Paper ET15

Front-Loading Statistical Programming Using Automated Synthetic Data

Faezeh Moshfeghifar, Novo Nordisk A/S, 2880 Bagsværd, Denmark
Hemanth Padmakar, Novo Nordisk A/S, 2880 Bagsværd, Denmark

ABSTRACT

An efficient submission of clinical trial data is crucial to successfully introducing new treatments to patients in need. Novo Nordisk's "Fit4Take-Off" aspiration aims to submit a New Drug Application to the pre-defined four markets within four days of the Last Patient Last Visit (LPLV). To support this aspiration, Novo Nordisk has developed an in-house application called Kassandra, which automatically generates trial-specific synthetic data for clinical trials across therapeutic areas.

Kassandra's synthetic data generation capabilities offer a solution to streamline clinical trial setup processes. It helps the Electronic Data Capture (EDC) system check procedures, ensuring a robust and adaptable data entry approach surpassing traditional methods. It also enables the front-loading of statistical programming tasks ahead of the First Patient First Visit (FPFV), allowing statistical programmers to initiate these tasks earlier in the trial conduct phase.

Kassandra's Python-based algorithm automatically injects synthetic data into the EDC system, utilizing the Application Programming Interface (API) injection method, which improves accuracy and efficiency compared to manual and Robotic Process Automation (RPA) approaches. Its user-friendly environment allows for easy modification of both standard and non-standard forms.

1. INTRODUCTION

The utilization of automation in clinical trial processes is steadily increasing due to its potential to expedite drug development, enhance data accuracy and quality, and reduce operational costs. One notable application of automation is the creation of synthetic data for Electronic Data Capture (EDC) systems. Synthetic data is precious in clinical trials as it can be used to test the EDC system's functionality in a "Training" environment and validate its accuracy without compromising patient privacy or the Production environment. Moreover, the timely generation of synthetic trial data can aid in identifying data quality issues, optimizing, and front-loading downstream processes such as the Study Data Tabulation Model (SDTM) and Analysis Data Model (ADaM) programming. Automating these processes can significantly reduce test time, improve efficiency of data management activities and setup analyses of the clinical data, reduce submission timelines, and ultimately accelerate the development of new drugs. This paper describes Novo Nordisk's approach to generating synthetic data for the EDC system and its benefits in enhancing data quality and accelerating drug development.

1.1 PROBLEM STATEMENT

Generating comprehensive and accurate synthetic data for clinical trials is a challenging task that requires populating patient data with all possible combinations of answers to each question on the Case Report Form (CRF). However, this can be time-consuming and resource-intensive, especially for large trials requiring many patients to cover all possible scenarios. In addition, clinical trials involve numerous dependencies and rules that must be followed to ensure data accuracy and reliability. For instance, if a patient becomes ineligible to continue the trial, all remaining visits for that patient must be skipped, and the “End of the Study” visit must be recorded for the patient. Failure to follow these rules can compromise the synthetic data’s accuracy, reliability, and usefulness.

The current approach to generating synthetic data is labor-intensive. It needs to provide a complete understanding of how well the synthetic data is covers all possible ways that individual questions on the forms could be answered. This issue can lead to incomplete or inaccurate synthetic data sets, compromising their usefulness to the downstream stakeholders. Therefore, there is a need for an automated approach that can generate comprehensive synthetic data more efficiently and effectively.

1.2 AUTOMATED DATA-ENTRY APPROACHES

Generating synthetic EDC data manually through the EDC user interface can be time-consuming and error-prone, limiting the accuracy and efficiency of synthetic clinical trial data entry. Therefore, a more practical

approach that leverages automation to streamline the process and improve the quality of the synthetic data generated is needed.
One example of an automated approach that relies on the user interface is the Robotic Process Automation (RPA) method. RPA methods automate routine tasks by mimicking user actions to enter data into computer systems, leading to greater accuracy and speed than manual entry. Automated data entry using RPA involves software robots or bots interacting with digital systems to capture and interpret applications for processing transactions, manipulating data, triggering responses, and communicating with other digital systems. Despite the efficiency of RPA, it has limitations. RPA systems require initial programming and continuous monitoring to handle exceptions and errors that may not be predictable at the outset. Additionally, RPA systems require updates to adapt to changes in trial protocols, data formats, or user interfaces.
A more advanced approach to automated data entry is using Application Programming Interfaces (APIs) to enter synthetic data directly. APIs provide a direct connection between different systems, allowing for real-time data exchange and integration. By leveraging APIs, an automated approach can provide a robust and adaptable solution for generating synthetic data and streamlining the data entry.

1.3 NOVO NORDISK'S CONTRIBUTION

Novo Nordisk has developed an in-house solution named Kassandra, which utilizes the power of APIs to enhance the efficiency and accuracy of clinical data entry. Kassandra eliminates the need for manual data entry by test engineers, automatically generating data values for each item and transmitting them to the EDC database via an API connection. Kassandra uses APIs to ensure a robust and adaptable approach to data entry that outperforms traditional methods by facilitating real-time data exchange and integration.

Additionally, Kassandra provides an overview of the coverage of each question, enabling test engineers to identify missing data and add it accordingly. This approach saves test engineers significant time and provides statistical programmers with early access to test data (SDTM format), enabling front-loading of trial programming activities. By enabling statistical programmers to commence tasks earlier, Kassandra can be part of decreasing the time between the LPLV and the final reporting of a clinical trial.

Our solution is one approach in the clinical development flow that streamlines the data entry process while maintaining high data integrity and compliance standards. Kassandra represents a step forward in using APIs for clinical data entry.

2. SYSTEM ARCHITECTURE AND METHODOLOGY

The architecture of Kassandra is designed to facilitate robust data handling and ensure seamless integration with the EDC system. The workflow is depicted in Figure 1, comprising two main components: a user interface and a data processing backend.

- Kassandra's **User Interface (UI)** is a web application accessible to internal Novo Nordisk users. It is designed for intuitive navigation and has features that enhance the user experience. Users can upload the trial-specific input files, including the trial protocol report, the configuration file, as well as user-specific interaction properties. The UI includes features for reporting errors, enabling users to flag any issues with the data generated and injected by Kassandra. Users can also receive real-time status updates on the progress of data processing tasks, enabling them to monitor the system's performance. An overview of the coverage of the answers to each question provides users with insights into missing data, helping EDC test engineers to identify and add the missing information. This feature saves time for test engineers and provides statistical programmers with early access to test data.
- **The Data Processing Backend** serves as Kassandra's engine, where the principal algorithm is executed. It is responsible for generating synthetic data and ensuring data integrity and compliance. The backend of Kassandra incorporates several modules, including the Data Loading, Sampling, Injection, and Logging modules. These modules work together to automate data entry and ensure data integrity and compliance. The Data Loading module imports and parses input data from diverse formats, while the Sampling module samples values for the different items according to the trial's constraints. The Injection module translates the sampled data into a format that is readable by the database of the EDC system and sends it via a secure API connection, while the Logging module tracks all system activities to support compliance and facilitate troubleshooting.

The development of Kassandra was conducted using Python, which was selected for its extensive libraries and strong support for scientific computing and data analysis. Essential libraries such as pandas for data handling, requests for API integration, and blob for data storage form the backbone of Kassandra's functionality. To maintain code quality and reliability, a combination of unit test and pytest frameworks facilitated rigorous testing, ensuring that each iteration of the software met the high standards required for clinical applications. These testing protocols were integrated into a continuous integration pipeline, ensuring that any changes to the software were thoroughly tested before being deployed. Using Python and these testing protocols helped to ensure that each incremental Kassandra release was robust, reliable, and capable of meeting the demands of automation.

2.1 DATA LOADING MODULE

The Data Loading module is responsible for importing and parsing input data from diverse formats, including the trial protocol, trial-specific assumptions, user-defined settings, and API connection metadata. Kassandra processes three principal types of input files containing study-specific metadata generated through the study setup process:

- **Trial Protocol Report:** This trial-specific file encapsulates the clinical trial protocol set forth in the Central Designer in a machine-readable format. It is crucial for ensuring the fidelity of the clinical data structure within the EDC Test environment.
- **Time and Events (TnE) Report:** The TnE report, another trial-specific document, outlines the timeline and event schedule of the clinical trial. It details the sequence and timing of data collection events and the specific forms to be completed at each visit, thereby stating the temporal framework for data entry.
- **Configuration File:** This file comprises additional metadata, including API configuration and user-specific interaction properties. It serves as the blueprint for system behavior and data handling protocols.

Each input file plays a critical role in Kassandra's operation, providing a comprehensive dataset for the EDC system to mirror the trial's design and procedural intricacies accurately. The Data Loading module ensures that the input data is properly formatted, error-free, and ready for processing by the other modules within the backend.

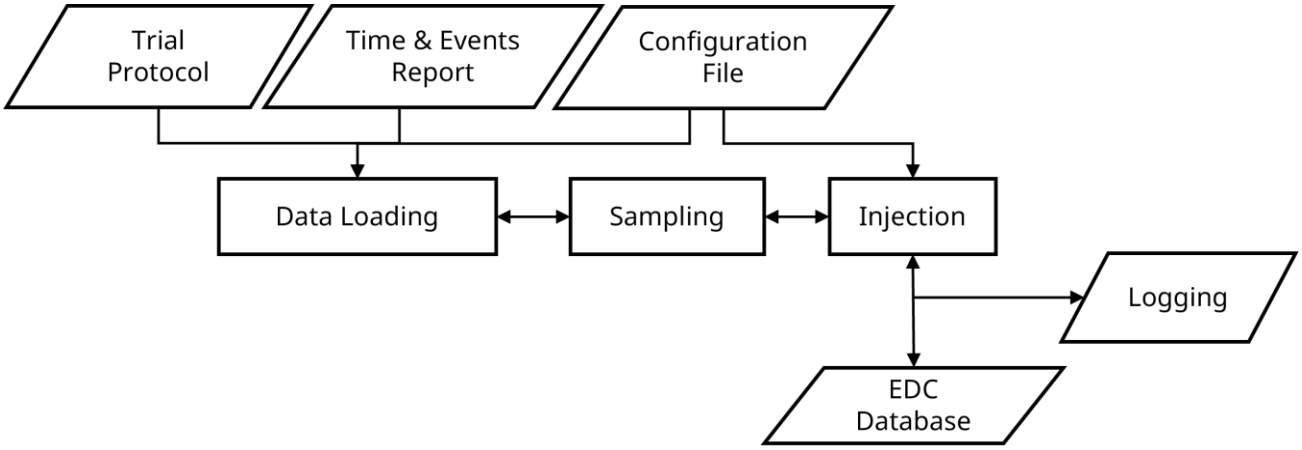

Figure 1. The Kassandra flowchart. This flowchart illustrates the different components of Kassandra's system architecture and methodology. The User Interface allows users to upload trial-specific input files while the Data Loading module imports and parses input data from diverse formats. The Sampling module generates synthetic data that reflects the trial's design and procedural intricacies, which is then reformatted by the Injection module and sent to the EDC database via a secure API connection. The Logging module tracks all system activities to support compliance and facilitate troubleshooting.

2.2 SAMPLING MODULE

The Sampling module generates synthetic data that reflects the trial's design and user specifications. It categorizes each item based on its type and samples the item values based on trial specifications and requirements. Additionally, it considers the dependencies among different forms and items involved in the trial.

By sampling all possible outcomes and filling in all combinations, this module facilitates a complete data flow check before FPFV, ensuring that all edit checks work as intended and enhancing the trial's validity and reliability. However, it is essential to note that the sampled data may not always be consistent with actual patient data. The Sampling module interacts with the Data Loading, Injection, and Logging modules to provide a comprehensive and accurate representation of data that enhances the trial's validity and reliability.

2.3 INJECTION AND API COMMUNICATION LAYER

The Injection module is responsible for reformatting the sampled data to align with the EDC database structure. The reformatting process is necessary to ensure the synthetic data is readable and can be integrated into the EDC system.

Once the sampled data is reformatted, Kassandra establishes a secure API connection with the EDC system to inject the data. The API connection allows for a seamless data transfer between Kassandra and the EDC system. In the event of any errors reported by the EDC system, Kassandra processes the error and resamples the item before injecting it again. This ensures the synthetic data is comprehensive and accurate, aligning with the trial's design and procedural intricacies.

3. DISCUSSION AND CONCLUSION

In conclusion, Kassandra represents an innovative approach to automating clinical trial processes, streamlining the data entry process while maintaining high data integrity and compliance standards. Kassandra's user-friendly environment gives users a clear overview of the sampled data. Its Python-based algorithm automatically injects the sampled data into the EDC system, utilizing the API injection method, which improves accuracy and efficiency compared to traditional approaches.

3.1 BENEFITS OF KASSANDRA TO VARIOUS STAKEHOLDERS

Kassandra's synthetic data generation capabilities offer significant downstream benefits to various stakeholder groups involved in clinical trial development and reporting. Kassandra enables stakeholders to test their systems and processes and front-load several programming activities before FPFV by providing early access to comprehensive and accurate synthetic data. Biostatistics is one area that can benefit from such synthetic data, as it can reduce the trial programming workload around and during Database Lock (DBL). The use of synthetic SDTM data to front-load programming ADaM datasets and Table, Figures, and Listings (TFLs) during the trial conduct phase can also benefit long-duration trials, where it can take years to see data points collected at late scheduled visits. Synthetic data can provide a comprehensive and accurate representation of data that reflects the trial's design and procedural intricacies across all visits and forms, minimizing the risk of human error while adhering to rigorous regulatory standards.

Additionally, EDC test engineers can benefit directly from Kassandra's synthetic data generation capabilities by eliminating the need for manual data entry. By easing their EDC system check task, Kassandra enables test engineers to test the EDC system's functionality and validate its accuracy. This significantly reduces test time and improves the core data management activities.

3.2 LIMITATION AND FUTURE WORK

While there are still areas for improvement, Kassandra represents a significant step forward in using APIs for clinical data entry. One limitation of Kassandra is the speed of injection through the API. The injection speed can sometimes be slow, and a value may have to be resampled multiple times before it is accepted by the EDC database, which can result in delays in the data entry process. Additionally, while the Sampling module generates synthetic data that accurately reflects the trial's design and procedural intricacies, there is still room for improvement in generating sampled data closer to the true values present in actual patient data. This could be an area for future development, as generating more accurate synthetic data would further enhance the reliability and validity for front-loading activities in downstream trial conduct activities.

Another potential future for Kassandra is to extend its synthetic data generation capabilities to include externally generated data, such as laboratory results or e-source data. While there is already an initial code for this, further development is needed to improve its accuracy and usability for users. This area for future development further enhances Kassandra's capability and streamlines the clinical trial data process. While Kassandra has made significant strides in clinical trial automation, there is still potential for future development in improving the speed and accuracy of its synthetic data generation capabilities.

ACKNOWLEDGEMENTS
We would like to express our gratitude to our colleagues within the Data Management Process and Innovation department and the Data Science Automation group at Novo Nordisk for their invaluable contributions and support to this project.

CONTACT INFORMATION
Contact the author at:
Author Name: FAEZEH MOSHFEGHIFAR
Company: NOVO NORDISK A/S
Address: 2880 BAGSVÆRD, DENMARK
Work Phone: +45-34487714
Email: FZMG@NOVONORDISK.COM