A Safety Framework for Critical Systems Utilising Deep Neural Networks

Xingyu Zhao[1], Alec Banks[2], James Sharp[2], Valentin Robu[1], David Flynn[1], Michael Fisher[3], and Xiaowei Huang[3(✉)]

[1] Heriot-Watt University, Edinburgh EH14 4AS, UK
{xingyu.zhao,v.robu,d.flynn}@hw.ac.uk
[2] Defence Science and Technology Laboratory, Salisbury SP4 0JQ, UK
{abanks,jsharp1}@dstl.gov.uk
[3] University of Liverpool, Liverpool L69 3BX, UK
{mfisher,xiaowei.huang}@liverpool.ac.uk

Abstract. Increasingly sophisticated mathematical modelling processes from Machine Learning are being used to analyse complex data. However, the performance and explainability of these models within practical critical systems requires a rigorous and continuous verification of their safe utilisation. Working towards addressing this challenge, this paper presents a principled novel safety argument framework for critical systems that utilise deep neural networks. The approach allows various forms of predictions, e.g., future reliability of passing some demands, or confidence on a required reliability level. It is supported by a Bayesian analysis using operational data and the recent verification and validation techniques for deep learning. The prediction is conservative – it starts with partial prior knowledge obtained from lifecycle activities and then determines the worst-case prediction. Open challenges are also identified.

Keywords: Safety cases · Quantitative claims · Reliability claims · Deep learning verification · Assurance arguments · Safe AI · Bayesian inference

1 Introduction

Deep learning (DL) has been applied broadly in industrial sectors including automotive, healthcare, aviation and finance. To fully exploit the potential offered by DL, there is an urgent need to develop approaches to their certification in safety critical applications. For traditional systems, safety analysis has aided engineers in *arguing* that the system is sufficiently safe. However, the deployment of DL in critical systems requires a thorough revisit of that analysis to reflect the novel characteristics of Machine Learning (ML) in general [2,10,27].

Compared with traditional systems, the behaviour of learning-enabled systems is much harder to predict, due to, *inter alia*, their "black-box" nature and the lack of traceable functional requirements of their DL components. The

A. Casimiro et al. (Eds.): SAFECOMP 2020, LNCS 12234, pp. 244–259, 2020.
https://doi.org/10.1007/978-3-030-54549-9_16

"black-box" nature hinders the human operators in understanding the DL and makes it hard to predict the system behaviour when faced with new data. The lack of explicit requirement traceability through to code implementation is only partially offset by learning from a dataset, which at best provides an incomplete description of the problem. These characteristics of DL increase apparent non-determinism [25], which on the one hand emphasises the role of *probabilistic measures* in capturing uncertainty, but on the other hand makes it notoriously hard to estimate the probabilities (and also the consequences) of critical failures.

Recent progress has been made to support the Verification and Validation (V&V) of DL, e.g., [23,47]. Although these methods may provide evidence to support low-level claims, e.g., the local robustness of a deep neural network (DNN) on a given input, they are insufficient by themselves to justify overall system safety claims. Here, we present a safety case framework for DL models which may in turn support higher-level system safety arguments. We focus on DNNs that have been widely deployed as, e.g., perception/control units of autonomous systems. Due to the page limit, we also confine the framework to DNNs that are fixed in the operation; this can be extended for online learning DNNs in future.

We consider safety-related properties including reliability, robustness, interpretability, fairness [6], and privacy [1]. In particular, we emphasise the assessment of DNN *generalisation error* (in terms of inaccuracy), as a major reliability measure, throughout our safety case. We build arguments in two steps. The first is to provide initial confidence that the DNN's generalisation error is bounded, through the assurance activities conducted at each stage of its lifecycle, e.g., formal verification on the DNN robustness. The second step is to adopt *proven-in-use/field-testing* arguments to boost the confidence and check whether the DNN is indeed sufficiently safe for the risk associated with its use in the system.

The second step above is done in a statistically principled way via Conservative Bayesian Inference (CBI) [8,46,49]. CBI requires only *limited and partial* prior knowledge of reliability, which differs from normal Bayesian analysis that usually assumes a *complete* prior distribution on the failure rate. This has a unique advantage: partial prior knowledge is more convincing (i.e. constitutes a more realistic claim) and easier to obtain, while complete prior distributions usually require extra assumptions and introduces optimistic bias. CBI allows many forms of prediction, e.g., posterior expected failure rate [8], future reliability of passing some demands [46] or a posterior confidence on a required reliability bound [49]. Importantly, CBI guarantees conservative outcomes: it finds the worst-case prior distribution yielding, say, a maximised posterior expected failure rate, and satisfying the partial knowledge. We are aware that there are other extant dangerous pitfalls in safety arguments [25,27], thus we also identify *open challenges* in our proposed framework and map them onto on-going research.

The key contributions of this work are:

a) A very first safety case framework for DNNs that mainly concerns *quantitative* claims based on structured heterogeneous safety arguments.

b) An initial idea of mapping DNN lifecycle activities to the reduction of decomposed DNN generalisation error that used as a primary reliability measure.

c) Identification of open challenges in building safety arguments for quantitative claims, and mapping them onto on-going research of potential solutions.

Next, we present preliminaries. Sect. 3 provides top-level argument, and Sect. 4 presents how CBI approach assures reliability. Other safety related properties are discussed in Sect. 5. We discuss related work in Sect. 6 and conclude in Sect. 7.

2 Preliminaries

2.1 Safety Cases

A safety case is a comprehensive, defensible, and valid justification of the safety of a system for a given application in a defined operating environment, thus it is a means to provide the grounds for confidence and to assist decision making in certification [12]. Early research in safety cases mainly focus on their formulation in terms of claims, arguments and evidence elements. The two most popular notations are CAE [12] and GSN [26]. In this paper, we choose the latter to present our safety case framework.

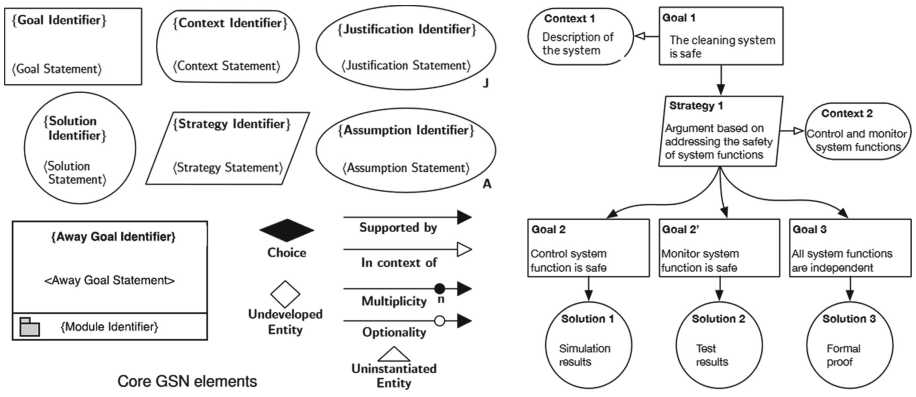

Fig. 1. The GSN core elements and an example of using GSN

Figure 1 shows the core GSN elements and a quick GSN example. Essentially, the GSN safety case starts with a top *goal* (claim) which then is decomposed through an argument *strategy* into sub-goals (sub-claims), and sub-goals can be further decomposed until being supported by *solutions* (evidence). A claim may be subject to some *context* or *assumption*. An *away goal* repeats a claim presented in another argument module. A description on all GSN elements used in this paper can be found in [26].

2.2 Deep Neural Networks and Lifecycle Models

Let (X, Y) be the training data, where X is a vector of inputs and Y is a vector of outputs such that $|X| = |Y|$. Let $\mathtt{X}$ be the input domain and $\mathtt{Y}$ be the set of labels. Hence, $X \subset \mathtt{X}$. We may use x and y to range over $\mathtt{X}$ and $\mathtt{Y}$, respectively. Let $\mathcal{N}$ be a DNN of a given architecture. A network $\mathcal{N} : \mathtt{X} \rightarrow \mathcal{D}(\mathtt{Y})$ can be seen as a function mapping from $\mathtt{X}$ to probabilistic distributions over $\mathtt{Y}$. That is, $\mathcal{N}(x)$ is a probabilistic distribution, which assigns for each possible label $y \in \mathtt{Y}$ a probability value $(\mathcal{N}(x))_y$. We let $f_{\mathcal{N}} : \mathtt{X} \rightarrow \mathtt{Y}$ be a function such that for any $x \in \mathtt{X}$, $f_{\mathcal{N}}(x) = \arg\max_{y \in \mathtt{Y}}\{(\mathcal{N}(x))_y\}$, i.e. $f_{\mathcal{N}}(x)$ returns the classification label. The network is trained with a parameterised learning algorithm, in which there are (implicit) parameters representing e.g., the number of epochs, the loss function, the learning rate, the optimisation algorithm, etc.

A comprehensive ML *Lifecycle Model* can be found in [4], which identifies assurance desiderata for each stage, and reviews existing methods that contribute to achieving these desiderata. In this paper, we refer to a simpler lifecycle model that includes several phases: initiation, data collection, model construction, model training, analysis of the trained model, and run-time enforcement.

2.3 Generalisation Error

Generalisability requires that a neural network works well on all possible inputs in $\mathtt{X}$, although it is only trained on the training dataset (X, Y).

Definition 1. *Assume that there is a ground truth function* $f : \mathtt{X} \rightarrow \mathtt{Y}$ *and a probability function* $O_p : \mathtt{X} \rightarrow [0, 1]$ *representing the operational profile. A network* $\mathcal{N}$ *trained on* (X, Y) *has a generalisation error:*

$$G_{\mathcal{N}}^{0-1} = \sum_{x \in \mathtt{X}} \mathbf{1}_{\{f_{\mathcal{N}}(x) \neq f(x)\}} \times O_p(x) \tag{1}$$

where $\mathbf{1}_{\mathtt{S}}$ *is an indicator function – it is equal to 1 when* $\mathtt{S}$ *is true and 0 otherwise.*

We use the notation $O_p(x)$ to represent the probability of an input x being selected, which aligns with the *operational profile* notion [35] in software engineering. Moreover, we use 0-1 loss function (i.e., assigns value 0 to loss for a correct classification and 1 for an incorrect classification) so that, for a given O_p, $G_{\mathcal{N}}^{0-1}$ is equivalent to the reliability measure *pfd* (the expected probability of the system failing on a random demand) defined in the safety standard IEC-61508. A "frequentist" interpretation of *pfd* is that it is the limiting relative frequency of demands for which the DNN fails in an infinite sequence of independently selected demands [48]. The primary safety measure we study here is *pfd*, which is equivalent to the generalisation error $G_{\mathcal{N}}^{0-1}$ in (1). Thus, we may use the two terms interchangeably in our safety case, depending on the context.

3 The Top-Level Argument

Figure 2 gives a top-level safety argument for the top claim **G1** – the DNN is sufficiently safe. We first argue **S1**: that all safety related properties are satisfied. The list of all properties of interest for the given application can be obtained by utilising the Property Based Requirements (PBR) [34] approach. The PBR method is a way to specify requirements as a set of properties of system objects in either structured language or formal notations. PBR is recommended in [2] as a method for the safety argument of autonomous systems. Without the loss of generality, in this paper, we focus on the major quantitative property: reliability (**G2**). Due to space constraints, other properties: interpretability, robustness, etc. are discussed in Sect. 5 but remain an undeveloped goal (**G3**) here.

More properties that have a safety impact can be incorporated in the framework as new requirements emerge from, e.g., ethical aspects of the DNN.

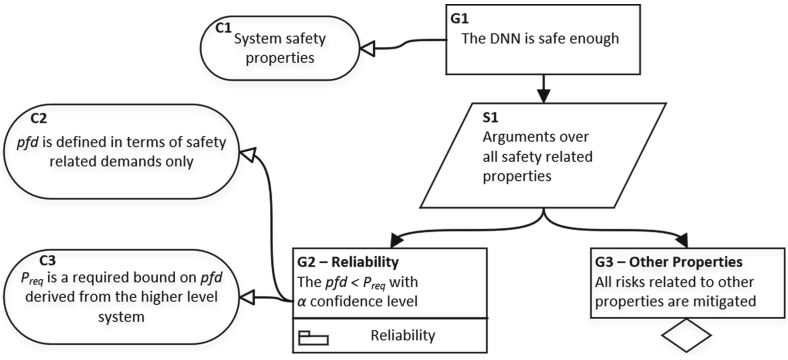

Fig. 2. The top-level safety argument

Despite the controversy over the use of probabilistic measures (e.g., *pfd*) for the safety of conventional software systems [29], we believe probabilistic measures are useful when dealing with ML systems since arguments involving their inherent uncertainty are naturally stated in probabilistic terms.

Setting a reliability goal (**G2**) for a DNN varies from one application to another. Questions we need to ask include: (i) What is the appropriate reliability measure? (ii) What is the quantitative requirement stated in that reliability measure? (iii) How can confidence be gained in that reliability claim?

Reliability of safety critical systems, as a probabilistic claim, will be about the probabilities/rates of occurrence of failures that have safety impacts, e.g., a dangerous misclassification in a DNN. Generally, systems can be classified as either: continuous-time systems that are being continuously operated in the active control of some process; or on-demand systems, which are only called upon to act on receipt of discrete demands. Normally we study the failure rate (number of failures in one time unit) of the former (e.g., flight control software)

and the probability of failure per demand (*pfd*) of the latter (e.g., the emergency shutdown system of a nuclear plant). In this paper, we focus on *pfd* which aligns with DNN classifiers for perception, where demands are e.g., images from cameras.

Given the fact that most safety critical systems adopt a *defence in depth design* with safety backup channels [28], the required reliability (p_{req} in **G2**) should be derived from the higher level system, e.g., a 1-out-of-2 (1oo2) system in which the other channel could be either hardware-only, conventional software-based, or another ML software. The required reliability of the whole 1oo2 system may be obtained from regulators or compared to human level performance (e.g., a target of 100 times safer than average human drivers, as studied in [49]). We remark that deriving a required reliability for individual channels to meet the whole 1oo2 reliability requirement is still an open challenge due to the dependencies among channels [30] (e.g., a "hard" demand is likely to cause both channels to fail). That said, there is ongoing research towards rigorous methods to decompose the reliability of 1oo2 systems into those of individual channels which may apply and provide insights for future work, e.g., [7] for 1oo2 systems with one hardware-only and one software-based channels, [28,48] for a 1oo2 system with one possibly-perfect channel, and [15] utilising fault-injection technique. In particular, for systems with duplicated DL channels, we note that there are similar techniques, e.g., (i) ensemble method [39], where a set of DL models run in parallel and the result is obtained by applying a voting protocol; (ii) simplex architecture [45], where there is a main classifier and a safer classifier, with the latter being simple enough so that its safety can be formally verified. Whenever confidence of the main classifier is low, the decision making is taken over by the safer classifier; the safer classifier can be implemented with e.g., a smaller DNN.

As discussed in [8], the reliability measure, *pfd*, concerns system behaviour subject to *aleatory* uncertainty ("uncertainty in the world"). On the other hand, *epistemic* uncertainty concerns the uncertainty in the "beliefs about the world". In our context, it is about the human assessor's *epistemic* uncertainty of the reliability claim obtained through assurance activities. For example, we may not be *certain* whether a claim – the *pfd* is smaller than 10^{-4} – is true due to our imperfect understanding about the assurance activities. All assurance activities in the lifecycle with supportive evidence would increase our *confidence* in the reliability claim, whose formal quantitative treatment has been proposed in [11,32]. Similarly to the idea proposed in [46], we argue that all "process" evidence generated from the DNN lifecycle activities provides initial confidence of a desired *pfd* bound. Then the confidence in a *pfd* claim is acquired incrementally through operational data of the trained DNN via CBI – which we describe next.

4 Reliability with Lifecycle Assurance

4.1 CBI Utilising Operational Data

In Bayesian reliability analysis, assessors normally have a prior distribution of *pfd* (capturing the *epistemic* uncertainties), and update their beliefs (the prior

distribution) by operational data. Given the safety-critical nature, the systems under study will typically see *failure-free* operation or very *rare failures*. Bayesian inference based on such non or rare failures may introduce dangerously optimistic bias if using a *Uniform* or *Jeffreys prior* which describes not only one's prior knowledge, but adds extra, unjustified assumptions [49]. Alternatively, CBI is a technique, first described in [8], which applied Bayesian analysis with only *partial* prior knowledge; by partial prior knowledge, we mean the following typical forms:

- $\mathbb{E}[pfd] \leq m$: the prior mean *pfd* cannot be worse than a stated value;
- $Pr(pfd \leq \epsilon) = \theta$: a prior confidence bound on *pfd*;
- $\mathbb{E}[(1 - pfd)^n] \geq \gamma$: prior confidence in the reliability of passing n tests.

These can be used by CBI either solely or in combination (e.g., several confidence bounds). The partial prior knowledge is far from a complete prior distribution, thus it is easier to obtain from DNN lifecycle activities (**C4**). For instance, there are studies on the generalisation error bounds, based on how the DNN was constructed, trained and verified [5,21]. We present examples on how to obtain such partial prior knowledge (**G6**) using evidence, e.g. from formal verification on DNN robustness, in the next section. CBI has also been investigated for various objective functions with a "posterior" flavour:

- $\mathbb{E}[pfd \mid \text{pass } n \text{ tests}]$: the posterior expected *pfd* [8];
- $Pr(pfd \leq p_{req} \mid k \text{ failures in } n \text{ tests})$: the posterior confidence bound on *pfd* [48,49]; the p_{req} is normally a small *pfd*, stipulated at higher level;
- $\mathbb{E}[(1 - pfd)^t \mid \text{pass } n \text{ tests}]$: the future reliability of passing t demands in [46].

Example 1. In Fig. 3, we plot a set of numerical examples based on the CBI model in [46]. It describes the following scenario: the assessor has θ confidence that the software *pfd* cannot be worse than ϵ (e.g., 10^{-4} according to SIL-4), then after n failure-free runs (the x-axis), the future reliability of passing t demands is shown on the y-axis. We may observe that stronger prior beliefs (smaller ϵ with larger θ) and/or larger n/t ratio allows higher future reliability claims.

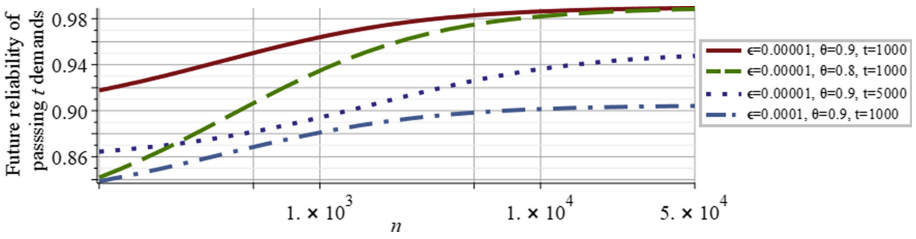

Fig. 3. Numerical examples based on the CBI model in [46]

Depending on the objective function of interest (**G2** is an example of a posterior confidence bound) and the set of partial prior knowledge obtained (**G6**), we choose a corresponding CBI model[1] for **S2**. Note, we also need to explicitly assess the impact of CBI model assumptions (**G5**). Published CBI theorems abstract the stochastic failure process as a sequence of independent and identically distributed (i.i.d.) Bernoulli trials given the unknown *pfd*, and assume the operational profile is constant [8,46,49]. Although we identify how to justify/relax those assumptions as open challenges, we note some promising ongoing research:

a) The i.i.d. assumption means a constant *pfd*, which may not hold for a system update or deployment in a new environment. In [31], CBI is extended to a *multivariate* prior distribution case coping with scenarios of a *changing pfd*, which may provide the basis of arguments for online learning DNNs in future.
b) The effect of assuming independence between successive demands has been studied, e.g., [20]. It is believed that the effect is negligible given non or rare failures; note this requires further (preferably conservative) studies.
c) The changes to the operational profile is a major challenge for all proven-in-use/field-testing safety arguments [27]. Recent research [9] provides a novel conservative treatment for the problem, which can be retrofitted for CBI.

The safety argument via CBI is presented in Fig. 4. In summary, we collect a set of partial prior knowledge from various lifecycle activities, then boost our posterior confidence in a reliability claim of interest through operational data, in a conservative Bayesian manner. We believe this aligns with the practice of applying management systems in reality – a system is built with claims of sufficient confidence that it may be deployed; these claims are then independently assessed to confirm said confidence is justified. Once deployed, the system safety performance is then monitored for continuing validation of the claims. Where there is insufficient evidence systems can be fielded with the risk held by the operator, but that risk must be minimised through operational restrictions. As confidence then grows these restrictions may be relaxed.

4.2 Partial Prior Knowledge on the Generalisation Error

Our novel CBI safety argument for the reliability of DNNs is essentially inspired by the idea proposed in [46] for conventional software, in which the authors seek prior confidence in the (quasi-)perfection of the software from "process" evidence like formal proofs, and effective development activities. In our case, to make clear the connection between lifecycle activities and their contributions to the generalisation error, we decompose the generalisation error into three:

$$G_{\mathcal{N}}^{0-1} = \underbrace{G_{\mathcal{N}}^{0-1} - \inf_{\mathcal{N}\in\mathtt{N}} G_{\mathcal{N}}^{0-1}}_{\text{Estimation error of}}\,\mathcal{N} + \underbrace{\inf_{\mathcal{N}\in\mathtt{N}} G_{\mathcal{N}}^{0-1} - G_{f,(X,Y)}^{0-1,*}}_{\text{Approximation error of}}\,\mathtt{N} + \underbrace{G_{f,(X,Y)}^{0-1,*}}_{\text{Bayes error}} \tag{2}$$

[1] There are CBI combinations of objective functions and partial prior knowledge haven't been investigated, which remains as open challenges.

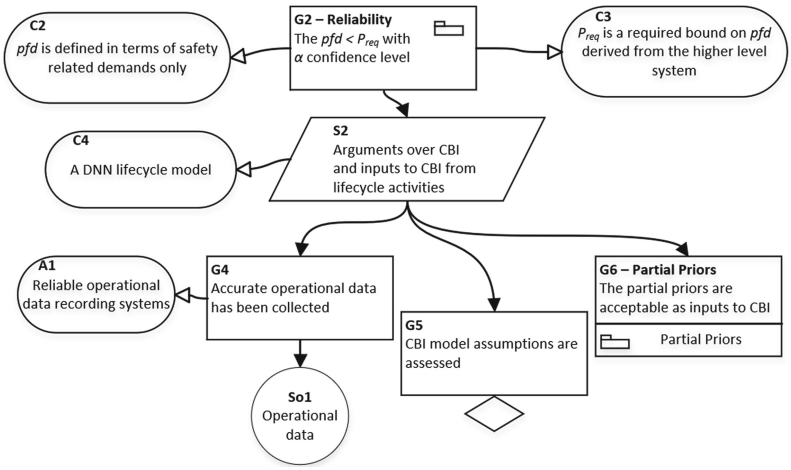

Fig. 4. The CBI safety argument

a) The *Bayes error* is the lowest and irreducible error rate over all possible classifiers for the given classification problem [19]. It is non-zero if the true labels are not deterministic (e.g., an image being labelled as y_1 by one person but as y_2 by others), thus intuitively it captures the uncertainties in the dataset (X, Y) and true distribution f when aiming to solve a real-world problem with DL. We estimate this error (implicitly) at the **initiation** and **data collection** stages in activities like: necessity consideration and dataset preparation etc.

b) The *Approximation error of* $\mathtt{N}$ measures how far the best classifier in $\mathtt{N}$ is from the overall optimal classifier, after isolating the Bayes error. The set $\mathtt{N}$ is determined by the architecture of DNNs (e.g., numbers of layers), thus lifecycle activities at the **model construction** stage are used to minimise this error.

c) The *Estimation error of* $\mathcal{N}$ measures how far the learned classifier $\mathcal{N}$ is from the best classifier in $\mathtt{N}$. Lifecycle activities at the **model training** stage essentially aim to reduce this error, i.e., performing optimisations of the set $\mathtt{N}$.

Both the Approximation and Estimation errors are reducible. We believe, the *ultimate goal* of all lifecycle activities is to reduce the two errors to 0, especially for safety-critical DNNs. This is analogous to the "possible perfection" notion of traditional software as pointed to by Rushby and Littlewood [28,42]. That is, assurance activities, e.g., performed in support of DO-178C, can be best understood as developing evidence of possible perfection – a confidence in $pfd = 0$. Similarly, for safety critical DNNs, we believe ML lifecycle activities should be considered as aiming to train a "possible perfect" DNN in terms of the two *reducible* errors. Thus, we may have some confidence that the two errors are both 0 (equivalently, a prior confidence in the *irreducible* Bayes error since the other two are 0), which indeed is supported by on-going research into finding globally optimised DNNs [17]. Meanwhile, on the **trained model**, V&V also

provides prior knowledge as shown in Example 2 below, and **online monitoring** continuously validates the assumptions for the prior knowledge being obtained.

Example 2. We present an illustrative example on how to obtain a prior confidence bound on the generalisation error from formal verification of DNN robustness [23,40]. *Robustness* requires that the decision making of a neural network cannot be drastically changed due to a small perturbation on the input. Formally, given a real number $d > 0$ and a distance measure $||\cdot||_p$, for any input $x \in \mathtt{X}$, we have that, $f_{\mathcal{N}}(x) = f_{\mathcal{N}}(x')$ whenever $||x' - x||_p \leq d$.

Figure 5 shows an example of the robustness verification in a one-dimensional space. Each blue triangle represents an input x, and the green region around each input x represents all the neighbours, x' of x, which satisfy $||x' - x||_p \leq d$ and $f_{\mathcal{N}}(x) = f_{\mathcal{N}}(x')$. Now if we assume $Op(x)$ is uniformly distributed (an assumption for illustrative purposes which can be relaxed for other given $Op(x)$ distributions), the generalisation error has a lower bound – the chance that the next randomly selected input does not fall into the green regions. That is, if ϵ denotes the ratio of the length not being covered by the green regions to the total length of the black line, then $G_{\mathcal{N}}^{0-1} \leq \epsilon$. This said, we cannot be certain about the bound $G_{\mathcal{N}}^{0-1} \leq \epsilon$ due to assumptions like: (i) The formal verification tool itself is perfect, which may not hold; (ii) Any neighbour x' of x has the same ground truth label of x. For a more comprehensive list, cf. [14]. Assessors need to capture the doubt (say $1 - \theta$) in those assumptions, which leads to:

$$Pr(G_{\mathcal{N}}^{0-1} \leq \epsilon) = \theta. \tag{3}$$

We now have presented an instance of the safety argument template in Fig. 6. The solution **So2** is the formal verification showing $G_{\mathcal{N}}^{0-1} \leq \epsilon$, and **G8** quantifies the confidence θ in that result. It is indeed an open challenge to rigorously develop **G8** further, which may involve scientific ways of eliciting expert judgement [36] and systematically collecting process data (e.g., statistics on the reliability of verification tools). However, we believe this challenge – evaluating confidence in claims, either quantitatively or qualitatively (e.g., ranking with low, medium, high), explicitly or implicitly – is a fundamental problem for all safety case based decision-makings [11,16], rather than a specific problem of our framework.

The sub-goal **G9** represents the mechanism of online monitoring on the validity of offline actives, e.g., validating the environmental assumptions used by offline formal verifications against the real environment at runtime [18].

Fig. 5. Formal verification on DNN robustness in an one-dimensional space

5 Other Safety Related Properties

So far we have seen a reliability-centric safety case for DNNs. Recall that, in this paper, reliability is the probability of misclassification (i.e. the generalisation error in (1)) that has safety impacts. However, there are other DNN safety related properties concerning risks not directly caused by a misclassification, like interpretability, fairness, and privacy; discussed as follows.

Interpretability is about an explanation procedure to present an interpretation of a single decision within the overall model in a way that is easy for humans to understand. There are different explanation techniques aiming to work with different objects, see [22] for a survey. Here we take the instance explanation as an example – the goal is to find another representation $\texttt{expl}(f_{\mathcal{N}}, x)$ of an input x, with the expectation that $\texttt{expl}(f_{\mathcal{N}}, x)$ carries simple, yet essential, information that can help the user understand the decision $f_{\mathcal{N}}(x)$. We use $f(x) \Leftrightarrow \texttt{expl}(f_{\mathcal{N}}, x)$ to denote that the explanation is consistent with a human's explanation in $f(x)$. Thus, similarly to (1), we can define a probabilistic measure for the instance-wise interpretability:

$$I_{\mathcal{N}} = \sum_{x \in \mathbf{X}} (f(x) \iff \texttt{expl}(f_{\mathcal{N}}, x)) \times O_p(x) \tag{4}$$

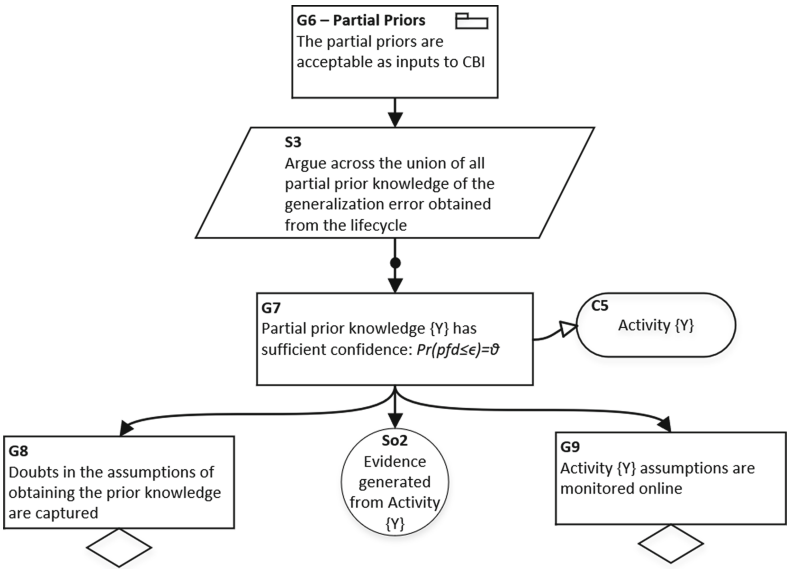

Fig. 6. A template of safety arguments for obtaining partial prior knowledge

Then similarly as the argument for reliability, we can do statistical inference with the probabilistic measure $I_{\mathcal{N}}$. For instance, as in Example 2, we (i) firstly

define the robustness of explanations in norm balls, measuring the percentage of space that has been verified as a bound on $I_{\mathcal{N}}$, (ii) then estimate the confidence of the robust explanation assumption and obtain a prior confidence in interpretability, (iii) finally Bayesian inference is applied with runtime data.

Fairness requires that, when using DL to predict an output, the prediction remains unbiased with respect to some protected features. For example, a financial service company may use DL to decide whether or not to provide loans to an applicant, and it is expected that such decision should not rely on sensitive features such as race and gender. *Privacy* is used to prevent an observer from determining whether or not a sample was in the model's training dataset, when it is not allowed to observe the dataset directly. Training methods such as [1] have been applied to pursue differential privacy.

The lack of fairness or privacy may cause not only a significant monetary loss but also ethical issues. Ethics has been regarded as a long-term challenge for AI safety. For these properties, we believe the general methodology suggested here still works – we first introduce bespoke probabilistic measures according to their definitions, obtain prior knowledge on the measures from lifecycle activities, then conduct statistical inference during the continuous monitoring of the operation.

6 Related Work

Alves *et al.* [2] present a comprehensive discussion on the aspects that need to be considered when developing a safety case for increasingly autonomous systems that contain ML components. In [10], a safety case framework with specific challenges for ML is proposed. [44] reviews available certification techniques from the aspects of lifecycle phases, maturity and applicability to different types of ML systems. In [27], safety arguments that are being widely used for conventional systems – including conformance to standards, proven in use, field testing, simulation and formal proofs – are recapped for autonomous systems with discussions on the potential pitfalls. Similar to our CBI arguments that exploit operational data, [24,33] propose utilising continuously updated arguments to monitor the weak points and the effectiveness of their countermeasures. The work [3] identifies applicable quantitative measures of assurance for learning-enabled components.

Regarding the safety of automated driving, [41,43] discuss the extension and adaptation of ISO-26262, and [13] considers functional insufficiencies in the perception functions based on DL. Additionally, [37,38] explores safety case patterns that are reusable for DL in the context of medical applications.

7 Discussions, Conclusions and Future Work

In this paper, we present a novel safety argument framework for DNNs using probabilistic risk assessment, mainly considering quantitative reliability claims, generalising this idea to other safety related properties. We emphasise the use of probabilistic measures to describe the inherent uncertainties of DNNs in safety

arguments, and conduct Bayesian inference to strengthen the top-level claims from safe operational data through to continuous monitoring after deployment.

Bayesian inference requires prior knowledge, so we propose a novel view by (i) decomposing the DNN generalisation error into a composition of distinct errors and (ii) try to map each lifecycle activity to the reduction of these errors. Although we have shown an example of obtaining priors from robustness verification of DNNs, it is non-trivial (and identified as an open challenge) to establish a quantitative link between other lifecycle activities to the generalisation error. Expert judgement and past experience (e.g., a repository on DNNs developed by similar lifecycle activities) seem to be inevitable in overcoming such difficulties.

Thanks to the CBI approach – Bayesian inference with limited and partial prior knowledge – even with sparse prior information (e.g., a single confidence bound on the generalisation error obtained from robustness verification), we can still apply probabilistic inference given the operational data. Whenever there are sound arguments to obtain additional partial prior knowledge, CBI can incorporate them as well, and reduce the conservatism in the reasoning [8]. On the other hand, CBI as a type of proven-in-use/field-testing argument has some of the fundamental limitations as highlighted in [25,27], for which we have identified on-going research towards potential solutions.

We concur with [27] that, despite the dangerous pitfalls for various existing safety arguments, credible safety cases require a heterogeneous approach. Our new quantitative safety case framework provides a novel supplementary approach to existing frameworks rather than replace them. We plan to conduct concrete case studies and continue to work on the open challenges identified.

Acknowledgements and Disclaimer. This work is supported by the UK EPSRC (through the Offshore Robotics for Certification of Assets [EP/R026173/1] and its PRF project COVE, and End-to-End Conceptual Guarding of Neural Architectures [EP/T026995/1]) and the UK Dstl (through projects on Test Coverage Metrics for Artificial Intelligence). Xingyu Zhao and Alec Banks' contribution to the work is partially supported through Fellowships at the Assuring Autonomy International Programme.

This document is an overview of UK MOD (part) sponsored research and is released for informational purposes only. The contents of this document should not be interpreted as representing the views of the UK MOD, nor should it be assumed that they reflect any current or future UK MOD policy. The information contained in this document cannot supersede any statutory or contractual requirements or liabilities and is offered without prejudice or commitment.

References

1. Abadi, M., et al.: Deep learning with differential privacy. In: ACM SIGSAC CCS'16 (2016)

2. Alves, E., Bhatt, D., Hall, B., Driscoll, K., Murugesan, A., Rushby, J.: Considerations in assuring safety of increasingly autonomous systems. Technical report NASA/CR-2018-220080, NASA, July 2018
3. Asaadi, E., Denney, E., Pai, G.: Towards quantification of assurance for learning-enabled components. In: EDCC 2019, pp. 55–62. IEEE, Naples, Italy (2019)
4. Ashmore, R., Calinescu, R., Paterson, C.: Assuring the machine learning lifecycle: Desiderata, methods, and challenges. arXiv preprint arXiv:1905.04223 (2019)
5. Bagnall, A., Stewart, G.: Certifying the true error: Machine learning in Coq with verified generalization guarantees. In: AAAI 2019, vol. 33, pp. 2662–2669 (2019)
6. Barocas, S., Hardt, M., Narayanan, A.: Fairness and Machine Learning. fairmlbook.org (2019). http://www.fairmlbook.org
7. Bishop, P., Bloomfield, R., Littlewood, B., Popov, P., Povyakalo, A., Strigini, L.: A conservative bound for the probability of failure of a 1-out-of-2 protection system with one hardware-only and one software-based protection train. Reliab. Eng. Syst. Saf. **130**, 61–68 (2014)
8. Bishop, P., Bloomfield, R., Littlewood, B., Povyakalo, A., Wright, D.: Toward a formalism for conservative claims about the dependability of software-based systems. IEEE Trans. Softw. Eng. **37**(5), 708–717 (2011)
9. Bishop, P., Povyakalo, A.: Deriving a frequentist conservative confidence bound for probability of failure per demand for systems with different operational and test profiles. Reliab. Eng. Syst. Saf. **158**, 246–253 (2017)
10. Bloomfield, R., Khlaaf, H., Ryan Conmy, P., Fletcher, G.: Disruptive innovations and disruptive assurance: assuring machine learning and autonomy. Computer **52**(9), 82–89 (2019)
11. Bloomfield, R.E., Littlewood, B., Wright, D.: Confidence: its role in dependability cases for risk assessment. In: DSN 2007, pp. 338–346. IEEE, Edinburgh (2007)
12. Bloomfield, R., Bishop, P.: Safety and assurance cases: past, present and possible future - an adelard perspective. In: Dale, C., Anderson, T. (eds.) Making Systems Safer, pp. 51–67. Springer, London (2010)
13. Burton, S., Gauerhof, L., Heinzemann, C.: Making the case for safety of machine learning in highly automated driving. In: Tonetta, S., Schoitsch, E., Bitsch, F. (eds.) SAFECOMP 2017. LNCS, vol. 10489, pp. 5–16. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-66284-8_1
14. Burton, S., Gauerhof, L., Sethy, B.B., Habli, I., Hawkins, R.: Confidence arguments for evidence of performance in machine learning for highly automated driving functions. In: Romanovsky, A., Troubitsyna, E., Gashi, I., Schoitsch, E., Bitsch, F. (eds.) SAFECOMP 2019. LNCS, vol. 11699, pp. 365–377. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-26250-1_30
15. Chen, L., May, J.H.R.: A diversity model based on failure distribution and its application in safety cases. IEEE Trans. Reliab. **65**(3), 1149–1162 (2016)
16. Denney, E., Pai, G., Habli, I.: Towards measurement of confidence in safety cases. In: International Symposium on Empirical Software Engineering and Measurement, pp. 380–383 (2011)
17. Du, S.S., Lee, J.D., Li, H., Wang, L., Zhai, X.: Gradient descent finds global minima of deep neural networks. arXiv e-prints p. arXiv:1811.03804 (Nov 2018)
18. Ferrando, A., Dennis, L.A., Ancona, D., Fisher, M., Mascardi, V.: Verifying and validating autonomous systems: towards an integrated approach. In: Colombo, C., Leucker, M. (eds.) RV 2018. LNCS, vol. 11237, pp. 263–281. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-03769-7_15
19. Fukunaga, K.: Introduction to Statistical Pattern Recognition. Elsevier, New York (2013)

20. Galves, A., Gaudel, M.: Rare events in stochastic dynamical systems and failures in ultra-reliable reactive programs. In: FTCS 1998, pp. 324–333. Munich, DE (1998)
21. He, F., Liu, T., Tao, D.: Control batch size and learning rate to generalize well: theoretical and empirical evidence. In: NIPS 2019, pp. 1141–1150 (2019)
22. Huang, X., et al.: A survey of safety and trustworthiness of deep neural networks. arXiv preprint arXiv:1812.08342 (2018)
23. Huang, X., Kwiatkowska, M., Wang, S., Wu, M.: Safety verification of deep neural networks. In: Majumdar, R., Kunčak, V. (eds.) CAV 2017. LNCS, vol. 10426, pp. 3–29. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-63387-9_1
24. Ishikawa, F., Matsuno, Y.: Continuous argument engineering: tackling uncertainty in machine learning based systems. In: Gallina, B., Skavhaug, A., Schoitsch, E., Bitsch, F. (eds.) SAFECOMP 2018. LNCS, vol. 11094, pp. 14–21. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-99229-7_2
25. Johnson, C. W.: The increasing risks of risk assessment: on the rise of artificial intelligence and non-determinism in safety-critical systems. In: The 26th Safety-Critical Systems Symposium, p. 15. Safety-Critical Systems Club, York, UK (2018)
26. Kelly, T.P.: Arguing safety: a systematic approach to managing safety cases. Ph.D. thesis, University of York (1999)
27. Koopman, P., Kane, A., Black, J.: Credible autonomy safety argumentation. In: 27th Safety-Critical System Symposium Safety-Critical Systems Club, Bristol, UK (2019)
28. Littlewood, B., Rushby, J.: Reasoning about the reliability of diverse two-channel systems in which one channel is "possibly perfect". TSE **38**(5), 1178–1194 (2012)
29. Littlewood, B., Strigini, L.: 'Validation of ultra-high dependability...' - 20 years on. Safety Systems, Newsletter of the Safety-Critical Systems Club 20(3) (2011)
30. Littlewood, B., Povyakalo, A.: Conservative bounds for the pfd of a 1-out-of-2 software-based system based on an assessor's subjective probability of "not worse than independence". IEEE Trans. Soft. Eng. **39**(12), 1641–1653 (2013)
31. Littlewood, B., Salako, K., Strigini, L., Zhao, X.: On reliability assessment when a software-based system is replaced by a thought-to-be-better one. Reliab. Eng. Syst. Saf. **197**, 106752 (2020)
32. Littlewood, B., Wright, D.: The use of multilegged arguments to increase confidence in safety claims for software-based systems: a study based on a BBN analysis of an idealized example. IEEE Trans. Softw. Eng. **33**(5), 347–365 (2007)
33. Matsuno, Y., Ishikawa, F., Tokumoto, S.: Tackling uncertainty in safety assurance for machine learning: continuous argument engineering with attributed tests. In: Romanovsky, A., Troubitsyna, E., Gashi, I., Schoitsch, E., Bitsch, F. (eds.) SAFECOMP 2019. LNCS, vol. 11699, pp. 398–404. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-26250-1_33
34. Micouin, P.: Toward a property based requirements theory: system requirements structured as a semilattice. Syst. Eng. **11**(3), 235–245 (2008)
35. Musa, J.D.: Operational profiles in software-reliability engineering. IEEE Softw. **10**(2), 14–32 (1993)
36. O'Hagan, A., et al.: Uncertain Judgements: Eliciting Experts' Probabilities. Wiley, Chichester (2006)
37. Picardi, C., Habli, I.: Perspectives on assurance case development for retinal disease diagnosis using deep learning. In: Riaño, D., Wilk, S., ten Teije, A. (eds.) AIME 2019. LNCS (LNAI), vol. 11526, pp. 365–370. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-21642-9_46

38. Picardi, C., Hawkins, R., Paterson, C., Habli, I.: A pattern for arguing the assurance of machine learning in medical diagnosis systems. In: Romanovsky, A., Troubitsyna, E., Bitsch, F. (eds.) SAFECOMP 2019. LNCS, vol. 11698, pp. 165–179. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-26601-1_12
39. Ponti Jr., M.P.: Combining classifiers: from the creation of ensembles to the decision fusion. In: SIBGRAPI 2011, pp. 1–10. IEEE, Alagoas, Brazil (2011)
40. Ruan, W., Wu, M., Sun, Y., Huang, X., Kroening, D., Kwiatkowska, M.: Global robustness evaluation of deep neural networks with provable guarantees for the hamming distance. In: IJCAI 2019, pp. 5944–5952 (2019)
41. Rudolph, A., Voget, S., Mottok, J.: A consistent safety case argumentation for artificial intelligence in safety related automotive systems. In: ERTS 2018 (2018)
42. Rushby, J.: Software verification and system assurance. In: 7th International Conference on Software Engineering and Formal Methods, pp. 3–10. IEEE, Hanoi, Vietnam (2009)
43. Schwalbe, G., Schels, M.: Concept enforcement and modularization as methods for the ISO 26262 safety argumentation of neural networks. In: ERTS 2020 (2020)
44. Schwalbe, G., Schels, M.: A survey on methods for the safety assurance of machine learning based systems. In: ERTS 2020 (2020)
45. Sha, L.: Using simplicity to control complexity. IEEE Softw. **18**(4), 20–28 (2001)
46. Strigini, L., Povyakalo, A.: Software fault-freeness and reliability predictions. In: Bitsch, F., Guiochet, J., Kaâniche, M. (eds.) SAFECOMP 2013. LNCS, vol. 8153, pp. 106–117. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40793-2_10
47. Sun, Y., Wu, M., Ruan, W., Huang, X., Kwiatkowska, M., Kroening, D.: Concolic testing for deep neural networks. In: ASE2018, pp. 109–119. ACM (2018)
48. Zhao, X., Littlewood, B., Povyakalo, A., Strigini, L., Wright, D.: Modeling the probability of failure on demand (pfd) of a 1-out-of-2 system in which one channel is "quasi-perfect". Reliab. Eng. Syst. Saf. **158**, 230–245 (2017)
49. Zhao, X., Robu, V., Flynn, D., Salako, K., Strigini, L.: Assessing the safety and reliability of autonomous vehicles from road testing. In: The 30th International Symposium on Software Reliability Engineering, pp. 13–23. IEEE, Berlin, Germany (2019)