Prototyping a Linked List Data Structure to Analyze MIMIC-III Data

Project Report

as part of IE525: Healthcare Delivery Systems

Group 10

Yuyuan Liu, Aman Tukrel, Malfrine Das

April 24, 2018

Abstract:

This report is based on the extraction, development, evaluation and analysis of a data structure on diabetic patients from the Medical Information Mart for Intensive Care III (MIMIC- III) database. After accessing the MIMIC- III database, SQL queries were executed to extract and collect relevant information on patients with Diabetes. Python was utilized to transform the data into a list of linked lists and develop a data structure viable for statistical analysis. The statistical analysis was performed using Minitab to discover a 30% higher average insulin amount administered to patients who died than the average insulin amount administered to patients who did not.

Table of Contents

1. Introduction

The Medical Information Mart for Intensive Care III (MIMIC-III) database was created at the Laboratory for Computational Physiology at the Massachusetts Institute of Technology (MIT) to facilitate data-driven clinical decision-making. MIMC-III integrates anonymous clinical, numeric, and waveform data of patients admitted to the Beth Israel Deaconess Medical Centre in Boston, Massachusetts. Though MIMIC-III offers researchers unprecedented access to healthcare data, significant data mining is often required to allow researchers to employ the data for analysis and model validation. Researchers at the Regenstrief Center for Healthcare Engineering (RCHE) would like to transform data from MIMIC-III into a linked list structure to better understand the data using neural network algorithms.

The objective of this project is to develop a proof-of-concept linked list data structure that researchers can use when employing neural network algorithms. To better frame the project, the data structure will be developed to answer the following question: Is there a relationship between a diabetic patient's mortality rate and the total amount of insulin they are administered? A simple question was chosen so group members may focus their attention on developing the data structure and assessing its viability for future use.

The project objective will be addressed in this paper in the following manner:

1. Background:
    a. A summary of MIMIC-III data and linked lists is provided in section 2.1 and 2.2, respectively
2. Method:
    a. An explanation of the processes used to query and transform the MIMC-III data is presented in Section 3.2 and 3.3, respectively
3. Results:
    a. The computation time of the linked list data structure is presented in section 4.1
    b. A statistical analysis on the queried data is provided in section 4.2
4. Discussion:
    a. A discussion on Computation Time is presented in section 5

b. A discussion on Mortality Rate, difference in insulin administration amounts to alive and deceased patients, and causation $\neq$ correlation is provided in section 5

5. Conclusion:
    a. Project takeaways and further recommendations are provided in section 6

2. Background

2.1. MIMIC-III

In this project, the data from MIMIC-III will be queried and analysed to address the project objective. MIMIC-III integrates anonymous, comprehensive, and widely accessible clinical data to improve validation and innovation in healthcare research. The database is comprised of data from the following sources:

- Hospital Electronic Health Records (EHR)
- Archives from critical care information systems
- Social Security Administration Death Master File

MIMIC-III is a relational database consisting of 26 tables that are linked using unique identifiers for patients, hospital events, and medical diagnoses. Table 1 describes the 9 data classes that the database is separated into.

Table 1: The 9 Different Data Classes of the Database

Class	Description
Billing	Coded data mostly for billing and administrative needs
Descriptive	Details regarding patient demographics, admission, discharge and mortality
Dictionary	Cross-referencing concept identifiers
Interventions	Medical procedures
Laboratory	Laboratory test results
Medications	Records of medications and medication orders
Notes	Free text notes provided by nurses and doctors
Physiologic	Nurse-verified patient vital signs on an hourly basis
Reports	Free text reports of electrocardiogram and imaging studies

MIMIC-III is not readily accessible to the public. To gain access to the database, researchers must provide CITI “Data or Specimens Only Research” training certification and

receive approval from a supervisor who already has access to the database. Due the sensitive nature of health records, users must ensure that throughout their research, patient data is protected. To protect data in the Purdue University cluster, researchers use Secure Shell protocol when remotely accessing MIMIC-III. To securely transfer data files, researchers use a secure transferring software called SecureFX.

2.2. Linked Lists

Data structures provide a way of organizing and storing data so they can be accessed and modified efficiently based on the requirements of an algorithm. Linked lists are a specialized and uncommon data structure. Each element in a linked list is a separate object, often referred to as a node. The node is comprised of two items: data and a reference to the next node in the linked list. The first node in a linked list is called the head and does not contain any data. Similarly, the last node is called the tail and it references a null object.

Since the nodes of a linked list are separate objects, they do not need to be stored contiguously in a computer's memory. This improves overall memory allocation and computational efficiency when adding and removing objects in a linked list especially in comparison to more conventional data structures like arrays and array lists. However, when searching and traversing through elements, linked lists are much more inefficient when compared to arrays and array lists. Researchers at RCHE would like to develop a linked list because it aggregates data chronologically based on the events that happen to a patient when they are in the hospital. Using this structure, researchers will have the ability to employ more sophisticated algorithms to analyse MIMIC-III data.

3. Method

3.1. Overview

Figure 1 is a flowchart of the 4 stages of the project. Relevant Electronic Health Records (EHR) data will be mined and cleaned from MIMC-III using SQL in DataGrip. EHR data provides information regarding the processes a patient goes through during their visit. Each process pertinent to our project was classified as an event type. Using these event types, a linked list structure that profiles a patient's hospital visit was developed. The linked list was queried and analysed to answer the question proposed in the project overview. Python was used to transform and query the linked list. Minitab was used to perform a statistical analysis on the resultant data. Based on the results, the viability of the linked list data structure for future was assessed. A URL to the project code can be found in Appendix A. It is publicly hosted on Github.

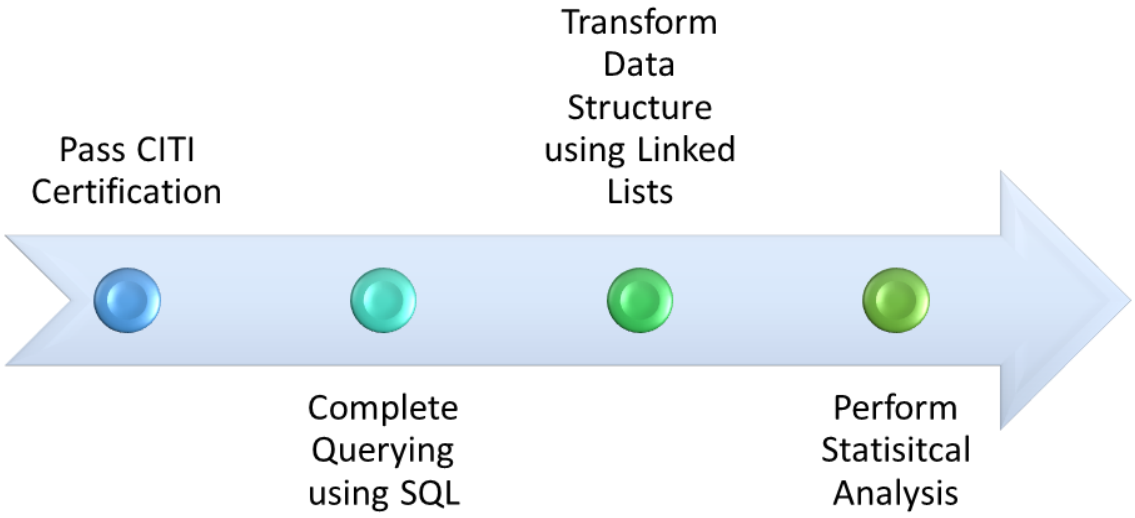

Figure 1: Basic project timeline

3.2. Querying MIMIC-III

Access to MIMIC-III was requested from researchers at MIT using physionet.org PuTTYgen was used to generate a public/private key pair for authorization. Using DataGrip, MIMIC-III was accesses using an SSH tunnel. The database was queried, and its output was exported as a CSV file.

Table 2 lists and describes the attributes that were collected from MIMIC-III and exported as CSV. Table 2 lists and describe the tables in MIMIC-III in order to retrieve the relevant data. The relevant data required to address the project question were found in the following tables:

- D_ICD_DIAGNOSES provides the definition of International Classification of Diseases Version 9 (ICD-9) codes for diagnoses, including the ICD9 code, short and long title for the diagnoses. By searching "diabetes" in long title, the ICD9 code for all diabetes was found.
- DIAGNOSES_ICD contains ICD diagnoses that are assigned to patients at the end of their hospital stay. The resultant set of diabetes patient IDs can be queried by intersecting this table with D_ICD_DIAGNOSES with diabetes' ICD9 codes.
- INPUTEVENTS_MV records input events for patients, including all relevant information such as patient ID, ICU stay ID, weight, caregiver ID, input item, amount, rate, item status, and event changes.
- ADMISSIONS provides information regarding a patient's admission to the hospital, including admission and discharge time, demographic information, the source of the admission, and so on

Table 2: The attribute, table name and description of queried data

Attribute	Table Name	Description
ICD9_CODE	D_ICD_DIAGNOSES	Diagnostic concept code
LONG_TITLE	D_ICD_DIAGNOSES	Brief definition for the given diagnosis code
SUBJECT_ID	DIAGNOSES_ICD + INPUTEVENTS_MV	Unique patient ID
HADM_ID	INPUTEVENTS_MV	Unique patient hospital stay ID
ORDERCATEGORYNAME	INPUTEVENTS_MV	Higher level information about the ordered

		medication/solution
STARTTIME	INPUTEVENTS_MV	Start time of an input/output event
ENDTIME	INPUTEVENTS_MV	End time of an input/output event
AMOUNT	INPUTEVENTS_MV	Amount of a drug or substance administered to the patient
ADMITTIME	ADMISSIONS	The date and time the patient was admitted to the hospital
DISCHTIME	ADMISSIONS	The date and time the patient was discharged from the hospital
DEATHTIME	ADMISSIONS	The time of in-hospital death for the patient, only present if the patient died in-hospital
HOSPITAL_EXPIRE_FLAG	ADMISSIONS	Binary flag which indicates whether the patient died

Figure 2 provides a visualization of the queries used to collect the required data. By searching "insulin" in INPUTEVENTS_MV table under the attribute ORDERCATEGORYNAME, the IDs of patients who were administered insulin as well as their hospital stay ID is retrieved. By combining the results of patient ID sets from DIAGNOSES_ICD and INPUTEVENTS_MV, the SUBJECT_ID of diabetes patients who were administered insulin is recovered. Using this SUBJECT_ID set, the hospital stay ID set, and adding "insulin" as a filter, the information regarding insulin administration start times, end times and amount were recovered. By using the same SUBJECT_ID set and HADM_ID set, data regarding admission time, discharge time, and, if applicable, death time, for the hospital visits that had insulin administration were obtained. Finally, all data sets were joined into one table by using patient ID and hospital visits ID as keys. The resultant set was ordered by patient ID and time.

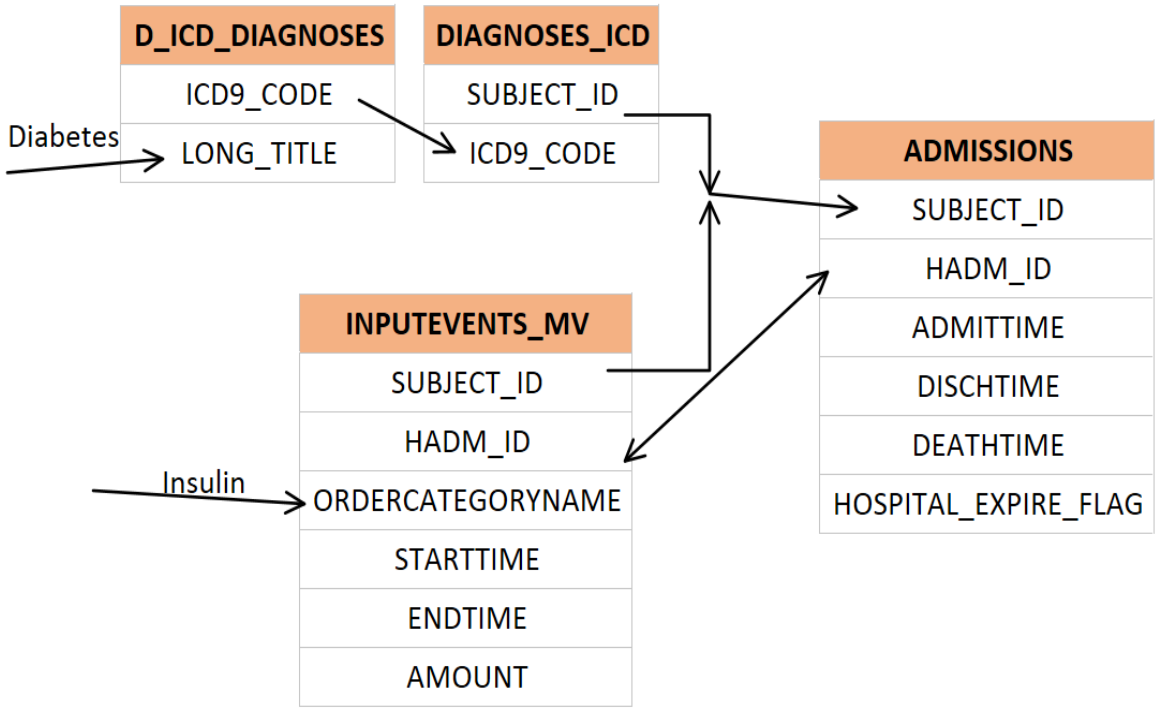

Figure 2. Visualization of MIMIC-III queries

3.3. Transforming Data

After the data was collected from MIMIC-III as a raw .csv file, it was transformed into a list of linked lists. The linked list class was named EHRLink and Figure 3 provides an overview of the EHRLink structure. For each unique patient ID found in the raw data, a corresponding EHRLink was created. Each EHRLink is comprised of a series of nodes, called events, that represent the processes a patient undergoes while at the hospital. Only the event types that were considered relevant to the project structure were included. The event types along with their corresponding properties are shown in Table 3.

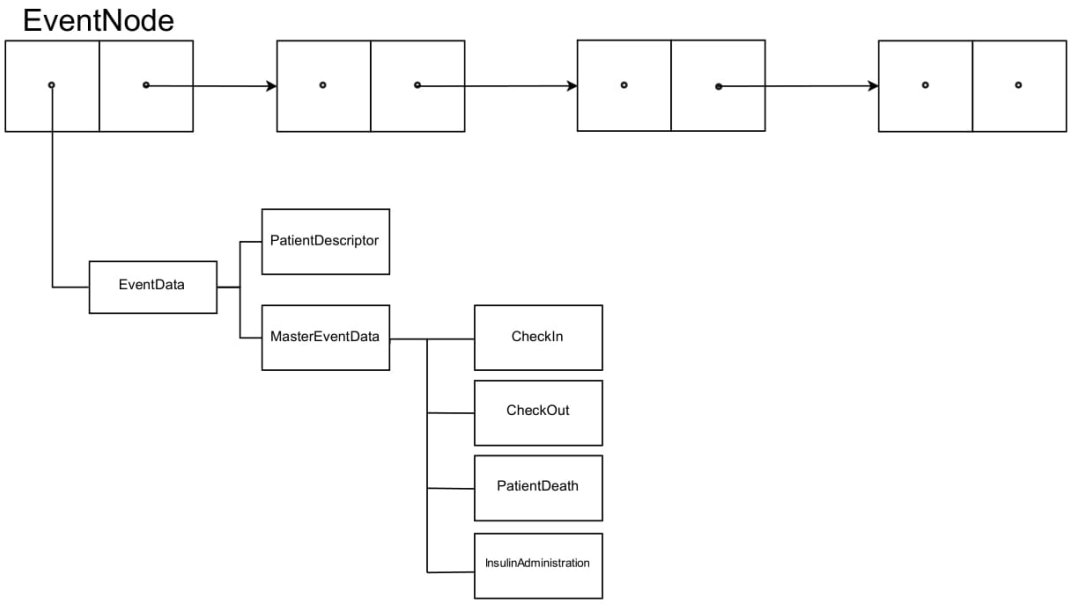

Figure 3: Overview of EHRLink Structure

Table 3: The Different Event Types, Class Names and Mimic- III Data Tag

Event Type	Event Type Class Name	MIMIC-III Data Tag
Patient Check-In	CheckIn	admittime, hadm_id
Patient Check-Out	CheckOut	dischtime, hadm_id
Patient Death	PatientDeath	deathtime, hadm_id
Insulin Administration	InsulinAdmin	starttime, endtime, hadm_id, amount
Patient Description	PatientDecsriptor	subject_id

The design principles used when developing EHRLink were scalability, readability, and computational efficiency. Since there are thousands of possible event types that can be developed if the linked list is implemented on a larger scale, it was essential that node references and data types were developed robustly. A class called MasterEventData was created to separate the reference and data section of each event node in EHRLink. Additional subclasses of

MasterEventData were created corresponding to the event types listed in Table 3. For now, this master class has properties for event timestamp and visit ID as well as methods for class instantiation and equality checking.

Table 4 below describes the methods developed for the EHRLink class. These methods were used to traverse through the list of EHRLinks and collect relevant data for analysis. Data was collected to determine the total number of visits, insulin administered, and death of each diabetic patient.

Table 4: The Methods Used to Develop EHRLINK Class

Method Name	Description
Add	Addition of a given node
Remove	Remove a given node
Search	Search for a given node
Size	Retrieve the size of the linked list

4. Results

4.1. Computation Time

Table 5 shows the average time it took to transform the raw MIMIC-III data into a list of EHRLinks and the time it took to traverse the list of EHRLinks to collect relevant data. The program was executed 1000 times to determine the average computation time.

Table 5: Average and Traverse Time

Transform Time	Traverse Time
13.9 s	8.84×10^{-2} s
2.36×10^{-4} s / row	2.03×10^{-5} s / patient

4.2. Descriptive Statistics

Tables B1, B2 and B3 and Figure 4 visualize the data queried from the list of EHRLinks. Figure 4 is a histogram of total patients, visits and total insulin per visit. In order to find a difference between the deceases and living, the mean values of total insulin administered, number of visits and insulin per visit were calculated by segregating the data into two parts: those who died and those who did not. Table 6 and Figure 5 compare the amount of insulin administered to patients who are still living and those who are deceased.

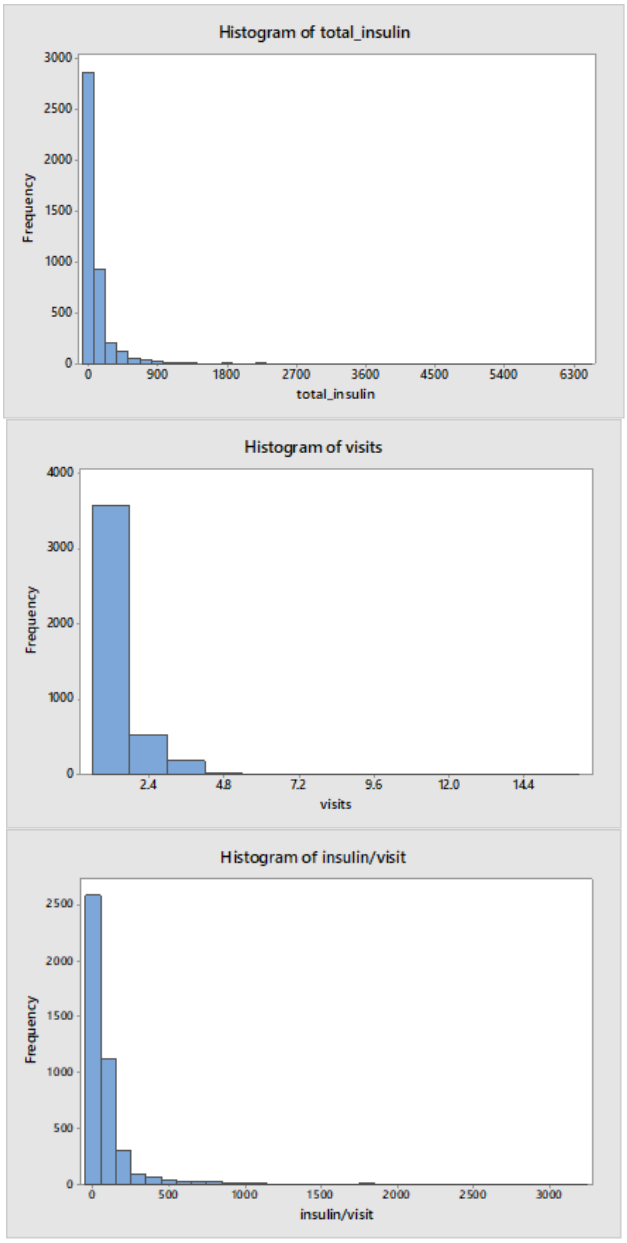

Figure 4: Histogram of total insulin, visits and insulin per visit

Table 6: Comparison of parameters for alive and deceased patients

Parameter	Alive	Deceased	Difference
insulin/visit	86.91	113.25	30.30 %
insulin	112.68	156.1	38.53 %
visits	1.2731	1.3545	6.39 %

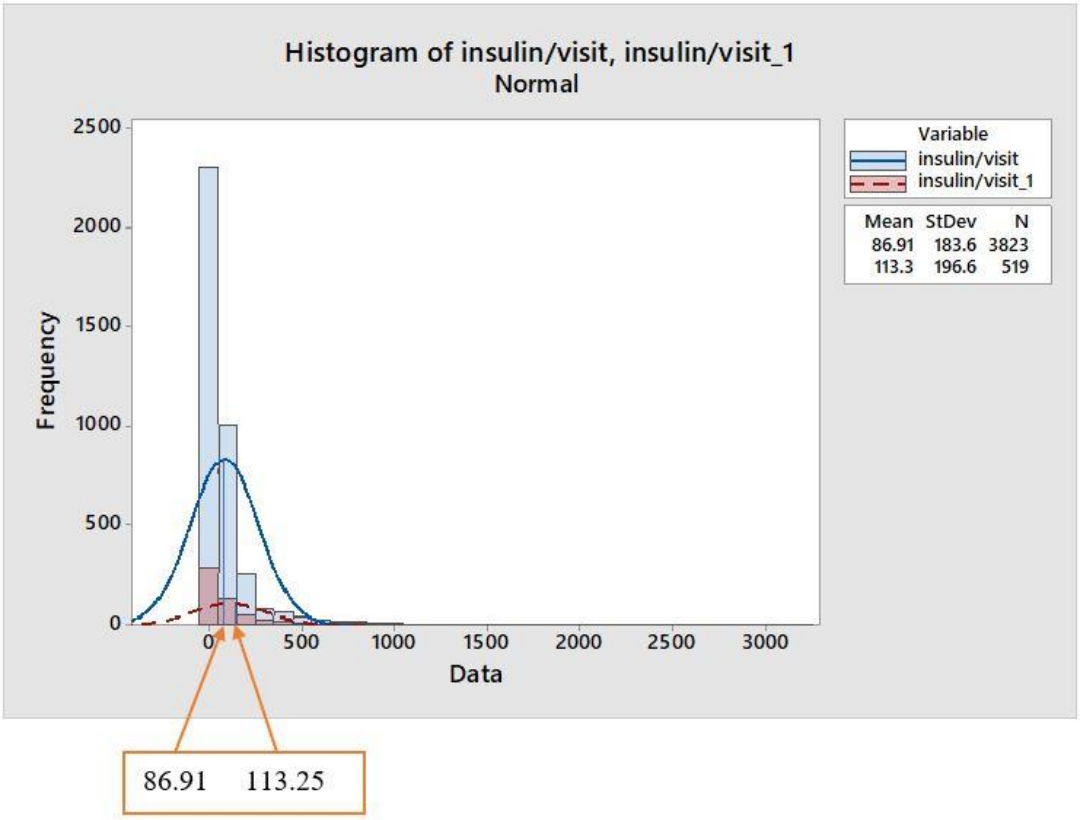

Figure 5: Comparison of means of insulin/visit to patients who died VS that to those who did not

5. Discussion

The data from section 4.2 makes one fact very evident: patients who died were administered 30% higher insulation amount per visit than the amount per visit to patients who did not. However, this data considers only 4342 patients, out of which 519 died. To make a more substantial claim, more data that is usable would be needed. According to a (Kitabchi, 1 March 2002), among patients admitted to the ICU, patients with newly diagnosed hyperglycemia had a 3-fold higher mortality rate (31%) than patients with a known history of diabetes or with normoglycemia who had ICU mortalities of 10% and 11.3%, respectively ($P < 0.01$). Our calculations show mortality rate of the patients to be 11.95%, which is close to the findings of the above paper.

Based on the results, it was concluded that the EHRLink structure that was developed provides a useful and effective method to aggregate data for future use. Due to the limited size of the raw data used, it was determined that further testing is required to conclusively prove if EHRLink is computationally efficient. Preliminary results show that it traversal time is quite minimal. Additionally, aggregating and transforming tabular data into EHRLink is relatively quite computationally costly. However, since it is likely that the transformation process is only executed once, the cost of computation is satisfactory. It is imperative that additional testing and development occur before a linked list data structure like EHRLink is implemented on a large scale. The design team recommends the following to be taken into consideration for future use:

1. Develop an importing / exporting method to store the ERHLink structure. Saving the EHRLink as an XML or JSON structure would likely be the most effective method. Python also has well established libraries, like pickle and shelf, that save the current python workspace variables. Though these libraries tend to be less reliable than XML or JSON structures, they are much quicker to implement.
2. Develop an automated method to create EventNodes. Currently, the EventNodes and their respectively node addition methods are hard coded into the Node class in order to complete project deliverables in a timely manner. In a traditional ICU setting, it is likely that there will be thousands of event types to account. If

EHRLink is to be implemented on a larger scale, it is recommended that a more scalable method for developing EventNode data classes.

3. Explore the possibility of implementing a hash table structure as opposed to a linked list. Like linked lists, the element addition method in hash tables have an average computation complexity of *O(1)*. However, hash tables are faster than linked lists when being traversed as they have an average computational runtime of *O(1).* It was assumed that a more in-depth analysis of data structure was completed prior to the prototyping phase of the project.
4. For the next phase of the project, it is recommended that a larger raw dataset be implemented in order to fully analyse the computational complexity of EHRLink. In this report, approximately 60 thousand rows of data were analysed in 12s. Since artificial intelligence algorithms require data in the order of gigabytes, it is recommended that the next phase use test data of similar size.

6. Conclusions

In this report, a prototype linked list data structure called EHRLink was developed in order to transform and analyse diabetic patient data from the Medical Information Mart for Intensive Care III (MIMIC-III) database. The feasibility of EHRLink was evaluating by addressing the following question: Is there a relationship between a diabetic patient's mortality and the amount of insulin administered to them while in intensive care. The following conclusions were drawn after analysing the patient data and the feasibility of EHRLink:

1. Patients who died were administered 30% higher insulin amount per visit than the amount per visit of patients who did not.
2. EHRLink provided an effective framework for aggregating data in order to perform events-based analysis.
3. Only a limited understanding of the computational efficiency of EHRLink could be ascertained. This is because the project question that was asked provided a very small amount of data (~20 MB).

4. It is recommended that an automated method be developed to create event nodes in the linked list. This is due the sheer magnitude of possible events types in a hospital that need to be accounted for. Additionally, it would be beneficial to also develop a method to export EHRLink data as an XML of JSON for stable data storage.
5. It is recommended that other data structures, like hash tables, be explored for this project because traversing through linked lists can often be computationally costly.

7. References

1. Guillermo E. Umpierrez, Scott D. Isaacs, Niloofar Bazargan, Xiangdong You, Leonard M. Thaler, Abbas E. Kitabchi. Hyperglycemia: An Independent Marker of In-Hospital Mortality in Patients with Undiagnosed Diabetes. *The Journal of Clinical Endocrinology & Metabolism*, Volume 87, Issue 3, 1 March 2002, Pages 978–982
2. MIMIC-III, a freely accessible critical care database. Johnson AEW, Pollard TJ, Shen L, Lehman L, Feng M, Ghassemi M, Moody B, Szolovits P, Celi LA, and Mark RG. Scientific Data (2016). DOI: 10.1038/sdata.2016.35. Available from: http://www.nature.com/articles/sdata201635

8. Appendix

8.1. Appendix A

https://github.com/malfrine/EHRBlockChain

8.2. Appendix B

Tables B1, B2, and B3 provide the descriptive statistics for the all patient, alive patients, and deceased patient, respectively.

Table B1: Descriptive Statistics: total_insulin, visits, insulin/visit (all)

Variable	N	N*	Mean	SE Mean	StDev	Minimum	Q1	Median	Q3	Maximum
total_insulin	4342	0	117.88	3.99	262.78	0	16	42	110	6440.5
Visits	4342	0	1.2828	0.0127	0.8339	1	1	1	1	16
insulin/visit	4342	0	90.05	2.81	185.34	0	14	36.17	88	3220.25

Table B2: Descriptive Statistics: insulin/visit, total_insulin, visits (not dead)

Variable	N	N*	Mean	SE Mean	StDev	Minimum	Q1	Median	Q3	Maximum
insulin/visit	3823	0	86.91	2.97	183.55	0	14	36	84	3220.25
total_insulin	3823	0	112.68	4.18	258.7	0	16	41	105	6440.5
Visits	3823	0	1.2731	0.0135	0.8372	1	1	1	1	16

Table B3: Descriptive Statistics: insulin/visit, total_insulin, visits (dead)

Variable	N	N*	Mean	SE Mean	StDev	Minimum	Q1	Median	Q3	Maximum
insulin/visit	520	0	113.02	8.62	196.52	-10	14	38.17	123.88	1764.5
total_insulin	520	0	155.8	12.6	288.3	-10	14.3	50.5	154	3194
visits	520	0	1.3538	0.0353	0.8057	1	1	1	1	6