Distribution Agreement

Signature:

______________________________ __________

Fatemeh Amrollahi Date

An Scalable and Interoperable Real-time Software Platform
for Forecasting the Onset-time of Sepsis

By

Fatemeh Amrollahi
Master of Science

Computer Science and Informatics

Shamim Nemati, Ph.D.
Advisor

Ashish Sharma, Ph.D.
Committee Member

Babak Mahmoudi, Ph.D.
Committee Member

Accepted:

Lisa A. Tedesco, Ph.D.
Dean of the James T. Laney School of Graduate Studies

Date

An Scalable and Interoperable Real-time Software Platform
for Forecasting the Onset-time of Sepsis

By

Fatemeh Amrollahi
B.Sc., Amirkabir University, Iran, 2015

Advisor: Shamim Nemati, Ph.D.

An abstract of
A dissertation submitted to the Faculty of the
James T. Laney School of Graduate Studies of Emory University
in partial fulfillment of the requirements for the degree of
Master of Science
in Computer Science and Informatics
2019

Abstract

An Scalable and Interoperable Real-time Software Platform
for Forecasting the Onset-time of Sepsis
By Fatemeh Amrollahi

Sepsis, a dysregulated inflammatory response to infection, is difficult to diagnose in advance of life-threatening physiological decompensations. Multiple studies have demonstrated improved outcomes when this condition is recognized and treated early. Nemati et al. have developed a real-time, high-dimensional machine learning algorithm capable of detecting sepsis four to six hours prior to clinical recognition, capable of substantially reducing the untoward effects associated with the condition. In this work, a software platform was developed that consumes live patient data, securely transports it into a cloud environment, and interprets it in real-time. Our approach leverages the benefits of cloud-based managed services that are scalable and fault tolerant. Though there are several pathways for extracting live data from electronic health records (EHR), the AIDEx platform proposed in this work is an EHR vendor-agnostic open-source solution that can be easily deployed in any clinical environments.

An Scalable and Interoperable Real-time Software Platform for Forecasting the Onset-time of Sepsis

By

Fatemeh Amrollahi
B.Sc., Amirkabir University, Iran, 2015

Advisor: Shamim Nemati, Ph.D.

A dissertation submitted to the Faculty of the
James T. Laney School of Graduate Studies of Emory University
in partial fulfillment of the requirements for the degree of
Master of Science
in Computer Science and Informatics
2019

Acknowledgments

Firstly, I would like to express my sincere gratitude to my advisor Professor Shamim Nemati for the continuous support of my Master study and related research, for his motivation, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis.

My sincere thanks also goes to Professor Ashish Sharma and Dr. Christopher Josef, for their insightful comments, stimulating discussions, and encouragement during this research. Without their precious support it would not be possible to conduct this research.

I would like to thank the rest of my thesis committee: Professor Babak mahmoudi, for his comments which helped me to widen my research from various perspectives.

Last but not the least; I would like to thank my father for supporting me spiritually in my life.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Sepsis : A Health Crisis

Sepsis is a syndromic, life-threatening condition that occurs when the body exerts an exaggerated response to infection and begins injuring its own internal organs [6]. Nearly 6% of the inpatient hospital population in the United States will carry a diagnosis of sepsis during their stay, incurring an aggregate hospital cost of $23.7 billion (USD) [5]. Even with the progress of diagnosis of sepsis, still 20% of septic patients progressed to deadly sever sepsis and sepsis shock.[8] When all hospital deaths are ultimately considered nearly 35% are attributable to sepsis, and interestingly, this extremely deadly condition lacks the notoriety of heart attacks which have a mortality rate of 2.7-9.6% and only cost the US $12.1 billion annually, roughly half of the cost of sepsis [5, 23].

In 2004 the Surviving Sepsis Campaign (SSC) promulgate the first of their many recommendations regarding treatment regimens for sepsis and septic shock in the form of consolidated evidence-based practice guidelines called sepsis bundles [8]. Numerous trials have demonstrated that outcomes for sepsis can be dramatically improved by early recognition of the condition and rapid treatment [1, 16, 3, 7]. The most recent recommendation from the SSC is a 1-hr bundle that in addition to obtaining diagnostic tests like cultures and lactate levels prescribes standard treatment with broad-spectrum antibiotics, IV fluid, and vasoactive drugs if necessary all within an hour of a sepsis diagnosis [12].

While there are effective protocols for treating sepsis once it has been diagnosed there still exists challenges in reliably identifying septic patients early in their course owing to the significant variability in the diseases presentation and etiologies.At present the Sepsis-3 guidelines have narrowed the constellation of signs and symptoms of sepsis into a definition that can be reliably used by clinicians and researchers to identify this life-threatening condition retrospectively, but this definition cannot be used to identify a patient who is experiencing the untoward effects of sepsis early in the disease's course.

1.2 Sepsis Prediction Algorithm

In recent years, the increased prevalence of electronic medical records (EMR) has spurned the development of machine learning based surveillance tools for detection or classification [9, 17, 18, 10, 20] and prediction [9, 19, 11, 4] of patients with sepsis or septic shock. However, the real-time implementation of a high-dimensional machine learning model in an ICU environment has not been successfully achieved.

In their recent work, Nemati et al. developed the Artificial Intelligence Sepsis Expert (AISE), a modified Weibull-Cox model that uses an EMR data to predict the onset of sepsis four to six hours in advance with an AUC of 0.85 [19]. The AISE development cohorts contained over 30,000 patients from multiple hospitals in the Emory Healthcare system and was validated using a cohort containing 50,000 patients from the MIMIC-III ICU database [19].

In this work, we developed a software platform, called Artificial Intelligence Decompensation Expert version 1.0 (AIDEx 1.0), to facilitate deployment of the AISE algorithm in a live clinical environment. This novel architecture and the associated user interface (UI) addresses the low clinical tolerance for false-alarms [11] as well as the interpretability and workflow integration requirements necessary for successful clinical decision support (CDS) systems [2, 22, 13]. The AISE algorithm, in conjunction with the AIDEx platform, was designed to make real-time predictions about who is or will become septic at virtually any institution with a modern EMR, affording clinicians the ability to intervene early and drastically reduce the impact of this deadly condition. In this work, we describe our containerize architecture that fetches patients records from a real-time EMR database and displays hourly sepsis risk score for each patient, in addition to the top contributing factors to the risk scores.

Chapter 2

A Real-time Software Platform for Forecasting the Onset of Sepsis

Sepsis, a deadly privilege disease in ICU,is cussed by extreme immune system host response to infection. Several studies have shown that early detection of sepsis and initiation antibiotics significantly reduce the risk of developing sever sepsis and greatly influence the survival outcome among patients in ICUs. Recent works on EMRs and laboratory results for clustering and predicting sepsis is promising, but most of existing works are limited to the static or low resolution dynamic features.Nemati et al. have developed a real-time, high-dimensional machine learning algorithm capable of detecting sepsis four to six hour ahead. In this work, we developed a platform to that consume live patient data, computes sepsis risk scores, and presents the population level surveillance in an interpretable manner. Our platform is capable of substantially reducing the untoward effects associated with the sepsis condition.

We integrated our platform with multiple source of live data, in our last development we consume data as FHIR resources and call it AIDEx0.1. AIDEx is capable of consuming live patient data, securely transport it into a cloud environment, and interpret it in real-time. In our other deployment of our platform we extract live data from ICU at Emory hospital we called this project SepsisApp on Tele-ICU. Though there are several pathways for extracting live data from electronic health records (EHR), our platform is an EHR vendor-agnostic open-source solution that can be easily deployed in every clinical environments. Our platform consumes patient data, computes sepsis scores, and presents patient information along with their sepsis risk score in an interpretable manner via a web-based dashboard. It is modular and can be deployed in any host machine with installed Linux operating system and Docker.

2.1 AIDEx: A FHIR Based Real-time Software Platform for Forecasting the Onset of Sepsis

2.1.1 Overview

AIDEx is a software platform that consumes patient data as a series of FHIR resources, computes sepsis scores, and presents them in an interpretable manner via a web-based dashboard. It is modular and consists of four containerized microservices that are deployed on the Google Cloud Platform (GCP) and orchestrated using the Google Kubernetes Engine (GKE). AIDEx does not include a FHIR server and instead it relies on a GCP managed service called the Cloud Healthcare API to provide patient data as FHIR resources. The environment is secured via a virtual private cloud and utilities have been deployed to push data from the institution to the cloud. The use of containerized microservices and GKE removes the need to install distinct applications and their associated dependencies on a host machine at various deployment sites. It also allows us to leverage the inherent scalability and fault tolerance that stems from the use of GCP and GKE.

The Healthcare API and its FHIR interfaces is the only services that is unique to GCP and when moving to a different cloud, we would leverage, or deploy our own, FHIR server. The AIDEx architecture however is inherently vendor neutral and relies on technologies that are available across all cloud vendors and is easy to replicate on-prem hardware. It also provides a common pattern that can be used in other applications that consume patient data in real-time and are predicting various clinical decompensations.

Google Cloud FHIR instance and Cloud Healthcare APIs

Fast Healthcare Interoperability Resources (FHIR) is a Interoperability healthcare standard from Health Level Seven International (HL7)for transferring, sharing and exchanging electronic health records (EHRs). Google cloud provides a FHIR store inside the datasets. FHIR stores specifies a set of FHIR resources to meet the data model demands of healthcare providers. FHIR provides support to model the majority of clinical data and billing use cases. Furthermore, FHIR implementation can be extended to meet additional clinical or organizational needs. FHIR in Google could is a 'RESTful' specification. This RESTful FHIR framework, performed transactions directly on the FHIR server using an HTTP request/response. The API describes the FHIR resources as a set of interactions on resources. Each resource on FHIR is an instance that is managed in collections by its type. The Cloud Healthcare APIs used by AIDEx provides full support for performing every valid operation on FHIR STU3 resources.

Kubernetes Deployment on Google Cloud

Kubernetes is a powerful software platform to manage containerized applications across multiple virtual or physical machines. A single Kubernetes cluster is comprised of multiple nodes, with each node representing a physical or virtual machine (VM). In the Google Cloud deployment of the AIDEx pipeline, a node represents an individual VM with a set of CPUs and memory resources. The AIDEx pipeline reserves three nodes which employs a total of three CPUs and 11 gigabytes (GB) of memory. Though each node can perform its own computational operations, the Kubernetes cluster manages the nodes as a single entity. A Kubernetes cluster creates a single powerful machine by pooling all the nodes resources and intelligently distributing them as they are needed to run instances of containerized applications.

Use of Kubernetes Pods and Docker Containers

Each of the containerized AIDEx microservices are wrapped in a high-level structure called a pod. A pod is a Kubernetes unit that is comprised of one or more containerized applications. The AIDEx pipeline is comprised of four distinct but interconnected services, and it uses Docker containers to isolate and run these services and their required dependencies. Figure2.1 provides a detailed diagram of different services of the AIDEx pipeline and shows how these services communicate. The use of containerized services via Docker entirely averts the issue of dependency conflicts and the need for customization that arises with a native installation of applications on multiple host machines. AIDEx is fault tolerant and in the event of service failure Kubernetes automatically recovers the service by instantiating one or more pods that host that service through the Kubernetes service manager. The Kubernetes cluster running the AIDEx pipeline was configured to instantiate with two replicas of the AIDEx Algorithm pod (which runs the machine learning algorithm that computes a sepsis risk score) and three replicas of AIDEx Server pod (which provides the front facing user interface). These replicas allow for efficient load balancing and scaling of the pipeline. All the pods created and used by the AIDEx pipeline are isolated from the outside world, prohibiting direct unauthorized interaction with these services. To communicate with a service running inside of a pod a channel must be opened using ingress; the Kubernetes network traffic controller. The AIDEx pipeline uses Load Balancer to add ingress for the UI service. Conversely, communication within the

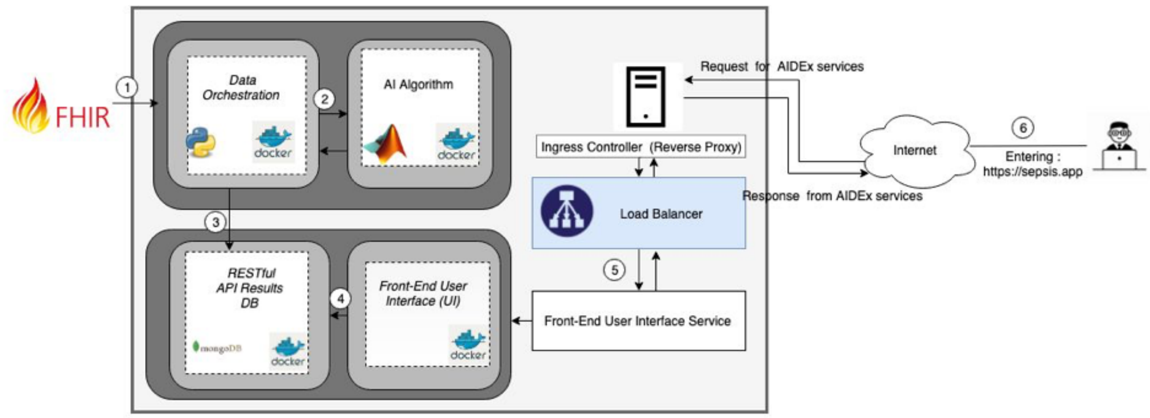

Figure 2.1: **AIDEx1.0 Architecture.** 1) Patient records (e.g. temperature, heart rate, etc.) are collected and prepared by the Data Wrangler 2) Two input files containing clean patient data are passed to the Deep AISE algorithm and a sepsis risk score is created and appended to the file 3) The sepsis risk score and patient features are saved to JSON files and stored in the ResultsDB 4) The user interface service makes an API call to acquire the necessary patient data from ResultsDB 5) The Kubernetes load balancer assess demand and handle the distribution of the workload across front-end user interface services 6) A clinician logs into the website and a request for patient data is made.

Kubernetes cluster is allowed to occur with very little restriction, making it very easy for one pod to share information with other pods on the cluster.

2.1.2 The AIDEx 1.0 Services

The AIDEx platform takes real-time patient data on hourly basis from an FHIR server and computes an interpretable risk score for sepsis that a clinician can access via a web page. The AIDEx pipeline is unique in its health system agnostic design and its use of a state-of-the-art machine learning algorithm capable of accurately identifying patients with sepsis early in their clinical course. Though this tool provides population-level surveillance of a large cohort of ICU patients, its real strength lays in its ability to provide clinicians with individual patient vital sign trends and features contributing to their risk score. Table 2.1 presents a summary of the services comprising the pipeline. In the following section, the functioning of each service is described in more details.

Data Orchestrator acquires live data from the FHIR store, prepares it for use by the algorithm, and stores calculations in the ResultsDB. ResultsDB Data Warehouse is a mongo data base to hold patient data and algorithm outputs. The data base accessible via a RESTful APIs.

Table 2.1: The Four Core Services of AIDEx1.0

Kubernetes Pod Name	Kubernetes Pod Function
Data Orchestrator	Acquires live data from the FHIR store, prepares it for use by the algorithm, and stores calculations in the ResultsDB
ResultsDB- Data Warehouse	A Mongo data base to hold patient data and algorithm outputs. The data base accessible via a RESTful APIs
Algorithm	Performs sepsis risk calculations with data provided by the Data Orchestrator
Server	Hosts and serves the interface via a webpage

1.Data Orchestrator - Clinical Data Harmonization

The Data Wrangler performs several distinct tasks, but its overall function is to shepherd real-time patient data from the Google cloud FHIR server instance to the Algorithm, and finally to the RESTful API mongo DB where it is stored. Figure2.2 illustrates the data flow diagram of the AIDEx pipeline. The Data Wrangler service begins its work by querying a live EMR FHIR database capturing the active patient whose observation modified and/or added to the FHIR store during the last hours. After capturing active patients from FHIR server patient recourse, the AIDEx queries the FHIR observation recourse for the clinical and labratory results of active patients over the last hour that are necessary for sepsis prediction under five core topics. Data arriving from an active EMR is not always ready for use by a machine learning algorithm and requires a series of pre-processing steps. The Data Wrangler begins by ensuring that each feature has a timestamp and an associated patient ID number. If a feature is missing or its validity cannot be ensured the value is subsequently discarded. Similarly, any duplicate values for a given timestamp are similarly discarded. Sometimes errors with data entry result in values that are not physiologically plausible. To minimize the impact of erroneous data, all extreme values are limited to a maximum and minimum value based upon the 95% confidence interval for each feature obtained from a pre-collected patient cohort. Following this pre-processing step, the Data Orchestrator service then makes an API call to the Deep AISE Algorithm service. Active patients prepared and standardized data are then transmitted from the Data Orchestrator service to the AISE Algorithm service as two input files.

The AISE Algorithm service runs a MATLAB executable that performs the risk calculation and returns an individual patients predicted sepsis risk score appended to the patients demographic input file in addition to producing the three most important factors contributing to the risk score. Then Data Orchestrator services final function is to provide a standardized interface for reading and writing data as JavaScript Object Notation (JSON) documents to the RESTful API ResultsDB. Each input file that has been created is converted into JSON documents corresponding to the timestamps of the following patient features: laboratory results and vital signs. Each JSON document contains timestamp and correspondence patient features, sepsis risk score, change in risk score over the last four hours, demographic information, and the three factors contribute the most to the high sepsis risk score. The front-end UI Server will later call these JSON documents and display the data within as part of a graphic user interface for interpretation by clinical team members.

2.Sepsis Prediction Algorithm

The AISE Algorithm Service accepts as its input two input files (a demographics file and a dynamic vital signs and laboratory file) which constitute all the input features to the AISE Algorithm [20]. When deployed the algorithm can alert clinicians 4-6

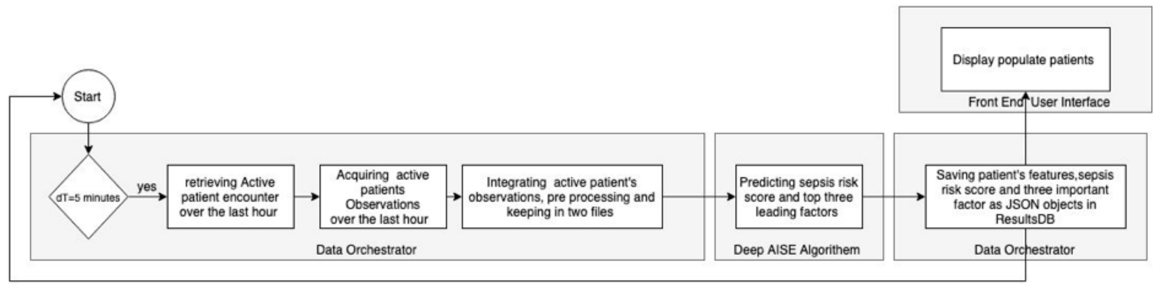

Figure 2.2: Sepsis risk score prediction platform flow diagram

hours before a patient meets the Sepsis-3 criteria. The output of the AISE Algorithm service is a sepsis risk score and the top three factors contributing to the sepsis risk score (see Fig. 3). The data returned by AISE Algorithm service along with all patient features are combined into JSON documents to be stored in the Data Warehouse. The Inputs of the AISE model and the machine learning model is explained further with more details.

Features used by AISE and the AISE machine learning Model

AISE algorithm used high resolution HR and BP time series with 2 second resolution through Bed Master System from all ICU patients 18 years old or older within Emory hospital and public ICU available data sets. Patients were followed through their ICU stay until they get discharged or developing sepsis according to Third International Consensus Definition Sepsis-3. Data orchestrator provides AISE algorithm on hourly basis with two input files. These files keep records of real-time High-resolution and dynamical features, laboratory results updated over the last hour, clinical features, and demographical features. Table2.2 presents the lists of all features used as inputs for predicting sepsis risk score by the AISE algorithm.

The AISE algorithm accurately predict the onset of Sepsis based on Sepsis-3 definition 4-12 hours prior to clinical recognition. We choose the AISE algorithm for our software pipeline because of its ability to adopt the dynamic of real-time high resolution EMR data such as HR and BP recorded and comes to our pipeline from

Bed Master system every hour. The AISE algorithm also outperforms other existing models in terms of predicting accurately the risk of sepsis with AUCROC 85% [20]. In AISE algorithm and based on Sepsis-3 definition the time of suspicious (t suspicious) is defined as earlier time stamp of sampling blood culture within 24 hours after initiation antibiotics or the initiation of antibiotics within 72 hours after sampling blood cultures for an infected patient. The onset time of sepsis (t sepsis) is then defined when there are 2 or more points change in a Sequential Organ Failure Assessment (SOFA score) up to 24 hours after suspicious time or 12 hours before suspicious time. AISE considered the onset time of sepsis as a minimum of suspicious time and time of sepsis. The AISE algorithm used modified Weibull-cox proportional hazards machine learning model to predict the sepsis time 4 hours prior to clinician recognition. Appendix A provides a more details for machine learning model used by AISE.

3.Data Warehouse - RESTful API Mongo DB

The AIDEx pipeline is designed to be scalable and capable of managing data streams from a large patient population. The patient data streams and the computed sepsis scores are transformed into a time-series JSON document. This JSON document captures the state of the document, at a specific time point, and includes the demographic information, clinical and lab values and the sepsis scores. These documents are stored in MongoDB a well know NoSQL document store that is highly scalable and has been used in a variety of clinical and research applications. The database is accessed via a REST API that is built using an OSGI based declarative middleware called Bindaas [21**? ?**]. Bindaas is a domain agnostic, standards-based, extensible, and easy to use middleware environment that can be used by a data provider to rapidly create and deploy a secure web service with a REST API. Other pods in the AIDEx pipeline,including the UI web application access MongoDB via this API.

Table 2.2: list of all features used by AISE algorithm

List of all features	
High-resolution and dynamical features	Standard deviation of RR intervals and MAP (RRSTD and MAPSTD), average multiscale entropy of RR and MAP(HRV1 and BPV1), and average multiscale conditional entropy of RR and MAP (HRV2 and MAPv2)
Laboratory Results	White blood count (WBC), Hemoglobin, Hematocrit, creatinine, Bilirubin and Bilirubin direct, platelets, INR,partial prothrombin time (PTT), Aspartate Aminotransferase (AST), Alkaline Phosphate, lactate, glucose, potassium, calcium, Blood urea nitrogen (BUN), phosphate, magnesium, chloride, B type Natriuretic Peptide (BNP), troponin, fibrinogen, pH, paCO2, HCO3, BaseExcess, SaO2, CRP, Sedimentation Rate, Ammonia (the last 3 one is not in our data set)
Clinical features	Mean arterial blood pressure (MBP),heart rate (HR), O2Sat , systolic blood pressure(SBP), diastolic blood pressure (DBP),respiratory rate, Temperature, Glasgow coma scale (GCS), partial pressure of arterial oxygen (Pao2), fraction of Inspired o2 (FIO2)
Demographic information	Care unit, surgery in the past 12 hours, Wound class, Surgical Specialty, Number of antibiotics in the past 12, 24 and 48 hours. Age, Charleston Comorbidity index (CCI), Mechanical Ventilation, maximum sofa score changes over the past 6 hours

4.Front End User Interface

One of the most common criticisms leveled against machine learning algorithms deployed in healthcare environments is their black box nature. The AIDEx pipeline addresses this issue through an interpretable, clinician-friendly display of patient data via the front-end UI Server. As seen in Fig. 3, the front-end UI includes a command center that gives a high-level overview of the ICU population and detailed view that presents detailed information including sepsis scores, clinical interpretations and vital signs. The default view is seen in Fig.2.3a demonstrates a population-level view that surveils all included ICU patients. Each patient is represented by a single card, and the front of each card contains the patients room number at the top, a sepsis score, a discharge readiness score, and finally a directional arrow with magnitude representing the acceleration (i.e. delta) of a patients sepsis risk score over the last four hours. The patient list is ranked according to the sepsis risk score with the most acute patients at the top of the list and a second UI see in Figure 3b is revealed. This patient-centric view reveals the top three factors contributing to the risk score in addition to the vital sign trends for the patient over the last 24 hours. The UI is a web page that was built using the popular JavaScript framework called Angular and Node.js. As previously described the Data Warehouse stores patient features in addition to the AISE Algorithm outputs in JSON files inside the Data Warehouse. This approach to data storage makes it simple for the node.js code to obtain patient data from the Data Warehouse for display in the user interface.

Figure 2.3: **AIDEx1.0 Server- Front End User Interface**-Fig a) A population view of all ICU patients. Each patient is represented by a single card that displays: a sepsis risk score (AISE), discharge readiness score (DRS), the acceleration of patients sepsis score (delta). Patients are listed descending based on their AISE score Fig. 3b) The detailed view for a single patient is displayed. Visible is the 12-hour trajectory of the patients sepsis risk score, and their vital signs. (A demonstration of this user interface can be found at https://sepsis.app username: demo, password: demo)

2.2 Sepsis App on Tele-ICU: A SQL Based Real-time Software Platform at Emory Hospital for Forecasting the Onset of Sepsis

Here and in this development SepsisApp pipeline consumes live data from ICU at Emory hospital. High-resolution real-time data are collected by BedMaster system which is a third-party integrated software connected to the hospitals General electric monitors. The System acquires and stores waveforms, and vital signs. Acquired data then stored in over 400 tables and archived by Philips team. In this project we need to query and achieve these live EMR records for the last hour and after computing the Sepsis Risk score, store them on our local mongo DB and present the sepsis risk score in eICU for the clinician at Emory Hospital. The stream on Live SepsisApp is similar to AIDEx and have four main services. In this deployment all the services are deployed by docker container and all the containers are managed via docker compose engine. All the services work in same way as AIDEx with small changes in Data Orchestrator. Data Orchestrator here query and capture the data from SQL server on hourly basis with resolution of 5 minutes and after applying some preprocessing steps post data via a local network created by docker-compose engine to a Deep Learning-based predictive algorithm. Below architecture of SepsisApp on eICU is explained with more detail.

2.2.1 Overview of SepsisApp on Tele-ICU

SepsisApp on eICU contains four containers including Data Wrangling, AISE algorithm, MongoDB, and User interface to present to sepsis risk score for care provider. All the services here are deployed via docker images and docker container technology.

Docker containers packages each software into standard units to easily be developed, deployed and transferred in any host machine. Containerized software will always run on the same Docker engine. Furthermore, Docker compose engine is another technology used on SepsisApp on eICU to manage the containers and recover them in the case of each container failure. Docker compose makes the SepsisApp on eICU also fault tolerant. On the other hand, via Docker compose engine we create a local network to connect our containers to each other inside the network and isolate them from outside world. The SepsisApp on eICU takes real-time patient data on hourly basis from a Archive DB. The Data Orchestrator post the acquired data from Archive after some preprocessing steps like what we explained above as two input files to the AISE container and via network bridge created by docker compose. The AISE Algorithm service compute the risk score and returns an individual patients predicted sepsis risk score appended to the patients demographic input file in addition to producing the three most contributing factors to the risk score. In last step, Data Orchestrator provides the RESTful API MongoDB with JSON documents. The Json document structure and MongoDB APIs spec are provided in Appendix B. The front-end UI Server exposed to the internet via port 3001 will later call these JSON documents and display the data to be used and interpret by clinical team members.

Chapter 3

Tests and Statistical methods

AIDEx is a live software platform integrated from four micro services. To evaluate the functionality and quality of the platform application and to identify the defects of the software to ensure the product is defect free we developed and test the system against different software testing strategy including Unit testing,unit integration testing, and system testing. To perform unit tests, individual units of a software have been tested. As UNIT test is considered as a white box test, We have tested every block of codes, and functionality of each of the methods from micro services. In these series of tests we aimed to validate the each unit of the software. In second level of the tests we performed unit integration testing to ensure that different units of the services are communicating with each other correctly. So through the Unit integration test we evaluate the functionality of each of the micro services, and ensure units working together in a perfect flow. All the APIs including Google Cloud Healthcare API are also tested. In last series of tests we performed system level tests. In this level of the software testing, we evaluate a functionality of a complete integrated system. Here we aim to asses the systems compliance with the specified requirements.Furthermore, system level tests proved that there is no data leakage inside our pipeline. Table 2 presents a summary of some of the tests descriptions and the correspondence results that we have preformed. Sampled synthetic de-identified data over 24 hours from Emory eICU data set is used in different level of the tests. Furthermore,The AIDEx has a centralized logging procedure. Components Logs including python info,error, and warning logs produced by Data Orchestra,AISE algorithm Python flask logs, and Bindaas logs are all accessible via Google Stack driver in this project.

For all continues and dynamic features every 24 hours the statistics of EHR records have been computed and kept. Maximum, minimum, median[25-75 percentile], and standard deviation(STD) of every dynamic features along with number of active patients on every hour have been reported by our Quality Control container (QC container). The QC container is apart from the pipeline and developed to check the data flow and perform the statistics. QC container also perform two-sided Wilcoxon rank sum test every week. Here QC compares two population from current and last week to assess the difference between two population with 95% confidence interval.

Table 3.1: Testing methods and descriptions

Component	Second column	Third column
Data Orchestrator (Healthcare APIs)	Send a patient observation and retrieve it	Successful
Data Orchestrator (Healthcare APIs)	Send 5 patients observations and retrieve them	Successful
Deep_AISE Algorithm and Orchestrator	send only lab values no demographic information (empty demo)	Deep_AISE will return a server error (500) and Orchestrator would not send any

Continued on next page

Table 3.1 – continued from previous page

Component	Test description	Third column
		thing to ReultsDB
Deep_AISE Algorithm and Orchestrator	Set all demographic information to be null except for encounter Here we have lab values for this patient	DeepAISE returns AISE score for the patient with the same encounter. Orchestrator send Json object based on labs with AISE score to the ResultsDB. demographic fields in Json are empty string.
Deep_AISE Algorithm and Orchestrator	send empty file as lab values for a patient with demographic information (empty lab)	Deep_AISE will return a server error (500) and Orchestrator would not send any thing to ReultsDB
Deep_AISE Algorithm and Orchestrator	All lab values for a patient set to be null except for lab date and encounter	DeepAISE returns AISE score for the patient with the same encounter. Orchestrator would not send Json object to the ResultsDB.

Continued on next page

Table 3.1 – continued from previous page

Component	Test description	Third column
Deep_AISE Algorithm and Orchestrator	Send lab values and demographic information for one patient and get the score for this patient	Successful
Deep_AISE Algorithm and Orchestrator	Change the type of one lab values from float to string	AISE container failed to remove this record
ResultsDB (submit single record API)	Submit a single Json file and retrieve it	Successful
ResultsDB (submit single record API)	Submit 5 Json files one by one and retrieve it	Successful
ResultsDB (FindAllTime API) and Orchestrator	Uploading 5 patients lab records and get all of the Json using this API	Successful
ResultsDB (PatientDeatils)	Uploading a patient record and request the ResultsDB for a details of this patient including the AISE score and feature trends	Successfully return the wanted fields for this patient

Chapter 4

Conclusion and Future directions

Early detection and treatment of sepsis is categorically one of the most important interventions that can be taken in a modern ICU.

In this work, we have developed the pipeline to detect and inform clinicians of a patients risk for developing sepsis. The described pipeline is a scalable, generalizable application that is poised to change the way clinicians identify and treat sepsis.

Our future works include integrating Health level seven (HL7) messages with pipeline. HL7 is set of standards for exchanging and transferring EHRs.

Also it would be interesting to build methods of matching patients according to their similarities with previously seen patients (patients-like-me). This may allow for optimizing treatment strategies.

Bibliography

[1] H. Bartimus J. Reif M. J. Schmidt A. K. Venkatesh, U. Avula and E. S. Powell. Time to antibiotics for septic shock: evaluating a proposed performance measure. *Am. J. Emerg. Med., vol. 31, no. 4, pp. 680683*, Apr. 2013.

[2] A. B. T. G. Awry. Sepsis early warning scoring systems: The ideal tool remains elusive! *J Crit Care*, 2018.

[3] A. Rhodes et al. The surviving sepsis campaign bundles and outcome: results from the international multicentre prevalence study on sepsis (the impress study). *Intensive Care Med., vol. 41, no. 9, pp. 16201628*, Sep. 2015.

[4] C. Lin et al. Early diagnosis and prediction of sepsis shock by combining static and dynamic information using convolutional-lstm. *2018 IEEE Int. Conf. Healthc. Inform. ICHI, pp. 219228*, 2018.

[5] C. Rhee et al. Incidence and trends of sepsis in us hospitals using clinical vs claims data, 2009-2014. *JAMA, vol. 318, no. 13, pp. 12411249*, Oct. 2017.

[6] M. Singer et al. The third international consensus definitions for sepsis and septic shock (sepsis-3). *JAMA, vol. 315, no. 8, pp. 801810*, Feb. 2016.

[7] R. Ferrer et al. Empiric antibiotic treatment reduces mortality in severe sepsis and septic shock from the first hour: results from a guideline-based performance improvement program. *Crit. Care Med., vol. 42, no. 8, pp. 17491755*, Aug. 2014.

[8] R. P. Dellinger et al. Surviving sepsis campaign guidelines for management of severe sepsis and septic shock. *Intensive Care Med., vol. 30, no. 4, pp. 536555*, Apr. 2004.

[9] T. Desautels et al. Prediction of sepsis in the intensive care unit with minimal electronic health record data: A machine learning approach. *JMIR Med. Inform., vol. 4, no. 3, p. e28*, Sep. 2016.

[10] K. K. Giuliano. Physiological monitoring for critically ill patients: testing a predictive model for the early detection of sepsis. *Am. J. Crit. Care Off. Publ. Am. Assoc. Crit.-Care Nurses, vol. 16, no. 2, pp. 122130; quiz 131*, Mar. 2007.

[11] P. J. Pronovost K. E. Henry, D. N. Hager and S. Saria. A targeted real-time early warning score (trewscore) for septic shock. *Sci. Transl. Med., vol. 7, no. 299, pp. 299ra122-299ra122*, Aug. 2015.

[12] M. M. Levy and A. Rhodes. The surviving sepsis campaign bundle: 2018 update,. *Crit. Care Med., vol. 46, no. 6, p. 4*, 2018.

[13] J. Norrie. The challenge of implementing ai models in the icu. *Lancet Respir. Med., vol. 6, no. 12, pp. 886888*, Dec. 2018.

[14])]ref22 online. Fielding dissertation: Chapter 5: Representational state transfer (rest). *Available:* $https://www.ics.uci.edu/fielding/pubs/dissertation/rest_arch_style.htm.$, $[Accessed: 25 - Feb - 2019]$.

[15]]ref23 Online. A nosql based research: Pacs deployed on a cloud. *Available: http://archive.rsna.org/2012/12031184.html*, [Accessed: 25-Feb-2019].

[16] J. Pryor M. A. Puskarich S. A. Sterling, W. R. Miller and A. E. Jones. The impact of timing of antibiotics on outcomes in severe sepsis and septic shock:

A systematic review and meta-analysis. *Crit. Care Med., vol. 43, no. 9, pp. 19071915*, Sep. 2015.

[17] Y. Halpern Y. Jernite N. I. Shapiro S. Horng, D. A. Sontag and L. A. Nathanson. Creating an automated trigger for sepsis clinical decision support at emergency department triage using machine learning. *PloS One, vol. 12, no. 4, p. e0174708*, 2017.

[18] K. G. Kuttler R. K. Keddington T. L. Allen S. M. Brown, J. Jones and P. Haug. Prospective evaluation of an automated method to identify patients with severe sepsis or septic shock in the emergency department. *BMC Emerg. Med., vol. 16, no. 1, p. 31*, 2016.

[19] F. Razmi M. D. Stanley G. D. Clifford S. Nemati, A. Holder and T. G. Buchman. An interpretable machine learning model for accurate prediction of sepsis in the icu. *Crit. Care Med., vol. 46, no. 4, pp. 547553*, Apr. 2018.

[20] G. D. Clifford S. P. Shashikumar, Q. Li and S. Nemati. Multiscale network representation of physiological time series for early prediction of sepsis. *Physiol. Meas., vol. 38, no. 12, pp. 22352248*, Nov. 2017.

[21] A. Sharma and Y. Saghar. A web services based system to create and query an on-demand virtualized pacs that integrates dicom imaging and non-imaging data. *Soc. Imaging Inform. Med. SIIM*, 2013.

[22] E. H. Shortliffe and M. J. Seplveda. in the era of artificial intelligence. *p. 2*, 2018.

[23] C. M. Torio and B. J. Moore. National inpatient hospital costs: The most expensive conditions by payer, 2013: Statistical brief 204. *Healthcare Cost and Utilization Project (HCUP) Statistical Briefs, Rockville (MD): Agency for Healthcare Research and Quality (US)*, 2016.