3.2 – Mitigating potential deviations

Practical guidance – automotive

Authors: Professor Robin E Bloomfield, Dr Gareth Fletcher (Adelard LLP), and Dr Peter Popov (City University)

Defence in depth and diversity are fundamental to achieving high levels of safety within complex systems. Diversity is a key concept and diverse redundancy is needed to counter common cause failures and epistemic uncertainties. It is a sound and widely used design principle in safety-critical applications. Lack of diversity was a key factor in the 2003 North American power blackout as non-diverse backup systems failed in the same way as the primary systems (p.60 [1]).

The key factor which determines how beneficial "design diversity" is, is the failure correlation between "diverse" components. Ideally, when one opts for "design diversity" one hopes that simultaneous channel failures either do not occur at all or, if they do, they are rare. A number of studies (e.g. [2][3]) with non-machine learning (ML) based software demonstrated that the gains from design diversity may be significant but are usually significantly lower than one may hope under the assumption that diverse components would fail (statistically) independently. One explanation for this is that independent designers and developers make similar mistakes because of the inherent difficulty of the problem that the algorithm is solving. The presence of these correlations and the non-independence of failures is a robust result, replicated across experiments sponsored by Nasa, the nuclear industry and others.

Defence in depth in the autonomous vehicle context can take a variety of forms – from hardening a particular functional block (e.g. by deploying design diversity), to building a resilient architecture optimised to detect a failure, confine its impact and recover from failure fast. In addition, diversity can be deployed within design and verification and validation (V&V) teams, between development and assessment organisations, in tool chains to try and avoid problems of complex tool reliability, and in V&V techniques [4].

The principles of how to deploy defence in depth are well-known and discussed widely in safety and security related standards and text books ([5][6][7]). For autonomous systems the challenge is how to deploy defence in depth with ML components. Such ML components may be used as "sensors" in a safety channel (e.g. to detect obstacles on the road) and also to implement an essential part of the functionality (e.g. in journey planners).

Diversity studies have been conducted with ML software too. For instance, a number of studies in the late 1990s examined the effectiveness of design diversity with ML used for character recognition. In these works (e.g. [8]) the authors made two observations:

1. The effectiveness of diversity is affected not only by whether diverse channels fail simultaneously, but also whether the failures are identical or not
2. Diversity between channels can be promoted by carefully planning how the channels are trained, although the practical advice provided by the authors on how this can be done efficiently is very limited

Summary of approach

Diversity is important and should be introduced systematically and explicitly in the system and development lifecycle. For the developer and system architect, there are many options to consider for the ML component including the use of real time ensembles, diverse training sets and different tool chains.

Recommendations are as follows:

1. The use of diversity to improve reliability and safety is a sound principle. In particular it should be used to achieve higher dependability of safety mechanisms. The stakeholders for a mobility service or deployment of autonomous vehicles should undertake a review of defence in depth and define a diversity and defence in depth strategy balancing the advantages of diversity with possible increases in complexity and attack surface.
2. Diversity should be considered in the system architecture to reduce the trust needed in a single ML component. Independence of failures should not be assumed and failure correlation should be considered based, where possible, on experimental data. For example, multiple sensors from different manufacturers should be deployed on independent channels within the autonomous vehicle.
3. There are a number of practicable ways in which diversity could be introduced into the ML lifecycle:
    - Software tools – different ML development platforms
    - ML model architectures and use of ensembles
    - Training data sets
    - Organisational diversity should also be considered with the ML development team independent from the testing and evaluation team
4. The use of diversity to partition the operating regime (e.g. into areas with different types of difficulty) should be considered and the benefits of using ensembles and voting should also be evaluated.
5. Care should be taken when retraining DNNs to ensure that any regression faults do not pose new sources of potential failures for autonomous vehicles (AVs) post retraining. The average performance of the network may have improved; however, this could have been at the expense of introducing regression faults.

Further details on this guidance can be found in [14].

Example of achieving defence in depth and diversity

Here two demonstrator systems are used to study how defence in depth and diversity may be achieved for an autonomous vehicle using ML components. The first is the TIGARS Experimental Vehicle (TEV), which is a modified Yamaha golf cart and has a use case of being a taxi on private property in which obstacle detection and adaptive cruise control are carried out by the installed autonomous systems. The second is Donkey car autonomous driving vehicle [9]. The Donkey car consists of the body of a Radio Control (RC) car, including motor and servo units, controlled by a Raspberry Pi computer and the Donkey car autonomous driving software (an open source python package using TensorFlow [10]). The Donkey car is used to study diversity in neural network ensembles.

Defence in depth and diversity studies on the TEV

The TEV has a typical autonomous vehicle architecture which was to investigate options for deploying defence in depth that are known to have been beneficial in other domains (e.g. sensors, processing information, algorithms). However, the assessment of the effectiveness of defence in depth is application specific and crucially depends on the correlation of failures between the diverse layers of defence.

The unified modelling language component diagram shown in Figure 1 captures a fragment of an architecture with ML components derived from the real architecture of the TEV.

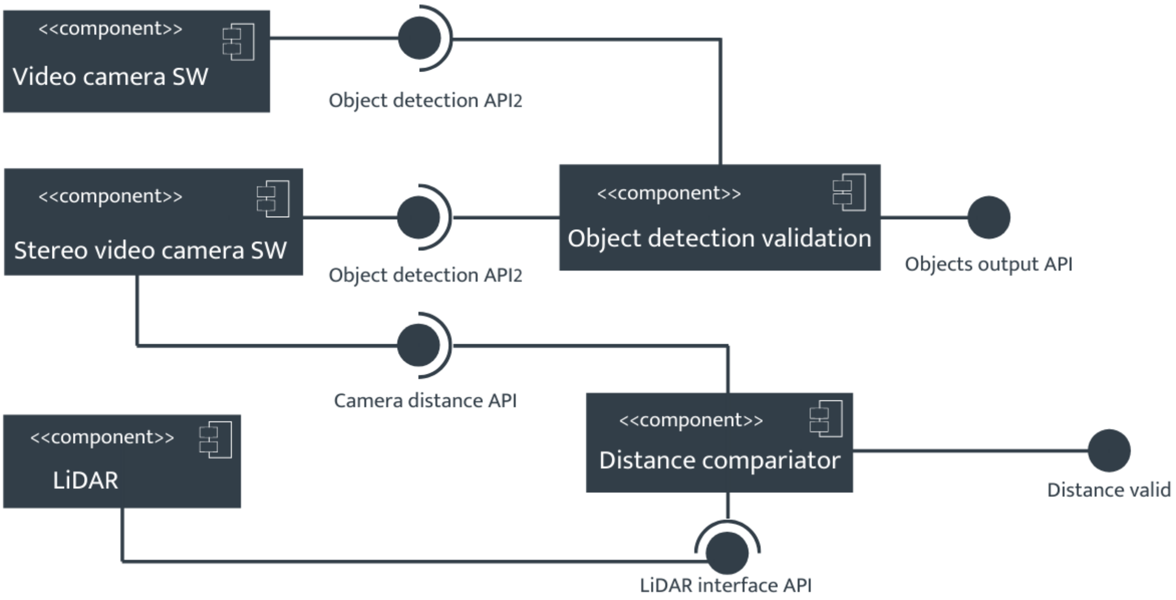

Figure 1 - Fragment of TEV architecture

To improve reliability both functions are implemented using “diverse” components (symmetric diversity); thus eliminating one type of common cause failure. Diversity in object recognition could be achieved by deploying two implementations of a CNN; two “functionally diverse” components are used for the distance measurement function too, one relying on the stereo camera as a sensor and the second on a LIDAR.

However, the two functions are clearly related (each of the channels implements the same functionality or the functionality of the channels is very similar), thus the outcomes from the two functions must be consistent: if objects are detected, the distance measurement should return a plausible value; if no objects are detected, the distance measurement function should return no value. In case of a disagreement between the channels the decision on which of the channels should be trusted is taken by an adjudicator (e.g. majority voting).

This is not possible in the TEV unless an additional channel is added or one of the two channels is trusted more than the other and the second channel is advisory (weakening the benefits of the diversity but still providing a checker/monitor). The TEV trusted the LIDAR distance information more as long as the object detection channels detected a vehicle and the stereo camera’s distance information was used as a checker. Assessing the effectiveness of such an arrangement would need a detailed analysis of the failure correlation between the two channels: the effectiveness would only be undermined if there were circumstances in which the stereo camera would produce correct measurements while the LIDAR-based measurement would produce incorrect output. Less common examples of asymmetric systems (e.g. the LIDAR being used as a checker of an object recognition system based on a stereo camera) are not covered by [11], but the model can be refined to cover the specifics of the TIGARS architecture.

Defence in depth and diversity using neural network ensembles

Neural network ensembles (NNE) adopt "software design diversity" in neural networks. An NNE uses a finite number of individual neural networks for the same learning problem, and the final output is jointly decided by all the outputs of these individuals via an adjudicator.

Diversity is sought by:

1. Diversifying the training data
2. Diversifying the structure, the objective function used in training and/or even the type of the neural networks used in the ensemble

Broadly the ensembles are trained either in parallel ("bagging") or sequentially ("boosting"). A recent survey of the current state-of-the-art in NNE is given in [12].

An asymmetric ensemble of models was tested in experimental trials with the Donkey car. A baseline ML model was used to perform an initial classification assessment – if the autonomous radio controlled car was on a straight or a corner part of the track. This is illustrated in Figure 2. Initial results on the offline test bed showed a significant improvement over a single model approach. Although, we did notice some confusion factor with the classifier model, where it would send some cases to the wrong specialised model (see [13] for more details on the studies).

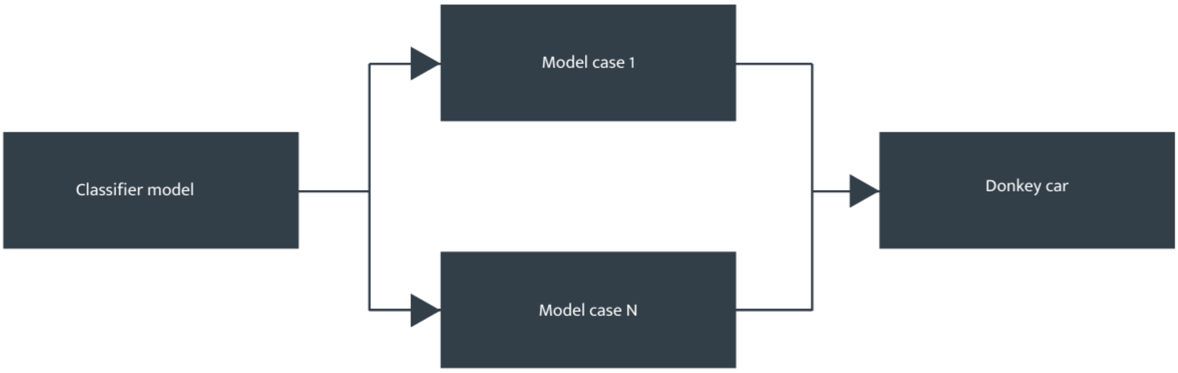

Figure 2 – Asymmetric diversity model architecture

Then, we used one of the two more specialised models (one for the corners, another for the straights) to provide the output steering angle predictions. This type of asymmetric diversity is highly dependent on the classification model being able to differentiate between the two cases well, else there is confusion of which specialised model to use, and reduced accuracy and reliability if the incorrect model was used to make the output predictions.

In the AV context the work on asymmetric systems points to two important issues:

- The insight provided by [11] about the limitations of fault injection experiments may apply in the AV context.

- Some form of trusted checkers are an essential part of the “safety kernel” for them to be able to guarantee a high level of safety.

References

- [1] U.S./Canada Power System Outage Task Force, Final Report on the August 14, 2003 Blackout in the United States and Canada: Causes and Recommendations (PDF). Energy.gov – Office of Electricity Delivery & Energy Reliability (Report), United States Department of Energy, April 2004.
- [2] Knight, J.C. and N.G. Leveson, An Experimental Evaluation of the Assumption of Independence in Multi-Version Programming. IEEE Transactions on Software Engineering, 1986. SE-12(1): p. 96-109.
- [3] Gashi, I., P. Popov, and L. Strigini, Fault Tolerance via Diversity for Off-The-Shelf Products: a Study with SQL Database Servers. IEEE Transactions on Dependable and Secure Computing, 2007. 4(4): p. 280-294.
- [4] NUREG/CR-7007, R. T Wood, R. Belles, et al, `Diversity Strategies for Nuclear Power Plant Instrumentation and Control Systems', US Nuclear Regulatory Commission, 2010. http://pbadupws.nrc.gov/docs/ML1005/ML100541256.pdf.
- [5] Fundamentals of Dependable Computing for Software Engineers (Chapman & Hall/CRC Innovations in Software Engineering and Software Development Series) Paperback – 10 Feb 2012, ISBN-10: 1439862559, ISBN-13: 978-1439862551.
- [6] NIST Special Publication 800-53, Revision 4, Security and Privacy Controls for Federal Information Systems and Organizations http://dx.doi.org/10.6028/NIST.SP.800-53r4.
- [7] ISA/IEC 62443 Cybersecurity Certificate Programs.
- [8] Partridge, D. and W. Krzanowski, Distinct Failure Diversity in Multiversion Software. 1997, University of Exeter, U.K.: Exeter, U.K.Available from: http://citeseerx.ist.psu.edu/viewdoc/versions?doi=10.1.1.30.4700.
- [9] Donkey Car, https://docs.donkeycar.com/, last accessed December 2019.
- [10] TensorFlow, https://www.tensorflow.org/ , last accessed December 2019.
- [11] Popov, P.T. and L. Strigini, Assessing Asymmetric Fault-Tolerant Software. 2010. p. 41-50. Available from: https://doi.org/10.1109/ISSRE.2010.10.
- [12] Li, H., X. Wang, and S. Ding, Research and development of neural network ensembles: a survey. Artificial Intelligence Review, 2018. 49(4): p. 455-479.
- [13] Fletcher G., Imai K., Matsubara Y. et. al., TIGARS Topic paper: Experimentation, D5.6.7 December 2019.
- [14] Bloomfield, R., Fletcher, G., Khlaaf, H., Ryan, P., Kinoshita, S., Kinoshit, Y., Takeyama, M., Matsubara, Y., Popov, P., Imai, K. and Tsutake, Y., 2020. Towards Identifying and closing Gaps in Assurance of autonomous Road vehicleS - A collection of Technical Notes Part 2. arXiv preprint arXiv:2003.00790.