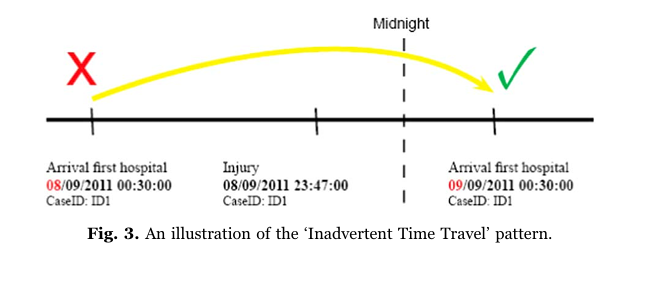
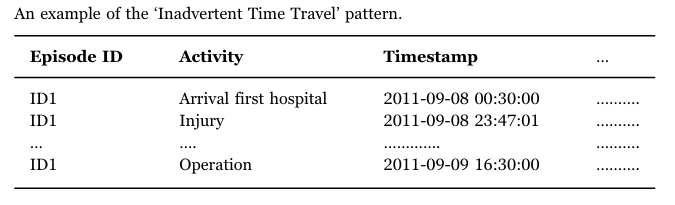
**InadvertentTimeTravel**

InadvertentTimeTravel occurs when erroneous timestamps are recorded due to their proximity to correct values, often leading to incorrect temporal ordering of events. This pattern impacts data accuracy and requires rectification by adjusting timestamps based on known rules to restore the correct event sequence.



**Detection**

This pattern can be detected by identifying incorrect event ordering.That is, given two activities a1 and a2 with a strict temporal ordering property such that a1 should always occur before a2, there exists at least a case with two events with activities a1 and a2 within correct ordering.

**Remedy**

Addressing this pattern in a generic manner requires knowledge of the minimum restrictions applicable to the ordering of all activities in the log.

**Algorithm**

Since it requires some intelligence to identify this pattern in a generic manner we decide to use a LLM model to detect this.

We give an event log to a LLM model and ask it whether the events are in order or not.

Below is an example:

Input events:

1. Operation
2. Injury
3. Arrival to hospital

import replicate

input = {

"top\_p": 1,

"prompt": """

are the given events in order? (say "yes" or "no" don't say anything else)

1) Operation

2) Injury

3) Arrival to hospital

""",

"temperature": 0.5,

"system\_prompt": "check the sequence of events for their order. If the events are in order say 'yes' else say 'no'. nothing else. Be simple",

"max\_new\_tokens": 500

}

for event in replicate.stream(

"meta/llama-2-70b-chat",

input=input

):

print(event, end="")

Output:

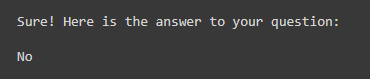


We have tested this using following LLMs

* *meta/llama-2-7b-chat*
* *meta/llama-2-13b-chat*
* *meta/llama-2-70b-chat*

*For most occasions meta/llama-2-70b-chat gives consistent answers. Other LLMs give answers in different formats.*

**

**

**Remedy:** We use the same LLM for identifying the correct event ordering.

import replicate

input = {

"top\_p": 1,

"prompt": """

give correct event order

1) Operation

2) Injury

3) Arrival to hospital

""",

"temperature": 0.5,

"system\_prompt": "give correct event ordering of events, don't give any explanation",

"max\_new\_tokens": 500

}

for event in replicate.stream(

"meta/llama-2-70b-chat",

input=input

):

print(event, end="")

Output:

