Research Challenges for Combined Autonomy, AI, and Real-Time Assurance

Tarek Abdelzaher
Computer Science
University of Illinois
Urbana-Champaign, IL, USA
zaher@illinois.edu

Sanjoy Baruah
Computer Science and Engineering
Washington University in St. Louis
St. Louis, MO, USA
baruah@wustl.edu

Chris Gill
Computer Science and Engineering
Washington University in St. Louis
St. Louis, MO, USA
cdgill@wustl.edu

Eugene Vorobeychik
Computer Science and Engineering
Washington University in St. Louis
St. Louis, MO, USA
yvorobeychik@wustl.edu

Ning Zhang
Computer Science and Engineering
Washington University in St. Louis
St. Louis, MO, USA
zhang.ning@wustl.edu

Xuan Zhang
Electrical and Systems Engineering
Washington University in St. Louis
St. Louis, MO, USA
xuan.zhang@wustl.edu

2021 IEEE Third International Conference on Cognitive Machine Intelligence (CogMI) | 978-1-6654-1621-4/21/$31.00 ©2021 IEEE | DOI: 10.1109/COGMI52975.2021.00029

***Abstract*—Advances in machine intelligence revolutionized a broad category of safety-critical and mission-critical applications, but important challenges remain when applying these solutions at the embedded network edge, as opposed to resource-rich contexts. What challenges stem from deploying cost-sensitive applications on lower-end devices to offer AI at the point of need? We present an overview of key research challenges that must be addressed to provide assurance of timing and other safety properties for resource-constrained systems involving autonomy and artificial intelligence on-line. We then describe a vision and agenda for research targeting those challenges.**

***Index Terms*—autonomy, AI, ML, neural networks, real-time**

I. Introduction

Before safety-critical systems may be deployed in the field, their safety properties must typically be *verified*. For many safety-critical Cyber-Physical Systems (CPS's), such safety properties have an inherently "real-time" aspect (e.g., that deadlines are met); the verification of such timing correctness properties is usually done by applying analysis and modeling techniques from *real-time scheduling theory* [1]. Real-time scheduling theory was initially developed for the verification of timing properties in relatively simple static systems that were designed to operate in one of a limited number of operational modes, and in reasonably predictable environments. Although the theory has since evolved significantly to deal with more complex dynamic systems operating within unpredictable environments, we believe additional effort is needed to further extend the concepts, techniques, and methods of real-time scheduling to make them applicable for the verification of timing safety properties of autonomous AI-based CPS's such as self-driving cars and unmanned aerial vehicles. The aforementioned challenges are further compounded by resource constraints in environments where the AI algorithms must execute on embedded edge devices as opposed in a resource-rich environment, such as the cloud.

For example, in drone cinematography, an unmanned aerial vehicle not only performs low-level functions such as object recognition and tracking, but also determines its own flight path to optimize the quality of the video it is recording of a subject, according to specific cinematographic objective functions. All of the above must fit on an embedded computing platform that is subject to size, weight, and power constraints to promote cost-efficient design and fuel-efficient operation. Similarly, self-driving cars must avoid collisions with other manned and unmanned vehicles, pedestrians, and features of the roadway while maintaining adequate rates of travel and navigating efficiently to a target destination. Mass production (at a scale of millions of units annually) implies that small per-unit cost savings have significant financial implications. This cost-sensitivity motivates careful use of lower-end computing platforms where possible, as opposed to resorting to over-provisioning. While many intelligent capabilities have been demonstrated in particular contexts with limited sources of potential interference, generalizing these capabilities and ensuring safety properties of these systems within a diverse range of complex operating environments and resource constraints poses important new challenges for CPS research.

II. Research Challenges and Approaches

A new generation of CPS research must address the aforementioned needs at the intersection of autonomy, AI, and real-time assurance. First, timing characteristics of modern AI components must be studied, and analytical models of those components' timing behavior must be developed so that their contributions to overall timing properties of the system can be gauged and verified. Second, especially for systems operating in complex and evolving environments, notions of prioritization must be expanded to encompass attention to the external environment (object tracking, projecting future trajectories) as well as the system's internal resources (scheduling, control, and system state estimation). Specifically, attention should be given to more critical elements of the environment in accordance with their physical properties and related

978-1-6654-1621-4/21/$31.00 ©2021 IEEE
DOI 10.1109/CogMI52975.2021.00029

safety constraints. Third, managing resources efficiently in the common cases, as well as in worst-case scenarios is essential. Fourth, resource access and availability, from a security perspective as well as a real-time one, also must be addressed. Fifth, adapting to rapidly changing environments, while maintaining adequate operational tempo should allow safe operation while maintaining suitable performance. Sixth, how to identify and mitigate side channel vulnerabilities in perception and other interactions with the environment also must be investigated. The combined impact of these advances has significant potential to enable a new generation of autonomous, artificially intelligent, safety-critical cyber-physical systems whose performance and safety properties (including timing, availability, security, and resilience) can be verified formally and validated empirically.

A. Timing Characterizations of Execution Workloads

In order to be able to provide timeliness guarantees, it is necessary that we have accurate characterizations of the timing requirements of the kinds of computations that must be performed in a time-sensitive manner. Obtaining such accurate characterizations in particularly challenging for complex autonomous safety-critical CPS's that are intended for operation in highly uncertain and widely dynamic environments, and that are, for cost and related reasons, implemented upon general-purpose computing platforms that are optimized for providing superior average-case performance (rather than minimizing worse-case timing durations).

In order to obtain strong empirical evidence of the computational resource requirements of modern AI components such as deep neural networks, extensive measurement experiments are needed upon a variety of implementation platforms of the kind that are widely used in implementing safety-critical autonomous CPS's today. On the basis of the evidence obtained via such measurements, abstract models can be constructed of run-time timing behavior that are both able to accurately characterize resource requirements and are amenable to the forms of analysis (such as schedulability analysis, timing analysis, etc.) that are needed in order to be able to prove timing safety properties.

B. Prioritization of Attention

An important challenge brought about by the need for resource efficiency is one of prioritizing machine attention [2]. Current execution pipelines of machine intelligence tasks (such as perception) generally treat the sensory input data stream as a uniform stream of the same priority. In reality, elements of the scene could be prioritized at a sub-frame level according to some notion of criticality. For example, intuition suggests that in autonomous driving, trees on the side of the road are not as important to track as pedestrians. Thus, trees could be observed at a much lower frame rate than the individuals. In general, static and immobile objects need not be observed as frequently as fast unpredictably-moving objects. In current systems, both are observed at the same frame rate, which is wasteful to resources and needlessly increases cost.

On the other hand, objects that may seem unimportant in the short term, may prove important in the near future. For example, a truck may obscure a vehicle that is passing it on the other side, relative to an autonomous vehicle that is tracking all nearby vehicles. Even though that other vehicle may be obscured from the autonomous vehicle's cameras and other sensors for a minute or more, its prior trajectory and the potential for it to emerge ahead of the truck and even move directly into the path of the autonomous vehicle must be considered, and attention and resources thus must be devoted to tracking it when it is visible and projecting its likely trajectory when it is not.

Addressing the above challenges introduces a "chicken-and-egg" problem. In order to segment the input and assign importance levels, priorities, deadlines, or effective observation rates to different objects or segments, these objects or segments must first be identified in the scene. Such identification, in itself, requires perception processing, thus negating the resource savings. In principle, however, it may possible to use auxiliary cueing sensors to determine areas of input that are more critical to observe. For example, a ranging sensor could be used to efficiently detect nearby obstacles and/or quickly-approaching objects allowing the corresponding areas of the visual input to be prioritized before any heavy-weight processing is done by the video perception pipeline [2], [3].

The above idea introduces several challenges. For example, what cueing sensors are appropriate? How to make cueing efficient so we do not end up spending the same amount of resources on cueing as we would have on actual whole seen processing? How to mitigate false positives and eliminate false negatives? What is a meaningful priority assignment? How to take advantage of redundancy among successful frames to further reduce processing overhead? In addition to model-based (and thus potentially explainable) tracking, prediction, and reasoning approaches, we believe real-time scheduling theory, which has studied priority-based resource-assignment extensively, is uniquely suited to answer some of these questions. Thus, it is important to investigate the use of concepts and techniques from real-time scheduling theory to better understand the role of priorities in achieving real-time assurance in autonomous AI-based CPS's.

C. Mechanisms for Resource Economy

Intelligent systems are data driven. Most computation is attributed to the processing of data-intensive pipelines to compute decisions based on observations. Given appropriate attention management mechanisms discussed above, another design decision is thus how to reduce resource consumption attributed to less critical input data segments. Several degrees of freedom can be explored, as discussed below.

1) Region-of-Interest Selection and Subsampling: One mechanism for reducing resource consumption is to process less critical data segments less frequently (i.e., sub-sample parts of the input stream). Other mechanisms are possible that rely on processing quality, not frequency. In essence, reducing processing frequency can be thought of as an extreme point

in the quality space, where the quality of processing of some inputs (namely, those that were skipped) has been reduced to zero. Examples of other points in that design space are presented below.

2) Early Exit and Imprecise Computations: A simple solution that is a compromise between *full processing* and *total skipping* of regions of input is to apply the concept of imprecise computations [4]. Since modern perception pipelines are based on neural networks, it becomes possible to design networks where processing additional layers increases the quality of outputs incrementally. Such a design was recently proposed for classification tasks [5]. The resources consumed by processing of lower-criticality inputs can thus be reduced by early exit mechanisms that stop neural network processing of less critical segments before all network layers have been executed, thus producing intermediate output quality.

3) Input Resolution Adjustment and Model Switching: Recent work demonstrated that another effective solution for reducing the amount of resources spent on processing less critical inputs is to simply reduce the resolution of such inputs [6] and use smaller neural networks to process the smaller inputs. It was shown, in fact, that this approach beats imprecise computations, described above, because for each network (of a different size), network weights can be specifically optimized for its corresponding tailored input size. This optimization is as opposed to early exit networks, where multiple exit points exist in *the same neural network* making it impossible to optimize neural network architecture and weights for each specific exit point. Rather, the choice of used weights becomes more of a compromise that jointly considers all possible exit points, making the network up to each exit point somewhat suboptimal for its size.

More work is needed to understand the advantages and limitations of imprecise computations versus model switching approaches. An obvious disadvantage of the latter, for example, is that all the different network versions need to fit in GPU memory upfront to avoid excessive context switching overhead. This need increases memory requirements compared to imprecise computations. Recent work attempts to reduce this problem by techniques such as weight virtualization [7]. It is also not clear if imprecise computations can be applied to all neural network functions, such as, for example, object detection. More work is needed to develop good imprecise computation models for different deep neural network inference tasks.

4) Neural Network Compression: Another solution to trade-off neural network quality for resource savings is neural network compression. Many algorithms for compressing neural networks have been proposed in recent years [8], [9]. It is therefore possible to use compressed networks for processing less important inputs, notwithstanding input size. While some compression techniques simply rely on numerical approximations (such as quantizing weights, using integer arithmetic instead of floating point, or removing smaller weights), other approximations are trained with specific loss functions in mind. For example, a network approximation can be trained in a manner that optimizes detection performance for a *subset* of object types [8]. Such approximations are particularly well-suited to a context where the compressed network should retain a higher level of output (e.g., classification) quality for a subset of object classes.

5) Computational Offloading: When applicable, some neural network computation can be offloaded to remote nodes. This solution is especially well-suited (as an alternative or supplement to quality reduction) for processing less critical regions, where challenges such as network outages will not interfere with critical operation. Neural network offloading brings about challenges in deciding the degree of compression for offloaded feature vectors and efficiently navigating the trade-off between the computational overhead of compression, the bandwidth needed for compressed data, and the degree of quality loss entailed. For example, recent work introduced an asymmetric encoder/decoder framework [10], where the compression of feature vectors to be offloaded is much more efficient than decoding of the compressed signal, motivated by the asymmetry between the edge nodes and central services.

6) Surrogate Sensing: Another idea in the trade-off space is to optimize for the common-case by using simpler sensors and inference algorithms in that case, and escalating processing to more computationally-complex components and sensors only when needed. For example, a security system might consume fewer resources when no motion is detected. Motion detection does not require complex neural network processing and thus can be used to short-cut the rest of the pipeline when appropriate. This simple concept can be scaled to other common scenarios, such as, for example, motion signatures that are limited in size making them unlikely to be caused by a human. In general, simple (e.g., compressed) neural networks can be used to detect a set of common conditions. When deviations from these conditions are detected, larger networks may be invoked.

The above performance differentiation mechanisms give rise to new models of computation that differentiate both the quality and resource consumption, on subframe basis, depending on input criticality. These capabilities present interesting modeling and resource allocation challenges that allow giving quality and timeliness guarantees to the processing of more critical data regions, while optimizing some aggregate performance measure, subject to capacity constraints, for other objects.

D. Timely Resource Access and Availability

A fundamental difference between CPSs and conventional computing systems is their interactions with the physical world. While many computation tasks in IT systems can be suspended for an extended period of time, calculations in CPSs often have hard deadlines, since time continues to elapse in the physical world. Computationally correct, but untimely results often have little to no value for system control, and can even destabilize the physical system.

The security of AI has received significant attention in recent years, where attacks often focus compromising the

confidentiality (such as membership interference attacks) and the integrity (such as adversarial examples and data poisoning). Little attention has been given to the availability aspect. However, with the growing importance of AI in cyber-physcial systems and the emergence of ubiquitous autonomy, the availiability of AI components is a pressing open challenge for the security and safety of these critical systems. Recently, it has been demonstrated that timing manipulation can lead to control destablization of cyber-physical systems [11]–[13].

Availability often implies timely access to system resources in the cyber-physical systems. There are two general attack vectors, attacking the AI components or the system it relies on to impact of timing behavior of the system. In [11], Shumailov et al. explored the use adversarial samples to cause significant (up to 30x) amount of additional energy consumption and delay on the DNN. The key observation behind the attack is that inputs of the same size can cause a DNN to consume significantly different amounts of energy and time due to use of hardware and algorithmic optimizations. In [12], Li et al. considers a different attack vector where the attacker leverages the resource contention to cause significant delay in the run-time characteristics of AI-powered control algorithms, leading to control destablization. We envision that in order to provide full system availability, it is important to take a hollistic approach in order to not only secure against timing anomalies in neural network execution but also to protect the platform the AI runs on.

E. Rapidly Changing Operating Environments

We have seen above that time-critical execution workloads in autonomous safety-critical CPS's may change significantly and rapidly; strategies are needed for accommodating such time-varying computational demands in an effective and efficient manner. *Elastic task models* [14]–[17], widely studied in the real-time scheduling literature, appear particularly appropriate for representing such workloads: these models possess the expressive capabilities to represent both the variation in the amount of computing that is needed by each individual task, and the task's resilience to being under-served (i.e., not receiving its entire requested amount of execution). Scheduling and schedulability-analysis algorithms have been developed for these task models that then seek to schedule them in a robust/ resilience manner as the workload changes during system execution time. We believe such elastic task models, suitably adapted, may prove useful for representing the kinds of dynamic workloads that are found in many safety-critical AI-based autonomous CPS's.

Another approach to dealing with a CPS's rapidly-changing computational needs would be to adapt the CPS's operational tempo in response to an increase/ decrease in the availability of computation. However, trade-offs between performance objectives and safety constraints make such elastic adaptation an ongoing consideration. For example, a self-driving car may slow down in dense and complex urban settings, while maintaining reasonable progress towards a planned destination - how to incorporate deadlines (time of arrival) as well as constraints (maximum safe rate of travel based on proximity and movement of pedestrians and other vehicles) within a constrained optimization procedure that *runs continuously to shape overall system behavior as it moves* is an important research challenge.

This in turn suggests that new formal models for *cyber-physical elasticity* through which a system may adapt its operational tempo or other timing aspects to respect constraints and optimize performance are needed. Tractability of exact techniques (e.g., full state space exploration) for off-line or on-line use of model-based approaches, and the potential role of stochastic and/or approximate approaches that can give strong probabilistic bounds on response timing, solution quality, etc., must be investigated to determine what can be done at run-time as the system is operating, versus what must be done *a priori* off-line.

These models also will have security implementations, and platform level approaches for applying them must minimize timing-based attack surfaces, e.g., through control algorithms that can accommodate additional timing jitter, because events they receive are time stamped. For both verification and mitigation of adversarial vulnerabilities, hardware-in-the-loop simulation platforms (e.g., combining ROS components running on a Jetson board with AirSim or CARLA) are likely to be valuable for studying timing induced vulnerabilities of self-driving cars, and autonomous drones, and other autonomous systems, under various scenarios.

F. Perceptual Side-Channel Vulnerabilities

While leveraging perceptual data streams for adaptive scheduling, such as in order to prioritize attention, presents tremendous opportunities for increasing the efficacy of resource utilization in real-time systems, these also create a new side-channel vulnerability in the form of physical attacks on AI-based perception processing. In particular, modern neural network architectures engaged in a variety of perceptual tasks have been shown to be vulnerable to *adversarial example* attacks, where digital inputs or even the external physical environment are maliciously manipulated to effect a change in prediction semantics, such as misclassification of objects in scenes [18]–[20]. One of the key research challenges in adaptive scheduling schemes will be to provide adequate assurance that these do not fall prey to such attack vectors, for example, by composing perceptual robustness guarantees (which are, by their nature, input-varying) with real-time scheduling guarantees.

III. Concluding Remarks and Future Work

This paper briefly outlined challenges in adapting AI for CPS applications, with a focus on deploying AI at the point of need – namely, on the embedded edge, where data are collected by a myriad of sensors that execute a new breed of analytics, accommodate resource constraints, optimize for a dynamic environment, and survive an expanded range of threats. The rationale for pushing intelligent computations to the edge in a broad category of CPS applications lies in

operational efficiency and resilience. By pushing computations to where data originate, needless dependence on remote or centralized resources is removed, thereby simultaneously improving end-to-end latency, robustness, security, and resource economy. The paper calls for a research agenda on resource management in the above context. In general computing systems, streamlining application development and operation necessitates the introduction of operating systems to address common challenges such as efficiency, robustness, scalability, and responsiveness. In systems, where cyber-physical capabilities intertwine physical edge resource management with intelligent computational artifacts, a new operating-system-like construct is needed in order to ensure that the execution of various decision loops involved at different spatial and temporal scale meets the challenges named above. Fundamentally, these challenges are partitioned onto (i) performance optimizations to significantly reduce the end-to-end latency, and computational and communication resource needs of intelligent components, and (ii) resilience solutions to guarantee correctness in the presence of a myriad of cyber threats. The paper invites multi-disciplinary efforts to address the above challenges.

ACKNOWLEDGMENT

Research reported in this paper was sponsored in part by the Army Research Laboratory under Cooperative Agreement W911NF-17-2-0196 and NSF under award CPS 20-38817, and in part by The Boeing Company. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory, NSF, Boeing, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

REFERENCES

[1] L. Sha, T. Abdelzaher, A. Cervin, T. Baker, A. Burns, G. Buttazzo, M. Caccamo, J. Lehoczky, A. K. Mok *et al.*, "Real time scheduling theory: A historical perspective," *Real-time systems*, vol. 28, no. 2, pp. 101–155, 2004.

[2] S. Liu, S. Yao, X. Fu, R. Tabish, S. Yu, A. Bansal, H. Yun, L. Sha, and T. Abdelzaher, "On removing algorithmic priority inversion from mission-critical machine inference pipelines," in *2020 IEEE Real-Time Systems Symposium (RTSS)*. IEEE, 2020, pp. 319–332.

[3] S. Liu, S. Yao, X. Fu, H. Shao, R. Tabish, S. Yu, A. Bansal, H. Yun, L. Sha, and T. Abdelzaher, "Real-time task scheduling for machine perception in in intelligent cyber-physical systems," *IEEE Transactions on Computers*, 2021.

[4] J. W. Liu, W.-K. Shih, K.-J. Lin, R. Bettati, and J.-Y. Chung, "Imprecise computations," *Proceedings of the IEEE*, vol. 82, no. 1, pp. 83–94, 1994.

[5] S. Yao, Y. Hao, Y. Zhao, H. Shao, D. Liu, S. Liu, T. Wang, J. Li, and T. Abdelzaher, "Scheduling real-time deep learning services as imprecise computations," in *2020 IEEE 26th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*. IEEE, 2020, pp. 1–10.

[6] Y. Hu, S. Liu, T. Abdelzaher, M. Wigness, and P. David, "On exploring image resizing for optimizing criticality-based machine perception," in *2021 IEEE 27th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*. IEEE, 2021, pp. 169–178.

[7] S. Lee and S. Nirjon, "Fast and scalable in-memory deep multitask learning via neural weight virtualization," in *Proceedings of the 18th International Conference on Mobile Systems, Applications, and Services*, 2020, pp. 175–190.

[8] S. Yao, Y. Zhao, A. Zhang, L. Su, and T. Abdelzaher, "Deepiot: Compressing deep neural network structures for sensing systems with a compressor-critic framework," in *Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems*, 2017, pp. 1–14.

[9] S. Yao, Y. Zhao, H. Shao, S. Liu, D. Liu, L. Su, and T. Abdelzaher, "Fastdeepiot: Towards understanding and optimizing neural network execution time on mobile and embedded devices," in *Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems*, 2018, pp. 278–291.

[10] S. Yao, J. Li, D. Liu, T. Wang, S. Liu, H. Shao, and T. Abdelzaher, "Deep compressive offloading: Speeding up neural network inference by trading edge computation for network latency," in *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*, 2020, pp. 476–488.

[11] I. Shumailov, Y. Zhao, D. Bates, N. Papernot, R. Mullins, and R. Anderson, "Sponge examples: Energy-latency attacks on neural networks," in *IEEE European Symposium on Security and Privacy (EuroS&P)*, 2021.

[12] A. Li, J. Wang, and N. Zhang, "Chronos: Timing interference as a new attack vector on autonomous cyber-physical systems," in *ACM Conference on Computer and Communications Security (CCS)*, 2021.

[13] R. Mahfouzi, A. Aminifar, S. Samii, M. Payer, P. Eles, and Z. Peng, "Butterfly attack: Adversarial manipulation of temporal properties of cyber-physical systems," in *2019 IEEE Real-Time Systems Symposium (RTSS)*. IEEE, 2019, pp. 93–106.

[14] G. C. Buttazzo, G. Lipari, and L. Abeni, "Elastic task model for adaptive rate control," in *1998 IEEE Real-Time Systems Symposium (RTSS)*, 1998.

[15] G. C. Buttazzo, G. Lipari, M. Caccamo, and L. Abeni, "Elastic scheduling for flexible workload management," *IEEE Trans. Comput.*, vol. 51, no. 3, pp. 289–302, Mar. 2002. [Online]. Available: http://dx.doi.org/10.1109/12.990127

[16] T. Chantem, X. S. Hu, and M. D. Lemmon, "Generalized elastic scheduling," in *2006 27th IEEE International Real-Time Systems Symposium (RTSS'06)*, 2006, pp. 236–245.

[17] ——, "Generalized elastic scheduling for real-time tasks," *IEEE Transactions on Computers*, vol. 58, no. 4, pp. 480–495, April 2009.

[18] A. Boloor, K. Garimella, X. He, C. Gill, Y. Vorobeychik, and X. Zhang, "Attacking vision-based perception in end-to-end autonomous driving models," *Journal of Systems Architecture*, vol. 110, p. 101766, 2020.

[19] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. X. Song, "Robust physical-world attacks on deep learning visual classification," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1625–1634.

[20] M. Sharif, S. Bhagavatula, L. Bauer, and M. K. Reiter, "Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition," in *ACM SIGSAC Conference on Computer and Communications Security*, 2016, pp. 1528–1540.

[21] D. Casini, T. Blaß, I. Lütkebohle, and B. B. Brandenburg, "Response-time analysis of ROS2 processing chains under reservation-based scheduling," in *31st Euromicro Conference on Real-Time Systems (ECRTS 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2019.