Intelligent Health Systems – From Technology to Data and Knowledge
E. Andrikopoulou et al. (Eds.)

doi:10.3233/SHTI250419

Design and Development of SPADE: A Synthetic Patient Data Generator for Testing Healthcare Data Processing System Capabilities

Tia HADDAD[a], Pushpa KUMARAPELI[a,1], and Souheil KHADDAJ[a]
[a] School of Computer Science and Mathematics, Kingston University London
ORCiD ID: Tia Haddad https://orcid.org/0009-0009-5187-1168
Pushpa Kumarapeli https://orcid.org/0000-0002-9552-6835
Souheil Khaddaj https://orcid.org/0000-0001-9477-4290

Abstract. SPADE (Synthetic PAtient Data Engine) generates realistic synthetic patient data to test healthcare data systems under extreme conditions. It simulates anomalies like outliers and spikes, facilitating system evaluation and contingency planning. Developed in Python, SPADE offers customizable data generation with future plans for API integration and a web-based interface.

Keywords. Synthetic Patient Data, Healthcare Data Systems, Software Testing

1. Introduction

Healthcare data processing systems are rapidly evolving, adopting cutting-edge big data processing techniques, cloud-native architectures, edge computing, and increasingly integrating machine learning and deep learning approaches. These systems support tasks like data extraction, analytics, and decision-making. Due to privacy regulations, evaluating them at scale with real patient datasets is challenging. Synthetic patient data offers an alternative, allowing for the generation of datasets without compromising privacy. While existing generators replicate real data [1], testing system limits requires datasets with exceptional characteristics, such as spikes or outliers [2]. We have developed SPADE (Synthetic Patient Data Engine), a system for generating synthetic patient data. Our objective is to facilitate extensive evaluation by pushing data processing systems to their limits, intentionally "breaking" these systems to enable the design and testing of robust contingency measures.

2. The Design Characteristics of SPADE

SPADE was developed using Python language, adopting iterative and incremental approaches. It first prompts to define the population characteristics, such as size, age, and gender distribution, to generate a data frame where each row represents a patient.

[1] Corresponding Author: Pushpa Kumarapeli; E-mail: p.kumarapeli@kingston.ac.uk.

Gender and age value distribution mimic real-world samples, e.g. the UK population. Future versions will support loading regional and country-specific profiles. Each patient receives an anonymised, customisable alphanumeric identifier. Users can add data frames for diagnoses, test outcomes, examinations, and prescriptions, with SPADE assigning appropriate column types. For example, blood pressure (BP) columns include dates, codes, and systolic/diastolic values, while disease diagnosis columns contain code and date pairs. Users can define data ranges and occurrences, such as specifying up to 10 BP recordings over specific number of years. SPADE also allows the introduction of null values, outliers, and invalid data to simulate real-world or extreme conditions. Data volumes can be scaled to simulate big data characteristics. It produces dataset of 100k records, with 30 columns under 30 seconds. Outputs can be in text file format, or integrated into database systems, including cloud based systems. The current version includes the core data generation engine (fig 1), with future updates planned to introduce interfacing options for both software testers and health informaticians.

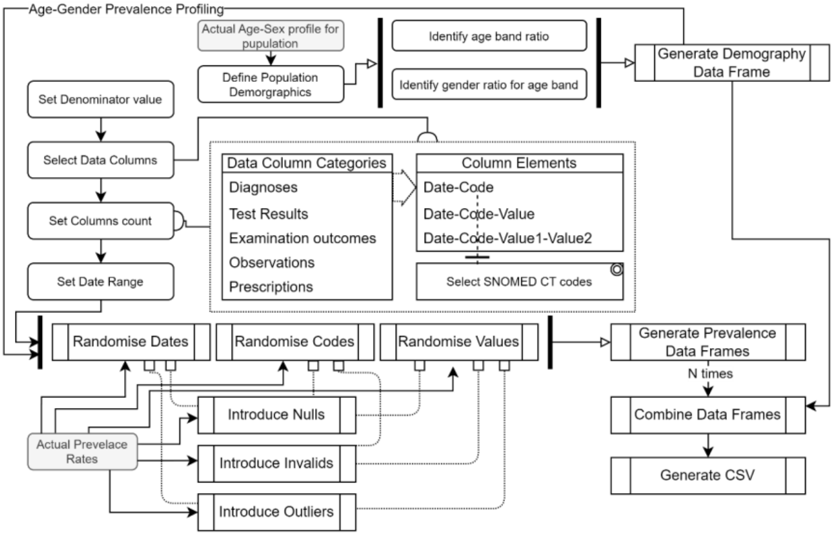

Figure 1. Overall process model of the SPADE system

3. Conclusions

SPADE is a synthetic patient data generator designed to test healthcare software by evaluating how systems perform with datasets that exhibit extreme characteristics. It can also be used by epidemiologists and data modelers to test models for reporting, visualisation, or machine learning. SPADE enables comprehensive stress-testing of data processing systems, facilitating contingency planning and workflow optimisation. Future versions will support multiple classification systems and introduce multiple interfacing options for easier access and integration.

References

[1] Goncalves A, Ray P, Soper B, Stevens J, Coyle L, Sales AP. Generation and evaluation of synthetic patient data. BMC medical research methodology. 2020 Dec;20:1-40.

[2] Helvik BE, Vizarreta P, Heegaard PE, Trivedi K, Mas-Machuca C. Modelling of software failures. Guide to Disaster-Resilient Communication Networks. 2020:141-72.