Real-Time EEG-Based Cognitive Workload Monitoring on Wearable Devices

Renato Zanetti [iD] , *Student Member, IEEE,* Adriana Arza [iD] , *Member, IEEE,* Amir Aminifar [iD] , *Member, IEEE,* and David Atienza [iD] , *Fellow, IEEE*

Abstract—**Objective: Cognitive workload monitoring (CWM) can enhance human-machine interaction by supporting task execution assistance considering the operator's cognitive state. Therefore, we propose a machine learning design methodology and a data processing strategy to enable CWM on resource-constrained wearable devices. Methods: Our CWM solution is built upon edge computing on a simple wearable system, with only four peripheral channels of electroencephalography (EEG). We assess our solution on experimental data from 24 volunteers. Moreover, to overcome the system's memory constraints, we adopt an optimization strategy for model size reduction and a multi-batch data processing scheme for optimizing RAM memory footprint. Finally, we implement our data processing strategy on a state-of-the-art wearable platform and assess its execution and system battery life. Results: We achieve an accuracy of 74.5% and a 74.0% geometric mean between sensitivity and specificity for CWM classification on unseen data. Besides, the proposed model optimization strategy generates a 27.5x smaller model compared to the one generated with default parameters, and the multi-batch data processing scheme reduces RAM memory footprint by 14x compared to a single batch data processing. Finally, our algorithm uses only 1.28% of the available processing time, thus, allowing our system to achieve 28.5 hours of battery life. Conclusion: We provide a reliable and optimized CWM solution using wearable devices, enabling more than a day of operation on a single battery charge. Significance: The proposed methodology enables real-time data processing on resource-constrained devices and supports real-time wearable monitoring based on EEG for applications as CWM in human-machine interaction.**

Index Terms—**Cognitive Workload Monitoring, Human-Machine Interaction, EEG, Wearable Devices, Edge Computing.**

I. INTRODUCTION

WHILE the level of automation in work environments is likely to rise in the years to come [1], human operators are still expected to play an essential role in system supervision (monitoring, diagnosis, and planning) [2]. In fact, automation has fundamentally changed human-machine interaction (HMI) in system operation by driving human operators from labour-intensive activities to mentally-intensive supervision tasks [2], [3]. Furthermore, despite reducing the amount of manual work, automated system supervision requires continuous levels of vigilance and prompt responses, thus it can be even more burdensome than previous tasks [2]. As a result, highly automated environments still suffer from performance fluctuations and human errors due to limited humans cognitive resources [3]–[5].

Furthermore, human errors are the foremost source of operation failures in highly automated and safety-critical environments [6]. Such failures can cause serious accidents, for instance, in rail routing and aerial traffic control [6]. Hence, beyond the need for solutions to maintain human performance, there is also an urge for improving safety in highly automated task execution. In this context, the assessment of the operator's cognitive workload (CW) in real-time has been already proposed for automation level regulation [5], [7], [8]. More specifically, CW can be utilized in HMI to increase the awareness of the operator's cognitive state, thereby enabling the regulation of operation demands automatically with different levels of the system's task execution assistance [8]–[10].

Although CW assessment is a challenging endeavour encompassing various factors (e.g., mental and physical demands, performance metrics, and frustration level) [7], it has been widely studied as a tool to assess operators' performance while executing tasks in diverse contexts [6]–[16]. Among CW assessment metrics, prior works considered physiological signals suitable for CW monitoring (CWM), regarding the possibility of continuous and unobtrusive data acquisition [5], [6], [17]. In particular, electroencephalography (EEG) has been successfully used to correlate brain activity variations to task demand levels [5], [6], [9], [11], [12].

However, traditional commercial EEG data acquisition systems are considered cumbersome for daily monitoring [18], and their use might be associated with social stigma [6], [19]. Even though there are commercial wireless EEG headsets on the market, using such systems for CWM would require EEG streaming to another platform for data processing. Streaming data reduces the system's reliability and requires more battery capacity to account for the power-hungry wireless link [20]. To overcome these limitations, we consider a CWM system that embeds in the same device both data acquisition and processing, thus guaranteeing a reliable CW assessment in real-

time and long-term operation (for at least 8 h/day). Moreover, such a system also has to be unobtrusive and easy to set up, minimizing interference with the task execution.

The latest smart wearable devices, employing the edge computing paradigm, can perform continuous data acquisition and on-board machine learning (ML) to achieve high classification rates [21]–[24]. Hence, we envision a wearable-based solution to address the previously stated requirements for CWM. Nevertheless, most wearable systems are based on resource-constrained platforms, presenting very limited processing capacity, memory, battery, and data transmission throughput [25]. Therefore, a wearable CWM system also requires a tailored data processing algorithm to reduce workload and RAM memory usage. Particularly in CW assessment, the algorithm RAM memory requirement is critical as it is necessary to process long time-series (various seconds of physiological data) [15], [16], [26].

In order to employ a resource-constrained wearable platform for CWM, in this work, we propose a machine learning (ML) design methodology and data processing strategy based on only four channels of EEG, designed to enable the use of a simple acquisition system to assess the operator's cognitive state. Particularly, we adopt the edge-computing paradigm, proposing a lightweight data processing scheme to tackle memory and battery life constraints [27]. The main contributions of this paper are:

- A solution for real-time CWM on wearable devices composed of an ML design methodology and a data processing strategy, both validated on an experimental data set. We reach an accuracy of 74.5% and a 74.0% geometric mean of sensitivity and specificity on unseen data.
- A feature extraction scheme for processing data of long time-series in multiple batches. The proposed scheme reduces the RAM memory footprint of our data processing strategy by 14x compared to a single batch processing, hence allowing its deployment in a resource-constrained embedded platform.
- A model size optimization strategy for reducing flash memory requirement in Random Forest (RF) low-level representation. By reducing our final model size by 27.5x, the proposed strategy enables the use of our RF model in flash-limited embedded wearable platforms.
- An implementation of the proposed data processing strategy for real-time CWM on a state-of-the-art wearable platform, which reaches 28.5 hours of operation on a single battery charge (225 $mA \cdot h$ battery).

The rest of the paper is organized as follows. Section II presents a state-of-the-art overview on CW assessment, mainly focusing on physiological-signal based monitoring and wearable systems for CWM. In Section III, we present our proposed methodology for designing and validating our CWM solution for wearable devices. Section IV provides more details regarding adopted machine-learning methodology for data processing design and validation. Section V presents our proposed system implementation for real-time CWM. Section VI details the experimental setup for evaluating the proposed methodology,

and Section VII presents the results and discussion. Section VIII concludes this work.

II. RELATED WORK

CW is a complex concept with various definitions in the literature, yet most of them relate cognitive engagement or capacity to task demands [6]. The CW concept is widely associated with human performance on task execution. Moreover, prior works advocated that CW yield information on the operator's cognitive state to regulate the level of task execution assistance on automation [5], [33]. Hence, the use of CW has the potential to enhance HMI, thus improving performance. However, continuous CWM requires unobtrusive assessment metrics and systems. Henceforth, we present prior methodologies for CW assessment. Moreover, we focus on previous applications targeting long-term and real-time subject monitoring based on physiological signals.

A. CW Assessment Measures

Prior research on CW assessment relies mainly on three types of measures: surveys [34], performance metrics [7], and physiological signals [5]. Among them, the application of surveys is the simplest solution, but it requires the interruption of the task for the subject to report on their cognitive condition. Performance metrics can be used to overcome this problem, assessing CW based on performance scores during task execution. However, such metrics are system dependent, thus not always available in real-world applications. On the other hand, physiological signals are considered a more comprehensive metric for CW assessment, relating responses from the central nervous system (CNS) and autonomic nervous system (ANS) to cognitive demand variations [5], [6]. Moreover, physiological signals' reliability for assessing CW has been validated in various settings, leveraging unobtrusive and continuous data acquisition during experimental tasks [5]–[13], [15], [16], [26].

In this paper, we adopt physiological signals to foster CWM use in HMI as we consider it the only suitable alternative for real-time assessment daily. In this direction, we assess prior works aiming at CWM based on physiological signals and machine learning. Table I presents works with high CW classification performance but also relates other applications on mental state monitoring with data processing on the edge.

Particularly, following a multimodal CWM approach, [11], [15], [16], and [26] employ various signals to monitor CW, achieving classification accuracy from 80.20% to 82% on unseen test sets and up to 93.61% on cross-validation. The wide range of classification performance is influenced by the considerable intra- and inter-subject response variability when coping with task demands [35] and is also related to the diversity of experimental setups available in the literature. Furthermore, the findings of [11] and [26] indicate that EEG features contributed the most to different levels of CW discrimination. Although [15] and [16] reach relatively high accuracies using only peripheral signals, such an approach requires the subject to wear various acquisition systems and sensors. Thus, it reduces its likelihood of adoption for daily use. Additionally, a wearable multimodal CWM approach would also require

Table I
STATE-OF-THE-ART (SoA): SELECTED WORKS ON MENTAL-STATE MONITORING APPLICATIONS USING PHYSIOLOGICAL
SIGNALS AND OTHER WEARABLE AND EDGE COMPUTING-BASED APPLICATIONS.

Work	Target Application	Wearables (MCU)	Signals	Win. size	Methods Applied	Detection Performance	System
[28]	Drowsiness	No	EEG, EOG	8 s	ELM, SVM (2 class)	Acc: 94.70%	Nihon-Koden EEG 2110
[23]	Stress	Yes	EEG, ECG, EDA	-	15 features; SVM (2 class)	Gmean: 86.94%	MindWave, Bio-Harness, Shimmer
[29]	Stress	Yes	ECG, RSP, SpO2, Accelerometer	35 s	16 features; SVM (2 class)	Acc: 96.7%	In-house ASIC
[30]	Stress	Yes	PPG, EDA	20 s	30 features; KNN (3 class)	Acc: 75%	In-house device
[31]	Seizure Detection	Yes Cortex-M4	EEG	4 s	54 features; RF (2 class)	Gmean: 94.50%	e-Glass
[32]	Gesture Recognition	Yes PULP SoC	EMG	2 s	8 features; HD computing (11 class)	Acc: 84.30%	BioWolf
[11]	CW	No	EOG, ECG, RSP, EEG (6 channels)	10 s	43 features; ANN (3 class)	Acc: 82%	Grass P511
[26]	CW	No	EOG, ECG, EEG (16 channels)	30 s	82 features; SVM (3 class)	Acc: 93.61%*	Nihon-Koden EEG
[12]	CW	No	EEG (14 channels)	4 s	4 features; SVR (3 class)	Acc: 69.0%	Emotiv Epoc
[10]	CW	No	EEG (25 channels)	2 s	26 features; LDA (2 class)	Acc: 85%	Biosemi ActiveTwo
[15]	CW	Yes	PPG, RSP, SKT, ECG	60 s	24 features; XG-Boost (2 class)	Acc: 80.20%	Biopack
[16]	CW	Yes Cortex-M3	SKT, ECG, EDA, PPG, RSP	60 s	13 features; RF (3 class)	Gmean: 81.70%	Shimmer Empatica E4
This work	CW	Yes Cortex-M4	EEG (4 channels)	56 s	17 features; RF (2 class)	Gmean: 74.0.%, Acc: 74.5%	e-Glass

EEG: electroencephalography; EOG: electrooculography; ECG: electrocardiography; RSP: respiration; EDA: electrodermal activity;
EMG: electromyography; PPG: photoplethysmogram; SKT: skin temperature; SpO2: oxygen saturation; Acc: accuracy;
Gmean: geometrical mean of sensitivity (Sen) and specificity (Spec);   * Reported on cross-validation.

more circuitry and higher algorithmic complexity, which poses higher energy requirements for data processing. Therefore, we chose to employ only EEG on our proposed CWM solution.

Prior works have found correlations between certain EEG band power oscillations and task complexity. For instance, the authors of [11] observed that theta (4 - 8 Hz) band activity from central electrodes on the scalp were the features that contribute the most to CW assessment. Moreover, theta band power is found to increase from low to high task demand variation, especially on the frontal cortex [5], [9], [33]. Meanwhile, the alpha (8 - 12 Hz) band power has shown an inverse correlation with task complexity [5], [9]. Finally, the beta (12 - 30 Hz) spectral band has also been used by various works to assess CW [5], [6]. Besides EEG band powers, entropies and statistical features have been also used for CWM [36], [37].

Most of the previous works employing EEG have made use of caps or headsets covering the midline scalp area. However, both EEG caps and headsets are considered cumbersome in daily use [18], [19]. On the other hand, [11] and [12] also reported salient features over the side frontal and temporal areas when classifying different levels of CW. Thus, there are opportunities for new wearable solutions exploring peripheral-scalp electrode positioning and also targeting the lack of light and easily deployable CWM devices.

B. Wearable Systems for CWM on the edge

Uncomfortable acquisition equipment, long subject preparation time, unreliable CW assessment, and short battery life are examples of hurdles preventing CW assessment in real-time for daily use. Consequently, these are problems we have to solve to provide operators' mental state awareness on HMI in highly automated systems. Additionally, as stated before, solving such problems entails the use of tailored wearable devices. Nonetheless, we found no complete wearable solution in the literature for CWM.

Indeed, we found a few recent works targeting CWM based on wearable and multimodal sensors [15], [16]. The authors of [15] proposed an ensemble-based machine learning technique (i.e., XGBoost) to assess subject CW, reaching 80.20% accuracy in differentiating between two levels of cognitive demands. However, even though targeting a CWM system, they do not implement their methodology on a wearable platform, relying on a commercial wireless system for data acquisition. On the other hand, in [16], the authors propose a self-aware machine learning algorithm for CWM during manual labour for wearable devices. It includes the estimation of the energy consumption for executing the feature extraction and inference stages of the proposed algorithm on an ARM Cortex-M3 development board. Nevertheless, they do not implement the algorithm in a specifically designed wearable device suiting their application needs. Hence, [16] does not consider the entire system design.

For instance, we can rely on inconspicuous system design to obtain a hidden wearable platform capable of performing unobtrusive data acquisition while the operator executes their daily tasks. Moreover, a simple wearable platform with few monitored physiological signals (i.e., EEG) can decrease subject
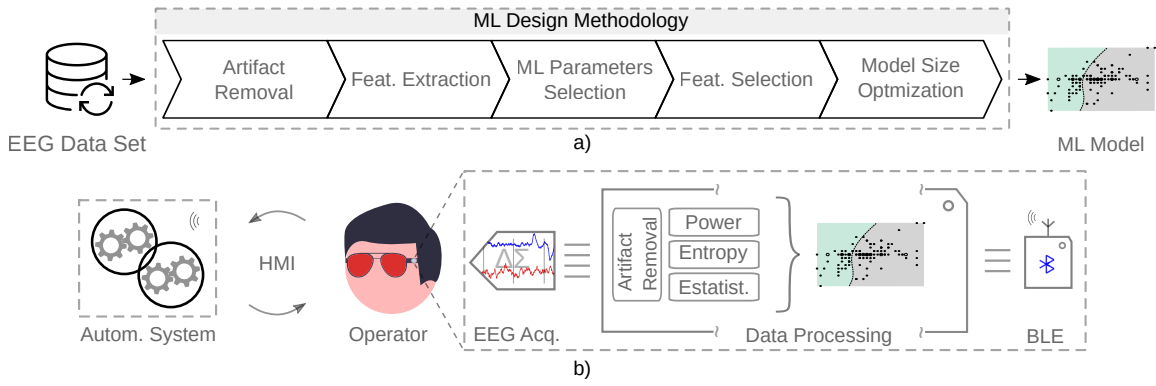
Fig. 1. a) ML-based design methodology for CWM on wearable devices; b) real-time CWM for HMI solutions.

preparation time and use less circuitry to reduce energy consumption. Additionally, by using the edge-computing paradigm to process the data on-board, we can assess CW at the data producer's site [27]. Hence, we reduce the latency and the device's energy consumption [38] by avoiding streaming data to another platform with more computation power. In fact, we found prior works targeting other applications such as epileptic seizure detection [31], [39], and gesture recognition [32], which reported battery life from 1.22 to 2.71 days of operation on a single charge. Finally, the use of EEG for CWM depends on new electrode placement strategies to avoid unwanted bulky design (e.g., the design proposed in [39]).

Nevertheless, the use of a new electrode placement might require further investigation into its viability for CW assessment. First, midline scalp positions were found to display the highest correlations with task demand variations among investigated electrodes. Second, the amount of RAM memory required to assess CW on wearable platforms is also a hard restriction to be addressed by the application design. For instance, there is no clear definition among previous CW assessment works regarding the required data window size (e.g., 10 s [11], 30 s [26], and 60 s [15]). To overcome these problems, we explore a design methodology based on machine learning to devise a lightweight data processing strategy. For instance, we divide the original data segment into smaller data batches to execute data filtering and feature extraction (cf., Subsection V-C).

III. COGNITIVE-WORKLOAD MONITORING AT THE EDGE

We aim to provide awareness of operators' cognitive state during HMI by monitoring CW in real-time with wearable devices. Moreover, we target a seamless integration of CWM and HMI, proposing an unobtrusive and easy-to-set-up solution to reduce interference with daily supervision-task execution. Hence, we present a solution composed of an ML design methodology for CW assessment based on only four EEG channels and a system implementation including a data processing strategy for CWM on a state-of-the-art wearable platform (Fig. 1). However, the use of wearable platforms poses a major challenge in obtaining a reliable CW assessment. The reduced set of electrodes available and the minimal data processing capacity limit the use of well-established artifact-removal methods in real-time, like those based on blind source separation [18].

Therefore, we propose a lightweight CWM solution to overcome such challenges and validate its viability experimentally. This solution is composed of an ML design methodology and a data processing strategy to reduce memory consumption while mitigating artifacts confound in the CW assessment. Our solution relies on three pillars: 1) identification and removal of artifacts along with band-pass filtering to mitigate baseline-wander and high-frequency noise; 2) a feature extraction scheme for processing data in small batches to reduce memory and smooth and dilute remaining artifact effects; 3) a feature selection and an ML model generation based on data of various subjects to increase the model likelihood of learning the actual underlying process relating variation of CW to the task demands while ignoring random artifact effects. These steps are further detailed in Sections IV and V.

A. CWM Design Methodology and Validation

We propose a flexible ML-based design methodology (Fig. 1a) and data processing strategy for our CWM solution, validating them on the experimental data set described in Subsection VI-A. Our proposed methodology is based on the RF classifier, considering a combination of past modelling success, lightweight inference execution, and model interpretability. We devise and validate our data processing strategy for CWM, obtaining the ML model to be integrated to our wearable device's firmware as the final output.

Our ML-based methodology is mainly built over five steps, see Fig. 1a. First, we remove artifacts and filter noise, then divide data in windows of various sizes. Second, we extract a total of 68 features from each segment, as described in Subsection IV-B4. Then, we select two important parameters: 1) the EEG window providing the best classification accuracy among the various tested segments; 2) the number of trees to be used in the RF model. In the fourth step, we perform a recursive feature elimination on cross-validation (RFECV) to reduce the model complexity while trying to improve classification performance. Finally, we use a hyperparameter optimization method to reduce the final RF model size (in bytes), allowing for its integration to the firmware of our resource-constrained wearable platform.

B. Real-Time CWM System Design

Fig. 1b presents our wearable system implementation targeting real-time CW awareness on HMI solutions. Our CWM system is based on a state-of-the-art wearable device for unobtrusive EEG monitoring. The platform is embodied on glasses temples, a suitable and inconspicuous design for daily use without mobility constraints. The hardware features four high-resolution EEG channels, based on a low-power analogue front-end, and a Bluetooth-low-power (BLE) system-on-chip. Moreover, an ultra-low-power microcontroller is embedded for data processing, reducing application battery requirement. Finally, it is entirely based on off-the-shelf and industrial-grade components, facilitating production and price scaling.

Our data processing strategy can be divided into three stages: 1) noise and artifact removal; 2) feature extraction (power, entropy, and statistical features); 3) cognitive-workload level classification based on an RF model. After each CW inference, the operator's CW state is sent through the BLE interface to the automated system for further use. Furthermore, targeting the implementation of our methodology on resource-constrained wearable devices, the data processing strategy requires adaptations, for instance, to reduce its memory footprint and processing requirements. In particular, the maximum EEG window size that could be processed on the wearable device is limited due to the relatively small amount of RAM memory available. To counteract this problem, we propose a data processing scheme allowing the division of each EEG window into smaller data batches. In the proposed scheme, the final feature values are obtained cumulatively or as averages of intermediate batch results. Hence, we can achieve one inference each batch processing. Section V-C provides a detailed description of the calculation of each chosen feature.

IV. Machine Learning Model Design and Validation

In this section, we describe the design methodology we have followed to develop and validate the proposed data processing methodology for our CWM solution. Particularly, we obtain a lightweight processing flow including a final RF model for a binary classification problem of CW states.

A. Cognitive-Workload Classification Algorithm

Implementing an ML-based data processing strategy on resource-constrained platforms requires the design of a lightweight algorithm, both in terms of RAM memory requirements and execution time. Among available ML algorithms, we chose RF [40] as it provides a robust and fast ensemble classification method (apart from previously stated qualities). Moreover, its algorithm includes two mechanisms to mitigate data overfitting: 1) *boostrapping*, which increases the training data set by sampling random subsets of observations with replacement; 2) taking random sets of features when considering splits for each node in a decision tree.

Although RF inference is relatively simple, storing an RF model in an embedded platform might require considerable amounts of flash memory. To tackle this problem, we explore a hyperparameter optimization to homogenize the maximum depth among all the trees in the forest. Hence, the matrices used to store the model at a low-level representation are shallower and less sparse. In practice, we train an RF model and parse it to a header file [41], which is then used on the wearable system's firmware project. The following subsections provide details on the ML method stages and final model optimization to achieve a small size for deployment on the application.

B. ML Design Methodology for CWM

Our ML-based design methodology considers the simulated search and rescue (SAR) mission data set described in Subsection VI-A, previously pre-processed as described in Subsection VI-B. We divide the available amount of data sessions in cross-validation (training and validation) and testing sets (as specified the Section VI-C). In this division, we consider each data acquisition session as a unit, thus data from an experiment per subject per day is never shared between different classification sets. The cross-validation set is used for evaluating parameters related to the ML model design, features selection and further optimization, which are presented in the following subsections. Moreover, in each step of our proposed ML design methodology, we employ 30-fold cross-validation to cover a representative amount of subject-data combinations. The testing data is only used to assess models' generalization performance as unseen data. Additionally, in the following steps, we guarantee that the same data are used throughout performance comparisons.

1) EEG Artifact Removal: EEG signals are typically in the order of tens of μV. Due to their small amplitude, EEG acquisitions are normally contaminated with physiological artifacts (e.g., blinking activity) and noise from various sources (e.g., 50 Hz line voltage). To mitigate the effect of such artifacts, we propose an artifact removal step composed of: a) an artifact identification stage; b) an artifact removal stage Details of its implementation are given in Subsection V-B. Additionally, we filter the EEG data between 3 and 18 Hz (4th order Butterworth filter) to further mitigate the remaining low-frequency artifacts and remove high-frequency noise.

2) Feature Extraction: As an exploratory step toward obtaining an appropriate EEG window size, we then segment the data in various window sizes (e.i., 8, 16, 24, 32, 40, 48, 56, and 64 s) with 60% overlapping. Such a range of EEG window sizes covers most of those proposed in prior works (Table I), allowing us to have a better overview of the ML performance. Thereafter, the segmented data is used for extracting the 17 features per channel (Subsection V-C), yielding a total of 68 features per segment. Additionally, to reduce inter-subject feature variability, we subtract the subject's baseline state, namely an average value of features obtained during a baseline condition.

3) ML Parameters Selection: We train a RF model for each previously defined window size (Subsection VI-A) and evaluate its effect on the classification scores (defined in Subsection VI-C). Moreover, we repeat this procedure for 100, 200, 300, 400, and 500 trees, also observing the effect of the number of trees on the model performance. Both parameters are selected taking into consideration the final performance obtained in this step. Additionally, we also observe that the

number of trees should be small enough to reduce the final model size (in bytes) without compromising the classification performance.

4) Feature Selection: After defining the window size and the number of trees, the next step in our design methodology is feature selection. It is intended for reducing the number of input variables, thus, potentially improving the performance of the RF model and reducing possible overfitting. Basically, we aim to increase the model likelihood to learn the actual underlying relationship of the training data by considering only the input features that are most representative for classifying the target variable. We execute the feature selection interactively by applying an RFECV method. We rank the features by using the RF returned feature weights and prune the less significant ones after each iteration of the used RFECV method, thus, producing a classification score for all subsets of features used. Finally, we select the subset of features that provides the highest classification score.

5) Model Size Optimization: By setting the hyperparameters related to the RF tree maximum-depth, we limit the maximum input/output pathway length in the model. Hence, we limit one dimension of the matrices used to store the RF model at a low-level representation and, in consequence, reduce the final model size in the microcontroller's flash memory. Three hyperparameters can affect the tree maximum-depth in the case of the MATLAB® RF algorithm, namely: 1) *'MaxNumSplits'*, the maximal number of decision splits (the larger the number, the deeper the tree); 2) *'MinLeafSize'*, the minimum number of leaf node observations (the smaller the number, the deeper the tree); 3) *'MinParentSize'*, the minimum number of branch node observations (the smaller the number, the deeper the tree). As a result, based on *'MaxNumSplits'* and *'MinLeafSize'*, we tune our RF model using Bayesian optimization [42], thus obtaining the out-of-bag quantile error estimate for various combinations of these hyperparameter's values. Considering this estimated error value, we choose new hyperparameters values to generate our final RF model.

C. Assessment of the ML Design Methodology for CWM

First, we assess our proposed ML design methodology for CWM based on only four EEG channels by comparing the results obtained using data from a full EEG-cap (19 channels). We follow our proposed methodology, but adopting a different set of numbers of trees on the first step of model design (i.e., 200, 400, 600, 800, and 1000) and discarding five times more features during the RFECV due to the higher number of features (i.e., 19 channels x 17 features).

Second, to verify the robustness of our methodology against artifacts, we obtain the classification results using an ICA-based method to remove EOG artifacts. Thus, in this second approach, we filter the data between 1 and 40Hz (Butterworth, 4th order, zero-phase) and use the EEGlab's *runica* [43] to decompose the data into independent components. Then, each component is verified for correlation with the available EOG reference. The components with a correlation coefficient superior to 0.8 are removed before back-projecting data into the original sensor space [10].

Finally, we assess the generalization and robustness of the proposed framework applying our methodology on STEW data set [12]. This data set contains EEG data from 48 subjects, which were acquired while subjects executed a multitasking activity. The experimental setup employed the Emotiv EPOC EEG headset sampling at 128 Hz (16 bits, 14 channels, right mastoid reference). STEW data set contains two files per subject, corresponding to two different tasks. Hence, we target a binary classification considering the task as the ground truth. Moreover, due to lack of data at rest condition, we skip the subject feature baseline removal. Additionally, we also include SoA comparison considering the studies in [15] and [10], which also target CWM on the same data set presented in Subsection VI-A.

V. REAL-TIME COGNITIVE-WORKLOAD MONITORING SYSTEM DESIGN

In this section, we detail aspects of our proposed system implementation for real-time CWM. In particular, we present firmware and hardware design characteristics needed to execute our proposed CWM data processing strategy.

A. System Operation

Using an ARM Cortex processor-based wearable device, we adopt the ARM CMSIS real-time operating system port (CMSIS-RTOS) as a base for our application's firmware development. To reduce processing overhead, we handle data acquisition and peripheral communications (e.g., BLE communication) employing direct memory access (DMA) and interrupt service routines (ISR). Moreover, RTOS message queues are used for data synchronization throughout the data processing stages. The RTOS is temporized using a real-time clock (RTC). The RTC facilitates deep low-power modes activation when using the CMIS-RTOS tickless operation mode, which is our target operating mode for reducing energy consumption.

We exploit the ARM CMSIS software library [44] for developing the data processing sub-routines. This library is optimized for ARM Cortex processors, which allows us to speed up data processing execution. Finally, we also use the microcontroller's floating-point unit (FPU) and its digital signal processing (DSP) instructions to further optimize the data processing strategy execution (i.e, filtering and feature extraction). Hence, both approaches contribute to reducing the overall system energy consumption by enabling longer periods in low-power mode. Moreover, similar strategies could be employed with other microcontrollers, considering their specific features such as HW accelerators, manufacturers data processing libraries, etc.

B. Real-Time Data Acquisition and Artifact Removal

Our proposed wearable system implementation relies on only four EEG channels placed next to the F7, T7, F8, and T8 electrode positions over the frontal and temporal lobes (Fig. 2), in a linked-mastoids referential montage. Data storage is handled using circular buffers, thus, allowing processing
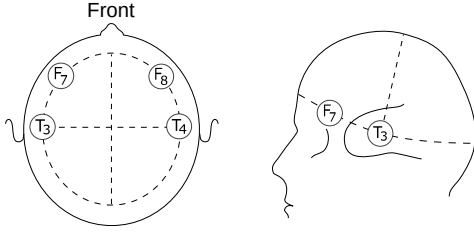
Front

Fig. 2. Proposed electrode placement by the 10-20 international system [49]

chunks of samples at each step. Tackling RAM memory constraints of the target wearable platform, we propose a batch data-processing scheme in which an EEG epoch is divided into various smaller data batches. Each data batch is independently processed and determines the amount of RAM needed.

After acquiring a data batch, we start executing the artifact removal step. As stated in the previous section, first, we identify artifacts within the acquired buffer. Then, we proceed with its removal. The artifact identification stage is executed in three steps. First, we filter the data between 0.5 and 30 Hz to remove baseline-wander. Next, we enhance artifact peaks by applying the relative-energy (RelEN) algorithm [45] to a frontal electrode signal (i.e., F7). Third, we detect peak locations on the RelEN signal considering the two threshold method described in [46]. The RelEn algorithm employs a long and a short data window to calculate the relative energy point by point. In this work, we empirically chose 4 s and 100 ms, respectively for the long and short windows. For the peak detection, we use as thresholds two and three times the standard deviation of the RelEN output per batch.

Following the method proposed in [47], we estimate the artifact using the Savitzky-Golay filter (3rd order, 41 coefficients) in a small region around the artifact peak position. As blinking are the main expected artifacts, we apply the method to 0.5 s of data around the detected peaks (i.e., 100ms before and 400ms after the peak) [48]. After artifact estimation, each trace is subtracted from the original signal.

Finally, we filter the data between 3 and 18 Hz to mitigate possible residual artifact energy and remove high-frequency noise. We use infinite impulse response (IIR) filters (Butterworth, 4th order) for all stages. The filter implementation is based on a second-order-section (SOS) biquad structure to mitigate the effect of filter coefficient quantization. In particular, we employ the CMSIS biquad cascade IIR filter using direct form I SOS structure on our application.

C. Feature Extraction Scheme

According to [5] and [6], several studies have assessed the correlation of CW to variations in task demand levels using EEG bandpowers (i.e., theta, alpha, beta). Besides employing bandpowers, we also expect changes in the signal complexity, amplitude, and waveform between different levels of CW [50], [51]. Therefore, we include features in the time domain and entropies that are able to capture such changes. To meet the proposed data processing scheme, we select and implement a set of features that can be either calculated cumulatively over all data batches or obtained as an average value. Given an EEG

time series X, with possible values $\{x_1, x_2, ..., x_n\}$, a total of 17 features are calculated per EEG channel as follows:

1) Time Domain Features: We estimate the standard deviation (*std*), and calculate the skewness and kurtosis. Besides, we also estimate the Hjorth activity (HA), mobility (HM), and complexity (HC) parameters [50], yielding a total of six time-domain features. All features but skewness and kurtosis are obtained cumulatively among data batches, which are calculated by averaging batch intermediate results. We obtain the mean using (1) and variance using (2) cumulatively, according to [52], [53]. Once reaching the end of a window size, *std* is then obtained from the *var*.

$$\overline{x}_k = \overline{x}_{k-1} + \frac{(x_k - \overline{x}_{k-1})}{k} \tag{1}$$

$$s_k = s_{k-1} + (x_k - \overline{x}_{k-1}) * (x_k - \overline{x}_k) \tag{2}$$

For $2 \leq k \leq n$, variance is estimate as $\sigma^2{}_k = s_k/(k-1)$ [52].

The last three features, the Hjorth parameters dependent on the variance of X and its first and second derivative, as can be observed in the equations below. The first and second derivative's variance are also estimated using (2).

$$HA(X) = var(X) \tag{3}$$

$$HM(X) = \sqrt{\frac{var(X')}{var(X)}} \tag{4}$$

$$HC(X) = \frac{HM(X')}{HM(X)} \tag{5}$$

2) Frequency Domain Features: We calculate the band power of the following EEG bands: theta (4 - 8 Hz), alpha (8 - 12 Hz), and lower beta (12 - 16 Hz). Moreover, we also obtain the following relative band power: 1) each of the aforementioned bands concerning the total power; 2) theta relative to the alpha band; 3) alpha relative to the beta band. The power features are obtained using Welch's Method to estimated the power spectral density (PSD). In this case, each data batch is a segment with 0% overlapping modified using a Hamming window [54]. More specifically, we employ the CMSIS fast Fourier transform (FFT) to obtain each segment's spectral power. It yields a total of 8 features in the frequency domain.

3) Entropy Features: Entropy features are a set of nonlinear measures that reflects the complexity of the EEG signals. In particular, we extract Shannon (SEn), Tsallis (TEn), and Rényi (REn) entropy, which are calculated as follows [51], [55]:

$$SEn = -\sum_{i=1}^{M} p(I_i) \cdot \log_2 p(I_i) \tag{6}$$

$$TEn = \frac{1 - \sum_{i=1}^{M}(p(I_i))^{\beta}}{\beta - 1} \tag{7}$$

$$REn = \frac{1}{1 - \alpha} \cdot \log_2 \left(\sum_{i=1}^{M}(p(I_i))^{\alpha} \right) \tag{8}$$

$P(I)$ is defined as the probability distribution of X given that the k_{th} possible element belongs to one of $\{I_i : i = 1, ..., M\}$

possible intervals of a histogram with M bins. Applying our data batch scheme, we first obtain a cumulative histogram (M=20 bins) of all data batches within a window size and then, estimate $P(I)$. Following [51], we adopt $\alpha = 2$ and $\beta = 2$ to calculate Rényi and Tsallis entropies.

VI. EXPERIMENTAL SETUP

In this section, we describe the database and performance metrics used for training and validating our proposed CWM design methodology. Also, we present the wearable platform used for our CWM system implementation and its firmware execution profiling test to assess current consumption and estimate our system battery life.

A. Data Set

We use the EEG data set acquired on the experiment presented in [15] and [10]. Such experiments aimed at assessing the drone operators' CW level from their physiological signals during a simulated search and rescue (SAR) mission. The experimental setup accounted for subject comfort during data acquisition, having them sitting in a comfort arm-chair in front of a medium-size screen and interacting with the simulator with a hand game-pad. This setup aims to reduce body and head movements to decrease the number of muscular artifacts. EEG signals were acquired using a Biosemi ActiveTwo system [56], at 2048 Hz, having electrodes placed according to the 10-20 international system [49].

A total of 24 volunteers (27.7 ± 4.8 years old) took part in the experiment, executing the experiments twice (on different days). All of them signed a written consent and the study has the approval of the Cantonal Ethics Commissions for Human Research Vaud and Geneva (no. PB2017-00295). The experimental setup is the following:

- It was used a simulator for SAR with drones that includes four different types of tasks, namely: 1) *baseline (B)*, an auto-pilot passive task; 2) *flying (F)* the drone through the centre of a set of waypoints (circles marking the pathway); 3) *mapping three objects (3M)*, a task of identifying three objects by colour by pressing a video-game-like control button over an auto-pilot flight; 4) *flying and mapping three objects (F3M)*, combining the two previous tasks to create the highest level of workload among all tasks.
- During each experimental session (approximately 56 min of data acquisition), a subject performed all the above-described tasks three times: 1) on the first trial, each task lasted five minutes, with three minutes of rest in-between; 2) during the second and third trials, tasks lasted three minutes each without rest in between. The task sequence was chosen randomly among participants.

The available data is grouped by experimental session, a session per subject per day of the experiment, yielding a total of 48 separated subsets of EEG (approximately 56 min of data per session). Finally, according to the averaged self-reported difficulty level stated in [10], the F is the second most challenging task. Therefore, we use data from *F3M* and *F* tasks for our ML binary model design and validation. Additionally, the data of the first trial *baseline* task is used to extract the subject's baseline CW condition (as described in Subsection IV-B).

B. Data Set Pre-Processing

In general ML-based approaches, the data set has to be pre-processed before use. In relation to the target data set, first, we modify the data to resemble the output of our proposed wearable platform. We use the EEGLab [43] re-reference tool to apply an average linked-mastoid reference. Additionally, we downsample the data to 256 Hz.

Second, there are various specific factors (e.g., skills, attention, engagement) that can contribute to a subject not perceiving a task as cognitively demanding. Hence, we remove data sessions presenting not observable CW variation between tasks. To this aim, we perform a pairwise statistical test to compare the features extracted from the two selected tasks within a data session (experiment per subject per day). Since the window size is further selected in our methodology, we assess the statistical difference for all previously defined ones using a non-parametric two-sided Wilcoxon rank-sum test (5% significance level) [57]. Finally, we discard the sessions having less than 33% (empirically chosen) of the total number of features with statistical differences.

C. CWM Model Performance Assessment

We execute the ML model design and validation methodology proposed in Section IV using MATLAB®. From the available number of data sessions, first, we sort the testing set (20%) for future model generalization assessment (unseen data). Second, we divide the remaining data into training (60%) and validation (20%) sets. All sessions have approximately the same amount of data points per task, thus our data set is balanced. Therefore, we evaluate the performance of our proposed ML design using the accuracy (Acc), sensitivity (Sens), specificity (Spec), and their geometric mean (Gmean) scores. These scores are defined as follows:

$$Acc = \frac{Tp + Tn}{Tp + Tn + Fp + Fn} \tag{9}$$

$$Sens = \frac{Tp}{Tp + Fn}, \qquad Spec = \frac{Tn}{Tn + Fp} \tag{10}$$

$$Gmean = \sqrt{Sens \cdot Spec} \tag{11}$$

where Tp, Tn, Fp, and Fn stand for true positive, true negative, false positive, and false negative, respectively.

D. Application Profiling and Battery Life Analysis

We target the e-Glass wearable device [39] as implementation platform for our real-time data processing described in Section V. e-Glass is designed using extensively validated off-the-shelf components, namely: 1) the STM32L476 ultra-low-power ARM Cortex-M4 microcontroller, featuring a 1 MB flash memory and a 128 KB RAM, and reaching up to 80 MHz clock; 2) the ADS1299 EEG Front-End, a complete EEG System-on-Chip in reduced package size; 3) the BlueNRG-MS, Bluetooth-low-energy (BLE) network processor.

The real-time data processing for CWM described in Section V was implemented for e-Glass platform using the CMSIS-RTOS. Among other advantages, the RTOS includes a run-time tracing utility we use to profile the processing workload per thread created. This tracing tool employs a secondary timing source (i.e., a microcontroller timer peripheral) to collect information on each thread's total execution time. This statistic includes the Idle task execution (automatically created by the RTOS), allowing us to estimate the amount of idle processing time. A second tracing tool we use is the data watchpoint and trace component (DWT), available on e-Glass' ARM Cortex-M4 microcontroller. DWT includes a clock cycle counter, the CYCCNT, which we use to determine the number of clocks used by specific parts of code, aiming at optimization and execution bottlenecks identification.

Considering the run-time statistic utility, we obtain the execution profile for our firmware over a 60-minute period (with 10 μs resolution). We process a 4 s data-batch while e-Glass is in run mode (lowest processing overhead). Moreover, we assess the firmware execution time breakdown by setting the clock counter for each main processing block (i.e, artifact removal, feature extraction, and the ML inference).

Finally, regarding battery life, we measured e-Glass' current consumption using a Fluke 8846A digital multimeter, 6.5 digit resolution, 100 μA to 10 A current range (with up to 100 pA resolution). The measures were taken at 4 Hz, considering e-Glass at three different operation schemes: 1) running the application, microcontroller always in run mode (main clock at 80 MHz); 2) microntroller in deep sleep mode, peripherals active; 3) running the application, activating RTOS tickless mode (switching from run-mode to deep-sleep mode when executing the Idle task). These three scenarios indicate the maximum and minimum current consumption, and the target operation mode current consumption (considering processing overhead to switch between modes).

VII. RESULTS AND DISCUSSION

This section presents the results for the validation of our proposed ML design methodology for CWM. We also present the comparison with a full-cap solution, the ICA-based artifact-removal technique usage, and the STEW data set. Finally, we present the execution profiling results and battery life-time estimation when implementing our proposed methodology on the e-Glass platform.

A. ML Model Design and Evaluation

The first step for our ML model design is to select the window size and RF number of trees. The obtained overall cross-validation Gmean scores show a relatively small variation for different window sizes and number of trees (ranging from $69.5-77.6\%$). The 56 s window size output the highest average Gmean across all number of trees when using 200 trees. Hence, we select the 56 s (with 60% overlap) window size and 200 trees as parameters to be used in the next steps of our ML design methodology.

Second, we proceed with the feature selection step aiming at reducing the number of features used by our RF model and,
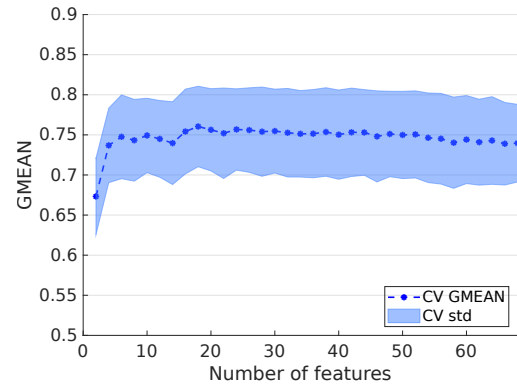

Fig. 3. Feature selection score evolution per number of most important features.

therefore, decrease the final computational load of this step on the real-time implementation. We apply the RFECV pruning the two least important features per iteration. Fig. 3 presents the cross-validation Gmean scores and their standard deviation for the RFECV process. We observe a fast rise in Gmean with the increment of a few features, reaching the maximum score value when using 18 features. From this point on, there is a slowly decreasing trend with the increment of the number of features.

Table II shows that the cross-validation Gmean increases from $74.7 \pm 4.3\%$ to 76.7 ± 5.0 % when the number of used features reduces from 68 to 18 features. Moreover, we also observe stable classification scores when using the selected set of features on the unseen test data. In terms of model size, models generated for the number of features (68 and 18) would require quite a considerable number of bytes to be stored on the microcontroller memory. Thus, a requirement of up to 3,126.0 KB (after compilation) of flash memory is not feasible for wearable platforms. Therefore, we need an additional model optimization step, towards model size reduction.

B. Optimization for Model Size Reduction and Generalization

In general, a decision tree (DT) model is a map of possible outcomes, connecting inputs (features) throught a pathway to each leaf (containing assigned labels). Each path includes nodes, branching the way toward a possible outcome according to feature weights. As a result, an RF model, an ensemble of DT models, can be stored on the microcontroller flash memory in groups of matrices containing the number of branches, path lengths, branch weights, labels for each leaf, etc. The sum of each matrix storage capacity is directly related to the model size in bytes. More specifically, the maximum input/output pathway length is one dimension of such matrices, thus, shallower trees produce smaller models. Moreover, although limiting the maximum tree depth causes the generation of more pathways (e.g., leading to more leaves, branches, weights), shallower trees also have higher input/output lengths homogeneity. Hence, shallower threes also produce less sparse storage matrices.

To achieve a shallower tree, we optimize the hyperparameters of our final RF model to reduce the maximum tree depth. In the case of RF, we employ a Bayesian optimization concerning the *MaxNumSplits* and *MinLeafSize* parameters and obtain the error curve in Fig. 4. For the chosen hyperparameters,

Table II
ML METHODOLOGY PERFORMANCE: THE INFLUENCE OF FEATURE SELECTION (RFECV) AND HYPERPARAMETER OPTIMIZATION IN PERFORMANCE SCORES (±STANDARD DEVIATION) AND MODEL SIZE – e-GLASS VERSUS FULL-CAP SOLUTION.

System Config.	RFE CV	Model Opt.	Feat. (total)	CV (%)		Testing (%)				Model Size (KB)
				Acc	Gmean	Acc	Gmean	Sens	Spec	
e-Glass	No	No	68	74.7 ± 4.3	74.0 ± 4.8	76.4	76.1	82.3	70.1	2,901.0
(4 ch.,	Yes	No	18	76.7 ± 5.0	76.0 ± 5.0	76.3	76.1	81.6	70.9	3,126.0
200 trees)	Yes	Yes	18	75.8 ± 4.7	75.1 ± 5.0	74.5	74.0	82.9	66.1	113.5
Full-Cap	No	No	323	80.6 ± 5.4	79.8 ± 6.0	81.2	81.0	76.7	85.7	5,646.9
(19 ch.,	Yes	No	33	82.3 ± 5.2	81.7 ± 5.7	80.8	80.4	72.3	88.6	4,814.5
400 Trees)	Yes	Yes	33	82.0 ± 5.2	81.3 ± 5.8	80.5	80.2	73.3	87.7	645.7

the small the *MaxNumSplits* the shallower a tree. Also, as larger the *MinLeafSize* as shallower a tree. Thus, we select the model minimum feasible point located at *MaxNumSplits*=17 and *MinLeafSize*=1 (lowest error) to generate our final model.

Table II presents the scores for our final model and its memory size after optimization. Our newly trained model considers the above selected hyperparameters, employing the feature set returning the highest Gmean score on the RFECV (18 features in this case). The final model requires only 113.5 KB of memory, a 27.5x size reduction compared to the one generated in the previous ML stage, which is leveraged by the RFECV. In case we optimize model size for the full feature set, we only achieve a 3.5x reduction (model size=824.5 KB). Additionally, optimize model's hyperparameters causes only a small variation on classification scores. Hence, we obtained a model showing marginal variation in classification scores but with a considerable size reduction, thus allowing us to fit it in our target platform flash memory. Nonetheless, regarding a trade-off between size and accuracy, another set of hyperparameters values can be chosen if further reduction is required. In particular, by using *MaxNumSplits*=10 and *MinLeafSize*=1, the model size is further reduced to 56.8 KB while achieving a $75.3 \pm 5.1\%$ and $76.0 \pm 4.8\%$ of cross-validation Gmean and Acc, respectively.

Finally, we use the final model to assess our proposed methodology generalization to the unseen testing set. Our results show a 74.0% Gmean score on the unseen test set (Sens=82.9% and Spec=66.1%) and 74.5% of Acc, which are within cross-validation score range. Thus, it indicates a good model generalization.

C. CWM Design Assessment Results

We evaluate the performance of our proposed ML design methodology for CWM and data processing strategy in three different scenarios, as previously stated in Subsection IV-C.

First, we apply the ML design methodology using data from a full EEG-cap, a common setup in the literature [5], [6], [10]. We obtain the highest cross-validation Gmean score using 400 trees. Thereafter, the RFECV and hyperparameter optimization results are also given in Table II. Results show only a 5.7% higher Gmean in CV (best one in average) and a 6.2% higher Gmean for the testing set after model optimization compared to the e-Glass solution. Similar differences are observed for the other performance scores.

Furthermore, analysing the feature ranking returned by RF on the RFECV, the 18 most important features are from 11 different EEG electrodes over the frontal/central region of the scalp (i,e.,
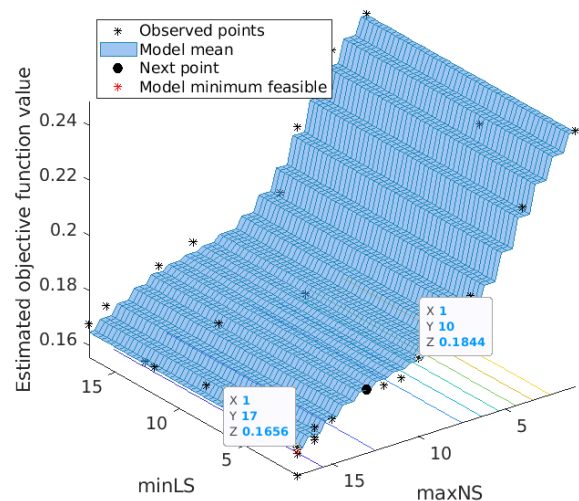

Fig. 4. Model optimization: out-of-bag quantile error vs complexity
.

F3, C3, P7, Fz, F4, Cz, C4, T8, O2, Fp1, and Fp2). Although providing slightly higher classification scores, such a solution would not be feasible on a resource-constrained embedded system. Such a system lacks RAM memory and processing capacity for real-time evaluation and poses considerable battery life restrictions. Besides, EEG caps or headsets are considered cumbersome and less accepted by possible users, for instance, due to social stigma [18], [19]. Therefore, the Gmean score difference between e-Glass and a full-cap solution is considered acceptable when observing the gains in usability, hardware simplification, processing workload reduction, and battery life. Additionally, as discussed in [6], human performance would be more affected in more extreme CW conditions and we could optimize the sensitivity of our proposed solution for those more extreme conditions (e.g., optimizing decision threshold).

Second, our ML design methodology and data processing scheme combines various steps to mitigate possible artifact confound on our proposed CWM solution. More specifically, we tailored an artifact removal algorithm for e-Glass, our target wearable platform. To assess the robustness of our methodology against artifacts, we employed an ICA-based method for artifact removal to replace our proposed algorithm. We obtain Gmean=73.1% and Acc=74.9% on the test set when using ICA (CV-Gmean=$73.1 \pm 5.9\%$ and CV-Acc=$73.8 \pm 5.5\%$). Hence, such results indicate that our methodology provide a similar CW assessment compared to a scenario in which includes a gold-standard artifact removal step is used.

Next, we apply our ML design methodology for CWM to

Table III
COMPARISON WITH THE SoA.

Work	Database	Signals	Accuracy
[10]	SAR	EEG (25 ch)	77%
[12]	STEW	EEG (14 ch)	69% (79.2%*)
[15]	SAR	PPG, RSP, SKT, ECG	80.20%
This work	SAR	EEG (4 ch)	74.5%
This work	STEW	EEG (4 ch)	83.6%

*Adjusting it for 2 classes, as explained in Subsection VII-C

the STEW data set to assess its generalization capacity [12]. Following our methodology, we select the EEG window size of 32 s and obtain 200 trees as parameters for generating the RF model in the feature selection and model optimization steps. As final results (see Table III), we achieve Acc=83.6% and Gmean=83.6% in the testing set, within the range of values obtained with our primary data set. Moreover, authors of [12] reported an Acc=69.2% for a three-class problem. However, a reformulation of their presented confusion matrix for a two-class problem, grouping [12]'s results for the moderate workload to the high condition count, returns Acc=79.2%. Such reformulation is considered to be reasonable as we observe that the median difference between moderate and high perceived workload is relatively higher than the low to moderate in a similar work [15]. Hence, our results are 4.4% superior to the ones inferred from the original study. These results indicate that our solution is robust and generally applicable albeit different in experimental setup and methodology of data analysis.

Finally, a direct comparison with SoA is shown in Table III. We achieve Acc=74.5% in the testing set, on par with the results of SoA works targeting CWM with various biosignals or using a full-cap solution. For instance, the authors in [10] achieve Acc=77% for the same classification problem on SAR database, but through the use of data from multiple channels. In turn, the work in [15] reports Acc=80.2% also for SAR database, but employing four peripheral physiological signals to classify between *baseline* and *F3M* tasks. Therefore, our proposed solution for CWM enables awareness of the operator's CW state on HMI, using edge wearable sensors with only four EEG channels.

D. Execution Profiling and Battery Life Estimation

We implement our proposed methodology for CWM on the e-Glass wearable platform. By profiling its execution for one hour, we obtained the RTOS task run-time breakdown given in Table IV. We can observe that 97.83% of the time e-Glass is executing the Idle task (waiting for execution demands from other tasks). Only 1.28% of the total time is used to run our proposed ML data processing strategy for CWM. The remaining time goes to processing overhead (e.g., battery charge control, BLE callbacks, acquired data parsing). Therefore, our proposed methodology is lightweight, allowing the e-Glass platform to execute it in a relatively short period to improve battery life.

Regarding the proposed ML approach, Table V displays its execution time breakdown when processing a 56 s EEG window size in 14 batches of 4 s. Each 4 s data batch is processed in approximately 56.4 ms (microcontroller main clock at 80 MHz), 60% of which is spent on feature extraction. Moreover, by

employing the proposed multiple data-batches feature extraction scheme, we reduce the RAM memory footprint for our data processing strategy by approximately 14x compared to a single batch processing. Hence, the proposed data processing scheme enables the algorithm execution on our wearable platform.

The inference period of 14.8 ms is quite small compared to the total time (1.8%). However, if we consider other applications using smaller window sizes, the inference time becomes significant. For instance, with an 8 s window size, the inference could account for 13.11% of the ML algorithm execution as its execution time is about the same considering the same number of features and trees. Similarly, for applications that require higher inference rates, the size of the model becomes a sensitive parameter when accounting for the total computation time. In such cases, the inference execution time might be in the same order of magnitude as the combined execution of pre-processing and feature-extraction steps.

Table IV
TASKS PROFILING: EXECUTION TIME BREAKDOWN.

System Tasks	Exec. Time (%)
Data management	0.29
Data processing	1.28
System manag.	0.59
RTOS TMR	0.02
Idle	97.83

Table V
ML PROFILING: EXECUTION TIME BREAKDOWN.

Data processing Tasks	Exec. Time (ms)
Filtering	46.59
Artifact Removal	276.01
Feature Extraction	481.63
RF inference	14.79
Total	804.23

Finally, Table VI provides e-Glass' current consumption in three different scenarios. The first scenario corresponds to the highest expected current consumption, on average, and the second to the lowest. In the third scenario, we activate the RTOS tickless operation mode, which is the target e-Glass operating mode of our application.

Current measurements were taken on the battery side as the average of a thousand samples. Due to the input capacitive/inductive power-supply filtering circuitry and the low frequency of high current events (e.g., a BLE transmission occurs each 1 s), the obtained average current values provide a reliable system power consumption assessment. Moreover, the current measurements are within the expected range, as follows:

- The first scenario includes the current consumption throughout the entire platform, i.e, including the microcontroller (estimated as approximately 14 mA using STmicroelectronics STM32CubeMX software v5.6), the EEG front-end (approximately 4.6 mA, based on the ADS1299 datasheet), BLE (mostly in sleep mode, drawing 3.5 μA). The difference is drawn by other circuitry (e.g., voltage regulators, pull-up resistor, etc.).
- During the second scenario, the microcontroller is kept in deep-low-power mode after peripheral configurations.

Hence, we observe that most of the current is consumed by the ADS1299 chip.

- Using the third scenario, we assess the effect of the RTOS tickless operation mode on the total current consumption. We observe a 1.47 mA increment in current consumption mostly related to the microcontroller operation to acquire and process the EEG data.

In terms of battery life estimation, when operating on the tickless mode, e-Glass can execute our proposed CWM algorithm for up to 28.59 hours on a single battery charge employing a 225 $mA \cdot h$ LiPo battery.

Table VI
MEASUREMENTS OF e-Glass' CURRENT CONSUMPTION DURING DIFFERENT OPERATION MODES.

System Operation Mode	Current (mA)
Active (Run mode - 80 MHz)	22.45
Low-power (Deep-low-power mode)	6.4
Tickless (Run/Low-power modes)	7.87
Battery life (225 $mA \cdot h$)	28.59 h

VIII. CONCLUSION

In this work, we have proposed a design methodology, and a data processing strategy for CWM on wearable devices aiming to provide a reliable and optimized tool to enable awareness of the operators' cognitive states on HMI. Then, we have validated our design methodology on an experimental data set, reaching a Gmean of 74.0% between sensitivity and specificity on unseen data.

The proposed lightweight ML-based data processing strategy for resource-constrained wearable devices implements two solutions for the main memory-related problems in such platforms, namely: 1) a data-batch processing scheme for feature extraction, reducing the RAM memory footprint of our real-time implementation by 14x compared to a single batch processing; 2) a hyperparameter optimization strategy aiming to obtain shallower RF models, lowering the final model size by 27.5x (requiring only 113.5 KB of flash memory).

Finally, we have profiled the execution of our proposed CWM data processing strategy in the e-Glass platform, in terms of processing workload and energy consumption. Hence, it requires only 1.28% of the available processing time to be executed, drawing only 7.87 mA during operation.

All in all, our methodology can be executed on a resource-constrained wearable platform, thus achieving 28.59 hours of operation on a single battery charge. Therefore, in this work we have provided a CWM solution for seamless integration into modern and highly automated systems to enhance HMI, leveraging conditions for higher performance and safer operation for more than a working shift.

REFERENCES

[1] E. Lodgaard and S. Dransfeld, "Organizational aspects for successful integration of human-machine interaction in the industry 4.0 era," *Procedia CIRP*, vol. 88, pp. 218–222, 2020.

[2] M. Mouloua *et al.*, "Human Monitoring of Automated Systems," in *Human Performance in Automated and Autonomous Systems*, M. Mouloua and P. A. Hancock, Eds. Boca Raton: Taylor & Francis, 2020, ch. 1st.

[3] R. Parasuraman and V. Riley, "Humans and Automation: Use, Misuse, Disuse, Abuse," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 39, no. 2, pp. 230–253, 6 1997.

[4] D. Kahneman, *Attention and Effort*. Englewood Cliffs, NJ: Prentice-Hall, 1973.

[5] G. Borghini *et al.*, "Measuring neurophysiological signals in aircraft pilots and car drivers for the assessment of mental workload, fatigue and drowsiness," *Neuroscience & Biobehavioral Reviews*, vol. 44, pp. 58–75, 7 2014.

[6] F. Babiloni, "Mental Workload Monitoring: New Perspectives from Neuroscience," in *Communications in Computer and Information Science*, vol. 1107. Springer, 11 2019, pp. 3–19.

[7] D. C. Evans and M. Fendley, "A multi-measure approach for connecting cognitive workload and automation," *International Journal of Human-Computer Studies*, vol. 97, pp. 182–189, 1 2017.

[8] P. Arico *et al.*, "Passive BCI in Operational Environments: Insights, Recent Advances, and Future Trends," *IEEE Transactions on Biomedical Engineering*, vol. 64, no. 7, pp. 1431–1436, 7 2017.

[9] B. Blankertz *et al.*, "The Berlin Brain-Computer Interface: Progress Beyond Communication and Control," *Frontiers in Neuroscience*, vol. 10, 11 2016.

[10] P. K. Jao *et al.*, "EEG Correlates of Difficulty Levels in Dynamical Transitions of Simulated Flying and Mapping Tasks," *IEEE Transactions on Human-Machine Systems*, pp. 1–10, 2020.

[11] G. F. Wilson and C. A. Russell, "Real-time assessment of mental workload using psychophysiological measures and artificial neural networks." *Human factors*, vol. 45, no. 4, pp. 635–643, 2003.

[12] W. L. Lim, O. Sourina, and L. P. Wang, "STEW: Simultaneous task EEG workload data set," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 26, no. 11, pp. 2106–2114, 2018.

[13] D. Miklody *et al.*, "Maritime Cognitive Workload Assessment," in *Symbiotic Interaction*, L. Gamberini *et al.*, Eds. Cham: Springer International Publishing, 2017, pp. 102–114.

[14] C. Tremmel *et al.*, "Estimating Cognitive Workload in an Interactive Virtual Reality Environment Using EEG," *Frontiers in Human Neuroscience*, vol. 13, p. 401, 11 2019.

[15] F. Dell'Agnola *et al.*, "Cognitive workload monitoring in virtual reality based rescue missions with drones," in *12th International Conference on Virtual, Augmented and Mixed Reality*, Copenhagen, 2020.

[16] G. Masinelli *et al.*, "Self-Aware Machine Learning for Multimodal Workload Monitoring During Manual Labor on Edge Wearable Sensors," *IEEE Design Test*, vol. 2356, no. c, p. 1, 2020.

[17] R. L. Charles and J. Nixon, "Measuring mental workload using physiological measures: A systematic review," *Applied Ergonomics*, vol. 74, pp. 221–232, 2019.

[18] V. Mihajlovic *et al.*, "Wearable, wireless EEG solutions in daily life applications: What are we missing?" *IEEE Journal of Biomedical and Health Informatics*, vol. 19, no. 1, pp. 6–21, 2015.

[19] C. Hoppe *et al.*, "Novel techniques for automated seizure registration: Patients' wants and needs," *Epilepsy and Behavior*, vol. 52, pp. 1–7, 2015.

[20] H. Mamaghanian *et al.*, "Compressed Sensing for Real-Time Energy-Efficient ECG Compression on Wireless Body Sensor Nodes," *IEEE Transactions on Biomedical Engineering*, vol. 58, no. 9, pp. 2456–2466, 9 2011.

[21] G. Surrel *et al.*, "Online Obstructive Sleep Apnea Detection on Medical Wearable Sensors," *IEEE Transactions on Biomedical Circuits and Systems*, no. 99, pp. 1–12, 2018.

[22] D. Sopic *et al.*, "Real-Time Event-Driven Classification Technique for Early Detection and Prevention of Myocardial Infarction on Wearable Systems," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 12, no. 5, pp. 982–992, 10 2018.

[23] S. Betti *et al.*, "Evaluation of an Integrated System of Wearable Physiological Sensors for Stress Monitoring in Working Environments by Using Biological Markers," *IEEE Transactions on Biomedical Engineering*, vol. 65, no. 8, pp. 1748–1758, 8 2018.

[24] A. Turner and S. Hayes, "The Classification of Minor Gait Alterations Using Wearable Sensors and Deep Learning," *IEEE Transactions on Biomedical Engineering*, vol. 66, no. 11, pp. 3136–3145, 11 2019.

[25] S. Sonune *et al.*, "Issues in IoT healthcare platforms: A critical study and review," in *2017 International Conference on Intelligent Computing and Control (I2C2)*. IEEE, jun 2017, pp. 1–5.

[26] J. Zhang, Z. Yin, and R. Wang, "Recognition of Mental Workload Levels Under Complex Human–Machine Collaboration by Using Physiological Features and Adaptive Support Vector Machines," *IEEE Transactions on Human-Machine Systems*, vol. 45, no. 2, pp. 200–214, 4 2015.

[27] W. Shi *et al.*, "Edge Computing: Vision and Challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.

[28] L.-l. Chen *et al.*, "Automatic detection of alertness/drowsiness from physiological signals using wavelet-based nonlinear features and machine learning," *Expert Systems with Applications*, vol. 42, no. 21, pp. 7344–7355, nov 2015.

[29] N. Attaran *et al.*, "Embedded Low-Power Processor for Personalized Stress Detection," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 65, no. 12, pp. 2032–2036, 12 2018.

[30] A. Golgouneh and B. Tarvirdizadeh, "Fabrication of a portable device for stress monitoring using wearable sensors and soft computing algorithms," *Neural Computing and Applications*, pp. 1–23, 6 2019.

[31] R. Zanetti, A. Aminifar, and D. Atienza, "Robust Epileptic Seizure Detection on Wearable Systems with Reduced False-Alarm Rate," in *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*. IEEE, 7 2020, pp. 4248–4251.

[32] V. Kartsch *et al.*, "An Energy-Efficient IoT node for HMI applications based on an ultra-low power Multicore Processor," in *2019 IEEE Sensors Applications Symposium (SAS)*. IEEE, 3 2019, pp. 1–6.

[33] A. Gevins and M. E. Smith, "Neurophysiological measures of cognitive workload during human-computer interaction," *Theoretical Issues in Ergonomics Science*, vol. 4, no. 1-2, pp. 113–131, 1 2003.

[34] S. G. Hart and L. E. Staveland, "Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research," 1988, pp. 139–183.

[35] F. Dell'Agnola, L. Cammoun, and D. Atienza, "Physiological characterization of need for assistance in rescue missions with drones," in *IEEE International Conference on Consumer Electronics (ICCE)*, IEEE. IEEE, 1 2018, pp. 1–6.

[36] P. Zarjam, J. Epps, and N. H. Lovell, "Beyond Subjective Self-Rating: EEG Signal Classification of Cognitive Workload," *IEEE Transactions on Autonomous Mental Development*, vol. 7, no. 4, pp. 301–310, 12 2015.

[37] P.-K. Liu *et al.*, "Entropy and Complexity Assisted EEG-based Mental Workload Assessment System," in *2019 IEEE Biomedical Circuits and Systems Conference (BioCAS)*. IEEE, 10 2019, pp. 1–4.

[38] M. Chiang and T. Zhang, "Fog and IoT: An Overview of Research Opportunities," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 854–864, dec 2016.

[39] D. Sopic, A. Aminifar, and D. Atienza, "E-Glass: A Wearable System for Real-Time Detection of Epileptic Seizures," in *Proceedings - IEEE International Symposium on Circuits and Systems*, vol. 2018-May. Institute of Electrical and Electronics Engineers Inc., 4 2018.

[40] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[41] P. Kendrick and D. Hasenfratz, "MATLAB TreeBagger model extraction," 2014. [Online]. Available: https://go.epfl.ch/rf2h

[42] MathWorks, "Tune Random Forest Using Quantile Error and Bayesian Optimization." [Online]. Available: https://go.epfl.ch/rfbopt

[43] A. Delorme and S. Makeig, "EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics including independent component analysis," *Journal of Neuroscience Methods*, vol. 134, no. 1, pp. 9–21, mar 2004.

[44] Arm Limited, "Cortex Microcontroller Software Interface Standard (CMSIS)," 2020. [Online]. Available: https://developer.arm.com/tools-and-software/embedded/cmsis

[45] S. Yazdani, S. Fallet, and J. M. Vesin, "A Novel Short-Term Event Extraction Algorithm for Biomedical Signals," *IEEE Transactions on Biomedical Engineering*, vol. 65, no. 4, pp. 754–762, 4 2018.

[46] L. Orlandic *et al.*, "REWARD: Design, Optimization, and Evaluation of a Real-Time Relative-Energy Wearable R-Peak Detection Algorithm *," in *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 7 2019, pp. 3341–3347. [Online]. Available: https://ieeexplore.ieee.org/document/8857226/

[47] D. Szibbo, A. Luo, and T. J. Sullivan, "Removal of blink artifacts in single channel EEG," *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, pp. 3511–3514, 2012.

[48] D. Hagemann and E. Naumann, "The effects of ocular artifacts on (lateralized) broadband power in the EEG," *Clinical Neurophysiology*, vol. 112, no. 2, pp. 215–231, 2001.

[49] G. H. Klem *et al.*, "The ten-twenty electrode system of the International Federation. The International Federation of Clinical Neurophysiology." *Electroencephalography and clinical neurophysiology. Supplement*, vol. 52, pp. 3–6, 1 1999.

[50] R. Jenke, A. Peer, and M. Buss, "Feature extraction and selection for emotion recognition from EEG," *IEEE Transactions on Affective Computing*, vol. 5, no. 3, pp. 327–339, 2014.

[51] U. R. Acharya *et al.*, "Application of entropies for automated diagnosis of epilepsy using EEG signals: A review," *Knowledge-Based Systems*, vol. 88, pp. 85–96, 2015.

[52] D. E. Knuth, "Accuracy of Floating Point Arithmetic," in *The Art of Computer Programming: Seminumerical Algorithms*, 2nd ed. Addison-Wesley Publishing Company, 1981, ch. 4th, p. 216.

[53] B. P. Welford, "Note on a Method for Calculating Corrected Sums of Squares and Products," *Technometrics*, vol. 4, no. 3, pp. 419–420, 1962.

[54] P. Stoica and R. Moses, "Nonparametric Methods," in *Spectral Analysis of Signals*, 1st ed. Prentice Hall, 2005, ch. 2, pp. 22–71.

[55] S. Tong *et al.*, "Parameterized entropy analysis of EEG following hypoxic-ischemic brain injury," *Physics Letters, Section A: General, Atomic and Solid State Physics*, vol. 314, no. 5-6, pp. 354–361, 2003.

[56] Biosemi, "ActiveTwo: 280-channel, DC amplifier, 24-bit resolution, biopotential measurement system with Active Electrodes." [Online]. Available: https://www.biosemi.com/products.htm

[57] W. Haynes, "Wilcoxon Rank Sum Test," in *Encyclopedia of Systems Biology*. New York, NY: Springer New York, 2013, pp. 2354–2355.