

Original Paper

Large Language Model–Based Critical Care Big Data Deployment and Extraction: Descriptive Analysis

Zhongbao Yang^{1*}, BS; Shan-Shan Xu^{1*}, MD; Xiaozhu Liu^{1*}, PhD; Ningyuan Xu², BS; Yuqing Chen³, MD; Shuya Wang³, MD; Ming-Yue Miao¹, MD; Mengxue Hou³, MD; Shuai Liu³, MD; Yi-Min Zhou³, MD; Jian-Xin Zhou¹, MD, PhD; Linlin Zhang³, MD, PhD

¹Department of Critical Care Medicine, Beijing Shijitan Hospital, Capital Medical University, Beijing, China

²School of Information Science and Technology, Beijing University of Technology, Beijing, China

³Department of Critical Care Medicine, Beijing Tiantan Hospital, Capital Medical University, Beijing, China

*these authors contributed equally

Corresponding Author:

Linlin Zhang, MD, PhD
Department of Critical Care Medicine
Beijing Tiantan Hospital, Capital Medical University
No.119 Nansihuanxi Road, Fengtai District
Beijing, 100070
China
Phone: 86 17611757717
Email: abluelemon@163.com

Abstract

Background: Publicly accessible critical care–related databases contain enormous clinical data, but their utilization often requires advanced programming skills. The growing complexity of large databases and unstructured data presents challenges for clinicians who need programming or data analysis expertise to utilize these systems directly.

Objective: This study aims to simplify critical care–related database deployment and extraction via large language models.

Methods: The development of this platform was a 2-step process. First, we enabled automated database deployment using Docker container technology, with incorporated web-based analytics interfaces Metabase and Superset. Second, we developed the intensive care unit–generative pretrained transformer (ICU-GPT), a large language model fine-tuned on intensive care unit (ICU) data that integrated LangChain and Microsoft AutoGen.

Results: The automated deployment platform was designed with user-friendliness in mind, enabling clinicians to deploy 1 or multiple databases in local, cloud, or remote environments without the need for manual setup. After successfully overcoming GPT’s token limit and supporting multischema data, ICU-GPT could generate Structured Query Language (SQL) queries and extract insights from ICU datasets based on request input. A front-end user interface was developed for clinicians to achieve code-free SQL generation on the web-based client.

Conclusions: By harnessing the power of our automated deployment platform and ICU-GPT model, clinicians are empowered to easily visualize, extract, and arrange critical care–related databases more efficiently and flexibly than manual methods. Our research could decrease the time and effort spent on complex bioinformatics methods and advance clinical research.

JMIR Med Inform 2025;13:e63216; doi: [10.2196/63216](https://doi.org/10.2196/63216)

Keywords: big data; critical care–related databases; database deployment; large language model; database extraction; intensive care unit; ICU; GPT; artificial intelligence; AI; LLM

Introduction

Critical care medicine has experienced an explosion of complex, heterogeneous data from various sources, such as electronic medical records, monitoring systems, imaging, registries, and omics platforms. This massive amount of

multimodal data, often referred to as “big data,” offers enormous potential for improving patient outcomes in intensive care units (ICUs) through studies and analytics.

There are many high-quality medical databases that excel in the dimensions of volume, velocity, veracity, variety, and value. Examples include the Medical Information Mart for

Intensive Care (MIMIC; MIMIC-III, MIMIC-IV, MIMIC-IV-Note, etc) at the Beth Israel Deaconess Medical Center in Boston [1-3], the Electronic Intensive Care Unit Collaborative Research Database (eICU-CRD) from 208 hospitals in the United States, and the Amsterdam University Medical Center Database (Amsterdam UMCdb) [4-6]. By integrating demographic, monitoring, laboratory, imaging, pharmacy, and waveform data, these databases provide a wealth of valuable information that has facilitated clinical research [7] and expanded the evidence base for clinical practice [8]. Many studies have demonstrated the potential of leveraging ICU big data for applications such as risk stratification, predictive models, treatment recommendations, and cohort phenotyping [9-12].

However, there are significant barriers to translating big data resources into clinical insights and decision support for frontline critical care teams. The complexity of ICU data requires specialized skills in data wrangling, mining, visualization, and interpretation, which most clinicians need to acquire. The gap between data science and clinical science hinders harnessing big data to improve ICU care [13,14]. Despite the promise of rich data and analytics, the adoption of actionable insights depends on accessible interfaces and informative data presentations [15].

Recent advances in medical natural language processing using artificial intelligence (AI) and large language models (LLMs) like GPTs have demonstrated the potential for intuitive human-computer interaction with health care data. Clinicians do not have to learn complex programming languages, and interfaces based on natural language and clinical logic can make big data analysis more accessible. An increasing body of research demonstrates that AI applications significantly enhance the efficiency and accuracy of disease diagnosis and treatment. Farič et al [16] developed an AI-based imaging software tool for detecting pulmonary nodules in chest computed tomography scans. Through qualitative semistructured interviews, they demonstrated the tool's usability and effectiveness as both a decision-support system and a "second reader" for clinical users. Sjoding et al [17] employed a deep convolutional neural network to identify acute respiratory distress syndrome findings on chest radiographs. The convolutional neural network demonstrated expert physician-level accuracy in acute respiratory distress syndrome radiograph detection, as validated through comprehensive quantitative assessments using both internal and external test sets. However, general domain language models still face great challenges in handling niche biomedical data and require customization and continuous iterative improvement.

Therefore, we seek to empower clinicians by simplifying database deployment and developing the intensive care unit-generative pretrained transformer (ICU-GPT). This tool aims to simplify data extraction via LLMs, liberating clinicians from complex bioinformatics methods to focus on clinical research and data analysis. With ICU-GPT, clinicians can focus on clinical research advancement and feel more capable and in control of their data analysis.

Methods

Overview

To achieve the availability and visualization of critical care-related databases and simplify the process of data extraction, we developed this platform in 2 steps. Step 1 was database deployment based on Docker container technology through the web-based client rather than by installing other applications. Step 2 was the development of ICU-GPT based on LLMs to generate Structured Query Language (SQL) queries for data extraction after request input.

Technologies of the Database Deployment and Visualization

Docker is an open-source platform that, by utilizing containerization technology, enables users to package the applications together with all their dependencies into an isolated container to achieve portability, consistency, and high segregation [18]. Database deployment involves the installation and configuration of health care databases onto PostgreSQL. This process ensures proper data storage and operational functionality. Based on Docker technology and customized PostgreSQL database initialization scripts [19], we aimed for a cross-platform automated deployment of critical care-related databases on different operating systems.

Database visualization leverages business intelligence tools to graphically represent the structure, content, and relationships within a database. This approach not only facilitates simple plotting but also provides a comprehensive and intuitive understanding of the database's table structure and overall architecture. We provide 2 web-based clients with business intelligence tools to achieve database visualization: Metabase and Superset. Metabase and Superset are open-source data analytics and visualization tools for easy data exploration [20,21]. With simple and convenient working interfaces, they allow users to extract data from various sources without writing SQL statements, including relational databases (MySQL, PostgreSQL, SQL Server, etc) and NoSQL databases. They also enable users to add plugins and visualization components to meet specific requests with high extensibility and customizability.

Technologies of ICU-GPT

With the appearance of the OpenAI GPT-3.5 in 2023, we decided to use LLMs to participate in the SQL query generation process [22]. An LLM is a natural language processing tool driven by AI technology and has recently emerged as a powerful tool across many areas of biomedicine. It can generate answers based on patterns observed during the pretraining stage, summarize large amounts of text, generate high-quality text from a short description, create code that can help support data analysis, produce images based on a verbal description, and much more.

ICU-GPT does not develop new LLM models but utilizes and is compatible with all OpenAI application programming interface (API) models. We chose OpenAI API-compatible models for ICU-GPT development for several reasons. First,

our research team is more familiar with OpenAI. There is also a large number of models compatible with the OpenAI API, including GPT-3.5, GPT-4o, and GPT-4o mini, offering wide applications and cost-effectiveness. Users can select the appropriate OpenAI API model based on their specific requirements. Moreover, OpenAI encompasses knowledge bases from the MIMIC databases, which other models lack, including *International Classification of Diseases (ICD)* codes and itemid data.

Using the BIRD and Spider, 2 extensive cross-domain datasets designed to evaluate the impact of comprehensive database content on text-to-SQL parsing, we conducted a comparative analysis of OpenAI API models with other models [23,24]. The OpenAI API models demonstrated consistently high performance in execution accuracy and reward-based valid efficiency within the BIRD dataset [25]. Furthermore, these models exhibited stable and superior performance in execution with values within the Spider dataset [26]. Although reports suggest that certain models may surpass OpenAI's performance, we were unable to include them in our comparison due to limitations in accessibility [27-30]. Moreover, due to the absence of a benchmark dataset specifically tailored for critical care-related databases, it remains challenging to directly compare ICU-GPT's performance with other models on such databases.

Ollama is a lightweight framework with high extensibility for building and running LLMs on local and cloud environments [31]. It provides an API for model creation, execution, and management as well as a prebuilt model library. It is compatible with the OpenAI API for dialogue creation. The development of ICU-GPT also involved 4 technologies: LangChain, Microsoft AutoGen, Pandas, and Gradio.

LangChain is an open-source Python library that provides the modules and tools to build AI applications based on LLMs [32]. It can easily integrate with LLMs to achieve text generation, question-answering, and translation. The LangChain SQL database object was adopted and adjusted in our platform to support databases with multiple schemas, such as MIMIC and eICU-CRD.

Microsoft AutoGen is a framework that can be customized and conversed with, facilitating the development of LLM applications and seamlessly allowing for human participation [33]. Userproxy is a specialized agent within Microsoft AutoGen that receives users' requests and generates appropriate prompts. GroupChat comprises 2 agents and Userproxy, working in tandem to facilitate the contemplation, execution, and refinement of users' requests.

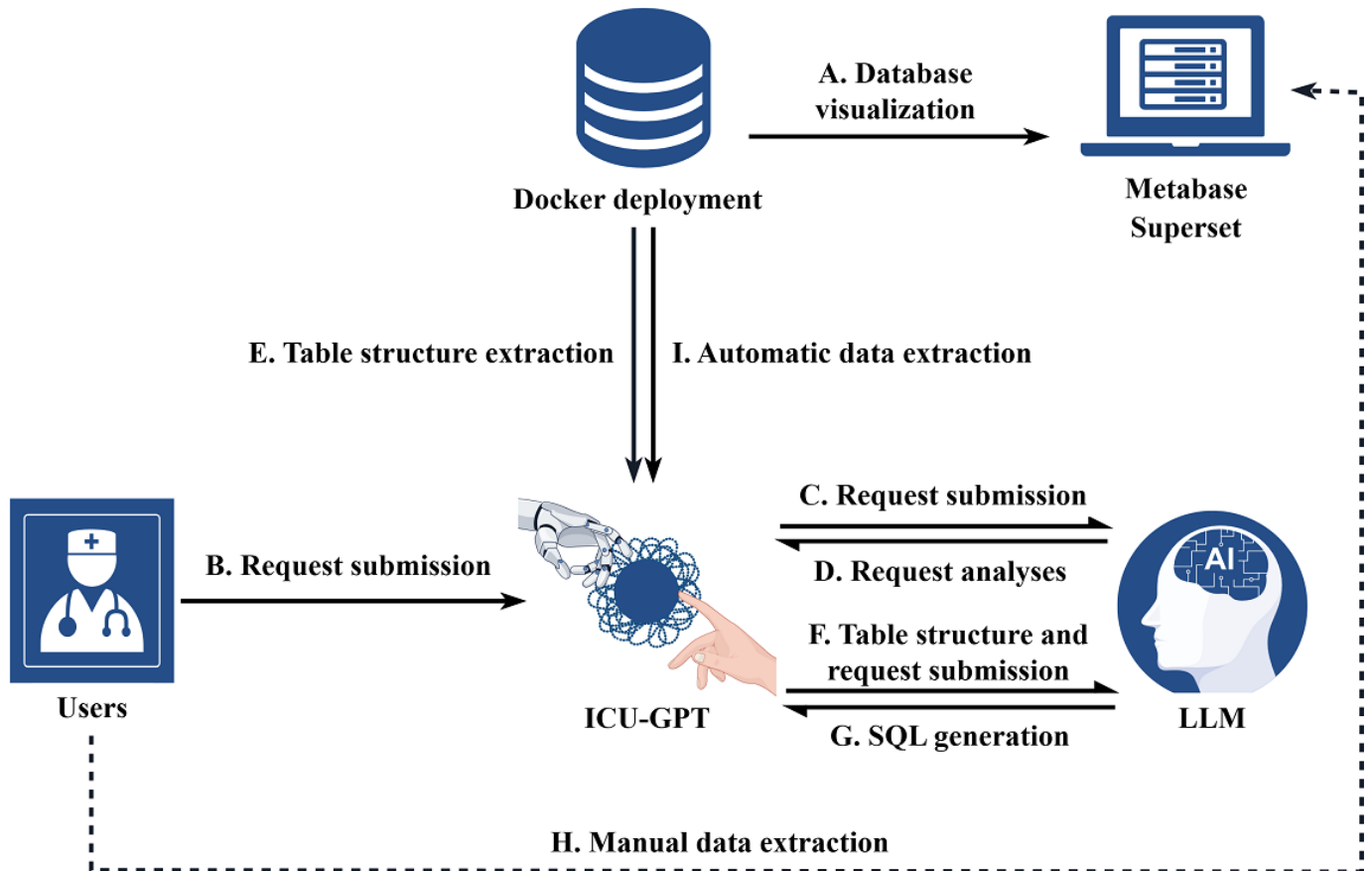
Pandas is a powerful Python-based library. It could provide accessible data structures to make data cleaning, transformation, and analysis more convenient [34]. Current data extraction functions are developed based on Pandas to achieve data processing.

Gradio is an open-source Python library. It enables developers to transform machine learning models and various functions into shareable web applications, facilitating a quick and easy showcasing of their work through intuitive and accessible interfaces [35].

Results

We summarized the workflow of database deployment, visualization, SQL generation, and data extraction in [Figure 1](#).

Figure 1. The procedures for the platform construction. The platform construction workflow encompasses several key components: database deployment based on Docker container technology; visualization powered by Metabase and Superset (A); SQL generation involving multiple rounds of interaction between users, LLMs, and ICU-GPT, including request submission, analysis, table structure extraction, and SQL generation (B to G); and data extraction (H and I). (A) Critical care–related database visualization based on Metabase and Superset. (B) Users send data extraction requests to ICU-GPT. (C) ICU-GPT sends requests to the LLM. (D) The LLM splits and analyzes the requests and sends them back to ICU-GPT. (E) ICU-GPT extracts table structure information from databases. (F) ICU-GPT sends requests and table structure information to the LLM. (G) The LLM generates SQL queries and sends them to ICU-GPT. (H) Users can choose to manually verify the SQL queries in Metabase and Superset. (I) ICU-GPT runs the SQL automatically for data extraction. ICU-GPT: intensive care unit–generative pretrained transformer; LLM: large language model; SQL: Structured Query Language. Figure created using Figdraw.



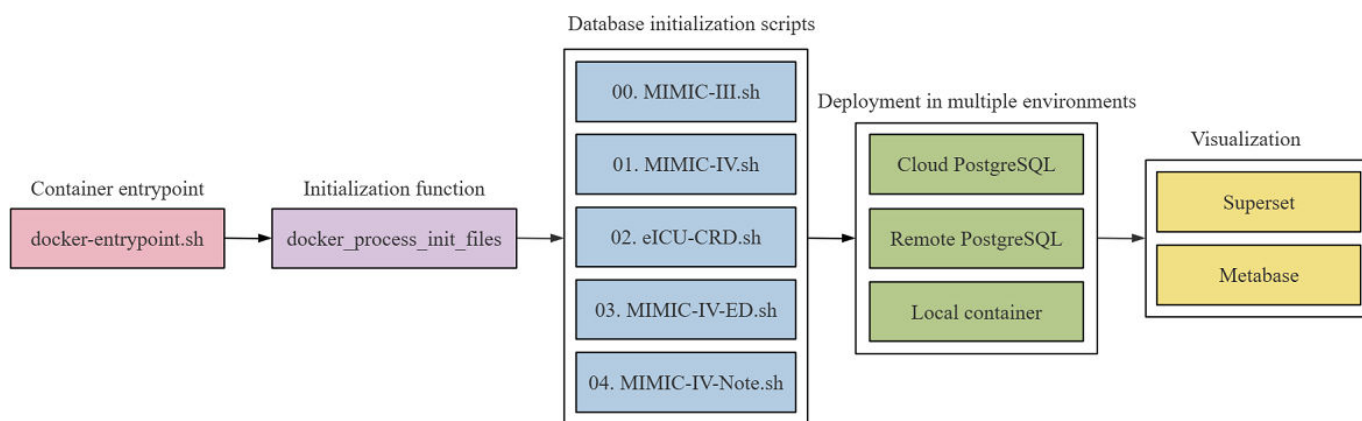
Database Deployment and Visualization

We adopted Docker technology to achieve the simplicity and flexibility of the cross-platform automated deployment of critical care–related databases. Functions of `docker_process_init_files` in `docker-entrypoint.sh` possess the mechanism of scanning shell files automatically in specific directories during the initial PostgreSQL container startup. The corresponding database initialization scripts are placed in the scanning path. Based on the mechanism, we named the files according to specific numbers to ensure the initialization process ran in the intended order and initialized the databases accordingly.

Our deployment platform supports several public critical care–related databases through PostgreSQL database

initialization scripts, including MIMIC-III, MIMIC-IV, MIMIC-IV-ED (Medical Information Mart for Intensive Care IV – Emergency Department), MIMIC-IV-Note, eICU-CRD, etc. Clinicians can select the corresponding database initialization scripts or combine multiple scripts according to their specific requests. This design also allows clinicians to initialize the database with multiple phases in local, cloud, or remote environments and support database upgrades with great flexibility. Through 2 business intelligence solutions, Metabase and Superset, data querying, extraction, and charting can be easily performed by the web-based client without the installation of any other applications (Figure 2).

Figure 2. Deployment and visualization of critical care-related databases. eICU-CRD: Electronic Intensive Care Unit Collaborative Research Database; MIMIC: Medical Information Mart for Intensive Care; MIMIC-IV-ED: Medical Information Mart for Intensive Care IV – Emergency Department.



SQL Query Generation and Data Extraction

We developed ICU-GPT to accurately assist in SQL query writing and provide a more intelligent and efficient approach to simplify data extraction.

Extensive research was conducted on SQL query generation. We found that most SQL query generation tools were only practical for tables under a single schema or merely supported a single table. The handling of databases with multiple schemas, such as MIMIC, was found to be poor. By thoroughly studying the source code of table information retrieval based on the LangChain SQL database object, we successfully overcame this deficiency and achieved multi-schema support.

Given the massive number of MIMIC tables, the actual tokens used to generate prompts exceed the upper limit of 16,000 tokens in OpenAI GPT-3.5. To address this issue, the table selection function was introduced to allow clinicians to generate SQL queries within a specific table range. This innovation not only solved the problem of the token limit but further improved the accuracy of SQL query generation. On the basis of this innovation, our platform is now highly flexible and available for more complex database structures.

Two agents and a Userproxy were established as a GroupChat using Microsoft AutoGen. One agent serves as the SQL engineer, responsible for SQL generation, while the other agent acts as the SQL expert, verifying and optimizing the SQL generated by the SQL engineer. A manager was also created to supervise the GroupChat and facilitate information interaction and agent selection within the group.

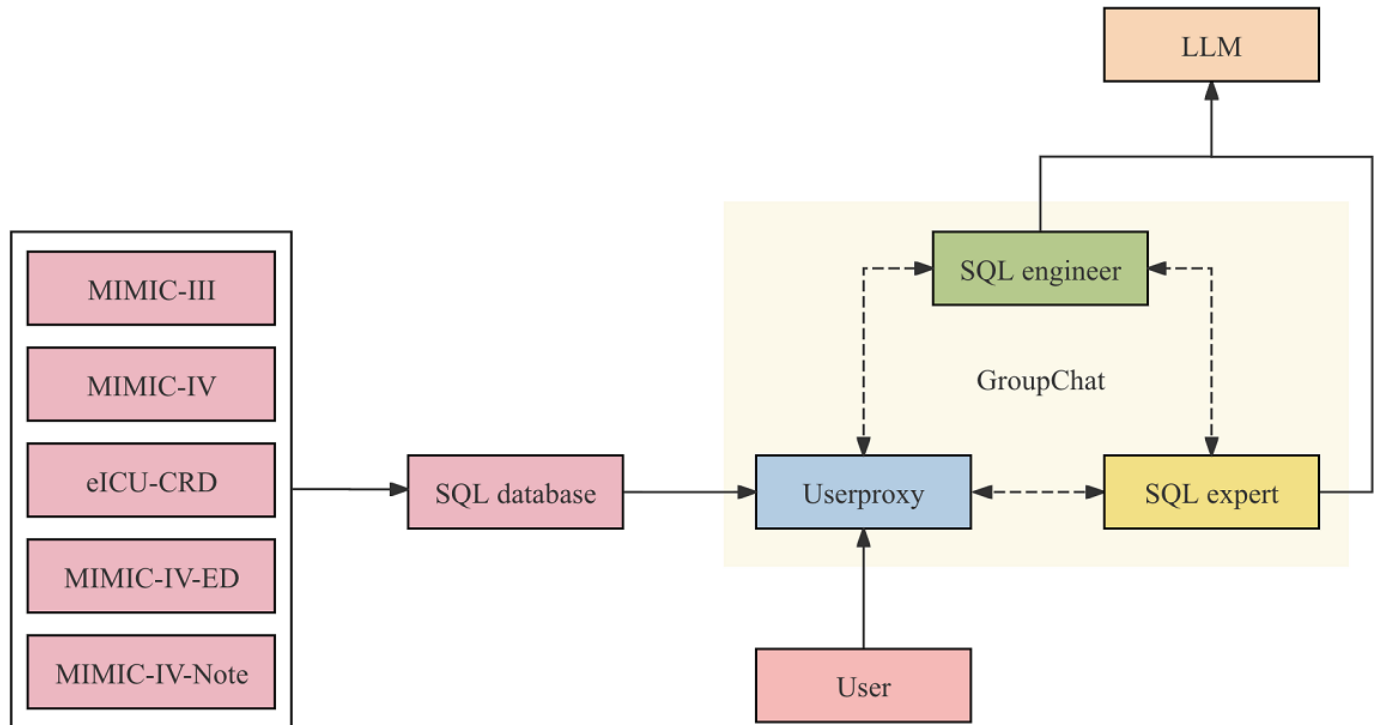
The Userproxy generates prompts based on clinicians' requests and tables specified by users, then sends them to the manager. The manager transmits these prompts to agents

and selects an appropriate agent to process the prompt using a relevant algorithm. This selected agent then sends the prompts to the LLM. The LLM generates SQL based on prompts and requests and then sends it back to the manager. The manager repeats this process, sending messages to agents and selecting the appropriate agent using the algorithm. The LLM receives prompts from the selected agent and interacts with the requests. The SQL expert evaluates the performance and accuracy of the generated SQL. If the SQL expert determines that the SQL accuracy is suboptimal, an iterative dialogue process is initiated to refine and improve the SQL as much as possible.

As AI cannot fully replace human intelligence, there are limitations when users input prompts based on their research objectives. This is particularly evident when the entered keywords are not comprehensive, synonyms or similar prompts are used, or even when typos occur. In such cases, ICU-GPT may struggle to fully comprehend the user's intent, potentially leading to incorrect or incomplete responses. To address this, SQL is not automatically executed after generation. Instead, users are given the opportunity to review and inspect the generated SQL themselves. If the results are unsatisfactory, users can further refine and optimize their prompts. This approach fosters a dynamic human-AI collaboration, maximizing SQL accuracy and user satisfaction (Figure 3) [36,37].

The code of the platform development is available online [38]. We warmly welcome and greatly appreciate any user feedback or new ideas, which can be submitted to us via GitHub. We will update ICU-GPT versions when the MIMIC and eICU-CRD databases are updated or when improved models become available. Some examples of how to achieve database deployment and SQL generation based on Python are presented in [Multimedia Appendix 1](#).

Figure 3. SQL query generation process. The GroupChat consists of the Userproxy, the SQL engineer, and the SQL expert. The manager is created to facilitate communication and information synchronization within the group. The SQL engineer is responsible for SQL generation and interaction with the LLM while the SQL expert is responsible for SQL optimization. The Userproxy generates prompts based on clinicians' requests and user-specified tables. The manager then selects appropriate agents to process these prompts using the LLM. The LLM generates SQL queries, which are subsequently evaluated by the SQL expert. If necessary, an iterative dialogue ensues to refine the SQL for optimal accuracy and effectiveness. eICU-CRD: Electronic Intensive Care Unit Collaborative Research Database; LLM: large language model; MIMIC: Medical Information Mart for Intensive Care; MIMIC-IV-ED: Medical Information Mart for Intensive Care IV – Emergency Department; SQL: Structured Query Language.



User Interface of ICU-GPT

We developed a front-end user interface using Gradio and ICU-GPT to simplify the SQL generation process. Users can download our prebuilt Docker mirroring in the Dockerfile and follow the instructions to achieve code-free SQL generation on the web-based client [39]. First, users select a database: MIMIC-III, MIMIC-IV, MIMIC-CareVue, or eICU-CRD

(Figure 4). The platform displays all table names under all schemas of the selected database. Users select the corresponding tables based on their requests (Figure 5). Users then input requests in the “Prompt” section and press “Submit” (Figure 6). Our platform supports prompts both in Chinese and English. Finally, the SQL bot outputs the SQL queries (Figure 7).

Figure 4. Users select databases on the ICU-GPT platform. ICU-GPT: intensive care unit–generative pretrained transformer.

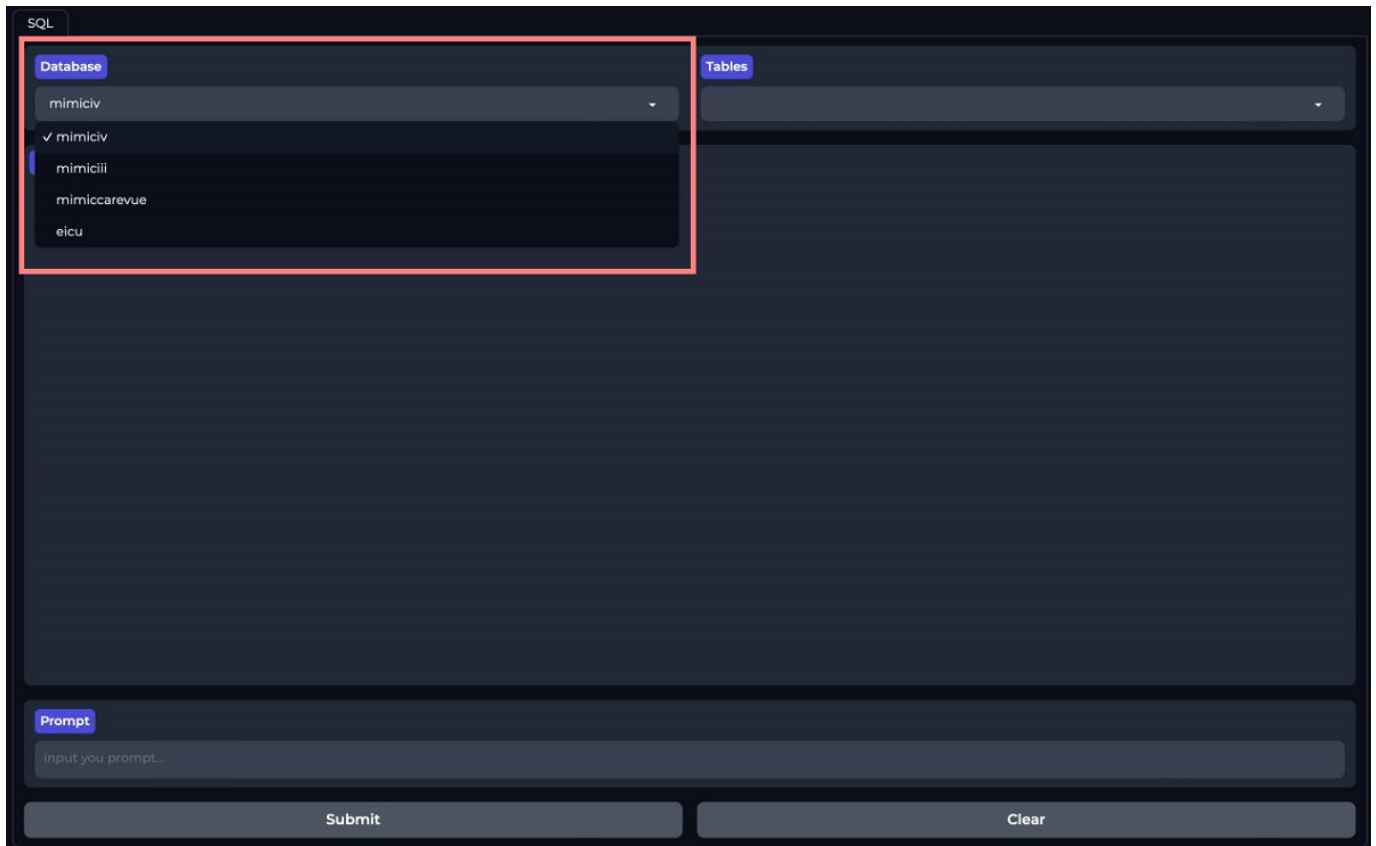


Figure 5. Users select tables on the ICU-GPT platform. ICU-GPT: intensive care unit–generative pretrained transformer.

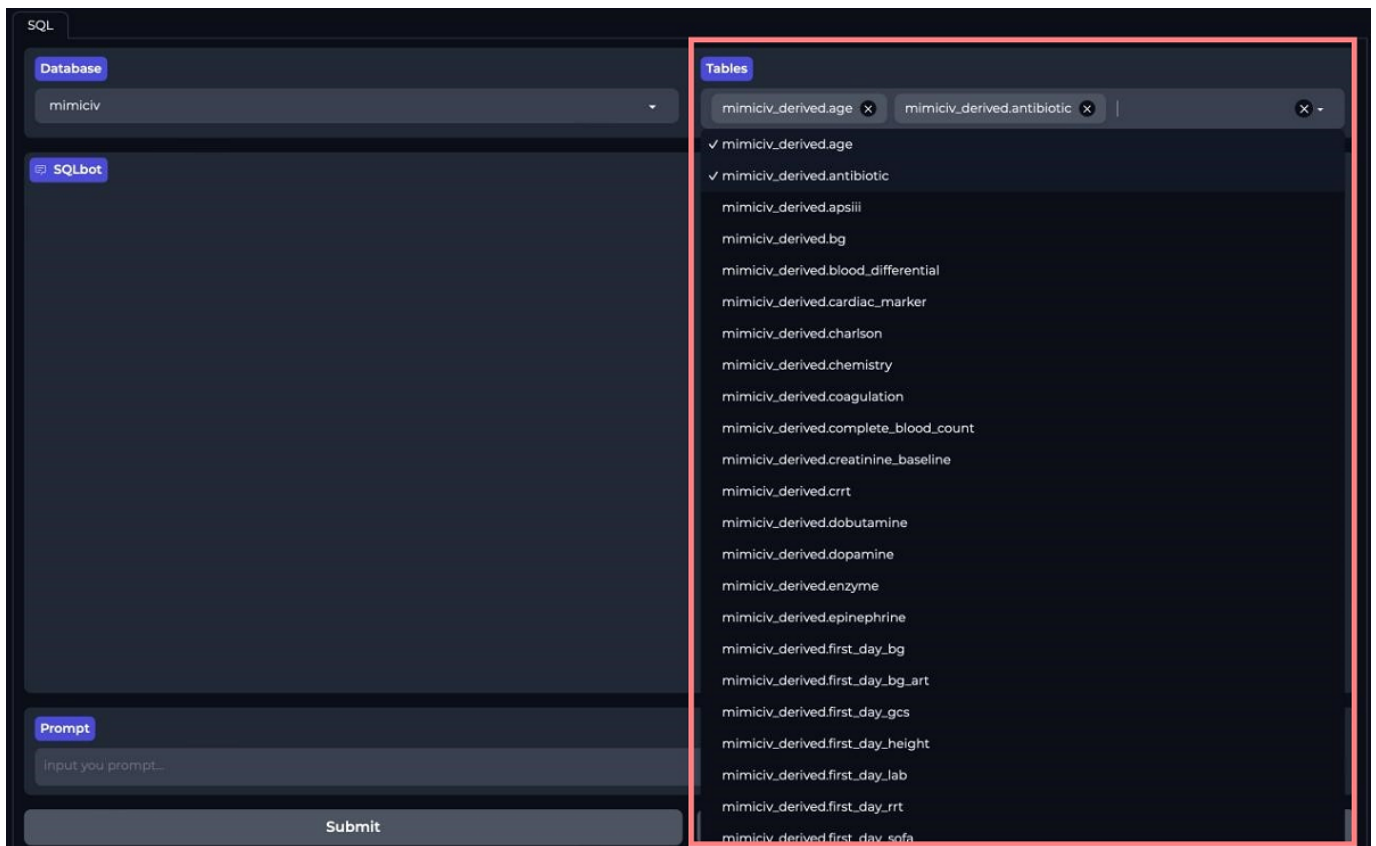


Figure 6. Users input and submit requests on the ICU-GPT platform. ICU-GPT: intensive care unit–generative pretrained transformer.

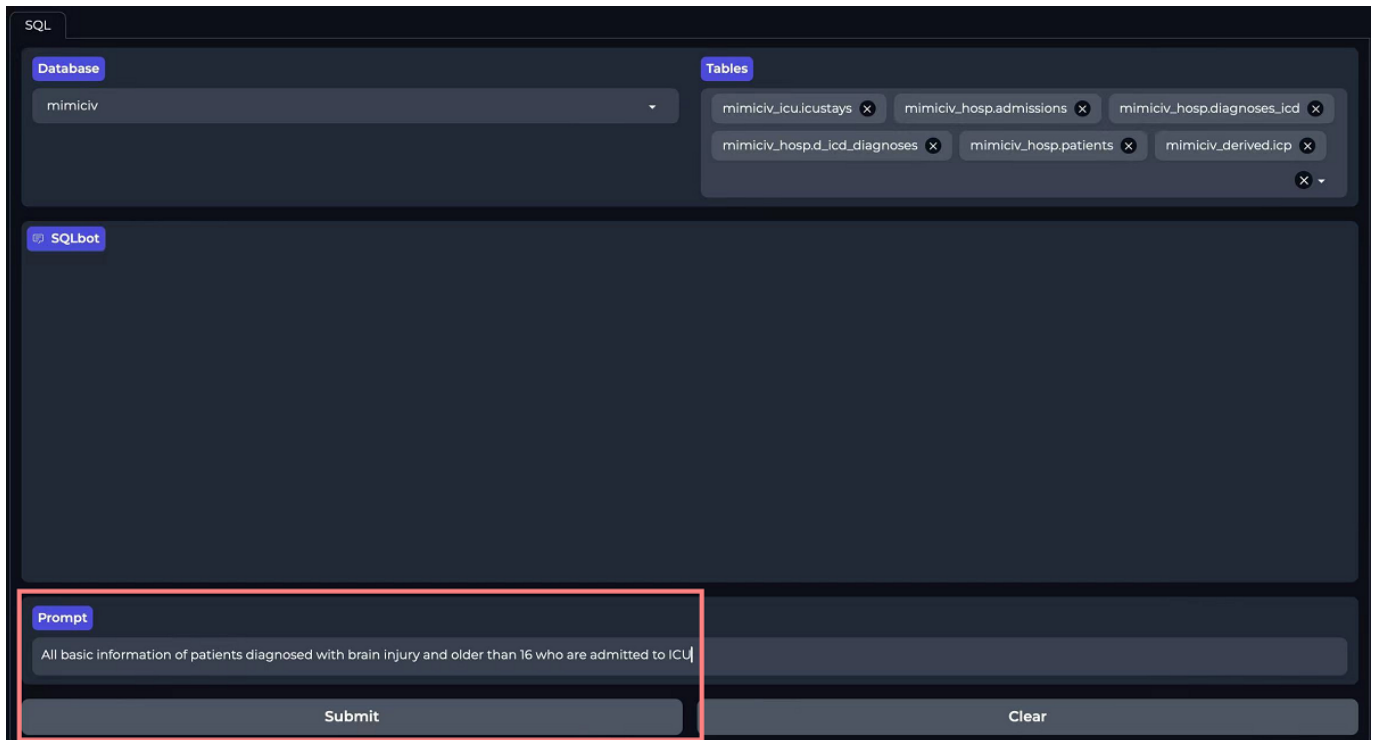
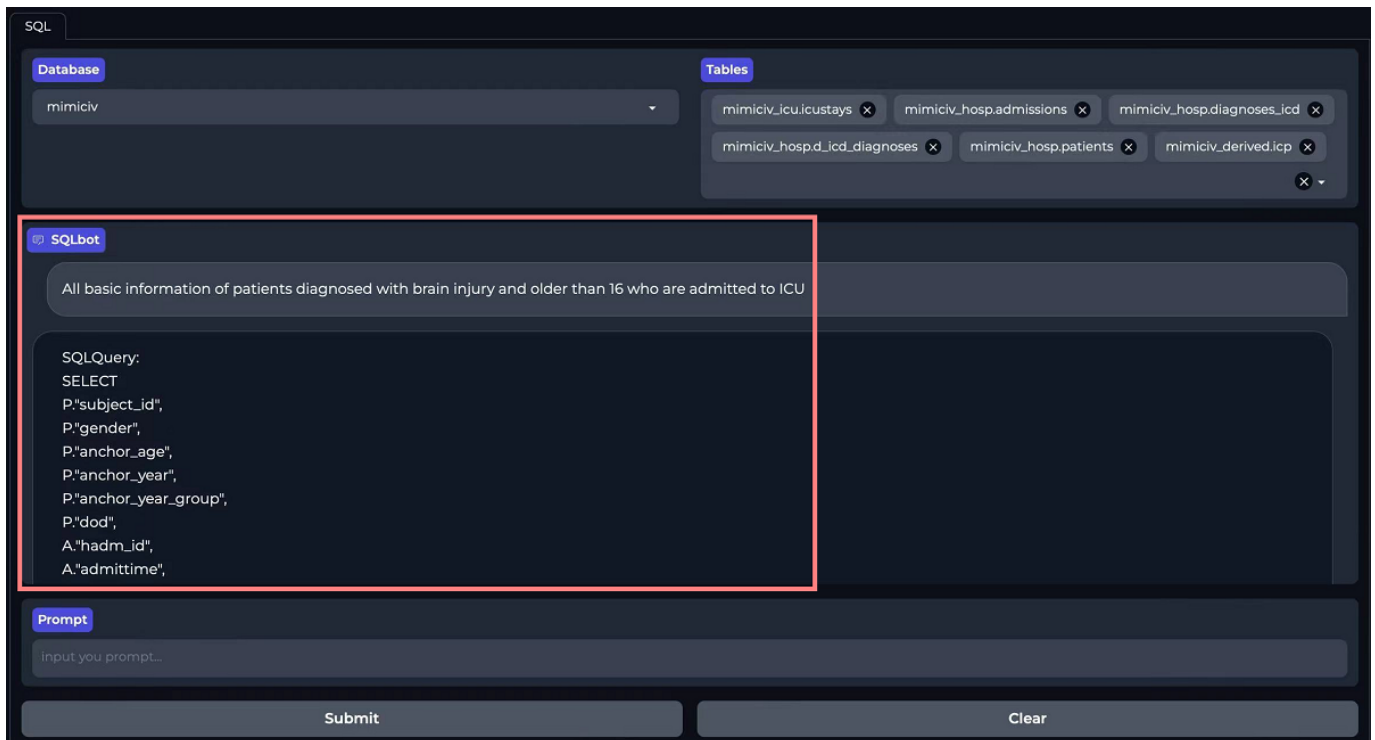


Figure 7. SQL output on the ICU-GPT platform. ICU-GPT: intensive care unit–generative pretrained transformer; SQL: Structured Query Language.



Discussion

Principal Results

In order to conduct the critical care–related database mining as efficiently and thoroughly as possible, we developed a platform in two steps: (1) we first implemented critical care–related database deployment by applying Docker container

technology with 2 visualization tools, Metabase and Superset; (2) ICU-GPT was then developed based on LLMs to achieve SQL query generation and data extraction. By fully utilizing the advantages of open-source technology and being highly replicable and scalable, this platform provides great convenience for clinicians.

The critical care–related public databases cover a large sample size with various aspects during hospitalization,

including demographic, physiologic, laboratory measurements, diagnoses, and medication administration data, which represents a certain degree of universality. While bringing tremendous clinical value, it is accompanied by the increasing complexity of database size and table structures. Even professional database administrators find it challenging to explore the table structures. Much time and effort would be spent on clinicians understanding the table structure and performing data extraction. Therefore, we simplified the complex and time-consuming process into 2 steps.

With Docker container technology, we first provided an efficient and reliable deployment solution to support the widespread application of critical care-related databases. With high repeatability and consistency, the automated deployment solution provides a professional and convenient database management and visualization tool for clinicians to arrange massive amounts of data.

After database deployment, we aimed to simplify the data extraction process. The Python peewee Object Relational Mapper library was considered for analyzing the database table structures automatically, binding them to peewee objects, and encapsulating the corresponding data extraction functions [40]. After multiple tests, although this approach could save the learning cost of SQL and database structures, programming foundation and data analysis experience in personalized extraction and data processing were still required. Despite extensive consideration of potential solutions for this problem, we could not achieve satisfactory outcomes, which led to a stagnation of the development process.

LLMs have recently emerged as powerful tools across many areas of biomedicine. As mentioned, they can rapidly summarize large amounts of text, generate high-quality text from a short description, create code that can help support data analysis, produce images based on a verbal description, and much more. Therefore, based on LLMs, we integrated LangChain and Microsoft AutoGen to develop the ICU-GPT. It not only overcame the token limit of GPT and supported multischema data, but it also achieved the seamless integration of SQL query generation and data extraction.

ICU-GPT empowers users with clinical expertise without data extraction proficiency to extract information from professional databases, significantly reducing the time spent on self-directed learning. However, quantifying the time saved and the efficiency gains from this learning process is complex, making it difficult to compare pre- and postimplementation efficiency improvements.

We agree that a comprehensive evaluation is crucial to demonstrate the system's effectiveness and reliability. While this study focuses on design and implementation, we are actively implementing a thorough evaluation of ICU-GPT's performance metrics. These include SQL query accuracy, system speed, user satisfaction, and the handling of large-scale datasets.

Limitations and Implications

Our study has several limitations. First, our platform only involved MIMIC-III, MIMIC-IV, and eICU-CRD to demonstrate the methodological feasibility and clinical applicability of ICU-GPT in database deployment and data extraction in critical care-related databases. We will expand the research achievements to other databases and enrich the prompt templates of ICU-GPT, such as the Amsterdam UMCdb and high time-resolution intensive care unit dataset (HiRID) [6,41]. Issues of data security and ethical procedures require further improvement. The MIMIC and eICU-CRD databases were deidentified prior to their release, replacing identifiers with random integers while preserving data integrity to maximize data privacy to the fullest extent possible. However, more private databases may be developed in the future. Special attention should be paid particularly to these databases. Adherence to health care regulations, such as the Health Insurance Portability and Accountability Act (HIPAA) in the United States or the General Data Protection Regulation (GDPR) in Europe, is crucial. Second, multiple tests were conducted to verify the reliability and stability of the platform. Systematic evaluation and validation of this platform are warmly welcomed to facilitate further exploration of critical care-related databases. Third, ICU-GPT mainly focuses on structured data SQL generation and analysis. We plan to leverage function calls and the Medical Concept Annotation Toolkit (MedCAT) to achieve unstructured data extraction and analysis like clinical notes from electronic health records in the future [42]. This will enable us to link the unstructured data to biomedical models, facilitating more comprehensive database mining. We will continue to enhance ICU-GPT and strive to achieve automated data processing through Pandas. Fourth, ICU-GPT, as a system built on existing LLMs, may inherit certain biases or limitations from the underlying models. These biases are a known challenge with LLMs due to the nature of their training on large-scale datasets, which may contain imbalances or inaccuracies. To mitigate these risks, we have implemented several measures. We chose OpenAI API compatible models for ICU-GPT development for its incorporation of knowledge bases from the MIMIC database, which includes crucial *ICD* codes and itemid data that other models lack. The system is designed to be used as a human-in-the-loop tool rather than an autonomous system, with users reviewing and validating the generated outputs. We also actively monitor the system's performance in real-world applications and incorporate user feedback to iteratively improve accuracy and reliability. Moreover, periodic audits are conducted to identify and address potential biases in the system's responses. Fifth, it is indispensable to note that AI cannot replace human intelligence and judgment. Cautious inspection and correction are required for platform improvement.

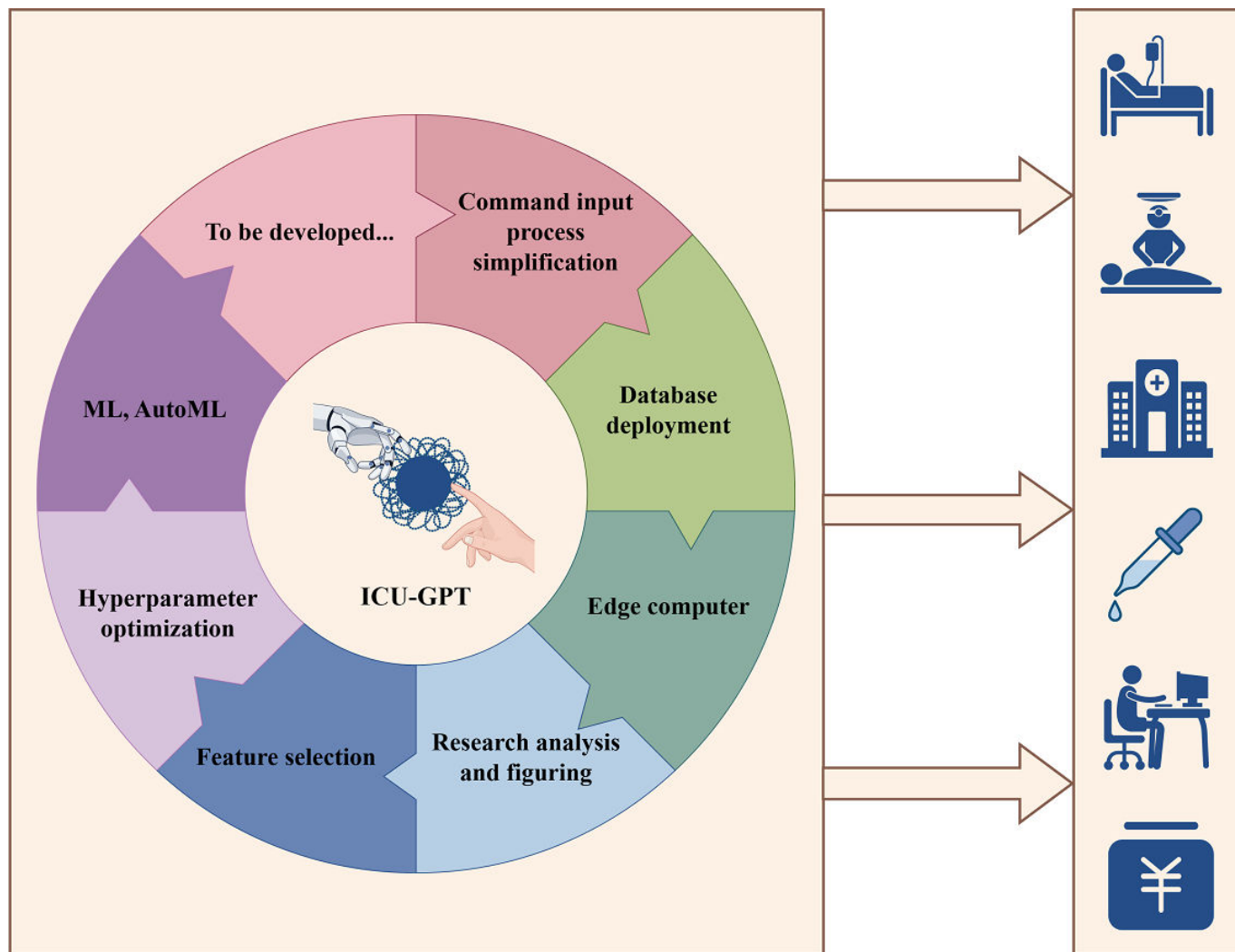
Despite the aforementioned limitations, this study addresses the development of ICU-GPT to simplify critical care-related database deployment and extraction via LLMs. Further work may focus on enhancing the intelligence level of ICU-GPT, such as achieving autonomous SQL generation

based on complex requests, rather than relying solely on fixed templates. It is also anticipated that researchers and developers will create advanced tools for the auxiliary diagnosis and treatment of critical illnesses, as well as sophisticated diagnostic decision support systems based on ICU-GPT.

The massive expansion of critical care-related databases and functions in ICU-GPT will offer enormous potential to uncover patterns, derive predictive models,

tailor interventions, and enable precision medicine through data science [43]. This will further hold immense potential and promising broad prospects for patient care, clinical work, hospital development, academic progress, administrative management, and business improvement (Figure 8). This area of research is ripe for exploration and could lead to groundbreaking advancements in the field.

Figure 8. Expansion of ICU-GPT functions and its potential benefits. ICU-GPT will bring significant benefits after the expansion of its functions, enhancing patient care, clinical work, hospital development, academic progress, administrative management, and business improvement. AutoML: automated machine learning; ICU-GPT: intensive care unit-generative pretrained transformer; ML: machine learning. Figure created using Figdraw.



Conclusions

Leveraging our automated deployment platform and ICU-GPT model, clinicians can now easily visualize, extract, and organize public critical care-related datasets with greater efficiency and flexibility compared to manual methods. By reducing the time and complexity of bioinformatics analysis, our platform and customized LLM enable clinicians without programming expertise to harness large ICU databases for

clinical research advancements. The end-to-end solution from deployment to analysis makes complex data more accessible to frontline ICU staff, unlocking the potential of big data to enhance evidence-based care and outcomes for critically ill patients. Overall, this work demonstrates the promise of thoughtful human-AI collaboration in transforming critical care delivery through data-driven insights.

Data Availability

The code of the developed platform is available online [38].

Authors' Contributions

ZY, SSX, and XL designed this study. ZY, XL, NX, YC, and SW completed the coding for the database deployment. ZY, SSX, MYM, MH, SL, and YMZ completed the coding for intensive care unit–generative pretrained transformer (ICU-GPT) development. SSX, XL, and LZ drafted the manuscript. LZ and JXZ revised the manuscript. All authors read and approved the final manuscript. JXZ and LZ contribute equally as corresponding authors.

Conflicts of Interest

None declared.

Multimedia Appendix 1

Case demonstration of the platform applicability based on Python.

[[DOCX File \(Microsoft Word File\), 50 KB-Multimedia Appendix 1](#)]

References

1. Johnson AEW, Pollard TJ, Shen L, et al. MIMIC-III, a freely accessible critical care database. *Sci Data*. May 24, 2016;3:160035. [Accessed 2024-11-10] [doi: [10.1038/sdata.2016.35](https://doi.org/10.1038/sdata.2016.35)] [Medline: [27219127](https://pubmed.ncbi.nlm.nih.gov/27219127/)]
2. Johnson AEW, Bulgarelli L, Shen L, et al. Author correction: MIMIC-IV, a freely accessible electronic health record dataset. *Sci Data*. Jan 16, 2023;10(1):31. [Accessed 2024-11-10] [doi: [10.1038/s41597-023-01945-2](https://doi.org/10.1038/s41597-023-01945-2)] [Medline: [36646711](https://pubmed.ncbi.nlm.nih.gov/36646711/)]
3. Johnson A, Bulgarelli L, Pollard T, Horng S, Celi LA, Mark R. MIMIC-IV (version 2.2). *PhysioNet*. 2023. [doi: [10.13026/6mm1-ek67](https://doi.org/10.13026/6mm1-ek67)]
4. Pollard TJ, Johnson AEW, Raffa JD, Celi LA, Mark RG, Badawi O. The eICU Collaborative Research Database, a freely available multi-center database for critical care research. *Sci Data*. Sep 11, 2018;5:180178. [doi: [10.1038/sdata.2018.178](https://doi.org/10.1038/sdata.2018.178)] [Medline: [30204154](https://pubmed.ncbi.nlm.nih.gov/30204154/)]
5. O'Halloran HM, Kwong K, Veldhoen RA, Maslove DM. Characterizing the patients, hospitals, and data quality of the eICU Collaborative Research Database. *Crit Care Med*. Dec 2020;48(12):1737-1743. [doi: [10.1097/CCM.0000000000004633](https://doi.org/10.1097/CCM.0000000000004633)] [Medline: [33044284](https://pubmed.ncbi.nlm.nih.gov/33044284/)]
6. Thorat PJ, Peppink JM, Driessen RH, et al. Sharing ICU patient data responsibly under the Society of Critical Care Medicine/European Society of Intensive Care Medicine Joint Data Science Collaboration: the Amsterdam University Medical Centers Database (AmsterdamUMCdb) example. *Crit Care Med*. Jun 1, 2021;49(6):e563-e577. [doi: [10.1097/CCM.0000000000004916](https://doi.org/10.1097/CCM.0000000000004916)] [Medline: [33625129](https://pubmed.ncbi.nlm.nih.gov/33625129/)]
7. Rahman A, Chang Y, Dong J, et al. Early prediction of hemodynamic interventions in the intensive care unit using machine learning. *Crit Care*. Nov 14, 2021;25(1):388. [doi: [10.1186/s13054-021-03808-x](https://doi.org/10.1186/s13054-021-03808-x)] [Medline: [34775971](https://pubmed.ncbi.nlm.nih.gov/34775971/)]
8. Chen H, Zhu Z, Zhao C, et al. Central venous pressure measurement is associated with improved outcomes in septic patients: an analysis of the MIMIC-III database. *Crit Care*. Jul 14, 2020;24(1):433. [doi: [10.1186/s13054-020-03109-9](https://doi.org/10.1186/s13054-020-03109-9)] [Medline: [32665010](https://pubmed.ncbi.nlm.nih.gov/32665010/)]
9. Desautels T, Calvert J, Hoffman J, et al. Prediction of sepsis in the intensive care unit with minimal electronic health record data: a machine learning approach. *JMIR Med Inform*. Sep 30, 2016;4(3):e28. [doi: [10.2196/medinform.5909](https://doi.org/10.2196/medinform.5909)] [Medline: [27694098](https://pubmed.ncbi.nlm.nih.gov/27694098/)]
10. Jeon ET, Lee HJ, Park TY, et al. Machine learning-based prediction of in-ICU mortality in pneumonia patients. *Sci Rep*. Jul 17, 2023;13(1):11527. [doi: [10.1038/s41598-023-38765-8](https://doi.org/10.1038/s41598-023-38765-8)] [Medline: [37460837](https://pubmed.ncbi.nlm.nih.gov/37460837/)]
11. Giannini HM, Ginestra JC, Chivers C, et al. A machine learning algorithm to predict severe sepsis and septic shock: development, implementation, and impact on clinical practice. *Crit Care Med*. Nov 2019;47(11):1485-1492. [doi: [10.1097/CCM.0000000000003891](https://doi.org/10.1097/CCM.0000000000003891)] [Medline: [31389839](https://pubmed.ncbi.nlm.nih.gov/31389839/)]
12. Godri Pollitt KJ, Peccia J, Ko AI, et al. COVID-19 vulnerability: the potential impact of genetic susceptibility and airborne transmission. *Hum Genomics*. May 12, 2020;14(1):17. [doi: [10.1186/s40246-020-00267-3](https://doi.org/10.1186/s40246-020-00267-3)] [Medline: [32398162](https://pubmed.ncbi.nlm.nih.gov/32398162/)]
13. Esteban A, Frutos-Vivar F, Ferguson ND, et al. Sepsis incidence and outcome: contrasting the intensive care unit with the hospital ward. *Crit Care Med*. May 2007;35(5):1284-1289. [doi: [10.1097/01.CCM.0000260960.94300.DE](https://doi.org/10.1097/01.CCM.0000260960.94300.DE)] [Medline: [17414725](https://pubmed.ncbi.nlm.nih.gov/17414725/)]
14. Fleuren LM, Klausch TL, Zwager CL, et al. Machine learning for the prediction of sepsis: a systematic review and meta-analysis of diagnostic test accuracy. *Intensive Care Med*. Mar 2020;46(3):383-400. [doi: [10.1007/s00134-019-05872-y](https://doi.org/10.1007/s00134-019-05872-y)] [Medline: [31965266](https://pubmed.ncbi.nlm.nih.gov/31965266/)]
15. Sendak MP, Ratliff W, Sarro D, et al. Real-world integration of a sepsis deep learning technology into routine clinical care: implementation study. *JMIR Med Inform*. Jul 15, 2020;8(7):e15182. [doi: [10.2196/15182](https://doi.org/10.2196/15182)] [Medline: [32673244](https://pubmed.ncbi.nlm.nih.gov/32673244/)]

16. Farič N, Hinder S, Williams R, et al. Early experiences of integrating an artificial intelligence-based diagnostic decision support system into radiology settings: a qualitative study. *J Am Med Inform Assoc*. Dec 22, 2023;31(1):24-34. [doi: [10.1093/jamia/ocad191](https://doi.org/10.1093/jamia/ocad191)] [Medline: [37748456](https://pubmed.ncbi.nlm.nih.gov/37748456/)]
17. Sjoding MW, Taylor D, Motyka J, et al. Deep learning to detect acute respiratory distress syndrome on chest radiographs: a retrospective study with external validation. *Lancet Digit Health*. Jun 2021;3(6):e340-e348. [doi: [10.1016/S2589-7500\(21\)00056-X](https://doi.org/10.1016/S2589-7500(21)00056-X)] [Medline: [33893070](https://pubmed.ncbi.nlm.nih.gov/33893070/)]
18. Docker. URL: <https://docs.docker.com> [Accessed 2024-06-05]
19. PostgreSQL. URL: <https://www.postgresql.org> [Accessed 2024-06-05]
20. Metabase. URL: <https://www.metabase.com> [Accessed 2024-06-05]
21. Superset. URL: <https://superset.apache.org> [Accessed 2024-06-05]
22. Key concepts. OpenAI. URL: <https://platform.openai.com/docs/introduction> [Accessed 2024-06-05]
23. DAMO-convai/bird. GitHub. URL: <https://github.com/AlibabaResearch/DAMO-ConvAI/tree/main/bird> [Accessed 2024-11-10]
24. taoyds/spider. GitHub. URL: <https://github.com/taoyds/spider> [Accessed 2024-11-10]
25. BIRD-SQL. URL: <https://bird-bench.github.io> [Accessed 2024-11-10]
26. Spider: Yale semantic parsing and text-to-SQL challenge. Language, Information, and Learning at Yale (LILY). URL: <https://yale-lily.github.io/spider> [Accessed 2024-11-10]
27. Marshan A, Almutairi AN, Ioannou A, Bell D, Monaghan A, Arzoky M. MedT5SQL: a transformers-based large language model for text-to-SQL conversion in the healthcare domain. *Front Big Data*. 2024;7:1371680. [doi: [10.3389/fdata.2024.1371680](https://doi.org/10.3389/fdata.2024.1371680)] [Medline: [38988646](https://pubmed.ncbi.nlm.nih.gov/38988646/)]
28. Qi J, Tang J, He Z, et al. RASAT: integrating relational structures into pretrained seq2seq model for text-to-SQL. arXiv. Preprint posted online on May 14, 2022. [doi: [10.48550/arXiv.2205.06983](https://doi.org/10.48550/arXiv.2205.06983)]
29. Li H, Zhang J, Li C, Chen H. RESDSQL: decoupling schema linking and skeleton parsing for text-to-SQL. arXiv. Preprint posted online on Feb 12, 2023. [doi: [10.48550/arXiv.2302.05965](https://doi.org/10.48550/arXiv.2302.05965)]
30. Wang P, Shi T, Reddy CK. Text-to-SQL generation for question answering on electronic medical records. arXiv. Preprint posted online on Jul 28, 2019. [doi: [10.48550/arXiv.1908.01839](https://doi.org/10.48550/arXiv.1908.01839)]
31. ollama/ollama. GitHub. URL: <https://github.com/ollama/ollama> [Accessed 2024-06-05]
32. langchain-ai/langchain. GitHub. URL: <https://github.com/langchain-ai/langchain> [Accessed 2024-06-05]
33. microsoft/autogen. GitHub. URL: <https://github.com/microsoft/autogen> [Accessed 2024-06-05]
34. McKinney W. Data structures for statistical computing in python. Presented at: Python in Science Conference; Jun 28-30, 2010; Austin, TX, United States. [doi: [10.25080/Majora-92bf1922-00a](https://doi.org/10.25080/Majora-92bf1922-00a)]
35. Abid A, Abdalla A, Abid A, Khan D, Alfozan A, Zou J. Gradio: hassle-free sharing and testing of ML models in the wild. arXiv. Preprint posted online on Jun 6, 2019. [doi: [10.48550/arXiv.1906.02569](https://doi.org/10.48550/arXiv.1906.02569)]
36. Wu Q, Bansal G, Zhang J, et al. AutoGen: enabling next-gen LLM applications via multi-agent conversation. arXiv. Preprint posted online on Aug 16, 2023. [doi: [10.48550/arXiv.2308.08155](https://doi.org/10.48550/arXiv.2308.08155)]
37. Customize speaker selection. AutoGen. URL: https://microsoft.github.io/autogen/docs/topics/groupchat/customized_speaker_selection [Accessed 2024-06-05]
38. ICU-GPT. GitHub. URL: <https://github.com/ICU-GPT/icu-gpt> [Accessed 2024-06-05]
39. Dockerfile of ICU-GPT. GitHub. URL: <https://github.com/ICU-GPT/icu-gpt/blob/main/icugpt/docker/Dockerfile> [Accessed 2024-06-05]
40. peewee. URL: <http://docs.peewee-orm.com> [Accessed 2024-06-05]
41. HiRID, a high time-resolution ICU dataset. PhysioNet. URL: <https://physionet.org/content/hirid/1.0> [Accessed 2024-06-05]
42. Kraljevic Z, Searle T, Shek A, et al. Multi-domain clinical natural language processing with MedCAT: the Medical Concept Annotation Toolkit. *Artif Intell Med*. Jul 2021;117:102083. [doi: [10.1016/j.artmed.2021.102083](https://doi.org/10.1016/j.artmed.2021.102083)] [Medline: [34127232](https://pubmed.ncbi.nlm.nih.gov/34127232/)]
43. Komorowski M, Celi LA, Badawi O, Gordon AC, Faisal AA. The Artificial Intelligence Clinician learns optimal treatment strategies for sepsis in intensive care. *Nat Med*. Nov 2018;24(11):1716-1720. [doi: [10.1038/s41591-018-0213-5](https://doi.org/10.1038/s41591-018-0213-5)] [Medline: [30349085](https://pubmed.ncbi.nlm.nih.gov/30349085/)]

Abbreviations

AI: artificial intelligence

Amsterdam UMCdb: Amsterdam University Medical Center Database

API: application programming interface

eICU-CRD: Electronic Intensive Care Unit Collaborative Research Database
GDPR: General Data Protection Regulation
HIPAA: Health Insurance Portability and Accountability Act
HiRID: high time-resolution intensive care unit dataset
ICD: *International Classification of Diseases*
ICU: intensive care unit
ICU-GPT: intensive care unit-generative pretrained transformer
LLM: large language model
MedCAT : Medical Concept Annotation Toolkit
MIMIC: Medical Information Mart for Intensive Care
MIMIC-IV-ED: Medical Information Mart for Intensive Care IV – Emergency Department
ML: machine learning
SQL: Structured Query Language

Edited by Jeffrey Klann; peer-reviewed by Alaa Marshan, Jorge Valverde-Rebaza; submitted 13.06.2024; final revised version received 30.11.2024; accepted 18.12.2024; published 12.03.2025

Please cite as:

Yang Z, Xu SS, Liu X, Xu N, Chen Y, Wang S, Miao MY, Hou M, Liu S, Zhou YM, Zhou JX, Zhang L
Large Language Model-Based Critical Care Big Data Deployment and Extraction: Descriptive Analysis
JMIR Med Inform 2025;13:e63216
URL: <https://medinform.jmir.org/2025/1/e63216>
doi: [10.2196/63216](https://doi.org/10.2196/63216)

© Zhongbao Yang, Shan-Shan Xu, Xiaozhu Liu, Ningyuan Xu, Yuqing Chen, Shuya Wang, Ming-Yue Miao, Mengxue Hou, Shuai Liu, Yi-Min Zhou, Jian-Xin Zhou, Linlin Zhang. Originally published in JMIR Medical Informatics (<https://medinform.jmir.org>), 12.03.2025. This is an open-access article distributed under the terms of the Creative Commons Attribution License (<https://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work, first published in JMIR Medical Informatics, is properly cited. The complete bibliographic information, a link to the original publication on <https://medinform.jmir.org/>, as well as this copyright and license information must be included.