Towards a Safety Argument for Autonomous Systems that Use Machine Learning

J. McCloskey, C. Allsopp, Chris Carter
j.mccloskey@fnc.co.uk

Abstract — With the rise of artificial intelligence technology, many industries are looking for ways to integrate artificial intelligence into their products, including for safety-related or safety-critical systems. Artificial intelligence is seen to be a useful tool to help eliminate or minimise human error, though comes with a variety of challenges in terms of safety. This paper looks into adopting a hazard-centric approach built of five principles when developing machine learning software algorithms to formulate a safety argument for autonomous systems that use machine learning.

1 Introduction

Artificial Intelligence (AI) including approaches such as Machine Learning (ML) is a trending topic as to how it can be integrated into safety-related and/or safety-critical systems. Hence it is required to consider how safety arguments for systems which exploit ML techniques and autonomous systems are made. A Dstl funded project led by Frazer-Nash Consultancy Ltd (Frazer-Nash) in partnership with Subject Matter Experts (SMEs) from across industry and academia, has developed a generic Safety Argument Structure for an Autonomous System that uses ML. Table 1 lists the industry and academia partners supporting the project.

Table 1: Industrial and Academic Partners

Partner	Area of Specialism on the Project
SeeByte Ltd	Integration of Autonomy and AI on Maritime Platforms
Bristol Robotics Laboratory	Embodied Artificial Intelligence within Autonomous Air Systems
Montvieux Ltd	Artificial Intelligence and Machine Learning Techniques
University of the West of England	Artificial Intelligence and Machine Learning Techniques
Ricardo Ltd	Autonomous Systems
University of Bristol	Functional Verification and Validation for Safety
University of York	Safety and Autonomy

The scope of the safety argument will be similar to that of a traditional product design safety case. The following ML techniques are within the scope of the argument: supervised learning, unsupervised learning, reinforcement learning and deep learning.

2 Hazard Centric and Principled Approach

Adopting a hazard-centric approach allows for; a simpler safety argument immediately focusing on the underpinning hazard and is agnostic of the domain. This approach is supported by previous work undertaken by Dstl [1]. The principles of assurance of ML software (MLSW) are:

- P1: Software safety requirements shall be defined to address the software contribution to system hazards.
- P2: Software detailed design shall embody the intent of the software safety requirements.
- P3: Software safety requirements shall be satisfied.
- P4: Hazardous behaviour of the software shall be identified and mitigated.
- P4+1: The confidence established in addressing the software safety principles shall be commensurate to the contribution of the software to system risk.

2.1 Principle 1

Identifying all hazards for an autonomous system that interfaces with the environment is challenging due to complexity and emergent behaviours, and also lack of specification.

In general, perception/interaction with the environment is required for an autonomous system. Identifying the complete range of scenarios may not be possible and hence it is not possible to identify all associated functional failures. To tackle this, a suggested approach is to use partial specifications.

For highly complex, high-authority software, it is possible that the design and implementation made to meet the individual requirements will interact which results in emergent behaviours from the interaction of all implemented requirements. This behaviour may not be safe.

2.2 Principle 2

The hierarchical decomposition of safety requirements is difficult for MLSW because of the highly inter-dependent nature of the MLSW artefacts and the trial-and-error aspects of the MLSW development process.

The inter-related nature of requirements decomposition and design decisions results in a network relationship. Incremental justifications are required at every design step to take account of inter-dependencies.

2.3 Principle 3

The MLSW learning process is inductive and can introduce different levels of non-determinism and lack of interpretability. This restricts/prohibits the use of analysis techniques to verify claims that safety requirements are satisfied. If verification by testing is the only option, then a vast number of test cases may be required and even then, it is not possible to prove the requirement is met for all potential inputs.

2.4 Principle 4

In order to minimise unanticipated behaviours and implementation errors during the development process, for every design decision, it is necessary to understand the interdependencies of the design commitments to date. A network ontology to track the dependencies may be a useful method in doing so. It is also important to identify the potential limitations/weaknesses associated with the selected datasets, learning algorithms, selected models and evaluation criteria.

2.5 Principle 4+1

For classical software, design standards require increasing rigour of development and assurance for increasing levels of software criticality – this is achieved through the mechanism of Safety Integrity Levels or Design Assurance Levels. No such levels yet exit for MLSW and hence bespoke confidence arguments are required until such times that a consensus on good practice is established.

3 Human Interaction with an Autonomous System

A safety argument for an autonomous system will need to demonstrate that all reasonably foreseeable safety risks associated with human interaction with the system are identified and managed to acceptable levels. One potential method identified to tackle this is the use of System Theoretical Process Analysis (STPA). STPA treats the human as a control function within the overall autonomous system (for instances where the human can monitor and take over from the autonomy) and provides information on how safety constraints due to human error could be violated.

Out-of-The-Loop Loss of Situational Awareness, presents a significant risk for the handover of control and the inability of an operator to regain effective control from the system to the operator. This is aggravated by issues such as passive fatigue [3], reduced operator vigilance [4], [5] and distraction by a secondary activity [6], [7].

Leveson [8] stresses the requirement for a human controller to have a model of an automated system's behaviour to enable effective interaction with the system. A safety argument for an autonomous system should demonstrate that the risks associated with divergence between the operator's mental model of the system and the system design are appropriately managed.

References

[1] E. Lennon, R. Ashmore, *Progress Towards the Assurance of Non-Traditional Software,* DSTL.

[2] P.J. Wilkinson, P.Kelly, *Functional Hazard Analysis for Highly Integrated Aerospace Systems,* Rolls-Royce Commercial Aero Engines Ltd., University of York.

[3] P.A. Hancock, P.A. Desmond, *Stress, Workload and Fatigue,* 2001.

[4] C. Neubauer, G. Matthews, L. Langheim, D. Saxby, *Fatigue and Voluntary Utilization of Automation in Simulated Driving,* 2012.

[5] D. Saxby, G. Matthews, J.S. Warm, E.M. Hitchcock, C. Neubauer, *Active and Passive Fatigue in Simulated Driving: Discriminating Styles of Workload Regulation and their Safety Impacts,* 2013.

[6] M. Regan, J. Lee, K. Young, *Driver Distraction: Theory, Effects and Mitigation,* 2009.

[7] J. De Winter, R. Happee, M.H. Martens, N.A. Stanton, *Effects of adaptive cruise control and highly automated driving on workload and situation awareness: A review of the empirical evidence,* 2014.

[8] N. Leveson, J. Thomas, *An STPA Primer,* 2013.

Author/Speaker Biographies

James McCloskey

James McCloskey leads Frazer-Nash Consultancy's Digital Assurance Group. He has 20 years of experience in the defence industry, much of this time spent focusing on the safety management and certification of Unmanned Aerial Systems.