Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 00 (2018) 000–000

Procedia Computer Science

www.elsevier.com/locate/procedia

Healthcare Interoperability and Pervasive Intelligent Systems (HiPIS)

Efficient Enriching of Synthesized Relational Patient Data with Time Series Data

Simon Schiff[a,*], Marcel Gehrke[a], Ralf Möller[a]

[a]*Institute of Information Systems, University of Lübeck, Ratzeburger Allee 160, 23562 Lübeck, Germany*

Abstract

Analysing data from electronic healthcare records allows for supporting decision making and thereby can improve healthcare. However, obtaining sufficient healthcare data required for machine learning analysis is challenging due to, e.g, privacy aspects of medical data. For machine learning tasks, carefully prepared synthesized medical records can be as good as real records, which is shown in [17]. Existing tools for medical data provision generate either relational records or streams of measurements over time, but not an appropiate combination of both. In this paper, we contribute an approach to enriching synthesized relational data with time series (longitudinal data) of real patients. We use Synthea to synthesize relational data and enrich the records with time series from the anonymized MIMIC III database. In our data integration scenario, we need to find the best match from the relational data to the time series data to obtain a sufficient amount of medical data for machine learning analyses. Our experiments show that we can enrich huge amounts of relational data with real time series data. However, without any processing optimizations, the runtime does not easily scale with the number of synthesized relational records. With several optimizations and using a distributed execution engine, such as Apache Spark SQL, we can efficiently enrich synthesized relational data with time series data.

Keywords: data enrichment, data integration, time series, relational data, medical data

1. Introduction

Electronic health records (EHRs) contain relational and temporal patient data. More specifically, EHRs contain relations between patients, medications, procedures, diagnoses, and streams of measurements over time. Analysing data from EHRs can help improving the overall health of patients and reduce healthcare costs [2]. The data can be used to generate prediction models [10, 11], for example to detect heart failures [18], and even unstructured treatment notes can be systemically analysed [14]. Unfortunately, machine learning techniques require collections of data which are not easily accessible. Further, exploitation of existing health-related data is very restricted due to important privacy concerns [5]. Therefore, in most cases the use of real datasets is not feasible as they are, e.g., only accessible inside a clinical organisation, if even, and in addition, scaling experiments can hardly be conducted on real data. Nonetheless,

* Corresponding author. Tel.: +49-451-3101-5736
E-mail address: schiff@ifis.uni-luebeck.de

a few anonymized EHR databases for research purposes exist, such as MIMIC III [8] and eICU [7]. Those databases are accessible to researchers, and the patients are fully anonymized, even though Emam et. al were able to re-identify patients [6], which is forbidden by the data use agreement. Researchers can test new algorithms with those anonymized datasets, reproduce test results from others, and compare algorithms. Additionally, having a limited set of data leads to known problems in machine learning, such as (i) always reusing the same dataset for machine learning can lead to overfitting, (ii) missing data to test for corner cases can lead to poor prediction models, and (iii) there might not be enough data to train the prediction model well [11].

A solution for obtaining data without such limitations is to synthesize realistic, but not real, patients. Hence, one can also make the data publicly available. Synthea [16] is a tool that synthesizes EHRs and models the whole lifespan of patients including relations such as drug usage, allergies, and observations of low frequency e.g. the body weight. A common EHR does not only consist of relations like drug usage or observations of low frequency, the largest part of an EHR consists of time series with measurements over time within the frequency of milliseconds. About 98% of the data in the MIMIC III database consists of measurements with high frequencies, having a very huge impact on the size of an EHR, which Synthea does not synthesize. Very often, these kinds of measurements are electrocardiographies (ECGs). FECGSYN [3] is a toolbox to generate ECGs based on several parameters. However, those parameters only have an impact on the ECG curves, but specific diseases or conditions are not specifiable. Thus, the generated ECG data does not match to the relational data of patients. Further, one might be interested in influences of multiple diseases on ECGs. Hence, we propose to enrich synthesized relational data with time series. We match generated relational data by Synthea with ECGs from MIMIC III. For the match from the two sources, we use six features, which suffices for our scenario, as the use of more features results in a more precise match, but decreases the number of matched patients. Additionally, we can use FECGSYN, not matched time series, and time series with an offset to prevent overfitting. With our approach, we generate patients on demand, where time series are consistent with the patient's medical history.

Related approaches to synthesize relational medical data without time series exist and synthesized medical data for predications are used. Murray et al. describe a way how to extract features from real databases for the creation of simulated patient datasets [12]. Cunningham et al. present a tool to simulate realistic enough patient datasets for the development and testing of medical systems, but they note that simulated datasets are not realistic in depth nor have a pedagogical value [4]. Johnson et al. demonstrate the usefulness of synthesized data to evaluate and compare algorithms in machine learning [9]. Watkins et al. use semi-synthetic data for disease surveillance with a hidden Markov model [17]. All these approaches have in common, that either relational or time series data is generated, but lack of a combination of both. With our solution, we present an approach how to obtain relational patient data with appropriate time series data.

Our paper has the following structure: We begin by presenting EHR data and dictionaries we use, followed by, how we solve our data integration problem by computing similarities between patients to enrich relational patient data with time series data. Finally, we evaluate our implementation and conclude how we could use the enriched patient data.

2. Medical Data

Training a prediction model requires a sufficient amount of data and obtaining medical data is not easy. Common ways to obtain medical data are anonymized databases which can be bought or accessed freely, synthesized databases, and tools to generate synthesized databases. In our experiments, we use MIMIC III, a freely accessible medical database containing both relational and time series data, and Synthea to synthesize relational data at scale. In the following, we present how medical data is classified using data dictionaries, MIMIC III, and Synthea.

2.1. Data Dictionaries

A patient can have several diseases diagnosed by a doctor, and based on diagnoses, procedures are performed. Diagnoses and procedures are typically described in natural language. Databases containing such descriptions only, without any classification, have a lot of disadvantages. Thousands of different diagnoses and procedures exist and there are several ways to describe each of them. Searching or comparing patients according to their diseases becomes nontrivial. Thus, a vocabulary for the classification of diagnoses and procedures is very useful and indeed several

proposals for such a vocabulary exist. One of them is the international statistical classification of diseases and related health problems (ICD) maintained by the world health organization. ICD is currently the most widely used statistical classification system in the world. Several revisions and modifications of ICD exists. ICD-9-CM is a clinical modification of the ninth revision created by the US National Center for Health Statistics, which is used in MIMIC III. Synthea uses the Systematized Nomenclature of Medicine SNOMED-CT vocabulary, where CT stands for clinical term.

Other vocabularies can be used to classify drugs. RxNorm provides normalized names and concept codes (RxCUI) for clinical drugs and drug delivery services. The norm only includes drugs, that are approved in the US. Another norm is the national drug code (NDC). A NDC consists of three segments. The first segment contains the labeler, the second one the product, and the last one the package size.

To migrate from one vocabulary to another, a mapping can be used, but some vocabularies are more fine grained than others. Therefore, an exact one to one mapping between different vocabularies is not always possible. To map ICD-9-CM codes to SNOMED-CT, a mapping from the national library of medicine exists, mapping 95% ICD-9-CM diagnostic codes to SNOMED-CT, which is divided into one-to-one (2/3) and one-to-many (1/3) mappings.

2.2. MIMIC III

Only a few anonymized databases containing EHRs are available, such as MIMIC III, which contains medical data from the Beth Israel Deaconess Medical Center in Boston. The data consists of information about different intensive care units (ICUs). Furthermore, the database contains tests, orders, billings, demographics, notes, and reports. MIMIC III is the only freely accessible critical care database of its kind [8], and there is a reason for such rarity.

The MIMIC III database is not a simple one-to-one copy of an existing clinical database. However, the database is integrated from three sources (i) archives from critical care information systems, (ii) hospital EHR databases, and (iii) Social Security Administration Death Master File. Procedures, diagnostics, and medications are standardized to dictionaries such as LOINC, ICD-9-CM, or RxNorm, including a relation between the codes and their definition, which is a labor-intensive process [1]. Patients are automatically de-identified in accordance with the Health Insurance Portability and Accountability Act [5]. The development of the database is ongoing, for example during datathons organised by MIT critical data. The structure of the data is hardly modified, but might not represent a real database of a clinic anymore. The whole database is available in a repository after filling out a data use agreement.

2.3. Synthesized Relational Patient Data using Synthea

Synthea is a tool to synthesize EHRs on demand. In Synthea the ten most frequent reasons for primary care encounters and the ten chronic conditions with the highest morbidity in the United States are modelled [16]. Only publicly available information, such as health statistics, are in use by Synthea to synthesize data. Hence, the data produced by Synthea is free of legal, privacy, security, and intellectual property restrictions.

Synthea is divided into modules. Each module contains a transition graph, which is used to model a disease or a treatment. The development and updating of modules heavily depends of the knowledge of experts from a medical domain that are usually not able to write a program in Java. Hence, Synthea provides a web interface for medical experts to develop modules. These modules are used during the data generation process to model the whole lifespan of a patient. Thus, a patient can have several diseases and treatments during his lifetime and some transition graphs will not terminate until the death of a patient. Synthea uses standardized dictionaries, e.g., procedure and diagnostic codes from SNOMED-CT and medication codes from RxNorm. Further, the output formats FHIR, CSV, CCDA, and human readable text are supported.

2.4. Data Integration

Datasets can have different units, structures, schemes, and semantics and the databases containing them can be different due to their implementation and query language. Thus, querying different datasets from various sources is a challenging task, even in the same context such as, for example, clinical databases. These differences are caused, e.g., by several different medical devices, ICUs, billing systems, units, and terminologies. One solution is data integration, which is the creation of a unified global view.

Formally, two different databases σ and σ' exists and the goal is to create a third database τ containing a global view. Data can be physically integrated into τ or queries over τ are automatically rewritten and sent to σ and σ'. Therefore, τ makes accessing and querying data easier, as τ hides the underlying data distribution and diversity [15].

3. Enriching Synthesized Patient Data

To improve the care of patients by analysing medical data, we need a sufficient amount of EHRs containing time series with measurements corresponding to the relational data. EHRs synthesized by Synthea are free of legal restrictions, but lack of time series. MIMIC III cannot be used as freely as data from Synthea, but contains time series data. Therefore, we combine synthesized relational data with time series from MIMIC III. To enrich synthesized patients, we search for similar patients from MIMIC III, and then only keep the synthesized patients and time series data of MIMIC III patients. Our approach allows for obtaining relational patient data of variable size containing time series, where patient data roughly fit to the time series data. Having the MIMIC III database and the ability to generate EHRs using Synthea, we perform the following steps:

(i) extract common features from MIMIC III and Synthea,

(ii) compute patient similarity based on the features, and

(iii) enrich relational patient data.

3.1. Features

We extract the following six patient features from MIMIC III and Synthea, which are not always explicitly given in the data, but are useful for comparing patients:

(i) procedures and diagnostics (disease),

(ii) gender,

(iii) age,

(iv) whether the patient is dead,

(v) ethnic, and

(vi) medications.

In MIMIC III, we calculate the age of a patient for each hospital stay, namely as a difference between his birth date and the time of hospital stay. In Synthea, we obtain the patient's age as a difference between his birth date and the time the patient was synthesized. Ethnicities are mapped manually as Synthea patients can have 30 different and MIMIC III 41 ethnicities. Synthea patient procedures and diagnostics are classified using SNOMED-CID and medications using RxNorm, but MIMIC III uses ICD-9-CM and NDC dictionaries for classification. As we are not able to manually map 2032 different procedures, 6984 diagnostics, and 4204 medications from MIMIC III to Synthea, we use mappings from the national library of medicine to map ICD-9-CM to SNOMED-CID and NDC to RxNorm.

Using data integration, we create a new database *match* with the schema from Figure 1 to access extracted features. The view `MIMIC_META` and `SYNTHEA_META` contain patients gender, ethnicity, age, and whether the patient is dead Additionally, `MIMIC` and `SYNTHEA` contain every SNOMED-CID and RxNorm code associated to a patient.

3.2. Similarity

To obtain `MATCH_1` as depicted in Figure 1, we join `MIMIC_META` with `MIMIC` and `SYNTHEA_META` with `SYNTHEA` using `PATIENT_ID` and then join both results by `SNOMED_CID`. We iteratively improve similarity by filtering out pairs, which are not similar, except for the age, where a difference of five years suffices. Every intermediate result is stored in a relation ranging from `MATCH_1` where pair of patients are only similar by `SNOMED_CID` to `MATCH_6` where pair of

```
MIMIC_META (PATIENT_ID, MIMIC_GENDER, MIMIC_ETHNIC, MIMIC_DEAD, MIMIC_AGE)
MIMIC (PATIENT_ID, SNOMED_CID, RX_NORM)
SYNTHEA_META (PATIENT_ID, SYNTHEA_GENDER, SYNTHEA_ETHNIC, SYNTHEA_DEAD, SYNTHEA_AGE)
SYNTHEA (PATIENT_ID, SNOMED_CID, RX_NORM)
MATCH_1 (MIMIC_ID, SYNTHEA_ID, MIMIC_GENDER, SYNTHEA_GENDER, MIMIC_ETHNIC, SYNTHEA_ETHNIC,
MIMIC_DEAD, SYNTHEA_DEAD, MIMIC_AGE, SYNTHEA_AGE, MIMIC_RX_NORM, SYNTHEA_RX_NORM,
ARRAY_COMMON_SNOMED_CID)
```

Fig. 1: Schema of database *match*

patients are similar by all six features. We unify all `MATCH_n` relations and for each pair we keep only the best match. Additionally, we count the common intersection of SNOMED-CID codes. We rank patients based on their similarity using a score:

$$\frac{\#\text{Common SNOMED-CID Codes}}{\#\text{SNOMED-CID of Synthea Patient}} \cdot \frac{\#\text{Common SNOMED-CID Codes}}{\#\text{SNOMED-CID of MIMIC III Patient}} \cdot \frac{\#\text{similar features}}{6} \tag{1}$$

which can be used to balance the number of matched patients and the number of features they have in common, as patients with a high similarity are rare. Thus, the score ensures that the patients have a similar medical history. A pair of patients having a high score are more likely similar than patients with a low score, as the score is higher the more features and SNOMED-CID codes are common. The score ranges between zero as pair of patients can have zero similar features and one as patients cannot have more than six similar features.

3.3. Enrichment

We keep pairs of patients, who have a score above a given threshold. As we match the medical history of patients, the linked time series approximately fit the enriched synthesized patient. Almost every MIMIC III patient has time series data and a similar synthetic patient, where time series approximately fit to the synthetic patient. Finally, we keep only synthetic patients data and time series data. Additionally, we can randomly enrich synthetic patients, with time series synthesized by FECGSYN, obtained from MIMIC III which are not used, or matched but slightly modified time series to prevent overfitting.

4. Evaluation

We expect different properties from our result of enriched synthesized relational patient data with time series data:

(i) the result should contain a large proportion of time series and synthesized relational patient data, to have a variety of patients in our result,

(ii) each pair of patients should be matched by using as much features as possible to obtain synthesized patients with time series matching their medical history, and

(iii) the processing should only take a reasonable amount of time to enrich a sufficient number of synthetic patients.

4.1. Matching

Unfortunately, we are not able to directly use all MIMIC III patients for the matching. Roughly 7000 MIMIC III patients do not have any corresponding medication, procedure, or diagnostic codes and in our experience, Synthea only synthesizes about 300 different SNOMED-CIDs, as they are based on the top ten primary care encounters and

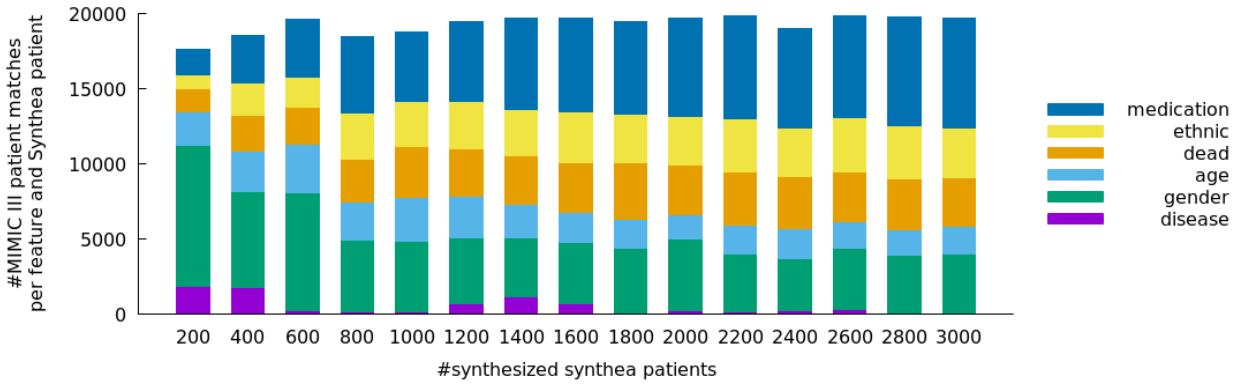


Fig. 2: Number of MIMIC III patient matches per feature and Synthea patient

chronic conditions with the highest morbidity. Therefore, we can only match about 19000 of the overall 46520 MIMIC III patients. For these 19000 patients, we find for nearly all synthesized patients a match. Further, we can use the remaining 27000 patients time series data to prevent overfitting. Therefore, we can match all time series data to synthesized relational patient data and also find for each synthesized patient a proper match.

Figure 2 shows the number of distinct matched MIMIC III patients per feature for 200 to 3000 synthesized patients. For each feature, the match also has to hold for the less restrictive features, as we compute the similarity between pairs of patients iteratively. Therefore, pairs of patients who are similar by age, are also similar with respect to gender and disease. For example, with 2000 synthesized patients, we find about 3300 MIMIC III patients, which match some synthesized patient by dead, age, gender, and disease, as described in subsection 3.2. Even more, we can match about 6500 MIMIC III patients, which are similar to at least one synthesized patient by all six features.

The number of matchable MIMIC III patients is fixed by 19000 whereas the number of synthetic patients varies. With an increasing number of synthesized patients, also the number of MIMIC III patients with a high similarity score (Equation 1) increases. If we synthesize 200 patients, then around 1600 MIMIC III patients have a similar synthetic patient in all six features. By synthesizing 3000 patients that number increases to 7200 MIMIC III patients.

Figure 3 depicts the number of distinct matched synthetic patients per feature. We obtain more synthetic patients than specified, because some synthetic patients are dead. For example, if we use Synthea to synthesize 3000 patients, then Synthea synthesizes 3352 patients (3000 alive and 352 dead). We find for 1400 synthetic patients of the 3000 a similar MIMIC III patient, who is similar in all six features. Each synthetic patient is at least similar to a MIMIC III patient by disease and gender. Regardless of how many patients we synthesize, we can match each synthesized patient with at least one similar MIMIC III patient and the number of features per synthetic patient is similar distributed, as

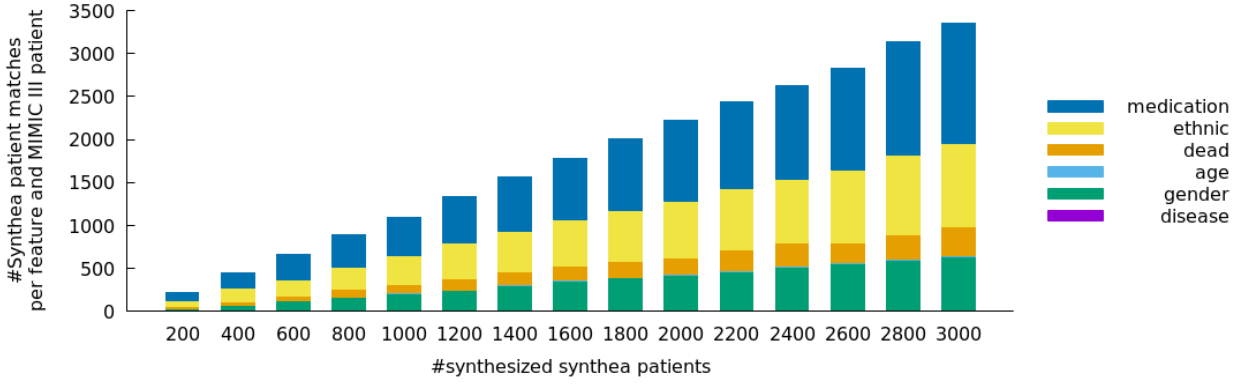
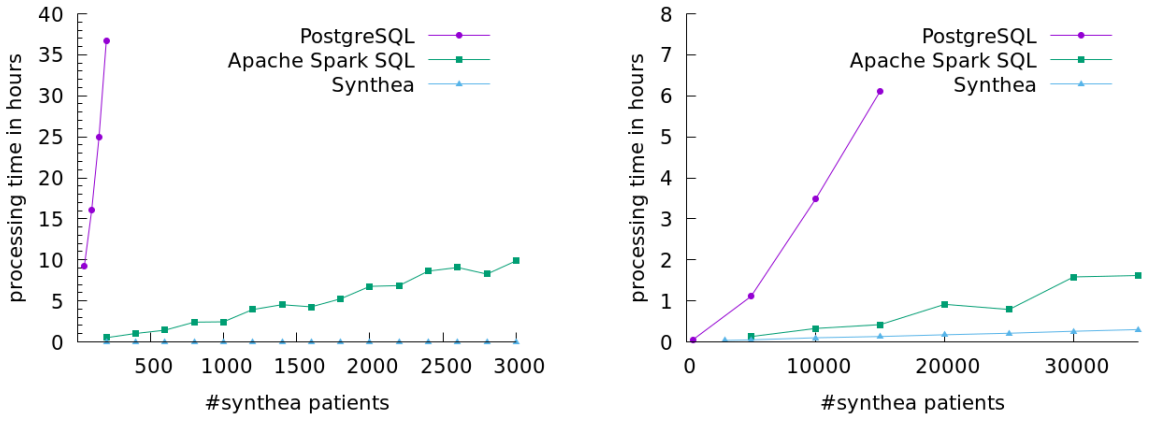


Fig. 3: Number of Synthea patient matches per feature and MIMIC III patient



(a) Processing times without optimizations

(b) Processing times with optimizations

Fig. 4: Processing time to enrich n synthesized patients

the number of MIMIC III patients is fixed. Finally, we select the best MIMIC III patients to enrich the synthesized patients, using our similarity score, Equation 1.

We achieve our goals i) and ii), namely that we can match each patient and thereby use all possible MIMIC III patients as well as having high similarity scores for these matches. We still need to show that we can find the matches in reasonable time.

4.2. Processing

The processing time strongly depends on the number of patients we enrich. Figure 4a shows the time is needed to enrich up to 3000 patients without any optimizations. We split the time it takes to generate the patients using Synthea and to enrich them using PostgreSQL or Apache Spark SQL, as we described in section 3. It roughly takes 4 minutes with Synthea to synthesize up to 3000 patients. Enriching a few patients, using PostgreSQL, already takes hours and therefore is not practical. With Apache Spark SQL, a distributed execution engine, installed on a 4 node cluster, were each node has 8 Intel Xeon E5-2620 v3 processor cores and 21 GB-Ram, we speed up the enriching. Using Spark SQL drastically reduces the time to enrich patients, but unfortunately still needs around 10 hours for 3000 patients, which is not reasonable either.

Thus, we deploy some optimizations: (i) reducing mapping files from around 8.5 million rows to 131, as the most of the contained codes are not in use by MIMIC III or Synthea, (ii) preprocessing `MIMIC_META` and `MIMIC`, and (iii) computing six different relations containing each pair of patients who are similar by one feature and then using these to iteratively improve the similarity, which avoids large cross products. In Figure 4b, the time to synthesize and enrich 0 to 35000 patients, using our optimization ideas, are depicted. The processing times are orders of magnitude faster. We can synthesize and enrich up to 5000 patients in a reasonable amount of time, using PostgreSQL. Synthea needs around 18 minutes to synthesize 35000 patients and we are able to enrich them in under 2 hours, using Spark SQL. Using our optimizations and a distributed execution engine, such as Spark SQL, allows us to efficiently enrich synthesized relational patient data with time series data on demand.

5. Summary

In this paper, we present an approach for enriching synthesized relational patient data with appropriate time series data. Synthesizing 2000 patients already leads to patient matches, where patients have at least the same disease and gender, and moreover we find for 35% of the MIMIC III matchable patients a synthetic patient, who is similar by all six features. By increasing the number of synthetic patients, we can match even more MIMIC III patients on all six features. We use PostgreSQL to store and process datasets synthesized by Synthea and from the MIMIC III dataset.

Even with the use of optimizations, such as pre-processing data, the use of indexes, and data cleansing, the processing times were initially found to be too high. Thus, we use Apache Spark SQL, a distributed execution engine, to speed up processing. The use of Apache Spark SQL together with the optimizations, we are able to efficiently generate large amounts of medical EHRs on demand.

In the future, we plan to use the data to test and evaluate STARQL (Streaming and Temporal Ontology Access with a Reasoning Based Query Language), which requires both relational and time series data [13].

Acknowledgements

This research originated from the Big Data project being part of Joint Lab 1, funded by Cisco Systems Germany, at the centre COPICOH, University of Lübeck

References

[1] Abhyankar, S., Demner-Fushman, D., McDonald, C.J., 2012. Standardizing clinical laboratory data for secondary use. Journal of biomedical informatics 45, 642–650.

[2] Bar-Dayan, Y., Saed, H., Boaz, M., Misch, Y., Shahar, T., Husiascky, I., Blumenfeld, O., 2013. Using electronic health records to save money. Journal of the American Medical Informatics Association 20, e17–e20.

[3] Behar, J., Andreotti, F., Zaunseder, S., Li, Q., Oster, J., Clifford, G.D., 2014. An ECG simulator for generating maternal-foetal activity mixtures on abdominal ECG recordings. Physiological measurement 35, 1537.

[4] Cunningham, J.A., Ainsworth, J.D., 2015. Simulating Realistic Enough Patient Records, in: MIE, pp. 35–39.

[5] Dernoncourt, F., Lee, J.Y., Uzuner, O., Szolovits, P., 2017. De-identification of patient notes with recurrent neural networks. Journal of the American Medical Informatics Association 24, 596–606.

[6] El Emam, K., Jonker, E., Arbuckle, L., Malin, B., 2011. A Systematic Review of Re-Identification Attacks on Health Data. PloS one 6, e28071.

[7] Goldberger AL, A.L., Glass L, H.J., Ivanov PCh, M.R., Mietus JE, M.G., Peng C-K, S.H., 2000. PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals. Circulation 101.

[8] Johnson, A.E., Pollard, T.J., Shen, L., Li-wei, H.L., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Celi, L.A., Mark, R.G., 2016. MIMIC-III, a freely accessible critical care database. Scientific data 3, 160035.

[9] Johnson, M.L., Pipes, L., Veldhuis, P.P., Farhy, L.S., Boyd, D.G., Evans, W.S., 2008. AutoDecon, a deconvolution algorithm for identification and characterization of luteinizing hormone secretory bursts: Description and validation using synthetic data. Analytical biochemistry 381, 8–17.

[10] Khalilia, M., Choi, M., Henderson, A., Iyengar, S., Braunstein, M., Sun, J., 2015. Clinical Predictive Modeling Development and Deployment through FHIR Web Services, in: AMIA Annual Symposium Proceedings, American Medical Informatics Association. p. 717.

[11] Marlin, B.M., Kale, D.C., Khemani, R.G., Wetzel, R.C., 2012. Unsupervised pattern discovery in electronic health care data using probabilistic clustering models, in: Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium, ACM. pp. 389–398.

[12] Murray, R.E., Ryan, P.B., Reisinger, S.J., 2011. Design and Validation of a Data Simulation Model for Longitudinal Healthcare Data, in: AMIA Annual Symposium Proceedings, American Medical Informatics Association. p. 1176.

[13] Özçep, Ö.L., Möller, R., Neuenstadt, C., 2014. A Stream-Temporal Query Language for Ontology Based Data Access, in: Joint German/Austrian Conference on Artificial Intelligence (Künstliche Intelligenz), Springer. pp. 183–194.

[14] Perera, S., Sheth, A., Thirunarayan, K., Nair, S., Shah, N., 2013. Challenges in Understanding Clinical Notes: Why NLP Engines Fall Short and Where Background Knowledge Can Help, in: Proceedings of the 2013 international workshop on Data management & analytics for healthcare, ACM. pp. 21–26.

[15] Smith, J.M., Bernstein, P.A., Dayal, U., Goodman, N., Landers, T., Lin, K.W., Wong, E., 1981. Multibase: integrating heterogeneous distributed database systems, in: Proceedings of the May 4-7, 1981, national computer conference, ACM. pp. 487–499.

[16] Walonoski, J., Kramer, M., Nichols, J., Quina, A., Moesel, C., Hall, D., Duffett, C., Dube, K., Gallagher, T., McLachlan, S., 2018. 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, 230–238.

[17] Watkins, R.E., Eagleson, S., Veenendaal, B., Wright, G., Plant, A.J., 2009. Disease surveillance using a hidden Markov model. BMC medical informatics and decision making 9, 39.

[18] Wu, J., Roy, J., Stewart, W.F., 2010. Prediction Modeling Using EHR Data: Challenges, Strategies, and a Comparison of Machine Learning Approaches. Medical care , S106–S113.