Deep Autoencoder for Real-Time Single-Channel EEG Cleaning and Its Smartphone Implementation Using TensorFlow Lite With Hardware/Software Acceleration

Le Xing, *Student Member, IEEE*, and Alexander J. Casson, *Senior Member, IEEE*

Abstract*—*Objective: **To remove signal contamination in electroencephalogram (EEG) traces coming from ocular, motion, and muscular artifacts which degrade signal quality. To do this in real-time, with low computational overhead, on a mobile platform in a channel count independent manner to enable portable Brain-Computer Interface (BCI) applications.** ***Methods:*** **We propose a Deep AutoEncoder (DAE) neural network for single-channel EEG artifact removal, and implement it on a smartphone via TensorFlow Lite. Delegate based acceleration is employed to allow real-time, low computational resource operation. Artifact removal performance is quantified by comparing corrupted and ground-truth clean EEG data from public datasets for a range of artifact types. The on-phone computational resources required are also measured when processing pre-saved data.** ***Results:*** **DAE cleaned EEG shows high correlations with ground-truth clean EEG, with average Correlation Coefficients of 0.96, 0.85, 0.70 and 0.79 for clean EEG reconstruction, and EOG, motion, and EMG artifact removal respectively. On-smartphone tests show the model processes a 4 s EEG window within 5 ms, substantially outperforming a comparison FastICA artifact removal algorithm.** ***Conclusion:*** **The proposed DAE model shows effectiveness in single-channel EEG artifact removal. This is the first demonstration of a low-computational resource deep learning model for mobile EEG in smartphones with hardware/software acceleration.** ***Significance:*** **This work enables portable BCIs which require low latency real-time artifact removal, and potentially operation with a small number of EEG channels for wearability. It makes use of the artificial intelligence accelerators found in modern smartphones to improve computational performance compared to previous artifact removal approaches.**

***Index Terms*—Deep learning, convolutional autoencoder, EEG artifact removal, smartphone, tensorflow lite.**

I. Introduction

ELECTROENCEPHALOGRAPHY (EEG) is a widely used non-invasive brain monitoring technique that allows neuronal electrical activities to be recorded from the scalp with excellent time resolution. It has been widely used in a range of applications from healthcare monitoring to many Brain-Computer Interfaces (BCIs) [1]. There is currently an increasing trend towards collecting EEG out-of-the-lab and out-of-the-clinic, in the more naturalistic settings found in people's daily life [2], [3], [4]. Remote monitoring can help increase access to EEG, allow longer term monitoring, and ensure users are presented with the same stimuli as they experience in their daily life [3].

To allow this portable monitoring, in recent years there has been a wide range of research focused on creating more user-friendly electrodes and electronics for more comfortable and wearable EEG [4], [5], [6], [7]. Reviews are given in [6], [8], [9], [10]. In wearable applications, typically (although not always), a smaller number of electrodes are used in order to decrease the set up time and improve comfort. An important direction in wearable EEG is in smartphone based EEG collection, with this allowing data streaming to a unit that users routinely keep with them. There are a number of manufacturers providing EEG units that connect with mobile phones, for example [11], [12], [13], and an increasing number of examples of on-phone processing of EEG data in real-time to present information to the user in neurofeedback and other applications [14], [15].

However, the EEG signal has a very low amplitude, at the micro-volt level, due to the attenuation when the EEG signal passes through the skull, tissue, and recording electrodes [16]. As a result, EEG signals are very vulnerable to a number of contamination sources, including physiologically originated artifacts such as Electrooculogram (EOG), Electromyogram (EMG), Electrocardiogram (ECG), and similar, and non-physiological sources such as electromagnetic interference and motion-induced artifacts, which lead to the real brain activity information being obscured [17]. This is particularly true for out-of-the-lab EEG data collection, as naturalistic settings potentially contain many sources of interference, and

Manuscript received 16 August 2023; revised 4 April 2024; accepted 24 May 2024. Date of publication 3 June 2024; date of current version 22 October 2024. *(Corresponding author: Alexander J. Casson.)*

Le Xing is with the Department of Electrical and Electronic Engineering, University of Manchester, U.K.

Alexander J. Casson is with the Department of Electrical and Electronic Engineering, University of Manchester, M13 9PL Manchester, U.K., and also with the Henry Royce Institute for Advanced Materials, M13 9PL Manchester, U.K. (e-mail: alex.casson@manchester.ac.uk).

Additional data supporting this publication is available at https://doi.org/10.48420/23093759.v2.

This article has supplementary downloadable material available at https://doi.org/10.1109/TBME.2024.3408331, provided by the authors.

This article has supplementary downloadable material available at https://doi.org/10.1109/TBME.2024.3408331, provided by the authors.

Digital Object Identifier 10.1109/TBME.2024.3408331

more motion, compared to lab experiments in a shielded room where participants are usually asked to remain stationary.

To overcome these sources of interference there are a wide number of algorithms available currently for EEG artifact removal. These have been systematically reviewed in [18], [19], [20], [21], [22]. Conventional methods, such as digital filtering, regression, frequency decomposition (including the Wavelet Transform (WT), and Empirical Mode Decomposition (EMD)) have limited efficiency when handling different artifacts [20]. Blind Source Separation (BSS) based methods, particularly Independent Component Analysis (ICA) show good performance on handling different artifacts, and have been popularly adopted in much research, becoming the current *gold standard* [18], [21]. Nevertheless, ICA-based algorithms still have several drawbacks in mobile EEG applications. Firstly, ICA approaches require multi-channel EEG data for separating out different signal sources, and ICA performance improves with an increasing number of channels [20], [23]. Secondly, in many cases the ICA algorithm is only a semi-automatic approach which requires artifact components in the source domain to be recognized manually by specialists for removal before signal reconstruction. Lastly, the independent components are separated via an iterative process with high computational costs, which is likely to cause processing latency and high power consumption when running on a portable platform.

Investigations are therefore ongoing into alternative artifact removal approaches which can be channel count independent, fully automated, and low computational complexity for real-time portable processing. In particular, Deep Learning (DL) based methods have received substantial attention recently. Various DL-based algorithms have been proposed for EEG artifact removal, with better performance shown compared with some traditional algorithms [24], [25], [26], [27], [28], [29], [30], [31], [32], [33], [34], [35], [36].

Amongst these DL based algorithms, Autoencoder-based neural networks have shown very good performance [22], [27], [33]. However, challenges still remain in three different aspects. Firstly, many existing autoencoder models, such as [30], [31], [33], investigated EOG and EMG artifact types, which are relatively easier to remove due to their clear time-frequency features. In comparison, motion artifacts are harder to remove, and very dominant in mobile EEG. Secondly, although [27], [35] proposed more comprehensive DL models, considering multiple types of artifact, their signal processing pipelines are relatively complicated with multiple networks combined for artifact detection and removal at the same time. This increases the computational complexity. Lastly, most of the studies only focused on processing contaminated EEG segments while neglecting a discussion on processing clean EEG segments. During an EEG experiment, periods of both artifact corrupted and non-corrupted data may be present, and when cleaning EEG data sections it is important to not introduce distortions into otherwise uncorrupted EEG sections. An extra artifact detection function can be added to separate out the contaminated segments and clean segments, and the artifact removal algorithm applied only to contaminated segments. However, autoencoder neural networks by intention are good at copying their input to their output as an intrinsic advantage, with this widely applied in image processing and data compression [37]. Therefore, the autoencoder should automatically reconstruct the signal if the input EEG is clean, and output the corrected signal when the input is contaminated, without any additional operations for segment differentiation. This will again reduce the computational complexity. However, this has not been verified in the works discussed above which have focused on processing corrupted EEG data segments.

Further, none of the aforementioned DL models go beyond offline algorithm validation on a standard computer. In principle, DL based algorithms are ideally suited to portable, smartphone based, real-time EEG data processing due to smartphone hardware evolution and breakthroughs in hardware development [38], [39]. A wide range of smartphone manufacturers have progressed their chipsets by integrating more powerful processors (central processing units (CPUs), graphics processing units (GPUs), digital signal processors (DSPs) together with neural processing units (NPUs)) targeted at running artificial intelligence applications. These therefore allow the hardware and software acceleration, and running, of DL models on mobile devices with faster computation times and lower power consumption [38]. However, to our knowledge smartphone hardware/software acceleration of DL EEG artifact removal algorithms has yet to be exploited.

Therefore, this paper proposes a Deep Autoencoder (DAE) algorithm for channel count independent (single-channel) EEG signal cleaning, targeting EOG, EMG and motion artifact removal, including a consideration of distortions introduced to otherwise clean data segments. Further, the algorithm is deployed on Android smartphones by using TensorFlow Lite with delegate based acceleration for providing a low power and rapid deep learning solution for mobile EEG signal processing. The on-smartphone implementation is amplifier independent, and so we only consider the real-time EEG processing on Android, not data streaming from a connected amplifier. For algorithm validation, the gold standard FastICA algorithm is also implemented on a smartphone for performance comparisons with the proposed DAE in terms of computation time, power consumption and memory usage. As a result, we present the first demonstration of on-smartphone DL models for EEG cleaning and open more opportunities in the practical applications of wearable EEG technology.

The remainder of this article is structured as follows. Section II presents the EEG datasets used, the design of the DAE network, the on-smartphone implementation, and the validation methods employed. Results are presented in Section III. These are then discussed in Section IV, together with a consideration of the limitations of this work.

II. Methods

A. Data Selection, Preparation, and Pre-Processing

Three different datasets, containing different types of EEG artifact were used, with the data processing pipeline shown in Fig. 1. The EEG datasets used are all publicly available, with the most important dataset selection criteria being that the dataset should contain both ground-truth clean and artifact contaminated

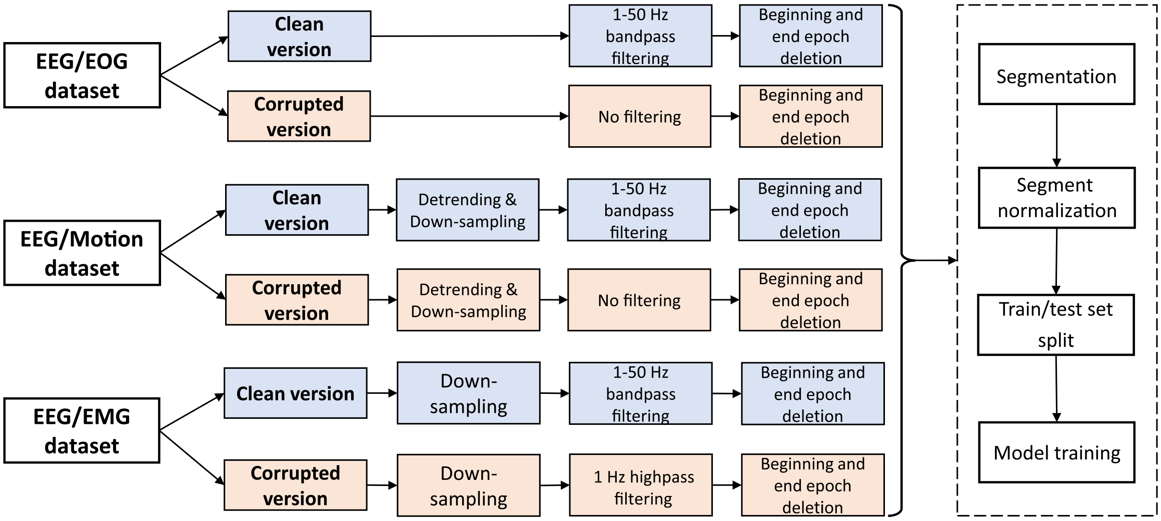
EEG/EOG dataset
Clean version
1-50 Hz bandpass filtering
Beginning and end epoch deletion
Corrupted version
No filtering
Beginning and end epoch deletion
EEG/Motion dataset
Clean version
Detrending & Down-sampling
1-50 Hz bandpass filtering
Beginning and end epoch deletion
Corrupted version
Detrending & Down-sampling
No filtering
Beginning and end epoch deletion
EEG/EMG dataset
Clean version
Down-sampling
1-50 Hz bandpass filtering
Beginning and end epoch deletion
Corrupted version
Down-sampling
1 Hz highpass filtering
Beginning and end epoch deletion
Segmentation
Segment normalization
Train/test set split
Model training

Fig. 1. Data processing pipeline of different EEG datasets before model training. Note that normalization is applied independently on a segment-by-segment basis.

versions of the same EEG recordings for the Deep Autoencoder (DAE) model training and evaluation. In total, 19 hours of EEG recording were used for model development and evaluation. Note that all filters applied are bidirectional, using the Python scipy `filtfilt` function.

1) EEG/EOG Dataset: For EOG contaminated data, the semi-simulated dataset in [40] was used. It contains 54 pairs of clean and corrupted EEG recordings from 27 participants, and each recording has 19 channels, with 2 reference electrodes at the mastoids, and a 200 Hz sampling frequency. In this dataset, the clean EEG signals were obtained from stationary participants and inspected by experts (in the original research team). The corrupted data were created from these clean records by linearly superimposing vertical and horizontal EOG epochs into the near-frontal EEG channels [40]. Mains noise was removed by the origianl research team.

In this study, we use 5 near-frontal channels, FP1, FP2, Fz, F7, and F8 channels. The clean data were first bandpass filtered between 1 and 50 Hz by using a second-order Butterworth filter, and 2 s at the beginning and end of the recording were deleted to eliminate edge effects. In parallel, 2 s epochs at the beginning and end were also deleted in the contaminated data version, but without any filtering. Our DAE is trained to operate on the raw corrupted version of the EEG data rather than a pre-filtered version of it.

2) EEG/motion Dataset: For motion artifact contaminated data, the dataset in [41] from PhysioNet [42], [43] was used. It contains 23 pairs of clean and corrupted EEG recordings, and each recording has 2 channels, with a 2048 Hz sampling frequency. The data was recorded in the EU, with a 50 Hz mains frequency, and no dedicated notch filtering was applied by the original research team.

In the data preparation stage, the raw data was first detrended and downsampled to 200 Hz to match the EEG/EOG dataset described above. The same bandpass filtering was applied and 5 s of data at the start/end of each record removed, with this slightly longer cut reflecting the longer duration of each record in this set compared to the EEG/EOG dataset.

For generating motion data with a clean reference, the clean and corrupted EEG signals were recorded simultaneously by two independent sensors placed close together. One was physically disturbed to produce motion interference, while the proximate sensor was undisturbed, giving a simultaneous clean recording from a very nearby location. On our visual inspection of the dataset we noted that the clean version EEG data commonly had low frequency drift, and so the data was linearly detrended before processing further.

3) EEG/EMG Dataset: For EMG contaminated data, the dataset in [24] was used. It provides a dataset containing clean EEG, and EOG and EMG artifact segments intended for Deep Learning algorithm development. Mains noise was removed by the origianl research team. This database contains 4514 clean EEG segments, 3400 EOG artifact segments, and 5598 EMG artifact segments, and each segment has a 2 s duration. We concatenated these segments to construct a longer duration recording which we could then re-segment (see Section II-B), noting that this would introduce minor discontinuities at each 2 s point to be removed by the filtering described below. The sampling frequency provided in the dataset was 256 Hz, with single channel data generated using an initial 64 channel dataset from [44]. In our data preparation stage, the concatenated raw data was first detrended and downsampled to 200 Hz. Afterwards, a 1–50 Hz bandpass filter was applied to the clean version data which could filter out the discontinuities introduced by concatenation and potentially eliminate distortions during the model training, and 5 s of data at the start/end of each record was removed. The corrupted version data was processed with only the 1 Hz cut-off filter to keep high-frequency interferes to be removed by the DAE.

Following the procedure in [24] we created 50 different pairs of clean/corrupted EEG signals, each with a duration of 5 minutes. To synthesize corrupted data, we randomly picked

EMG periods from the database and linearly superimposed them onto clean EEG segments. For each pair, the randomly picked EMG segments were different, but the same EMG segment could have appeared in more than one pair. The Signal-to-Noise Ratio (SNR) of the contaminated signal was controlled to be in the range -14 dB to -1 dB, based on

$$\text{SNR} = 20 \times \log \frac{RMS(\text{Clean EEG})}{RMS(\text{Corrupted EEG})} \tag{1}$$

where RMS is the Root Mean Square of the EEG amplitude in each 5 minute signal.

We created 5 minute signals, which are then segmented in Section II-B, to allow EMG artifacts that start and stop within a data segment, not only on segment boundaries.

4) Clean EEG Segment Subset: To avoid the network introducing distortions when presented with clean EEG data where no artifact removal is required, 4622 pairs of clean data were included within the dataset for model training. In these, the clean and *corrupted* data are essentially the same record, which is artifact-free. A 4 s segment of data (see Section II-B) was considered as clean when the correlation coefficient between the *corrupted* version and the ground-truth clean version was greater than 0.95, otherwise the segments were considered to contain artifacts. During the model training and testing, the clean input segments were part of each respective data set, rather than being separated as an independent set. For reporting performance, in addition to considering each of the datasets described previously, we calculate the reconstruction performance on the clean data segments in isolation, and show some specific example cases, to demonstrate the performance on this type of data.

B. Data Segmentation and Normalization

Our DAE network was developed to operate on 4 s epochs of data. The data as described in Section II-A were segmented into 4 s epochs, with this selected as the previously used segment length in [27]. 50% overlap was applied during the signal segmentation increasing the number of segments present. As a result, 3450 pairs of segments from the EEG/EOG dataset, 6215 pairs of segments from the EEG/motion dataset, and 7450 pairs of segments from the EEG/EMG dataset were obtained in total. These figures include the 4622 pairs of clean data, with some artifact free periods naturally present in each EEG recording. For training, all the segmented datasets were grouped together and split into training sets (80%, 13,692 pairs including 3708 pairs of clean input), validation sets (10%, 1711 pairs including 457 pairs of clean input) and test sets (10%, 1712 pairs including 457 pairs of clean input).

Before being passed to the network, data was normalized using the *max-min* approach, applied on a segment-by-segment basis:

$$x_i = \frac{s_i - s_{\min}}{s_{\max} - s_{\min}} \tag{2}$$

where s represents a 4 s EEG segment, $s_{\max}$ and $s_{\min}$ represent the maximum and minimum value of the signal segment s, and s_i is the i^{th} data point of EEG segment s. x is the normalized result. This rescales the data to be in the range 0 and 1, and can be applied in real-time to unseen data for the smartphone implementation. Note that the data is not de-normalized after reconstruction, and so the reconstructed EEG output signal is also in the range 0 to 1. This output is then detrended (mean removed) so that plotted signals are in the approximate range -0.5 to 0.5.

C. Deep Autoencoder Architecture

Our used Deep Autoencoder (DAE) structure is shown in Fig. 2. The autoencoder neural network was developed using the TensorFlow Machine Learning library and Keras API [45] in Python 3.9.

Let x, $\widetilde{x}$, and y denote the noramlised raw EEG input (which may or may not contain artifacts depending on the segment), ground-truth clean EEG, and reconstructed EEG output, respectively. As illustrated in Fig. 2, in the offline model training stage, the raw EEG signal, x, was inputted into the DAE network, with the ground-truth clean EEG version, $\widetilde{x}$, provided as the reference training target. The loss function was set as the Mean Squared Error (MSE) between the reconstructed EEG, y, and the output of the autoencoder and the ground-truth clean EEG, $\widetilde{x}$, using

$$L_{MSE} = \frac{1}{n}\sum_{i=1}^{n} \|\widetilde{x}_i - y_i\|_2^2 \tag{3}$$

where i represents the i^{th} data point of the segment being considered and n is the segment length in samples, $n = 800$ in this paper.

Therefore, the DAE was trained to map the corrupted signal to the clean signal while minimizing the loss function. A similar concept of denoising autoencoder networks has been used for image denoising [46] and speech signal enhancement [47]. The encoder stage is expected to extract and compress the key information of the input signal into a low-dimensional latent feature domain, and then the decoder stage learns to attenuate the artifact-related components and at the same time keep as many as possible signal-related components, to reconstruct the autoencoder output as close as possible to the reference. Our DAE contains seven 1-dimensional convolutional layers in total, and was trained for 1000 epochs with a learning rate at 0.001 with an Adam optimizer. The model was trained on a standard laptop (Intel Core i7, 16 GB RAM), requiring approximately 9 hours per run.

The trained DAE model can be used both offline for performance validation (Section II-D) and online on a smartphone for real-time data processing (Section II-E). As our training samples contain both originally clean signals and contaminated EEG segments, our DAE is expected to both remove artifacts from contaminated EEG sections, as well as pass through originally clean sections without introducing distortions. In contrast, the Recurrent Neural Network based autoencoder in [27] had a discriminator neural network prior to the EEG denoising stage, classifying the input into clean and contaminated segments and only contaminated signals were input into the denoising stage. Our trained model is designed to process all EEG segments, regardless of whether the signal epoch is originally clean or contaminated, within one neural network to reduce the complexity for online implementation.

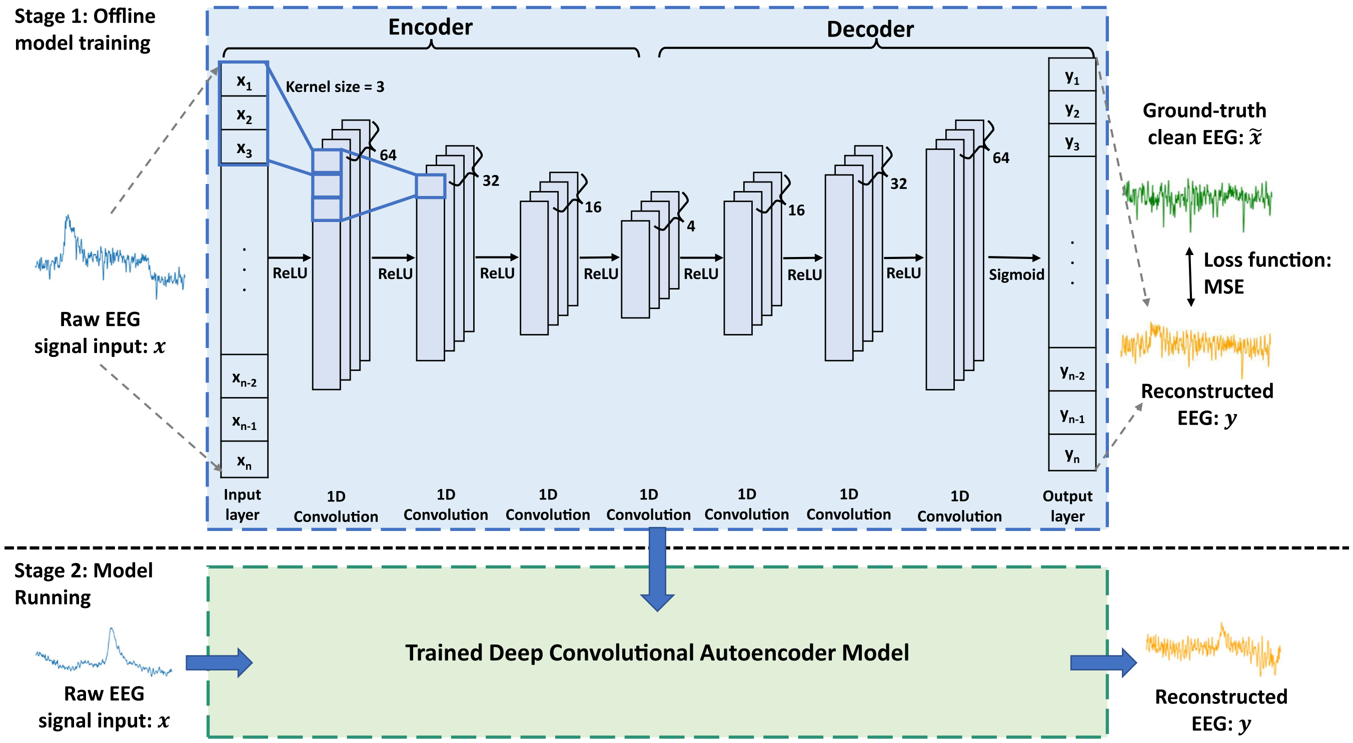

Fig. 2. Proposed deep convolutional autoencoder architecture for clean EEG reconstruction and EEG artifact removal. Stage 1 is the offline model training and Stage 2 is the model running online (in a smartphone) or offline (for validation purposes).

D. Artifact Removal Performance Evaluation

We use multiple metrics to show the performance of our approach, with the metrics previously used in EEG artifact removal validation research [24], [26], [27], [28]. The metrics are calculated on each EEG segment in the test dataset and the mean and standard deviation calculated for each different artifact type. The DAE output, y, is not further filtered before comparison.

1) Relative Root-Mean-Square Error: The Relative Root-Mean-Square Error (RRMSE) depicts the difference between the reconstructed EEG and the ground-truth clean EEG. Lower RRMSE represents better performance. In this study, RRMSE was used for both time and frequency domain comparisons between the ground-truth and reconstructed EEG signals, defined as

$$RRMSE_t = \frac{RMS(y - \widetilde{x})}{RMS(\widetilde{x})} \tag{4}$$

and

$$RRMSE_f = \frac{RMS(PSD(y) - PSD(\widetilde{x}))}{RMS(PSD(\widetilde{x}))} \tag{5}$$

where $RRMSE_t$ and $RRMSE_f$ indicate the RRMSE in the time and frequency domains respectively. PSD represents the Power Spectral Density (PSD) of each EEG segment, calculated using the Welch transform with a Hanning window (window size 200 samples, 50% overlap) and the number of FFT points was kept the same length of the segment at 800.

2) Correlation Coefficient: The Correlation Coefficient (CC) describes the similarity between the ground-truth clean EEG and reconstructed EEG in the time domain. It is calculated as

$$CC = \frac{Cov(y, \widetilde{x})}{\sqrt{Var(y)Var(\widetilde{x})}} \tag{6}$$

where Cov is the covariance and Var the variance. Higher values represent better performance.

3) Scalogram Similarity Comparison: Finally, we compare the time-frequency domain representation of the signals. The scalogram of each data segment was generated via the Continuous Wavelet Transform (CWT) with the Morlet wavelet. The normalized 2D cross-correlation function (`norm2corr` function in MATLAB) was used for calculating the similarity between the scalograms from reconstructed and ground-truth EEG. A similar method has been used previously for image matching [48], image registration [49], and scalogram comparison [50].

4) Offline Comparison Between DAE and ICA: For performance comparison between the proposed DAE and ICA as the current EEG gold standard, we used one example record (number 54) from the EEG/EOG dataset. This contains a 19 channel EEG, sufficient for applying ICA. We used the first 24 s of data (the largest integral multiple of 4 s shorter than the 27 s record length).

For our DAE model, we did the same as described above, now feeding all 19 EEG channels into the trained DAE for cleaning. Note that this record was not included in the model training and validation, and only 5 channels data from it were included for the model testing reported in the other sections of

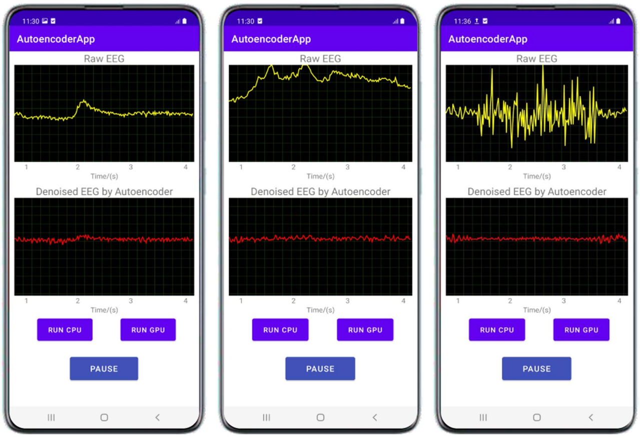

Fig. 3. Screenshot of the DAE Android App for EEG cleaning, including EOG (left), motion (middle), and EMG (right) artifact examples.

this paper. For ICA, we used the FastICA algorithm in EEGLAB and the automatic artifact component labeling and rejection, to get 19 channel ICA cleaned EEG. We then calculated the RRMSE and CC between ground-truth clean EEG and both algorithm-cleaned EEG signals for each channel, in both the time and frequency domains. Note the calculations were performed for normalized data, and the mean and standard deviations across the 19 channel data are provided.

E. TensorFlow Lite Deployment

After the offline algorithm performance evaluation, the DAE model was converted and exported as a compressed TensorFlow Lite (TF-Lite) model file (TF-Lite version 2.11), which is a commonly used framework for machine learning on mobile and edge devices [51]. Then, the DAE model was deployed onto an Android smartphone (Samsung A51, CPU: Octa-core Exynos 9611 (8 cores), and GPU: Mali-G72 MP3) and we created an EEG cleaning demo App. The application was developed by using Android Studio and the Java programming language.

For the on-smartphone algorithm demonstration, a synthetic example EEG signal with various EEG artifacts was pre-stored in the smartphone App in advance. (We do not embed data streaming from a real EEG amplifier in this current demo App.) The synthetic EEG signal lasts approximately 73 s, and contains a series of EOG, motion, and EMG artifact segments, with a sampling frequency of 200 Hz. When the App is launched, the EEG signal goes through the same pre-processing as on the computer (segmentation and normalization), and each 4 s segment is streamed into the DAE for artifact removal and reconstruction. At the same time, the corrupted data and reconstructed data are plotted in real-time by using the Canvas Java Class for visual comparison. Screenshots of the App are shown in Fig. 3 with EOG, motion, and EMG artifact removal examples. The live demo App is available at DOI: https://doi.org/10.48420/23093759.v2 in the android_app branch of the repository.

We investigated two different implementations, using the TF-Lite API to enable acceleration. After importing the TF-Lite model into the Android project, sample code is automatically generated on how to use the model. We modified the sample code to adapt to our data type by using the *ByteBuffer* Java Class for 1-Dimensional data input. In the Android *Activity* where the TF-Lite model is executing, the *Model.Options* Java Class was used to specify whether the model is about to be run on a CPU (default option) or GPU. After setup of the options, accelerator delegates can be enabled. As a result, the first model runs on the CPU with XNNPACK delegates enabled. The second model runs on the smartphone GPU with GPU delegates enabled.

In addition, we also compared the artifact cleaning performance and on-smartphone costs between the default TF-Lite model without optimization, and the optimized TensorFlow-lite model by using post-training quantization methods: (1) TF-Lite int-8 model by using dynamic range quantization; and (2) TF-Lite float-16 model by using float-16 quantization. The optimized models have reduced model sizes for computation in edge devices.

F. On-Smartphone Computational Cost Evaluation

For assessing the on-smartphone performance we measured the computational costs including the processing time, battery/power consumption, and memory usage on our Samsung A51 Android smartphone. We also implemented on-smartphone ICA to allow comparison to this gold standard approach.

To implement ICA on an Android smartphone, the open-source Statistical Machine Intelligence and Learning Engine (Smile) Java library [52] was used. We used the FastICA algorithm, operating on 32 channels of data. For comparing DAE and ICA performance we used the example EEG dataset from the EEGLAB ICA tutorial [53]. This example EEG dataset has 32 channels with a 256 Hz sampling frequency, lasting about 4 minutes. To use this dataset, we downsampled the data to 200 Hz and then imported the data to the Android phone. The FastICA method was used for signal decomposition with the following settings: 32 expected independent components, differentiable function *Kurtosis* type, ICA tolerance 0.001, and maximum iterations 1000. Note that the FastICA used here is only performing decomposition, excluding any component discarding and signal reconstruction steps.

To quantify performance we assessed the following parameters:

1) Computational Time: The time to process a 4 s segment of data was measured, for both the DAE TF-Lite model and the FastICA algorithm. Note that the computational time measured here includes the data loading time from phone memory and the algorithm execution time. More than 1000 segments were processed and the average and standard deviation of run times are reported for the DAE model, and 200 running times for FastICA. Assuming the DAE model processes data channel-by-channel, given a multi-channel recording with the matrix size of 32 × 800 (32 channel and 4 s recording duration), the averaged computational time of the DAE was multiplied by 32 for comparison, as FastICA computes 32 EEG channels simultaneously.

Different durations of EEG signal were used and the computational time was measured for each case. Here, the 32 channel

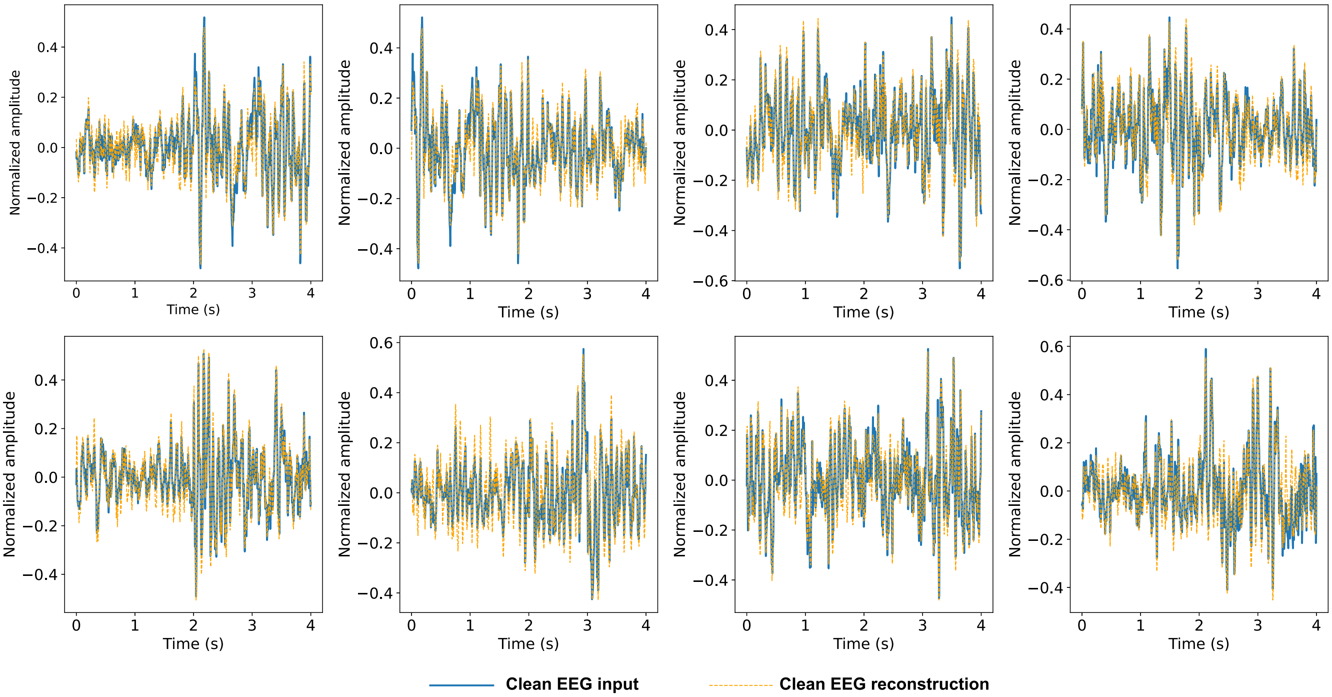

Fig. 4. Eight 4 s examples of originally clean EEG reconstruction examples. For each subplot, the clean EEG (DAE input, x), and the reconstructed EEG (DAE output, y) are displayed for comparison.

signals were loaded from the example EEG dataset for FastICA decomposition, with incremental durations at 4, 8, 12, 16, 20, 24, 28, 32, and 40 s. Again, the measures of the DAE TF-Lite model were measured for single channel data and multiplied by 32 for comparison. Note that the model was not re-trained for different segment lengths, it is the same 4 s model applied multiple times to process a larger window.

2) Battery Consumption: The battery consumption during App execution was measured using Battery Historian, a tool for inspecting battery usage on Android smartphones [54]. The DAE TF-Lite model and FastICA algorithm were set to run for 1 h while measuring the battery consumption. The experiments were done 3 times for each setup.

3) App Memory Usage: The smartphone App memory usage was measured via the Android Studio Profiler during App execution. This measurement reflects whether these algorithms are memory dependent, to gain insight into whether the App and algorithms can be affected by other Apps using the phone resources.

III. Results

A. Artifact Removal Performance

1) Clean EEG Reconstruction: Clean EEG reconstruction examples are shown in Fig. 4. A very good match is seen, with the reconstructed EEG signals keeping the time-frequency features of the input EEG segments as desired. The proposed DAE model is able to reconstruct clean EEG inputs, those which are not corrupted by artifacts, keeping the output signal close to the input EEG. The RRMSE (average ± standard deviation across all segments) between clean input EEG and reconstructed EEG output is 0.27 ± 0.07 and 0.30 ± 0.10 in the time and frequency domains respectively. The CC is 0.96 ± 0.02.

For the originally artifact-free EEG segments in the raw recording, signal reconstruction is very important. However, previous research, such as [24], [26], [27] only focused on the removal of artifacts for contaminated EEG segments, while ignoring clean EEG reconstruction. The results here show that clean EEG can be passed through with minimal distortion introduced, and so a separate discriminator to identify clean vs. corrupted EEG segments is not intrinsically required.

2) EOG Artifacts: Fig. 5 shows examples of EOG artifacts being removed, including horizontal and vertical eye movement artifacts. The large EOG peaks in the raw contaminated EEG have been effectively attenuated in the reconstructed EEG, despite some residual artifacts being present in some cases. The average $RRMSE_t$ and $RRMSE_f$ across all segments are 0.52 ± 0.10 and 0.52 ± 0.21 respectively, and the CC is 0.85 ± 0.07.

3) Motion Artifacts: Motion artifacts can be abrupt EEG signal jumps or slow signal baseline wandering due to the displacement of electrodes, and both types of motion artifact were removed by the DAE. Examples are shown in the supplementary material, Fig. S1. The $RRMSE_t$ and $RRMSE_f$ were 0.72 ± 0.11 and 0.68 ± 0.18, and CC 0.70 ± 0.11. We found from visual inspection that the EEG motion artifact dataset from [41] may still have some strong motion disturbance in the *clean* EEG versions. As a result, even the clean version recordings are not *clean enough*, containing residual motion contamination, decreasing the performance metrics for this dataset.

4) EMG Artifacts: We tested different EMG artifact cases, including both when the whole segment is contaminated by

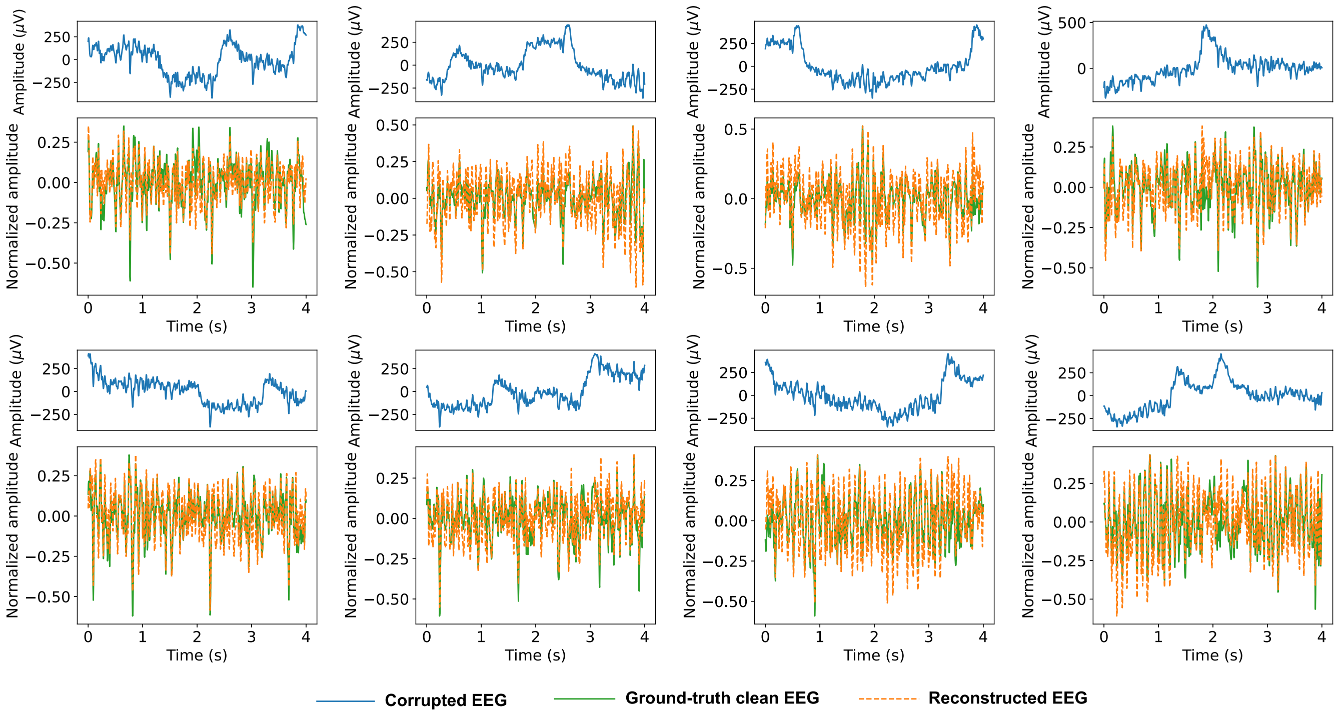

Fig. 5. Eight 4 s EOG artifact removal examples including horizontal and vertical eye movement artifacts. For each subplot, the top panel is the raw contaminated EEG segment s with the amplitude in micro-Volts (Blue), i.e., x except that the signal displayed is not normalized. The bottom panel shows the reconstructed EEG signal y (Orange), and the ground-truth clean EEG $\tilde{x}$ (Green) for comparison.

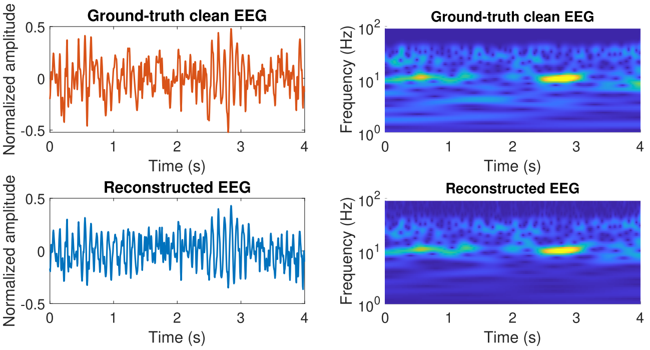

Fig. 6. Example of a pair of ground-truth clean EEG $\tilde{x}$ and reconstructed EEG traces y from the EEG/EOG artifact set and their corresponding scalograms for similarity comparison.

TABLE I
Scalogram Similarity Comparisons for Different Data Sets

Artifact type	Scalogram similarity
Clean EEG reconstruction	0.92 ± 0.03
EOG artifact removal	0.85 ± 0.08
Motion artifact removal	0.66 ± 0.14
EMG artifact removal	0.76 ± 0.13

TABLE II
Offline Artifact Removal Performance Comparison Between DAE and FastICA

Metric	DAE-cleaned	FastICA-cleaned
$RRMSE_t$	0.18 ± 0.01	0.22 ± 0.06
$RRMSE_f$	1.12 ± 0.23	0.39 ± 0.18
CC_t	0.88 ± 0.02	0.70 ± 0.14
CC_f	0.97 ± 0.01	0.93 ± 0.07

(CC_t, CC_f represents the correlation coefficients in the time and frequency domains.).

artifact and when the EMG artifact starts/stops during the segment. The DAE model is able to handle both conditions, the artifact does not need to be aligned with a segment start/end. The $RRMSE_t$ and $RRMSE_f$ were 0.60 ± 0.17 and 0.60 ± 0.19 respectively, and CC 0.79 ± 0.12. Example visulisations are given in the supplementary material, Fig. S2.

5) Scalogram Similarity Comparison: An EEG/EOG artifact removal example of the scalogram similarity comparison is shown in Fig. 6. In the scalogram of the reconstructed EEG, the main time-frequency features are clearly kept, despite some differences particularly at lower frequencies. The two bursts of EEG alpha activity (at ~10 Hz) are maintained. The similarity of the example scalograms shown is 0.91. The results across all artifact types are shown in Table I. These values have a high degree of consistency with the CC results, as well as providing good complementarity to the RRMSE measures in the time and frequency domain separately.

6) Offline Performance Comparison Between DAE and ICA: Taking one example EEG dataset from the EEG/EOG dataset for performance testing between our DAE model and FastICA, the performance metrics are shown in Table II. In the time domain, the DAE-cleaned EEG has lower RRMSE and

TABLE III
COMPARISON OF ARTIFACT REMOVAL PERFORMANCE BETWEEN DIFFERENT DEEP LEARNING ALGORITHMS

Artifact type	Metric	Gao *et al.* [26]	Zhang *et al.* [27]	Zhang *et al.* [24]	Yin *et al.* [36]	**This work (TensorFlow)**	**This work (TF-Lite)**
EOG artifacts	$RRMSE_t$	0.11	<1.00	0.27–0.68	0.34 ± 0.04	0.52 ± 0.10	0.52 ± 0.10
	$RRMSE_f$	0.09	–	0.25–0.65	–	0.52 ± 0.21	0.51 ± 0.21
	CC	0.99	>0.80	0.79–0.94	0.93 ± 0.02	0.85 ± 0.07	0.85 ± 0.07
Motion artifacts	$RRMSE_t$	0.27	<0.71	–	–	0.72 ± 0.11	0.72 ± 0.10
	$RRMSE_f$	0.28	–	–	–	0.68 ± 0.18	0.70 ± 0.18
	CC	0.96	>0.79	–	–	0.70 ± 0.11	0.70 ± 0.11
EMG artifacts	$RRMSE_t$	0.33	<0.55	0.29–1.26	0.29 ± 0.01	0.60 ± 0.17	0.60 ± 0.17
	$RRMSE_f$	0.33	–	0.26–1.30	–	0.60 ± 0.19	0.60 ± 0.18
	CC	0.94	>0.84	0.52–1.04	0.94 ± 0.004	0.79 ± 0.12	0.79 ± 0.12
Clean EEG	$RRMSE_t$	–	–	–	–	0.27 ± 0.07	0.28 ± 0.07
	$RRMSE_f$	–	–	–	–	0.30 ± 0.10	0.32 ± 0.12
	CC	–	–	–	–	0.96 ± 0.02	0.96 ± 0.02

higher CC with the ground-truth clean EEG, showing better time-domain data correlations, compared with the ICA-cleaned EEG. In the frequency domain, despite higher $RRMSE_f$ between the ground-truth and DAE-cleaned EEG, which is caused by data normalization, the higher CC_f shows better frequency domain coherence for this dataset compared with FastICA-cleaned EEG.

B. Comparison to Other Deep Learning Based Models

Table III shows a comparison of the EEG cleaning performance between our work and other similar deep learning models. Note that [26], [27] reported the same metrics as us directly, while [24] reported metrics only using boxplots rather than providing numeric results. We made use of the *GRABIT* toolbox [55] in MATLAB to extract values from the graphs. The minimal and maximal values across their 4 models were used to represent the metrics value range. For comparison with non-autoencoder based approaches, Table III also reports the performance from [36] as a recent approach based on hybrid Generative Adversarial Network (GAN)/CNN/Transformer networks (named GCTNet) which uses the same semi-simulated EOG and EMG datasets as our work.

From Table III, our proposed DAE shows a similar performance to the Gated Recurrent Unit autoencoder based pipeline proposed in [27], and the 4 different approaches (Fully-connected neural network, simple Convolutional Neural Network (CNN), complex CNN, and Recurrent Nural Network) proposed in [24]. The GCTNet in [36] shows better performance on EOG and EMG artifact removal than our method. The CNN-LSTM (Long Short-Term Memory) model reported in [26] gives the best performance, with a very high CC obtained for several artifact types, EOG (0.99), motion (0.96), and EMG (0.94), notably higher than our average values. However, this model contains 18.79 million network parameters. In contrast, our model is intended for real-time on-smartphone implementation and contains only 16,357 parameters in total. Our RRMSE values are within the range of values obtained by [24]. Our CC values in general suggest better performance than the RRMSE values, with this expected due to the normalization used giving an intrinsically small denominator (< 1) for the RRMSE calculation.

TABLE IV
TF-LITE COMPUTATIONAL COST COMPARISONS ON-SMARTPHONE 32 CHANNEL CONFIGURATION

Algorithm	DAE on CPU with XNNPack delegate	DAE on GPU with GPU delegate	FastICA (decomposition only)
Computational time (s)	0.14	0.34	1.28
Power usage (mAh/h)	0.43	29.03	89.94
Memory usage (MB)	101.9	124.4	115.5

Table III includes our results for both the TensorFlow model and the TF-Lite model using the default conversion option without optimization. These have very similar performances for artifact cleaning, indicating no substantial performance loss was caused by the conversion process. Similar results were obtained with the int-8 and float-16 TF-Lite models producing similarly close performance metrics and so for compactness these are not reported here.

C. On-Smartphone Computational Cost

Using the default TF-lite conversion settings, the single-channel EEG cleaning by the DAE TF-Lite model running on our Samsung smartphone on the CPU with XNNPack delegate requires an average of approximately 4 ms to process a 4 s EEG segment. In comparison, the GPU with GPU delegate is slower, requiring around 10 ms. Compared to the segment duration of 4 s, and an inter-sample period of 5 ms for a 200 Hz sampling frequency, this processing time is very acceptable for single-channel EEG processing in real-time applications. In contrast, processing a single test segment on the development laptop required approximately 10 ms.

Table IV summarizes the computational costs measured in terms of computational time, memory usage, and power consumption in the multi-channel configuration to allow comparison to the FastICA implementation. The DAE TF-Lite model has the best performance when running on a CPU, requiring the shortest processing time and lowest battery consumption. The GPU with hardware acceleration consumes slightly more time and power for processing.

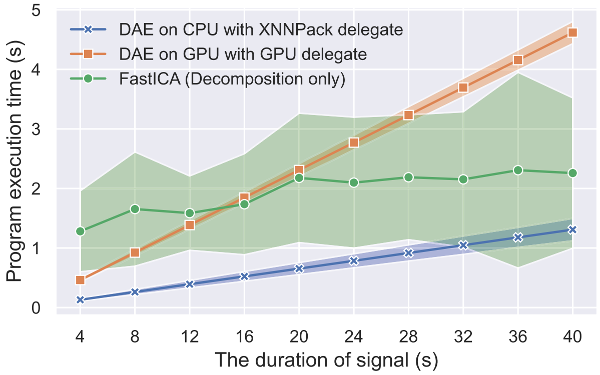

Fig. 7. Program execution time depending on the segment size of EEG processed for 32 channels. The dots show the average processing time and shaded areas show the standard deviation.

TABLE V
Computational Cost Comparisons Between On-Smartphone TF-Lite Models for 32 Channel Configuration

	Default TF-Lite no optimization		TF-Lite int-8 model		TF-Lite float-16 model	
	CPU	GPU	CPU	GPU	CPU	GPU
Computational time (s)	0.14	0.34	0.15	0.34	0.15	0.33
Power usage (mAh/h)	0.43	29.03	0.37	27.65	0.27	20.25
Memory usage (MB)	101.9	124.4	102.2	122.8	101.1	123.6

The trend for on-CPU running to offer the best performance is seen further in Fig. 7 which compares the on-smartphone computational time between the FastICA algorithm and DAE TF-Lite model at different EEG segment durations for the 32 channel data. The DAE TF-Lite model running on the CPU with XNNPack delegate (blue line) shows the fastest computational speed over all EEG durations. Again, the GPU delegate (orange line) is slower than the CPU in the test demo App as the signal duration increases. While on the Desktop/Cloud GPUs are often used to accelerate artificial intelligence implementations, for real-time deployment in the smartphone demo App we note that: (1) The test EEG data are pre-stored in the smartphone memory, and the data need to be transferred from the memory to the processor, and finally to the GPU for execution, which causes delay due to data loading. (2) This DAE TF-Lite model is relatively small with a limited number of parameters, which benefits the computation of the processor, rather than the GPU which is more helpful when the model is more complicated with numerous trainable parameters. (3) The GPU may be superior particularly for 2D image processing. However, the current model is developed for single-channel data processing, which may not utilise the strengths of a GPU. Similar effects of CPU implementations being faster than GPU implementations on smartphones have been observed in other works [56].

In comparison, the measured computational time of the FastICA decomposition (green line in Fig. 7) is slower than DAE TF-Lite with CPU, but faster than the DAE TF-Lite model with GPU when the signal duration increases. However, the execution time of FastICA decomposition has a high variance (the green shaded area), fluctuating in the approximate range of ±1 s or more. In contrast, the DAE TF-Lite model shows very stable computational performance with very low variance, important for real-time use allowing the processing latency to be known. In addition, the FastICA demo App implemented only computes the signal decomposition to produce the independent components. In reality, a complete ICA algorithm for EEG artifact removal includes signal decomposition, bad component recognition and rejection, and EEG reconstruction from the cleaned Independent Components. These steps are expected to add considerable processing time.

Table IV summarizes the computational needs of the different approaches, here using a 4 s segment length. The FastICA algorithm requires the longest processing time, even though only signal decomposition is being performed, and consumes the most power at 89.94 mAh/h. This is hundreds of times higher than the DAE model with CPU delegate. In terms of memory usage, the different cases give very similar values, ranging from 101 to 125 MB, showing no substantial differences between DAE TF-Lite models and the FastICA algorithm. This suggests that the algorithms are not memory-dependent and easy fit on modern smartphones.

Table V shows the computational costs of the different optimized TF-lite models running on the smartphone. For all the TF-Lite models the CPU is faster than the GPU, and there is a very similar computational time across the different types of model optimization. Also, the memory usage for different models is similar. The primary benefit of the optimization is in power consumption, with this noticeably reduced by applying optimization to the TF-Lite model.

IV. Discussion

Our results show that deep learning based EEG artifact removal has been achieved, in real-time, running on a smartphone. Reconstruction accuracy is in-line with previous works such as [24] although generally lower than very deep networks such as those in [26].

The proposed autoencoder neural network model was built based on symmetrical 1-dimensional convolutional layers and the trained model size deliberately kept modest with only 16,357 parameters in total. (The encoder and decoder parts have 8180 and 8177 trainable parameters respectively.) However, it is foreseeable that more complicated models, such as those in [26], could give better performance but sacrifice computational performance for on-smartphone use. Our approach takes only 4 ms to process 4 s of single-channel data, when using the CPU. While some applications may be able to allow more processing time than this, ours is set to allow processing in less than the time period between two EEG sample points.

Our work has a number of limitations. Optimization of the current model could be achieved by adding more hidden layers or using recurrent neural networks, which might be a good fit for time-series data such as EEG. More training datasets could potentially improve the model's accuracy, with more robust EEG artifact removal performance. We used public datasets where both a clean and a corrupted copy of the data was available due to the experiment carried out, but there is a wide range of *standard* EEG experiment data that is also publicly available if a reference free training approach can be devised. As a result of the above, our datasets were not from low channel count, wearable, EEG datasets, where our artifact removal approach might ideally be used. This is a limitation of the work.

Also, as a limitation of our work, note that the open dataset from [24] used 2 s epochs, which we concatenated together to form longer signals. We applied filtering from 1–50 Hz in the ground truth record to remove potential signal discontinuities at join points. Nevertheless, this could present a systematic aspect to this dataset (one of three used here) which may warrant consideration when using other datasets which are epoched differently. Further, the EEG/motion dataset did not have a notch filter applied by the original research team. In our work it uses the same processing pipeline as the other datasets; that is no notch filter being applied. This gives a learning/test case where mains interference is potentially present in recordings, but an important limitation to note is that the data was recorded in the EU, with a 50 Hz mains frequency. The learning may thus be systematically different to US based recordings where a 60 Hz mains frequency is present.

For improving reconstruction performance further there are of course many different deep learning architectures that could be investigated. Our approach is based upon a deep autoencoder, and so a clear next step might be to investigate variational autoencoders [34]. Further work on *denormalization* would also be beneficial. Our algorithm was developed by using segment-by-segment normalization rather than batch normalization over the whole signal. This was important to allow real-time operation as each segment can be processed independently, and avoids any potential leakage of training information into the test data via the normalization. However, this approach makes it hard to denormalize the cleaned data and return an output in μV. The normalization parameters are calculated on the corrupted data, which is generally much higher in amplitude than the cleaned data, so these same parameters cannot be used for denormalization. As a result, the current model could only output the cleaned EEG segments in the normalized amplitude range 0–1 before detrending, without recovering to the original EEG amplitude. This is similar to the approach taken in [24], [26], [27]. The morphology and spectral features are preserved, but not the absolute amplitude. Many EEG-based applications, such as Steady-State Visual Evoked Potentials (SSVEP)-based robotics control [57], [58], spectral-based motor imagery classification [59], and alpha entrainment [60], [61], do not rely on the absolute EEG amplitude, instead making use of the relative spectral information, minimizing the impact of this limitation. Nevertheless, our approach is not currently suitable for real-time artifact removal situations where averaging between trials or recovery of the absolute EEG amplitude is required. Further investigations are required to determine the most suitable way to perform amplitude recovery if required.

V. Conclusion

This paper proposed a Deep Autoencoder neural network which could effectively attenuate or remove EOG, motion, and EMG artifacts from an EEG signal. A comparison between this work and other papers shows a good EEG cleaning performance, with the benefit of less model complexity. Importantly, the proposed algorithm was successfully implemented on an Android smartphone, and was demonstrated to have faster computational time and lower power consumption than a gold standard ICA-based algorithm. The application does not yet receive real-time streamed data, as this is EEG unit/driver dependent, but we believe this can be readily added in future versions. Overall, our demonstration effectively promotes deep learning from offline algorithm validations to real-world smartphone applications, which will benefit next-generation wearable and mobile EEG systems and portable Brain-Computer Interface applications.

References

[1] A. J. Casson et al., "Electroencephalogram," in *Seamless Healthcare Monitoring*,T. Tamura and W. Chen, Eds. Berlin, Germany: Springer, 2018, pp. 45–81.

[2] A. J. Casson, "EEG goes home," *Clin. Neurophysiol.*, vol. 142, pp. 254–255, 2022.

[3] J. Minguillon et al., "Trends in EEG-BCI for daily-life: Requirements for artifact removal," *Biomed. Signal Process. Control*, vol. 31, pp. 407–418, 2017.

[4] V. Mihajlović et al., "Wearable, wireless EEG solutions in daily life applications: What are we missing?," *IEEE J. Biomed. Health Inform.*, vol. 19, no. 1, pp. 6–21, Jan. 2015.

[5] L. Xing, J. C. Batchelor, and A. J. Casson, "Opportunities and challenges for flexible and printable electrodes in electroencephalography," in *Proc. IEEE Int. Conf. Flexible Printable Sensors Syst.*, 2021, pp. 1–4.

[6] A. J. Casson, "Wearable EEG and beyond," *Biomed. Eng. Lett.*, vol. 9, no. 1, pp. 53–71, 2019.

[7] X. Gao et al., "Interface, interaction, and intelligence in generalized brain–computer interfaces," *Trends Cogn. Sci.*, vol. 25, no. 8, pp. 671–684, 2021.

[8] X. Gu et al., "EEG-based brain-computer interfaces (BCIs): A survey of recent studies on signal sensing technologies and computational intelligence approaches and their applications," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 18, no. 5, pp. 1645–1666, Sep./Oct. 2021.

[9] G. Niso et al., "Wireless EEG: A survey of systems and studies," *NeuroImage*, vol. 269, 2022, Art. no. 119774.

[10] C. He et al., "Diversity and suitability of the state-of-the-art wearable and wireless EEG systems review," *IEEE J. Biomed. Health Inform.*, vol. 27, no. 8, pp. 3830–3843, Aug. 2023.

[11] mBrainTrain, 2023. [Online]. Available: https://mbraintrain.com/

[12] Muse, 2023. [Online]. Available: https://choosemuse.com/

[13] Dreem, 2023. [Online]. Available: https://dreem.com/

[14] N. K. Jacob, H. O. Kings, and A. J. Casson, "A smartphone based platform for portable non-invasive light and sound neuromodulation," in *Proc. IEEE 42nd Annu. Int. Conf. Eng. Med. Biol. Soc.*, 2020, pp. 5228–5231.

[15] C. Lustenberger et al., "Auditory deep sleep stimulation in older adults at home: A randomized crossover trial," *Commun. Med.*, vol. 2, no. 1, 2022, Art. no. 30.

[16] E. H. T. Shad, M. Molinas, and T. Ytterdal, "Impedance and noise of passive and active dry EEG electrodes: A review," *IEEE Sensors J.*, vol. 20, no. 24, pp. 14565–14577, Dec. 2020.

[17] M. K. Islam, A. Rastegarnia, and Z. Yang, "Methods for artifact detection and removal from scalp EEG: A review," *Neurophysiologie Clinique/Clin. Neurophysiol.*, vol. 46, no. 4/5, pp. 287–305, 2016.

[18] D. Gorjan et al., "Removal of movement-induced EEG artifacts: Current state of the art and guidelines," *J. Neural Eng.*, vol. 19, 2022, Art. no. 011004.

[19] S. Song and A. D. Nordin, "Mobile electroencephalography for studying neural control of human locomotion," *Front. Hum. Neurosci.*, vol. 15, 2021, Art. no. 749017.

[20] W. Mumtaz, S. Rasheed, and A. Irfan, "Review of challenges associated with the EEG artifact removal methods," *Biomed. Signal Process. Control*, vol. 68, 2021, Art. no. 102741.

[21] C. Q. Lai et al., "Artifacts and noise removal for electroencephalogram (EEG): A literature review," in *Proc. IEEE Symp. Comput. Appl. Ind. Electron.*, 2018, pp. 326–332.

[22] S. Sadiya, T. Alhanai, and M. M. Ghassemi, "Artifact detection and correction in EEG data: A review," in *Proc. IEEE/EMBS 10th Int. Conf. Neural Eng.*, 2021, pp. 495–498.

[23] A. S. Janani et al., "How many channels are enough? Evaluation of tonic cranial muscle artefact reduction using ICA with different numbers of EEG channels," in *Proc. IEEE 26th Eur. Signal Process. Conf.*, 2018, pp. 101–105.

[24] H. Zhang et al., "EEGdenoiseNet: A benchmark dataset for deep learning solutions of EEG denoising," *J. Neural Eng.*, vol. 18, no. 5, 2021, Art. no. 056057.

[25] W. Sun et al., "A novel end-to-end 1D-ResCNN model to remove artifact from EEG signals," *Neurocomputing*, vol. 404, pp. 108–121, 2020.

[26] T. Gao et al., "EEG reconstruction with a dual-scale CNN-LSTM model for deep artifact removal," *IEEE J. Biomed. Health Inform.*, vol. 27, no. 3, pp. 1283–1294, Mar. 2023.

[27] W. Zhang et al., "Two-stage intelligent multi-type artifact removal for single-channel EEG settings: A GRU autoencoder based approach," *IEEE Trans. Biomed. Eng.*, vol. 69, no. 10, pp. 3142–3154, Oct. 2022.

[28] E. Brophy et al., "Denoising EEG signals for real-world BCI applications using GANs," *Front. Neuroergonomics*, vol. 2, 2022, Art. no. 44.

[29] F. Lopes et al., "Automatic electroencephalogram artifact removal using deep convolutional neural networks," *IEEE Access*, vol. 9, pp. 149955–149970, 2021.

[30] N. M. N. Leite et al., "Deep convolutional autoencoder for EEG noise filtering," in *Proc. IEEE Int. Conf. Bioinf. Biomed.*, 2018, pp. 2605–2612.

[31] H.-A. T. Nguyen et al., "A deep sparse autoencoder method for automatic EOG artifact removal," in *Proc. IEEE 19th Int. Conf. Control Automat. Syst.*, 2019, pp. 346–351.

[32] Y. Sumiya et al., "NR-GAN: Noise reduction GAN for mice electroencephalogram signals," in *Proc. Int. Conf. Biomed. Imag. Signal Process.*, 2019, pp. 94–101.

[33] B. Yang et al., "Automatic ocular artifacts removal in EEG using deep learning," *Biomed. Signal Process. Control*, vol. 43, pp. 148–158, 2018.

[34] J. F. Hwaidi and T. M. Chen, "A noise removal approach from EEG recordings based on variational autoencoders," in *Proc. IEEE 13th Int. Conf. Comput. Automat. Eng.*, 2021, pp. 19–23.

[35] S. Saba-Sadiya et al., "Unsupervised EEG artifact detection and correction," *Front. Digit. Health*, vol. 2, 2021, Art. no. 608920.

[36] J. Yin et al., "A GAN guided parallel CNN and transformer network for EEG denoising," *IEEE J. Biomed. Health Inform.*, early access, May 23, 2023, doi: 10.1109/JBHI.2023.3277596.

[37] D. Bank, N. Koenigstein, and R. Giryes, "Autoencoders," 2020, *arXiv:2003.05991*.

[38] A. Ignatov et al., "AI benchmark: All about deep learning on smartphones in 2019," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop*, 2019, pp. 3617–3635.

[39] K. Kotowski, K. Stapor, and J. Ochab, *Deep Learning Methods in Electroencephalography*. Berlin, Germany: Springer, 2020, pp. 191–212.

[40] M. A. Klados and P. D. Bamidis, "A semi-simulated EEG/EOG dataset for the comparison of EOG artifact rejection techniques," *Data Brief*, vol. 8, pp. 1004–1006, 2016.

[41] K. T. Sweeney et al., "A methodology for validating artifact removal techniques for physiological signals," *IEEE Trans. Inf. Technol. Biomed.*, vol. 16, no. 5, pp. 918–926, Sep. 2012.

[42] A. L. Goldberger et al., "PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, pp. e215–e220, 2000.

[43] PhysioNet, "Motion artifact contaminated fNIRS and EEG data," 2012. [Online]. Available: https://doi.org/10.13026/C2988P

[44] C. Hohyun et al., "EEG datasets for motor imagery brain computer interface," *Gigascience*, vol. 6, no. 7, pp. 1–8, 2017.

[45] F. Chollet et al., "Keras," 2015. [Online]. Available: https://keras.io

[46] L. Gondara, "Medical image denoising using convolutional denoising autoencoders," in *Proc. IEEE 16th Int. Conf. Data Mining Workshops*, 2016, pp. 241–246.

[47] X. Lu et al., "Speech enhancement based on deep denoising autoencoder," in *Proc. Interspeech*, 2013, pp. 436–440.

[48] K. Briechle and U. D. Hanebeck, "Template matching using fast normalized cross correlation," *Int. Soc. Opt. Eng.*, vol. 4387, pp. 95–102, 2001.

[49] Y. R. Rao, N. Prathapani, and E. Nagabhooshanam, "Application of normalized cross correlation to image registration," *Int. J. Res. Eng. Technol.*, vol. 3, no. 5, pp. 12–16, 2014.

[50] L. Xing and A. J. Casson, "3D-printed, directly conductive and flexible electrodes for personalized electroencephalography," *Sensors Actuators A, Phys.*, vol. 349, 2023, Art. no. 114062.

[51] T. S. Ajani, A. L. Imoize, and A. A. Atayero, "An overview of machine learning within embedded and mobile devices–optimizations and applications," *Sensors*, vol. 21, no. 13, 2021, Art. no. 4412.

[52] H. Li, "Smile," 2014. [Online]. Available: https://haifengl.github.io

[53] EEGLAB, "Tutorials," 2024. [Online]. Available: https://eeglab.org/tutorials/06_RejectArtifacts/RunICA.html

[54] Google, "Battery historian," 2017. [Online]. Available: https://github.com/google/battery-historian

[55] J. Doke, "GRABIT," 2016. [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/7173-grabit

[56] A. Das et al., "Enabling on-device smartphone GPU based training: Lessons learned," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops Affiliated Events*, 2022, pp. 533–538.

[57] G. R. Muller-Putz and G. Pfurtscheller, "Control of an electrical prosthesis with an SSVEP-based BCI," *IEEE Trans. Biomed. Eng.*, vol. 55, no. 1, pp. 361–364, Jan. 2008.

[58] A. Güneysu and H. L. Akin, "An SSVEP based BCI to control a humanoid robot by using portable EEG device," in *Proc. IEEE 35th Annu. Int. Conf. Eng. Med. Biol. Soc.*, 2013, pp. 6905–6908.

[59] W. Wu et al., "Classifying single-trial EEG during motor imagery by Iterative Spatio-Spectral Patterns Learning (ISSPL)," *IEEE Trans. Biomed. Eng.*, vol. 55, no. 6, pp. 1733–1743, Jun. 2008.

[60] L. J. Arendsen et al., "Entraining alpha activity using visual stimulation in patients with chronic musculoskeletal pain: A feasibility study," *Front. Neurosci.*, vol. 14, 2020, Art. no. 828.

[61] K. Lopez-Diaz et al., "Alpha entrainment drives pain relief using visual stimulation in a sample of chronic pain patients: A proof-of-concept controlled study," *Neuroreport*, vol. 32, no. 5, pp. 394–398, 2021.