system adopts a more permissive policy, prioritizing usability and ensuring that legitimate users can obtain information without unnecessary restrictions.

Therefore, the relative magnitude of k and r determines whether the chatbot operates in a safety-oriented regime ($k > r$) or in a usability-oriented regime ($r > k$), directly influencing the balance between security and accessibility in the decision-making process.

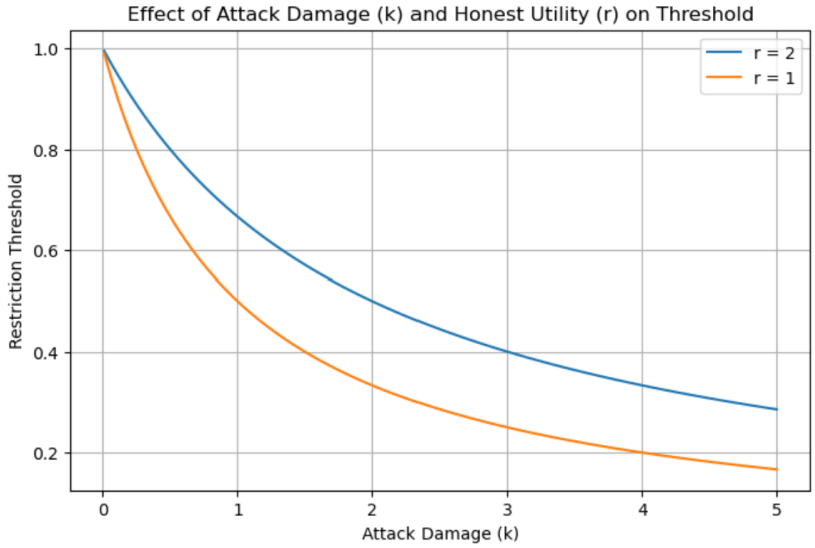

Figure 4.6: Threshold as a function of attack damage k for different values of r.

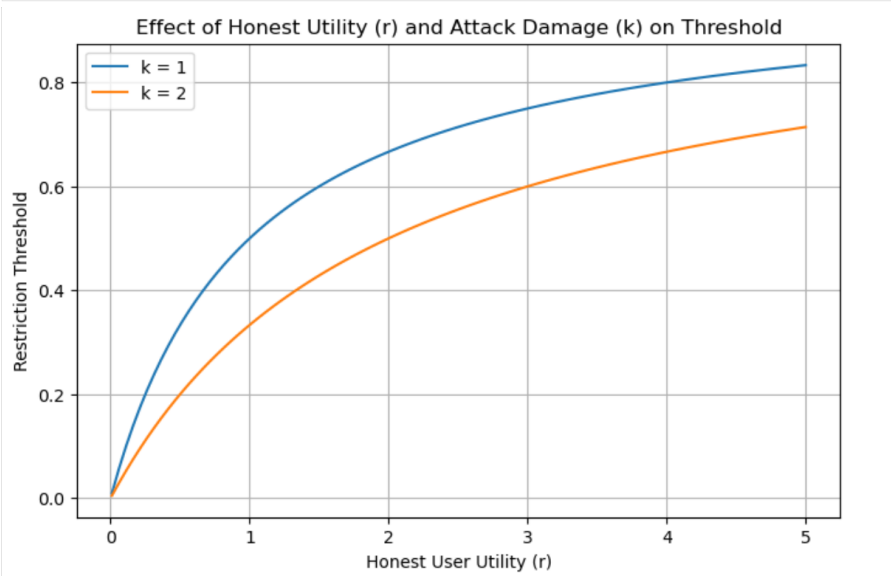

Figure 4.7: Threshold as a function of honest user utility r for different values of k.

Effect of Clarification Effectiveness (Δ_1, Δ_2)

In addition to the allow and restrict actions, the chatbot can request clarification in order to reduce uncertainty about the user's intent. The effectiveness of clarification is represented by the parameters $_1$ and Δ_2, which model the degree to which each clarification step improves the system's information about the user type.

The expected utility of clarification is modeled as

$$EU(\text{Clarify}_1) = (1-p)\Delta_1 k + p\Delta_1 r - c$$

$$EU(\text{Clarify}_2) = (1-p)\Delta_2 k + p\Delta_2 r - c$$

where c represents the cost associated with performing a clarification step.

Higher values of Δ_1 or Δ_2 indicate that clarification is more informative, increasing the expected utility of asking additional questions. As a result, the region in which clarification becomes the optimal action expands.

On the other hand, if the clarification effectiveness is low, the benefit of requesting additional information decreases, and the chatbot will more frequently switch directly to either allowing or restricting the request.

Consequently, the parameters Δ_1 and Δ_2 influence the size of the intermedi-

ate uncertainty region in which clarification is preferred over immediate decisions.

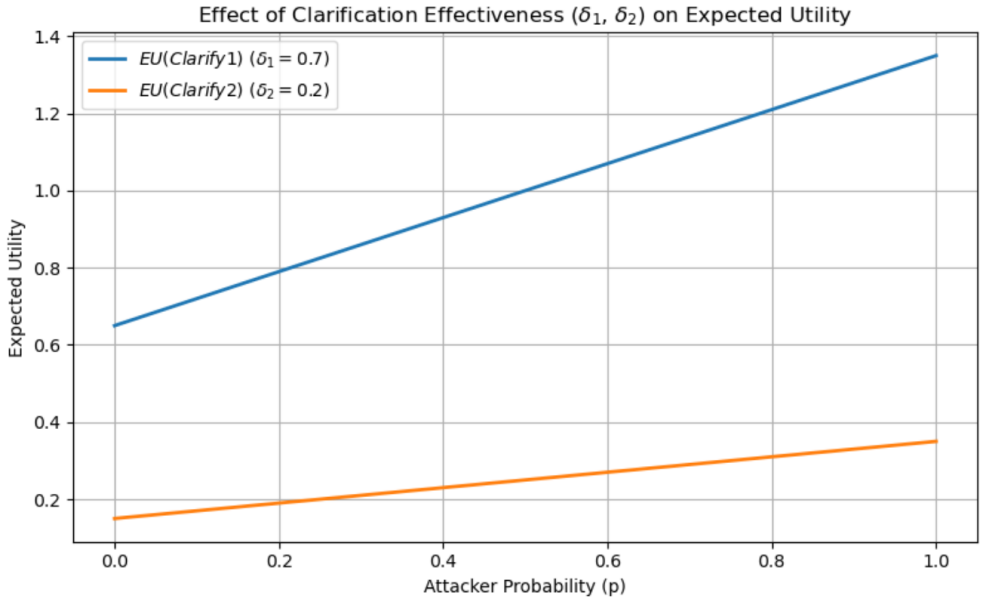

Figure 4.8: Effect of clarification effectiveness (Δ_1, Δ_2) on the expected utility of clarification actions. The parameters $\Delta_1 = 0.7$ and $\Delta_2 = 0.2$ correspond to the values used in the implemented chatbot application.

To illustrate the effect of the clarification effectiveness parameters, Figure 4.8 shows the expected utility of the two clarification actions as a function of the attacker probability p. In this experiment, the clarification parameters are fixed to $\Delta_1 = 0.7$ and $\Delta_2 = 0.2$, which correspond to the same values used in the implemented chatbot application.

As shown in the figure, the expected utility of the first clarification step is higher than that of the second clarification step across the entire range of attacker probabilities. This occurs because Δ_1 represents a more informative clarification stage, while Δ_2 reflects a weaker additional improvement in the system's knowledge about the user's intent. Consequently, the first clarification step provides a greater contribution to reducing uncertainty compared to subsequent clarification attempts.

4.2.2 Combined Risk–Utility Threshold Analysis

Figure 4.9 illustrates how the optimal restriction threshold μ^* varies under different risk and utility conditions. The blue curve represents the effect of hon-

est user utility r, while the orange curve represents the effect of attacker-related risk parameter k. Together, these curves demonstrate how the chatbot adjusts its restriction policy depending on the balance between safety and usability.

As shown in the figure, the restriction threshold decreases as the attacker-related parameter k increases. This behavior indicates that when the potential damage or incentive for malicious users becomes higher, the chatbot adopts a more conservative strategy. In such cases, the system blocks queries at lower estimated risk levels in order to reduce the probability of harmful responses.

In contrast, the restriction threshold increases as the honest user utility parameter r grows. When the benefit of serving legitimate users becomes larger, the chatbot adopts a more permissive strategy and allows a greater number of queries to pass through the system. This behavior reflects the need to preserve usability and avoid unnecessarily restricting helpful interactions.

Overall, these results highlight the trade-off between safety and accessibility in medical chatbot systems. The optimal decision threshold must balance the risk posed by potential attackers against the value of providing useful responses to legitimate users. These insights support the probability-based decision mechanism implemented in the proposed AttackGuard framework, where user queries are classified into honest, clarify, and attacker regions based on their estimated risk level.

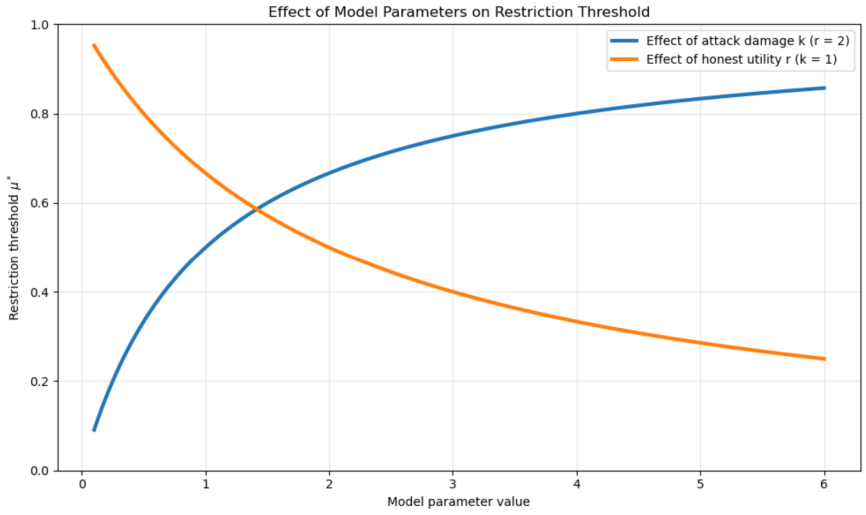

Figure 4.9: Optimal restriction threshold p^* under varying attacker risk parameter k and honest user utility r. The figure illustrates how the chatbot adapts its restriction policy depending on the balance between system safety and usability.

4.3 Implications for Real-World Medical Chatbot Systems

This section discusses the practical implications of the proposed model for real-world medical chatbot systems. It explains how the theoretical signaling-game framework can be integrated into a chatbot architecture to support decision-making under uncertainty. The section also highlights the trade-off between safety and usability, showing how the system adapts its behavior by allowing, clarifying, or restricting responses based on estimated risk. Additionally, it emphasizes the role of game theory in modeling strategic user behavior and in guiding the design of safer and more reliable AI chatbot systems.

4.3.1 Comparison with Real AI Chatbot Behavior

The proposed signaling-game framework reflects how real medical AI chatbots operate in practice. In real systems, a chatbot usually does not know the true intention of the user and must estimate the level of risk associated with each query, as commonly done in aligned language models [35].

To demonstrate how the model can be used in practice, the theoretical framework was integrated into a modified medical chatbot architecture based on an open-source chatbot system. A dataset was added containing estimated probabilities of malicious or unsafe user behavior. These probabilities represent the likelihood that a query comes from an attacker or from a user trying to obtain harmful medical advice, similar to modern safety-oriented systems [23].

When a query is received, the chatbot evaluates it using a trained probability model and assigns a risk score. This score corresponds to the posterior belief $\mu(A|m)$ in the signaling-game model. The chatbot then compares this probability with predefined decision thresholds obtained from the game-theoretic analysis. If the probability of malicious intent is too high, the chatbot restricts the response instead of providing potentially unsafe medical information.

This process follows the same logic as the Bayesian decision-making described in the theoretical model. The chatbot cannot directly observe the true type of the user and must infer it from the available signals and past data. This shows that the theoretical model can be applied to practical AI safety mechanisms in real chatbot systems. A key challenge in medical chatbot design is balancing safety and usability. Systems that are too restrictive may block useful information for

legitimate users. On the other hand, systems that are too permissive may provide unsafe or misleading medical advice. This trade-off has also been emphasized in recent work on contextual safety and over-refusal in language models [49].

The proposed model represents this trade-off using utilities and decision thresholds. Allowing a response is beneficial when the user is honest but can be harmful when interacting with an attacker. Restricting a response prevents potential harm but may reduce user satisfaction.

The clarification mechanism provides an intermediate option, allowing the chatbot to ask additional questions before making a final decision. The analysis shows that the optimal chatbot strategy depends on several factors, such as the probability of encountering an attacker, the possible damage from unsafe responses, the benefit of helping honest users, and the cost of additional verification steps.

As these parameters change, the chatbot may switch between allowing, clarifying, or restricting responses. This adaptive behavior reflects how modern medical AI systems try to provide helpful information while minimizing risk.

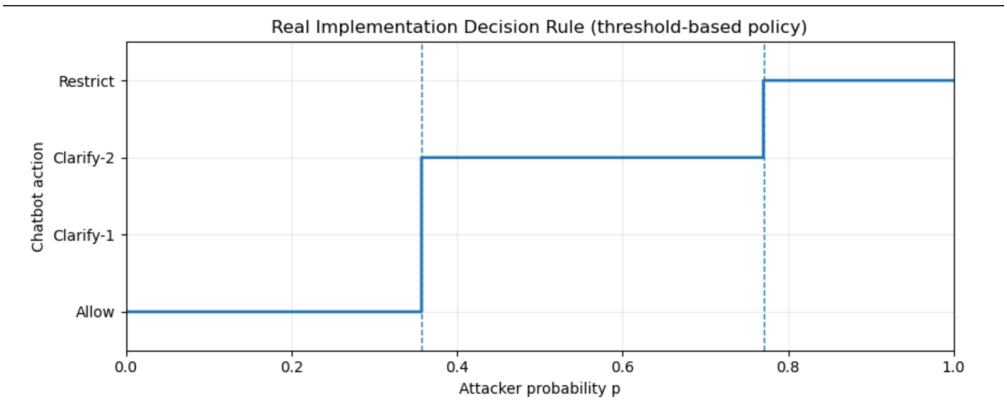

Figure 4.10: Threshold-based decision rule for the implemented medical chatbot. Depending on the estimated attacker probability p, the system chooses between allowing the response, requesting clarification, or restricting the query.

5

Conclusions

The development of AI-based conversational systems in healthcare introduces important challenges related to safety, reliability, and responsible information disclosure. Medical chatbots are expected to provide useful guidance to legitimate users while preventing the disclosure of potentially harmful information. A fundamental difficulty in this interaction arises from uncertainty about user intent: the chatbot cannot directly observe whether a user is acting honestly or attempting to exploit the system.

This thesis addressed this challenge by adopting a game-theoretic perspective on the interaction between users and medical chatbots. In particular, the problem was modeled as a signaling game with asymmetric information, where the user acts as the sender and the chatbot acts as the receiver. The sender possesses private information about their type—honest or malicious—while the chatbot must infer this type based on observable signals, namely the user's queries. By framing the interaction in this way, the chatbot's decision-making process can be analyzed as a strategic response under uncertainty.

The proposed model introduces a strategic decision framework based on expected utility maximization. Within this framework, the chatbot evaluates alternative actions—allowing a response, requesting clarification, or restricting the query—according to their expected outcomes under different probabilities of adversarial behavior. The analysis demonstrates how decision thresholds naturally emerge from the payoff structure, defining regions in which each action becomes the optimal strategy. In particular, clarification requests play a critical role in intermediate uncertainty scenarios, allowing the system to gather addi-

tional information before committing to a final decision.

Through this analysis, the thesis highlights how game-theoretic reasoning provides a principled approach to balancing two competing objectives in medical AI systems: maintaining safety by limiting harmful information disclosure and preserving usefulness by assisting legitimate users. Rather than relying solely on static filtering mechanisms, the signaling game model enables the system to adapt its behavior dynamically based on estimated risk and strategic interaction patterns.

5.1 Contributions and Implications

This thesis makes the following key contributions:

- The formulation of the interaction between a user and a medical chatbot as a signaling game with asymmetric information.
- The development of a utility-based decision framework that models chatbot responses as strategic choices under uncertainty.
- The identification of probability thresholds that determine when the chatbot should allow, clarify, or restrict a request.
- The demonstration that clarification steps can function as an information-gathering mechanism that improves decision quality in uncertain environments.

More broadly, the findings illustrate how concepts from game theory can be applied to the design of safer AI systems. By explicitly modeling the strategic behavior of users and the uncertainty faced by the chatbot, the proposed framework contributes to a deeper understanding of decision-making in adversarial human–AI interactions.

5.2 Future Work

Several directions remain open for future research and could further improve the proposed framework.

- **Modeling more realistic user behavior.** Future work could expand the model to capture a wider variety of user types beyond just "honest" and "attacker," representing different risk levels and real-world interactions with medical chatbots [17].

- **Multi-turn and extended interactions.** The current model handles only a few interaction steps. Future research could support longer, dynamic conversations using multi-stage signaling games, allowing the chatbot to adapt its strategy over extended exchanges [33].
- **Improved detection of risky queries.** Machine learning approaches, such as text classification or anomaly detection, could be employed to identify potentially harmful or suspicious queries, enhancing the chatbot's decision-making process.
- **Improving safety and reliability.** Strategies for controlling information disclosure and avoiding unsafe medical advice remain critical for future work, ensuring that chatbots remain both helpful and secure.

In conclusion, this thesis demonstrates that game-theoretic models offer a powerful framework for analyzing and designing decision-making strategies in medical chatbot systems. By treating the interaction as a signaling game under uncertainty, it becomes possible to formally reason about safety, information disclosure, and user intent. These insights provide a foundation for the development of more robust and trustworthy AI systems in healthcare and other safety-critical domains.

References

[1] A. Abd-Alrazaq, A. Rababeh, M. Alajlani, et al. "Overview of Chatbots in Healthcare: A Scoping Review". In: (2019). URL: https://www.sciencedirect.com/science/article/abs/pii/S1386505619307166.

[2] F. Abtahi, F. Seoane, I. Pau, et al. "Data Poisoning Vulnerabilities Across Healthcare AI Architectures: A Security Threat Analysis". In: (2025). URL: https://arxiv.org/abs/2511.11020.

[3] Anthropic. *HH-RLHF Dataset*. 2023. URL: https://huggingface.co/datasets/Anthropic/hh-rlhf.

[4] Leonardo Badia and Thomas Marchioro. *Game Theory: A Handbook of Problems and Exercises*. Università degli Studi di Padova, 2022. URL: https://www.research.unipd.it/handle/11577/3470935.

[5] Leonardo Badia and Thomas Marchioro. "On the anarchy of multiple false data injectors for age of incorrect information in sensor networks". In: *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*. 2025, pp. 1–6.

[6] Leonardo Badia et al. "Age of Information for Machine Learning Tasks With Mobile Edge Computing Offloading". In: *Proc. IEEE Int. Symp. Personal Indoor Mobile Radio Commun. (PIMRC)*. 2025, pp. 1–6.

[7] Leonardo Badia et al. "Ambiguous Data Injection Impacting Age of Incorrect Information: A Bayesian Game Analysis". In: *Proc. IEEE Int. Symp. Personal Indoor Mobile Radio Commun. (PIMRC)*. 2025, pp. 1–6.

[8] Leonardo Badia et al. "Medical self-reporting with adversarial data injection modeled via game theory". In: *Proc. Int. Conf. Commun. Signal Proc. Appl. (ICCSPA)*. 2024, pp. 1–6.

[9] Yuntao Bai et al. "Constitutional AI: Harmlessness from AI feedback". In: *arXiv preprint arXiv:2212.08073* (2022). URL: https://arxiv.org/pdf/2212.08073.

[10] Emily M. Bender et al. "On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?" In: *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency* (2021), pp. 610–623. DOI: 10.1145/3442188.3445922.

[11] Jean-Emmanuel Bibault et al. "Healthcare ex Machina: Are conversational agents ready for prime time in oncology?" In: *Clinical and Translational Radiation Oncology* 16 (2019), pp. 55–59. URL: https://doi.org/10.1016/j.ctro.2019.04.001.

[12] C. Blease, M. H. Bernstein, J. Gaab, et al. "Artificial Intelligence and the Future of Primary Care: Safety and Ethical Considerations". In: *BMJ Health & Care Informatics* 26.1 (2019), e100017. URL: https://www.jmir.org/2019/3/e12802/?utm_source=chatgpt.com.

[13] R. Bommasani et al. "On the Opportunities and Risks of Foundation Models". In: *Proceedings of the National Academy of Sciences* (2023). URL: https://arxiv.org/abs/2108.07258?utm_source=chatgpt.com.

[14] Valeria Bonagura et al. "Strategic interaction over age of incorrect information for false data injection in cyber-physical systems". In: *IEEE Trans. Control Netw. Syst.* 12.1 (2025), pp. 872–881.

[15] Tom Brown et al. "Language Models are Few-Shot Learners". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 1877–1901. URL: https://arxiv.org/abs/2005.14165.

[16] Alessandro Buratto et al. "Energy minimization for participatory federated learning in IoT analyzed via game theory". In: *Proc. Int. Conf. Artif. Intell. Inform. Commun. (ICAIIC)*. 2024, pp. 249–254.

[17] L. Carmichael, S. M. Poirier, C. K. Coursaris, et al. "Users' Information Disclosure Behaviors during Interactions with Chatbots: The Effect of Information Disclosure Nudges". In: *Applied Sciences* 12.24 (2022), p. 12660. DOI: 10.3390/app122412660.

[18] J. Clark et al. "Game-Theoretic Approaches for Secure AI". In: *IEEE Access* (2022). URL: https://ieeexplore.ieee.org/.

[19] J. Clusmann et al. "Prompt Injection Attacks on Large Language Models in Oncology". In: *arXiv preprint arXiv:2407.18981* (2024). URL: https://arxiv.org/abs/2407.18981.

[20] Drew Fudenberg and Jean Tirole. *ijory*. MIT Press, 1991. URL: http://www.library.fa.ru/files/tirole-game.pdf.

[21] Anna V. Guglielmi and Leonardo Badia. "Analysis of Strategic Security Through Game Theory for Mobile Social Networks". In: *Proceedings of the IEEE International Workshop on Computer-Aided Modeling Analysis and Design of Communication Links and Networks (CAMAD)*. Lund, Sweden: IEEE, June 2017. URL: https://www.dei.unipd.it/~badia/papers/2017_06_CAMAD.pdf.

[22] Andreas Holzinger et al. "What Do We Need to Build Explainable AI Systems for the Medical Domain?" In: *arXiv preprint* (2017). URL: https://arxiv.org/abs/1712.09923.

[23] Hakan Inan et al. "Llama Guard: LLM-based input-output safeguard for human-AI conversations". In: *arXiv preprint arXiv:2312.06674* (2023). URL: https://arxiv.org/abs/2312.06674.

[24] D. Ippolito et al. "Survey of Vulnerabilities in Large Language Models Revealed by Adversarial Attacks". In: *arXiv preprint arXiv:2310.10844* (2023). URL: https://arxiv.org/pdf/2310.10844.

[25] Niveen O. Jaffal, Mohammed Alkhanafseh, and David Mohaisen. "Large Language Models in Cybersecurity: Applications, Vulnerabilities, and Defense Techniques". In: *MDPI* (2025). URL: https://www.mdpi.com/2673-2688/6/9/216.

[26] Michael Kearns and Aaron Roth. *The Ethical Algorithm: The Science of Socially Aware Algorithm Design*. Oxford University Press, 2019. URL: https://global.oup.com/academic/product/the-ethical-algorithm-9780190948207.

[27] L. Laranjo, A. G. Dunn, H. L. Tong, et al. "Conversational Agents in Healthcare: A Systematic Review". In: *Journal of the American Medical Informatics Association* 25.9 (2018), pp. 1248–1258. URL: https://academic.oup.com/jamia/article/25/9/1248/5053593.

[28] Lavita. *ChatDoctor-HealthCareMagic-100k Dataset*. 2023. URL: https://huggingface.co/datasets/lavita/ChatDoctor-HealthCareMagic-100k.

[29] R. W. Lee, J. Suh, et al. "Vulnerability of Medical LLMs to Prompt Injection When Providing Medical Advice". In: *JAMA Network Open* (2025). URL: https://jamanetwork.com/journals/jamanetworkopen/fullarticle/2842987.

[30] Weitao Li et al. "Citation-Enhanced Generation for LLM-based Chatbots". In: *arXiv preprint arXiv:2402.16063* (2024). URL: https://arxiv.org/abs/2402.16063.

[31] MDN Contributors. *Cross-Origin Resource Sharing (CORS) configuration Practical implementation guides*. Accessed: 2026-04-04. Mozilla Developer Network. 2025. URL: https://developer.mozilla.org/en-US/docs/Web/Security/Practical_implementation_guides/CORS.

[32] MDN Contributors. *POST request method HTTP Reference*. Accessed: 2026-04-04. Mozilla Developer Network. 2025. URL: https://developer.mozilla.org/en-US/docs/Web/HTTP/Reference/Methods/POST.

[33] Roger B. Myerson. *Game Theory: Analysis of Conflict*. Harvard University Press, 1991. URL: https://www.hup.harvard.edu/catalog.php?isbn=9780674341166.

[34] OpenAI. *GPT-4 Technical Report*. 2023. URL: https://openai.com/research/gpt-4.

[35] Long Ouyang et al. "Training language models to follow instructions with human feedback". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 27730–27744. URL: https://arxiv.org/abs/2203.02155.

[36] James Pawlick and Quanyan Zhu. "A Stackelberg Signaling Game for Cyber Deception". In: *IEEE Transactions on Information Forensics and Security* (2017). URL: https://doi.org/10.1109/TIFS.2017.2718483.

[37] James Pawlick and Quanyan Zhu. "Quantitative Models of Imperfect Deception in Network Security Using Signaling Games with Evidence". In: *arXiv preprint* (2017). URL: https://arxiv.org/abs/1703.05484.

[38] Benedetta Picano, Alessandro Buratto, and Leonardo Badia. "Joint Communication and Inference User Allocation in LLM Native Networks". In: *Proc. IEEE Int. Conf. Machine Learning Commun. Netw. (ICMLCN)*. 2025, pp. 1–6.

[39] Pinecone. *Pinecone Vector Database*. 2023. URL: https://www.pinecone.io/.

[40] Pydantic Contributors. *Schema — Pydantic v1.10 Documentation*. Accessed: 2026-04-04. Pydantic Documentation. 2024. URL: https://docs.pydantic.dev/1.10/usage/schema/.

[41] skydev9293. *Clinical-ChatBot*. https://github.com/skydev9293/Clinical-ChatBot. Accessed: 2026-04-03. 2025.

[42] W3Schools.com. *React Introduction W3Schools*. Accessed: 2026-04-04. W3Schools. 2026. URL: https://www.w3schools.com/react/react_intro.asp.

[43] Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. "Jailbroken: How does LLM safety training fail?" In: *Advances in Neural Information Processing Systems* 36 (2023), pp. 80079–80110.

[44] Laura Weidinger et al. "Ethical and Social Risks of Harm from Language Models". In: *arXiv preprint* (2021). URL: https://arxiv.org/abs/2112.04359.

[45] Lu Xu et al. "Chatbot for Health Care and Oncology Applications Using Artificial Intelligence and Machine Learning: Systematic Review". In: *JMIR Cancer* 7.4 (2021), e27850. DOI: 10.2196/27850. URL: https://pmc.ncbi.nlm.nih.gov/articles/PMC8669585/.

[46] W. Xu and K. K. Parhi. "A Survey of Attacks on Large Language Models". In: *arXiv preprint arXiv:2505.12567* (2025). URL: https://arxiv.org/abs/2505.12567.

[47] Y. Xue et al. "Evaluation of the Current State of Chatbots for Digital Health: Scoping Review". In: *Journal of Medical Internet Research* 25 (2023), e47217. URL: https://www.jmir.org/2023/1/e47217/.

[48] Z. Zhang, L. Huang, et al. "Health-ORSC-Bench: A Benchmark for Over-Refusal and Safe Completion in Healthcare". In: *arXiv preprint* (2026). URL: https://arxiv.org/abs/2601.17642.

[49] Zhiwei Zhang et al. "FalseReject: A resource for improving contextual safety and mitigating over-refusals in LLMs via structured reasoning". In: (2025). URL: https://arxiv.org/abs/2505.08054.

[50] Zustand Documentation. *Introduction – Zustand*. https://zustand.docs.pmnd.rs/learn/getting-started/introduction. Accessed: 2026-04-04. 2025.

Acknowledgments

After a journey full of challenges and efforts, I am pleased to express my sincere gratitude to everyone who made this achievement possible.

For my family, I would like to express my deepest gratitude to my parents, who believed in me and supported me in every step of this academic journey. Every piece of advice and every word of encouragement has motivated me to achieve even more. I would also like to thank my sister and my brother, who mean everything to me. I wish them endless success in their own journeys.

A special thanks goes to my uncle and his family, without whom this journey would have been much more difficult. They have been a great support to me throughout these two years. I would also like to thank my cousin, my translator, who helped me a lot during these two years with my not-so-perfect Italian.

I would like to thank all my friends, and especially Tea, with whom I started this journey, making our stay in a new and unfamiliar city much easier for each other.

I am also grateful to all the people I met during these two years. I never imagined I would meet such wonderful people who made this journey less lonely and much more meaningful, always being there for a new and exciting adventure.

Finally, I would like to thank my supervisor, Leonardo Badia, who guided me with dedication and professionalism throughout these months, helping me achieve meaningful results. I am truly grateful for every meeting and every piece of support provided—your contribution has been invaluable to me.

My gratitude goes to all of you who were part of my journey; each of you was an extra strength for me.