AFRL SUNY Trusted AI Challenge Series

Topic #1: Verification of Autonomous Systems

Sponsor: AFRL

Objective: The verification of autonomous systems has largely relied on formal and heuristic testing approaches to verify simple behaviors such as safety properties (e.g. the avoidance of bad states). This topic seeks novel approaches for the verification of autonomous systems that learn and interact in their environments subject to more complex behavioral properties.

Description: Formal approaches to verification have enabled a sense of reliability and resiliency in systems whose dynamics are well-understood and whose complexity is reasonably bounded. However, the advent of autonomous agents powered by deep neural networks challenge these assumptions. Indeed, the dynamics of deep neural networks, while understood, are not interpretable in a form that is amenable to traditional verification algorithms. Some progress has been made for the formal verification of deep neural networks. For example, the use of linear programming and branch-and-bound techniques to mathematically model and verify properties of restricted neural networks of modest size at inference time has shown promise. However, the current literature focuses on verifying basic properties such as safety (i.e. avoid a given set of outputs), reachability (i.e. ensure a given output is observed at some point), and robustness (e.g. bound the proportion of misclassifications). While these properties provide a good starting point, the richness of properties available in traditional verification that come in the form of various logics such as Linear Temporal Logic (LTL) and the Mu-calculus is largely missing. A simple example specification that exists outside of the aforementioned properties is "ensure proposition A holds until B is reached" (e.g. avoid hostile areas of the map until ammo is replenished). We seek solutions in this direction for the verification of learning agents subject to a richer language of specifications. These solutions may be applied at design and/or test time of the underlying learning model, during the learning and/or inference phases of the learning algorithms, and can be within the context of various learning areas, such as data analytics (e.g. classification and regression) and decision-making (e.g. reinforcement learning and planning).

Unlike formal approaches to verification, some heuristic approaches to verification focus on creating behavioral and edge tests to evaluate the output of a system given specific inputs. Recently, these concepts have been adapted for static machine learning tasks in Natural Language Processing. Unfortunately, unlike a static machine learning task, an autonomous agent's environment is ever changing, and it is infeasible to design tests for every feasible input. Although behavioral testing will still be an important aspect of the verification pipeline, it must overcome unique challenges. Particularly concerning is that, 1) undesirable behavior may only

be exhibited after a large number or complicated sequence of events, and that, 2) undesirable behavior may appear stochastically as autonomous systems are not deterministic. The need for this is intuitive and well-documented as evidenced by the DARPA Assured Autonomy program, the Guarding AI Against Robustness Deception (GARD) program, and the AFOSR Agile Test and Evaluation portfolio of investments.

Guidance: Prospective performers should develop or adapt a formal verification or heuristic testing approach to verify autonomous systems subject to complex behavioral properties. Complex, in this sense, entails going beyond reachability and simple robustness properties and may entail the specification of behaviors to be derived from existing logics, such a Linear Temporal Logic (LTL) or Computation Tree Logic (CTL), among others. Potential solutions include the use of linear programming by representing the autonomous agent as a simple neural network that can be reasoned over as a simplex, the use of whitebox and differential testing by leveraging neuron coverage as an analogue to traditional code coverage metrics in software testing, or other approaches. The performers must clearly document any assumptions made on the autonomous system model and its learning and inferencing dynamics, to include datasets or simulation environments used and evaluation metrics proposed or adopted from the literature. There are no restrictions on the application domain and this may include classification, regression, natural language processing, reinforcement learning, and planning.

Summary: We seek novel approaches to assist users in creating baseline and edge tests for autonomous agents that address these challenges. Of primary interest are approaches that enable users to define and search for undesirable behaviors, and that provide statistical guarantees on performance.