Assured Autonomy

Dr. Sandeep Neema/I2O

SafeAI 2019
Honolulu, HI
January 27, 2019
DARPA/I2O Overview
I2O Strategic Thrusts

**CYBER:** Win at cyber

**ANALYTICS:** Understand the world

**SYMBIOSIS:** Partner with machines
Win in the cyber domain (plan and act)

- Plan X
- Transparent Computing (TC)
- Enhanced Attribution (EA)
- Active Cyber Defense (ACD)
- Network Defense
- Cyber Hunting at Scale (CHASE)
- Harnessing Autonomy for Countering Cyberadversary Systems (HACCS)
- Active Social Engineering Defense (ASED)

Operate through cyber attacks (manage insecurity)

- Cyber Grand Challenge
- Extreme DDOS Defense (XD3)
- Edge-Directed Cyber technologies (EdgeCT)
- Mission-Oriented Resilient Clouds (MRC)
- Cyber Fault-tolerant Attack Recovery (CFAR)
- Rapid Attack Detection, Isolation and Characterization Systems (RADICS)
- Building Resource Adaptive Software Systems (BRASS)

Harden systems against cyber attack (build in security)

- High Assurance Cyber Military Systems (HACMS)
- Vetting Commodity IT Software and Firmware (VET)
- Space/Time Analysis for Cybersecurity (STAC)
- Mining and Understanding Software Enclaves (MUSE)
- Clean-slate design of Resilient, Adaptive, Secure Hosts (CRASH)
- SafeWare
- Leveraging the Analog Domain for Security (LADS)
- Cyber Assured System Engineering (CASE)
- Dispersed Computing

DISTRIBUTION A. Approved for public release: distribution unlimited
Trustworthy information

- Broad Operational Language Translation (BOLT)
- Deep Exploration and Filtering of Text (DEFT)
- Low Resource Languages for Emergent Incidents (LORELEI)
- Robust Automatic Transcription of Speech (RATS)
- Brandeis
- Media Forensics (MediFor)

Understanding

- Quantitative Crisis Response (QCR)
- Memex
- Big Mechanism
- XDATA
- Insight
- Causal Exploration
- SocialSim
- Active Interpretation of Disparate Alternatives (AIDA)
- Modeling Adversarial Activity (MAA)
- World Modelers
- Warfighter Analytics Using Smartphones for Health (WASH)
- Synergistic Discovery and Design (SD2)
- Data-Driven Discovery of Models (D3M)

Partnership with machines

- Communicating with Computers
- Probabilistic Programming for Advancing Machine Learning (PPAML)
- Explainable AI
- Assured Autonomy
Learning systems are essential for real-world autonomy

But, safety matters...

How can we assure that learning systems are safe and correct?
Traditional V&V approaches are not tenable... Traditional testing will require exorbitant time and money: 11B miles, 500 years, $6B - *Driving to Safety, RAND Corp. Report, 2016*

**Table 1. Examples of Miles and Years Needed to Demonstrate Autonomous Vehicle Reliability**

<table>
<thead>
<tr>
<th>Statistical Question</th>
<th>How many miles (years(^a)) would autonomous vehicles have to be driven...</th>
<th>Benchmark Failure Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) without failure to demonstrate with 95% confidence that their failure rate is at most...</td>
<td>275 million miles (12.5 years)</td>
<td>(A) 1.09 fatalities per 100 million miles?</td>
</tr>
<tr>
<td>(2) to demonstrate with 95% confidence their failure rate to within 20% of the true rate of...</td>
<td>8.8 billion miles (400 years)</td>
<td>(B) 77 reported injuries per 100 million miles?</td>
</tr>
<tr>
<td>(3) to demonstrate with 95% confidence and 80% power that their failure rate is 20% better than the human driver failure rate of...</td>
<td>11 billion miles (500 years)</td>
<td></td>
</tr>
</tbody>
</table>

\(^a\) We assess the time it would take to compete the requisite miles with a fleet of 100 autonomous vehicles (larger than any known existing fleet) driving 24 hours a day, 365 days a year, at an average speed of 25 miles per hour.

“The notion that autonomous systems can be fully tested is becoming increasingly infeasible as higher levels of self governing systems become a reality... the standard practice of testing all possible states and all ranges of inputs to the system becomes an unachievable goal. Existing TEV methods are, by themselves, insufficient for TEV of autonomous systems; therefore a fundamental change is needed in how we validate and verify these systems.”
- *OSD TEV&V Strategy Report, May 2015*
Develop rigorous design and analysis technologies for continual assurance† of learning-enabled autonomous systems, in order to guarantee safety properties in adversarial environments

†assurance: a positive declaration intended to give confidence
Assurance Challenges

Implementation

Verification

Requirements

Source: xkcd.com/1838/
Safety Assurance for Non-Learning vs. Learning Systems

Non-Learning System

Learning-Enabled System (LES)

Unsure Region

Known Unsafe Region

Unknown Unsafe Region

Conceptual State-space
Focus Areas

Design-Time
- Specifications
- Verification
- Simulation
- Testing

Operation-Time
- Monitoring
- Safety-Aware Learning
- Dynamics Assurance Cases
- Assurance Measure
Simulation vs. Verification

Waypoint Navigation using Neural Networks

Counter-Examples Found Using Formal Verification

Source: HRL
Verifying Deep Neural Networks (DNNs)

Airborne Collision-Avoidance System X (ACAS X)

Inputs are from sensor readings about the environment

Baseline: Generate a look-up table that maps each state (120M) to a score for each of 5 advisories
- Strong left (SL); Weak left (WL); Strong right (SR); Weak right (WR); Clear of conflict (COC)

FAA considering alternate implementation that uses deep neural networks

8 fully-connected layers w/Rectified Linear Unit (ReLU) activations; 600K weights

Output is an advisory: Strong left (SL); Weak left (WL); Strong right (SR); Weak right (WR); Clear of conflict (COC)

Can we provide assurance for this DNN with respect to correctness?
Verifying Deep Neural Networks (DNNs)

Reluplex: tool to formally verify properties of DNNs
- Verified DNNs an order of magnitude larger than previously possible
- ...but, only worked with DNNs containing Rectified Linear Unit (ReLU) activation functions

Recent Results: Extended Reluplex to develop a new DNN verification tool (Marabou) that provides:
- Improved scalability: new custom linear programming engine
- Enhanced coverage: support for max pooling & convolution layers
- Improved usability: Python interface with support for TensorFlow
Verifying Systems containing DNNs

Sensor Input → Environment → Automatic Emergency Braking System → Controller → Plant

Spec: $G (\text{dist (ego vehicle, env object)} > \Delta)$

Goal: Verify that the system meets the Spec

Verifying Systems containing DNNs

CPS Analyzer: Identify Region of Uncertainty (ROU)

ML Analyzer: Explore Region of Uncertainty


This misclassification may not be of concern

But this one is a real hazard!
Adaptive Cruise Control System

Ego (follower) car (1) tracks velocity of lead car and (2) maintains a safe distance

**Specification:** \( D_r(t) \geq \frac{D_{safe}(t)}{2} \), where \( D_{safe}(t) = D_{default} + T_{gap} \times V_{ego}(t) \)
Specification-Guided Verification for Neural Networks in Closed-Loop Systems

HyST: https://github.com/verivital/hyst

nnv + nnmt https://github.com/verivital/nnv

Output reachable set computation

1-2 orders of magnitude less runtime and memory using Specification-Guided Partitioning
Simulation-based Verification
SCENIC - Scenario Description Language

• *Scenic* is a **probabilistic language** for modeling environments of learning-based CPS
• A Scenic program defines *distributions over environment scenes*

```python
from gta import Car, curb, roadDirection
ego = Car
spot = OrientedPoint on visible curb
badAngle = Uniform(1.0, -1.0) * (10, 20) deg
Car left of (spot offset by -0.5 @ 0), facing badAngle relative to roadDirection
```

• Several use cases, including generating data, verification/simulation, capturing environment assumptions, etc.

Example scenario: a badly-parked car

Source: UC-Berkeley

D. Fremont et al., “Scenic: Language-Based Scene Generation”, 2018
Focus Areas

**Design-Time**
- Specifications
- Verification
- Simulation
- Testing

**Operation-Time**
- Monitoring
- Safety-Aware Learning
- Dynamics Assurance Cases
- Assurance Measure
Assurance Measure

Assurance Monitoring Based on Inductive Conformal Prediction

- *Nonconformity measure* evaluates the degree to which a new example disagrees with the LEC input-label relationship of old examples

- Evaluate how different is a given example compared to a set of calibration examples (it is assumed that the examples are i.i.d.)

- Compute a prediction interval such that the actual output belongs to the interval with high confidence

Keep car on track [-1..1]
Addressing data distribution shift

what if \( p_{\text{test}}(x) \neq p_{\text{train}}(x) \)?

Idea 1: fit \( p_{\text{train}}(x) \), and test for new \( x^* \) if \( p_{\text{train}}(x) \) is small

how to model \( p_{\text{train}}(x) \)?

VAE (Kingma & Welling ’13)

\( z \sim \mathcal{N}(0, I) \)

\( x \sim p(x|z) \)

Better idea: transform \( x^* \) into an in-distribution image!

\( x \sim p_{\text{train}}(x) \)

\( x^* \sim p_{\text{test}}(x) \)

Source: UC-Berkeley

McAllister, Kahn, Clune, Levine (in submission)
Addressing data distribution shift

use these estimates to compute variance

\( x^* \sim p_{\text{test}}(x) \)

Source: UC-Berkeley

McAllister, Kahn, Clune, Levine (in submission)
Platforms and Phase I Challenge Problems
Integration and Experimentation Platforms

Boeing (UAV)

Northrop (UUV)

TARDEC (AMAS)

ONR (Sea Hunter)*

Source: Boeing

Source: Northrop Grumman

Source: TARDEC

*anticipated
Centerline Tracking:
LEC produces inputs to navigation and steering

Object Detection:
Perception LEC produces inputs to vehicle control
Northrop Grumman Challenge Problem

Underwater Infrastructure Survey and Surveillance Missions

With LECs
TARDEC Challenge Problems

**Slope Estimation:**
Assure an LEC that estimates the slope of the terrain based on LIDAR point cloud data.

**Stability Control:**
Assure an LEC that controls the vehicle: a) slippery surfaces; b) controlled skid & steer, sharp curves; and c) a mix of improved roads and off-roads/fire roads.
• Learning technologies paramount to achieve higher levels of autonomy

• We are developing foundational techniques for assurance of learning-enabled systems

• Evaluation and transition through Boeing, NG, and TARDEC Autonomous Platforms