Monitoring and Optimization of Daily Activities

Potential in Environmental Adaptation and Habit Monitoring

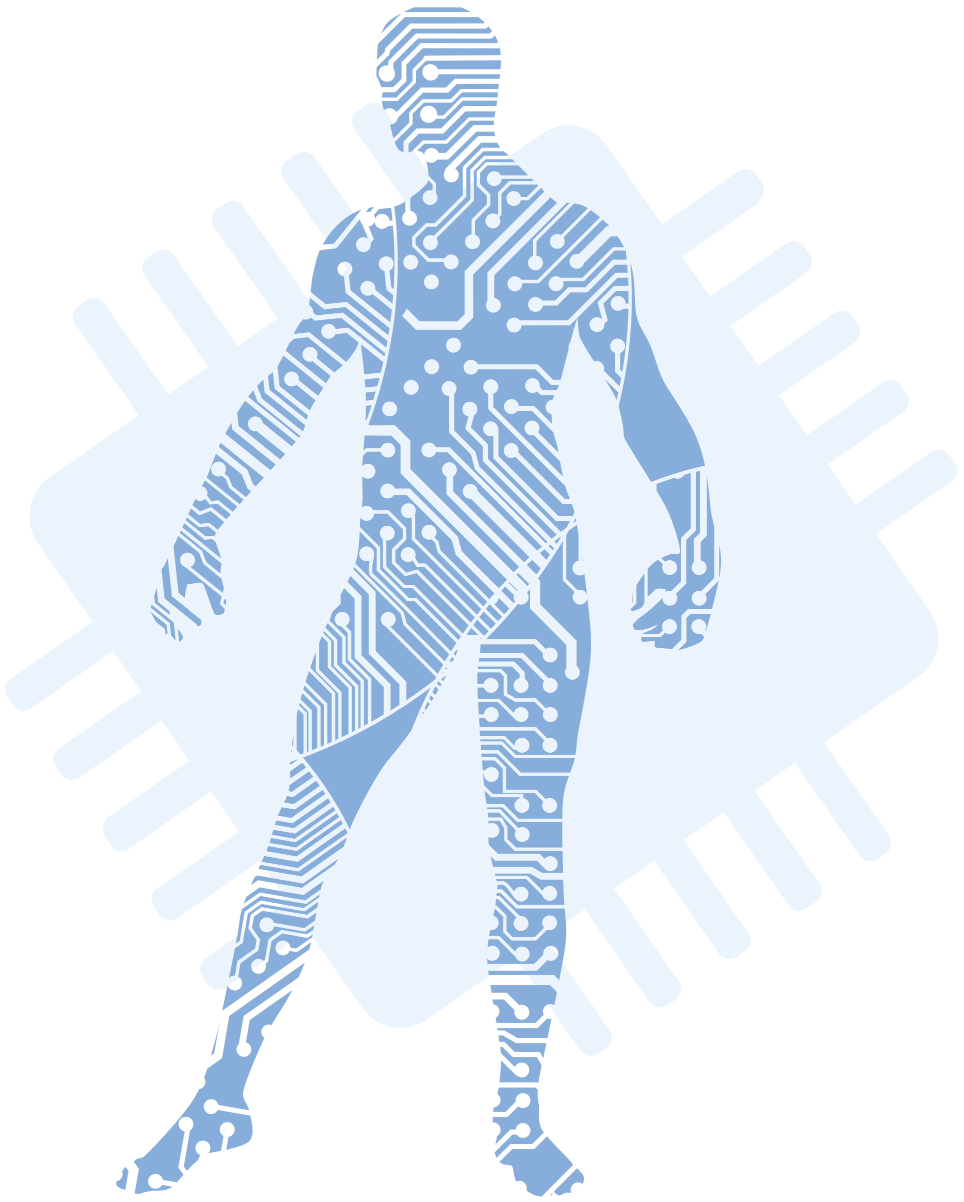
Stefan Andrei Alexandru

Project goals

This project wants to track a person's daily behavior to find their habits and discover patterns, which can then be used to improve their health and quality of life.

It is possible to get information like:

- The typical order of daily activities.
- The average length of each activity.
- The variation in daily routines
- Patterns or connections between activities (for example, if 'toilet' is always followed by 'sleep')

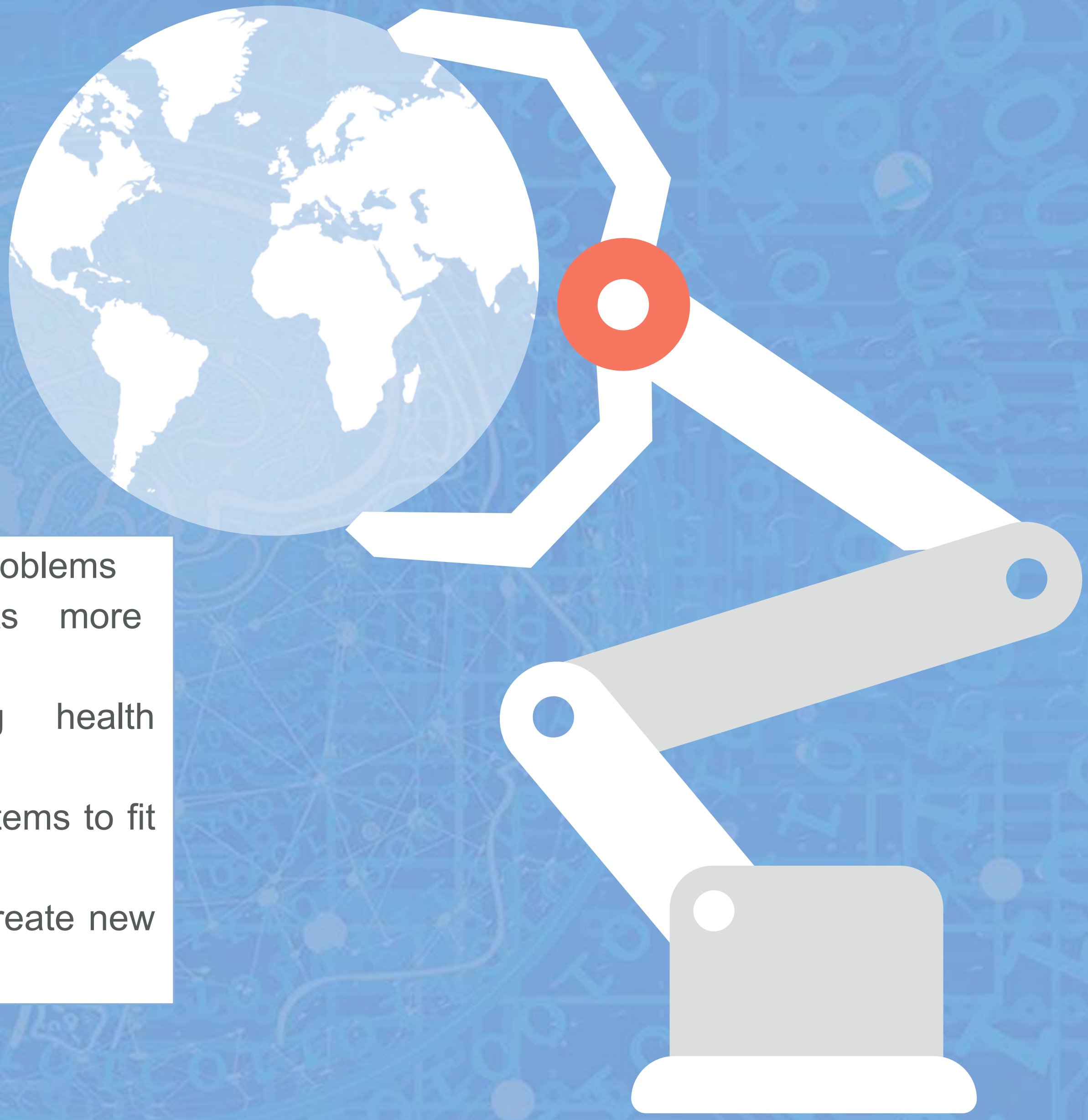


Practical implementations

Having the activity pattern helps predict the person's needs, allowing the environment to be adjusted to their needs.

For example, like a robot or a smart home, the environment can be set up for daily activities like showering or sleeping. Also, unusual activity can be spotted compared to the usual pattern, improving quality of life and preventing potential problem

Who is it useful for?



- Elderly: Tracking daily routines and finding health problems
- People with disabilities: Making environments more accessible and personal
- Chronic or cognitive conditions: Improving health management and preventing risks
- Smart home technology: Improving automated systems to fit daily habits
- Research: Studying behavior to improve life and create new technology

PROCESS MINING

Process mining is a method that looks at data from computer systems to understand, track, and improve business processes. By using logs of activities, it helps to see and analyze how work flows, finding problems, deviations, and areas that can be improved.

Using this approach, we apply process mining to study people's behaviors and routines. Instead of business processes, we analyze data from daily activities like movements and tasks. All actions were recorded in an XES(eXtensible Event Stream) file, a standard format for event logs.

DATASET

Activity performed at home

The implemented dataset contains logs of activities within daily habits, monitoring common actions like eating, watching TV, working, or using the bathroom, along with the exact time each action is performed. It also preserves the sequence in which these actions take place.

The dataset includes the following details:

- **Number of traces**: 43, representing individual sequences of activities performed by the person.
- **Number of events**: 4196, indicating the total number of recorded actions within these traces, capturing various daily activities and their exact timestamps.

Implementation

Step 1
Loading Dataset

Loads an XES file log with daily activities and timestamps

Step 2
Exploring Data

Analyzes the structure, trace count, and key activities

Step 3
Filtering Log

Filters traces that end with an "End" activity

Step 4
Discovering Models

Applies process mining algorithms to discover process models

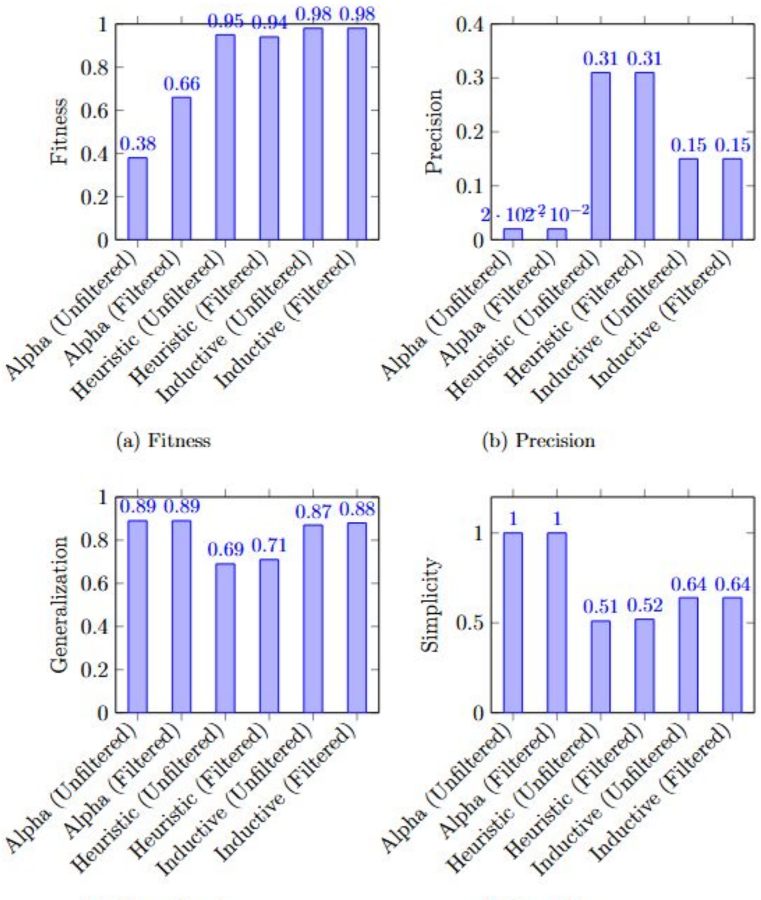
Step 5
Visualizing Models

Displays discovered models as Petri nets

Step 6
Evaluation and Conformance

Evaluates model accuracy and checks trace conformance.

Results



(a) Fitness

(b) Precision

(c) Generalization

(d) Simplicity

Model	Fitness	Precision	Generalization	Simplicity
Alpha Miner (Unfiltered)	0.38	0.02	0.89	1.0
Alpha Miner (Filtered)	0.66	0.02	0.89	1.0
Heuristic Miner (Unfiltered)	0.95	0.31	0.69	0.51
Heuristic Miner (Filtered)	0.94	0.31	0.71	0.52
Inductive Miner (Unfiltered)	0.98	0.15	0.87	0.64
Inductive Miner (Filtered)	0.98	0.15	0.88	0.64

- **Fitness**: Indicates the model's match with the log data. High fitness means the model explains most behaviors
- **Precision**: Assesses the model's ability to avoid behaviors not in the log. High precision means only observed behaviors are included
- **Generalization**: Evaluates the model's performance in new situations. High generalization means the model can handle unseen scenarios
- **Simplicity**: Measures the model's ease of understanding. High simplicity means the model is compact and easy to explain

Alpha Miner (Unfiltered)

- **Fitness (0.38)**: Low, The model poorly reproduces observed behavior in the log; many traces are incomplete or unrecognized
- **Precision (0.02)**: Very Low, The model is overly permissive, generating many sequences not present in the log
- **Generalization (0.89)**: High, Strong ability to generalize to new scenarios not observed in the log
- **Simplicity (1.0)**: High, ensuring ease of interpretation

The Alpha Miner model is not very accurate or comprehensive based on the results, but it is easy to understand and can be useful in scenarios where generalization is important

Alpha Miner (Filtered)

- **Fitness (0.66)**: Moderate, with room for improvement; the model reproduces much of the observed behavior
- **Precision (0.02)**: Very Low, The model is overly permissive, generating many sequences not present in the log
- **Generalization (0.89)**: High, Strong ability to generalize to new scenarios not observed in the log
- **Simplicity (1.0)**: High, ensuring ease of interpretation

With the filter applied, Alpha Miner improves slightly in fitness but still has low precision

Heuristic Miner (Unfiltered)

- **Fitness (0.95)**: High, the model closely matches the behaviors in the log
- **Precision (0.31)**: low, behaviors not in the log are still generated
- **Generalization (0.69)**: Moderate, it may not easily adapt to new situations
- **Simplicity (0.51)**: Moderate, the model is fairly complex

Heuristic Miner is accurate and closely matches the data, but it might create behaviors not seen in the data and has some difficulty generalizing. It is moderately complex

Heuristic Miner (Filtered)

- **Fitness (0.94)**: Very good, but slightly lower than the unfiltered version
- **Precision (0.31)**: Remains the same
- **Generalization (0.71)**: Slightly improved compared to the unfiltered version
- **Simplicity (0.52)**: Slightly more complex than the unfiltered version.

With the filter, Heuristic Miner remains accurate, but precision and generalization are still moderate

Inductive Miner (Unfiltered)

- **Fitness (0.98)**: Very high, the model almost perfectly reproduces the behavior in the data
- **Precision (0.15)**: Low, it allows some flexibility by generating unobserved behaviors
- **Generalization (0.87)**: High, the model is good at generalizing
- **Simplicity (0.64)**: Moderate, the model is relatively complex compared to others.

Inductive Miner is highly accurate with good generalization ability, but it's precision is low, allowing for some unobserved behaviors.

Inductive Miner (Filtered)

- **Fitness (0.98)**: Remains very high, similar to the unfiltered version
- **Precision (0.15)**: Stays low, meaning unobserved sequences are still generated
- **Generalization (0.88)**: Slightly improved compared to the unfiltered version
- **Simplicity (0.64)**: Remains the same

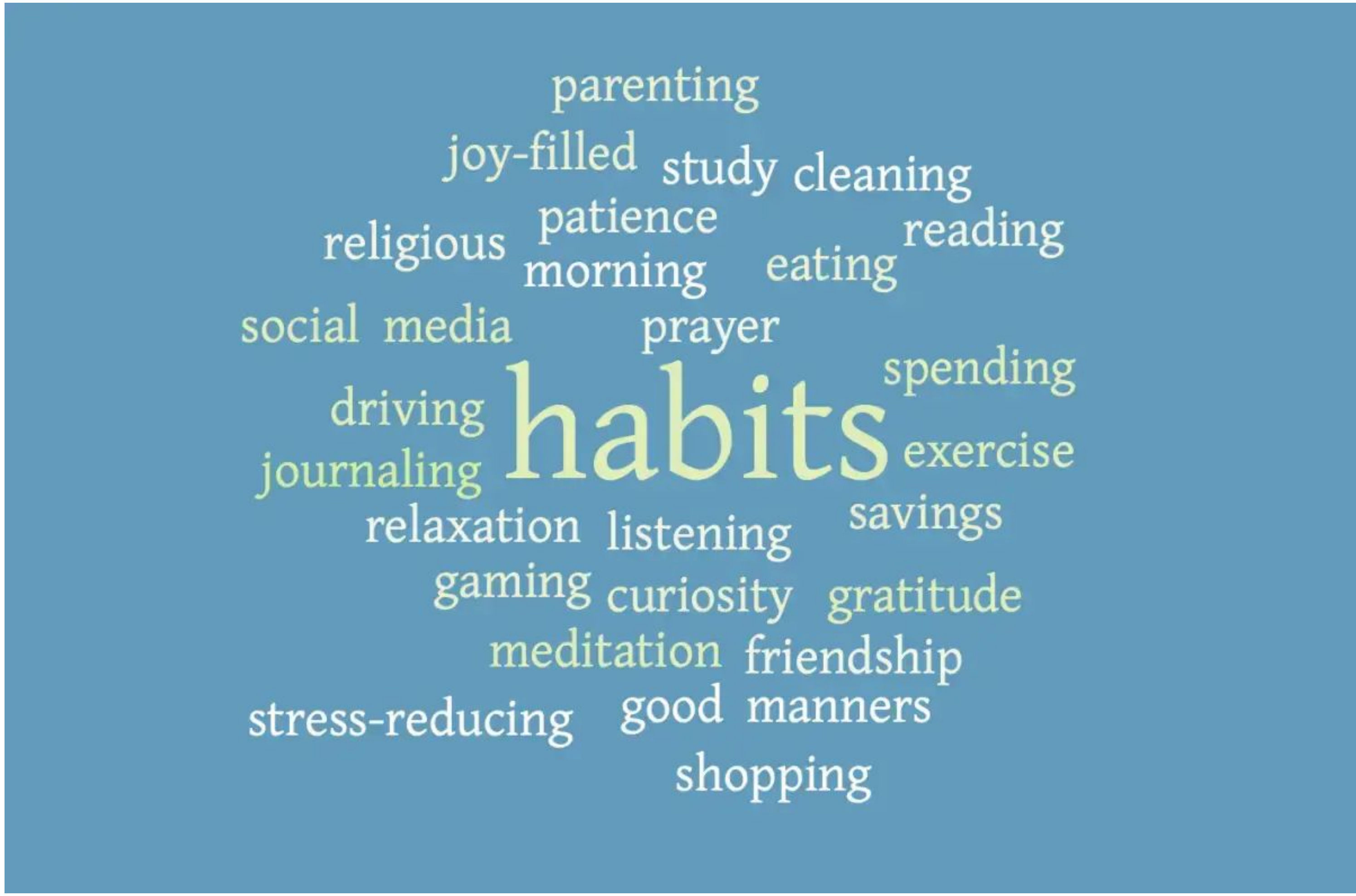
The filter slightly improves generalization, but the model maintains the same low precision.

Conclusion on Model Results

Inductive Miner excels in fitness and generalization but has low precision, which could limit its use

Heuristic Miner offers a good balance between fitness, precision, and complexity

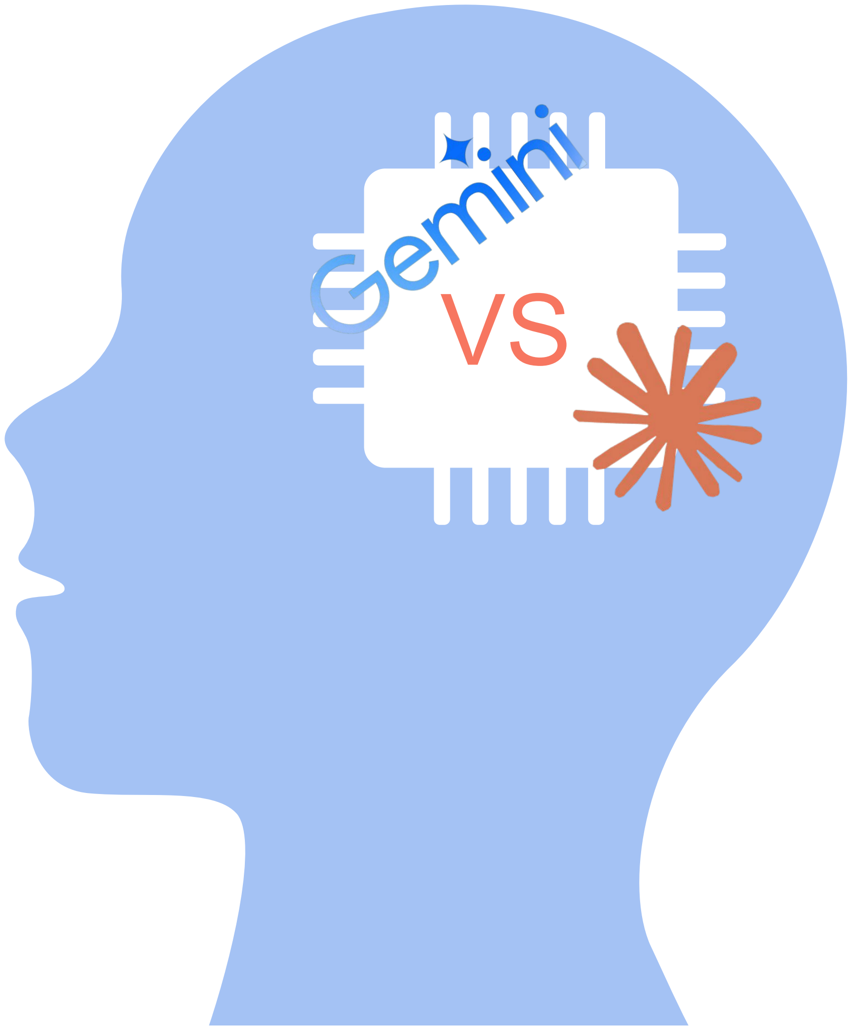
Alpha Miner, while generalizing well, has low fitness and precision, making it less effective than the other models



DATA AUGMENTATION

Gemini vs Claude

To increase the amount of available data, data augmentation was performed using Large Language Models (LLMs). Specifically, two models, Gemini and Claude, were chosen. These models were tasked with expanding the existing set of activities by simulating behaviors that align with the established patterns.



Original dataset

Model	Fitness	Precision	Generalization	Simplicity
Alpha Miner (Unfiltered)	0.38	0.02	0.89	1.0
Alpha Miner (Filtered)	0.66	0.02	0.89	1.0
Heuristic Miner (Unfiltered)	0.95	0.31	0.69	0.51
Heuristic Miner (Filtered)	0.94	0.31	0.71	0.52
Inductive Miner (Unfiltered)	0.98	0.15	0.87	0.64
Inductive Miner (Filtered)	0.98	0.15	0.88	0.64

Augmented dataset with Claude

Model	Fitness	Precision	Generalization	Simplicity
Alpha Miner (Unfiltered)	0.39	0.06	0.90	1.0
Alpha Miner (Filtered)	0.65	0.06	0.90	1.0
Heuristic Miner (Unfiltered)	0.94	0.33	0.69	0.51
Heuristic Miner (Filtered)	0.94	0.33	0.70	0.52
Inductive Miner (Unfiltered)	0.98	0.14	0.88	0.64
Inductive Miner (Filtered)	1.00	0.07	0.83	0.61

Augmented dataset with Gemini

Model	Fitness	Precision	Generalization	Simplicity
Alpha Miner (Unfiltered)	0.41	0.10	0.90	1.0
Alpha Miner (Filtered)	0.67	0.05	0.89	1.0
Heuristic Miner (Unfiltered)	0.93	0.37	0.62	0.50
Heuristic Miner (Filtered)	0.95	0.33	0.68	0.51
Inductive Miner (Unfiltered)	0.98	0.20	0.89	0.63
Inductive Miner (Filtered)	1.00	0.07	0.84	0.64

Results of Dataset Augmentation

The results from the two dataset variants showed small improvements in some areas and some declines in others. The models were able to generate only a few hundred extra examples compared to the original dataset. The filtered model did not show big improvements because many of the generated traces did not meet the required standards

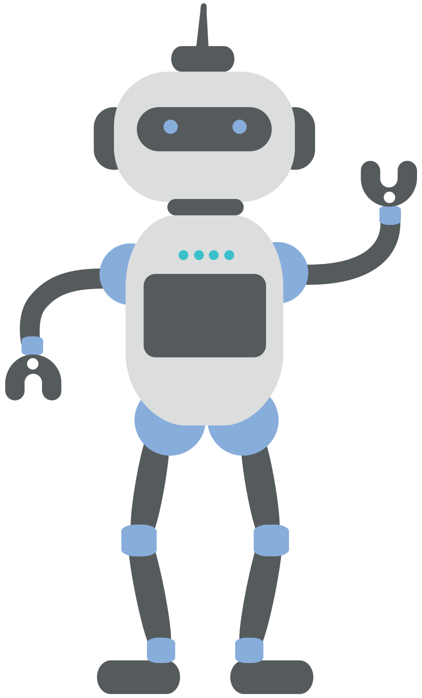
Limitations of LLMs in

Dataset Augmentation

Gemini and Claude proved to be ineffective in generating additional data. This is because the generated traces often lacked logic or did not represent the habits already present in the dataset.

LLMs had difficulty with this task because the dataset is closely related to the behavior of a specific individual, which is hard to imitate. The new data generated did not follow the same pattern as the existing ones and were too few to have a significant impact on the results.

Conclusions



The results from the first implementation are good, considering the dataset is not very large. A complex behavior pattern has been identified, but there is room for improvement in the future.

For better results in the future, we should increase the number of examples in the dataset and improve its quality by grouping similar activities into one class to make the implementation easier.

Thank You