

Received 15 December 2023, accepted 1 January 2024, date of publication 23 January 2024, date of current version 2 February 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3357510

APPLIED RESEARCH

An Extreme-Edge TCN-Based Low-Latency Collision-Avoidance Safety System for Industrial Machinery

MARCELLO ZANGHIERI¹, (Graduate Student Member, IEEE),
FABRIZIO INDIRLI², (Graduate Student Member, IEEE), ANTONIO LAPELLA³,
GIACOMO MICHELE PUGLIA⁴, FELICE TECCE⁵, FRANCESCO PAPARIELLO⁶,
GIULIO URLINI⁶, LUCA BENINI^{1,7}, (Fellow, IEEE),
AND FRANCESCO CONTI¹, (Member, IEEE)

¹Department of Electrical, Electronic and Information Engineering, University of Bologna, 40136 Bologna, Italy

²Department of Electronics, Information and Bioengineering, Politecnico di Milano, 20133 Milan, Italy

³SCM Group S.p.a., 47921 Rimini, Italy

⁴Bitron Industrie S.p.A., 10095 Grugliasco, Italy

⁵STMicroelectronics, 80022 Arzano, Italy

⁶STMicroelectronics, 20864 Agrate Brianza, Italy

⁷Integrated Systems Laboratory, Department of Information Technology and Electrical Engineering, ETH Zürich, 8092 Zürich, Switzerland

Corresponding author: Marcello Zanghieri (marcello.zanghieri2@unibo.it)

This work was supported in part by the EU Horizon 2020 (ECSEL Joint Undertaking) Project AI4DI under Grant 826060, and in part by the EU Horizon 2020 Project BonsAPPs under Grant 101015848.

ABSTRACT Modern manufacturing industry relies on complex machinery that requires skills, attention, and precise safety certifications. Protecting operators in the machine's surroundings while at the same time reducing the impact on the normal workflow is a major challenge. In particular, safety systems based on proximity sensing of humans or obstacles require that the detection is accurate, low-latency, and robust against variations in environmental conditions. This work proposes a functional safety solution for collision avoidance relying on Ultrasounds (US) and a Temporal Convolutional Network (TCN) suitable for deployment directly at the edge on a low-power Microcontroller Unit (MCU). The setup allowed to acquire a sensor-fusion dataset with 9 US sensors mounted on a real industrial woodworking machine. Applying incremental training, the proposed TCN achieved sensitivity 90.5%, specificity 95.2%, and AUROC 0.972 on data affected by the typical acoustic noise of an industrial facility, an accuracy comparable with the State-of-the-Art (SoA). Deployment on an STM32H7 MCU yielded a memory footprint of 560 B ($3\times$ less than SoA), with an extremely low latency of 5.0 ms and an energy consumption of 8.2 mJ per inference (both $>2.3\times$ less than SoA). The proposed solution increases its robustness against acoustic noise by leveraging new data, and it fits the resource budget of real-time operation execution on resource-constrained embedded devices. It is thus promising for generalization to different industrial settings and for scale-up to wider monitored spaces.

INDEX TERMS Collision avoidance, embedded systems, incremental learning, microcontroller, public dataset, real-time, temporal convolutional networks (TCN), time series, TinyML, ultrasounds.

I. INTRODUCTION

Nowadays several industrial sectors employ autonomous moving machinery that can constitute a source of hazard and must therefore be operated by workers with specific training and skills, with well-defined safety practices and working

conditions. A major concern is safeguarding operators' health. Solutions to do so with a reduced impact on the workflow of the machinery aiming at achieving high productivity and safety are currently an active field of research & development.

Industrial machines can be equipped with sensors that enable them to continuously monitor their surroundings in an automated way. This enables safeguards that halt

The associate editor coordinating the review of this manuscript and approving it for publication was Xiong Luo.

operations and drive the machinery to a safe state if people or dangerous obstacles are detected. The technical challenge in this scenario is to make the detection robust against variations in environmental conditions across multiple deployment sites (i.e., in “space”) and across several operational conditions of the same site (i.e., in “time”). Safety systems operating in an automated way belong to the domain of *functional safety* [1], where protection is framed and implemented as an *active, input-output* system. The safety function is the action generated in response to the processed input. Functional safety does not include *passive* systems (e.g., thermal insulation or fire-resistant doors) but involves electronics, software, and actuators.

Recently, methods based on Machine Learning (ML), and specifically Deep Learning (DL), have been gaining adoption in domains such as machine vision and data analytics [2]. Deep Neural Networks (DNNs) can now be regarded as a mature methodology in data analysis. Hence, DNNs are also promising for information processing tasks of active systems for functional safety since they can integrate multiple data streams and extract information from them. As to execution, ML/DL algorithms can be run in the cloud or at the edge, i.e., locally on a platform closely connected to the devices acquiring data [3], [4]. More specifically, recent advances in the field of Tiny Machine Learning (Tiny ML) [5], [6], [7] are enabling the porting of real-time ML inference onto embedded computing platforms with strict constraints in terms of memory or power envelope, such as microcontroller units (MCUs) [8], sometimes equipped with accelerators for ML/DL. For safety-critical systems, processing the data near the sensors can enhance reliability and ultra-low latency and increase the trust in ML-based solutions in industries such as manufacturing, mobility, and robotics [9], [10].

Safety solutions relying on ML/DL make it necessary to elaborate and advance the international standards that regulate functional safety. The major challenge is that the current versions of international standards do not cover novel, most recently introduced technologies and paradigms. This is an issue since innovative methods or algorithms can not be certified by definition. Hence, innovative solutions, even if proven effective for operators’ safety and production efficiency, undergo a large delay before inclusion into a new version of a standard; in turn, inclusion happens when a solution is mature and able to induce industrial interest in its inclusion. For this reason, the adoption of ML/DL-based solutions in Electro-Sensitive Protective Equipment (ESPE) systems [11] has not been addressed yet by any industrial safety standard. The authors’ stance in this regard is that interest from the industry must be fostered by showcasing innovative prototypes able to demonstrate the power of ML/DL for functional safety: this is the direction of the research presented in this work.

This work targets the specific domain of industrial woodworking machinery. It proposes a functional safety prototype for collision avoidance based on ultrasound (US) sensing and processing based on a Temporal Convolutional

Network (TCN), a DNN specialized for time series. The system is able to detect persons or obstacles in the field of view of the US sensors, which are mounted on the woodworking machine in such a way as to probe the space of operation of the machine’s moving parts. A detection triggers a stop of machine movement in real time. In detail, the contribution is multiple:

- This work implements a system based on 9 US sensors, an FPGA, and an MCU, mounted onto an industrial woodworking machine.
- The setup is used to collect a dataset for the detection task (i.e., clear space vs. human or obstacle), representative also of the acoustic noise conditions typical of an industrial facility, which are challenging since they impact the US signals; this curated dataset contains a total of 5085 US signal windows organized in 170 runs of the system in different obstacle and noise conditions.
- A TCN trained and tested for the purpose achieved sensitivity 96.7%, specificity 99.1%, and AUROC 0.993 in the absence of acoustic noise.
- In the presence of noise, exploiting an incremental learning technique proved that the proposed setup and model are able to leverage increasing amounts of data, attaining sensitivity 90.5%, specificity 95.2%, and AUROC 0.972.
- Deployment of the proposed TCN on the STM32H743ZI MCU yielded a profiling which outperforms the State-of-the-Art (SoA) TCN model for the task [12]: memory footprint of 560 B ($3\times$ smaller than SoA), with a latency of 5.0 ms and energy consumption of 8.2 mJ per inference (both $2.3\times$ less than SoA).

The proposed solution improves detection robustness against acoustic interference characteristic of a manufacturing environment, working with a resource budget fit for real-time execution on resource-constrained edge computing platforms. Table 1 reports a scheme of the advances of this work compared to the SoA represented by Conti et al. [12]. The proposed paradigm is generalizable to different sectors; in particular, the limited hardware requirements allow the scale-up of the approach, enabling adoption in scenarios with more sensors and, thus, wider monitored space in terms of the number of machines and extent of the probed areas.

For reproducibility and advance in the research & development community, this work also releases the curated dataset¹ and publishes open-source the developed code.²

II. RELATED WORK

A. SAFETY SYSTEMS IN INDUSTRIAL WOODWORKING MACHINERY

1) GENERAL

Industrial woodworking machines typically have a static base and a moving cabinet that slides horizontally at speed

¹https://github.com/MarcelloZanghieri2/collision_avoidance_ultrasound_dataset

²https://github.com/MarcelloZanghieri2/edge_tcn_collision_avoidance

TABLE 1. Contribution of this work in terms of the advances compared to the SoA represented by Conti et al. [12].

WORK	# of US sensors	Realized dataset		Method for accuracy on noisy data	Embedded system		Reproducibility	
		novel?	size		deployed @ edge?	real-time?	published dataset?	published code?
Conti et al. [12]	1	✓	227	data augmentation up to 1000×	✓	✓	✗	✗
This work	9	✓	5085 (22.4×)	incremental learning (augmentation just 64×)	0.32× parameter memory < 0.43× energy	< 0.43× latency	✓	✓

up to 1 m/s and operates over a working surface of the order of 4 m × 1.5 m [13]. Overall, these machines have a length of 5 - 10 m, a width around 5 m, and a height of 1 - 3 m [14], [15]. The moving cabinet can hit operators or objects, causing severe injuries or damage. In general, existing machine models rely on both active and non-active safety systems [13], [14], [15], [16], [17]. Non-active safety includes simple elements such as enclosures of the working units by fences, lateral curtain guards, transparent hatches, or perspex windows, based on the desired tradeoff of protection vs. accessibility and visibility. LEDs signal the machine status in real-time with a simple color code. This work focuses on more advanced active safety systems.

Active safety is based on real-time anti-collision systems required to operate while machines work at medium or maximum speed in a premise shared with workers performing regular work in the surroundings. Since detecting hazardous situations forces the machine to a safe-state mode, which can be unlocked only manually, erroneous automatic detection can cause a slowdown in the workflow. Active safety systems include: soft bumpers that stop the machine in case of accidental contact with persons or objects; pressure-sensitive floor mats; photocell barriers that detect the approach of persons or objects, automatically reduce the speed of the machine, and restore the maximum speed when the obstacle leaves the area; laser scanners that only enable the machine to start after the operator has left the area; automatic verification of the locking systems' positioning.

The proposed setup exploits US signals, processed for detecting objects or people within the space of operation of the machine. Compared to existing solutions, the proposed setup has several advantages. First, the proposed solution is a proximity sensor designed to trigger the stop of the machine before a collision, in contrast to bumpers. As to established collision avoidance systems, the existing laser scanners only probe a horizontal plane (at a height < 1 m above the floor) [14], whereas the proposed ultrasound sensors probe a 3D field of view. Compared to all alternative setups, including photocell barriers, the proposed solution can improve its detection accuracy during its lifetime since the proposed DNN benefits from incremental learning from data acquired in new conditions.

It is worth remarking that the technical documentation uses the term *collision avoidance* also for potential collisions between machinery's equipment or between tools and

material, handled during the virtual prototyping of the piece and the simulation and scheduling of numerical control positioning [13], [15], [16]; this kind of internal collision is not related to the topic of this work. It is also worth stressing that this work does not deal with inner systems for safety or maintenance such as air conditioning of electrical components or automatic lubrication.

2) BASED ON ULTRASOUNDS AND DL

A relevant earlier work tackling US-and-DL-based functional safety for woodworking machinery is by Conti et al. [12], who employ TEMPONet, a TCN previously applied to embedded biosignal processing in real-time [18], [19]. The previous work by Conti et al. [12] stemmed from the same project as this paper but only has the nature of a technical report documenting an incomplete stage of the research. Although a direct accuracy comparison is not viable since [12] relies on a different 1-channel dataset, it is possible to highlight several advancements (also reported in Table 1): (i) the proposed system mounts 9 ultrasound sensors, whereas the previous work mounted just 1; (ii) this work releases the dataset open-source; (iii) this work employs a smaller DNN, reducing the hardware resources and latency budget for execution; (iv) this work tackles a noisy environment by implementing an incremental training protocol instead of brute-force data augmentation.

B. RATIONALE OF THIS WORK IN RELATION TO THE ESTABLISHED FUNCTIONAL SAFETY STANDARDS

All safety equipment applied on industrial machines must get certified according to standards, such as the ones by the International Electrotechnical Commission (IEC) (covering electrical, electronic, and related technologies), that define the Safety Integrity Level to be met. Machinery-halting safety systems such as [12] and the one presented in this work fall under the regulations concerning non-contact Electro-Sensitive Protective Equipment (ESPE) sensors (e.g., photodiodes). More in detail, IEC 61508 [1] regards any electrical/electronic/programmable electronic (E/E/PE) for functional safety systems, such as sensors, control logic, or actuators, and also microprocessors; EN IEC 61496 [11] focuses on the requirements of design, building, and verification of systems based on non-contact ESPEs to detect persons in a safety system, focusing on indoor environments; EN IEC

62046 [20] addresses ESPEs for human detection for safety, focusing on industrial environments with machinery.

The authors are aware that novel ML/DL-based research & development prototypes such as the one presented in this work are not covered by current standards, nor can they receive certification in the short term. This limitation means that, as of today, developing finalized products based on the presented proof-of-concept is not possible. *The purpose of this work is to push research and technical expertise ahead of current standards and certifications.* The authors' motivation in undertaking the present research is to showcase how promising data-driven safety systems are, intending to incentivize both technical exploration and regulatory interest. The authors believe that this line of research, in addition to improving the SoA (II-A2) as to hardware-software figures of merit, will stimulate the attention from the industry for this class of approaches and methods, prompting a push for the inclusion of ML/DL-based functional safety into the future versions of the standards.

III. MATERIALS AND METHODS

A. TARGETED WOODWORKING MACHINE

The specific industrial woodworking machine used in this work is an SCM Morbidelli X200 [21], depicted in Fig. 1. This machine features two panels with a 3×2 and a 1×3 array of US sensors, as shown in Fig. 2. Fig. 3 schematizes the spatial configuration of the proposed proximity sensing system. This setup was used to collect data on the machine and test the accuracy and performance of the proposed solution. This setup is easily generalizable to machines and environments in different industrial sectors.

B. SYSTEM ARCHITECTURE

The hardware architecture of the proposed system is shown in Fig. 4 and relies on transducers that emit US pulses and sense the echo if a pulse hits an obstacle; if a detection happens, the system outputs a stop signal to the machine control. The main elements of the system are (i) the US sensors and their drivers, (ii) a Lattice FPGA for low-latency data collection, and (iii) a Nucleo-144 board mounting an STM32H743ZI MCU . This system performs both data collection and obstacle detection.

The data are acquired by a 2×3 plus 1×3 configuration of 9 Multicomp Pro MCUSD14A58S9RS-30C ultrasonic ceramic transducers.³ Using 9 sensors instead of just 1 is one of the key advances compared to [12]. Each transducer works both as an emitter and as a sensor for sound waves with a frequency between 30 kHz and 50 kHz; the sensing consists in emitting US pulses and receiving the echo reflected by obstacles. Each sensor is operated by a Texas Instruments PGA460, which integrates a low-noise amplifier, a programmable time-varying gain stage, a 12-bit ADC, and a DSP.⁴ The configured ADC resolution was set to 8 bits, producing `uint8` data, which is a convenient

format for a DNN quantized to 8 bits; the sampling frequency was set to 100kHz, and the sampling duration was set to 20.48ms-windows.

The low-power Lattice ECP5 LFE5U-85F FPGA⁵ collects the data from all 9 sensors. It communicates with the sensors via USART, and configures the resolution, sampling rate, and sampling duration at start-up. Then, the FPGA transmits the package of $2048 \text{ samples} \times 9 \text{ channels}$ 8-bit to the MCU via SPI. The motivation for using an FPGA for data aggregation is that the STM32H743ZI MCU does not have enough external interfaces; the FPGA allows to receive data from all 9 US sensors and convey them to the MCU through a single interface (i.e., the SPI).

The task of the MCU is to receive the data from the FPGA, run the DNN, and command the machinery to stop upon detection. The MCU is an STM32H743ZI,⁶ mounted on a STM32 Nucleo-144 board.⁷ This MCU mounts an ARM Cortex-M7⁸ processor with double-precision FPU operating at 480 MHz, 2 MB of Flash memory, 1 MB of SRAM (with 192 kB of tightly coupled scratchpad memory for real-time tasks), 4 DMA controllers, and peripherals such as UART/USART, SPI, Ethernet, and GPIO lines. Upon reception of the $2048 \text{ samples} \times 9 \text{ channels}$ data, the MCU executes the DNN inference. If the outcome is positive, the MCU raises a GPIO connected to the controller of the industrial machine , which halts the machine .

All the listed hardware elements are commercial components. The motivation for this choice is that the purpose of this work is not to profile specific hardware elements but to test whether the task is viable with commonly available hardware. In particular, there is no need for high-precision ultrasound sensors since accurate acoustic waveforms are irrelevant for a binary detection task in the presence of acoustic noise. In general, different component choices are not expected to alter the prototype's performance in terms of latency and accuracy. Profiling or designing dedicated components is out of the scope of this work. Different component choices to adapt the system to specific use cases do not limit the conclusions of the methodology proposed in this work.

In a more optimized iteration of the system, the FPGA+MCU assembly can be avoided by either (i) deploying the TCN model onto the FPGA, removing the MCU, or (ii) replacing the FPGA with one or more commercial off-the-shelf ICs (or by an FPGA chosen to be as small and inexpensive as possible) performing the data aggregation, keeping the net on an MCU. The latter option has the advantage of programmability for specific use cases with environmental conditions so diverse and challenging to require to adapt more than the net's parameters, e.g., the net's structure or additional processing stages. However, this kind of optimization is out of the scope of this work

⁵<https://www.latticesemi.com/Products/FPGAandCPLD/ECP5>

⁶<https://www.st.com/en/microcontrollers-microprocessors/stm32h743zi.html>

⁷<https://www.st.com/en/evaluation-tools/nucleo-h743zi.html>

⁸<https://developer.arm.com/Processors/Cortex-M7>

³<https://octopart.com/mcUSD14a58s9rs-30c-multicomp-30988352>

⁴<https://www.ti.com/product/PGA460>



FIGURE 1. The SCM Morbidelli X200 industrial woodworking machine used in this work (picture source: SCM Group [22]).



FIGURE 2. Configuration of the 9 US sensors mounted on the machine. The sensors are the grey metal round elements on the panels; the circular black pieces are washers for fastening the panels. Compare with Fig. 3.

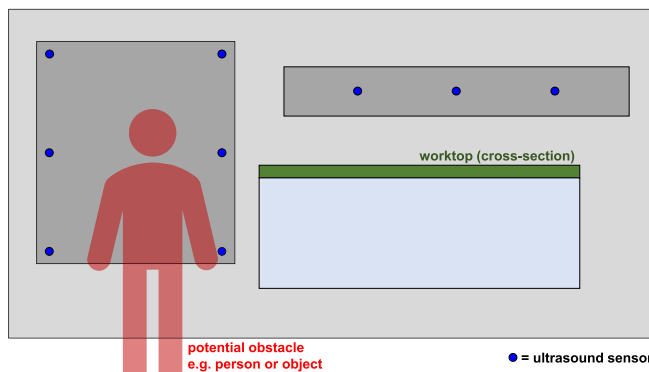


FIGURE 3. Spatial organization of the proposed proximity sensing system. The 3 US sensors on the moving cabinet over the worktop proved useful in preliminary tests to better sense the space surrounding the working table and obstacles at the far end of the working table. Compare with Fig. 2.

since the FPGA+MCU assembly has enough performance to make the realized prototype an effective proof-of-concept

at this applied research stage (as exposed in the results in Section IV).

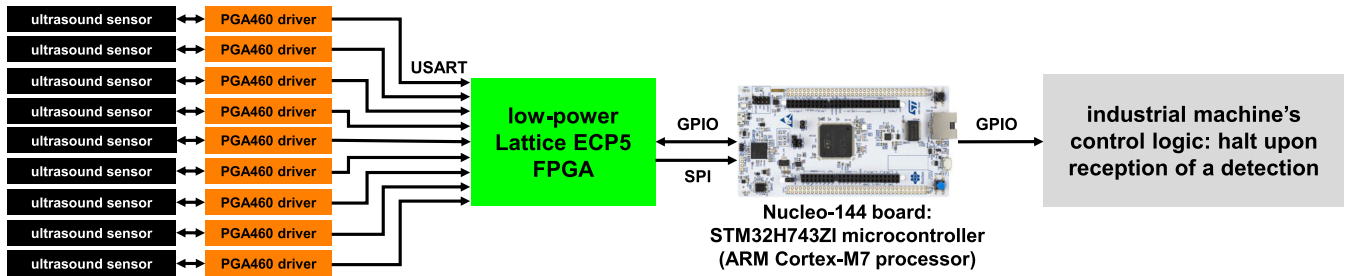


FIGURE 4. Schematic of the system architecture. Sensor fusion on data acquired from 9 sensors is one of the key proposed improvements compared to the SoA [12].

C. DATA ACQUISITION

The dataset acquisition followed three criteria: (i) framing the ML application as a detection task, i.e. a binary classification task *presence-vs-absence* of an obstacle; (ii) create environmental conditions analogous to the ones of the industrial facilities where the target woodworking machine typically operates; (iii) collect enough data to allow for a good DNN's recognition accuracy even on data pertaining to diverse conditions. Time windows of US signals were collected with and without obstacles creating a US response echo; the different used obstacles were people, dummies, and wood panels, also in a joint fashion. In addition to the two classes presence-vs-absence of an obstacle, two varying conditions produced more diverse data representative of real variable working situations:

- obstacle-sensor distance, varied from 0.5 m to 2.0 m;
- application of a compressed-air jet, recreating the environmental noise of the machinery's room, varying the pressure level from 0.0 bar (i.e., no noise) to 3.0 bar and the jet-sensor distance from 0.5 m to 1.5 m, both in presence and in absence of an obstacle.

First, 5 collections of data were acquired without noise, then 3 collections with noise. In noisy acquisitions, the compressed-air jet was always on, and the pressure value was constant while running each acquisition. Section IV-A reports the detailed structure of the signals and of the whole dataset.

D. INCREMENTAL LEARNING PROTOCOL

Incremental learning on the dataset involved experiments with incremental splits of the noisy data, i.e., collections 6-to-8. In particular, collections 6, 7, and 8 were merged and randomly split into three blocks of equal size with stratification (i.e., the same proportion of collections in each block). These blocks are denoted as the noisy data's *first third*, *second third*, and *last third*. The incremental experiments use the following splits:

- **Experiment 0:** training on collections $1 \cup 3 \cup 5$ and validation on collections $2 \cup 4$; this experiment involves no noisy data and is a control on the acquisition system and the quality of the data;
- **Experiment 1:** training on noiseless data, and validation on the last $\frac{1}{3}$ of noisy data; this experiment measures

how well a model can generalize to noisy data after only seeing noiseless data in training;

- **Experiment 2:** training on noiseless data plus the first third of noisy data, and validation on the last third of noisy data;
- **Experiment 3:** training on noiseless data plus the first and second thirds of noisy data, and validation on the last third of noisy data.

Experiments 1, 2, and 3 show the model progressively larger amounts of noisy data at training time; this allows assessing how much the proposed setup can benefit from incremental learning on newly-acquired data to improve detection. The validation set is the same across Experiments 1 to 3 for a fair comparison of the results.

This diverse dataset and its incremental protocol are a key advance compared to [12], where the incremental learning scenario is simulated by mere aggressive augmentation up to $1000\times$ of a single collection of 227 single-channel signal windows (i.e., $22\times$ fewer examples than the 5085 acquired in this work).

It is important to remark that incremental training is not meant to be run in real-time: real-time is only required for inference, which is part of the online pipeline of acquisition-transmission-processing. When the operators desire new data to improve the detection under specific challenging conditions, the system can collect new data and store them to a server (e.g., via the MCU's Ethernet), which retrains the net by including the new data and sends the updated model parameters back to the MCU. This process is not meant to be real-time because the new acquisition and the retraining typically need human supervision and iterations. Typically, the bottleneck is not transmission or latency but resides in (i) data acquisition, which requires materially preserving or reproducing the conditions of interest, and (ii) the search for the training settings able to fit both the old and the new data. This process occurs at the time scale of human manual experimentation, not at the time scale of the online acquisition-transmission-execution pipeline.

E. TEMPORAL CONVOLUTIONAL NETWORK: STRUCTURE, TRAINING, AND DEPLOYMENT

Temporal Convolutional Networks (TCNs) are a category of Convolutional Neural Networks (CNNs) specialized for time series. TCNs are based on 1D convolutions along the time

TABLE 2. Detailed structure of the proposed TCN, including the breakdown of all layers' memory footprint and computational load. All layers are sequential in a feed-forward fashion so that each layer's output format is the input format of the next one. As to sizes, the numbers of tensor elements directly correspond to the memory occupancy in bytes, thanks to 8-bit quantization. The field "# MAC" refers to the number of Multiply-and-Accumulate (MAC) operations.

LAYER	Size			# MAC	
	input	parameters (weights + biases)	output		
1	convolutional kernel $k = 3$ padding $p = 1$ stride $s = 2$	9×2048	112	4×1024	114688 (76.1 %)
2		4×1024	52	4×512	26624 (17.7 %)
3		4×512	26	2×256	6656 (4.4 %)
4		2×256	14	2×128	1792 (1.2 %)
5		1×128	7	1×64	448 (0.30%)
6		1×64	4	1×32	128 (0.08%)
(flattening)		1×32	0	32	0
7	fully connected	32	264	8	264 (0.18%)
8		8	72	8	72 (0.05%)
9		8	9	1	9 (0.01%)
COMBINED	Total parameters: 560 Max consecutive maps: $22.5 \cdot 10^3$ (layer 1)			Total MAC: $151 \cdot 10^3$	

dimension, and they outperform Recurrent Neural Networks on image segmentation, object detection, and biosignal processing [19], [23], [24], [25]. TCNs have also proven amenable to hardware-friendly parallelization strategies that reduce inference latency and energy consumption [26].

The proposed TCN has 6 convolutional layers followed by 3 linear layers, and Table 2 reports the net's complete structure. The input \mathbf{x} is a 2048 samples \times 9 channels `uint8` US signal window produced as per III-B and III-C. The 6 linear layers have 4, 4, 2, 2, 1, and 1 output channels, all with kernel size $k = 3$, full padding (i.e., zero-padding with length $p = 1$), and stride $s = 2$. The 3 linear layers have size 32-to-8, 8-to-8, and 8-to-1; the final scalar represents the input's score $\hat{y}_{\text{soft}} = \text{TCN}(\mathbf{x}) \in [0, 1]$, which is the soft (i.e., not yet binarized) assignment for the binary classification. All layers have batch-norm (BN) and ReLU activation except the last linear layer, which flows into a sigmoid. After training, BN folding is applied to merge each BN with its previous layer, slightly reducing the number of parameters and operations.

This TCN has just 560 parameters and requires just $151 \cdot 10^3$ Multiply-and-Accumulate (MAC) operations; The activation memory footprint is the maximum consecutive activation maps, i.e., input and output of a single layer; this is reached in the first convolutional layer with $22.5 \cdot 10^3$ activations (9×2048 input plus 4×1024 output. With 8-bit quantization (explained in the next paragraphs), the parameters and activations memory footprints amount to 560 B and 22.0 KiB, respectively.

This size makes the net very hardware-friendly for resource-constrained embedded platforms for computation and memory requirements. Moreover, it directly processes the raw signals without any handcrafted feature extraction or pre-processing, thanks to automatized feature learning at training time: this avoids time-consuming feature engineering and computation latency before inference.

Training consisted in 2 epochs in `float32`, followed by Post-Training Quantization (PTQ) to 8 bit and 16 epochs of Quantization-Aware Training (QAT). Quantization to 8-bit reduces the parameters memory requirement to 560 B and the activations memory requirement to 22.0 KiB, which are both $\frac{1}{4}$ of their `float32` counterparts. Both stages of training used balanced binary cross-entropy loss, Adam optimizer, initial learning rate 10^{-4} , and minibatch size 64. Both PTQ and QAT used the technique of Parameterized Clipping activation (PACT) [27].

Both trainings exploited the augmentation of the training set by a factor $64\times$, which consisted in producing 64 altered versions from each original US window by applying two transformations:

- a scaling by a factor from a uniform random distribution on $[0.95, 1.05]$, followed by casting back to `uint8`;
- a temporal shift by a random amount from a uniform distribution on $\{-25, \dots, +25\}$ samples.

This augmentation scheme is similar to [12]; still, the advances of this work allow to achieve accurate detection with $15\times$ milder augmentation (i.e., $64\times$ instead of $1000\times$), thanks to the inherent richness of the novel dataset (III-C).

In this setup, the sources of randomness are augmentation, net initialization, and stochastic minibatching. So, each of the Experiments 0, 1, 2, and 3 involved 64 repetitions to get statistics about the detection metrics.

The TCN was implemented using Python 3.8, PyTorch 1.9.0, and the open-source quantization library QuantLib.⁹ The TCN quantized to 8 bit was exported in ONNX format and deployed onto the STM32H743ZI MCU using the environment STM32CubeIDE 1.12.0 for code generation and exploiting X-CUBE-AI 8.0.0, a software extension for configuring DNN inference execution on STM32 MCUs using ARM CMSIS kernels. The stages in STM32CubeIDE

⁹<https://github.com/pulp-platform/quantlib>

or X-CUBE-AI did not include any further quantization or compression.

F. EVALUATION METRICS

This work targets both classification metrics, which measure the correctness of the TCN's detection, and deployment metrics, which quantify the computation and resource budget required by the TCN on the STM32H743ZI MCU.

The addressed classification metrics are the ones typical of detection (i.e., binary classification) on unbalanced data:

- *sensitivity* (synonym of *True Positive Rate* (TPR) or *recall*): the fraction of actual positives correctly detected:

$$\text{sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}}; \quad (1)$$

- *specificity* (synonym of *True Negative Rate* (TNR)): the fraction of actual negatives correctly classified:

$$\text{specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}}; \quad (2)$$

- *balanced accuracy* (synonym of *macro-average accuracy*): the average of sensitivity and specificity.
- Area Under the Receiver Operating Curve (AUROC).

All these metrics are independent of the class imbalance in the data, as opposed to naïve unbalanced accuracy. The pair sensitivity-specificity provides a more complete characterization than the pair precision-recall often used for binary classification since the latter pair does not consider the number of True Negatives; in contrast, the former pair considers all four possible outcomes. As to AUROC, it is independent of the threshold Θ used to determine the estimated hard labels $\hat{y}_{\text{hard}} \in \{0, 1\}$ from the TCN(\cdot) model's output soft labels $\hat{y}_{\text{soft}} \in [0, 1]$ for each input \mathbf{x} :

$$\hat{y}_{\text{hard}} = \begin{cases} 1 & \text{if } \hat{y}_{\text{soft}} = \text{TCN}(\mathbf{x}) > \Theta \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

Thus, the AUROC is methodologically interesting because it allows assessing the detection correctness independently from the specific sensitivity-specificity tradeoffs fixed in different application use cases. Since sensitivity and specificity depend on the choice of the discrimination threshold, the reported results refer to the threshold values tuned to maximize the balanced accuracy to report an example of tradeoff.

The addressed deployment metrics profile the workload of the real-time on-edge computation: memory footprint of the model; latency per inference; power consumption of inference, measured in working conditions $f_{\text{clock}} = 480$ MHz and $V_{\text{dd}} = 3.3$ V, via a USB power meter averaging over 30 s while executing inferences in loop; and energy per inference, determined as $\text{power} \times \text{latency}$.

IV. EXPERIMENTAL RESULTS

A. DATASET

Fig. 5 and Fig. 6 show the typical behaviour of the acquired signals. All windows begin with the final segment of the US

burst, which saturates the ADC's `uint8` dynamic range and carries no information about the class. However, this segment is a valuable control for diagnostics since it always presents the same timing across different recordings.

The initial saturation in each sensor's data only comes from the final segment of the US burst of that sensor. This check consisted in the following experiment. The procedure to check whether the burst from sensor i affected sensor $j \neq i$ involved running obstacle-less, noise-less runs starting sensor i 's acquisition 10 ms after sensor j 's acquisition. This means that sensor i 's burst emission happens between time 0 ms and time 10 ms of sensor j 's acquisition (which is 20.48 ms in total, as per III-B). So, if cross-sensor interference is present, it is visible in the first half window of sensor j 's data. Looping i and $j \neq i$ over all 9 sensors showed no cross-sensor interference for any (i, j) pair. This means that the adopted sensor placement causes no interference across sensors in the burst emission stage.

Later in the window, after the initial saturation due to the final segment of the emitted US burst, the echo carries the information of interest. As shown in Fig. 6, the noise resulting from a compressed air jet strongly affects the pattern of the echo signal envelopes. This makes it hard to devise handcrafted features that intuitively discriminate obstacle echoes from intensity due to noise, making classification hard, especially concerning specificity and the occurrence of false positives. This confirms the motivation for the recourse to DNNs (particularly TCNs) capable of automatic feature-learning at training time to obtain a data-driven feature extraction based solely on optimizing detection accuracy.

The realized dataset has the structure reported in Table 3. The whole dataset consists of 8 *collections*, and each collection is composed of 10 to 30 *runs*, for a total of 170 runs. The choice of the terms *collections* and *runs* is to avoid ambiguous naming such as *acquisitions*, *samples*, or *sessions*. Each collection corresponds to a value of the distance of the compressed-air jet, when applied; globally, the dataset contains 5 collections without noise and 3 collections with pressure noise. Within each collection, the different runs correspond to a choice of the obstacle-sensor distance and the jet's pressure and orientation (if applied). Between runs, the whole system was turned off and on. So, runs are homogeneous subsets of the dataset since they contain 2048-sample windows acquired in identical conditions of all settings, namely obstacle-sensor distance, compressed air jet pressure, and compressed air jet distance. Each collection consists of runs acquired with the same compressed air jet distance (if present), hence containing 2048-sample windows that are diverse due to varying obstacle-sensor distance and compressed air jet pressure (if present).

B. ACCURACY RESULTS

Fig. 7 and Table 4 report the detection results of the four experiments conducted as per Subsection III-D. Statistics are computed over the 64 repetitions of each experiment, performed to account for the fluctuations introduced by

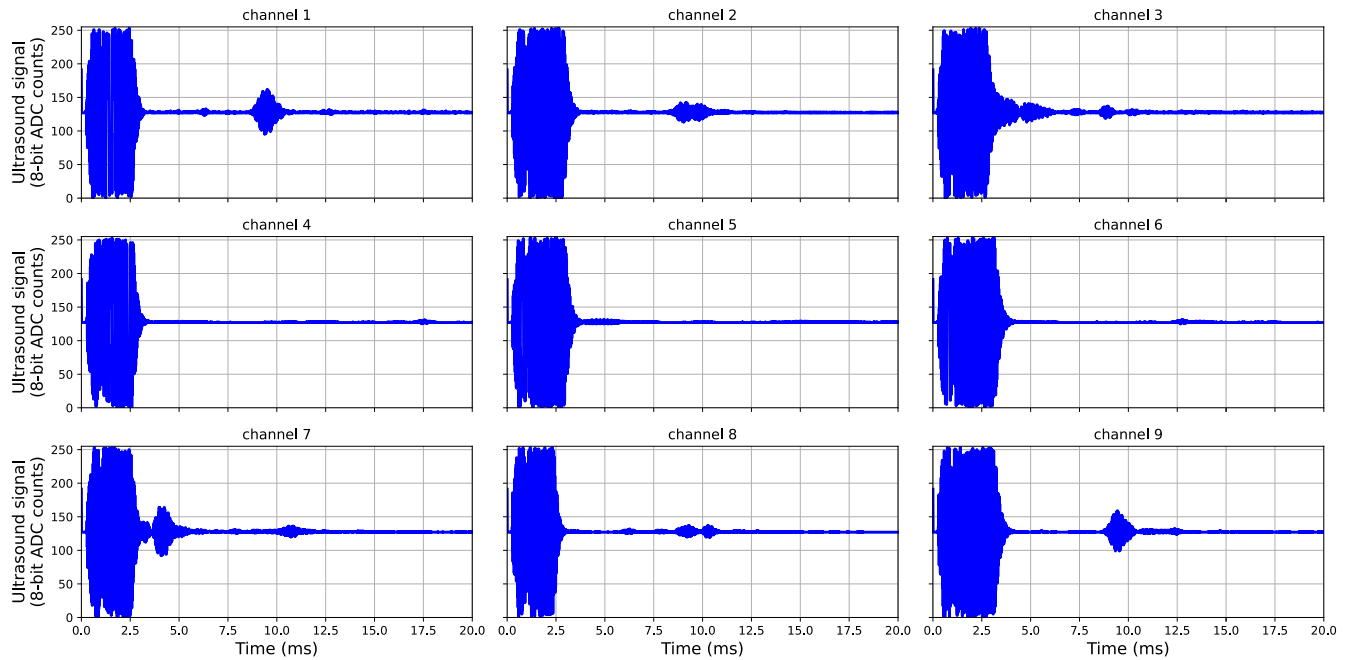


FIGURE 5. Example of a US window with obstacles and without noise (collection 1, run 10, window 1; all 9 channels, all samples except the last 48). It is possible to see the initial US burst, the subsequent silence, and the echoes received by the sensors facing obstacles.

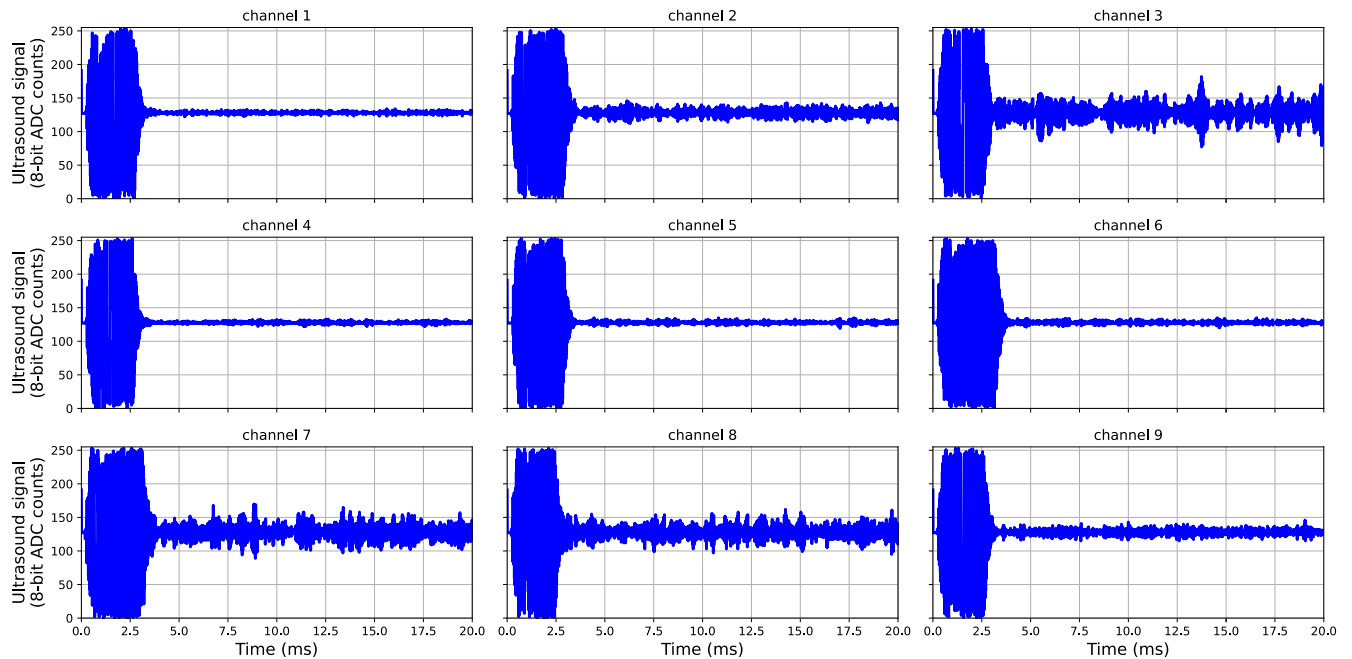


FIGURE 6. Example of a US window without obstacles and with noise from the compressed air jet (collection 8, run 50, window 1; all 9 channels, all samples except the last 48). The sensors most affected by noise sense an amplitude comparable to obstacles' echoes in the absence of noise (Fig. 5).

the sources of randomness in the process, namely data augmentation, initialization of the net's parameters, and minibatching for stochastic gradient descent, as explained in Subsection III-E. Fig. 7 shows that the experimental distributions obtained for the detection metrics are highly skewed, as can be seen from the asymmetric IQR ranges, whiskers, and outliers; therefore, median \pm Mean Absolute Deviation (MAD) is a convenient choice for summarizing each experiment in a way that is more robust and less sensitive

to skewness compared to average \pm standard deviation. The MAD is defined as

$$\text{MAD} \triangleq \text{median}_i (|a_i - \tilde{a}|) \quad (4)$$

where a_i 's are the accuracy values of a single repetition, and \tilde{a} is the experiment's median. It is worth remarking that, due to the non-linearity of the median, the median balanced accuracy is not the average of median sensitivity and median specificity, in general. The next subsections expose

the results of Experiments 0, 1, 2, and 3 (structured as per Subsection III-D), discussing each experiment individually.

1) EXPERIMENT 0

Experiment 0 is based on noiseless data for both training and validation (details in III-D). Therefore, this experiment is a check for the setup and the produced data. The outcome of this experiment is positive since all detection metrics (namely sensitivity, specificity, balanced accuracy, and AUROC – explained in Subsection III-F) have a median $> 97\%$. For instance, these results show a key successful sanity-check in that the working surface (Fig. 1, Fig. 2, and Fig. 3) is correctly discriminated from the added obstacles, despite being itself a physical object in the sensors' field of view.

2) EXPERIMENT 1

Experiment 1 consists in training on noiseless data and validation on noisy data (details in III-D). This experiment yields a balanced accuracy and an AUROC collapsed to values compatible with the chance level, which is $\frac{1}{2}$ for both these detection metrics. This collapse shows that recognition of noisy data is impossible if the model has never seen data affected by the compressed air jet pressure at training time; this confirms the motivation of the chosen protocol for data collection and incremental learning.

3) EXPERIMENT 2

Experiment 2 adds $\frac{1}{3}$ of the noisy data to the training set (details in III-D). The results of this first step of incremental learning on noisy data show that the detection in the presence of noise strongly surpasses the chance level, yielding a sensitivity of $85.0 \pm 5.5\%$ and all other metrics $> 88\%$. This experiment crucially proves that the data contain a pattern also in the presence of noise and that this pattern is strong enough to allow for an accurate data-driven detection.

4) EXPERIMENT 3

Experiment 3 adds a further $\frac{1}{3}$ of the noisy data to the training set (details in III-D). In this experiment, all the detection metrics except specificity further increase compared to Experiment 2. Specificity stays constant since it only decreases by 1.2%, and the new value is consistent with Experiment 2 within the variability $MAD = 2.8\%$. This experiment proves that the proposed system and DL setup are able to leverage increasing amounts of data to improve its accuracy on the challenging real working conditions of the industrial facility's environment.

Discussing detection metrics with an end-to-end view requires explaining what happens if the proposed system fails to detect an obstacle. In this case, a collision can happen between the obstacle and a machine's soft bumper; this kind of collision is not dangerous since bumpers are part of the active safety system that stops the machine in case of contact (as explained in Subsection II-A1). In general, it is possible to create even more redundancy by combining the proposed systems with any of the existing SoA active

safeguards illustrated in II-A1, such as pressure-sensitive floor mats, photocell barriers, or laser scanners.

C. PERFORMANCE AND MEMORY FOOTPRINT RESULTS

Table 5 reports the results of the TCN profiling, compared with [12]. For a fair comparison, since [12] dealt with just 1 input channel, that net is also extended to support 9 input channels as the new data. Memory footprints refer to TCN quantized to 8 bit. Memory footprints of activations are determined as the maximum sum of two consecutive feature maps since batch normalizations and ReLUs can be computed in place; for all models, the maximum-size pair is the input-output of the first convolutional layer, which occupies $(C_{in}T_{in} + C_{hid\ 1}T_{hid\ 1})$ bytes, where $T_{in} = T_{hid\ 1} = 2048$ samples, C_{in} is 1 channel for [12] and 9 channels for extended- [12] and the proposed new net; and $C_{hid\ 1}$ is 2 channels for [12] and extended- [12], and 4 channels for the proposed new net.

The energy consumption per inference was determined based on the power draw measured experimentally, which is $(1.63 \pm 0.01)W$, which is in the same range as the previous work [12]. Overall, the results show that the proposed new TCN improves all the deployment metrics, except the RAM used for activations, which is the same as the reference model, i.e., 22.0 KiB (2.1% of the total 1 MiB available on the STM32H743ZI MCU). The advantage of the new compact model lies in the latency and energy consumption per inference reduced by $> 2.27\times$ compared to [12].

It is essential to note that a latency of $\Delta t_{infer} = 5.0$ ms/inference does not imply a rate of $1/\Delta t_{infer} = 200$ inferences/s. In general, the entrance of operators or objects into the space spanned by the moving parts of the machine (corresponding to the field of view of the sensors) can be detected using an inference rate much lower than 200 inferences/s. For specific applications, the inference rate choice is based on the use case's requirements. Moreover, a higher inference rate gives some degrees of freedom for post-processing operations such as majority voting or averaging of the scores to make accuracy more robust. In situations that do not require a high inference rate, using an MCU with lower performance is possible, continuing to satisfy the real-time requirements.

It is worth discussing the latency results in more detail. The speed $v_{cabinet}$ of the machine's moving parts is of the order of 1 m/s (as explained in II-A1), and the speed v_{obs} of potential obstacles, i.e., people and objects in the surroundings, is typically lower. Assuming the worst case, i.e., the machine's moving cabinet and an obstacle moving towards each other from a distance d , the maximum allowed stopping time T_{max} is

$$T_{max} = \frac{d}{v_{cabinet} + v_{obs}}. \quad (5)$$

A conservative estimate of T_{max} , which corresponds to a conservative upper bound on latency, can be obtained assuming $v_{cabinet} = 1$ m/s, $v_{obs} = 1$ m/s, and $d = 0.5$ m,

TABLE 3. Dataset of ultrasound windows realized for setup validation and incremental learning. The dataset consists of *collections*; in turn, every collection contains *runs*. Each run contains data acquired with the same obstacle-sensors distance, and the air jet pressure (if present) and air jet distance. Each collection contains runs corresponding to different obstacle-sensors distances and air jet pressures, but the same air jet distance.

COLLECTION	Environment conditions	Number of runs	Windows per run	Total windows in collection	Negative class runs	Negative class windows	Positive class runs	Positive class windows	Class-imbalance of windows (neg – pos)
1	no noise	10	30 (15 in run 9)	285	2	45	8	240	15.8% – 84.2%
2		16	30	480	3	90	13	390	18.8% – 81.3%
3		30	30	900	10	300	20	600	33.3% – 66.7%
4		20	30	600	20	600	0	0	100.0% – 0.0%
5		24	30	720	24	720	0	0	100.0% – 0.0%
6	compressed-air jet noise at 1.0 – 1.5 m	20	30	600	0	0	20	600	0.0% – 100.0%
7		25	30	750	25	750	0	0	100.0% – 0.0%
8	compressed-air jet noise at 0.5 m	25	30	750	25	750	0	0	100.0% – 0.0%
All		170		5085	109	3255	66	1830	64.0% – 36.0%

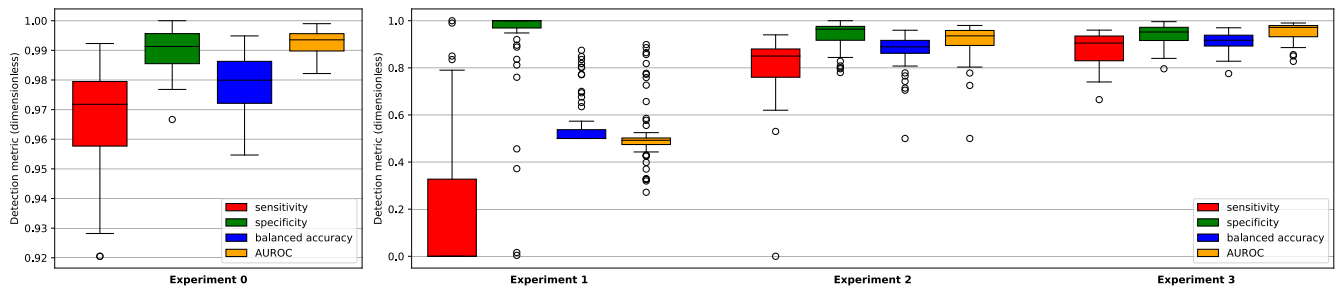


FIGURE 7. Experimental distributions of the detection metrics obtained for Experiment 0 (validation of setup and data) and Experiments 1-to-3 (incremental training on noisy data); for the details of the experimental protocol, see Subsection III-D. Notice the different y-scales in the two plots. The lower (resp., upper) whisker is set at the lowest datum above $Q_1 - 1.5$ IQR (resp., $Q_3 + 1.5$ IQR), with Q_1 and Q_3 the first and third quartiles respectively, and $IQR \triangleq Q_3 - Q_1$ the interquartile range. The general trend shows high accuracy in Experiment 0, the collapse in Experiment 2, and the incremental recovery in Experiments 2 and 3. Moreover, the asymmetric IQR ranges, whiskers, and outliers highlight high skewness; this motivates the recourse to median \pm Mean Absolute Deviation (MAD) for more robust summaries compared to average \pm standard deviation.

TABLE 4. Detection metrics results for Experiments 0, 1, 2, and 3 (protocol detailed in Subsection III-D). Distributions are summarized as median \pm Mean Absolute Deviation (MAD). This chart complements Fig. 7 by reporting quantitatively the high accuracy of Experiment 0, the collapse in Experiment 1, and the recovery in Experiments 2 and 3.

	Detection metric (median \pm MAD)			
	sensitivity	specificity	balanced accuracy	AUROC
Experiment 0	0.967 \pm 0.018	0.991 \pm 0.006	0.980 \pm 0.007	0.993 \pm 0.002
Experiment 1	0 \pm 0	1 \pm 0	0.5 \pm 0	0.493 \pm 0.016
Experiment 2	0.850 \pm 0.055	0.964 \pm 0.028	0.889 \pm 0.028	0.936 \pm 0.031
Experiment 3	0.905 \pm 0.030	0.952 \pm 0.028	0.917 \pm 0.024	0.972 \pm 0.013

TABLE 5. Results of the profiling of the proposed TCN's deployment and execution.

MODEL	Memory		Operations (kMAC)	Latency (ms)	Energy per inference (mJ)	
	weights (B)	activations (KiB)				
Conti <i>et al.</i> [12]	original: 1 input channel	1044	6.0	227	11.4	18.6
	expanded to 9 input channels	1792	22.0	630	> 11.4	> 18.6
This work	560 (0.32 \times)	22.0 (same)	151 (0.24 \times)	5.0 (< 0.44 \times)	8.2 \pm 0.2 (< 0.44 \times)	

which is the shortest distance used in our dataset (III-C). These values yield

$$T_{\max} = \frac{0.5 \text{ m}}{1.0 \text{ m/s} + 1.0 \text{ m/s}} = 0.25 \text{ s.} \quad (6)$$

This underestimate of the maximum allowed latency is 12 \times the acquisition time of the signal, i.e., 20.48 ms (III-B), and 50 \times the computation latency of the TCN inference, i.e., 5.0 ms. The US sensors can detect obstacles at a maximum distance of 2 m to 2.5 m, so more time is generally available.

Even in the worst case, the proposed system's latency for data acquisition and processing is one order of magnitude shorter than the available time: the proposed solution has a latency sufficiently short for the task, with a significant margin for future scenarios with faster-moving cabinets and obstacles.

V. CONCLUSION

This work proposes a solution for collision avoidance applied to the use case of a real woodworking industrial machine. This work also publicly releases a novel curated sensor-fusion dataset for US-based proximity sensing of the machine surroundings and implements a TCN setup. The proposed TCN is able to increase its detection accuracy by exploiting new data, thus proving able to tackle real industrial environments that are challenging due to noise. At the same time, the proposed TCN has a complexity low enough to fit embedded platforms' memory, latency, and energy constraints, as proven by deploying it onto an edge MCU. The proposed solution can be easily applied to different premises and machines; in particular, it is more hardware-friendly than the available SoA models, and its limited hardware requirements allow the setup to scale up to monitor larger environments.

ACKNOWLEDGMENT

The authors GIACOMO MICHELE PUGLIA and FELICE TECCE were with DPControl S.r.l., Nocera Inferiore, Italy.

REFERENCES

- [1] International Electrotechnical Commission (IEC), *Functional Safety of Electrical/Electronic/Programmable Electronic Safety-Related Systems—Part 1: General Requirements*, document IEC 61508-1:2010, ed. 2.0, 2010. [Online]. Available: <https://webstore.iec.ch/publication/5515>
- [2] A. Shrestha and A. Mahmood, "Review of deep learning algorithms and architectures," *IEEE Access*, vol. 7, pp. 53040–53065, 2019.
- [3] K. Cao, Y. Liu, G. Meng, and Q. Sun, "An overview on edge computing research," *IEEE Access*, vol. 8, pp. 85714–85728, 2020.
- [4] F. Wang, M. Zhang, X. Wang, X. Ma, and J. Liu, "Deep learning for edge computing applications: A state-of-the-art survey," *IEEE Access*, vol. 8, pp. 58322–58336, 2020.
- [5] D. L. Dutta and S. Bharali, "TinyML meets IoT: A comprehensive survey," *Internet Things*, vol. 16, Dec. 2021, Art. no. 100461.
- [6] P. P. Ray, "A review on TinyML: State-of-the-art and prospects," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 34, no. 4, pp. 1595–1623, Apr. 2022.
- [7] Y. Abadade, A. Temouden, H. Bamoumen, N. Benamar, Y. Chtouki, and A. S. Hafid, "A comprehensive survey on TinyML," *IEEE Access*, vol. 11, pp. 96892–96922, 2023.
- [8] A. Burrello, M. Scherer, M. Zanghieri, F. Conti, and L. Benini, "A microcontroller is all you need: Enabling transformer execution on low-power IoT endnodes," in *Proc. IEEE Int. Conf. Omni-Layer Intell. Syst. (COINS)*, Aug. 2021, pp. 1–6.
- [9] R. S. Peres, X. Jia, J. Lee, K. Sun, A. W. Colombo, and J. Barata, "Industrial artificial intelligence in Industry 4.0—Systematic review, challenges and outlook," *IEEE Access*, vol. 8, pp. 220121–220139, 2020.
- [10] *Artificial Intelligence for Digitizing Industry*, May 2019.
- [11] International Electrotechnical Commission (IEC), *Safety of Machinery Electro-Sensitive Protective Equipment—Part 1: General Requirements and Tests*, document IEC 61496-1:2020, ed. 4.0, 2020. [Online]. Available: <https://webstore.iec.ch/publication/63115>
- [12] F. Conti, F. Indirli, A. Latella, F. Papariello, G. M. Puglia, F. Tecce, G. Urilini, and M. Zanghieri, *AI-Powered Collision Avoidance Safety System for Industrial Woodworking Machinery*. Copenhagen, Denmark: River Publishers, 2021.
- [13] SCM Group. (2022). *Morbideilli M100/M200. Numerical-Controlled Machining Centres*. [Online]. Available: <https://www.scmgroup.com/en/scmwood/products/machining-centres.c874/cnc-machining-centres-for-routing-and-drilling.878/morbideilli-m100-200.32314>
- [14] HOMAG Group. (2023). *CNC Processing Center CENTATEQ P-210*. [Online]. Available: <https://www.homag.com/en/product-detail/cnc-processing-center-centateq-p-210>
- [15] Biesse Group. (2023). *Rover K FT. CNC Machining Centre*. [Online]. Available: <https://www.biesse.com/ww/wood/cnc-work-centres/rover-k-ft>
- [16] Biesse Group. (2023). *Rover K Smart. CNC Machining Centre*. [Online]. Available: <https://www.biesse.com/ww/wood/cnc-work-centres/rover-k-smart>
- [17] Biesse Group. (2023). *Rover A 12/15/18. CNC Machining Centre*. [Online]. Available: <https://www.biesse.com/ww/wood/cnc-work-centres/rover-a-1215>
- [18] M. Zanghieri, S. Benatti, A. Burrello, V. Kartsch, F. Conti, and L. Benini, "Robust real-time embedded EMG recognition framework using temporal convolutional networks on a multicore IoT processor," *IEEE Trans. Biomed. Circuits Syst.*, vol. 14, no. 2, pp. 244–256, Apr. 2020.
- [19] M. Zanghieri, S. Benatti, A. Burrello, V. J. K. Morinigo, R. Meattini, G. Palli, C. Melchiorri, and L. Benini, "SEMG-based regression of hand kinematics with temporal convolutional networks on a low-power edge microcontroller," in *Proc. IEEE Int. Conf. Omni-Layer Intell. Syst. (COINS)*, Aug. 2021, pp. 1–6.
- [20] International Electrotechnical Commission (IEC). *Safety of Machinery—Application of Protective Equipment to Detect the Presence of Persons*, document IEC 62046:2018, ed. 1.0, 2018. [Online]. Available: <https://webstore.iec.ch/publication/27263>
- [21] SCM Group. (2019). *Morbideilli X200/X400. CNC Nesting Machining Centres for Drilling and Routing*. [Online]. Available: https://www.scmgroup.com/products/docs/CDL/Morbideillix200-x400rev00mag19_Ing.pdf
- [22] Höchsmann. *SCM Morbidelli X200*. [Online]. Available: <https://wtp.hoechsmann.com/it/lexikon/40505/morbideillix200>
- [23] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager, "Temporal convolutional networks for action segmentation and detection," *CoRR*, vol. abs/1611.05267, Dec. 2016.
- [24] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *CoRR*, vol. abs/1803.01271, Sep. 2018.
- [25] M. Zanghieri, A. Burrello, S. Benatti, K. Schindler, and L. Benini, "Low-latency detection of epileptic seizures from iEEG with temporal convolutional networks on a low-power parallel MCU," in *Proc. IEEE Sensors Appl. Symp. (SAS)*, Burrel, Benin, Aug. 2021, pp. 1–6.
- [26] A. Burrello, A. Dequino, D. J. Pagliari, F. Conti, M. Zanghieri, E. Macii, L. Benini, and M. Poncino, "TCN mapping optimization for ultra-low power time-series edge inference," in *Proc. IEEE/ACM Int. Symp. Low Power Electron. Design (ISLPED)*, Jul. 2021, pp. 1–6.
- [27] J. Choi, Z. Wang, S. Venkataramani, P. I. Chuang, V. Srinivasan, and K. Gopalakrishnan, "PACT: Parameterized clipping activation for quantized neural networks," *CoRR*, vol. abs/1805.06085, Apr. 2018.



MARCELLO ZANGHIERI (Graduate Student Member, IEEE) received the M.Sc. degree in physics from the University of Bologna, Italy, in 2019, where he is currently pursuing the Ph.D. degree in data science and computation with the Energy-Efficient Embedded Systems Laboratory (EES Lab), DEI Department, under the supervision of Prof. Luca Benini. In 2019, he held a Research Scholarship with EES Lab, with a focus on research studentship. His research interests include time series analysis with machine learning and deep learning, focusing on ultrasounds, EMG, and EEG, to advance human-machine interaction based on low-power computing platforms.



and compilers for neural

FABRIZIO INDIRLI (Graduate Student Member, IEEE) received the M.Sc. degree in computer science and engineering from Politecnico di Milano, Italy, in 2019, where he is currently pursuing the Ph.D. degree in information technology. Since 2020, he has been collaborating with STMicroelectronics, where he was a Software Design Engineer. He has coauthored several scientific articles for both industrial and academic projects. His research interests include tiny machine learning



the design and verification of hardware platforms for video surveillance and HW-accelerated neural network inference. He coauthored several scientific articles and filed three patents. He contributed and is contributing to several EU-funded projects.

FRANCESCO PAPARIELLO received the M.Sc. degree (summa cum laude) in computer science engineering from the University of Pisa, Italy, in 2000. Since 2000, he has been with STMicroelectronics in different countries (France, Switzerland, and Italy). He has been responsible for the design and development of a retargetable multi-core simulation infrastructure. He is currently a Senior Member of the Technical Staff with STMicroelectronics. He also contributed to



ANTONIO LATELLA received the M.Sc. degree in computer and automation engineering from Marche Polytechnic University, Ancona, Italy, in 2018. Since 2018, he has been with SCM Group S.p.a., Rimini, Italy. In 2020, he was involved in European projects in the research and development field. He is currently involved in the development of digital and industrial IoT services related to machines in the after-sales and assistance fields.



GIULIO URLINI received the M.Sc. degree in electronic engineering from the University of Brescia, Italy, in 1999. Since 2000, he has been with STMicroelectronics, Italy, where he is currently a Senior Program Manager.



vehicle charging stations.

GIACOMO MICHELE PUGLIA received the M.Sc. degree in computer science from the University of Trento, Italy, in 2014. Since 2014, he has been a Firmware-Software Developer in various capacities. From 2019 to 2021, he was with DPControl S.r.l., Italy, where he dedicated his efforts to projects centered around image processing and neural networks. Since 2022, he has been employed with Bitron, Italy, as a Firmware and Software Developer for electric



Mac Van Valkenburg Award, the 2020 EDAA Achievement Award, the 2020 ACM/IEEE A. Richard Newton Award, and the 2023 IEEE CS E. J. McCluskey Award.

LUCA BENINI (Fellow, IEEE) received the Ph.D. degree from Stanford University. He is currently a Full Professor with Università di Bologna, Italy. He holds the Chair of Digital Circuits and Systems with ETH Zürich, Switzerland. His research interests include energy-efficient parallel computing systems, smart sensing micro-systems, and machine learning hardware. He is a fellow of ACM and a member of the Academia Europaea. He was a recipient of the 2016 IEEE CAS



sectors, such as power management and biomedical. He is currently involved in projects aimed with acquiring and processing video streams and sensor interfacing.

FELICE TECCE received the degree in electronic engineering from the University of Naples Federico II, in 2019. In the same year, he specialized in digital electronics and began his working career in the same year with DPControl S.r.l., as a Digital Designer on FPGA devices. Since 2022, he has been with the Analog MEMs and Sensors (AMS) Research and Development Group, STMicroelectronics (ST), Italy, as a Front-End ASIC Designer, taking part in projects related to different operating



with a particular focus on system-on-chips for artificial intelligence applications. His research work has resulted in more than 70 publications in international conferences and journals and was awarded several times, including the 2020 IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I: REGULAR PAPERS Darlington Best Paper Award.

FRANCESCO CONTI (Member, IEEE) received the Ph.D. degree in electronic engineering from the University of Bologna, Italy, in 2016. From 2016 to 2020, he held a research grant with the University of Bologna and was a Postdoctoral Researcher with ETH Zürich. He is currently a tenure-track Assistant Professor with DEI Department, University of Bologna. His research is centered on hardware acceleration in ultra-low power and highly energy-efficient platforms,

...

Open Access funding provided by 'Alma Mater Studiorum - Università di Bologna' within the CRUI CARE Agreement