# Using Deep Learning for Schedulability Verification in Safety-Critical Systems

Sanjoy Baruah (Speaker) [*] Pontus Ekberg [†]

Deep Learning is already widely used in autonomous Cyber-Physical Systems (CPS's) such as self-driving cars, unmanned aerial vehicles, humanoid robots, etc., for purposes of perception – to enable the autonomous CPS to estimate its position, build a map of obstacles in its surroundings, and detect and track external objects such as pedestrians. Currently, research efforts are underway to also apply Deep Learning (DL) to *speed up computation* in CPS's: this is particularly meaningful for autonomous CPS's that are not tethered to the power grid and hence must make do with relatively simple computing platforms on board.

The *sporadic task model* [1] is widely used for modeling the timing behavior of CPS's that primarily comprise collections of independent recurrent processes executing upon a shared computing platform. Such systems are commonly scheduled during run-time using the Earliest-Deadline First (EDF) or some Fixed-Priority (FP) scheduling discipline.

We have been investigating the use of Deep Neural Networks (DNNs) to classify EDF- and FP-scheduled sporadic task systems as either *schedulable* –guaranteed to meet all deadlines– or not. These problems are known to be intractable even upon single-processor platforms: NP-hard for FP [2] and coNP-hard for EDF [3]. And while we were able to train DNN-based classifiers to be fairly accurate ($> 90\%$) in classifying task systems as schedulable or not, even for very small task systems (comprising just 2–3 tasks) we were unable to train DNNs that made no classification errors whatsoever. Classification errors are of two kinds: a *false negative* error classifies a schedulable system as being unschedulable, whereas a *false positive* error classifies an unschedulable system as being schedulable. While false negative errors have the unfortunate effect of causing schedulable systems to be to be needlessly rejected as unschedulable, *false positive errors constitute a safety hazard* since a potentially unschedulable system is misidentified as being schedulable. We must be able to eliminate all false positive errors if we are to use DL for schedulability-analysis for safety-critical systems.

**Proposed Approach.** To eliminate the possibility of false positive errors, we propose that when DNNs are used for schedulability-analysis and classify a system as being schedulable, they be additionally required to generate a justification for this decision in the form of a ***certificate*** – see Figure 1. We require that this certificate must be verifiable by a (different) algorithm that is based on 'traditional' algorithmic techniques in that it does not make use of Deep Learning and related AI techniques; it is only if this verification algorithm agrees that the certificate validates schedulability do we deem the system specifications to have passed the schedulability-analysis test.

[*]`baruah@wustl.edu`. Washington University in St. Louis, USA.

[†]`pontus.ekberg@it.uu.se`. Uppsala University, Sweden.

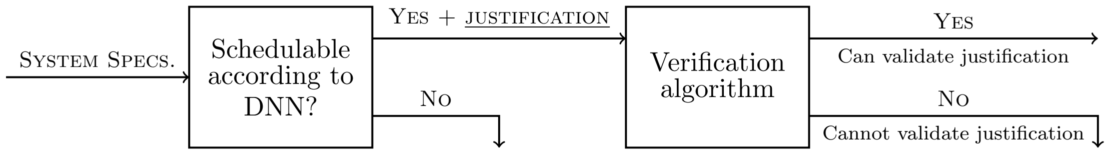
System Specs.
Schedulable according to DNN?
Yes + justification
No
Verification algorithm
Yes
Can validate justification
No
Cannot validate justification

Figure 1: A Framework for DNN-based Schedulability Verification

Proposition 1 immediately follows from the definition of the complexity class NP:

**Proposition 1.** Restricting that the module labeled "Verification algorithm" in Figure 1 have no worse than polynomial running time, it is necessary and sufficient for a schedulability condition to belong to the complexity class NP in order for it to be verifiable using the framework of Figure 1. □

Hence, in order to determine whether a schedulability-analysis problem can be verified or not using DL via the framework presented in Figure 1, it is necessary to demonstrate its membership or non-membership in the complexity class NP. To prove that a schedulability-analysis problem belongs to NP, one must furnish a polynomial-time verification algorithm for the problem. To demonstrate non-membership in NP (and hence unsuitability for solution using the framework of Figure 1), we must show the problem to be hard for some complexity class that is believed to be distinct from NP (i.e., they contain problems $\notin$ NP). Thus, *showing a schedulability-analysis problem to be hard for any complexity class believed to be distinct from* NP *provides substantial evidence that it is not amenable to solution using the framework of Figure 1.* It turns out that upon preemptive uniprocessors, FP schedulability $\in$ NP whereas EDF schedulability is coNP-complete (and hence unlikely to be in NP), and indeed we have been able to instantiate the framework of Figure 1 to do FP schedulability analysis, but not EDF schedulability analysis.

**Dealing with problems $\notin$ NP.** Real-time scheduling theory research has traditionally sought to obtain efficient algorithms for determining schedulability; if a particular schedulability analysis problem is shown to be computationally hard, one approach has been to try to identify sub-problems that are solvable in polynomial time[1]. If some schedulability analysis problem that arises frequently in practice is $\notin$ NP and therefore not amenable to solution using the framework of Figure 1, one could, by analogy, seek to identify sub-problems of this problem that are in NP. We have investigated this approach for the coNP-hard problem of EDF schedulability analysis [6], and have identified a variety of different (partially overlapping) subsets of the space of all EDF-schedulable sporadic task systems, that are each $\in$ NP and hence each subset can be verified using the framework of Figure 1. One could therefore instantiate a copy of the framework of Figure 1 for each of these subsets, by training different DNNs to generate verifiable justifications (certificates) for each of these subset, and consider a task system to be verified schedulable if any of these DNNs is able to generate a verifiable certificate.

[1]For instance, FP and EDF schedulability analysis are intractable in general [2, 3] as stated, but exact polynomial-time schedulability tests are known [4, 5] for *harmonic* task systems.

## References

[1] Giorgio C. Buttazzo. *Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications*. Second edition, 2005.

[2] Pontus Ekberg and Wang Yi. Fixed-priority schedulability of sporadic tasks on uniprocessors is NP-hard. In *2017 IEEE Real-Time Systems Symposium, RTSS 2017, Paris, France, December 5-8, 2017*, pages 139–146. IEEE Computer Society, 2017.

[3] Friedrich Eisenbrand and Thomas Rothvoß. EDF-schedulability of synchronous periodic task systems is coNP-hard. In *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms*, January 2010.

[4] V. Bonifaci, A. Marchetti-Spaccamela, N. Megow, and A. Wiese. Polynomial-time exact schedulability tests for harmonic real-time tasks. In *Proceedings of the 34th Real-Time Systems Symposium (RTSS)*, pages 236–245, Dec 2013.

[5] Thi Huyen Chau Nguyen, Werner Grass, and Klaus Jansen. Exact polynomial time algorithm for the response time analysis of harmonic tasks. In Cristina Bazgan and Henning Fernau, editors, *Combinatorial Algorithms*, pages 451–465, Cham, 2022. Springer International Publishing.

[6] Sanjoy Baruah and Pontus Ekberg. Towards Efficient Explainability of Schedulability Properties in Real-Time Systems. In Alessandro V. Papadopoulos, editor, *35th Euromicro Conference on Real-Time Systems (ECRTS 2023)*, volume 262 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 2:1–2:20, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.