A SAT-based approach to rigorous verification of Bayesian networks

Ignacy Stępka, Nicholas Gisolfi, Artur Dubrawski

September 13, 2024

Content

- Motivation
- Background (DARPA Triage Challenge)
- Compilation & encoding
  - Compilation to an ODD
  - Encoding to a Boolean algebra formulae
- Verification queries
  - High-level idea
  - If-Then
  - Feature Monotonicity
- Use cases
- Future work

Why verification?

- Deploy ML systems in safety-critical real-world applications
- Verify model's adherence to properties desired by subject experts
- Ensure that the model will not ever inflict otherwise easily preventable harm
- Leverage the robust predictive capabilities of ML systems in real-world safety-critical scenarios

Problem Motivation

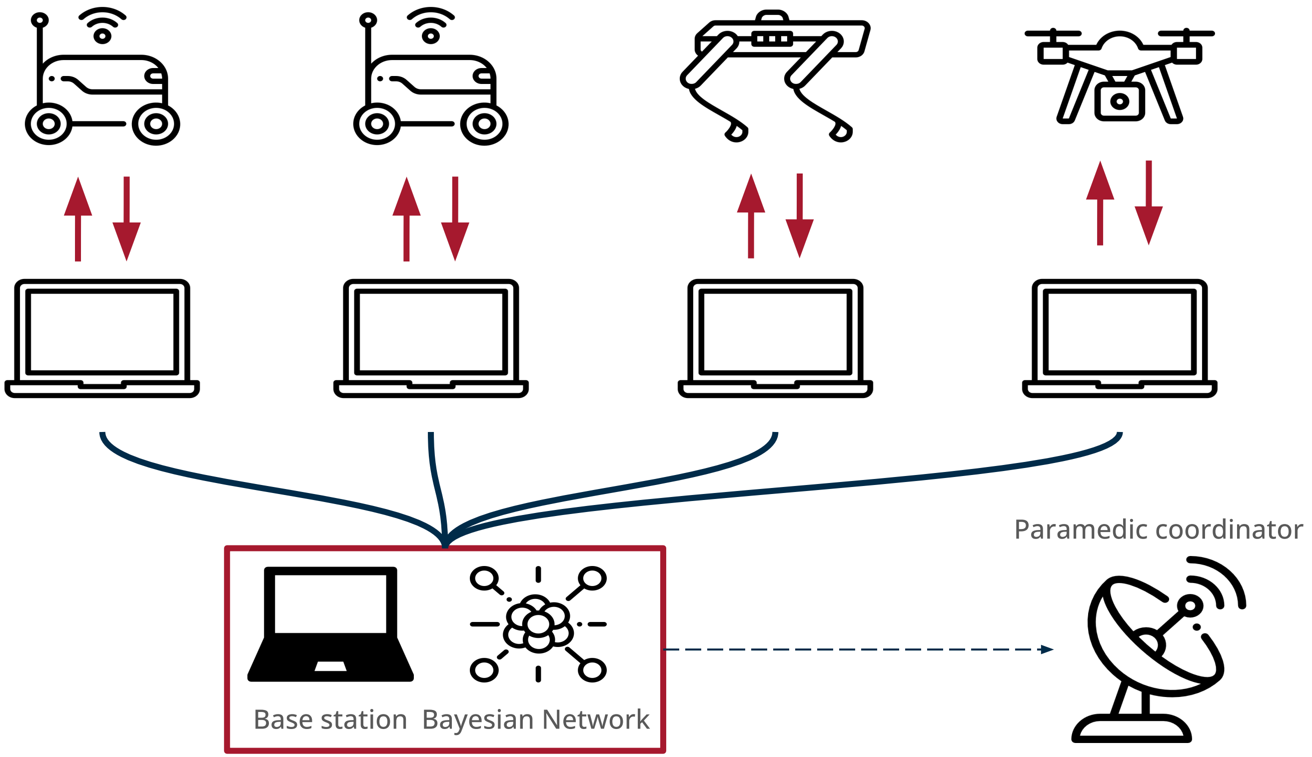
DARPA Triage Challenge

Image source: https://triagechallenge.darpa.mil

DARPA Triage Challenge

- Mass casualty setting: collapsed building, train crash, terrorist attack etc..
- Limited number of paramedics available right away
- Need: rapid assessment of casualty severity and prioritization (triage) for paramedics to maximize survivability of as many people as possible
- Group of robots equipped with various sensors to feed algorithms assessing vital signs and conditions (e.g., breathing rate, injury patterns)
- Strict and well-established medical guidelines for performing triage (e.g., SALT method)

Sketch of the setup



What to verify?

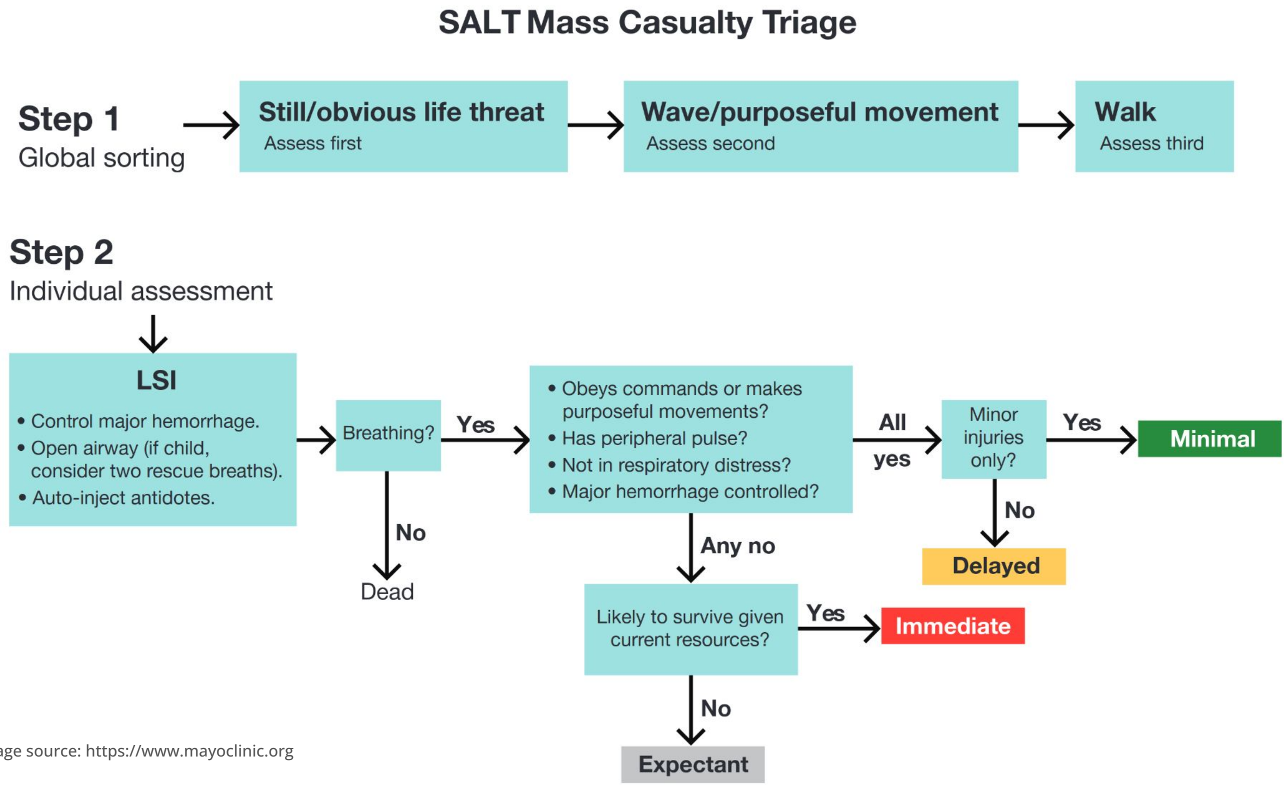
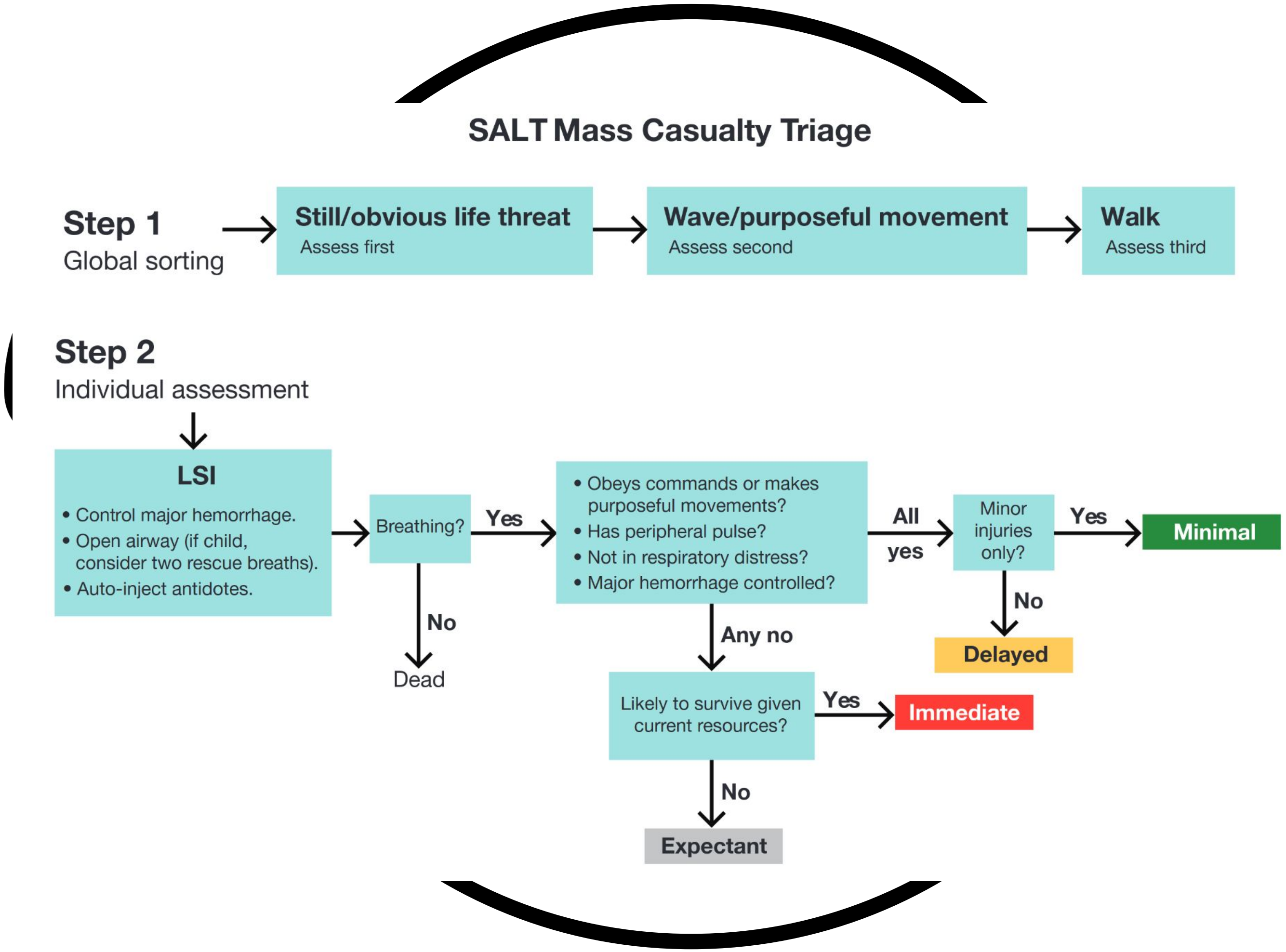
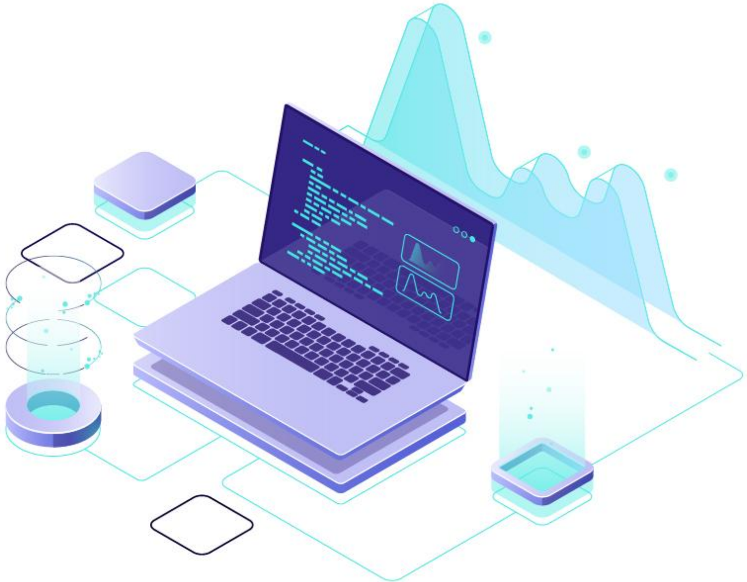


Image source: https://www.mayoclinic.org


SALT Mass Casualty Triage
Step 1
Global sorting
Still/obvious life threat
Assess first
Wave/purposeful movement
Assess second
Walk
Assess third
Step 2
Individual assessment
LSI
• Control major hemorrhage.
• Open airway (if child, consider two rescue breaths).
• Auto-inject antidotes.
Breathing?
Yes
No
Dead
• Obeys commands or makes purposeful movements?
• Has peripheral pulse?
• Not in respiratory distress?
• Major hemorrhage controlled?
All yes
Minor injuries only?
Yes
Minimal
No
Delayed
Any no
Likely to survive given current resources?
Yes
Immediate
No
Expectant

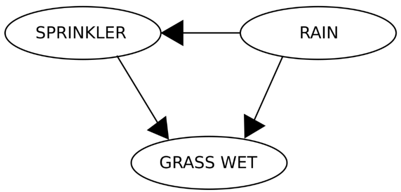
Bayesian Network

Compilation & encoding



Bayesian Network

RAIN	SPRINKLER T	SPRINKLER F
F	0.4	0.6
T	0.01	0.99



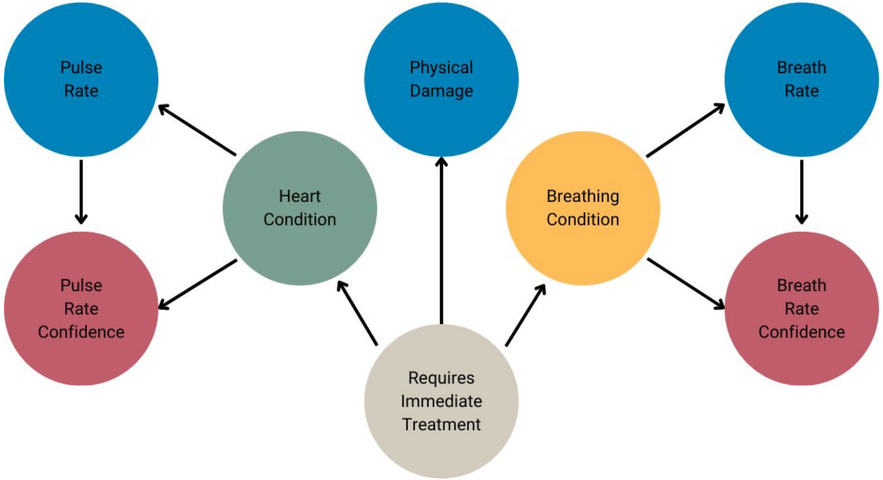
RAIN T	RAIN F
0.2	0.8

SPRINKLER	RAIN	GRASS WET T	GRASS WET F
F	F	0.0	1.0
F	T	0.8	0.2
T	F	0.9	0.1
T	T	0.99	0.01

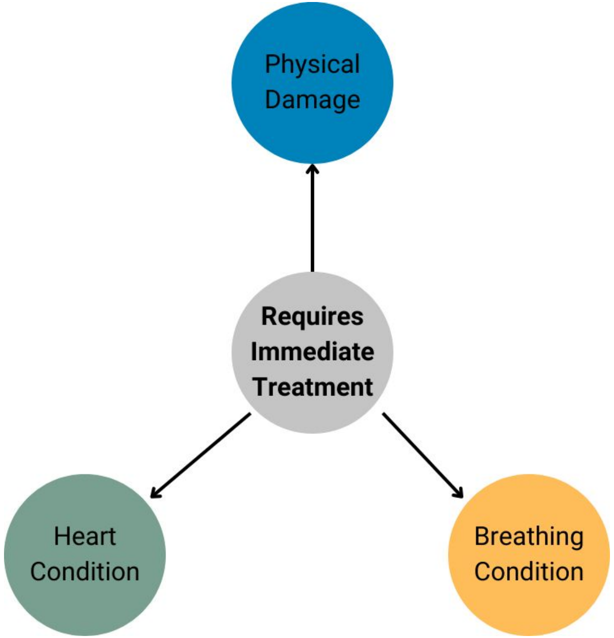
Image source: https://en.wikipedia.org/wiki/Bayesian_network

- **Structure:** directed acyclic graph (DAG)
- **Nodes:** Represent random variables
- **Edges:** Indicate conditional dependencies between variables.
- **Conditional Probability Tables (CPTs):** Each node is associated with a CPT that quantifies the effect of the parent nodes.

Ex. Bayesian Network for Triage

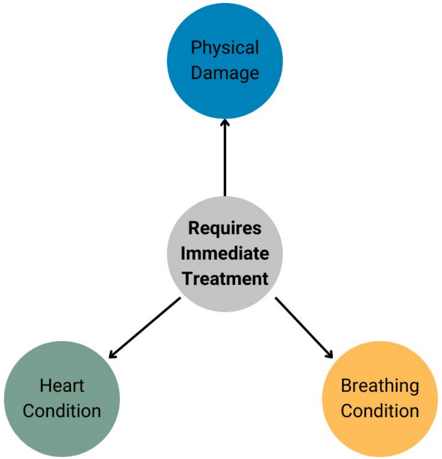


Let’s simplify



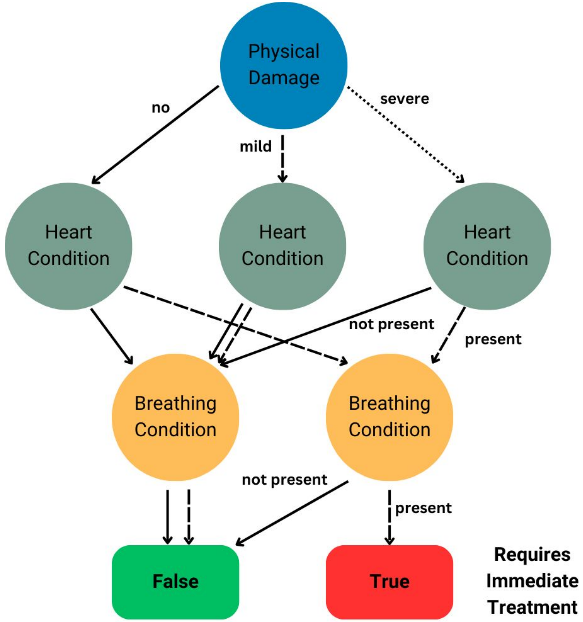
Compilation

Bayesian Network



compile

Multivalued Decision Diagram (MDD)

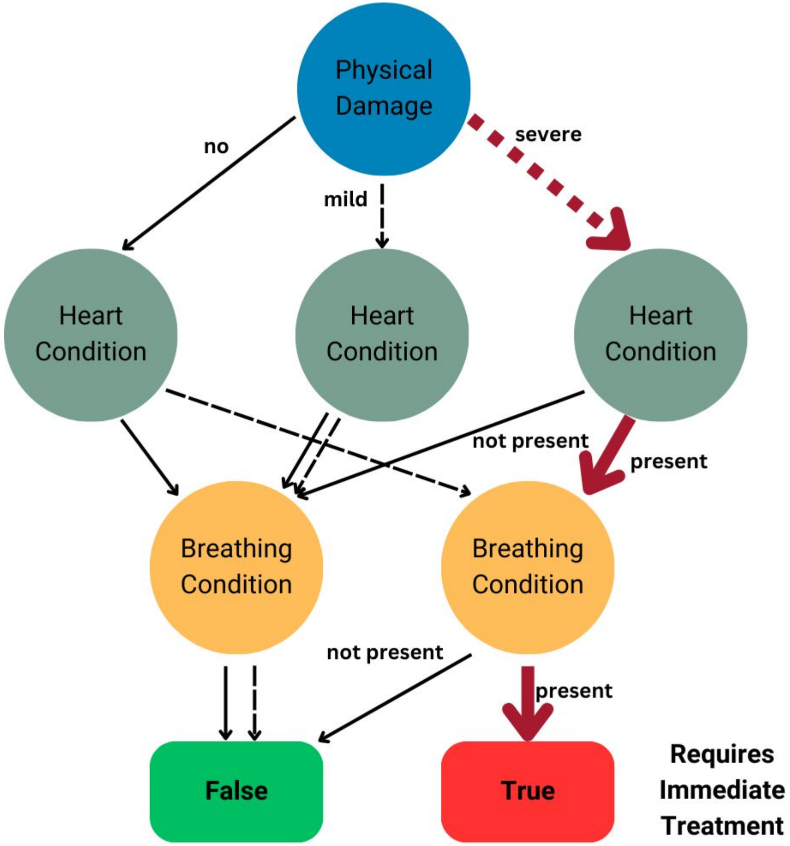


Shih et al. 2019 " Compiling Bayesian Network Classifiers into Decision Graphs"

Physical Damage = Severe

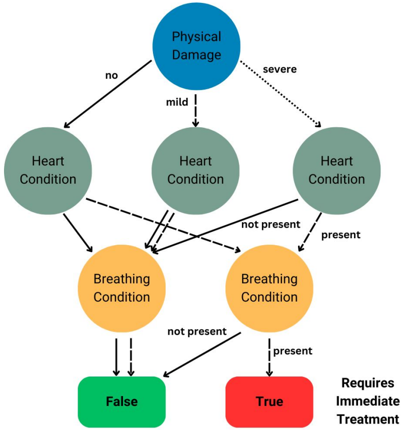
Heart Condition = Present

Breathing Condition = Present



Encoding

Multivalued Decision Diagram (MDD)



Encoding table

	Logical formulation	CNF equivalent
T1:	$v_i \rightarrow \vee_j \epsilon_{ij}$	$\neg v_i \vee ALO_j(\epsilon_{ij})$
T2:	$\epsilon_{ij} \rightarrow v_i$	$\neg \epsilon_{ij} \vee v_i$
T3:	$\epsilon_{ij} \rightarrow \mu_{ij}$	$\neg \epsilon_{ij} \vee \mu_{ij}$
T4:	$\epsilon_{ij} \rightarrow x_{ij}$	$\neg \epsilon_{ij} \vee x_{ij}$
T5:	$\mu_{ij} \wedge x_{ij} \wedge v_i \rightarrow \epsilon_{ij}$	$\neg \mu_{ij} \vee \neg x_{ij} \vee \neg v_i \vee \epsilon_{ij}$
P1:	$v_i \wedge x_{ij} \rightarrow \epsilon_{ij}$	$\neg v_i \vee \neg x_{ij} \vee \epsilon_{ij}$
P2:	$v_i \rightarrow \exists_j \delta_{i-1j}$ where $v_i \neq \rho$	$\neg v_i \vee_j \delta_{i-1j}$
P3:	$x_{ij} \rightarrow \exists \epsilon_{ij}$	$\neg x_{ij} \vee \epsilon_{ij}$
P4:	$EO(v$ for all v at the same level $i)$	$ALO(v) \wedge AMO(v)$

Table I

encode

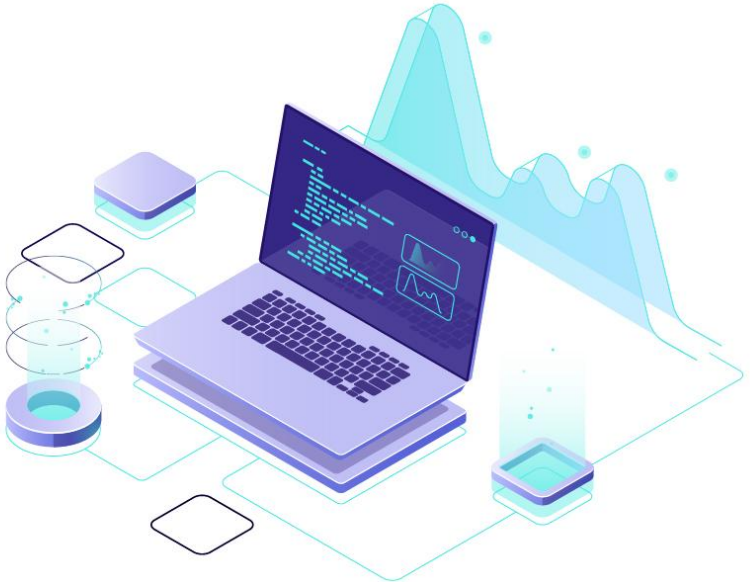
Conjunctive Normal Form (CNF)

$$(A \vee B) \wedge (A \vee B \vee C) \wedge \ldots$$

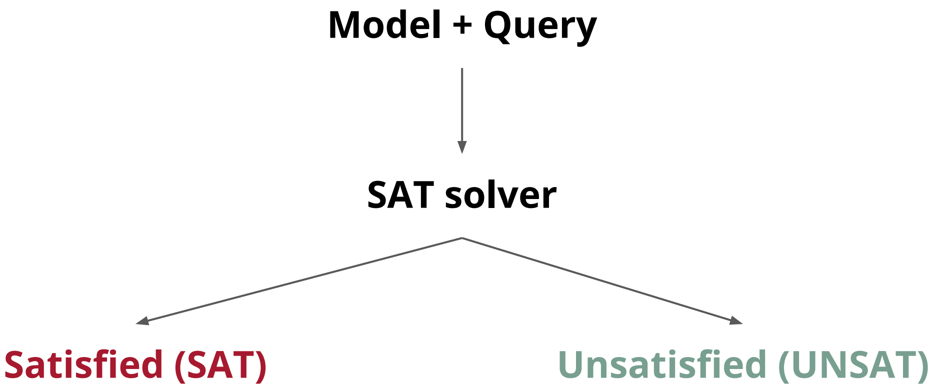
Abio et al. 2015 “On cnf encodings of decision diagram”

Verification queries

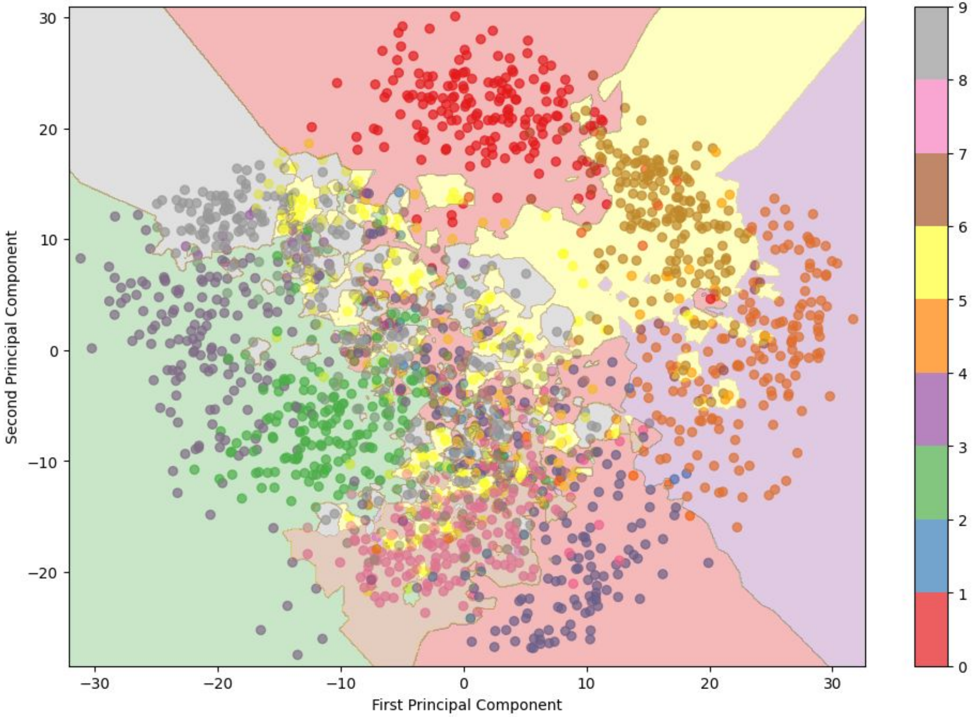
If-then & feature monotonicity



Template of verification

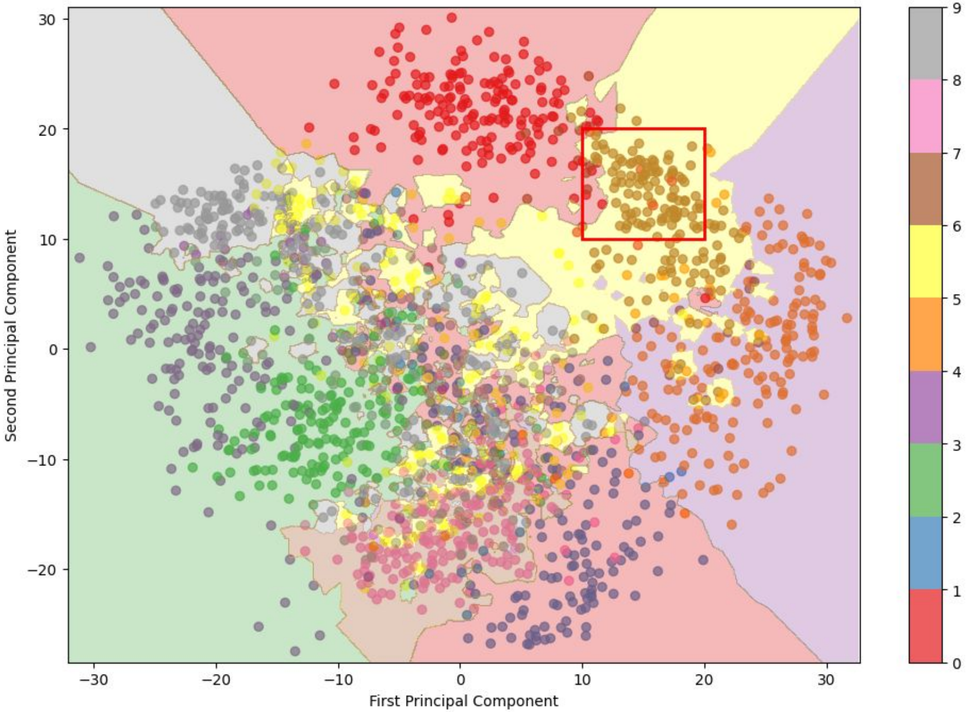


If-then rules (ITR)



If-then rules (ITR)

IF in the box
THEN class = 6



Make it a contradiction

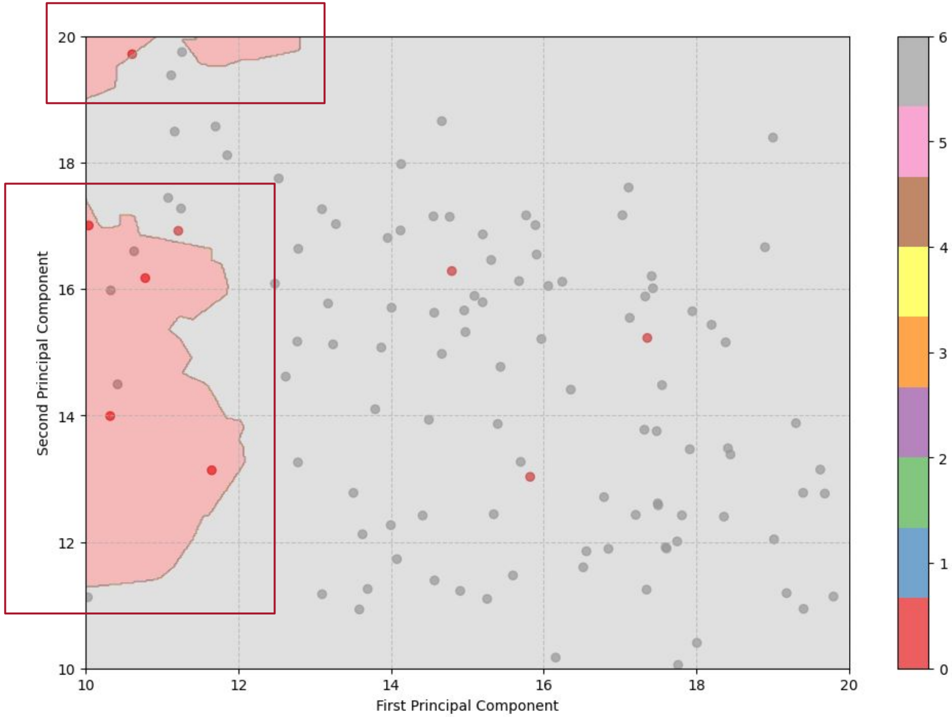
IF in the box
THEN class = 6

contradiction →

IF in the box
THEN class =/= 6

Verify the assertion

Not pure!



If-then rules (ITR)

Query:

IF
Pulse rate >= Insufficient and
Pulse rate confidence >= 50%

THEN
Requires immediate treatment >= True

Counterexample:

Pulse rate = Insufficient
Pulse rate confidence = 75%
Breath rate = Good
Physical damage = No

Requires immediate treatment = **False**

If-then rules (ITR)

Pulse_Rate >= Insufficient and
Pulse_Rate_Confidence >= 50%

Algorithm 1 If-Then Rules ($ITR_{M,R,c}$) Verification

```
Require: M                                                      ▷ Encoded model
Require: R ▷ Constraints on input. In each tuple, the first element is a variable, and
    the second is a threshold index.
Require: c                                         ▷ Prescribed output (class label)
1: r ← count(R)                            ▷ Set r to the number of premises in R
2: F ← ∅                                ▷ Initialize set F of 'and' separated literals
3: CNF_X ← ∅.                       ▷ Initialize set CNF_X of 'and' separated literals
4: For i from 1 to r:                                  ▷ Iterate over list of premises
    1. X, t ← R[i]
    2. l ← card(X)                ▷ Assign to l the number of unique variable values
    3. C ← ∅.                          ▷ Initialize set C of 'or' separated literals)
    4. For j from t to l:                      ▷ Iterate over X variable values above t
       (a) C ∪ X_j
    5. CNF_X ∪ C.
5: F ∪ M                                                            ▷ Add model
6: F ∪ CNF_X                                          ▷ Add correct constraint ranges
7: F ∪ Y_{1−c}                                ▷ Add the outcome of the undesired class
8: ITR_{M,R,c} ← assert F
```

Feature monotonicity

(partial assignment)

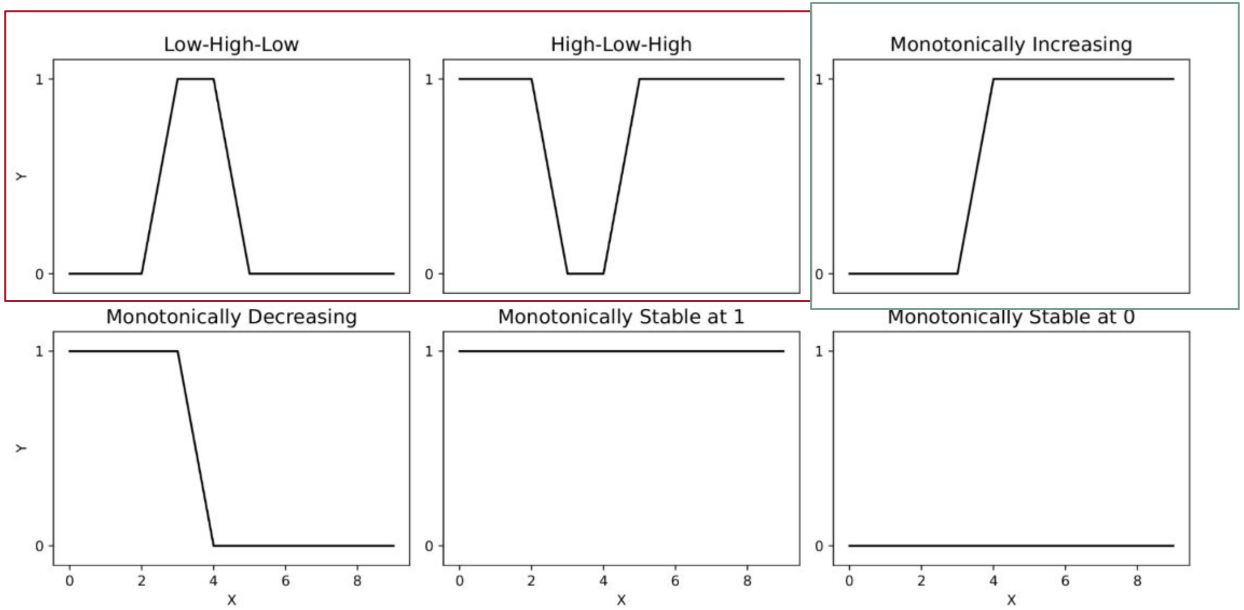
Given
Pulse rate = Low
and
Heart condition = Present

(query)

Does **Requires immediate treatment**
Pulse rate confidence?
have a monotonic relationship with
feature **Pulse rate confidence**?

Feature monotonicity

Requires Immediate Treatment



Pulse rate confidence?

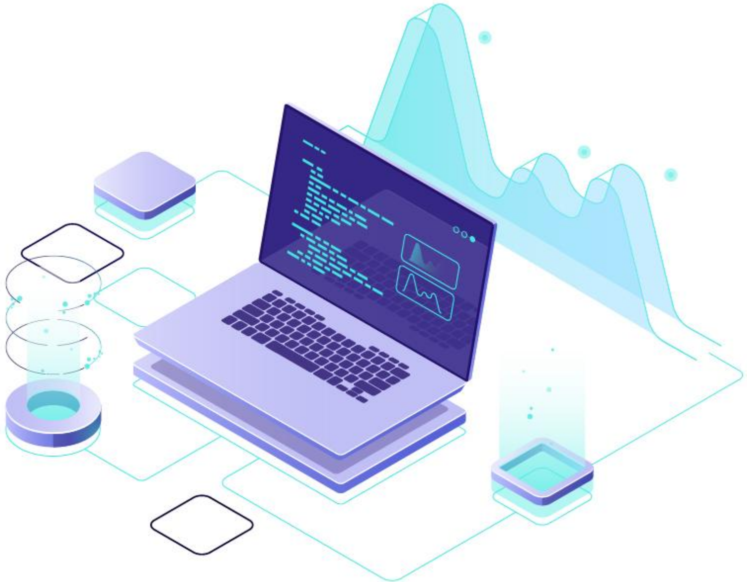
Feature monotonicity

Algorithm 2 Feature Monotonicity (FMO_{M,ϕ_{X^*},x_i}) Verification

Require: M ▷ Encoded model
Require: ϕ_X ▷ Partial assignment
Require: x_i ▷ Feature to check the monotonicity on

1: Create three copies of M: $M1$, $M2$, $M3$
2: $T \leftarrow \emptyset$ ▷ Create an empty CNF formula (operator **and** between elements)
3: For t from 1 to 2:
   1. $F \leftarrow \emptyset$ ▷ Create an empty CNF formula (operator **and** between elements)
   2. $F \cup M1 \cup M2 \cup M3$ ▷ Add models' literals
   3. $F \cup \phi_X$ ▷ Add partial assignment over all variables
   4. $F \cup (i_{x_i}^{M1} < i_{x_i}^{M2})$ ▷ Add inceasing assignment order on x_i in adjacent models
   5. $F \cup (i_{x_i}^{M2} < i_{x_i}^{M3})$
   6. If $t = 1$ then:
      (a) $F \cup (Y^{M2} > Y^{M1})$ ▷ Add outcome β_{LHL}
      (b) $F \cup (Y^{M2} > Y^{M3})$
      else:
      (a) $F \cup (Y^{M2} < Y^{M1})$ ▷ Add outcome β_{HLH}
      (b) $F \cup (Y^{M2} < Y^{M3})$
   7. $\tau \leftarrow$ assert F ▷ Assert the entire formula and return true or false
   8. $T \cup \neg\tau$ ▷ Add negation of the verification result
4: $FMO_{M,\phi_{X^*},x_i} \leftarrow$ assert T ▷ Get the final result of the verification query

Runtime experiments

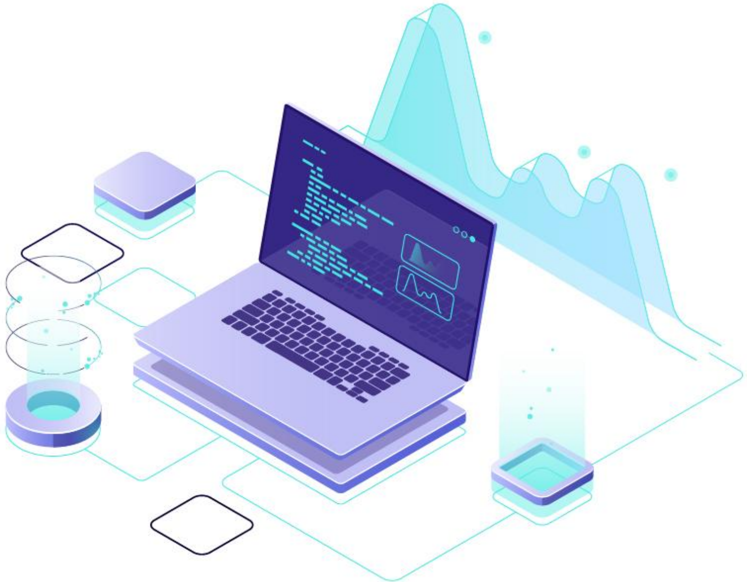


Time efficiency

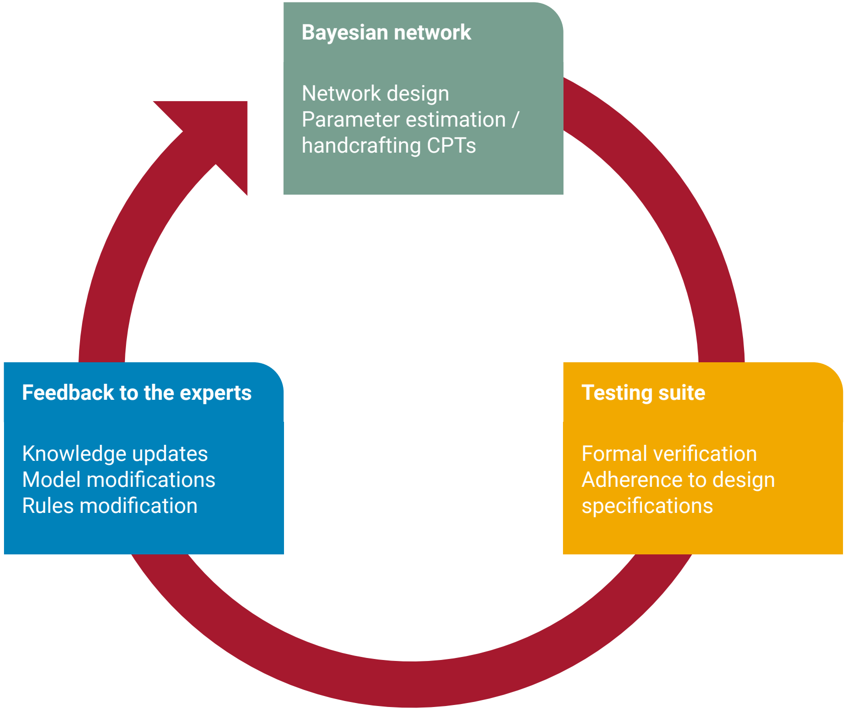
Model	Size (nodes)	Compilation time [s]	VQ#1 If-then [ms]	VQ#2 F. Mono [ms]
admission	5	2.39	0.79	16.54 (SAT)
asia	8	2.30	0.38	12.23 (SAT)
child	20	7.12	7.63	33.81 (SAT)
corical	20	7.22	3.61	13.78 (SAT)
alarm	37	253.53	38.34	166.08 (SAT)
win95pts	76	315.17	34.94	204.21 (SAT)

Use case example #1

Sanity checks for DARPA Triage

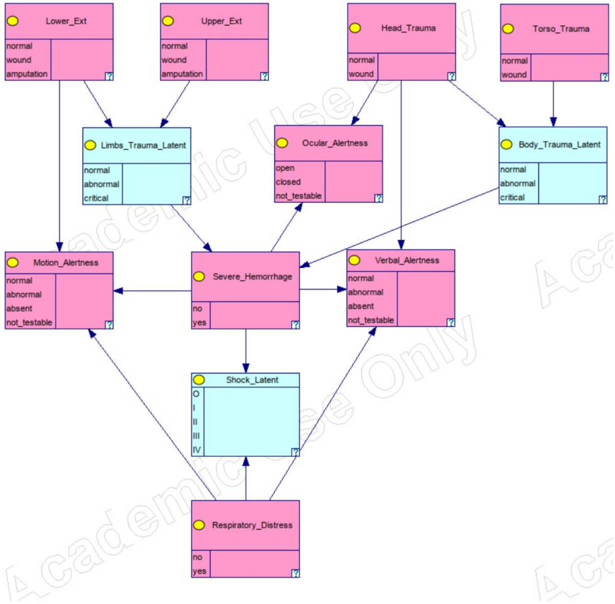


Verification & improvement lifecycle



Bayesian Network in DARPA Triage

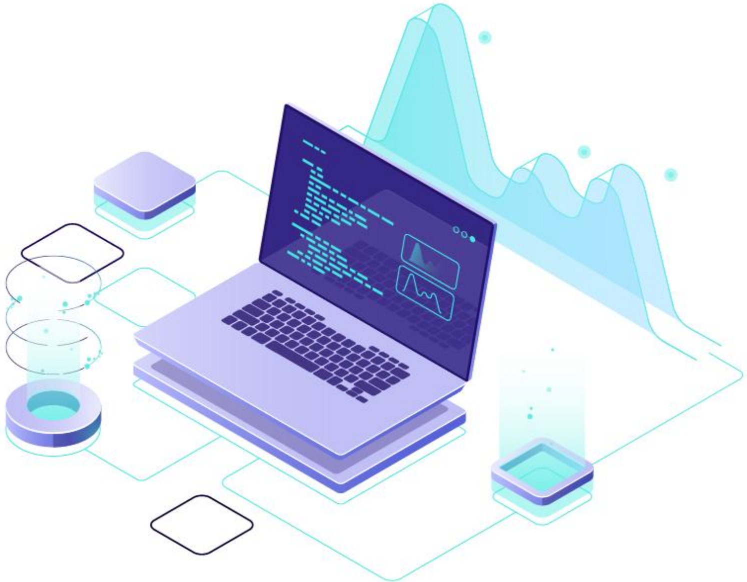
- Initially no access to any data
- Need to handcraft the structure of the Bayesian network in collaboration with domain experts (i.e., medical professionals)
- Later, probabilities can be adjusted with the actual data
- Generate rules from SALT and domain knowledge
- Iterate on the manual development of the Bayesian network with SALT compliance verification in between iterations
- Later, use the same approach for data-driven version



Actual Bayesian Network employed in the challenge

Use case example #2

Automated design specifications testing



Automated verification & validation

Specify verification queries according to design specifications

or

Discover queries from data, e.g., by extracting “pure” regions from the data

	BN0	BN1	BN2	BN3	BN4	BN5	BN6	BN7	BN8	BN9
UNSAT	0	1	0	0	7	2	0	3	0	4
SAT	11	10	11	11	4	9	11	8	11	7
Compliance %	0.00	9.09	0.00	0.00	63.64	18.18	0.00	27.27	0.00	36.36
Test Accuracy %	70.03	72.43	70.03	68.10	72.30	72.33	72.37	72.27	72.40	68.10

Thank you!

Paper & code

- Contact: istepka@andrew.cmu.edu
- If you find this talk interesting please do reach out and/or star our github repository

Thank you