Workflow Characterization of a Big Data System Model for Healthcare Through Multiformalism

Tancredi Covioli[1], Tommaso Dolci[1][0000−0002−1403−7766], Fabio Azzalini[1][0000−0003−0631−2120], Davide Piantella[1][0000−0003−1542−0326], Enrico Barbierato[2][0000−0003−1466−0248], and Marco Gribaudo[1][0000−0002−1415−5287]

[1] Politecnico di Milano, Piazza Leonardo da Vinci, 32, 20133 Milano, Italy
tancredi.covioli@mail.polimi.it
{tommaso.dolci,fabio.azzalini,davide.piantella,marco.gribaudo}@polimi.it
[2] Università Cattolica del Sacro Cuore, Via della Garzetta 48, 25133 Brescia, Italy
enrico.barbierato@unicatt.it

Abstract. The development of technologies such as cloud computing, IoT, and social networks caused the amount of data generated daily to grow at an incredible rate, giving birth to the trend of Big Data. Big data has emerged in the healthcare field, thanks to the introduction of new tools producing massive amounts of structured and unstructured data. For this reason, medical institutions are moving towards a data-based healthcare, with the goal of leveraging this data to support clinical decision-making through suitable information systems. This comes with the need to evaluate their performance. One of the techniques commonly used is modeling, which consists in performing an evaluation of a model of the system under analysis, without actually implementing it. However, to make an adequate performance assessment of Big Data systems, we need a diversity of volumes and speeds that, due to the sensitivity of data concerning healthcare, is not available. While in other fields this problem is usually solved through the use of synthetic data generators, in healthcare these are few and not specialized in performance evaluation. Therefore, this work focuses on the creation of a synthetic data generator for evaluating the performance of a Big Data system model for healthcare. The dataset used as a reference for creating the generator is MIMIC-III, which contains the digital health records of thousands of patients collected over a time span of multiple years. First, we perform an analysis of the dataset, adopting multiple distribution fitting techniques (e.g., phase-type fitting) to model the temporal distribution of the data. Then, we develop a generator structured as a multi-module library to allow the customization of each component, specifically we propose a multiformalism model to reproduce the patient behavior inside the hospital. Finally, we test the generator by evaluating the performance in different scenarios. Through these experiments, we show the granular control that the generator offers over the synthetic data produced, and the simplicity with which it can be adapted to different uses.

Keywords: performance evaluation · synthetic data generation · Big Data · healthcare data.

1 Introduction

Big Data is flooding the healthcare field thanks to the introduction of new tools for continuous patient monitoring, producing massive amounts of structured and unstructured data every day. For this reason, medical facilities are moving towards data-driven healthcare, with the goal of leveraging this incredible source of information to support clinical decision-making and public health management. This naturally comes with the request to elaborate a multitude of heterogeneous data, resulting in the emergence of new system architectures and new methods to evaluate their performance. One of the evaluation techniques frequently used is modeling, which consists in creating a model of the system under analysis, and performing the evaluation on it instead of implementing and testing the actual system. In order to make an adequate performance assessment of Big Data–centered systems, we need a diversity of volumes and workloads that, due to the sensitivity of data related to healthcare, it is usually not available. In other fields, this problem is solved through the use of synthetic data generators, but in the field of healthcare there are just few available and not specialized for performance evaluation [6].

In this paper we create a synthetic data generator for the evaluation of performances of a Big Data system model focusing on healthcare data. Reference data for the generator is gathered from two sources: *a)* MIMIC-III dataset [15], a large freely-available database comprising de-identified health-related observations (called "events") associated with over forty thousand patients who stayed in intensive care units (ICU) at the Beth Israel Deaconess Medical Center between 2001 and 2012; *b)* MIMIC-III Waveform Matched Dataset [18], a dataset associated to the original MIMIC-III containing the digitized vital signals ("waveforms") recorded for all patients that stayed in ICU.

The rest of the paper is organized as follows: Section 2 describes the state of the art, Section 3 illustrates MIMIC-III and the analyses performed on it, Section 4 introduces our synthetic data generator, Section 5 shows the experimental evaluation of the generator, Section 6 sums up and concludes the paper.

2 Related Work

As previously mentioned, the advent of Big Data in healthcare led to the need of new systems for data processing and performance evaluate. The first challenge is finding datasets that contain both large volumes of structured and unstructured data, and with a fine enough temporal granularity to identify the peak usage in order to make a proper and complete performance assessment. The scarcity of publicly available datasets that meet these characteristics causes the need to consider generating synthetic data. The features sought for these datasets are:

- Data heterogeneity, namely the need to contain structured, partially structured and unstructured data.
- A high level of temporal granularity, namely the need to cover events that occur even in fairly short time intervals.

– A volume of data large enough to make the simulator realistic.

In literature, publicly available datasets with the characteristics sought are very few. One of them is the New Zealand National Minimal dataset, a collection of public and private hospital discharge information, including clinical information [14]. Although it meets the volume requirement, covering the discharges of all patients in New Zealand public hospitals since 1993, it is not considered granular enough for the purposes presented (it does not contain records collected about patients during their stay but only information recorded during their discharge) and not heterogeneous enough (containing only tabular data).

Another dataset that we considered is the ChestX-ray8 dataset, a collection of X-ray images of over 30000 patients [25]. Unfortunately, the radiological reports associated with each image are not included in the dataset, severely limiting its heterogeneity and, therefore, its usefulness. The only dataset deemed large, heterogeneous and granular enough to be used as a reference for the creation of the generator is MIMIC-III [15].

2.1 Synthetic Generators

A commonly used methodology related to machine learning for generating synthetic data in the healthcare field are Generative Adversarial Networks (commonly referred to as GANs). A GAN is a kind of artificial intelligence algorithm based on an adversarial training system, where two competing models (a generator model and a discriminator model) are trained against each other [11]. GANs been used for the development of medGAN [7], a generator that focuses on privacy preserving synthetic health data which, however, is only able to generate binary or at most integer values, severely limiting its usefulness. An evolution of medGAN is provided by healthGAN [26], which is capable of the simulation of real-distributed values, while remaining limited in the heterogeneity of the simulated data. In [19] is proposed SmoothGAN, a new approach for the generation of high quality synthetic data that maintains important relations and factors of the original data; it is not focused on following the time distribution of the original data though and, moreover, it does not seem to have a functioning implementation. GANs have also been used previously in [5] to model events contained in the MIMIC-III dataset, but their focus is again not on the timing of the data, which are apparently omitted from the output of the generator.

The most notable synthetic data generator that we found not based on GANs is provided by Synthea, a tool suite focusing on the generation of synthetic health records that cover the entire lifetime of the patients [24], but that lacks the granularity to be used for performance evaluation purposes.

2.2 Workflow Characterization by Multiformalism

The multiformalism modeling approach aims to facilitate the coordination and integration of models that focus on different aspects or components of a system by employing heterogeneous formalisms. This approach also enables the creation of macro-models by combining specific submodels that use the most appropriate formalism for the problem at hand, while ensuring unity and coherence in the overall model. Different approaches exist regarding the implementation of multiformalism modeling concepts. For instance, some approaches combine multiformalism with multisolution, while others allow the choice of solver to be decoupled from the formalism and bound to general characteristics of the model. Additionally, different multiformalism modeling frameworks vary in terms of the number and types of formalisms that are allowed. Some frameworks employ a fixed set of known formalisms, which simplifies the implementation of proper single or multiple solvers and enables significant optimizations. Other frameworks allow an unlimited number of formalisms and require means to incorporate new formalisms, solution processes, and solvers.

Scientific literature offers a vast family of multiformalism tools, such as SHARPE [23], SMART [8] and DEDS [2] where the type of formalisms is limited. In other tools, the limitation is overcome, as the range of formalisms can be augmented in different ways: see, for example, AToM^3 [16] and Möbius [9], OsMoSys [10], and finally SIMTHESys [1].

3 MIMIC-III

The analysis performed on MIMIC datasets are centered around gathering an understanding of the process behind the generation of its events, focusing on the interaction process between patients and hospital and its duration. MIMIC datasets contain data associated with thousands of patients collected over 10 years, including, in addition to a large amount of structured tabular data collected and placed temporally with minute accuracy, a collection of medical notes written in plain English and issued by the caregivers of the patients. Moreover, it is associated with a database containing a multitude of physiological signals (waveforms) recorded during the ICU stays of the patient, which further increases the heterogeneity of the dataset.

From now on, when talking about MIMIC, unless clearly specified, we will mean to consider the combination of both the original MIMIC-III dataset and the MIMIC-III Waveform Matched Dataset. The objective of this analysis is to find a set of distributions to be used to model the time of acquisition of the events described in MIMIC. To do so, we focus on the process of interaction between the patient and the system, specifically looking at the different events that are registered at each stage of such interaction process. Fig. 1 shows the evolution of the number of patients inside the considered hospital per year and during a day. It is interesting that it does not show the classical self-similarity present in classical sources, like internet traffic, but it has a lot of variability during a year. Once these stages have been identified, we are able to look at the distribution of their duration and model it with some fitting techniques and tools. Finally, the same fitting procedures will be applied to model the inter-time between the registrations of the events contained in MIMIC.

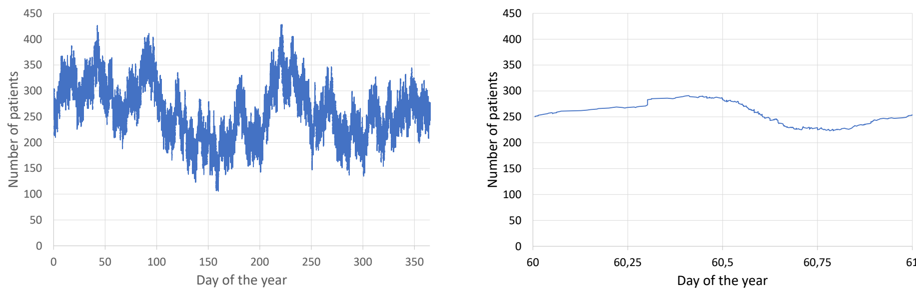


Fig. 1. Evolution of the number of patients during a year and during a day.

Once the distributions of the duration of the interaction stages and of the inter-arrival time of the events for each stage are determined, we use them as the foundation for the synthetic data generator.

3.1 Design Decisions

Due to the sensitive nature of the data that comprises MIMIC, the authors needed to perform a de-identification procedure before the dataset was made available to the public. In particular, for each individual patient, the dates have been shifted into the future by a random offset. Nonetheless, all the time intervals between two entries relative to the same patient are kept intact; therefore, only a handful of time information associated to each timestamp are still valid after the random shift, i.e., the day of the week, the seasonality, and the time of the day. This procedure deeply influenced our methodology: the de-identification procedure made an analysis of the exchange of data between the patients as a whole and the hospital system impossible and, for this reason, we decided to focus on analyzing the duration of the interactions of the patients singularly.

3.2 Stages of Interaction with the Hospital

Fig. 2 shows the possible temporal evolution of patients inside the hospital, focusing on their permanence in the Intensive Care Units. After multiple refinements, the identified stages of the interaction process, represented as time intervals to be modeled, are the following:

1. time spent in an ICU;
2. time interval after the hospital admission and before the start of the first ICU stay;
3. time spent in the hospital between two consecutive ICU stays;
4. time between the end of the last ICU stay and the end of the entire hospital stay;
5. total time in the hospital, which shall be considered as the sum of the times listed above (unless the patient is not admitted in an ICU, in which case the hospital time shall be computed separately);
6. time between the end of a hospital stay and the beginning of the next one.

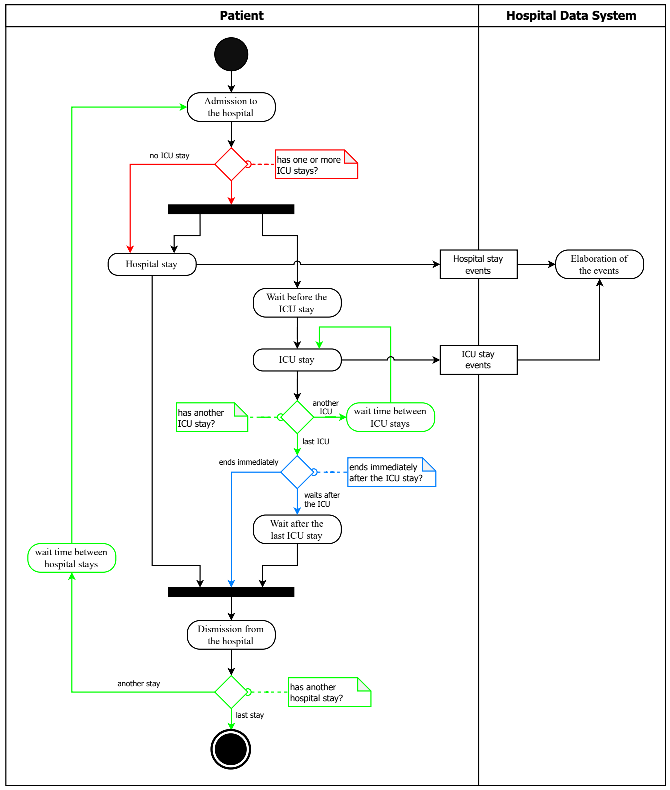


Fig. 2. Activity diagram representing the stages of the interaction process between the patient and the hospital system.

MIMIC considers two main categories of events: those associated with the specific ICU stay of the patient (e.g., automatic measurements of their blood pressure made during their ICU permanence) and those associated with the general hospital stay of the patient (e.g., laboratory results collected from their cultures). The former are generated only during the first stage of the ones listed above, i.e., the time spent in ICU, while the latter are generated through all the other intervals except the last one, during which no event shall be generated, since the patient is not in the premise of the hospital. Each of these categories is then split into multiple kinds of event, like laboratory events or note events, each stored in its own table in the dataset.

3.3 Classification

Before focusing on the distributions that can be fit to model the interaction stages identified in the previous section, we decide to split both the events stored in MIMIC and the interactions into *classes*. This decision is necessary to avoid

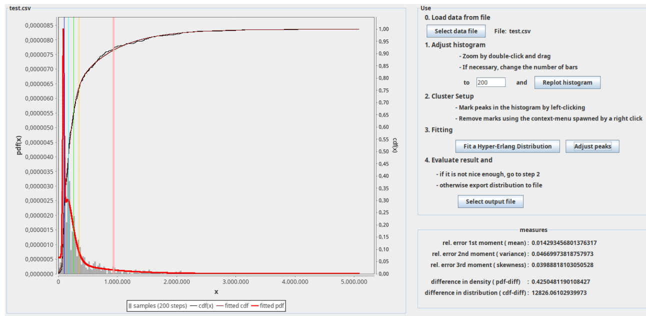


Fig. 3. Interface of HyperStar for distribution fitting.

considering a single distribution to model all the events produced by all the patients during their stays, which would inevitably worsen the fitting results. The chosen classes are based on easily observable features of the data in order to maintain the grouping process simple. Such features, though, shall be relevant to the medical field and distinctive enough to split the dataset in comparable portions. For instance, the link between gender and health conditions is well known [17], and the link between ethnicity and health conditions, although not as common, has also been documented and studied for a long time [22]. Finally, the features chosen for the classification are the following:

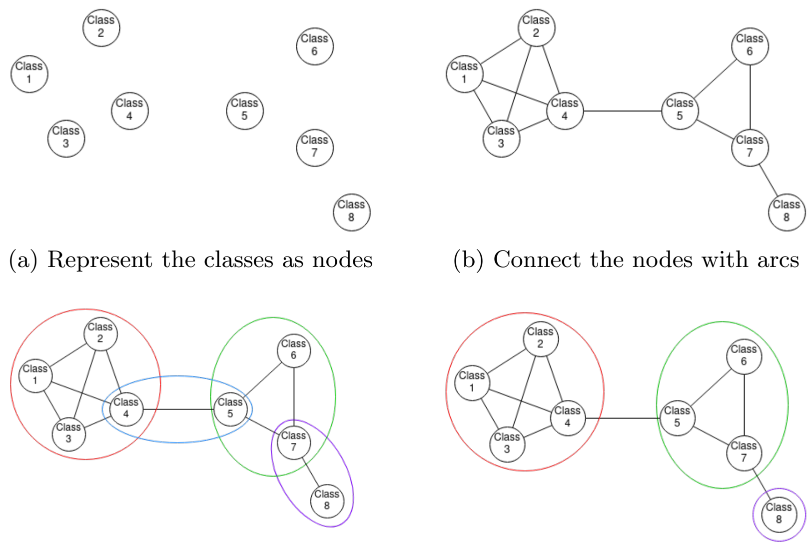
- the gender of the patient (either Male or Female);
- the age of the patient (a categorical attribute with values 0-45, 45-65, 65-75, 75-100 and over 100 years old);
- the day of the week when the hospital stay begins.

The combination of these features leads to a total of 70 classes in which the patients, the admissions and the associated events can be divided.

3.4 Distribution Fitting

After the classification step, we are ready to fit some distributions to the interaction stages previously defined and to the events registered in MIMIC. Using HyperStar [20], a simple tool for distribution fitting, we fit the duration of each interaction stage to Phase-Type distributions [13], as shown in Fig. 3. They are commonly employed for performance evaluation thank to their adaptability to any kind of empirical distribution [4]. In particular, the time intervals are fitted to Hyper-Erlang distributions.

Since HyperStar requires the intervention of the user during the fitting process and we have 70 distributions to consider (one for each class) for each of the interactions identified, the time complexity would grow excessively. For this reason, we opt for bringing together those classes that have a similar empirical distribution. To find similar classes, we leverage the Kolmogorov-Smirnov test,



(a) Represent the classes as nodes (b) Connect the nodes with arcs

(c) Maximal complete sub-graphs (d) Assign shared nodes

Fig. 4. Visualization of the steps performed to make the groups.

commonly used to assess whether two samples come from the same distribution [3]. For each of the interaction stages identified, we perform the test on all possible couples of classes, and cluster together those that exhibit the same distribution. In Fig. 4 is shown a sample of the result, where the nodes represent the different classes, the arcs connect the classes with the same distribution according to the Kolmogorov-Smirnov test, and the circles represent the groups formed. The events, on the other hand, are fitted using exponential distributions, one for each class and for each kind of event registered in MIMIC. The procedure chosen to fit an exponential to the inter-time between the recording of the events of each kind is based on the method of moments, which focuses on achieving the same average of the considered sample. Since most of the kinds of events registered in MIMIC are associated with one or more attributes that provide a clear indication of the time of their creation, this procedure is used for them. Events that require a specific procedure to be fit, due to the way they are structured, are considered separately and specific procedures are employed for each of them. Waveforms are also fitted using exponential distributions.

4 Development of the Generator

The objective features which we aim to obtain during the development of the generator are fine-tunability and adaptability, in line with the requirements of data and workload diversity for the performance evaluation of Big Data systems [12]. **Fine-tunability** is intended as as the possibility of changing the parameters of the identified distributions from those obtained from the analysis performed and the ability to control the synthetic data generated as output. **Adaptability** is intended as the ability to adapt the generator to different analysis procedures and, potentially, different datasets, without requiring excessive and complex modifications. The generator is divided into the following three modules:

– **Configuration module**, which contains the necessary components for reading and managing the outputs of the analysis.
– **Classification module**, containing the components intended to model the classification made during the analysis phase.
– **Generation module**, containing the components needed to generate the synthetic events.

4.1 Configuration Module

In order to allow for an easier customization of the parameters of the generator, the management of the information obtained from the analysis is centralized within a single module. The configuration module consists in two main components: *a)* the **Manager** class, responsible for providing the other components of the generator with the information obtained from the analysis; **b)** the *Configuration dictionary*, which contains the default file paths where to find the outputs of the analysis. To avoid the use of hard-coded strings in the other components of the generator, some enumerations are also introduced to enclose and group the keys of the previously mentioned dictionary. As we will see later on, they are also used to model the interaction stages identified during the analysis and the kinds of events associated with each of them.

4.2 Classification Module

The classification module collects the enumerations intended to model the groups obtained by the classification procedures previously described. To do so, it uses two enumerations, namely `PatientClass` (used to model the classifications based on the gender and the age of the patients) and `AdmissionClass` (which models the classifications based on the weekday at which the hospital stay started).

4.3 Generation Module

The generation module is the main module of the generator, in charge of the generation of the synthetic events from the distributions fitted during the analysis. Its components follows a layered structure. Each layer is comprised of a single class that implements the `EventsGenerator` interface, which defines the `get_events` and `get_waveforms` methods, used to retrieve the events and the waveforms that result from the generation. Each component, when asked to generate the events, creates an instance of the component of the following layer and routes the request to them, up until the `Interaction` layer is reached, which ultimately generates the events.

This structure is chosen to allow the user to finely control the events to be generated by choosing which layer to request the generation of the events to. For example, if the user is interested in the generation of events of multiple patients, regardless of their classification group, s/he can simply use the `Hospital` class, which is able to generate the required events, once provided with the number of patients to consider and the time distance between their first admission to the hospital. If instead the user wants to consider the events of the patients of a specific age group, the `Patient` class may be used. The classification group to consider is decided by the Hospital layer, that chooses the age and the gender of the patient, and by the Patient layer, that chooses the day of the week on which the admission begins. The chosen classification group is passed on to the following layers by the corresponding element of the enumeration discussed in Section 4.2.

The distributions to be used for the generation of the events and the other outputs of the analysis (such as the probability of the patient having a certain age or a certain gender) are requested to the `Manager` class by the components of the generation module. The `Manager` class is meant to be instantiated by the user and provided with the configuration dictionary to retrieve the outputs of the analysis; a default configuration dictionary, previously introduced in the configuration module, may also be used. The created instance can then be provided to the layer intended to use. During the creation of the following layers, the instance of the Manager is passed on to allow the generation of the events.

As previously stated, the only layer that effectively performs the generation of the events is the **Interaction** layer. This layer models the stages of the Interaction process identified previously and, differently to the other layers, contains more than one class. In particular, the Interaction layer contains two classes, both extending the abstract class *Interaction*:

- the `StayInteraction` class, which models all the interaction stages that generate the events associated with the entire hospital stay.
- the `ICUInteraction` class, which models all the interaction stages that generate the events associated with only the ICU stay.

To generate the synthetic events, the two classes request to the instance of the `Manager` class (either passed on by the previous layers or provided by the user) the parameters of the exponential distributions fitted during the analysis. The duration of the interaction is provided by the Admission layer, which requests the necessary parameters to the Manager to generate it according to the phase-type distributions fitted during the analysis; during this exchange, information about the specific interaction stage considered is communicated through the elements of the enumerations introduced in the configuration module. The events are represented in the generator by the `Event` class, regardless of their category or type. The waveforms are represented by an extension of such class.

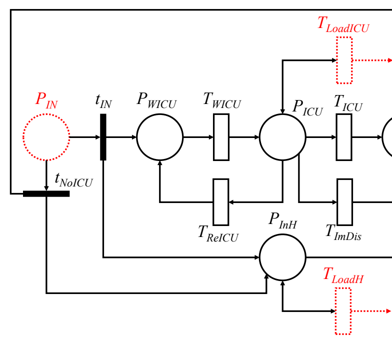


Fig. 5. The GSPN model for the activity diagram shown in Fig. 2.

5 Experimental Evaluation

To show the potentiality of the approach, we propose a multiformalism model which uses the GSPN shown in Fig. 5 to reproduce the patient behavior described in Fig. 2. In particular, new patients enters the hospital in place P_{IN}. Immediate transitions t_{IN} and t_{NoICU} determines whether the patient will need at least a visit to the ICU or if she will only require regular hospitalization. Waiting time to enter or re-enter the ICU is modeled by place P_{WICU} and timed transition T_{WICU}. Permanence in ICU is modeled by place P_{ICU}. In this case three possible events can occur: a patient can conclude her permanence in the ICU, and be immediately dismissed (transition T_{ImDis}), she can be hospitalized for a subsequent period (T_{ICU}), or re-enter the ICU after another period (T_{ReICU}). Normal hospitalization is modeled by place P_{WDis} and timed transition T_{WDis}, while the dismiss is modeled by place P_{Dis} and immediate transition t_{Dis}. Finally places P_{InH} and P_{OUT} account respectively for the total number of patients in the hospital (both ICU and non-ICU), and the patients that are leaving the hospital. Requests produced by patient in the ICU and in the hospital, are modeled by infinite server timed transitions $T_{LoadICU}$ and T_{LoadH}. The firing rate of both transitions is controlled by the marking of the place corresponding to the number of patients either in ICU (P_{ICU}), or inside the hospital (P_{InH}). The type of request is modeled by the color of the token being produced during firing of the corresponding transition: either (H–$jobs$) or (ICU–$Jobs$).

The information system of the hospital is modeled with the Queing Network model shown in Fig. 6, where the GSPN component shown if Fig. 5 is represented with a rounded box. There are two classes of jobs, representing respectively the requests incoming from normal patients (H–$jobs$) and from the ones in ICU (ICU–$Jobs$). Patients arrives to the hospital according to source λ_{Pat}, produces jobs with the GSPN submodel, and leaves the system through immediate transition T_{END} and sink σ_{Pat}. In this simplified scenario, the system is modeled by a C-server queue, with finite capacity K, and drop policy. In this context, all distributions are considered exponential, and queues works in FCFS order.

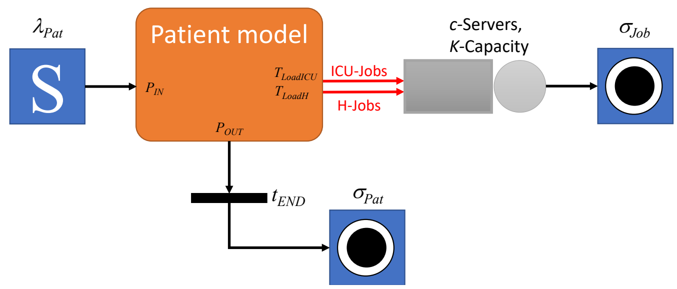


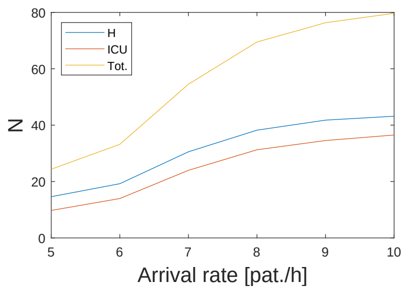
Fig. 6. The multiformalism model of the considered scenario.

The considered parameters are reported in Table 1. To avoid privacy issues, data used are not the exact ones computed by the fitting procedure and the generator previously described, but sufficiently similar to make results meaningful.

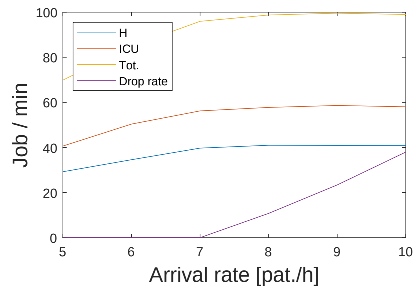
Parameter	Value	Parameter	Value	Parameter	Value
λ	$5 \dots 10$ p/h	C	40	K	100
S_{ICU}	20 s	S_H	30 s	t_{IN}	9
$\lambda_{LoadICU}$	30 r/h	λ_{LoadH}	6 r/h	t_{NoICU}	1
T_{WICU}	4 h	T_{ReICU}	120 h	T_{ImDis}	72 h
T_{ICU}	24 h	T_{WDis}	48 h		

Table 1. Model parameters.

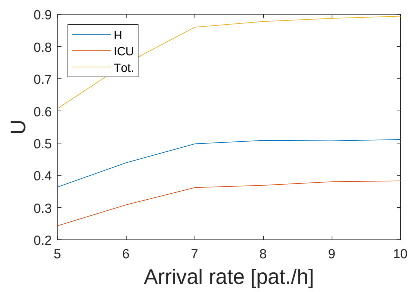
The performance analysis and the design of the model is performed using *JMT*, a suite of tools for the performance evaluation of computer systems [21]. With the generated events, we are able to evaluate the system's performances, showing through an analysis of various performance indices (such as throughput, response time, and utilization), considering an increase in the arrival rate. Fig. 7a shows the evolution of the arrival rate of requests: the flex at around $\lambda = 7$ patients per hour shows the possibility of having reached the maximum capacity of the system. This is confirmed by Fig. 7b, where throughput and drop rates are shown. As expected, with more than $\lambda > 7$ patients per hour, requests start being dropped, showing that the considered number of servers is not sufficient to handle this peak of requests. It is interesting to see that although the system is experiencing drops, it still does not reach 100% utilization, as shown in Fig. 7c: this is a consequence of the fact that the losses are due to the high variability of the input traffic: most of them occur during bursts, making then the system work at a very low utilization during normal operation times.



(a) Number of requests



(b) Throughput and drop rate



(c) Utilization

Fig. 7. The evolution of number of requests, throughput and drop rate, and utilization of the servers as function of the arrival rate λ.

6 Conclusions

In this work, we have developed a synthetic data generator to be used for the evaluation through modeling techniques of a Big Data System. As a reference for the creation of the generator, we relied on MIMIC, a publicly available dataset containing the measurements and observations made on patients of a hospital during a multiple years period and recorded by the data system of the hospital in question. With the aim of obtaining additional information on the temporal distribution of such events, we first analyzed the interaction process between the patients and the hospital system, focusing on the time periods during which the events described in MIMIC were recorded in the system. These time periods were then modeled through appropriate distributions, taking advantage of distribution fitting techniques and tools; these same techniques were then used to model the inter-times of each kind of event. The results obtained, such as the parameters of the fitted distributions and the structure identified for the interaction process between the patients and the hospital system, were used as the foundation for the generator's architectural choices. To allow a granular control over the output of the generator, a layered architecture was adopted during its development. Moreover, throughout development, the ability to customize the generator was foregrounded, to make it easy to adapt it to different analysis procedures. Finally, to test the possibilities just described, two experiments were carried out, focused on evaluating the performance of a simple model of a Big Data system in two different scenarios. They resulted in a success, showing how easy it is to tune the output of the generator and to change its components.

Recently, MIMIC saw the release of a newer version, named MIMIC-IV, which has a slightly different structure and considers longer time frames. To date, it does not yet contain the waveforms records, but their introduction has been initiated. A possible future work might involve using data from MIMIC-IV as a reference for a new version of the generator.

Acknowledgements This work has been partially supported by the Health Big Data Project (CCR-2018-23669122), funded by the Italian Ministry of Economy and Finance and coordinated by the Italian Ministry of Health and the network Alleanza Contro il Cancro. Additionally, we are grateful to Letizia Tanca and Giuseppe Serazzi for their advice during the definition of this work and the support in the revision of this paper.

References

1. Barbierato, E., Bobbio, A., Gribaudo, M., Iacono, M.: Multiformalism to support software rejuvenation modeling. In: 2012 IEEE 23rd International Symposium on Software Reliability Engineering Workshops. pp. 271–276. IEEE (2012)
2. Bause, F., Buchholz, P., Kemper, P.: A toolbox for functional and quantitative analysis of deds. In: Computer Performance Evaluation: Modelling Techniques and Tools 10th International Conference, Tools' 98 Palma de Mallorca, Spain September 14–18, 1998 Proceedings 10. pp. 356–359. Springer (1998)
3. Berger, V.W., Zhou, Y.: Kolmogorov–Smirnov Tests. In: Encyclopedia of Statistics in Behavioral Science. John Wiley & Sons, Ltd (2005)
4. Bladt, M.: A Review on Phase-type Distributions and their Use in Risk Theory. ASTIN Bulletin: The Journal of the IAA **35**(1), 145–161 (May 2005)
5. Chin-Cheong, K., Sutter, T., Vogt, J.E.: Generation of heterogeneous synthetic electronic health records using GANs. In: Workshop on Machine Learning for Health at the 33rd Conference on Neural Information Processing Systems (2019)
6. Choi, E., Biswal, S., Malin, B., Duke, J., Stewart, W.F., Sun, J.: Generating multi-label discrete patient records using generative adversarial networks. In: Machine learning for healthcare conference. pp. 286–305. PMLR (2017)
7. Choi, E., Biswal, S., Malin, B., Duke, J., Stewart, W.F., Sun, J.: Generating Multi-label Discrete Patient Records using Generative Adversarial Networks. In: Proceedings of the 2nd Machine Learning for Healthcare Conference. pp. 286–305 (2017)
8. Ciardo, G., Jones, III, R.L., Miner, A.S., Siminiceanu, R.I.: Logic and stochastic modeling with SMART. Perform. Eval. **63**, 578–608 (June 2006)
9. Clark, G., Courtney, T., Daly, D., Deavours, D., Derisavi, S., Doyle, J., Sanders, W., Webster, P.: The mobius modeling tool. In: Proceedings 9th International Workshop on Petri Nets and Performance Models. pp. 241–250 (2001)
10. Franceschinis, F., Gribaudo, M., Iacono, M., Mazzocca, N., Vittorini, V.: Towards an object based multi-formalism multi-solution modeling approach. In: Proceedings

of the Second International Workshop on Modelling of Objects, Components, and Agents. pp. 47–66 (2002)
11. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial networks. Communications of the ACM **63**(11), 139–144 (2020)
12. Han, R., Lu, X., Xu, J.: On Big Data Benchmarking. In: Big Data Benchmarks, Performance Optimization, and Emerging Hardware. pp. 3–18 (2014)
13. Harchol-Balter, M.: Real-world workloads: High variability and heavy tails. Performance modeling and design of computer systems: Queueing theory in action pp. 347–348 (2013)
14. Health, M.o.: National Minimum Dataset (hospital events) (2012)
15. Johnson, A.E., Pollard, T.J., Shen, L., Lehman, L.w.H., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Anthony Celi, L., Mark, R.G.: Mimic-iii, a freely accessible critical care database. Scientific data **3**(1), 1–9 (2016)
16. Lara, J.d., Vangheluwe, H.: Atom 3: A tool for multi-formalism and meta-modelling. In: Fundamental Approaches to Software Engineering: 5th International Conference. pp. 174–188 (2002)
17. Legato, M.J., Bilezikian, J.P.: Principles of Gender-specific Medicine. Gulf Professional Publishing (2004)
18. Moody, B., Moody, G., Villarroel, M., Clifford, G., Silva III, I.: Mimic-iii waveform database matched subset. PhysioNet (2017)
19. Rashidian, S., Wang, F., Moffitt, R., Garcia, V., Dutt, A., Chang, W., Pandya, V., Hajagos, J., Saltz, M., Saltz, J.: SMOOTH-GAN: Towards Sharp and Smooth Synthetic EHR Data Generation. In: Artificial Intelligence in Medicine. pp. 37–48 (2020)
20. Reinecke, P., Krauß, T., Wolter, K.: Phase-type fitting using hyperstar. In: Computer Performance Engineering: 10th European Workshop, EPEW 2013, Venice, Italy, September 16-17, 2013. Proceedings 10. pp. 164–175. Springer (2013)
21. Serazzi, G., Casale, G., Bertoli, M.: Java Modelling Tools: an Open Source Suite for Queueing Network Modelling and Workload Analysis. In: Third International Conference on the Quantitative Evaluation of Systems. pp. 119–120 (2006)
22. Tang, H.: Confronting ethnicity-specific disease risk. Nature genetics **38**(1), 13–15 (2006)
23. Trivedi, K.S.: SHARPE 2002: Symbolic Hierarchical Automated Reliability and Performance Evaluator. In: Proceedings of the 2002 International Conference on Dependable Systems and Networks. p. 544 (2002)
24. Walonoski, J., Kramer, M., Nichols, J., Quina, A., Moesel, C., Hall, D., Duffett, C., Dube, K., Gallagher, T., McLachlan, S.: Synthea: An approach, method, and software mechanism for generating synthetic patients and the synthetic electronic health care record. Journal of the American Medical Informatics Association **25**(3), 230–238 (Mar 2018)
25. Wang, X., Peng, Y., Lu, L., Lu, Z., Bagheri, M., Summers, R.M.: Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2097–2106 (2017)
26. Yale, A., Dash, S., Dutta, R., Guyon, I., Pavao, A., Bennett, K.P.: Generation and evaluation of privacy preserving synthetic health data. Neurocomputing **416**, 244–255 (Nov 2020)