

Impact of Information Technology on the Healthcare Sector

Nate Hanrahan, Saransh Rakshak

Due: April 27, 2025

Contents

Introduction	1
Data Preparation	2
Loading Data	2
Data Information	2
Cleaning Data Column Names	3
Merging Dataframes	3

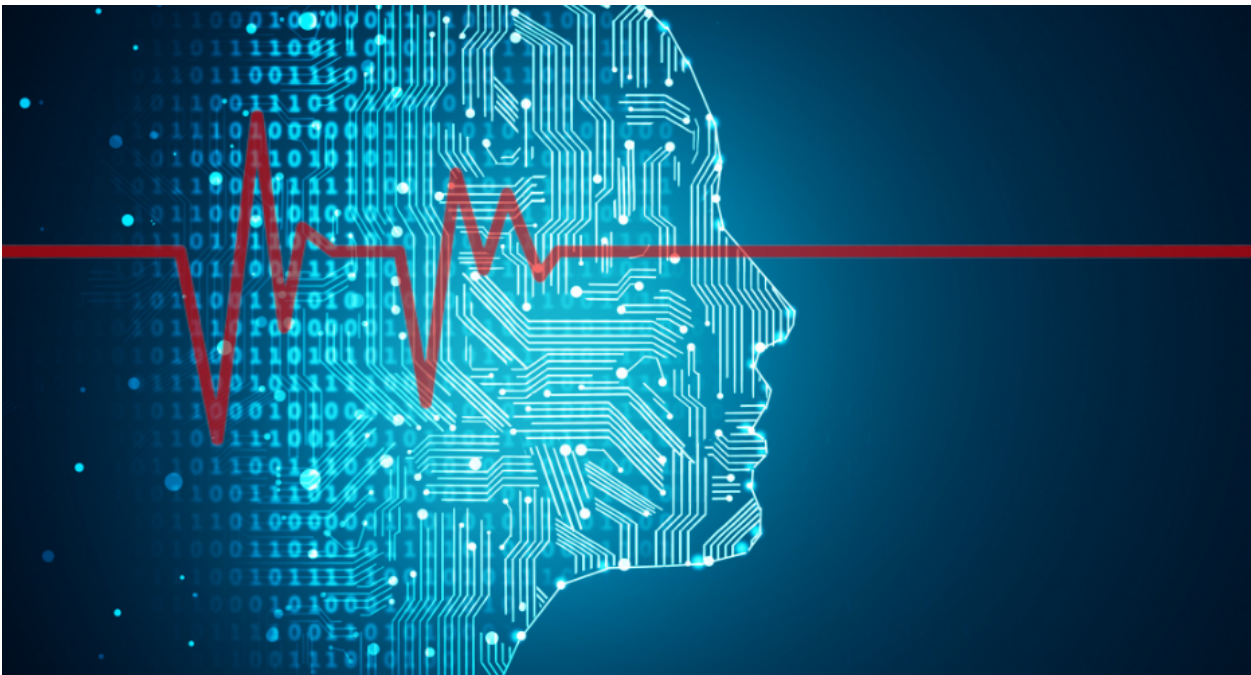


Figure 1: AI Healthcare

Introduction

The 2017 Healthcare Information and Management Systems Society (HIMSS) Conference & Exhibition was one of the world’s largest and most influential healthcare IT events. Held from February 19–23, 2017

at the Orange County Convention Center in Orlando, Florida, the event brought together over 40,000 healthcare and IT professionals, including clinicians, policymakers, researchers, tech executives, industry leaders & more. With more than 1200 exhibits and hundreds of educational sessions, HIMSS 2017 served as a dynamic forum for exploring the constantly growing convergence of healthcare and technology. Key themes included cybersecurity, patient engagement, tele-medicine, and deep/machine learning integration for assistive technology, highlighting the growing need for connected and trained systems that support a rapidly growing healthcare landscape.

Since the 2017 conference, groundbreaking technologies have emerged with the potential to alter how healthcare data is managed, analyzed, and protected. Among these, Deep Learning has shown exceptional promise in areas such as medical imaging, diagnostics, and clinical decision assistance - offering accuracy beyond normal practice. A growth in ethical tech like differential privacy - which enable the sharing and analysis of sensitive health data while preserving individual privacy — shows an increasing concern about earning user trust in the era of big data and personalized medicine. These innovations represent both technical progress and also a shift in how healthcare systems operates at massive scale while maintaining trust, security, and ethical responsibility.

One of the key enablers behind the rise of these advanced data technologies is the significant improvement in computing infrastructure - especially through the widespread adoption of Graphics Processing Units (GPUs). Technology like NVIDIA's CUDA have transformed GPUs from gaming hardware into essential tools for scientific computing, enabling massive parallel processing that powers modern AI and machine learning applications. As the cost of computing has decreased, high-performance tools—such as dedicated GPUs—have become more accessible and standardized, even in personal and enterprise-level systems. This democratization of computing power has allowed a broader range of researchers, developers, and healthcare organizations to experiment with and deploy data-intensive solutions that were once limited to well-funded labs or tech giants.

The focus areas of HIMSS 2017 and the emergence of next-generation information technologies point toward a radical shift for the healthcare sector. As artificial intelligence, improvements in patient privacy, and scalable data infrastructure get better, they allow the possibility of data-driven personalized care. Our report explores how these technologies are driving change across the healthcare sector - from hospital workflows and patient experience, to large-scale population health management. By analyzing data sets from HIMSS 2017, we will begin to understand how convergence of healthcare and technology is redefining the foundations of patient care.

Data Preparation

Loading Data

```
# load data
HAEntity <- read.csv("data/HAEntity.csv")
HA_NPI <- read.csv("data/HAEntityNPI.csv")
UseOfIT <- read.csv("data/UseOfITComponent.csv")
Pharma <- read.csv("data/Pharmacy.csv")
PharmaProds <- read.csv("data/PharmacyProduct.csv")

# head(PharmaProds)
```

Data Information

1. Our first collection of datasets comes directly from the HIMSS 2017 server.

A. *HAEntity*:

- Description: This table contains the demographic information for all the facilities and systems tracked.
- Link: This table links to the other tables by the 'HAEntityID'. This table also links to itself by 'HAEntityID'.

B. *HA_NPI*:

- Description: This table provides information regarding medication prescriber's NPI (National Provider Identifier).
- Link: This table is connected to HAEntity by 'SurveyId' and 'HAEntityID'.

C. *Pharmacy*:

- Description: This table provides an overview of the pharmacy department, including FDA bar code regulated products.
- Link: This table links to the other tables by the 'HAEntityID'.

D. *PharmacyProduct*:

- Description: This table contains the vendor and product detail for the pharmacy department.
- Link: This table links to the other tables by the 'HAEntityID'.

Cleaning Data Column Names

```
HAEntity <- HAEntity %>%  
  rename(Care_Type = Type)  
  
HA_NPI <- HA_NPI %>%  
  rename(ProviderIdentifier = id)  
  
UseOfIT <- UseOfIT %>%  
  rename(Tech_Type = Type)  
  
Pharma <- Pharma %>%  
  rename(Pharma_Id = Id)  
  
PharmaProds <- PharmaProds %>%  
  rename(Pharma_Prod_Type = Type) %>%  
  rename(Pharma_Prod_Id = Id)
```

Merging Dataframes

```
# left join NPI to HAEntity  
data <- HAEntity %>%  
  left_join(HA_NPI, by = c("SurveyId", "HAEntityId")) %>%  
  left_join(UseOfIT, by = c("SurveyId", "HAEntityId"))
```

```
## Warning in left_join(., UseOfIT, by = c("SurveyId", "HAEntityId")): Detected an unexpected many-to-many relationship
## i Row 138 of 'x' matches multiple rows in 'y'.
## i Row 12429 of 'y' matches multiple rows in 'x'.
## i If a many-to-many relationship is expected, set 'relationship =
## "many-to-many" to silence this warning.
```

```
# join Pharma/Prods
data <- data %>%
  left_join(Pharma, by = "HAEntityId") %>%
  left_join(PharmaProds, by = "HAEntityId")
```

```
## Warning in left_join(., PharmaProds, by = "HAEntityId"): Detected an unexpected many-to-many relationship
## i Row 210 of 'x' matches multiple rows in 'y'.
## i Row 1683 of 'y' matches multiple rows in 'x'.
## i If a many-to-many relationship is expected, set 'relationship =
## "many-to-many" to silence this warning.
```

```
# check mismatches from splitting SurveyId cols
mismatched <- which(!is.na(data$SurveyId.y) & (data$SurveyId.x != data$SurveyId.y | data$SurveyId != data$SurveyId.y))
if (length(mismatched) == 0) {
  data <- data %>%
    select(-SurveyId, -SurveyId.y) %>%
    rename(SurveyId = SurveyId.x)
} else {print("Corrupted SurveyId Values - Mismatch")}

colnames(data)
```

```
## [1] "HAEntityId" "SurveyId"
## [3] "ParentId" "UniqueId"
## [5] "EntityNo" "Name"
## [7] "HAEntityTypeId" "HAEntityType"
## [9] "CBSA" "MedicareNumber"
## [11] "Address1" "Address2"
## [13] "City" "State"
## [15] "Zip" "Phone"
## [17] "Website" "Fax"
## [19] "EmailConvention" "ProfitStatus"
## [21] "ServicePopulation" "Care_Type"
## [23] "YearOpened" "OwnershipStatus"
## [25] "NofFTE" "VendorSelStrategy"
## [27] "NofBeds" "NofStaffedBeds"
## [29] "FreeStanding" "SameISSystem"
## [31] "AcuteId" "NofPhysicians"
## [33] "NofHCCareVisits" "FTETotal"
## [35] "DataCenterAcuteId" "PhysFT"
## [37] "PhysAffiliated" "PhysTotal"
## [39] "IsImaging" "PhysResidents"
## [41] "PhysHospitalists" "PhysOther"
## [43] "NofIntensiveCareBeds" "AverageDailyCensus"
## [45] "Latitude" "Longitude"
## [47] "IsIDSACOCClassified" "IsIDSPCMHCertified"
## [49] "IsIDSPlanningACO" "IsIDSPlanningPCMH"
## [51] "NoOfNeonatalIntensiveCareBeds" "PhysicianExtenders"
```

## [53] "PercVirtServers"	"PercVirtComputers"
## [55] "County"	"DisasterMsgId"
## [57] "IsNICUPresent"	"ProviderIdentifier"
## [59] "NPI_Number"	"ServiceType"
## [61] "Id"	"ComponentId"
## [63] "Component"	"ComponentOther"
## [65] "Perc"	"Tech_Type"
## [67] "Pharma_Id"	"ADM"
## [69] "ADMPlanned"	"ADMTimeFrame"
## [71] "ADMPerc"	"Robot"
## [73] "RobotPlanned"	"RobotTimeFrame"
## [75] "DeptMedical"	"DeptED"
## [77] "DeptOR"	"DeptOther"
## [79] "DeptOtherName"	"EMAR_CPOE"
## [81] "CarouselsTimeFrame"	"CarouselsPlanned"
## [83] "Carousels"	"InventoryMgmt"
## [85] "PharmaceuticalSupplyVendorID"	"PharmaceuticalSupplyVendorName"
## [87] "Pharma_Prod_Id"	"VendorId"
## [89] "VendorName"	"ProductId"
## [91] "ProductName"	"Pharma_Prod_Type"
## [93] "NofDevices"	"ContractYear"
## [95] "ContractMonth"	"Qty"

2. The following sets come from ...