CHOC Team

Group 13

Evan Leong

Marcus Linture

Kenny Yu

Kelly Hu

Calvin Nguyen

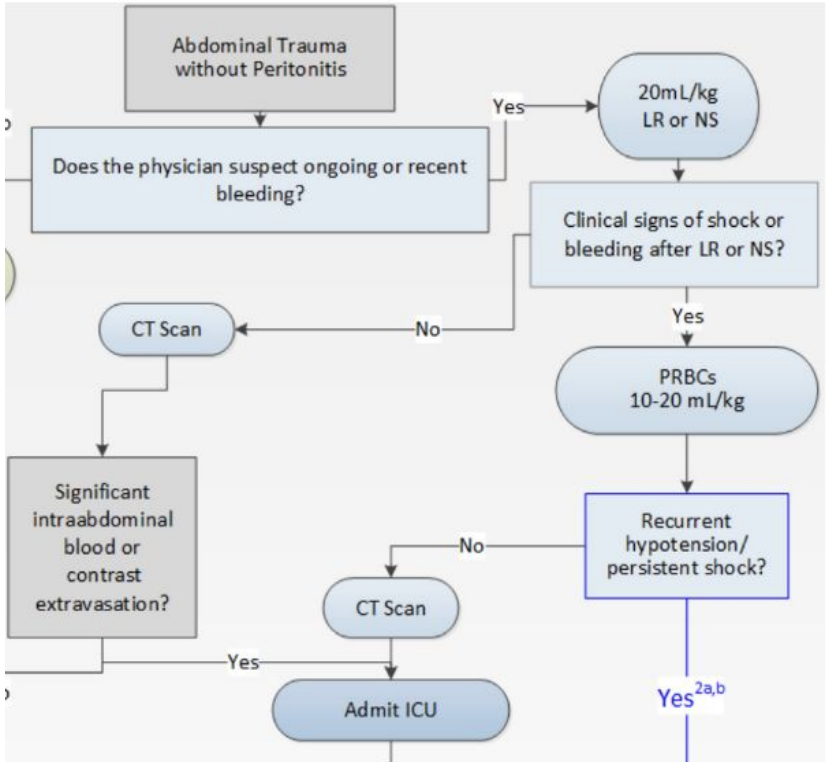
Introduction / The Problem

- Children's Hospital of Orange County
- Children in accidents that suffer from solid organ injuries (SOI)
- Assist clinical providers in determining whether or not clinical intervention is needed, based on ATOMAC guidelines
- Better understand the extreme values and cut points for bleeding in children through simulations

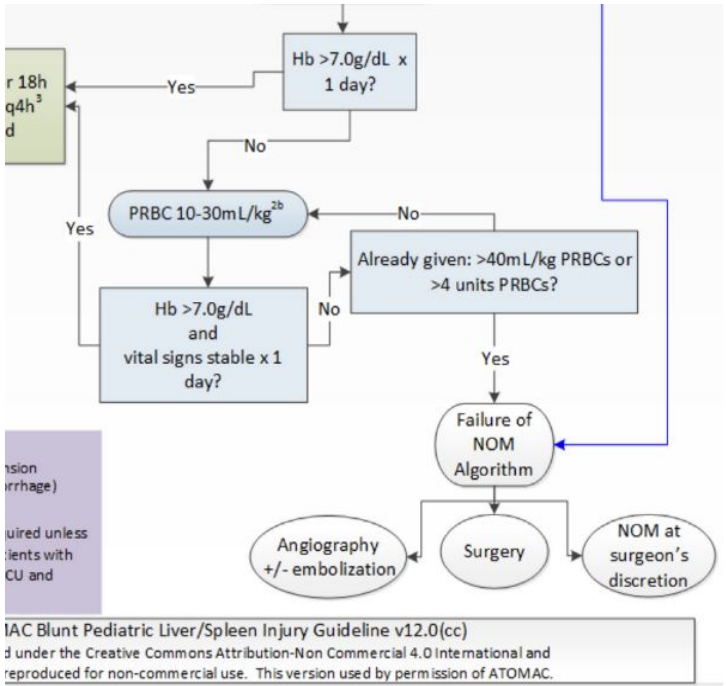
The Solution

- **Lagrangian Neural Network**
- **ATOMAC algorithm**
  - What is it?
  - How does it help?
- **Used by**
  - Surgeons
  - Doctors at CHOC
- **Front-end interface for easier accessibility**

Above



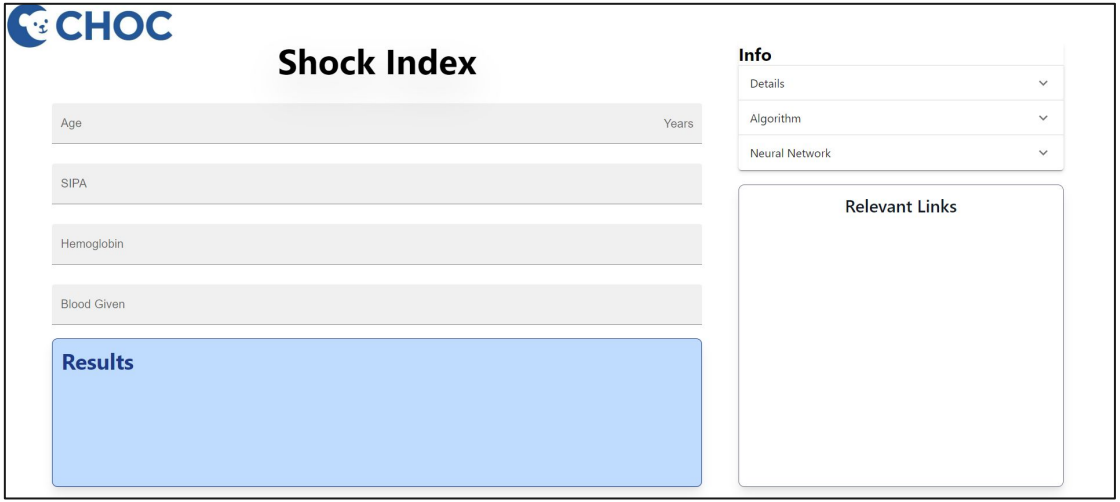
Below



AC Blunt Pediatric Liver/Spleen Injury Guideline v12.0(cc)
d under the Creative Commons Attribution-Non Commercial 4.0 International and
reproduced for non-commercial use. This version used by permission of ATOMAC.

of blunt liver or spleen injury. LR=lactated ringers; NS=normal saline; PRBCs=packed red blood cells; CT = computed
in; NPO=nothing by mouth; q6h=every 6 hours; NOM=nonoperative management; ICU=intensive care unit; q2h=every
Research Network; PedSRC = Pediatric Surgery Research Consortium; For Abdominal CT prediction rules, see (1) Streck

Interface



- **User inputs data**
  - Run through model
- **SIPA:** Shock Index - Pediatric Adjusted
  - Elevated SIPA = higher mortality rate
- **See if child is in shock**
  - Warning color and sound if in shock

Neural Network

- Preprocessing of Data:
  - feature selection
- Neural Network Training and Adjustments:
  - Goal: Lagrangian Networks
- SIPA Calculator
- Accuracy Testing:
  - Cross-validation
- Output: Yes, persistent shock / No, no action needed

Technical Details, Novelty and Challenges

- **Interface**
  - React
  - TypeScript
  - HTML & Tailwind CSS
  - Material UI
- Undefined Requirements

- **Machine Learning Model**
  - Lagrangian NN with ATOMAC algorithm
  - Extreme resources needed
    - Preprocessing, training, testing

Next steps

- Preprocess data
  - 200+ features, some temporal → combined features
- Start with less complex ML methods
  - kNN, SVM, PCA
- Build to Lagrangian NN
- Integration to front end

Thank You!
Any questions?