A Hierarchical HAZOP-Like Safety Analysis for Learning-Enabled Systems

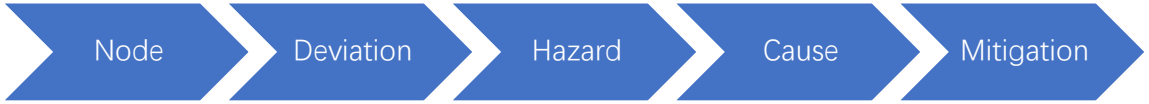
Yi Qi, Philippa Ryan Conmy, Wei Huang, Xingyu Zhao, Xiaowei Huang

UNIVERSITY OF LIVERPOOL

ADELARD part of nccgroup

What's the HAZOP?

A **hazard and operability study** (HAZOP) is a structured and systematic examination of a complex plan or operation in order to identify and evaluate problems that may represent risks to personnel or equipment.



- **Node**: Define a node from the pipeline according to the functional area.
- **Deviation**: Select guide word and attribute (parameter), applying the operational deviation to the node.
  - **Guide word**: Each guide word is a short word to create the imagination of a deviation of the design/process intent. The most commonly used guide words are: no, more, less, as well as, part of, and so on.
  - **Attributes:** Attributes are closely related to nodes and are usually the subject of the action being performed.
- **Hazard**: Hazard is a source of potential damage, harm, or adverse health effects on something/someone.
- **Cause**: Identify the causes of the deviation in the node.
- **Mitigation**: Enlist existing safeguards mitigating or preventing the hazards.

Motivation

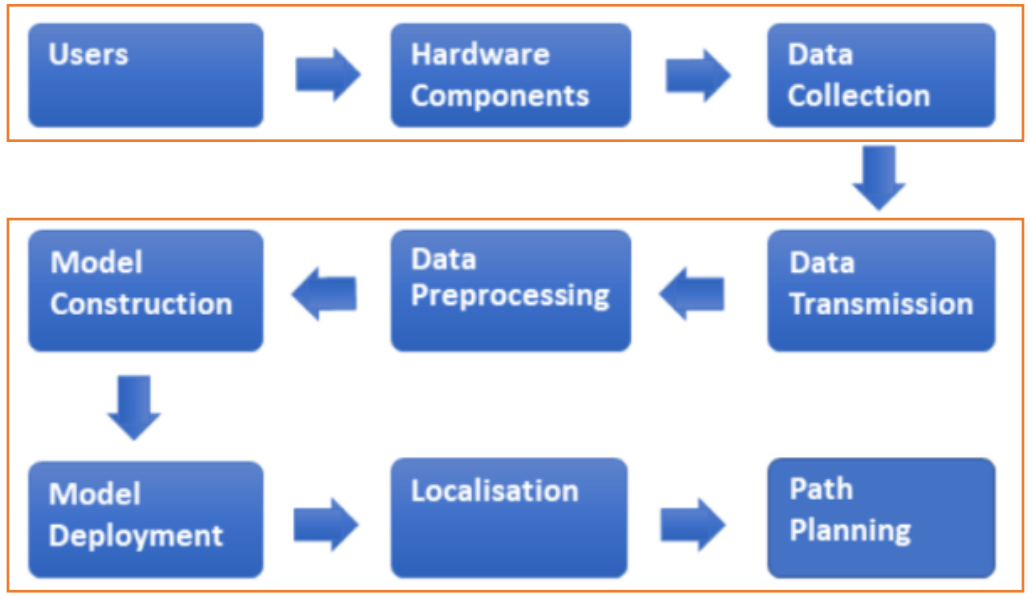


Figure 1: Workflow diagram of the running example

Traditional HAZOP will cover it.

Will traditional HAZOP still cover it?

- The emergence of complex systems with ML components tested HAZOP's abilities.
- Technical challenges raised by the novel characteristics of ML.

Overview of HILLS

HILLS is a Hierarchical HAZOP-Like method for LES.

- **System Level**: HILLS at the system level largely follow HAZOP.

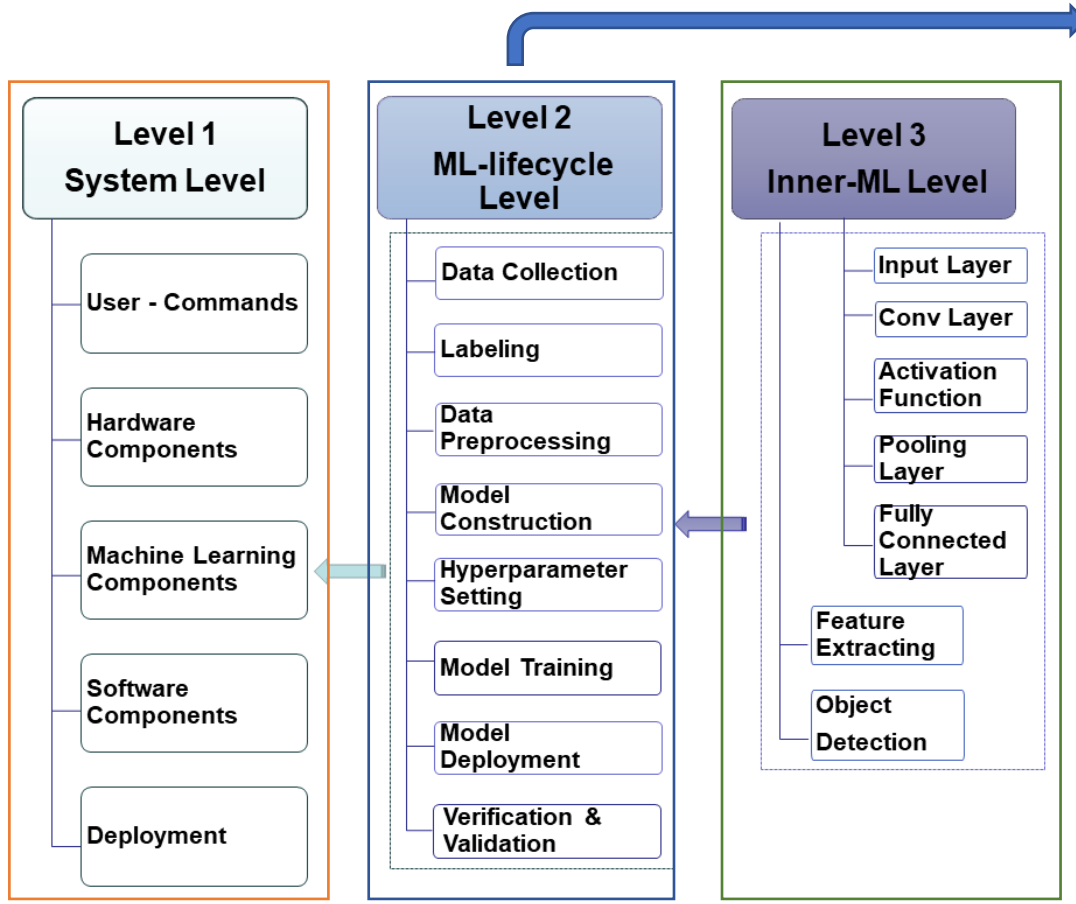


Figure 2: The 3-level hierarchical structure of HILLS

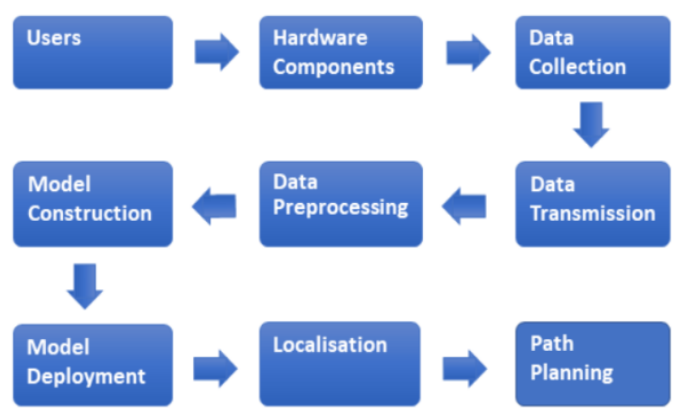
- **ML-lifecycle Level**: Considering mainly the human factors involved in the development of the ML models. On the other hand, the hazards at the system level may be attributed to the hazards at the ML-lifecycle level. We can analyse the direct human factor (labeling errors, operation errors) and security issues (adversarial attacks) at this level.

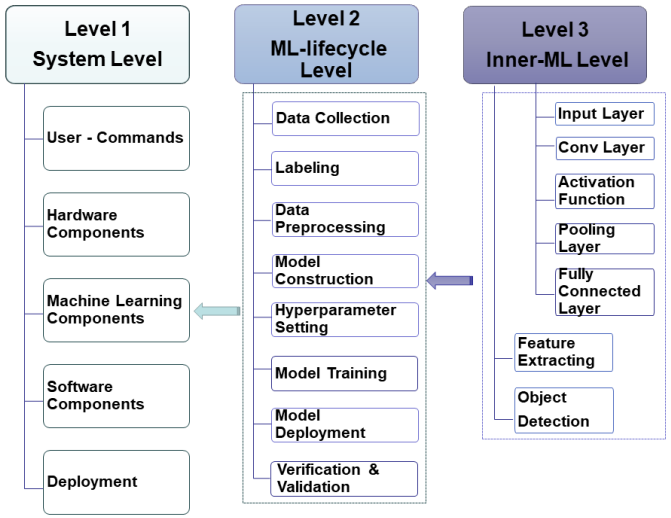
- **Inner-ML Level**: At the inner-ML level, HILLS takes the method of extracting basic layers of an ML component to form a model for analysis.

UNIVERSITY OF LIVERPOOL

ADELARD part of nccgroup

Running Example





Nodes in each level in SOLITUDE example

Level	Node	Description
System level	Node 1	User
System level	Node 2	Hardware components
System level	Node 3	Data transmission
ML-lifecycle level	Node 4	Data collection
ML-lifecycle level	Node 5	Labeling
ML-lifecycle level	Node 6	Data preprocessing
ML-lifecycle level	Node 7	Hyperparameter setting
ML-lifecycle level	Node 8	Model deployment
Inner-ML level	Node 9	Feature Extracting
Inner-ML level	Node 10	Object Detection
ML-lifecycle level	Node 11	Localisation

UNIVERSITY OF LIVERPOOL

ADELARD part of nccgroup

Partial Results

System level analysis (partial)

Node	Deviation	Hazard	Cause	Mitigation
Data transmission (Flow from camera to classifier)	No action	Erratic trajectory	No data from sensor (transient)	Acoustic guidance system
Data transmission (Flow from camera to classifier)	No action	Erratic trajectory	No data from sensor (transient)	Situational awareness (route mapped and planned in advance)
Data transmission (Flow from camera to classifier)	No action	Erratic trajectory	No data from sensor (transient)	Maximum safe distance maintained if uncertain
Data transmission (Flow from camera to classifier)	No action	Insufficient energy/power	No data from sensor (permanent)	Camera health monitor (e.g. sanity check for blank images)
Data transmission (Data flow)	Part of action	Erratic trajectory	Corrupted sensor data	Reliable camera (robust to environment etc.)
Data transmission (Data value)	Wrong value	Loss of communication	Hardware breakdown	Hardware monitor
Data transmission (Data value)	Wrong value	Loss of communication	Information conflict/lag	Maximum safe distance maintained if uncertain

Example 1. *At the system level, we discovered several hazards from the running example, some of them are summarised in Table 1. For example, one of the hazards is "erratic trajectory", suggesting that the robot moves into an unsafe area. This hazard is associated with a deviation "no action" where "no" is the guide word and "action" is the attribute. One of the causes of this hazard is "no data from sensor", which can be mitigated with e.g., the deployment of an acoustic guidance system as a duplicated perception component based on another sensor .*

Example 2. *Some hazards, such as "erratic trajectory", may appear in different nodes, which suggests that they may occur more often and have the higher priority to be mitigated.*

Partial Results

ML-lifecycle level analysis (partial)

Node	Deviation	Latent-hazard & Threat	Cause	Mitigation
Labeling (Manually label data)	Wrong label	Low prediction accuracy	Users make mistake with labeling	Keep classifier accuracy/reliability for critical objects >X
Labeling (Manually label data)	Wrong label	Low prediction accuracy	Users make mistake with labeling	Sanity check for ground truth and label attribute
Labeling (Manually label data)	Incapable label	Low prediction accuracy	Data itself is incomplete	Keep classifier accuracy/reliability for critical objects >X
Labeling (Manually label data)	Incapable label	Low prediction accuracy	Data itself is incomplete	Sanity check for ground truth and label attribute
Data collection	Attacked	Data Poisoning	Input data is contaminated	Detection based on data provenance
Data preprocessing	Part of data washing	Incorrect data ranges	Data washing incomplete	Consistency Check (e.g. Value range)
Hyperparameter setting	Wrong setting	Inappropriate hyperparameter	User make mistake with setting	Sanity check to hyperparameter
Hyperparameter setting	Wrong setting	Inappropriate hyperparameter	Unsuitable hyperparameter for setting	Continuing monitor to hyperparameter
Model deployment	Attacked	Robustness Attacks	Insert a calculated disturbance into the input data	Defensive Distillation
Model deployment	Attacked	Backdoor	Insert disturbance into the input data	XAI explain to input
Localisation	No Localisation	Lose estimation of position	Hardware (sensors) breakdown	Situational awareness (route mapped and planned in advance)
Localisation	No Localisation	Lose estimation of position	Hardware mismatch	Common time to synchronise data and results
Localisation	Wrong Localisation	Misposition	Slip rate too large	Situational awareness (route mapped and planned in advance)
Localisation	Wrong Localisation	Misposition	Combination miss between hardware and ML	Common time to synchronise data and results

Example 3. *There is a deviation "Attacked", whose threats are various attacks, e.g., evasion attacks, backdoor attacks, and data poisoning attacks. Their respective cause is usually that a certain entity in the training or inference of a ML model (e.g., input instance, model structure, training, dataset) is perturbed, modified, or contaminated. Their respective mitigation can be very specific.*

Partial Results

Inner-ML level analysis (partial)

Node	Deviation	Latent-hazard & Threat	Cause	Mitigation
Feature extracting	Imprecise extracting	Wrong outputs	Less layers	Using deeper layers
Feature extracting	Wrong extracting	Wrong outputs	Wrong hyperparameter setting	Using Explainable AI (XAI) to locate
Feature extracting	Wrong extracting	Wrong outputs	Unsuitable kernel size setting	Kernel size need to match dataset size
Feature extracting	Wrong extracting	Dying ReLU problem	Learning rate setting too large	Choosing suitable learning rate for ReLU (activation function)
Feature extracting	Wrong extracting	Losing information of figures	Unsuitable parameter setting in pooling layer	Evaluate whether need pooling layer
Feature extracting	Wrong extracting	Losing information of figures	Unsuitable parameter setting in pooling layer	Choose an appropriate pooling type

Example 4. *When the ML component has wrong output, we can get from the inner-ML level analysis that this may be related to the setting of the hyperparameter. Explainable AI (XAI) helps users locating.*

Example 5. *Unsuitable parameter setting in activation function or pooling layer also make specific threats. It also lead to wrong outputs or losing part of information of figures. At the inner-ML level, we focus on the ML model structure itself.*

Partial Results

- Qualitative Analysis

QUAL studies the connections between levels, with the guide words as entry points.

Example 6. *We use "no" as an example. We can get a deviation "no action" at the system level, and have the deviation "no localisation" at ML-lifecycle level. Given they share the same guide word, we should consider whether the "no localisation" has a causality relation with the "no action"*

- Quantitative Analysis

Example 7. *For threat nodes with no incoming arrows, such as T 2.i and T 3.i, we may set the probability of their occurrence infinitely close to 100 percent (99% as in Figure).*

Example 8. *There may be multiple children nodes at different levels for a parent node. In Figure 1, the threat T 2.i has two causes, C2.a and C3.a, at ML-lifecycle level and Inner-ML level, respectively. While the two causes may be mitigated separately as they belong to different levels,*

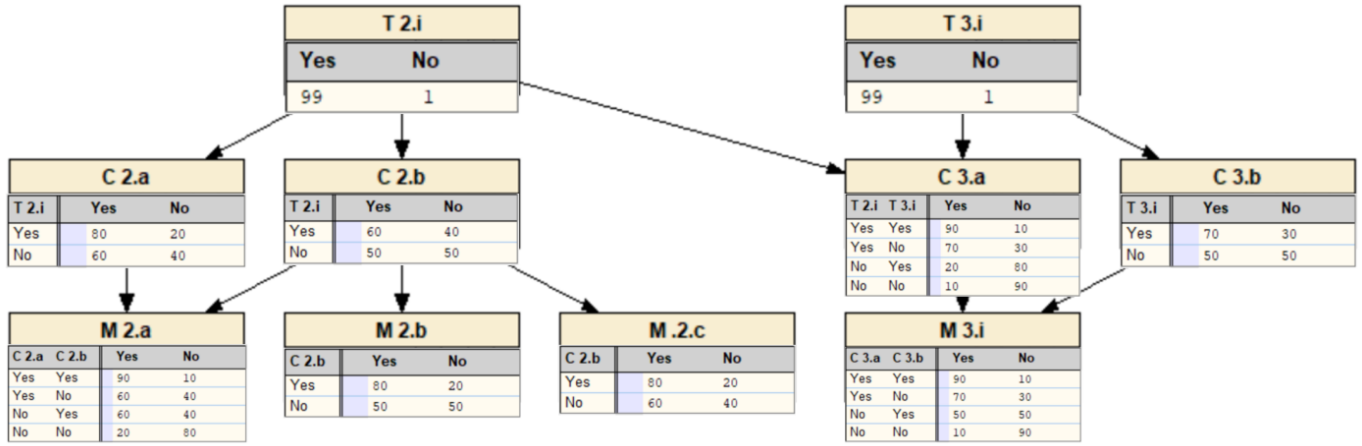


Table 3: A BN fragment (with illustrative probabilities)

UNIVERSITY OF LIVERPOOL

ADELARD part of nccgroup

Conclusion

- We propose a hierarchical HAZOP-like method, HILLS for the safety analysis of LESs.
- We try to propose to study formal analysis of the relations both qualitatively and quantitatively.
- HILLS is applied to a practical example of AUVs, with the discovery of new causes and mitigation related to ML.
- HILLS complements HAZOP when working with LESs.

Thank you watching

Any Questions and Comments

yiqi@liverpool.ac.uk

UNIVERSITY OF LIVERPOOL

ADELARD part of nccgroup