On Fixing the Cruise Driverless Model

By

Robert Hink, Ph.D.

Table of Contents

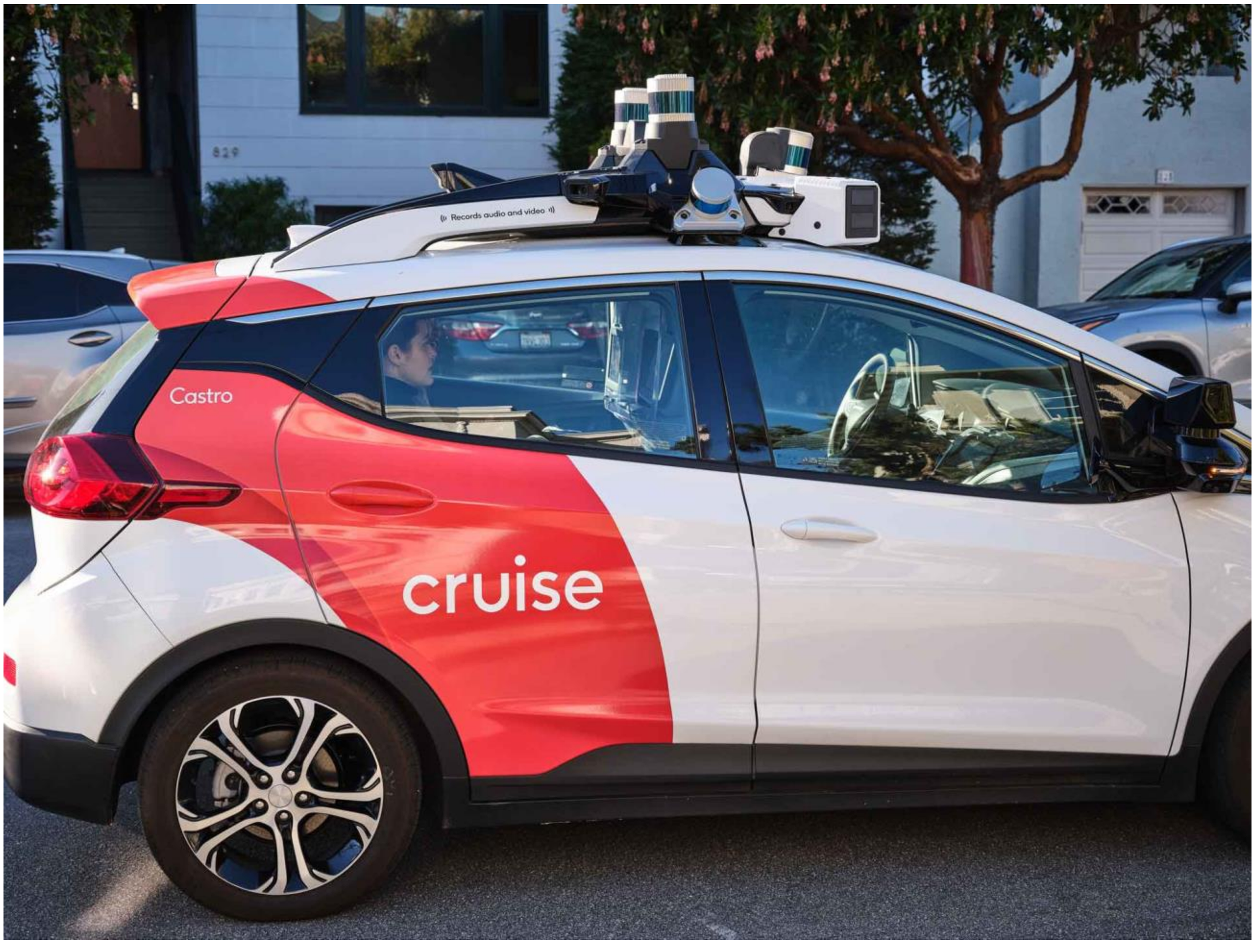

Figure 1: Cruise DAV with Passenger

Summary

Despite Cruise's excellent driverless-vehicle safety record benchmarked against human ridehail services (Human vs Cruise Ridehail Safety), the California Department of Motor Vehicles (DMV) has suspended Cruise's certification to operate Driverless Autonomous Vehicles (DAVs) in San Francisco. This suspension is due to recent incidents involving Cruise DAVs. This report suggests that the problem may be that the DAV Machine Learning (ML) model has not been exposed to infrequently encountered situations. As a consequence, on occasion, the DAVs make incorrect choices. Perhaps the simplest solution is the implementation of a Knowledge-Based System (KBS), an Artificial Intelligence (AI) technology that can incorporate human-driver knowledge to supervise the DAV's decision-making thereby correcting any errors.

Introduction

This report is intended to show that the performance of Cruise DAVs can be improved by incorporating knowledge or heuristics into their control of the vehicle. DAVs are strongly influenced by experience. For the most part, DAVs do quite well drawing only on experience. They do less well when they find themselves in exceptional situations. In short, they seem to have difficulty reasoning through complex and unusual situations.

This dichotomy is not unlike how humas behave. When performing some overlearned tasks, humans can go into autopilot. Driving is an example of such a task. When a situation arises that requires more attention, humans typically go into cognitive mode (i.e., requiring memory, learning, and reasoning) to deal with it (2 Types of Thinking).

Artificial intelligence (AI) has given us a technology to emulate the cognitive style of thinking. This technology is known as Knowledge-Based Systems (KBSs) (Knowledge-Based Systems).

KBSs can address the problems Cruise has experienced recently. This approach may be implemented expeditiously and inexpensively.

Incident

On August 17, 2023, a Cruise DAV carrying one passenger collided with a SFFD fire truck on the intersection of Turk and Market Streets (Incident 8/17/2023) in San Francisco. The DAV had failed to stop in the presence of an emergency vehicle approaching with lights flashing and siren blaring in the same lane going in the opposite direction. This action is in violation of the California Vehicle Code (CVC 21806). The signal light was green. The fire truck entered the intersection without slowing in violation of 8.A.451.05-D Internation Association of Fire Chiefs Driving Policy. It was later determined that both vehicles were at fault.

This incident serves as an example of an avoidable mishap.

Hypothesis

The incident cited above illustrates an occasion when the DAV's performance deviated from what an experienced human driver should do, namely pull over, if possible, and come to a stop. The fact that the light was green should be irrelevant to the human driver.

Cruise DAVs have a great deal of experience with green lights. In fact, almost invariably, a red light is followed by a green light. Almost equally frequently, green lights signal the vehicle to proceed.

Also, the presence of an emergency vehicle with siren sounding and lights flashing means the opposite (i.e., do not proceed). DAV's also have some experience with emergency vehicles.

When the two events, green light and emergency vehicle in the area, occur together, there is a conflict. Which event dominates? Our hypothesis is that the one which is more frequent or more highly trained wins. In this case, the green light meaning to proceed is chosen.

Methods

In this study, we trained a Random Forest Classifier (RFC) on simulated data. The data consisted of two features: Emergency Medical Vehicle (EMV) presence and Signal Light State (SLS). EMV could be either "present" or "absent"; SLS could be either "green" or "red". The label identifies the appropriate action: either "go" or "stop".

There are four combinations:

EMV	SLS	Action
absent	green	go
absent	red	stop
present	green	stop
present	red	stop

The absent-green combination is represented in about 60% of the instances; absent-red and present-red constitute the other 40% of the instances. The present-green combination does not appear in the dataset. This combination is the unseen probe that mimics the incident from August 17.

The dataset of 1,000 instances was split at 33% leaving 670 trials for training and 330 trials for testing.

Instance Counts

Absent-Green	**Absent-Red**	**Present-Green**	**Present-Red**
596	200	0	204

Results

The model learned the classifications perfectly for the three feature combinations presented. The accuracy score was 100%.

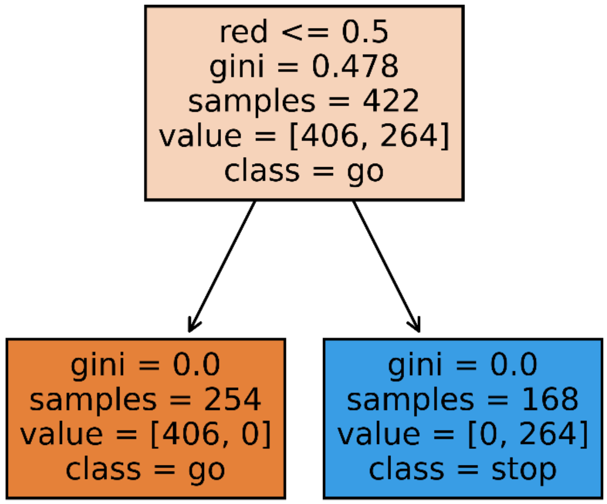

Figure 2: Random Forest Classifier

Figure 2 shows the first decision rule is to go to the left node if the signal's redness is less than 0.5. Since "red" can only be 0 or 1, the rule says to go left if redness is 0 (i.e., the signal is green). The value attribute shows that the rule classifies 406 cases as "go" (i.e., signal is green) and 264 cases as "stop" (i.e., signal is red). The base rate then favors "go". The fact that the Gini index is 0 at both leaf nodes says that the classification rule is perfect, no impurity.

Please note that the classifier is able to divide the data based upon signal redness since there are no present-green instances in the data. In other words, any redness means stop; there is no counter example of present-green in the dataset which also should evoke a "stop" response.

Figure 3 shows how the instances are distributed as hits, correct rejections, false alarms, and misses (Confusion Matrix). Hits and correct rejections appearing in the top left, bottom right cells, respectively, are the only cells populated; there are no errors of any kind occurred.

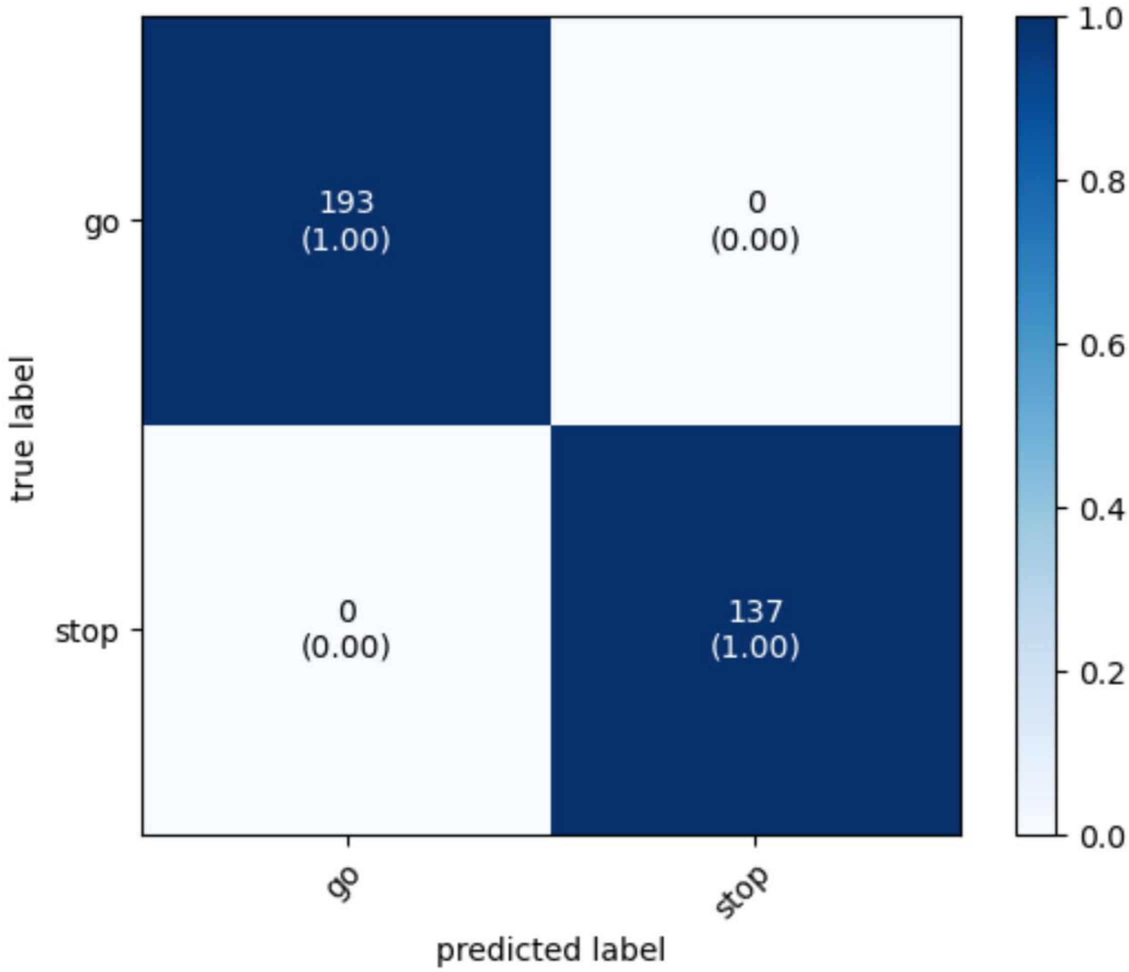

Figure 3:Confusion Matrix

Finally, we asked our model to predict the action when present-green is presented. The response was “go”. This is how the DAV reacted at Turk St. at Market St. on August 17 when it collided with the fire truck.

We trained a new model where the four combinations of feature values are more closely balanced.

Instance Counts (balanced)

Absent-Green	Absent-Red	Present-Green	Present-Red
252	223	251	274

This dataset has “green” in 503 instances and 497 “red” instances.

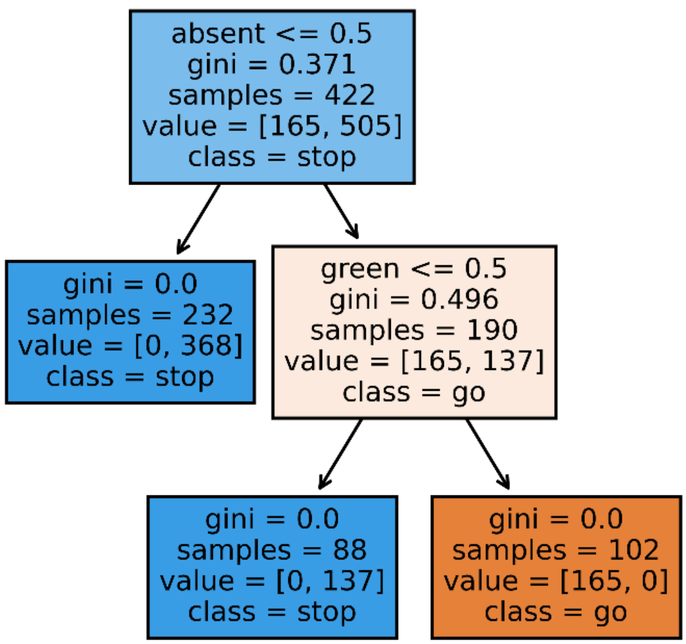

Figure 4: Random Forest Classifier (balanced)

Figure 4 displays the decision tree with balanced data. Now the top node tests for EMV presence or absence. If present, the tree goes to the left node recommending “stop”. If absent, the classifier tests for signal greenness. This subtree is similar to that using the unbalanced data. If not green (i.e., red), it passes to the left terminal node recommending “stop”. Otherwise, it passes to the right terminal node recommending “go”.

The Confusion Matrix (c.f. Figure 5) shows no erroneous classifications. When asked to predict the action given the test probe of present-green, the model responded “stop”, the correct action. Despite the slight probabilistic edge given to “green” in the balanced dataset. The model made the correct choice simply because it was trained to do so.

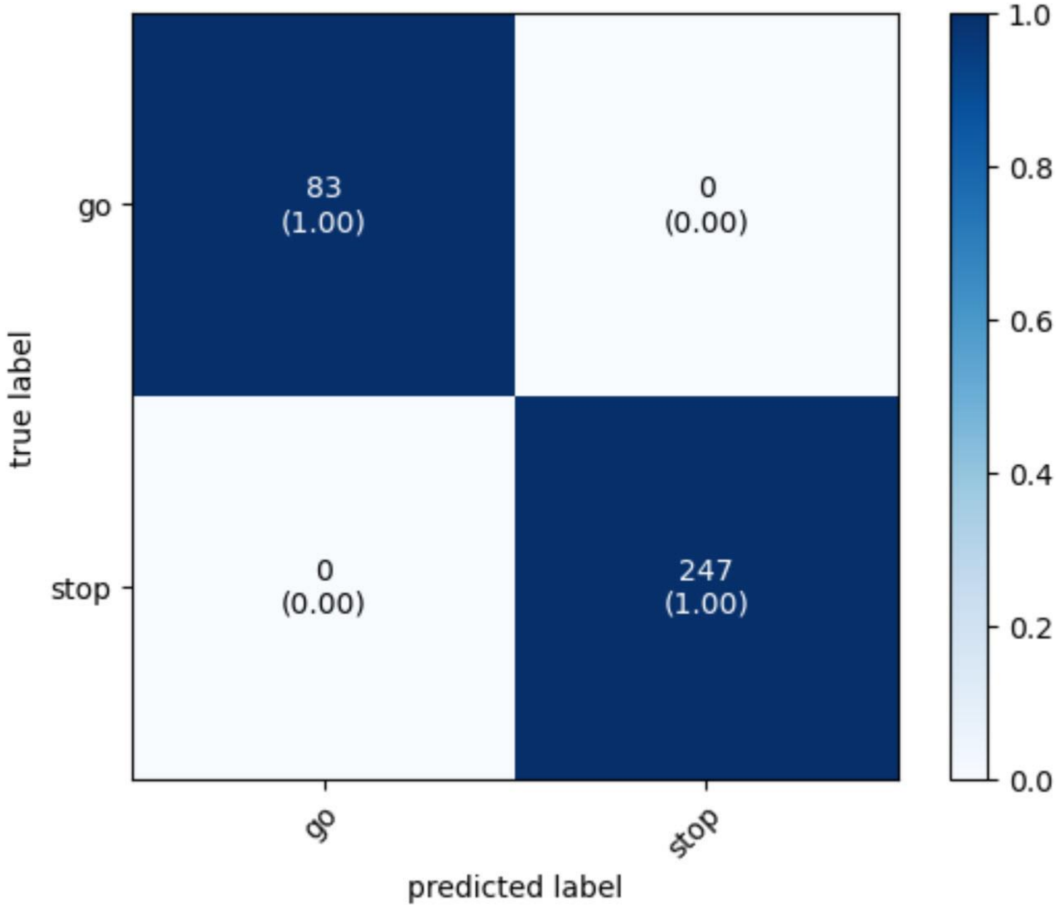

Figure 5: Confusion Matrix (Balanced)

Discussion

The comparison of the unbalanced and balanced models suggests that Cruise may be able to correct the DAV problems with more comprehensive data, perhaps simulated data.

That approach may be successful. Other companies have had limited success, though. Tesla, for example, relied heavily on simulated data to implement their Fully Self Driving feature (FSD), but for urban driving, many issues persist (Tesla FSD Critique). Also, from personal experience, I never saw my Model S take the proper exit from a freeway when driving in FSD mode. That failure suggests that Teslas when in FSD mode on the freeway are actually in Autopilot Mode. FSD mode seems to be more appropriate for street driving. The Teslas have difficulty transitioning between the two modes.

There is an alternative approach. That would be to employ a KBS as a supervisor. In situations where it disagrees with the intended action, it could intervene and take control. Otherwise, it would pass through the intended action.

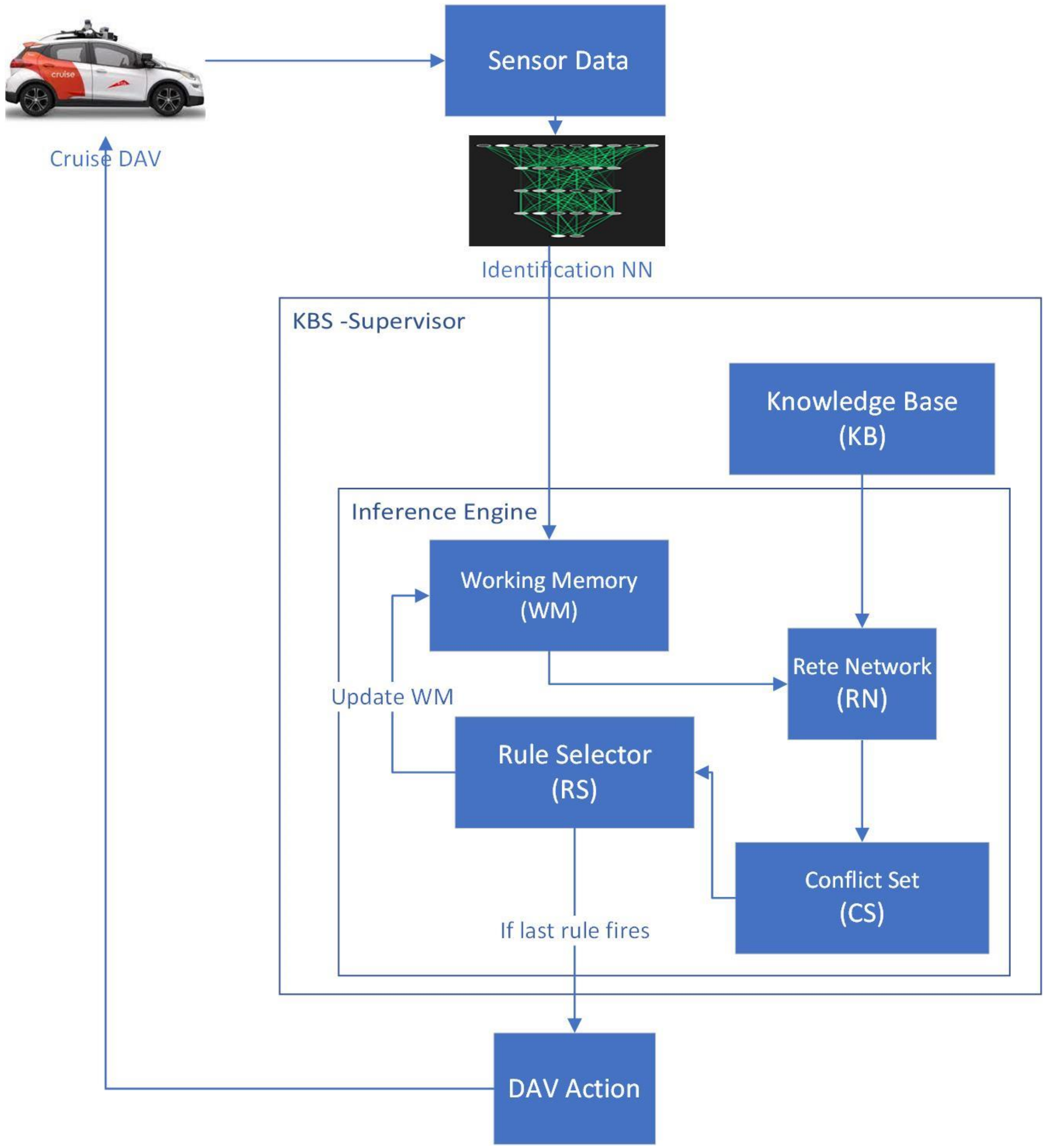

Figure 6: KBS - Supervisor

Figure 6 shows the elements of KBS - Supervisor. There are 2 main components to the KBS architecture: the Knowledge Base (KB) and the Inference Engine (IE). The KB is the repository of the if-then rules, also called Production Rules (PRs), and the IE contains the apparatus for applying those rules to specific situations.

The IE has Working Memory (WM) which contains Working Memory Elements (WMEs), objects describing the state of the environment. The Rete Network (RN) is an indexing algorithm for matching the WMEs to the test(s) specified in the if-part of the PRs. It ensures that tests matching WMEs against conditions from PRs are performed only once thereby improving efficiency.

On a given pass of the IE, the RN runs tests of the WMEs against the PRs in the KB. The PR(s) that are eligible to fire are passed to the Conflict Set (CS). The Rule Selector (RS) uses a set of criteria to select a single PR to execute (i.e., to fire). These criteria might be to select the PR with the most conditions or choose the PR with the highest priority. The RS also removes the selected PR from the CS. This action prevents the same rule from firing again to the same WME(s) making the PR "refractory".

If the PR that fired creates new WME(s), the IE starts the process over again looking for new rule matches. In this way, the IE chains the PRs together forming a line of reasoning.

When the IE can find no more PRs to fire, the process terminates until more WMEs are inserted into WM. The DAV action should come from the last PR to fire.

We applied this approach to our unbalanced model that chose to "go" with "present-green". The readout below shows that the KBS noticed the presence of the EMC. It fired rule p-0 which prints the message, "stopping for emv".

```
Decoded Features: EMV-Fact(emv=present), Signal-Fact(signal=green)
stopping for emv
```

[160]:

```
Productions:
p-0: IF V(emv) << Fact(emv=present) THEN emv_present(net, emv)
p-1: IF V(signal) << Fact(signal=red) THEN sig_red(net, signal)
p-2: IF AND(V(emv) << Fact(emv=absent), V(signal) << Fact(signal=green)) THEN emv_absent_and_sig_green
(net, emv, signal)

Facts:
f-0: Fact(signal=green)
f-1: Fact(emv=present)

WMEs:
(f-0 ^signal green)
(f-1 ^emv present)
(f-1 ^__fact_type__ <class '__main__.Fact'>)
(f-0 ^__fact_type__ <class '__main__.Fact'>)
```

Only 30 lines of new code were needed to correct the error. Moreover, KBSs can explain their reasoning by displaying the sequence of PRs that fired and why they did so making them much easier to debug and maintain than ML models.

The fact that the model classifies the training data perfectly is a red flag that the model is overfit (Overfitting). Overfitting means the model cannot make reasonable guesses when presented with unknown and previously unseen data. [1]

Besides overfitting, data leakage (Data Leakage) also may be a problem. In our case, the test data and the training data are composed of the same combination of feature values. The result is good accuracy, but performance is poor with never-before-seen data.

In fact, we observed precisely that effect.

[1] Please note that we made no attempt to regularize our model. Our objective was to simulate the DAV's behavior from Incident 8/17/2023 and not to build a generalizable model.

There is one final point of note concerning the model: the forest consists of a single tree. Random forest classifiers employ a bagging strategy (Bagging). The idea is that multiple decision trees trained on different samples of the dataset produce better estimates when the action selected wins an election where each tree casts a vote for its preferred class. In our case, 10 trees were requested, but only 1 was needed. This finding indicates that the problem we presented to the RandomForestClassifier was simple, much simpler than controlling a DAV.

One may argue that the results presented here cannot really be compared to Cruise's model. The problem here is over-simplified compared to that of autonomous driving. Our model is overfit and trained with data leaks. I doubt this is the case for the Cruise model. These are all valid points. Nevertheless, our analysis highlights a potential issue facing all DAV ridehail companies: DAVs only know about what they have experienced, and they have not experienced everything that may occur.

I, for one, would like to see those Cruise DAVs roaming the streets of San Francisco again looking for fares. The KBS approach may be the best way to achieve that end.

Conclusion

DAVs are driven by data not knowledge. In other words, DAVs act according to what they have been shown only. Experience may be sufficient for predictable environments, but not so when the unpredictable occurs.

We claim that knowledge is needed to cover the unpredictable.

A KBS, using heuristics and the regulations from state vehicle codes, may inform the DAV what it should do despite its inclination to do something else.