JOHNS HOPKINS
INSTITUTE *for* ASSURED AUTONOMY

RADICS:

Runtime Assurance of Distributed Intelligent Control Systems

Brian Wheatman, Jerry Chen, Tamim Sookoor, and **Yair Amir**

www.dsn.jhu.edu/radics/

Why Assure AI

- AI systems are optimized for the average case.
  - They have a long tail of edge cases that can lead to failures
  - Ideally, we would get the benefits of AI without the cost of these edge cases
- Reinforcement learning (RL) algorithms are difficult to reason about and have non-intuitive behavior.
  - RL algorithms break in nonintuitive ways due to phenomena such as reward hacking and specification gaming
  - These algorithms often have a fat tail of edge cases which can never be fully trained away
- We introduce Runtime Assurance of Distributed Intelligent Control Systems (RADICS)
  - RADICS combines an invariant-based Black-Box Monitor with a White-Box Monitor that evaluates the confidence of the machine learning algorithm.
  - These two monitors ensure correctness while maintaining good performance

Black-Box Monitoring

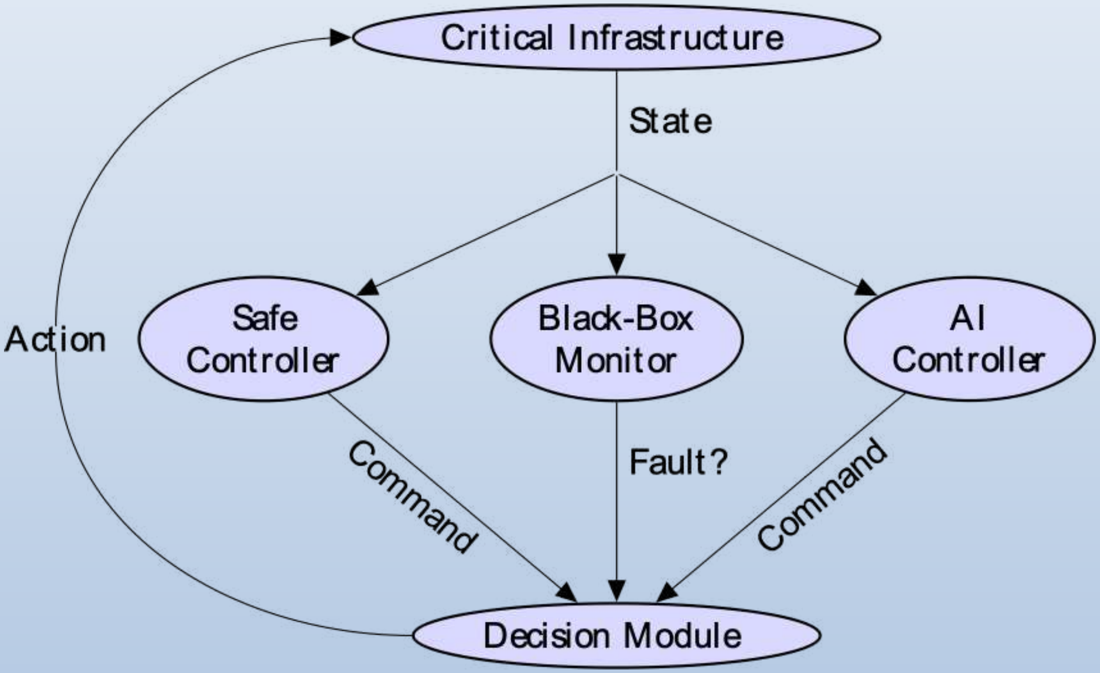
Black-Box monitoring is a standard approach to create dependable systems

The systems work roughly as follows:

State is collected and passed to a trusted controller, an AI controller and a monitor.

Each controller proposes an action

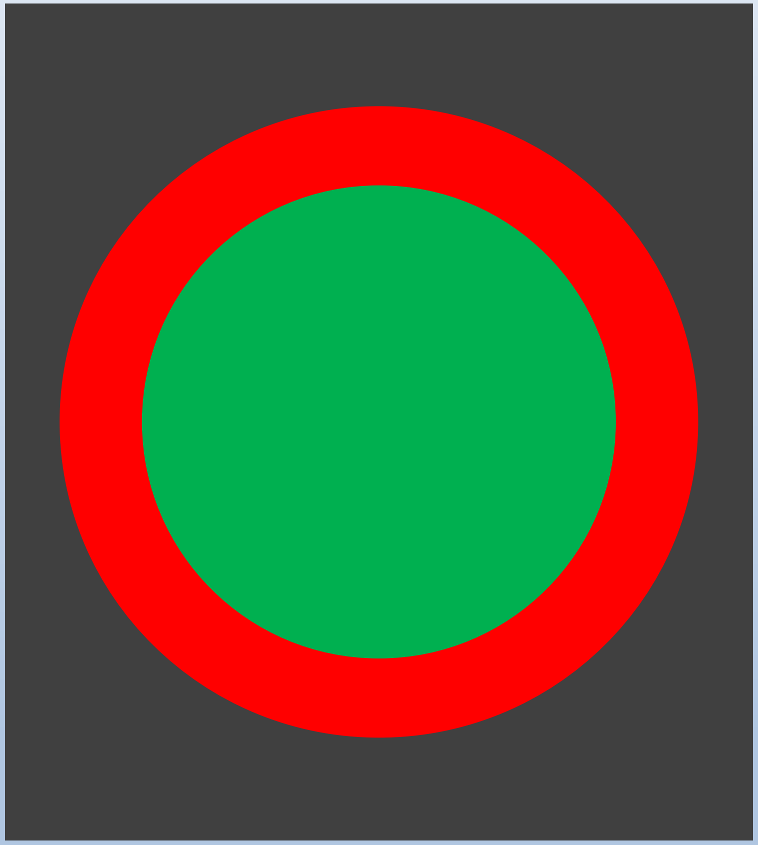
The decision module uses the output of the monitor to determine which action should be performed



Black-Box Monitoring

Black-Box monitoring can ensure system correctness as long as it takes some sufficiently long amount of time to go from the good state to the bad state

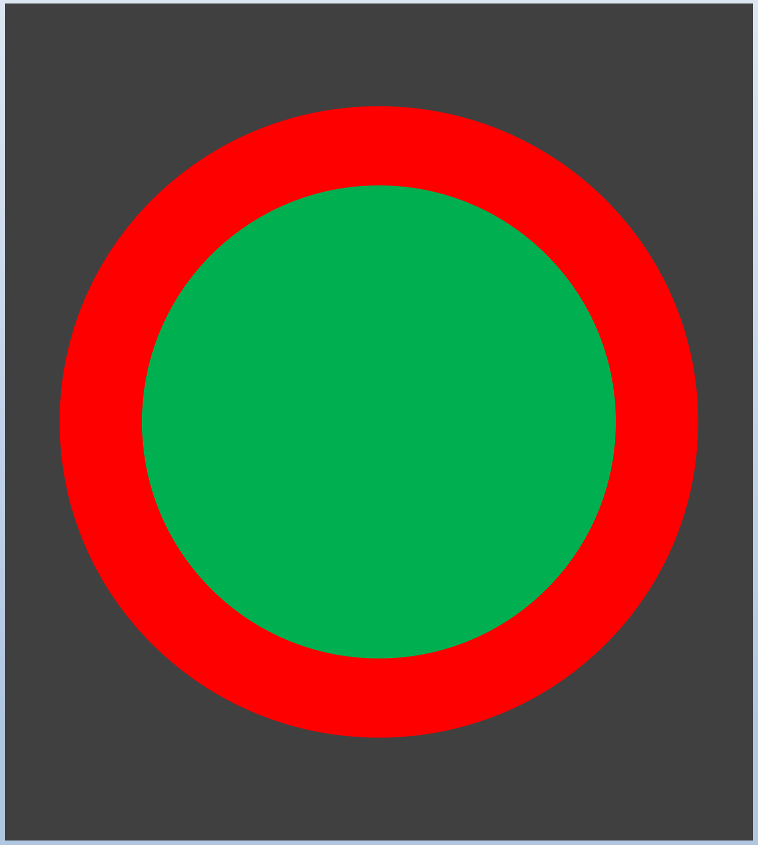
The monitor determine how far away from the bad state the system is in, and if it is close to a bad state the safe controller's action is performed until the state is sufficiently safe again.



Black-Box Monitoring - Limitations

While the decision to switch to the safe controller is straightforward the decision to switch back can be more complicated.

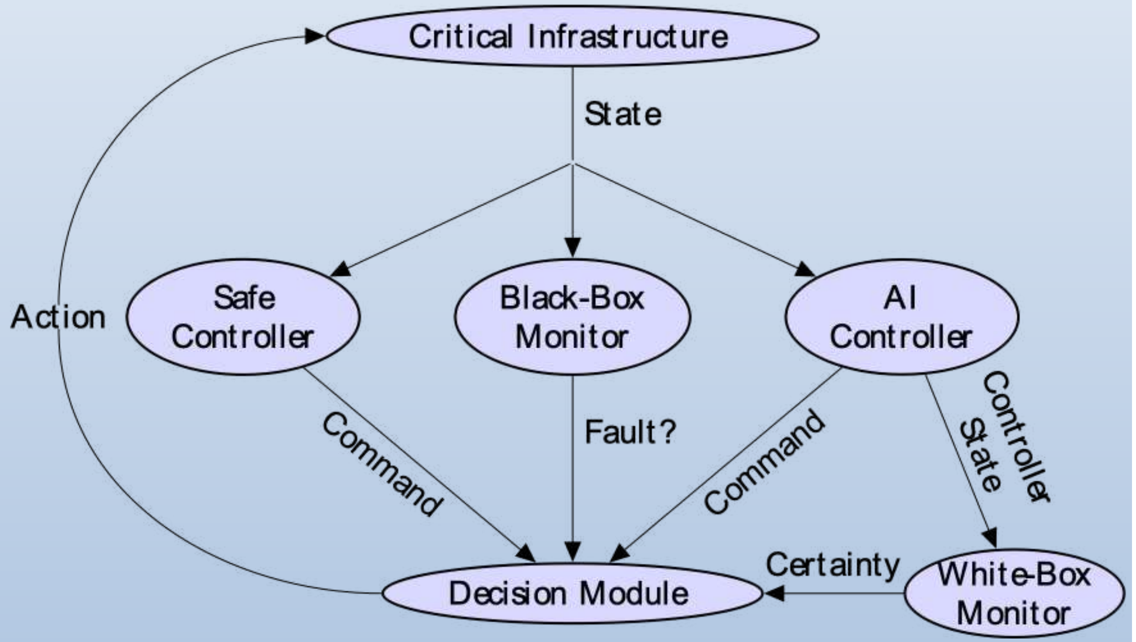
If the overall state of the world has not changed between when the system went from the green region to the red, we will likely oscillate between the controllers, and while we can stay "correct" performance will suffer



RADICS Approach

Adding a white-box monitor that determines how confident the AI controller is in its own proposed action can improve performance.

If the white-box monitor determines that AI controller is not confident in its action, it can switch to the safe controller sooner, or not switch back from the safe controller as quickly as just with the black-box monitor



RADICS Approach

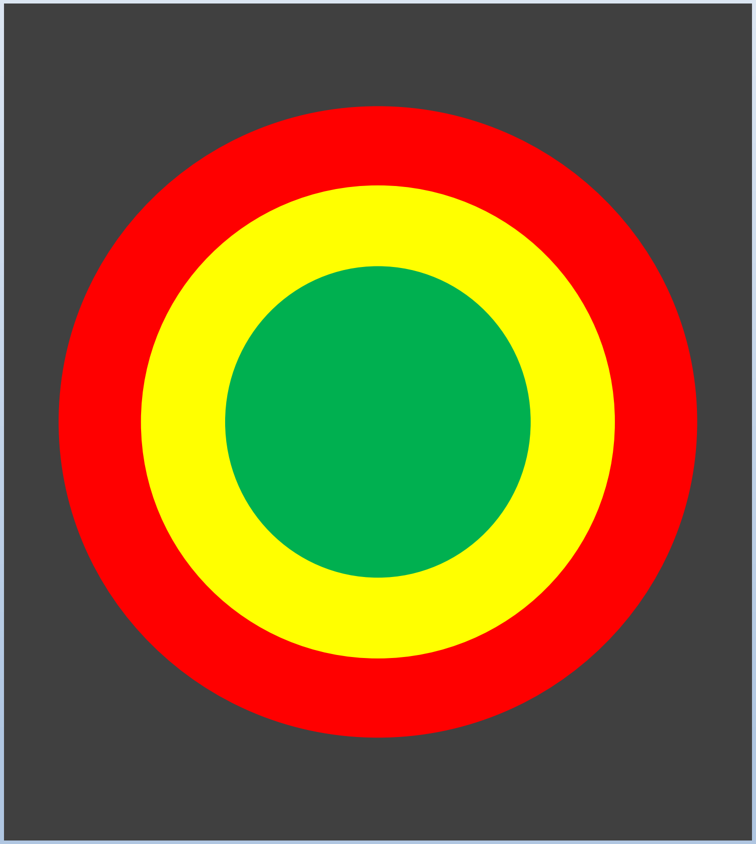
We still have all the safety of black box monitoring, but we introduce a new region to help with the overall system performance

The distance into the yellow region the decision module allows the system to go is dependent on the white-box monitor

RADICS Limitations

- RADICS only works for systems which have safe algorithms
  - If no static algorithm is known then we cannot assure the system

- RADICS also requires the time it takes to reach system failure to be non trivial



Traffic Control Testbed

We evaluate RADICS on a simple traffic control problem

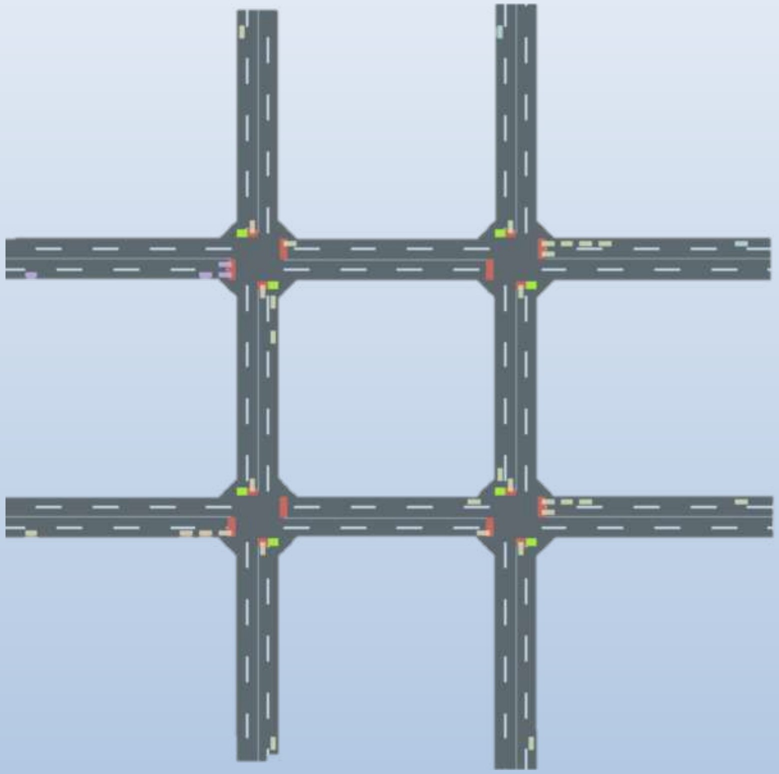
The goal is to control the four traffic lights to maximize the average speed of the cars

The traffic control testbed is based on the **SUMO** and **Flow** traffic simulation frameworks

Currently a monolithic 2x2 grid topology

Safe controller ensures that no car will wait more than once at each intersection

AI controller trained on a certain traffic pattern (500 vehicles per hour arriving on each incoming edge) for 80,000,000 steps representing 8,000,000 real-life seconds.



JOHNS HOPKINS
INSTITUTE *for* ASSURED AUTONOMY

Evaluation

Four strategies:

Safe Controller: a simple timer-based approach with synchronized traffic light changes ensuring that cars will wait at most once to go through an intersection

AI Controller: deep reinforcement learning controller using the Flow framework

Blackbox Monitoring: switch to the Safe controller when the average speed is below a certain threshold (4 m/sec); switch to the AI controller when the average speed is above a certain threshold (5 m/sec)

RADICS: use Whitebox Monitoring in addition to Blackbox monitoring. Our current Whitebox monitor simulates the system forward assuming the near future is similar to the near past to determine the confidence in the AI controller.

Videos of each strategy can be found at www.dsn.jhu.edu/radics/

Evaluation

The four strategies are evaluated using an elaborated scenario with three segments. The first segment deals with traffic that the AI controller trained for. The second segment introduces a new traffic pattern that is designed to trip the AI controller. The third segment returns to the normal traffic, evaluating the ability of the AI-based strategies to recover.

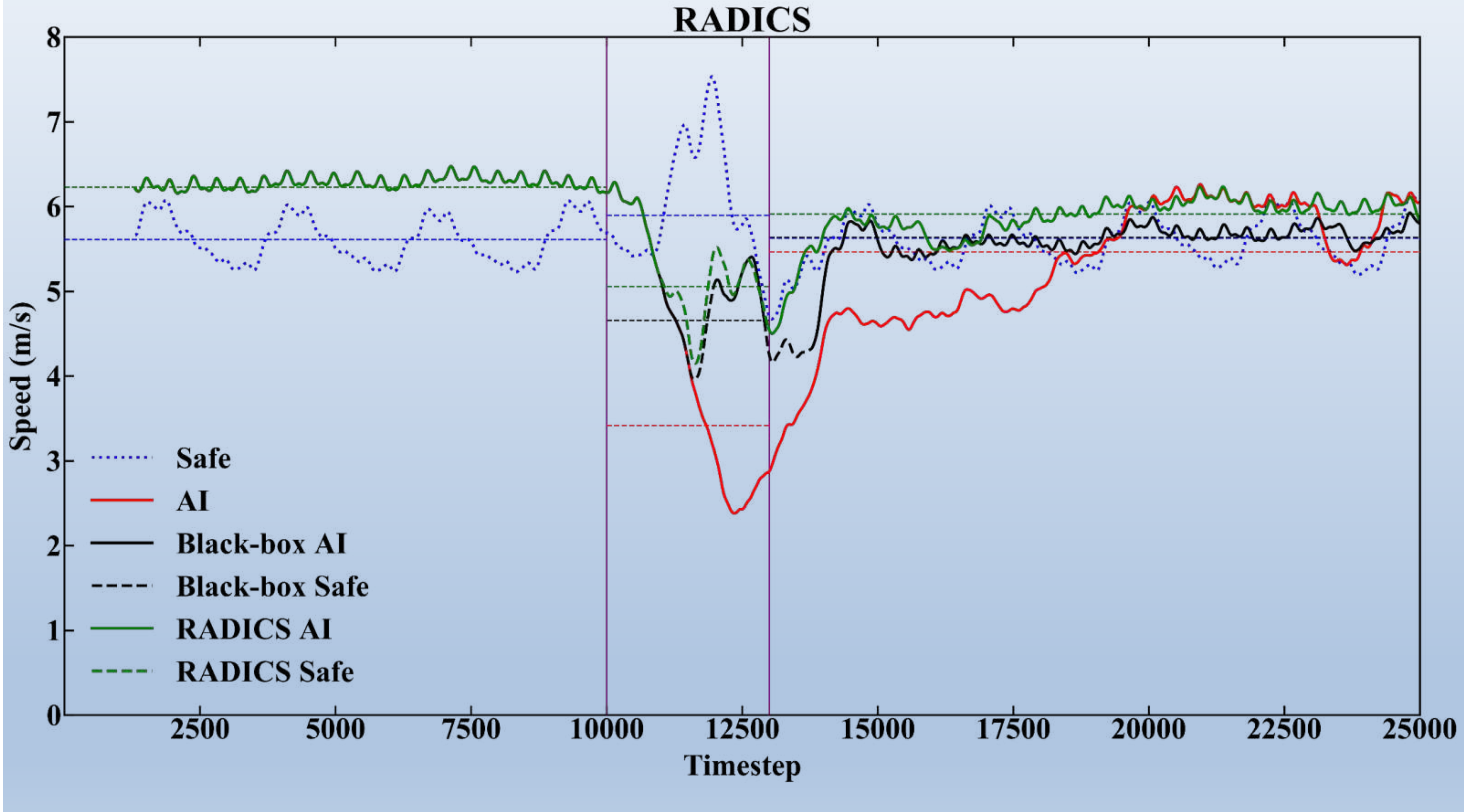
Specifically:

Segment 1: 1-10000 simulation steps (1000 seconds) with normal traffic the AI was trained on (500 vehicles per hour arriving on each incoming edge)

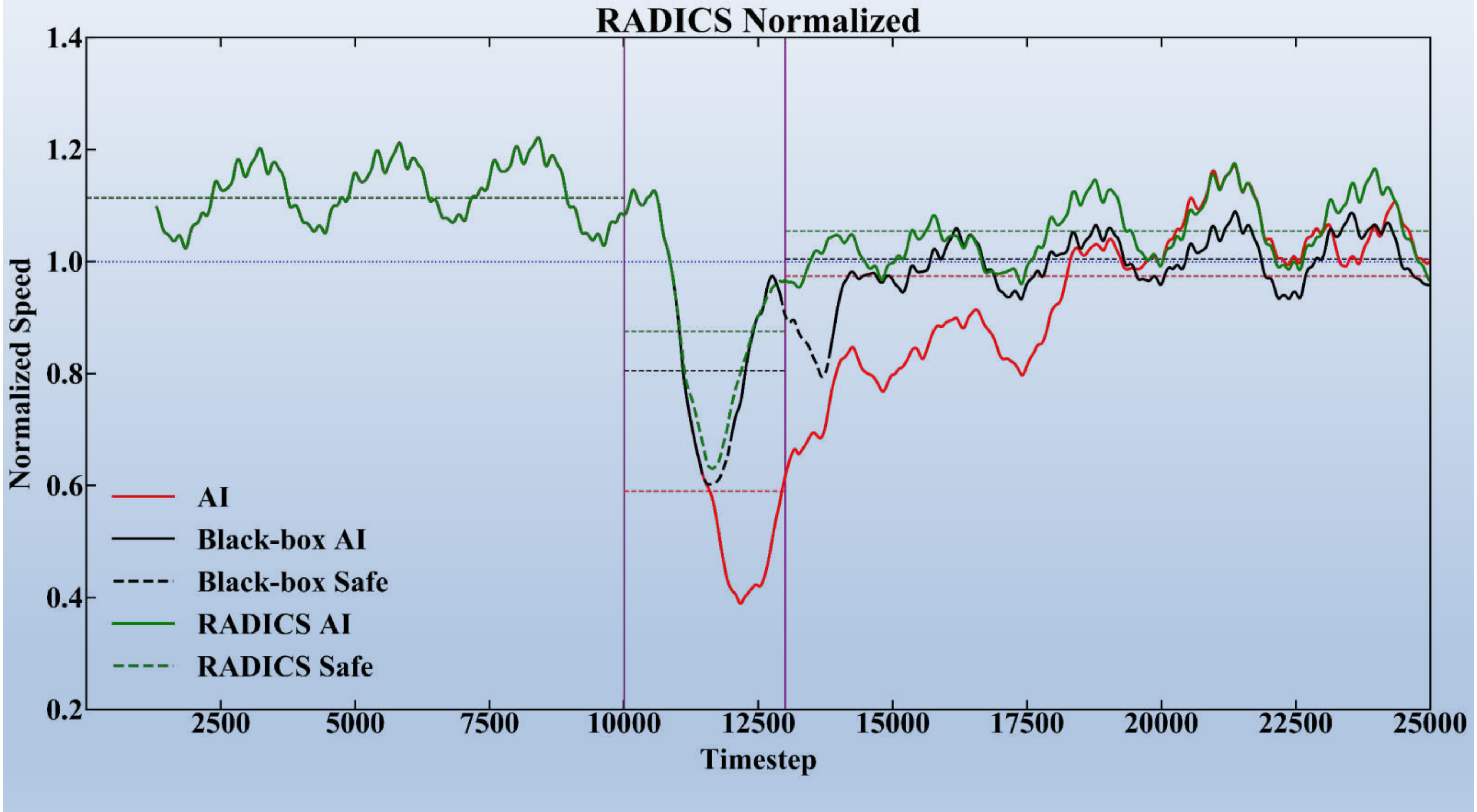
Segment 2: 10001 – 13000 simulation steps (5 minutes) with anomalous traffic that trips the AI in an unintuitive way (100 vehicles per hours arriving on each incoming edge except for one edge that gets, as before, 500 vehicles per hour)

Segment 3: 13001 – 25000 (20 minutes) simulation steps with normal traffic again

Evaluation


RADICS
Speed (m/s)
0
1
2
3
4
5
6
7
8
2500
5000
7500
10000
12500
15000
17500
20000
22500
25000
Timestep
Safe
AI
Black-box AI
Black-box Safe
RADICS AI
RADICS Safe

Normalized Evaluation



Evaluation

Segment 1: 1-10000 simulation steps (1000 seconds) with normal traffic the AI was trained on (500 vehicles per hour arriving on each incoming edge)

Segment 2: 10001 – 13000 simulation steps (5 minutes) with anomalous traffic that trips the AI in an unintuitive way (100 vehicles per hours arriving on each incoming edge except for one edge that gets, as before, 500 vehicles per hour)

Segment 3: 13001 – 25000 (20 minutes) simulation steps with normal traffic again

Controller	Overall	Segment 1	Segment 2	Segment 3
Safe controller	5.65	5.61	5.90	5.63
AI Controller	5.53	6.23	3.42	5.47
Black-Box	5.76	6.23	4.66	5.64
RADICS	5.94	6.23	5.06	5.91

Videos of each run can be found at www.dsn.jhu.edu/radics/

Conclusion and Future Work

RADICS ensures system correctness while still maintaining good performance.

While RADICS has several limitations, including the need to have a safe controller that provides an assured behavior / acceptable performance, it still can fit a wide range of applications

Future work involves running on more complicated scenarios, scaling to any traffic topology, and testing different styles of white box monitoring

We are also looking for other domains where these ideas may be applicable

For more information: www.dsn.jhu.edu/radics/