2.3 Implementing requirements using machine learning (ML)

Practical guidance – space

Authors: ACTIONS demonstrator project

This guidance describes the process of following the AMLAS framework to implement safety requirements throughout the development of the ML components of the ACTIONS project. Two ML applications demonstrated in the ACTIONS project are referenced throughout this guidance:

- **Emergency response service** This is the space segment of the ACTIONS project. Autonomous active fire detection is performed on-board an earth observation satellite, and an alert is sent to emergency response services on the ground.
- **Burnt area detection** This is the ground segment of the ACTIONS project. Burnt area detection is performed autonomously on data products created on-board the earth observation satellite, for commercial applications supporting recovery from wildfire events.

Emergency response service

The emergency response service utilises data autonomy, where decisions made on-board the small Earth observation satellite by the ML component have safety implications. Missed active fire detections and/or misdirection of the emergency services on the ground risks damage to property, the natural environment, and potentially to human life.

This section describes the steps of safety requirements definition, and the development stages of the ML component (Data Management, Model Development, Model Verification and Model Deployment) where the requirements are implemented.

System safety requirements

Safety requirements defined for the small satellite system of the ACTIONS project determined that the response service must be truthful, accurate and timely. Four system level requirements were defined:

- **REQ-SAFE-ER-1** - The Emergency Response Service shall determine the location of a visible active fire within 200 m of its true location.
- **REQ-SAFE-ER-2** - The Emergency Response Service shall inform emergency services of a visible active fire within 3 hours of it starting.
- **REQ-SAFE-ER-3** - The Emergency Response Service shall positively identify 95% of all visible active fires acquired by the satellite instrument within the area of interest.
- **REQ-SAFE-ER-4** - The Emergency Response Service shall falsely indicate visible active fires in the area of interest at a rate not exceeding current fire alert service.

ML safety requirements

Of the ACTIONS system safety requirements, three were allocated to the ML component: all except the timeliness requirement. The capability of the system to make timely alerts relies

on a constellation of satellites in orbit, so this requirement remained allocated at system level.

The ML safety requirements concerning performance were defined as follows:

- **MLSR1** – All points of the mask generated by the ML component shall be less than 6 pixels outside the boundary of the area of the real fire.
- **MLSR2** - The ML component shall correctly identify the presence of a fire that satisfies the Schroeder[1] conditions in a frame for 95% of real fires.
- **MLSR3** - The ML component shall not identify the presence of a fire in a frame where there is not a real active fire more than 52[2] times per month.
- **MLSR4** - ML performance requirements shall be satisfied for all data across the range of featured present in the operating scenarios.

Specific details of ML requirements were iterated over as the development data and architecture of the ML model were each defined, also considering the operational scenarios of the system. These requirements were then implemented throughout the development lifecycle of the ML component (Data Management, Model Development, Model Verification and Model Deployment).

Data management

The selection of development data was an integral step in the ML development process. The availability of relevant, labelled satellite datasets is a crucial factor in this domain, and selection will inform choices made in the later stages of Model Learning and Model Verification.

When selecting from available datasets, a list of benefits and limitations is useful. Considerations may include:

- Volume of samples available. Is the set large enough to create a training and validation set, as well as a distinct test set?
- Licensing restrictions. Is the data publicly available to be used for the purposes intended, commercial or otherwise?
- Truth labelling. Do the data samples have one label per image, or are masks available to enable classification at the pixel level? Are additional labels provided to include other relevant information, for example cloud level or land use type? What kind of accuracy has the labelling process achieved?
- The spectral range, and the pixel depth present in the image data. Is the data relevant to and compatible with the data that is expected to be received by the ML component on-board?
- The ground spatial distance of the imagery. What area on the ground does a pixel of data represent?

[1] The sensor tuned conditions for active fire detection set out by Wilfrid Schroeder, Patricia Oliva, Louis Giglio, Brad Quayle, Eckehard Lorenz, and Fabiano Morelli. Active fire detection using Landsat-8/OLI data. Remote Sensing of Environment (Elsevier), 185:210 – 220, 2016. ISSN 0034-4257. doi:10.1016/j.rse.2015.08.032.

[2] NASA FIRMS is being considered as the gold standard for FPs (at 52 instances a month), therefore equivalent or better performance is safe.

- The level of processing applied to the raw data acquired by the satellite sensor. Is this processing compatible with the data that is expected on-board?

Test and verification data

In the ACTIONS project, a large test set in the same format as, but independent to, the development set was used to internally evaluate model performance.

It is common for a model to have been trained on a development set of discrete tiles. This format is not fully relevant to the data captured by sensors on-board a satellite. This discrepancy was addressed in the ACTIONS project by testing performance on much larger arrays of continuous data as early as possible, to understand and address insufficiencies.

Data processing

If there are expected to be different levels of pre-processing applied to the raw sensor data acquired by the satellite, to that which has been applied to the data available during development, then some compensation, and potentially additional training and testing will be necessary. Any other processing steps, such as selecting spectral bands or cropping imagery to multiple tiles of a particular size suitable for the ML model, must also be performed on-board to the pre-processed data that is acquired by the ML component.

Data augmentation

Some examples of augmentations that can be applied to satellite image arrays are flipping, rotating, shifting, zooming, adding noise, and altering pixel intensity. Analysis of the dataset will be useful in selecting which augmentations may benefit training, and visualising augmented samples is useful for understanding their effect.

Data fusion

Data fusion involves the combination of multiple sources of data to add value to or improve the accuracy of the dataset. Sources of data which can be combined may include image data, extracted information, signals such as Automatic Identification System (AIS) or radio frequency sensor measurements, and other onboard telemetry.

Data assurance with AMLAS

The choices made when generating the datasets used during model development and verification were outlined in the AMLAS Data Generation Log along with an explanation of how they met key data requirements of relevance, completeness, accuracy, and balance. A ‘perfect’ dataset is elusive, but this process will expose limitations, and any areas where improvements are possible. The verification dataset for the onboard ML component was generated independently of the model developer in line with the AMLAS framework.

Model learning

When selecting a model for use onboard a small satellite, some domain specific factors will inform the model choice. Due to memory and power constraints, a smaller model architecture is typically necessary. Inference speed is an important factor for on-board processing and a simpler model, with fewer parameters, will also make fault diagnosis more straightforward.

The AMLAS Model Development Log produced in ACTIONS described details of the semantic segmentation model architecture and its parameters, and iterations as the model was evaluated against test data. Other choices documented included image tile dimension and the selection of three spectral bands from the data that were found to be most successful for fire detection.

Model verification

The model was executed against the verification set, which tested the model to the limits of successful operation. Verification data samples with certain features were sourced independently of the model developer. This assured the verification process and exposed the limitations of model performance. For example, the model was found to generate false positive alerts when tested against data from heavily built-up Tokyo, so the system should not be deployed in these conditions.

In the earth observation domain, it may be that compatible satellite data with the required features is available, but not as a labelled dataset. This was a challenge encountered in ACTIONS, and it was necessary to perform a thorough visual analysis of results and to evaluate model performance by eye.

Model deployment

Following model verification, the ML component should ideally be integrated into the system within which it will operate, and then tested. In the space domain this is often impractical, as was the case for deploying the ACTIONS system. As an alternative, integration testing was carried out in a simulated environment with target hardware in the loop across a set of defined operational scenarios.

The simulated test environment was designed to be as similar as possible to the actual deployment environment. This involved running the model on the hardware that will be used in space and simulating a pass of the satellite over the area of interest. A large area of continuous satellite data was sourced for this purpose, containing some instances of known wildfire.

When using simulation in the space domain, it may be worthwhile to introduce some of the sensor anomalies and single event effects that could be encountered onboard the satellite. The effects of input data distortion to the model can be explored, and earlier stages of development perhaps revisited to improve robustness.

It is important that the ML component is tested as robustly and meaningfully as possible before deployment to the target system environment, where opportunity for intervention is limited. Once the ML component is deployed and actual flight data is available, some in-orbit updates may be possible to improve performance or robustness with revised model weights.

Erroneous behaviour

The testing and verification stages of the ACTIONS ML component informed what kind of erroneous behaviour may be expected during actual deployment. Consideration was made to how the model output would be used within the system, and how the expected erroneous behaviour would be handled to minimise negative impact.

For example, the goal of the ACTIONS onboard system is to alert the emergency services to only uncontrolled and unintentional fires, but it was accepted that the model is not able to make this distinction. Therefore, one of the expected erroneous behaviours is to generate alerts for sources of intense heat and light that do not require an emergency response. In response to this erroneous behaviour, the system was designed to provide a compressed image in addition to the text alert, enabling a distinction to be more confidently made by an expert on the ground.

ML component evaluation

Test results were used to evaluate how well the ML component implemented safety requirements of the ACTIONS emergency response service system.

Throughout the various testing stages of the ML component development cycle, availability of data was a key challenge which was addressed with visual validation.

However, labelled test data was available to generate the AMLAS Internal Test Results, which featured clear performance metrics calculated across the dataset. The evaluation examples provided below specifically consider the Internal Test Results and how they meet the defined safety requirements.

Notes on performance metrics

The selection of pixel classification metrics was an important consideration for enabling meaningful evaluation of model performance. Pixel accuracy may be measured and used to calculate false positive and false negative classifications, but this was found to be an unsuitable performance metric due to a significant class imbalance present in the active fire detection datasets used for training and testing. Alternative metrics for the comparison of images were selected. These were Intersection over Union (IoU) and Mean Iou (a mean of the IoU scores calculated for the non-fire pixel class and the active fire pixel class).

Example 1

Requirement:

- The Emergency Response Service shall determine the location of a visible active fire within 200m of its true location.

Evaluation against internal test data:

- Bounded areas will be generated from the model output mask. The location details of these will be communicated to the emergency services within a text alert to quickly communicate the location of discrete, visible, active fires.
- The bounded areas are derived from the model output masks which have a ground spatial resolution of 30 metres per pixel.
- Analysis of the IoU and Mean IoU scores, between the model output masks and the truth masks, shows that the model meets the requirement of being accurate to within 200m. Recorded error is comfortably less than 6 pixels (180m) in any direction, when executing the model against the labelled internal test data.

Figure 1 – Active fire detection performed on a test image.

Example 2

Requirement:

- The Emergency Response Service shall positively identify 95% of all visible active fires acquired by the satellite instrument within the area of interest.

Evaluation against internal test data:

- False negatives were calculated through analysis of IoU scores. A classification was considered a false negative if the score was below a threshold, and the model classified an entire chip as containing no fire where fire was present.
- Across the labelled internal test data, a false negative rate of 0.8% was found. The model positively identifies 99.2% of all visible active fires across the internal test data.

Example 3

Requirement:

- The Emergency Response Service shall falsely indicate visible active fires in the area of interest at a rate not exceeding current fire alert service.

Evaluation against internal test data:

- False positives were calculated through analysis of IoU scores. A classification was considered a false positive if the score was below a threshold, and the model classified an entire sample tile as containing fire where no fire was present.
- Across the labelled internal test data, a false negative rate of 0%* was found.

*It is important to note that this rate is calculated using 'truth' labels which have been algorithmically generated. It is likely there are samples present in the labelled development and test data that have been falsely classified as containing fire. *False positives were later exposed through analysis of model performance during verification and simulation testing.*

Burnt area detection

This commercial application of the ACTIONS project does not have safety-critical requirements, i.e., there is no direct threat to human life from inaccuracies in the output products. As a ground segment, it is also inherently more easily adaptable post-launch, than the on-board system. Combined, these two features reduce the levels of assurance required

in the ground segment, relative to the onboard system. These features are typical of commercial applications from ground segments, so the more lightweight assurance approach applied is likely to be appropriate for most applications.

Although not safety-critical, all commercial applications require levels of accuracy and assurance that enable robust products of value to be delivered to the end-users. Without some level of assurance, the commercial value of downstream products will be reduced and may ultimately damage the financial health of both provider and end-user.

Implementation of the AMLAS framework is expected to lead to a higher quality end product but its rigorous implementation also bears a financial cost to the provider, without necessarily creating a higher financial value product. Consequently, levels of assurance of commercial applications can be considered as a trade-off between rigour and cost and will vary from application to application.

Implementation of AMLAS

The AMLAS framework has been considered throughout the development of the commercial application. A brief overview of some of the key aspects of this implementation are provided in this section. Each stage of the AMLAS framework was considered to varying degrees, with main areas of focus on: Data Management, Model Verification, and Model Deployment.

In terms of ML Safety Assurance and ML Requirements Assurance, discussions with potential end-users were conducted to establish the initial commercial assurance requirements from which the ML requirements could be derived, based on the overall system design.

The AMLAS framework incorporates an iterative approach, with feedback resulting in earlier stages being revisited as development proceeds. This resulted in changes to requirements to improve clarity and testability based on better understanding of the system and user-requirements.

Data management

Data management focused on identifying and preparing a suitable reference data for training and verification, with assurance addressing: Relevance, Completeness, Accuracy, and Balance. Key elements included:

- Use of satellite imagery equivalent to the onboard system design
- Coverage of the relevant geographic area and land cover types
- Coverage of a range of fire events (severity, seasons, wildfire/prescribed, forest types, etc.)
- Use of a well-established and understood dataset
- Balance of a range of burn severity levels and forest types, to provide assurance under different conditions

Observed limitations (e.g. time-lag to post-fire images, bias towards clear-sky) in the resulting datasets, have been acknowledged and potential routes to improvement or mitigation have been proposed but not implemented.

Model verification

Model verification used this dataset to establish that the ML requirements were met. This verification stage did not fully comply with the AMLAS framework, as the verification dataset was not truly independent, in that it was developed alongside the training dataset by the same team. Although not truly independent, best efforts were made to maximise independence (e.g., sampling from distinct fire events to minimise effects of spatial correlation), with decisions on how to split the available reference data made prior to model development commencing.

Further qualitative assurance has also been provided through the full system demonstration of the evolution or a single large fire event over time. A second level of assurance such as this, can serve as useful and potentially more efficient method of testing the impact (if any) of limitations in the training and verification data, without incurring the full expense of augmentation/redesign. Results from secondary qualitative assurance may help guide and streamline further iterations of model development within the AMLAS framework.

Model deployment

A third level of assurance was provided through integration testing, where the focus was testing under more extreme but also realistic active fire conditions. Issues of false positive detection of damage under conditions of thick smoke/cloud were raised through these additional qualitative assurance stages. These highlight the need for a further iteration through the stages of AMLAS framework, potentially revisiting the Data Management and Model Verification stages to augment or redesign the training data, or potentially considering changes to the wider system (e.g., introducing an independent smoke/cloud detection stage).

Integration testing also provides additional assurance of the onboard system, allowing verification of the following: onboard geolocation, the locations of active fire detections, and that outputs from the onboard system fit within the ground segment system design (e.g., the correct satellite channels are present).

Space and ground segment differences

Applying the AMLAS framework to the ground segment is undoubtably useful for assurance purposes, providing a clear understanding of system performance and limitations. However, the degree to which its various stages need to be applied may depend on the nature of the application and the end-user.

Key differences are outlined as follows:

- **Safety considerations** The non-safety critical nature of the commercial application fundamentally reduces the levels of assurance required. Without a threat to life, applying same levels of rigour in assurance become less logical, particularly when considered alongside commercial costs and expectations from end-users. Requirements for downstream commercial applications will typically have more flexibility, with relaxation of requirements still providing a commercially viable product.
- **Time sensitivity** Differences in timing and mechanisms for delivery in the ground segment, also reduce the levels of assurance required (e.g., there is time for additional internal QA before products are delivered that is not available in the rapid and direct delivery of onboard products.

- **Opportunity for intervention** The ground segment presents more opportunities for iterative development and improvement once operational, in terms of ease of implementation and the types of improvements possible (e.g., addition of new, external data layers).

Because of these differences, a more lightweight interpretation of the AMLAS framework should be considered, where assurance can still be provided but to a level that matches the application and end-user. The principles of the AMLAS framework remain valid but focus should be applied to key aspects (e.g., ensuring accurate and representative training data), with less emphasis on features that have potential to be resolved by alternative approaches to system design (e.g., introducing cloud detection). Such an approach is likely to yield a product with high levels of assurance and accuracy under a known set of conditions. Any outstanding inaccuracies or limitations can be acknowledged and potentially mitigated for through changes to system design or requirements that still satisfy end-user needs.

Summary of approach

1. Define system safety requirements.
2. Define ML safety requirements.
3. Complete the ML component lifecycle process (Data Management, Model Learning, Model Verification and Model Deployment).
4. Evaluate how the ML component meets requirements at each stage, iterating back through steps of the cycle where necessary.
5. Generate and maintain AMLAS artefacts to assure and validate every stage of development and justify conclusions made during evaluation against requirements.
6. The degree to which the AMLAS framework is followed may be defined as appropriate to the safety criticality of the ML application, as outlined in the discussion of differences between the space and ground segments of the ACTIONS project.