Project Title

Integration of Heterogeneous Data and Analytics in The Medical Domain

Hasib Ul Alam
Tanay Gaherwar
Md Kamrul Hasan

Course: IMPRO Project-WiSe 19/20
Supervisor
Dr. Marcela Charfuelan
Dr. Holmer Hemsen
Date: 07/02/2020

Contents

Abstract. *Data integration is crucial for any kind of organization. For any big organization, maintaining data with different departments is difficult. The challenging task in this data integration is heterogeneity. When it comes to the application of the healthcare system, heterogeneity is a big issue. But proper data integration in the medical domain will not only help the doctors to make decisions but also save time. In this project, we build up a scalable integrated model over heterogeneous medical data using MIMIC II [1]. Finally, we visualize historical events and signal via dashboard using Elasticsearch and Kibana.*
Keywords*: Heterogeneity, Medical Data, JSON, Kibana, MIMIC II, Elasticsearch*

1. Introduction

In this computer era, data is growing very fast. For many reasons, it is very necessary to process and analyze data. Sometimes, data analytics is used for making very important decisions. With the growth of data, medical data is also expanding. In the healthcare system, different types of structured, semi-structured and unstructured data exist. There is also a variance in data types in the medical domain. There is clinical data, ICU data and so on. In clinical data, doctors and patients can receive from various heterogeneous sources like Electronic Health Record (EHR), different laboratory tests, patient demographic data, etc. In ICU or time-series data, we can get heart rate, pulse, blood pressure from various devices that provide continuous data with a timestamp. Now we can see the difficulties of heterogeneity. There is heterogeneity in clinical and also in the waveform. Some patients have only clinical data and some patients also have both clinical and waveform data. For those patients who have both data, we have to integrate both form data to monitor and make proper decisions about patient health. Our goal is to make a scalable data integration model for Clinical and ICU data. Another difficulty for working with a medical domain is data availability. As these data are very sensitive, it is hard to get open access to data sources. For our integration data modeling, we used MIMIC II [1] data. In MIMIC II data, there are two types of data: clinical and waveform. There is also a mapper to match the patient data in clinical and in waveform. Using python, we created a JSON file for both clinical and waveform data of a patient as an integrated data model. By using Elasticsearch and Kibana [2], we finally visualize different types of patient data using Kibana visualization tools and queries. We used two approaches to create integrated model as JSON. The first one is a nested object JSON file and another one is flattened JSON files which we described in the later sections. In the later sections, we will discuss the related works for our project and our proposed system and implementation approach. Then we will discuss data processing and visualization techniques. In the end, we will show some limitations to our works and future works.

2. Related Works

In the last decades, there is a lot of research work related to the integration of clinical and ICU data. As there are lots of difficulties to make scalable integrated data models for clinical and time series ICU data, they tried to use different database platforms for time-series and without time-series data. Closing the Data Loop [3] which is an integrated open-access analysis platform for mimic databases. They used mimic iii data [1]. Using the python script, they converted the waveform data into CSV (Comma-separated values)

and then load those data into SciDB. But for clinical data, they used Postgress. For integration, analysis, and visualization, they used R and Shiny. A polystore system for the health care system is proposed in this [4] paper. They used mimic ii data for their system experimentation. They used two back-ends: Postgress for patient clinical data and SciDB for waveform data which is time series. After storing data, they visualize data based on cross-DB queries over different databases. In this [5] paper, they mainly focused on different heterogeneous clinical data. They used an index based information integration system. They designed a Healthcare Act Indexing Information Model (HAIIM) according to the act-centered view of healthcare from HL7 RIM [6]. They used a centralized index which leads to a virtual clinical data center. After that, an integrated query can be run over those heterogeneous clinical data. Hadoop based medical data integrated system is used in this paper [7]. They mainly focused on clinical data. In their system, there are two modules: data integration and data management. In this meta dimensional approach [8], They used clinical data and genomics data sources. For integrating those data, they used Bayesian Networks. After integrating the data, they did some statistical analysis using neural networks. The problem with this approach is that all incoming data are structured. They used SQL and their system has no support for waveform data. In web based personal health care system [9], A patient can collect and manage their respective health information. This project considered clinical data like medical history,past surgeries, medications and so on. They SQLDB DCM4CHE server for their project implementation. From the above studies, we found that there is no such system that can integrate waveform and clinical data in a common platform.

3. Proposed System

One of the themes we saw common in most of the works we referred to was that when systems were dealing with heterogeneous data (one being time-series, other being non-time-series) these systems used different tools to save them, or they relied on modeling them in different databases and cross querying between them. Our goal was to overcome this handicap, and model both forms of data in the same system. The options we narrowed down to were using (i) Influxdb with Grafana for visualization or (ii) Elasticsearch with Kibana for visualization. After probing, we chose Elasticsearch. The reason for this decision was that: (i) Elasticsearch is compatible with JSON data- a format which suited the structure of MIMIC data as it has several headings and subheadings, and also because it was a format all three of us were comfortable with. (ii)Elasticsearch's inherent support for both clinical data and time-series data through its visualizer in Kibana called Timelion. Timelion was built specifically with time-series data in mind. This automatically made Elasticsearch an easy option because of its support for both time-series and clinical data within one tool, reducing the need to model them separately for separate databases.

3.1. Challenges With Modeling

Within Elasticsearch, we further faced modeling challenges. Some of them being:

(i) Elasticsearch treats each row of data inputted as a different object- this leads to a huge performance challenge while inserting large amounts of data into the model. Fortunately, Elasticsearch is made to handle large datasets, and therefore querying performance is not compromised. Since medical data that we had isn't the sort of dataset where new data is frequently added, this drawback does not affect our system considerably.

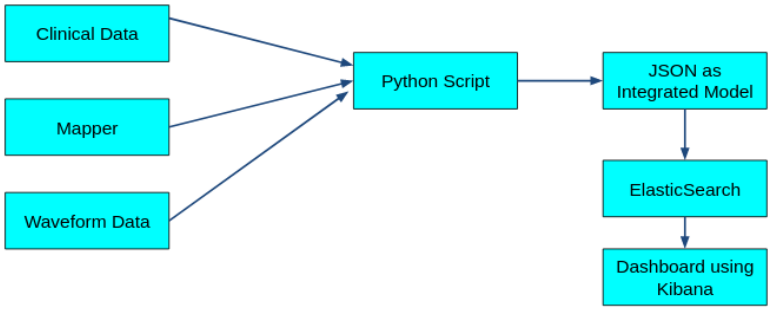

Figure 1. Problem overview

(ii) Nested Objects do not get recognized in Kibana- This problem didn't show up till later, because the nested object is a very intuitive design for MIMIC data and is acceptable for Elasticsearch. Our original model (see model description) was based on using the nested object in JSON format, but since Kibana can not recognize it, we had to remodel our data to eliminate the nested object structure.

(iii) Kibana Dashboards cannot handle different time spans- since the MIMIC dataset has data scarcity: Frequent data is only available for short spans, with different timings between time-series and clinical data, and every patient's data is for a different time period. All these factors make it harder to compare data from different patients in Kibana.

4. Implementation

4.1. Data Collection Description

For our project implementation, we used freely available MIMIC II (Multiparameter Intelligent Monitoring in Intensive Care) data[1]. This data was collected over 7 years from 2001. MIMIC II databases are collected over four categories [10]: 1) Bedside monitoring signal data (waveform) 2) Clinical data 3) Hospital electronic archive 4) Mortality data from SSDI (Social Security Death Index). There are two main data types: Clinical and Waveform. There are over 25,000 patients in this database. Among them, approximately 20,000 are adults and around 5000 are neonates [10]. In waveform data, 3000 patients are included. From 3000 patients in waveform data, 2500 patients are matched with clinical. For every patient, there are multiple ids possibilities. In our project, we used clinical ids and waveform ids. To identify clinical patients from waveform data, we used a mapper files which are included in the data sources.

4.1.1. Waveform Data

Waveform data contains time-series data of physiologic measurements. For signals like blood pressure and RESP in waveform,it gives 125 samples per second. For numerics like HR: heart rate Respiration rate, it gives 1 sample per second/minute. In Fig:02, we can see the sample data for waveform.

```
'Time and date','HR','ABPSys','ABPDias','ABPMean','PAPMean','CVP','PULSE','RESP','SpO2','NBPSys','NBPDias','NBPMean'
'hh:mm:ss dd/mm/yyyy','bpm','mmHg','mmHg','mmHg','mmHg','mmHg','bpm','pm','%','mmHg','mmHg','mmHg'
'[03:10:00 23/08/2012]',0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,-,-,-
'[03:11:00 23/08/2012]',0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,-,-,-
'[03:12:00 23/08/2012]',0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,-,-,-
'[03:13:00 23/08/2012]',0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,-,-,-
```

Figure 2. Waveform data sample

4.1.2. Clinical Data

Intensive care unit data of thousands of patients is stored in clinical databases. There are overview data categories that exist in clinical data [11]: 1) General 2) Physiological 3) Medications 4) Lab Tests 5) Fluid Balance 6) Notes and Reports. Sample data for a patient in clinical data shows on Fig: 03.

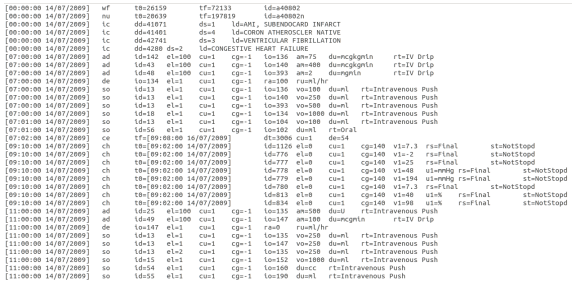

Figure 3. Clinical Data Sample

4.1.3. Mapper

There is a mapper file for matching the clinical data and waveform data for a patient. In mapper file, there are also general information for patients like sex, birthdate and record time.Part of the mapper file is shown in Fig: 04.

Clin. Record	Wave. Record	Sex	Age	Birthdate	Waveform Record Date
s00001	a42216	F	60	22/10/1954	12/11/2014
s00020	a40962	F	76	03/07/1939	30/05/2016
s00033	a42362	M	87	23/10/1929	02/03/2017
s00079	a40392	M	53	19/04/1963	20/09/2016
s00079	a41123	M	53	19/04/1963	20/09/2016
s00123	a40438	M	57	16/08/1954	29/09/2011
s00124	a40616	M	71	12/11/1944	16/07/2016
s00135	a41228	M	52	28/06/1961	29/05/2014
s00138	a40909	M	47	19/10/1963	24/11/2010
s00151	a41282	F	90+	**/**/****	27/06/2016
s00151	a41406	F	90+	**/**/****	24/06/2016
s00175	a42514	M	90+	**/**/****	05/11/2015
s00177	a41497	M	71	10/07/1939	10/03/2011
s00184	a43000	M	60	15/10/1954	05/04/2015
s00194	a41202	M	44	30/04/1967	13/06/2011
s00208	a42425	M	44	22/07/1966	15/08/2010
s00214	a41482	M	67	11/05/1947	28/11/2014
s00214	a42032	M	67	11/05/1947	02/12/2014
s00217	a42236	M	66	25/06/1945	28/11/2011

Figure 4. Segmentation of entire mapper (clinical and waveform data)

4.2. Integrated model

The main goal of our work is to make an integrated model for clinical and waveform data. After creating a common platform, we can easily visualize the data for a patient. In Fig:05, we can see the whole picture of our integrated model. A patient is related to clinical and waveform data. In clinical data, there are lots of events that happen during

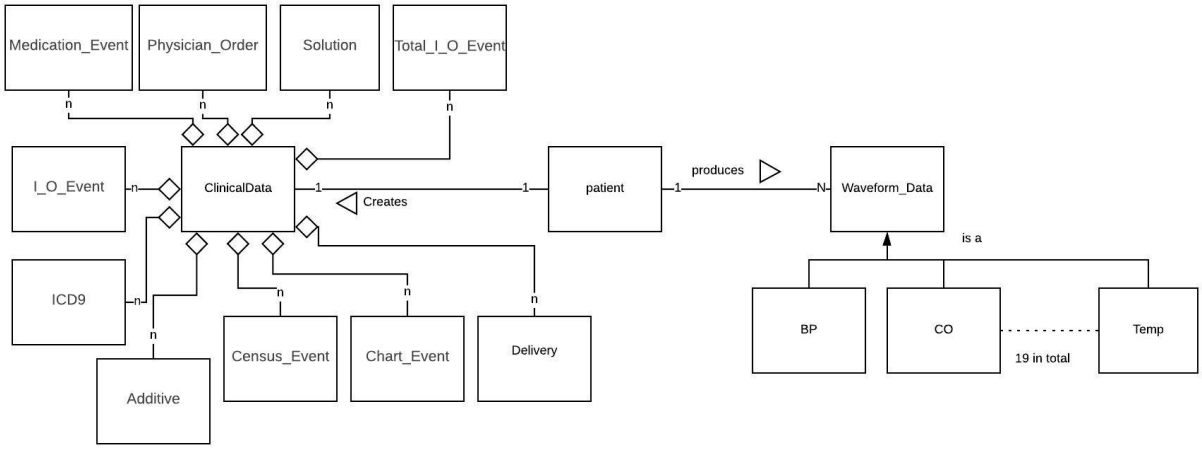

Figure 5. Conceptual model of integrated system

a specific period. From these clinical events, we can retrieve the patient's disease symptoms, medications and so on. From waveform data, we can make the time series diagram to see the ups and downs in heart rate pulse, blood pressure and so on. In the logical diagram in Fig: 06, we can see how data organized in different components of our system. At the top level, there will be general patient information like Ids, sex, and date of birth. Then the patient data will be decomposed into two parts like clinical data and waveform data. The clinical data includes all of its events and waveform data contains signal and numeric data. For better understanding, we also created a tree diagram like Fig: 07 from the logical diagram of our system. After doing studies on different tools for processing and visualizing time-series data, we decided that it would be better to use Elasticsearch which takes JSON format data. That is why we made a common JSON file for every patient which includes clinical and waveform data.After creating an index in Elasticsearch, we load the JSON files into that index of Elasticsearch.

Our first approach (Fig:08) was to create individual documents for each patient file. For that, we needed to create nested objects for each patient record. In this approach, we put all the data of a single patient in a single JSON object. It will contain 5 dictionaries. One for age, one for date of birth, one for id, one for clinical data and another for waveform data. Waveform data is a list where it contains each record as a dictionary. In Clinical Data we have one dictionary where each key is a source. For example ICD9, solutions, delivery and so on. Inside each dict we have a list of all rows as dictionaries based on their source. We uploaded this to elastic search but when the elastic search found data that has nested array / or list it does not map them by themselves. So we created our custom mapping for it and then fed that into elastic search. This model works perfectly on elasticsearch. But unfortunately, when we tried to integrate this with kibana it failed as currently, Kibana cannot perform aggregations across fields that contain nested objects. It also cannot search on nested objects when Lucene Query Syntax is used in the query bar [12]. So in our second approach (Fig:09), we created documents for each line. A single document will contain a timestamp, patient id or cid, and its source and relevant values if it is patient data. If it is from wave data it will contain cid, source, waveform id, all relevant measurements. In this way, if we want to filter for only wave data we can do it

using the source field. In this way, if we want we can also put wave data and clinical data in a separate index. But currently, we are putting them in the same index.

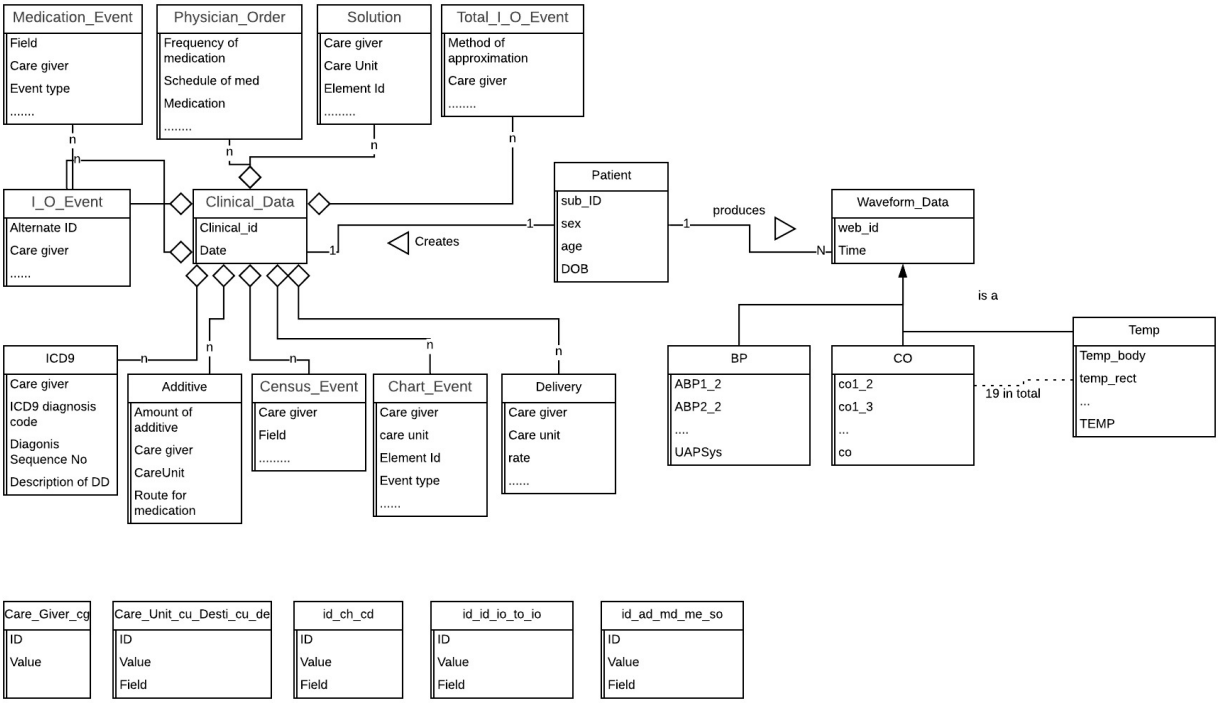

Figure 6. Logical diagram of integrated system

5. Visualization

For data storing and visualization, we used Elasticsearch and Kibana. When we loaded our JSON file in Elasticsearch, it did not work directly. First, we created an index in Elasticsearch as per the requirements of our data. After that, we edited the index schema by mapping some fields in medical data. It did not detect the timestamp data. We specify the date field in the index manually. First, we tried to create one common dashboard for both time-series data and clinical data. For one specific dashboard, we can set only one specific time. But for clinical and waveform data, the data collection times are different. That is why if we want to see waveform data for a while, we could not see the data for clinical in that period and vice versa. That is why we created one dashboard for the clinical data and one dashboard for the waveform data. For our experiment, we used pie chart for clinical data visualization and for waveform, we used Timelion which is a time-series data visualization tools for Kibana. For clinical, we run some queries, which shows the patient's symptoms list and vice versa. Kibana has great tools for filtering data. We can specify the field and also the patient id for visualizing data.

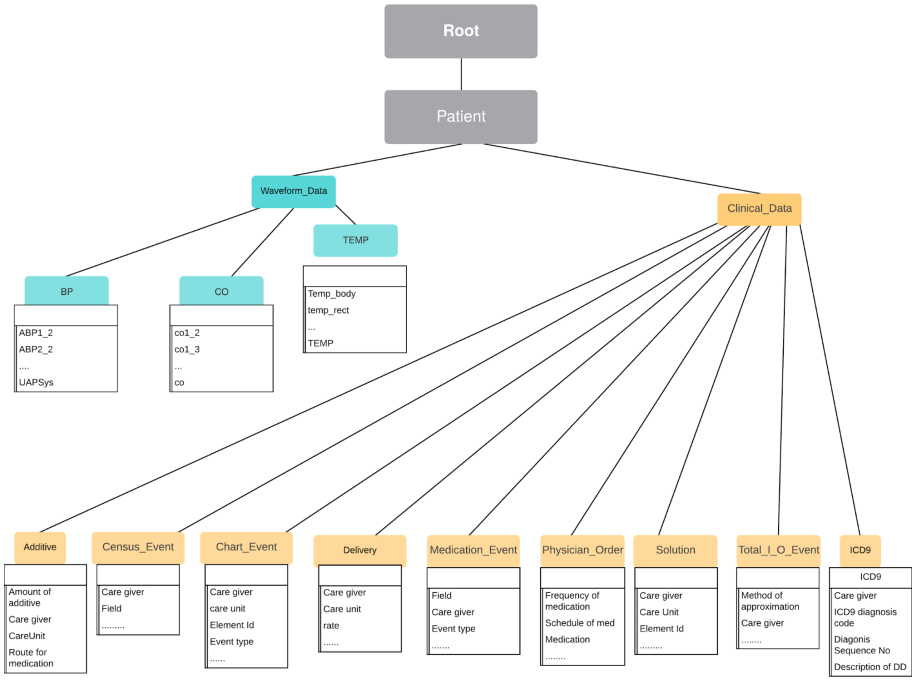

Figure 7. Tree view of the integrated system

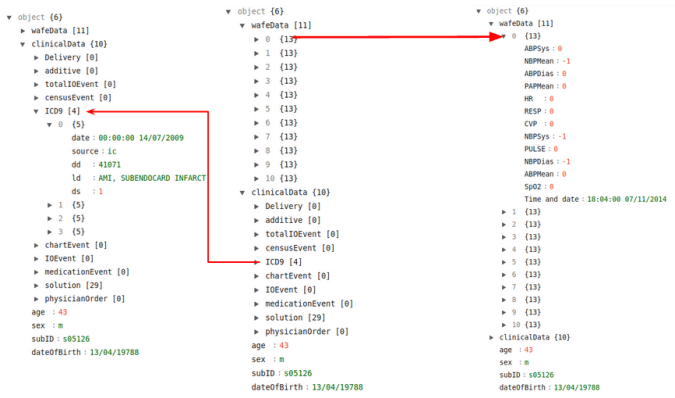

Figure 8. Sample JSON model for a patient: First Approach

5.1. Sample Output For Clinical Data

From Fig: 10, we can see all disease symptoms for all patients. The outer circle represents the patient ids and the inner circle represents the symptoms. But we can filter to see the patient's id list for a particular diseases and also for specific diseases symptoms, we can view the patient ids like Fig:11.

5.2. Sample Output For Waveform Data

In Fig-12, We display the different time-series measurements data like HR, Blood pressure and so on. We can also view the data by combining different signal over time.

```
▼ array [13508]
  ▼ 0  {16}
       Time and date : 18:04:00 07/11/2014
       HR   : 0
       ABPSys : 0
       ABPDias : 0
       ABPMean : 0
       PAPMean : 0
       CVP  : 0
       PULSE : 0
       RESP : 0
       SpO2 : 0
       NBPSys : -1
       NBPDias : -1
       NBPMean : -1
       source : wf
       wid  : a40012n
       cid  : s00318
```


Figure 9. Sample JSON model for a patient: Second Approach

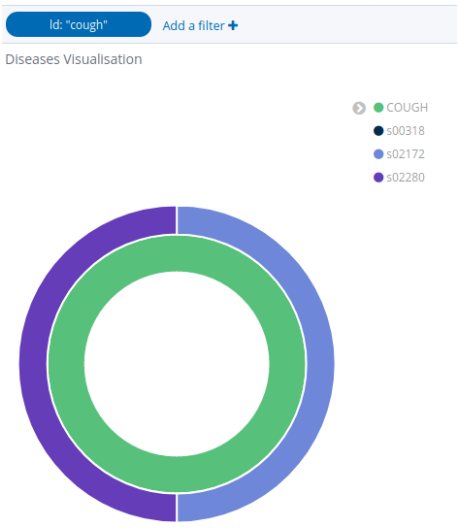

Figure 10. Clinical data visualization: First Filtering

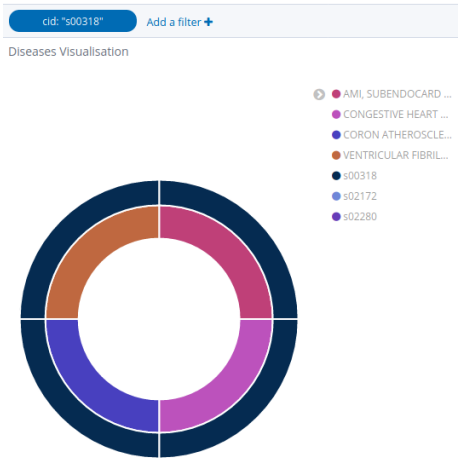

Figure 11. Clinical data visualization: second Filtering

6. Limitation and Future Works

6.1. Limitation

As discussed previously, Kibana has its limitations with the dashboard. The common time for all visual elements is inconvenient. The range of queries that can be performed and the speed with which the data is handled is quite good. Our source of data MIMIC database also has its limitations, as the data is not that regular, nor is it consistent as there are several missing values.

Figure 12. Different timeseries measurements for a patient

6.2. Future Works

For future works, we would like to recommend modeling the MIMIC data in Influxdb and visualization through Grafana (or another competing database system) for comparing benchmarks. The challenge of handling both time-series and clinical data in the same system would make an interesting comparison between the two. For Kibana, future works could include the possibility of trying to query data specific to a medical Doctor's/nurses' needs which would require inputs from someone from the medical domain to form queries that give relevant insights.

7. Conclusion

In conclusion, we showed that it is possible to create one system to accommodate both time-series and non-time-series data. We can see that Elasticsearch is appropriate for such a task with its support for both kinds of data, and its support for JSON, which in our use-case: the MIMIC medical data suits the model. We also observed that modeling data to fit Elasticsearch and Kibana is tricky, and if our system is expecting new data to be added frequently, these tools are not suitable because of the upload performance handicaps. That said, if the data update does not happen frequently, Elasticsearch along with Kibana offers various kinds of visualizations that give really good insights, and allows us to make complicated queries and visualize them not only with standard line-graphs, pie and bar charts, but also Timelion a time-series specific visualization.

References

[1] Mimic ii. `https://archive.physionet.org/mimic2`, Online; accessed 2020.

[2] Elasticsearch. `https://www.elastic.co/de/`, Online; accessed 2020.

[3] D. J. Stone L. A. Celi, R. G. Mark and R. A. Montgomery. Big data' in the intensive care unit: closing the data loop. *American Journal of Respiratory and Critical Care Medicine*, page 1157–1160, 2013.

[4] M. Stonebraker M. Balazinska U. Cetintemel V. Gadepally J. Heer B. Howe J. Kepner T. Kraska et al. A. Elmore, J. Duggan. A demonstration of the bigdawg polystore system. *VLDB*, pages 1908–1911, 2015.

[5] Huilong Duan et al. Jiye An, Xudong Lu. An act indexing information model for clinical data integration. *IEEE*, pages 1099–1102, 2007.

[6] Hl7 reference information model. `http://www.hl7.org/v3ballot/html/infrastructure/rim/rim.htm`, Online; accessed 2020.

[7] Y. Wang D. Y. Tong W. W. Yin J. S. Li D. M. Lyu, Y. Tian. Design and implementation of clinical data integration and management system based on hadoop platform. *ITME*, pages 76–79, 2015.

[8] Andreas Koop Nick Antonopoulos Moeez M. Subhani, Ashiq Anjum. Clinical and genomics data integration using meta-dimensional approach. *UCC*, pages 416–421, 2016.

[9] Agarwal A Pandya AS Ved V, Tyagi V. Personal health record system and integration techniques with various electronic medical record systems. pages 91–94, 2011.

[10] Mimic user guide. `https://mimic.physionet.org/archive/mimic-ii-guide.pdf`, Online; accessed 2020.

[11] Clinical data description. `https://archive.physionet.org/mimic2/mimic2_clinical_overview.shtml`, Online; accessed 2020.

[12] Nested object. `https://www.elastic.co/guide/en/kibana/current/nested-objects.html`, Online; accessed 2020.