Journal of Community Healthcare and Development
Vol. 1, No. 1 (2020), pp.25-34
http//dx.doi.org/10.33832/jchd.2020.1.1.05

NODE-RED MICROSERVICE BEDSIDE SYSTEM FOR OUTCOME PREDICTION OF PATIENTS WITH SUSPECTED SEPSIS: USEFUL STUDY FOR THE CORONAVIRUS OUTBREAK AND BEYOND (STAGE 1 OF THE WORK)

Jinan Fiaidhi[1*], Sabah Mohammed[1] and Sami Mohammed[2]

[1]Lakehead University, Department of Computer Science,, Canada
[2]University of Victoria, Department of Computer Science, Canada,
[1]jfiaidhi@lakeheadu.ca, [1]mohammed@lakeheadu.ca, [2]smohamme@uviv.ca

Abstract— THE covid-19 virus is humanity's newest enemy, with the potential to prematurely end millions of lives. COVID-19 is out of our control right now and we're going to have to get innovative if we want to catch up with it. To control this new coronavirus, we need to understand it first. To date, the most reliable information on the clinical syndrome resulting from COVID-19 comes from recently published data out of Wuhan, China that was published in the Journal of the American Medical Association (JAMA) on February 7, 2020. The most common symptoms attributed to COVID-19 infection requiring hospitalization were fever, fatigue, and a dry cough. A majority of those cases also had low white blood cell counts and abnormal blood clotting. The patients hospitalized only 26% were sick enough to be treated in an intensive care unit (ICU) – of these, approximately 60% developed respiratory failure and 31% developed septic shock. According to the Global Sepsis Alliance COVID-19 does indeed cause sepsis[1]. Sepsis is "a life-threatening organ dysfunction caused by a dysregulated host response to infection." As there are no proven therapies effective against the virus itself, it is apparent that the best COVID-19 care includes good sepsis care. Early clinical recognition of sepsis can help in reducing the effects of having sever COVID-19 cases. This paper attempt to address the issue of building a bedside infrastructure where the healthcare worker can use it to select existing workflows or creating new workflows for predicting the outcome of new cases and whether these cases will likely to have a sepsis shock. In our next stage we will address the issue of developing effective predictors for sepsis in its early stages and whether it these cases will likely to develop a sepsis shock. The first stage uses the notion of Node-Red microservices to build a bedside Such Node-Red environment is considered as a new visual tool that caregivers can use machine learning from the bedside.

Keywords— **Microservices, Clinical Workflow Interoperability, SEPSIS, Bedside Visual Tools, Node-Red**

1. INTRODUCTION

Sepsis is a major health crisis all over the world and one of the leading causes of death resulting as many as 5.3 million deaths per year globally [1]. Despite the high incidence

Received: January 3, 2020
Reviewed: March 3, 2020
Accepted: April 6, 2020

[1] https://www.global-sepsis-alliance.org/news/2020/4/7/update-can-covid-19-cause-sepsis-explaining-the-relationship-between-the-coronavirus-disease-and-sepsis-cvd-novel-coronavirus

ISSN: 2652-6026 JCHD

of sepsis and the professional critical care society's work in defining the clinical criteria that aid sepsis recognition, the fundamental need for early detection and treatment remains unmet [2]. However, sepsis prediction is a difficult task as sepsis can be triggered by a variety of pathogen caused by variety of infections including viruses, bacteria, fungi or parasites as well as it is sensitive to many other factors like acute illness, underlying chronic diseases, and complications associated with infection. Adjusting for these factors is essential for evaluation of new therapies. The purpose of the present study is to determine variables readily identifiable at the bedside that predict the outcome of patients in intensive care unit (ICU) with sepsis. Identifying those at risk for sepsis and initiating appropriate treatment, prior to any clinical manifestations, would have a significant impact on the overall mortality and cost burden of sepsis [3]. Some medical researchers consider sepsis to have three stages. The first stage is the least severe and usually has symptoms of fever and an increased heart rate. The second stage is more severe and is characterized by symptoms of difficulty breathing and possible organ malfunctions, while the third is the most severe stage (septic shock or severe sepsis) with life-threatening low blood pressure. However, not all researchers agree with these stages.

Development of machine learning algorithms to predict severe sepsis and septic shock and evaluate the impact on clinical practice and patient outcomes are among the most important priorities of several researchers agenda [4, 5]. However, the current machine learning models aiming to predict sepsis from electronic health records (EHR) or datasets do not accurately predicts sepsis due to the heterogeneity of the sepsis conditions despite the emerging importance of machine learning in prognosis and treatment [6]. However, looking at a recent systematic review performed in PubMed, Embase.com and Scopus on studies that uses retrospective sepsis data, individual machine learning models have shown the ability to accurately predict sepsis onset ahead of time [7]. However, these models have not shown wider acceptance and remains as an alternative to traditional bedside sepsis scoring systems. According to [8], there is no clinically validated system exists for real-time prediction of sepsis onset especially those that can work at bedside. According to seminal article published at BMJ by Seneviratne [9] recently, very few of the machine learning algorithms ever make it to the bedside; and even the most technology-literate academic medical centers are not routinely using AI in clinical workflows. Commonly, however, at the emergency (IUC) departments they use pen and paper to a large extent when recording emergency care procedures and measurements. During treatment the patient is the main focus and because of this, recording of measurements done could be delayed or in worst case forgotten during stressful situations. The new face of healthcare needs to be defined by a confluence between technical complexity and the healthcare system workflow fabric in which the technical complexity is embedded [10]. This new face should include an ecosystem that involve pragmatic interoperability that focus on workflows and their integration [11]. The goal of this paper is to derive a classification to sepsis that can work at the bedside and be integrated with the dynamics of the clinical workflow of IUC departments using prospective datasets.

2. EMERGENCY DEPARTMENT (BEDSIDE) ECOSYSTEM CARE PLANNING

There are many demands placed on staff working in emergency departments such as the currently witnessed coronavirus outbreak, overcrowding, bed shortages and long waiting times for patients. Despite these demands clinical care needs to be carefully assessed, planned and documented [12]. Although the emerging technologies provided variety of medical sensors to enable most of the work to be passive, still we need some active role to assess new cases and select the relevant care plan. The interoperability of between what is passive and what is active is another challenging issue that need to be solved. Among successful models that can enforce such interoperability is the goal that could be achieved by different approaches and among them the federated the Joint Directorate Laboratories (JDL) model which was researched in 1993 to deal with multi-level data fusion interoperability and integration, primarily was for military systems [13]. The idea was that you could look at the state of the world considering different entities as "atomic units" of your "World View". Level 1 Fusion, for example, was called "Object Refinement" and was focused on fusing multiple heterogeneous data sources to obtain information about individual objects (e.g., people, vehicles, buildings, etc.). So-called Higher-Level Fusion dealt with Levels 2-4, which were named "Situation Refinement", "Threat Refinement", and "Process Refinement". Note, there is a big conceptual jump from Level 1 to Level 2. Level 1 consists of real, measurable objects that exist in time and space. The other levels are concepts and exist in "linguistic space" instead of physical space. In 1998, the JDL model was upgraded to more general, less militaristic language. There was also a "Level 0" added. So the levels now were "Sub-Object Assessment", "Object Assessment", "Situational Assessment", "Impact Assessment", and "Process Refinement". Sub-Object Assessment referred to using data to resolve things smaller than individual objects. So it could be an arm, or it could refer to integrating ("fusing") consecutive reflected pulses from a radar to form a signal [14]. In 2015, a researcher [15], extended the four levels JDL model into five level one where Level 5: "User Refinement" involves humans to be "in the loop" affecting the products of all the lower levels.

However, the increasing amount of information available for business planning motivates the adoption of machine learning methods for addressing specific situation assessment or risk. A special focus needs to be placed on prognosis, namely, the capability to estimate and anticipate events of interest regarding assets and production processes. There lies indeed at the core challenge of business modeling and planning from a data science perspective: data-driven prognostic approaches aim at predicting when an abnormal behavior is likely to arise within the monitored process, providing further insights such as its severity and impact on the business performance. For this reason, it becomes particularly interesting to characterize normality properly towards unveiling degradation patterns or trends. Thus, adding a sixth level to JDL to include learning analytics response is one of the most important direction for business modeling and planning. The idea of having the sixth level (L6.2) is to characterize the business values (including user care values and workflows) through qualitative measures (aka sound care plans) as well as to identify behavioral patterns of interest on the basis of the data monitored from the application process by means of quantitative analytics and machine learning models. This acquired knowledge (valued and thick data as well as quantitative indicators) can be then used to tackle a wide variety of planning problems, including focused prediction, classification and anomaly detection, among others. Furthermore, since data and services are the most important asset that a business owns, and in today’s climate of data security and stewardship it’s more important than ever to trace how that data was produced and the journey it has undertaken through the workflow of services leading to its present state. What we need in this direction is a strong data and services governance platform. This platform represents another architectural layer (L6.1) to the JDL model where the layout of the system is developed as a suite of workflows of small services, each running in its own process and communicating with lightweight mechanisms using a special workflow server. The benefits of level 6.1 are many, ranging from an increase in development productivity, to better business-IT alignment, agility, scalability, and technology flexibility. Figure 1 illustrates our overall bedside care planning framework (BCP) that can be used for predicting outcomes of several of the frequent emergency department cases as defined by the chief complaints [16] to be the grouped into 11 categories: respiratory, gastrointestinal (GI), undifferentiated infection (UDI), influenza-like illness (ILI), lymphatic, skin, neurological, pain, dental, alcohol and musculoskeletal syndromes. These categories were used by other healthcare surveillance programs with slight modification. Chief complaints that could not be grouped into the above 11 syndromes should assigned to "unclassified".

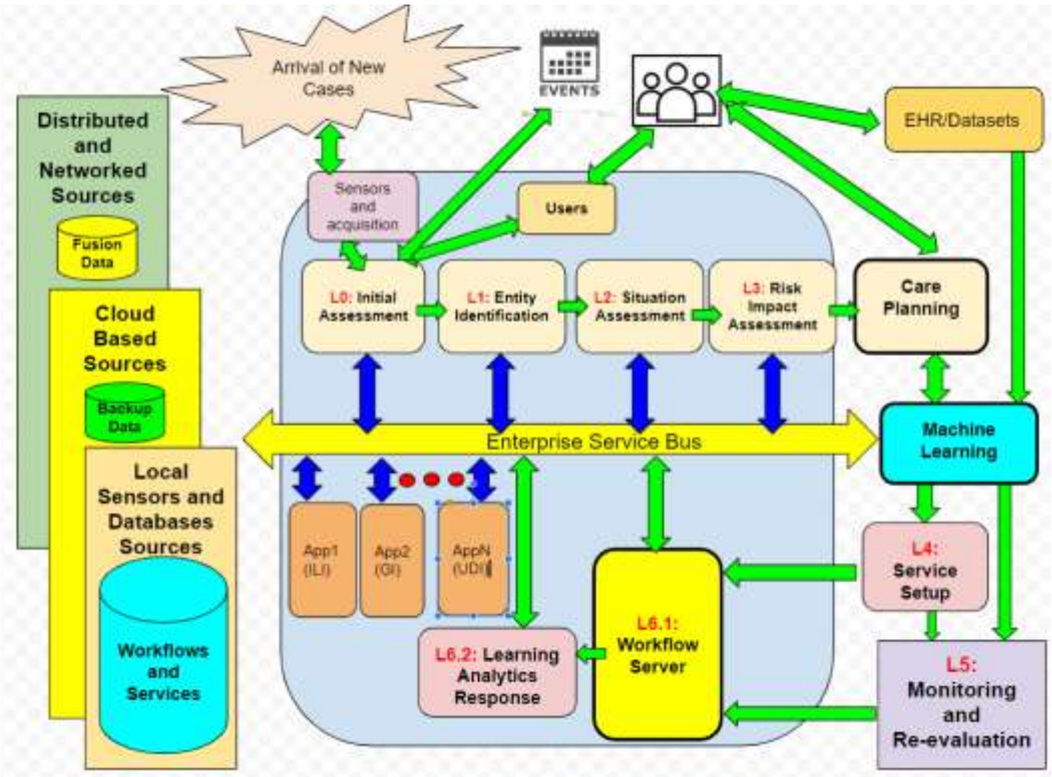


Fig. 1 Bedside Care Planning involving Machine Learning Framework (BCP)

However, BCP defines the infrastructural layer (upper layer) of our architecture. The lower layer that deals with actual business implementation, this is the workflow DSL and the corresponding step implementations. The workflow layer is essentially a highly abstracted version of the ***decode*** **->** ***handle*** **->** ***interpret*** pipeline which defines the different microservices of the applications workflows. This architecture is inspired by the Helland's design [17]. The workflow server takes this pipeline and separates concerns even further by drawing service boundaries between each of these operations

- Workflow Triggers (decode)
- Workflow Executor (handle)
- Workflow Response (interpret)

The execution pattern of a microservice application at the workflow server can be described as a time sequence of Web service invocations. Each web service is categorized into either a trigger or an action in WoT architecture [18]. A trigger is either a publication of some information or a signal that an action (actuation) took place. An action is a task to be executed whenever a trigger is fired. In this approach, instead of producing Apps as a complete runtime artifact, developers can design intermediary artifacts, as accessible controls parts for typical transportation tasks. Using these controls, transporters are enabled to construct their own tasks autonomously by using resources of their own choice from across the WoT. The microservice based transportation system translates the time sequence of trigger and action executions to a time sequence of network flows. A network flow is a traffic information between two communicating endpoints. Zapier and IFTTT or many other alternatives makes this task easy. It lets transporters to integrate everyday apps (*e.g.*, existing legacy app or a newly created workflow) and automate the transportation business processes. Figure 2 illustrates the idea behind using WoT Microservice platform.

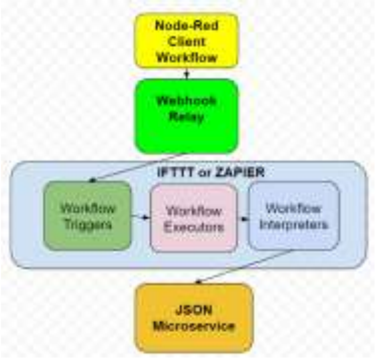
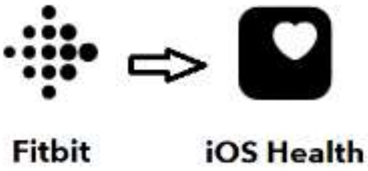
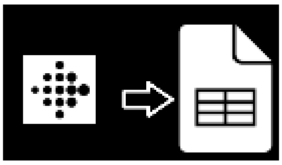


Fig. 2 The layout of Node-Red Microservice

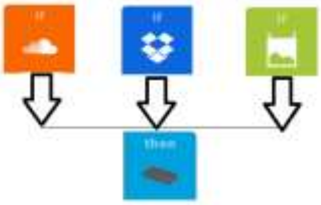
Using the IFTTT platform, the "IF THIS" keyword is used to identify the triggers and "THEN THAT" is used to identify the execution and interpretation part. The trigger is activated by changes that occur within other web services such as Fitbit. Upon activation you may assign an action like pushing a notification to your iOS Health app:



WoT platform may create a new microservice workflow for logging your complete Fitbit data in a spreadsheet:



Moreover, we can add branching logic to microservices to run different actions based on different conditions (*e.g.*, if A happens in your trigger app, then do X. If B happens, then do Y, and so on). Tools like Paths can be used with Zapier to add such branching logic and produce a workflow of microservices:



WoT platforms like IFTTT or Zapier, however, may represent decent workflow composition tool to get started on simple point-to-point integrations but what should you do when you want to tackle higher complexity that often arises when we are dealing with healthcare workflows? Fast-growing healthcare enterprises need to incorporate higher level frameworks to accommodate the various complex scenarios. In this direction we are experimenting with more sophisticated workflow wrappers like the Node-RED. With Node-Red, healthcare workers can produce fully customizable workflows with flexible connector operators such as loops, data storage, array mapping, branching, and if/then conditionals and many more. Node-RED[2] is a programming tool used for wiring together hardware devices, APIs, functions and online services by representing them as nodes. One can use his mobile device to hook to such services via creating a bucket through webhook relay[3] and installing node-red on the mobile phone (*e.g.*, Android[4]). Figure 3 illustrates our overall microservice to classify new cases arriving at an emergency department as SEPSIS or not.

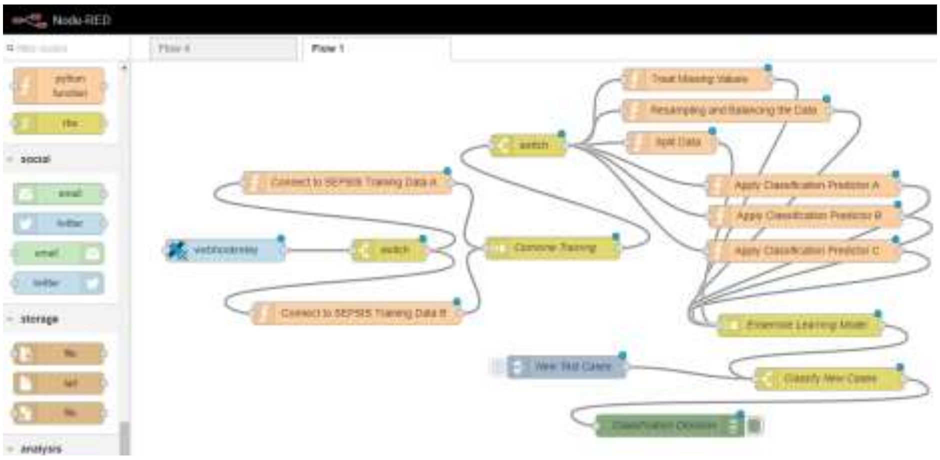
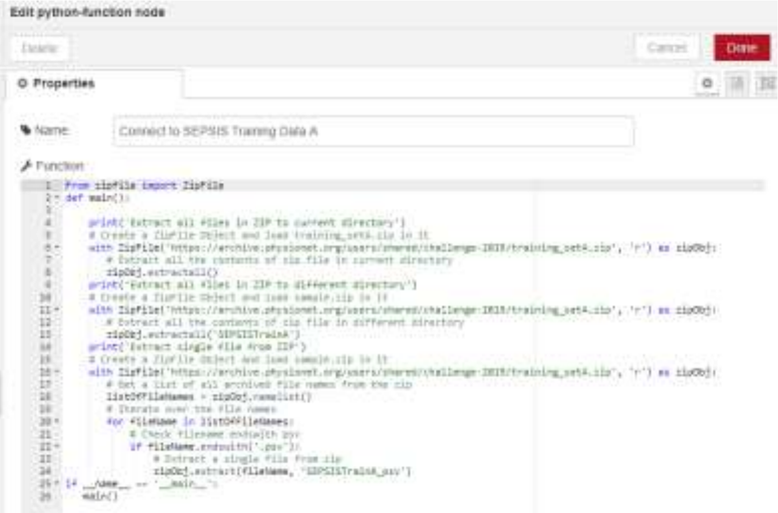


Fig. 3 Node-Red Microservice Workflow for Classifying New Cases for SEPSIS

All what we need after we define our workflow is to define the important functions that contributes to this microserviceSince the task of this paper is find best predictors for SEPSIS, we decided to use Python as our Node-Red functions programming. For this we will need to add a Python function node tour Node-Red dashboard. For this we will need to add a Python shell through the executing the following command:

npm install node-red-contrib-pythonshell

Figure 4 illustrate an example of the python function for reading one SEPSIS training data from the PhysioNet link.



Name: Connect to SEPSIS Training Data A

Function

```
from zipfile import ZipFile
def main():

    print('Extract all files in ZIP to current directory')
    # Create a ZipFile Object and load training_setA.zip in it
    with ZipFile('https://archive.physionet.org/users/shared/challenge-2019/training_setA.zip', 'r') as zipObj:
       # Extract all the contents of zip file in current directory
       zipObj.extractall()
    print('Extract all files in ZIP to different directory')
    # Create a ZipFile Object and load sample.zip in it
    with ZipFile('https://archive.physionet.org/users/shared/challenge-2019/training_setA.zip', 'r') as zipObj:
       # Extract all the contents of zip file in different directory
       zipObj.extractall('SEPSISTrainA')
    print('Extract single file from ZIP')
    # Create a ZipFile Object and load sample.zip in it
    with ZipFile('https://archive.physionet.org/users/shared/challenge-2019/training_setA.zip', 'r') as zipObj:
       # Get a list of all archived file names from the zip
       listOfFileNames = zipObj.namelist()
       # Iterate over the file names
       for fileName in listOfFileNames:
           # Check filename endswith psv
           if fileName.endswith('.psv'):
               # Extract a single file from zip
               zipObj.extract(fileName, 'SEPSISTrainA_psv')
if __name__ == '__main__':
   main()
```

Fig. 4 A Python Function in Node-Red for Extracting Zipped Data.

[2] https://nodered.org/

[3] https://webhookrelay.com/

[4] https://nodered.org/docs/getting-started/android

The overall architecture of our bedside predictive system based on the notion of Node-Red microservices can be represented in Figure 5.

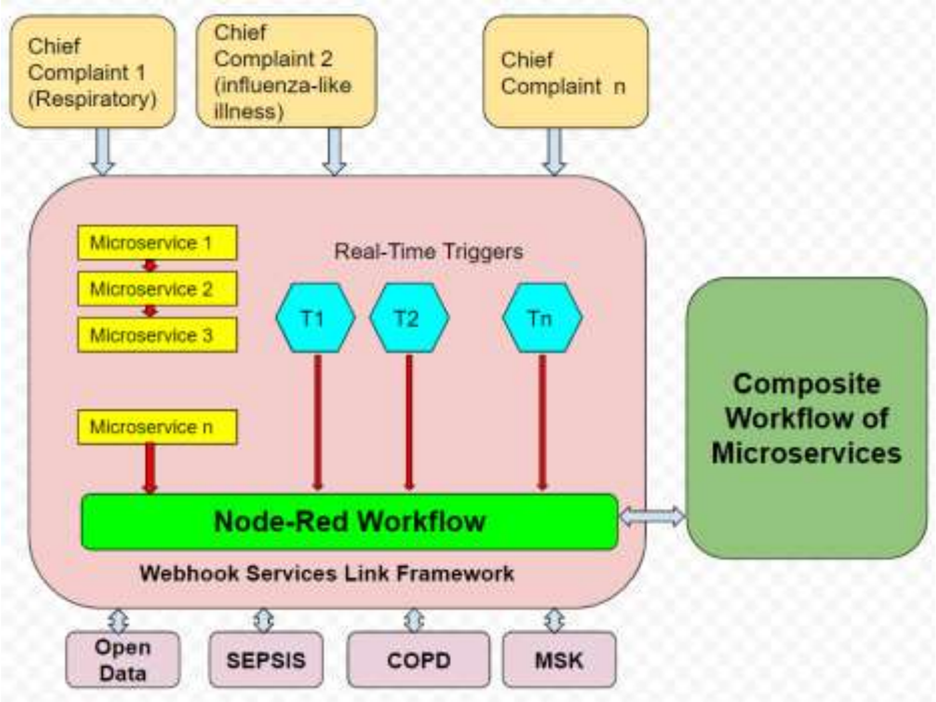

Chief Complaint 1 (Respiratory)
Chief Complaint 2 (influenza-like illness)
Chief Complaint n
Microservice 1
Microservice 2
Microservice 3
Microservice n
Real-Time Triggers
T1
T2
Tn
Composite Workflow of Microservices
Node-Red Workflow
Webhook Services Link Framework
Open Data
SEPSIS
COPD
MSK

Fig. 5 The Envisioned Emergency Department Node-Red Microservices Architecture

3. PREPARING FOR DEVELOPING EFFECTIVE PREDICTORS FOR SEPIS AND SEPSIS SHOCKS:

Building effective predictors requires the availability of relevant datasets. In this direction, there are good number of SEPSIS datasets and machine learning classification models that can help in mission to build such models and incorporates them at our Node-Red bedside visual environment. Table I lists some of the notable SEPSIS datasets that we can use in stage 2 of this research.

Table I. List of Notable SEPSIS Datasets

SEPSIS Dataset	Reference Link
Pediatric Hospital Inpatient Sepsis	https://healthdata.gov/dataset/pediatric-hospital-inpatient-sepsis-performance-measures-facility-2015-and-2016
Adult Hospital Inpatient Sepsis	https://healthdata.gov/dataset/adult-hospital-inpatient-sepsis-performance-measures-and-mortality-facility-2015-and-2016
NICU Infant Sepsis	https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6386402/bin/pone.0212665.s002.docx
Multicentre UCSF Sepsis	https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5829820/
Mimic-III	https://mimic.physionet.org/
Biopharma Sepsis	https://github.com/prise6/aVirtualTwins/wiki/03-Sepsis-Dataset
PhysioNet 2019 Sepsis Challenge	https://archive.physionet.org/users/shared/challenge-2019/

However, for building a robust predictors we are suggesting to start training the various likely predictors on the PhysioNet 2019 challenge on Sepsis dataset as it is the most recent and comprehensive data extracted from patients at the bedside which include 1361672 patients' records with 41 attributes. Table II lists the PhysioNet Sepis dataset attributes.

Table II. PhysioNet 2019 Sesis Challenge Dataset Attributes

Vital Signs (Col 1-8)	Lab Values (Col 9-34)	Demographic (Col 35-40)	Sepsis Outcome (Col 41)
HR: Heart rate (beats per minute) **O2Sat:** Pulse oximetry (%) **Temp: Temperature (**Deg C**)** **SBP:** Systolic BP (mm Hg) **MAP:** Mean arterial pressure (mm Hg) **DBP:** Diastolic BP (mm Hg) **Resp:** Respiration rate (breaths per minute) **EtCO2:** End tidal carbon dioxide (mm Hg)	**BaseExcess:** Measure of excess bicarbonate (mmol/L) **HCO3:** Bicarbonate (mmol/L) **FiO2:** Fraction of inspired oxygen (%) **pH:** N/A **PaCO2:** Partial pressure of carbon dioxide from arterial blood (mm Hg) **SaO2:** Oxygen saturation from arterial blood (%) **AST:** Aspartate transaminase (IU/L) **BUN:** Blood urea nitrogen (mg/dL) **Alkalinephos:** Alkaline phosphatase (IU/L) **Calcium:** (mg/dL) **Chloride:** (mmol/L) **Creatinine:** (mg/dL) **Bilirubin_direct:** Bilirubin direct (mg/dL) **Glucose Serum glucose:** (mg/dL) **Lactate Lactic acid:** (mg/dL) **Magnesium:** (mmol/dL) **Phosphate:** (mg/dL) **Potassium:** (mmol/L) **Bilirubin_total:** Total bilirubin (mg/dL) **TroponinI:** Troponin I (ng/mL) **Hct:** Hematocrit (%) **Hgb:** Hemoglobin (g/dL) **PTT:** partial thromboplastin time (seconds) **WBC:** Leukocyte count (count*10^3/µL) **Fibrinogen:** (mg/dL) **Platelets:** (count*10^3/µL)	**Age:** Years (100 for patients 90 or above) **Gender:**Female (0) or Male (1) **Unit1:** Administrative identifier for ICU unit (MICU) **Unit2:** Administrative identifier for ICU unit (SICU) **HospAdmTime:** Hours between hospital admit and ICU admit **ICULOS:** ICU length-of-stay (hours since ICU admit)	For sepsis patients, SepsisLabel is **1** For non-sepsis patients, Sepsis Label is **0**.

However, the roadmap for identifying the best sepsis predictors and integrating them with the Node-Red visual environment is described in Figure 6.

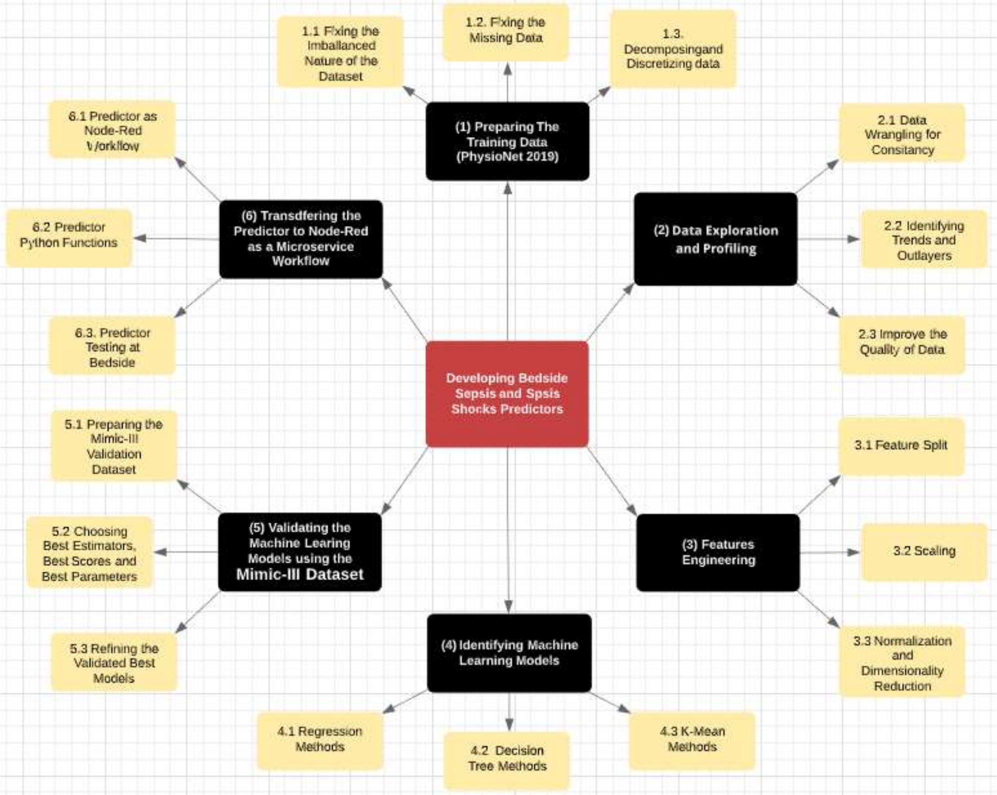

1.1 Fixing the Imballanced Nature of the Dataset
1.2. Fixing the Missing Data
1.3. Decomposingand Discretizing data
6.1 Predictor as Node-Red Workflow
(1) Preparing The Training Data (PhysioNet 2019)
2.1 Data Wrangling for Consitancy
6.2 Predictor Python Functions
(6) Transdfering the Predictor to Node-Red as a Microservice Workflow
(2) Data Exploration and Profiling
2.2 Identifying Trends and Outlayers
6.3. Predictor Testing at Bedside
2.3 Improve the Quality of Data
Developing Bedside Sepsis and Spsis Shocks Predictors
5.1 Preparing the Mimic-III Validation Dataset
3.1 Feature Split
5.2 Choosing Best Estimators, Best Scores and Best Parameters
(5) Validating the Machine Learing Models using the Mimic-III Dataset
(3) Features Engineering
3.2 Scaling
5.3 Refining the Validated Best Models
(4) Identifying Machine Learning Models
3.3 Normalization and Dimensionality Reduction
4.1 Regression Methods
4.2 Decision Tree Methods
4.3 K-Mean Methods

Fig. 6 Roadmap for Integrating Best Sepsis Predictors at the Node-Red Environment

4. CONCLUSIONS

The incorporation of machine learning into clinical medicine holds promise for substantially improving healthcare delivery. With the use of bedside visual tools that incorporates machine learning has the potential to improve patient and caregiver Satisfaction.. This stage 1 of the research illustrate how Node-Red and the notion of the visual workflow microservices can be imployed to integrate predictors for critical cases like the Septic Shocks. This research is an ongoing work in progress to develop innovative healthcare ecosystem that can work at the bedside.

ACKNOWLEDGMENT

This research is kind of extracts from the NSERC Discovery Grants 2020 Proposals of the first two authors. It has been submitted to the IEEE LISS 2020 July 25-28, 2020 Budapest, Hungary. This paper is an extension from the IEEE LISS 2020 paper.

REFERENCES

[1] Fleischmann, Carolin, André Scherag, Neill KJ Adhikari, Christiane S. Hartog, Thomas Tsaganos, Peter Schlattmann, Derek C. Angus, and Konrad Reinhart. “Assessment of global incidence and mortality of hospital-treated sepsis. Current estimates and limitations.” American journal of respiratory and critical care medicine, vol. 193, no. 3, (2016), pp. 259-272.

[2] Liu, Luchen, Haoxian Wu, Zichang Wang, Zequn Liu, and Ming Zhang. “Early Prediction of Sepsis From Clinical Datavia Heterogeneous Event Aggregation.” arXiv preprint arXiv:1910.06792 (2019).

[3] Nemati, Shamim, Andre Holder, Fereshteh Razmi, Matthew D. Stanley, Gari D. Clifford, and Timothy G. Buchman. “An interpretable machine learning model for accurate prediction of sepsis in the ICU.” Critical care medicine, vol. 46, no. 4, (2018), pp. 547-553.

[4] Giannini, Heather M., Jennifer C. Ginestra, Corey Chivers, Michael Draugelis, Asaf Hanish, William D. Schweickert, Barry D. Fuchs et al. “A Machine Learning Algorithm to Predict Severe Sepsis and Septic

Shock: Development, Implementation, and Impact on Clinical Practice.” Critical care medicine, vol. 47, no. 11, (2019), pp. 1485-1492.

[5] Desautels, Thomas, Jana Hoffman, Christopher Barton, Qingqing Mao, Melissa Jay, Jacob Calvert, and Ritankar Das. “Pediatric severe sepsis prediction using machine learning.” bioRxiv, (2017), pp. 223-289.

[6] Ibrahim, Zina M., Honghan Wu, Ahmed Hamoud, Lukas Stappen, Richard JB Dobson, and Andrea Agarossi. “On classifying sepsis heterogeneity in the ICU: insight using machine learning.” Journal of the American Medical Informatics Association, vol. 27, no. 3, (2020), pp. 437-443.

[7] Fleuren, Lucas M., Thomas LT Klausch, Charlotte L. Zwager, Linda J. Schoonmade, Tingjie Guo, Luca F. Roggeveen, Eleonora L. Swart et al. “Machine Learning for the Prediction of Sepsis, a Systematic Review and Meta-Analysis of Diagnostic Test Accuracy.” (2019).

[8] Nemati, Shamim, Andre Holder, Fereshteh Razmi, Matthew D. Stanley, Gari D. Clifford, and Timothy G. Buchman. “An interpretable machine learning model for accurate prediction of sepsis in the ICU.” Critical care medicine, vol. 46, no. 4, (2018), pp. 547-553.

[9] Seneviratne, Martin G., Nigam H. Shah, and Larry Chu. “Bridging the implementation gap of machine learning in healthcare.” BMJ Innovations, vol. 6, no. 2 (2020).

[10] Mohammed, Sabah, Jinan Fiaidhi, and Sami Mohammed. “Pragmatic Interoperability for Extreme Automation and Healthcare Interoperability & Continuity.” IEEE LISS 2020 Conference Preprint, TechRxIv (2020). https://www.techrxiv.org/articles/Pragmatic_Interoperability_for_Extreme_Automation_and_Healthcare_Interoperability_Continuity/11728956.

[11] Weber, Jens H., and Craig Kuziemsky. “Pragmatic interoperability for ehealth systems: the fallback workflow patterns.” In 2019 IEEE/ACM 1st International Workshop on Software Engineering for Healthcare (SEH), IEEE., (2019), pp. 29-36.

[12] Dunnion, Mary E., and Miriam Griffin. "Care planning in the emergency department." International emergency nursing 18, no. 2 (2010): 67-75.

[13] Liggins, Martin E.; Hall, David L.; Llinas, James (2008). Multisensor Data Fusion, Second Edition: Theory and Practice (Multisensor Data Fusion). CRC. ISBN 978-1-4200-5308-1.

[14] Mark Meloon, What is an example of data fusion? How does it help in big data? Is it data analysis technique?, Sep 19, 2014, Available Online: https://www.quora.com/What-is-an-example-of-data-fusion-How-does-it-help-in-big-data-Is-it-data-analysis-technique

[15] Erik Blasch, One decade of the Data Fusion Information Group (DFIG) model, Proceedings Volume 9499, Next-Generation Analyst III; 94990L (2015) https://doi.org/10.1117/12.2176934

[16] Event: SPIE Sensing Technology + Applications, 2015, Baltimore, Maryland, United States

[17] Brillman, Judith C., Tom Burr, David Forslund, Edward Joyce, Rick Picard, and Edith Umland. "Modeling emergency department visit patterns for infectious disease complaints: results and application to disease surveillance." BMC medical informatics and decision-making 5, no. 1 (2005): 4.

[18] Helland, Pat. "Life beyond Distributed Transactions: an Apostate's Opinion." In CIDR, vol. 2007, pp. 132-141. 2007.

[19] Sun, Long, Yan Li, and Raheel Ahmed Memon. "An open IoT framework based on microservices architecture." China Communications 14, no. 2 (2017): 154-162.