Real-Time Motion Artifact Removal in fNIRS with Denoising Autoencoder at the Edge

Jinchen Li
HUB of Intelligent Neuro-engineering (HUBIN), DSIS
University College London (UCL)
London, UK
jinchen.li.23@ucl.ac.uk

Yunjia Xia
HUB of Intelligent Neuro-engineering (HUBIN), DSIS
University College London (UCL)
London, UK
yunjia.xia.18@ucl.ac.uk

Jingyu Lei
HUB of Intelligent Neuro-engineering (HUBIN), DSIS
University College London (UCL)
London, UK
jingyu.lei.22@ucl.ac.uk

Robert J. Cooper
DOT-HUB, Department of Medical Physics and Biomedical Engineering
University College London (UCL)
London, UK
robert.cooper@ucl.ac.uk

Hubin Zhao
HUB of Intelligent Neuro-engineering (HUBIN), DSIS
University College London (UCL)
London, UK
hubin.zhao@ucl.ac.uk

***Abstract*—Functional near-infrared spectroscopy (fNIRS) can be used to measure cortical hemodynamics, with advantages such as non-invasiveness, high spatial resolution, wearability, ease of use and relatively low cost. These features make it potentially suitable for translational applications such as brain-computer interface (BCI), neurofeedback, and personalized healthcare. However, fNIRS signals are susceptible to motion artifacts (MAs), which can obscure physiological information. Herein, we propose to deploy a denoising autoencoder (DAE) network on an STM32 microcontroller for real-time multichannel MA removal at the edge. The DAE model was trained on fNIRS data augmented with simulated MAs. It was deployed on the edge device without any performance degradation, outperforming the conventional wavelet-based methods. With an inference time of 38 ms, this implementation is well-suited for real-time processing of multi-channel fNIRS data. Additionally, the low memory usage and CPU workload of the model make it ideal for deployment on diverse microcontroller platforms. This work holds the potential to enable the wider applications of wearable fNIRS in practice.**

Keywords*—*fNIRS, motion artifact, DAE, edge processing, real-time, multichannel

I. Introduction

Brain-computer interfaces (BCIs) offer a valuable tool for patients with disabilities, enabling them to control external devices and communicate with their environments [1]. Among various modalities used in BCI, functional near-infrared spectroscopy (fNIRS) has become a widely adopted method for measuring brain activities [1, 2]. fNIRS offers several advantages, including non-invasiveness, high spatial resolution, wearability, ease of use and relatively low cost [3, 4]. This technique leverages the distinct absorption spectra of oxy-hemoglobin and deoxy-hemoglobin, two important molecules for oxygen transport, within the near-infrared range [5]. By measuring changes in near-infrared light intensity as it passes through the brain and scalps, fNIRS can provide insight into relative hemoglobin concentration change, which further indicates change in brain activities [6]. However, fNIRS is susceptible to motion artifacts (MAs), which result from subject movements during data collection, and cause unexpected fluctuations optical coupling, leading to spurious peaks or drifts in time-series data, thereby affecting the quality and accuracy of the measurements [7, 8].

Traditional time-series MA removal techniques, such as wavelet filtering, often require manual parameter tuning and are associated with long processing time [9]. In contrast, learning-based strategies, such as denoising autoencoders (DAEs), eliminate the need for parameter fine-tuning [10], and have shown promising results in MA removals with higher accuracy and reduced processing time [7]. However, most of deep learning-based MA removal algorithms are still performed offline and rely on benchtop computers, limiting the efficiency and wearability of fNIRS systems [11, 12]. For patients with disabilities to use BCI to control the external devices, the fNIRS output must be both fast and accurate. Noises such as MAs can be misinterpreted as brain activities, compromising the accuracy of BCI, especially in environments with significant movements.

To implement deep learning models for real-time MA removal without compromising the wearability of fNIRS technologies, accelerating the deep learning model is essential. One approach is cloud-based processing, where the computational power of the cloud is leveraged for acceleration, and the processed data was then transmitted back to the fNIRS device [13]. An alternative approach is edge processing [5], allowing for the optimization and acceleration of deep learning models on local (edge) devices [14]. Given that the deep learning models used for real-time MA removal in fNIRS are relatively straightforward, the computational power of cloud processing could be optional. By eliminating the need for data transmission, edge processing not only offers improved real-time performance, but also ensures data privacy, as all information remains on the local devices. Therefore, edge processing can potentially be a more suitable solution for implementing real-time MA removal in fNIRS.

Herein, we propose a DAE network, trained on data augmented with simulated MAs, and deployed on an STM32 microcontroller for real-time multichannel MA removal in fNIRS.

II. Method

The methodological framework of this study, illustrated in Fig. 1, outlines the key steps involved in training, optimizing, and deploying the deep learning model for MA removal on an STM32 microcontroller. First, during the model training phase, multichannel fNIRS data was collected and constructed into a dataset for training the DAE model. Once trained, the model was converted to a format compatible with edge devices, enabling deployment on the edge microcontroller. In the end, after the hardware deployment of the model, its capability for multichannel real-time processing was evaluated using testing dataset.

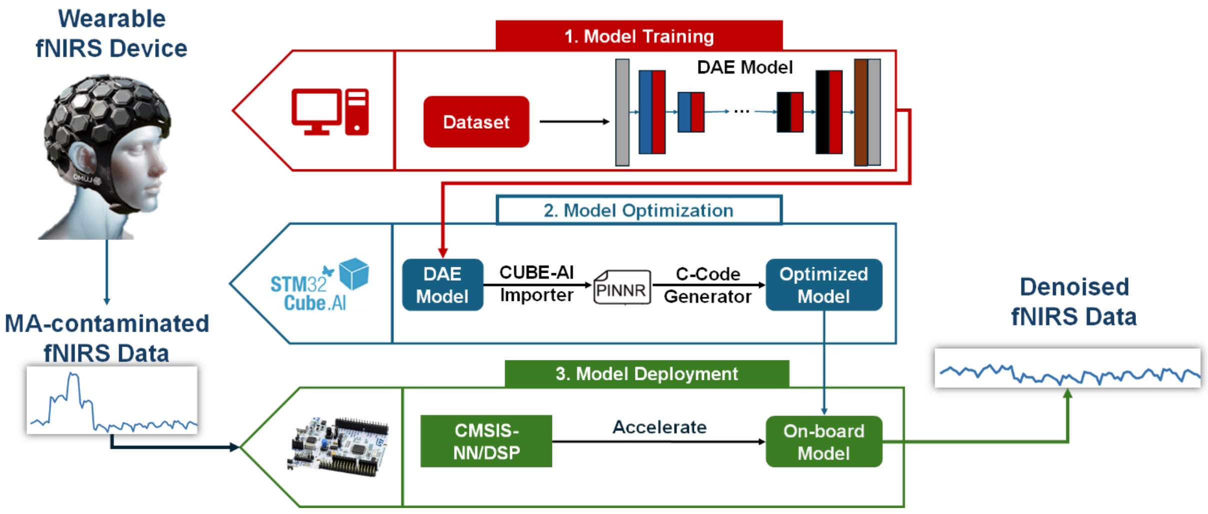

Fig. 1. Workflow of the proposed DAE model for STM32 microcontroller implementation: training, optimization, and on-board deployment.

A. Data Collection and Dataset Construction

The dataset was collected from 9 participants using a high-density fNIRS device (LUMO—Gowerlabs Ltd., UK) with approximately 3000 high quality channels at two wavelengths (735 and 850 nm) and a sampling rate of 6.67 Hz [15]. Each participant contributed approximately 40 minutes of data, during which they were instructed to remain still to minimize the MAs. Ethical approval for all experimental procedures was granted by the Committee for Research Ethics at UCL, London, under Application No 6860.017, 6860.018, and 17599.002.

In the preprocessing phase, several steps were taken to prepare the data. First, the raw intensity data collected from subjects was converted to optical density (OD) using the equation:

$$OD = -ln\left(\frac{I(t)}{\bar{I}}\right) \tag{1}$$

where *I(t)* is the raw light intensity at time *t*, $\bar{I}$ is the average intensity across the channel during the entire experiment.

Next, to ensure data quality, channels were pruned using the *enPruneChannel* function from HomER2 [16], which filters out channels based on their power range and signal-to-noise ratio. Channels were identified as active if their signal-to-noise ratio was above 12.5 [17].

Following channel pruning, each channel was normalized using z-score normalization [18], which accounts for both the mean and standard deviation, improving robustness against the large variability often observed in fNIRS data:

$$OD_{norm} = \frac{OD - mean(OD)}{std(OD)} \tag{2}$$

where OD_{norm} represents normalized *OD*, *mean(OD)* and *std(OD)* stand for the mean and standard deviation value of the *OD*, respectively.

After preprocessing, data from 7 participants were used to construct the window-based dataset, while data from the remaining 2 participants were used to create the continuous dataset. Since the DAE model in this study is designed to process real-time signals using a window-based strategy, the window-based dataset was used to train the model, whereas the continuous dataset was used to simulate real time data transfer.

To generate the window-based dataset, the signals were then segmented into 15-second windows (100 data points) with 50% overlap. One or two simulated MAs were randomly introduced as spikes in each window. This process resulted in paired segments of MA-contaminated and clean data, yielding a total of 2,133,664 segment pairs. The dataset was subsequently divided into a training set (80%), validation set (20%). For the continuous dataset, no segmentation was applied, and randomly generated MAs were introduced into each channel. This dataset was used to further evaluate the model's performance in processing continuous fNIRS signals.

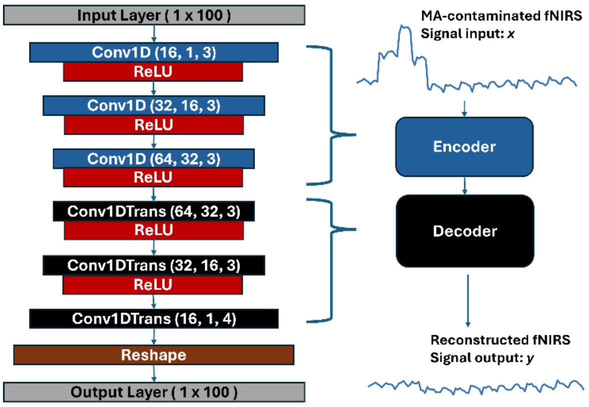

Fig. 2. Proposed DAE architecture for real-time MA removal in fNIRS.

B. DAE Architecture & Training

The architecture of the proposed DAE moder is depicted in Fig. 2. Previous work with similar architectures have been successfully applied for tasks such as image denoising and speech signal enhancement [19]. In our design, the encoder compresses the input data points into a low-dimensional latent space, while the decoder reconstructs the signal, attenuating artifact-related components. The input size is 1 × 100, and both the encoder and decoder consist of three convolutional layers with ReLU activations, except that the final convolutional layer does not have an activation layer. Due to the reduction in spatial dimensions during convolutional operations, rounding can lead to dimensional mismatches between the encoder's output and the decoder's input. To address this, a reshape layer is introduced after the final convolutional layer, ensuring that the output matches the original input size.

The loss function was set as the mean squared error (MSE) between the reconstructed fNIRS, y, and clean fNIRS, $\tilde{x}$, using (3) where N represents the segment length in samples and i indexes the data points in the window. The DAE was trained to minimize the loss function, mapping the MA-contaminated signal to the clean ones.

$$MSE = \frac{1}{N}\Sigma_{i=1}^{N}(\widetilde{x_i} - y_i)^2 \qquad (3)$$

The DAE network was developed using the TensorFlow Machine Learning library in Python 3.9 [20]. It was trained for 10 epochs with a learning rate at 0.001 with an Adam optimizer. The trained DAE model was tested offline for performance validation and then deployed on the STM32 microcontroller for multichannel real-time processing.

C. *TF-Lite Deployment*

After training, the DAE model was converted and compressed into a TF-Lite file, a commonly used format for deploying AI models on edge devices. The model was then imported into STM32CubeIDE [21] with X-Cube-AI extension [22], a toolbox developed by *ST Microelectronics* to facilitate the deployment of neural networks onto the STM32 microcontrollers. As shown in Fig. 1, the TF-Lite file was imported and translated into a platform-independent neural network representation (PINNR), which serve as an intermediate representation that standardize the uploaded deep learning model, making it potentially suitable to be further applied across different hardware platforms in the future. Using PINNR, the C-code generator produced an optimized STM32 project for deployment on the STM32 micro-controller (NUCLEO F401RE, ARM 32-bit Cortex-M4 CPU).

Following the successful deployment, the computational performance metrics, including CPU usage, memory usage and inference time, were measured using the default testing program generated by X-Cube-AI. Based on the measured inference time, the theoretical maximum number of channels that can be processed in real-time was determined. This was calculated by taking the ratio of available processing time to the inference time per channel.

D. *Peformance Evaluation of MA Removal*

The performance of the model was compared to a wavelet-based algorithm, with the wavelet implementation sourced from HomER2. Additionally, the performance of the model on both the benchtop computer (Intel Core i9, 32GB RAM) and STM32 microcontroller were evaluated to compare if there was any degradation in performance after deployment on the edge microcontroller. An UART protocol was utilised to facilitate data transfer between the computer and the edge microcontroller. The testing dataset was transmitted to the microcontroller using a 15-seconds of sliding window with 50% overlap. Normalization was applied using the mean and standard deviation from the previously processed window. The quality of the processed signals was then assessed using MSE and Correlation Coefficient (CC).

III. RESULTS & DISCUSSION

A. *TF-Lite deployment*

The computational performance of the trained DAE model on the STM32 microcontroller (NUCLEO F401RE, ARM 32-bit Cortex-M4 CPU) is shown in the Table I. The model requires an average of 3195675 CPU cycles per inference, with a low CPU workload of 3%. Regarding to complexity, the model performs a total of 371,329 multiply-accumulate operations. In terms of memory usage, the model occupies 536 bytes of stack memory and does not utilize any heap memory. The minimal CPU and memory usages of the model demonstrate that the STM32 microcontroller still holds sufficient capacity for any extra tasks or increased model complexity when needed.

The average inference time is 38.043 ms, with the inference time in the kernel taking 38.003 ms, this is sufficient to support real-time MA removal in fNIRS. For an fNIRS device with sampling frequency of 6.67 Hz, the available processing time is 0.15 s, allowing the model to process up to 3 channels simultaneously. By decreasing the overlap, the number of the parallel-processed fNIRS channels correspondingly increases, thought with a slight trade-off in processing delay. For example, for 15-seconds processing window with 50% overlapping, the available processing time extends to 7.5 s, enabling the deployed model to simultaneously process up to 197 fNIRS channels in parallel. However, this study does not assess the theoretical maximum number of channels constrained by RAM and flash memory. If memory limitations impact processing efficiency, deploying the platform on a microcontroller with larger memory capacity could effectively mitigate this issue.

TABLE I. COMPUTATIONAL PERFORMANCE OF DAE ON THE STM32 MICROCONTROLLER

Metric	***Value***
Average CPU Cycles	3,195,675
Multiply-Accumulate Operations	371,329
CPU Workload	3 %
RAM (KB)	10.25
Flash (KB)	61.07
Used Stack (bytes)	536
Used Heap (bytes)	0
Average Inference Time (ms)	38.043
Inference Time in Kernel (ms)	38.003
Inference Time in User (ms)	0.025

B. *Peformance Evaluation of MA Removal*

The performance of MA removal between wavelet-based method and the proposed DAE model, that evaluated on both the benchtop computer and STM32 microcontroller, is presented in Table II. Fig. 3, a visual representation of the data from Table II, illustrates that the proposed DAE model achieved significantly lower MSE and higher CC compared to the wavelet method. In particular, the lower standard deviation of the MSE and CC indicates that the proposed DAE model can offer more accurate and stable performance than the wavelet-based approach. Furthermore, the nearly identical MSE and CC values between the benchtop computer and STM32 microcontroller for DAE model indicate that the DAE has been smoothly deployed on the microcontroller without any degradation in performance.

The comparison of MA removal algorithms applied to a sample from the testing dataset is shown in Fig. 4. The MA-contaminated signal contains multiple MA spikes, respectively denoised by the DAE and wavelet methods.

However, the signal reconstructed by the DAE is notably more similar to the ground truth clean signal. Importantly, in periods unaffected by MA, the model did not introduce any distortion, and preserved the physiological signal such as heart rate and respiratory rate. Previous DAE models often inadvertently filtered out both MAs and physiological signals [7, [19]. However, in certain applications, such as systemic physiology augmented functional near-infrared spectroscopy [22] and multimodal BCIs [6], the preserving physiological signals is essential for capturing comprehensive and complementary brain-physiological activities, widening the practical applications of fNIRS.

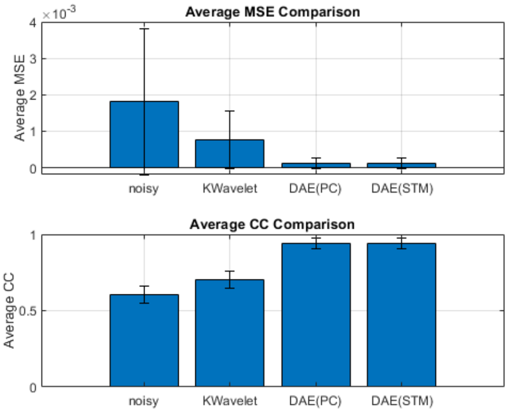

Fig. 3. Comparisions of average MSE and CC of MA removal using both wavelet and DAE, on the benchtop computer and STM32 microcontroller respectively.

TABLE II. PERFORMANCE COMPARISON OF DIFFERENT APPROACHES OF MA REMOVAL

	MSE	*CC*
STM DAE	0.0001 ± 0.0001	0.9402 ± 0.0349
PC DAE	0.0001 ± 0.0001	0.9402 ± 0.0349
Wavelet	0.0008 ± 0.0008	0.7015 ± 0.0562

In this study, only simulated MAs used because the training approach and evaluation metrics relied on the availability of clean signals as ground truth. Incorporating real MAs during data collection would make it challenging to obtain corresponding clean signals for training and performance evaluation. Therefore, future work will focus on developing appropriate methodologies for assessing the denoising performance of the proposed platform on signals with real-world MAs. Upon validation, the proposed processing platform will be integrated into an fNIRS device and evaluated in real-world settings to verify its performance in practical applications.

IV. CONCLUSION

In this work, the proposed DAE model has been deployed on an STM32 microcontroller with encouraging performance for multichannel real-time MA removal of fNIRS signals at the edge. The approach outperformed traditional wavelet-based methods. The proposed model required minimal computational resource, making it suitable for deployment across various edge platforms when needed. With an inference time short enough to simultaneously process up to 197 channels of fNIRS data, the model proved its parallel real-time processing capability for fNIRS systems with a relatively large channel counts. Additionally, the successful deployment of the model on an edge microcontroller without performance degradation underscores its potential for edge processing in fNIRS systems with a wearable form factor. In particular, the results of this study demonstrate that deep learning models, such as the DAE utilized here, can be deployed on edge devices without performance degradation while achieving promising efficacy in multichannel real-time MA removal of fNIRS. This DAE-enabled edge device is expected to be integrated into existing fNIRS systems in the future as a self-contained processing unit for an initial attempt of the development of a new-generation of AI-empowered wearable high-density fNIRS technology. In short, this work enables more accurate and efficient real-time processing at the edge, thereby enhancing the usability of wearable multichannel fNIRS technologies in diverse, real-world applications in wider environments, such as real-time BCI, online neurofeedback, personalized healthcare, and beyond.

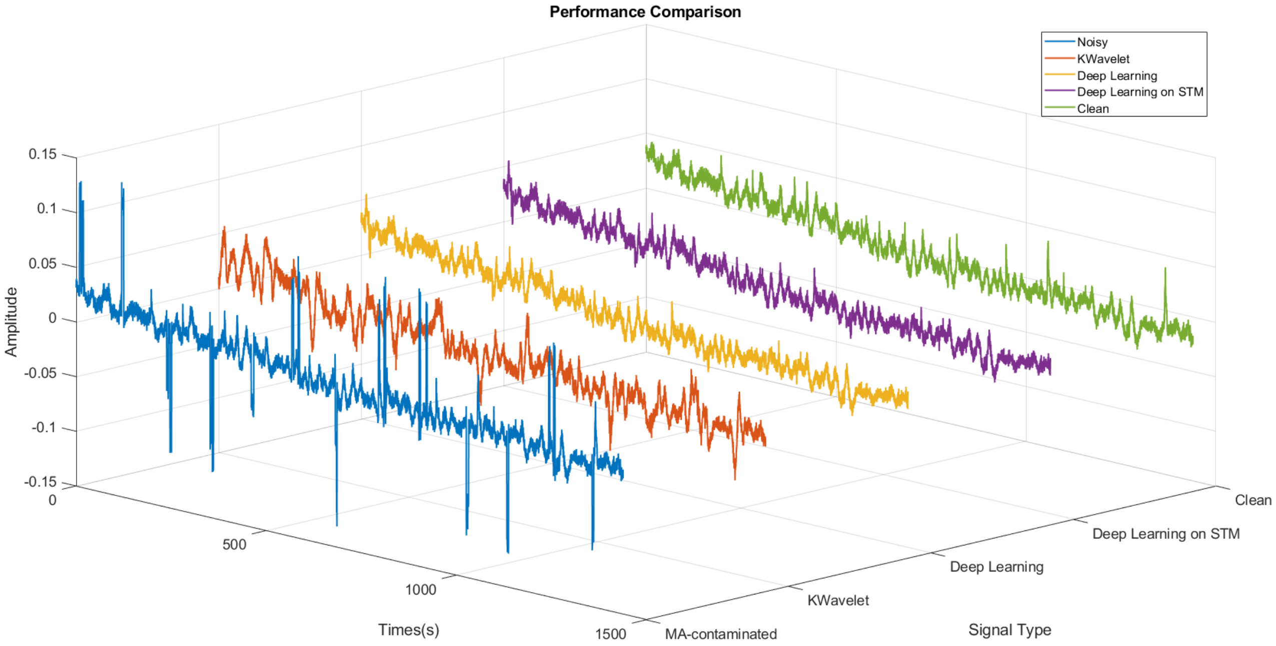

Fig. 4. Comparison of typical MA removal algorithms over a sample from the testing dataset.

References

[1] M. J. Khan and K. S. Hong, 'Hybrid EEG-FNIRS-based Eight-Command Decoding for BCI: Application to Quadcopter Control', *Front. Neurorobot.*, vol. 11, no. FEB, 2017, doi: 10.3389/fnbot.2017.00006.

[2] R. Zimmermann, L. Marchal-Crespo, J. Edelmann, O. Lambercy, M. Fluet, R. Riener, M. Wolf, and R. Gassert, 'Detection of Motor Execution using a Hybrid fNIRS-Biosignal BCI: A Feasibility Study', *J. Neuroeng. Rehabil.*, vol. 10, no. 1, pp. 1–15, 2013, doi: 10.1186/1743-0003-10-4.

[3] Q. Yang and S. Ge, 'Classification of a Single Channels fNIRS Signal for a Brain Computer Interface', *Proc. - 2014 7th Int. Conf. Biomed. Eng. Informatics, BMEI 2014*, no. Bmei 2014, pp. 46–50, 2014, doi: 10.1109/BMEI.2014.7002740.

[4] N. Naseer and K. S. Hong, 'fNIRS-based Brain-Computer Interfaces: A Review', *Front. Hum. Neurosci.*, vol. 9, no. JAN, pp. 1–15, 2015, doi: 10.3389/fnhum.2015.00003.

[5] Y. Xia, E. M. Frijia, R. Loureiro, R. J. Cooper, and H. Zhao, 'An FPGA-based, Multi-Channel, Real-Time, Motion Artifact Detection Technique for fNIRS/DOT Systems', *Proc. - IEEE Int. Symp. Circuits Syst.*, pp. 1–5, 2024, doi: 10.1109/ISCAS58744.2024.10558489.

[6] J. Chen, Y. Xia, X. Zhou, E. E. Vidal-Rosas, A. Thomas, R. Luoreiro, R. J. Cooper, T. Carlson, and H. Zhao, 'fNIRS-EEG BCIs for Motor Rehabilitation: A Review', *Bioengineering*, vol. 10, no. 12, p. 1393, Dec. 2023, doi: 10.3390/bioengineering10121393.

[7] Y. Gao, H. Chao, L. Cavuoto, P. Yan, U. Kruger, J. Norfleet, B. Makled, S. Schwaitzberg, S. De, and X. Intes, 'Deep Learning-based Motion Artifact Removal in Functional Near-Infrared Spectroscopy', *Neurophotonics*, vol. 9, no. 04, 2022, doi: 10.1117/1.nph.9.4.041406.

[8] B. Molavi and G. A. Dumont, 'Wavelet-based Motion Artifact Removal for Functional Near-Infrared Spectroscopy', *Physiol. Meas.*, vol. 33, no. 2, pp. 259–270, 2012, doi: 10.1088/0967-3334/33/2/259.

[9] S. Brigadoi, L. Ceccherini, S. Cutini, F. Scarpa, P. Scatturin, J. Selb, L. Gagnon, D. Boas, and R. J. Cooper, 'Motion Artifacts in Functional Near-Infrared Spectroscopy: A Comparison of Motion Correction Techniques Applied to Real Cognitive Data', *Neuroimage*, vol. 85, pp. 181–191, 2014, doi: 10.1016/j.neuroimage.2013.04.082.

[10] Y. Zhao, H. Luo, J. Chen, R. Loureiro, S. Yang, and H. Zhao, 'Learning-based Motion Artifacts Processing in fNIRS: A Mini Review', *Front. Neurosci.*, vol. 17, p. 1280590, 2023, doi: 10.3389/fnins.2023.1280590.

[11] J. W. Barker, A. L. Rosso, P. J. Sparto, and T. J. Huppert, 'Correction of Motion Artifacts and Serial Correlations for Real-Time Functional Near-Infrared Spectroscopy', *Neurophotonics*, vol. 3, no. 3, p. 031410, 2016, doi: 10.1117/1.nph.3.3.031410.

[12] R. Huang, K. S. Hong, S. C. Bao, and F. Gao, 'Real-time motion artifact suppression using convolution neural networks with penalty in fNIRS', *Front. Neurosci.*, vol. 18, no. August, pp. 1–15, 2024, doi: 10.3389/fnins.2024.1432138.

[13] E. Waks, 'Advancing fNIRS Neuroimaging through Synthetic Data Generation and Machine Learning Applications', pp. 1–21, 2024, [Online]. Available: http://arxiv.org/abs/2405.11242

[14] D. O'Keeffe, T. Salonidis, and P. Pietzuch, 'Frontier: Resilient Edge Processing for the Internet of Things', *Proc. VLDB Endow.*, vol. 11, no. 10, pp. 1178–1191, 2018, doi: 10.14778/3231751.3231767.

[15] 'LUMO--Gowerlabs'. Accessed: Oct. 03, 2024. [Online]. Available: https://www.gowerlabs.co.uk/lumo

[16] 'HOMER2'. Accessed: Oct. 19, 2024. [Online]. Available: https://homer-fnirs.org/

[17] E. E. Vidal-Rosas, H. Zhao, R. W. Nixon-Hill, G. Smith, L. Dunne, S. Powell, R. J. Cooper, and N. L. Everdell, 'Evaluating a new generation of wearable high-density diffuse optical tomography (HD-DOT) technology via retinotopic mapping in the adult brain', *Opt. InfoBase Conf. Pap.*, vol. 8, no. 2, pp. 1–24, 2021, doi: 10.1117/12.2615354.

[18] I. G. and Y. B. and A. Courville, *Deep Learning*, vol. 29, no. 7553. 2016. Accessed: Oct. 24, 2024. [Online]. Available: http://deeplearning.net/

[19] L. Xing and A. J. Casson, 'Deep Autoencoder for Real-time Single-channel EEG Cleaning and its Smartphone Implementation using TensorFlow Lite with Hardware/software Acceleration', *IEEE Trans. Biomed. Eng.*, 2024, doi: 10.1109/TBME.2024.3408331.

[20] 'Welcome to Python.org'. Accessed: Oct. 12, 2024. [Online]. Available: https://www.python.org/

[21] 'STM32CubeIDE - Integrated Development Environment for STM32 - STMicroelectronics'. Accessed: Oct. 27, 2024. [Online]. Available: https://www.st.com/en/development-tools/stm32cubeide.html

[22] 'X-CUBE-AI - AI Eexpansion Pack for STM32CubeMX - STMicroelectronics'. Accessed: Oct. 23, 2024. [Online]. Available: https://www.st.com/en/embedded-software/x-cube-ai.html