A comparison of stochastic and deterministic approaches for EEG signal denoising

How does a stochastic algorithm (Independent Component Analysis) compare to a deterministic iterative algorithm (Empirical Mode Decomposition) in removing artifacts in simulated EEG time series data?

Extended Essay in Computer Science

Word Count: 4,000

How does a stochastic algorithm (Independent Component Analysis) compare to a deterministic iterative algorithm (Empirical Mode Decomposition) in removing artifacts in simulated EEG time series data?

ANONYMOUS AUTHOR(S)

Additional Key Words and Phrases: electroencephalogram, timeseries, denoising

1 INTRODUCTION

Electroencephalography (EEG) is a widely used neuroimaging technique that measures electrical brain activity [3]. EEG data record the brain's electric potentials, essentially neural activity [4], in microvolts and represents a time series - a sequence of measurements taken at regular intervals over time

EEG has proven invaluable in various fields, including clinical diagnosis and monitoring of neurological conditions like epilepsy [14], cognitive neuroscience for identifying patterns of brain activity [16], and in brain-computer interfaces, establishing a communication pathway between the brain and external devices[15]. However, EEG recordings are often affected by artifacts, unwanted signals that contaminate the acquired data, hindering accurate analysis [18]. Consequently, denoising, the removal of noise in data, becomes vital in EEG analysis and usage.

The evaluation of denoising algorithms lacks standardized protocols and benchmark datasets. Different studies often use varied evaluation metrics, noise models, and performance criteria, making direct comparisons of different algorithms' efficacy challenging. The variability in EEG data and the lack of consensus in evaluation methodologies hinder the determination of the most efficient algorithm for various EEG denoising tasks. Developing effective artifact removal methods is thus essential to enhance the quality and reliability of EEG signals [10].

In recent years, several algorithms have been proposed for artifact removal in EEG data. Independent Component Analysis (ICA) and Empirical Mode Decomposition (EMD) have gained considerable attention - 34% of papers citing EEG denoising utilize ICA, while 8% employ EMD [10].

ICA is a stochastic algorithm, incorporating randomness, making the behavior and outcome of the algorithm potentially variable across different runs, even with the same input. Its effectiveness and flexibility in denoising are based on several premises: statistical independence and instantaneous mixing of the source signals [10], the observed signal's dimensionality being greater than or equal to the source signals' dimensionality [11, 17] and the non-Gaussian nature of the sources [10].

On the other hand, EMD is a deterministic iterative algorithm, meaning its outcome and its intermediate steps can be precisely determined from its initial state and inputs/ EMD is employed to break down a signal into intrinsic mode functions (IMF), which capture distinct oscillatory components. It operates iteratively, meaning it repeats the decomposition process multiple times to extract these components [19].

This manuscript aims to compare the performance of two denoising algorithms, Independent Component Analysis (ICA) and Empirical Mode Decomposition (EMD), in removing artifacts from simulated EEG time series data. The focus is on ocular artifacts, which are EEG noise from eye blinks.

The research assesses how well each algorithm can separate the desired EEG signal from ocular artifacts, evaluating them based on time complexity, space complexity, Root Mean Squared Error (RMSE), and Normalized Mean Squared Error (NMSE). These metrics provide a comprehensive understanding of the trade-offs and advantages of each algorithm in terms of denoising quality and computational resource utilization. This study employs simulated EEG data for controlled experimentation, allowing for a systematic evaluation of the algorithms' performance against a known ground truth for artifact presence and characteristics.

2 METHODOLOGY

2.1 Experimental Setup

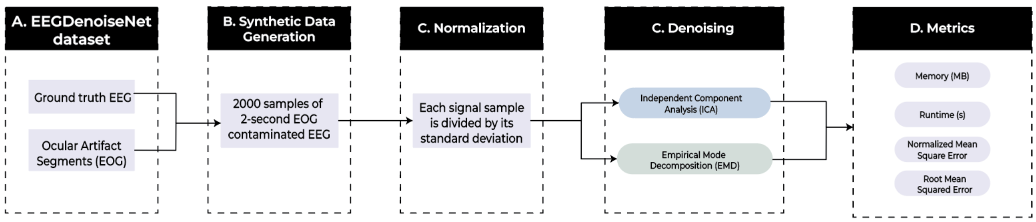

Fig. 1. Methodology depicting the process of artifact removal efficacy comparison between EMD and ICA on EEG data

Previous research indicates that the Signal-to-Noise Ratio (SNR) for EEG data contaminated with ocular artifacts typically ranges from -7 to -1 dB [21]. Taking this into consideration, for each SNR within the specified range, 5 trials will be conducted. In each trial, 7 sets of 2000 samples of 2-second ocular-artifact-contaminated EEG will be generated. These samples will be stored in a Python list, with each sample comprising an array of 512 float values. This array represents 2 seconds of EEG data at a 257 Hz sampling rate.

To accurately gauge the memory utilization for our dataset, it is imperative to underscore that variables in Python, such as floats, are not mere primitive data types but are implemented as objects, inherently consuming additional memory beyond their nominal data size. As a result, the actual memory footprint for one sample, containing 512 floats, substantially exceeds the simplistic computation of 4096 bytes. Consequently, the aggregate memory required for the dataset of 2000 samples is considerably greater than the initial calculation of 8.192 megabytes (MB), a fact that is critical to incorporate in our evaluation of the denoising algorithms' memory efficiency.

Finally, it is important to control the environment during the conduct of the experiment. Thus, the experiment will be conducted on the same system. The relevant details of the system can be found in the table below. The experiment will be conducted through a Python Notebook on PyCharm, an integrated development environment used for programming in Python.

Component	**Detail**
CPU	Model: MacBook Air M1 2020 Chip 8-core CPU 7-core GPU
Memory	8GB
Storage	256GB SSD
Operating System	macOS Ventura 13.2.1
Python version	Python 3.9.13

Table 1. Components and Details of the Experimental Environment

2.2 Simulated EEG Dataset

To compare the two algorithms, I will be synthetically generating a contaminated dataset as this offers controlled conditions with varying Signal-to-Noise ratios. This synthetic dataset provides a benchmark “clean” signal, facilitating a direct evaluation of each algorithm’s ability to remove artifacts. Moreover, this approach not only bypasses the logistical challenge of participant recruitment but it also eliminates the ethical concerns of handling real patient information and the concerns of variability in real-world datasets.

This study employs EEGDenoiseNet [22], a benchmark EEG dataset for training and testing Deep Learning denoising models. EEGDenoiseNet comprises quantitative time series data from public repositories, provided in platform-independent .npy format. It contains 4514 clean EEG segments and 3400 ocular artifact segments, allowing the synthesis of up to 15347600 unique ocular-artifact-contaminated EEG segments. The dataset underwent preprocessing and standardization, resulting in 2-second epochs per sample.

Equation (1) allows the creation of contaminated signals by linearly combining pure EEG segments with electroencephalography (EOG) segments[22].

$$y = x + \lambda \cdot n \tag{1}$$

In this context, y represents the combined one-dimensional signal of EEG and artifacts, while x represents the clean EEG signal considered as the reference or ground truth. The term n denotes ocular artifacts. The hyperparameter λ is used to regulate the SNR in the contaminated EEG signal y. To adjust the SNR of the contaminated segment, the parameter λ can be modified following equation (2).

$$\text{SNR} = 10 \log \frac{RMS(x)}{RMS(\lambda \cdot n)} \tag{2}$$

in which the root mean squared (RMS) value is defined as equation (3).

$$\text{RMS} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} g_i^2} \tag{3}$$

where N denotes the number of temporal samples in the segment g and g_i denotes the i^{th} sample of a segment g. Given that each signal spans 2 seconds and is sampled at a rate of 256 Hz, $N = 512$. Notably, lower λ represents higher SNR, as less EOG artifacts are added in the EEG signal. In return, lower SNR means higher noise level.

Before the generated dataset undergoes denoising through EMD or ICA, I first normalized the contaminated EEG segment and the ground-truth EEG segment following equation (4) where σ_y is the standard deviation of the contaminated EEG signal segment y. This ensures that the scale of the data is consistent, which is crucial for comparison.

$$\hat{x} = \frac{x}{\sigma_y}, \hat{y} = \frac{y}{\sigma_y} \tag{4}$$

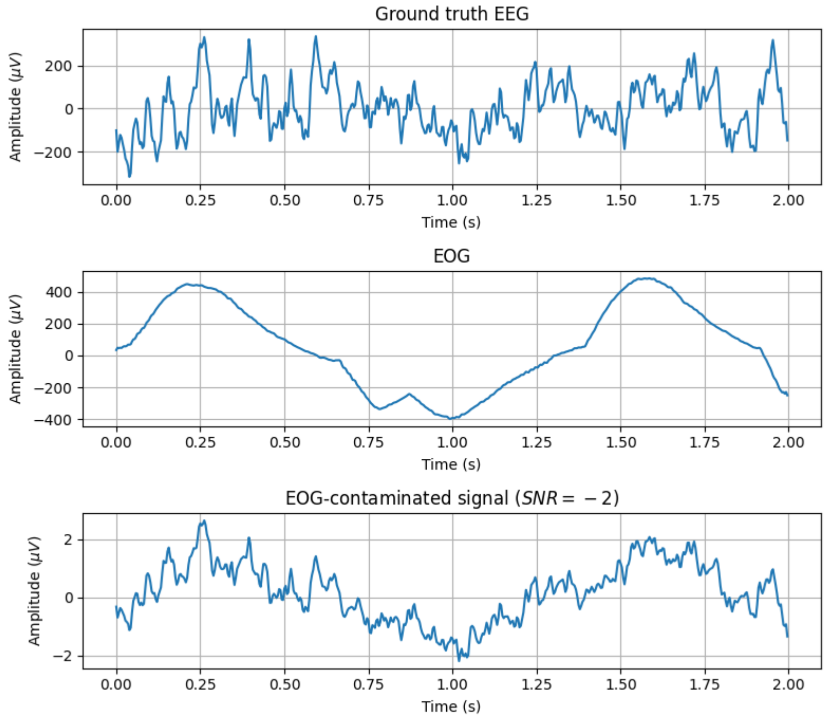

Fig. 2. Sample of synthetically generated data

Finally, Figure 2 above shows a sample of a synthetically generated EOG-artifact-contaminated EEG signal while Figure 3 illustrates how a contaminated signal would look across various SNRs.

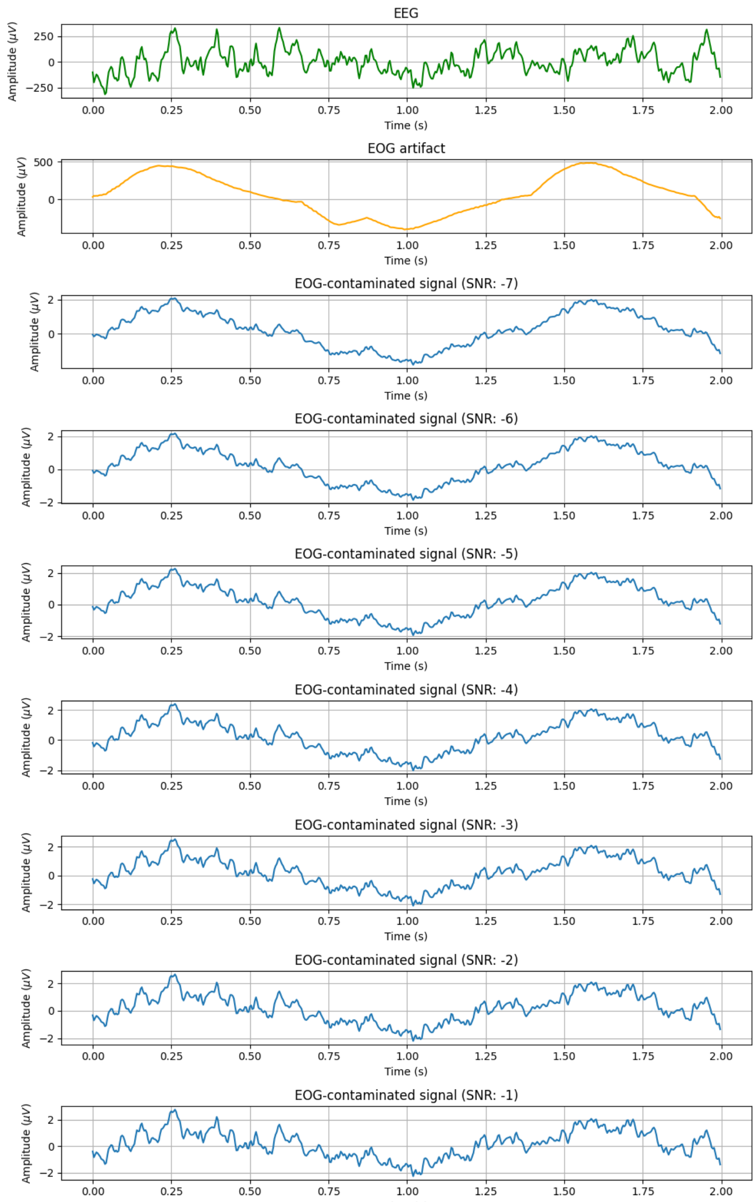

Fig. 3. A sample of a synthetically generated dataset in varying SNRs

2.3 Empirical Mode Decomposition

Empirical Mode Decomposition (EMD), introduced by Norden Huang in the late 1990s as part of the Hilbert-Huang Transform, is a technique designed to decompose a time series or signal into intrinsic mode functions (IMFs). EMD is predicated on the concept that signals consist of multiple fast and slow oscillations superimposed upon one another[7]. The technique effectively extracts these IMFs, each having a well-behaved envelope, to capture the various scales or modes inherent in the data, making it particularly well-suited for analyzing complex, nonlinear, and non-stationary signals [7].

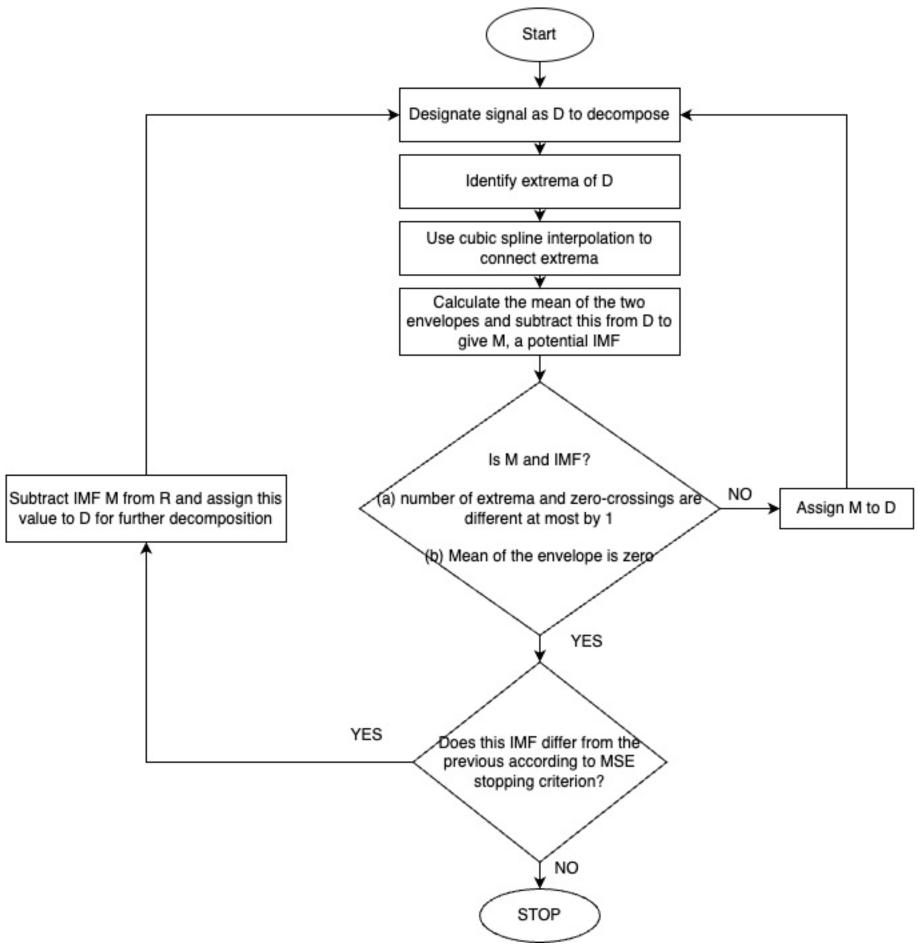

Fig. 4. Flow diagram of the Empirical Mode Decomposition Algorithm

EMD operates iteratively, as shown in Figure 4, performing sifting and decomposition to isolate simpler oscillatory components from complex data patterns. Algorithmically identifying local extrema, it employs cubic spline interpolation for computationally efficient envelope construction. The mean of these envelopes is computed and subtracted from the signal to yield an IMF candidate, which is iteratively refined against predefined criteria. Upon meeting a stopping criterion—like zero crossings or energy thresholds—the recursive sifting ceases, concluding the decomposition.

Several stopping criteria can be employed in EMD to determine when to halt the iterative process of extracting IMFs[19], as seen in Table 2.

Stopping Criterion	**Description**
Cauchy-Type Convergence	Stop the sifting process when the difference between two consecutive outputs, often measured by Mean Squared Error (MSE), is small enough to indicate convergence.
Number of Extrema and Zero Crossings	Requires the number of extrema and zero crossings to be equal or differ at most by one; stopping occurs when this condition is violated.
Fixed Number of Siftings	A predetermined number of siftings is performed for each IMF, stopping the process after this count is reached.
Threshold of Standard Deviation (SD)	The sifting stops when the standard deviation of the difference between consecutive siftings falls below a set threshold.
Energy Ratio	Stops the process when the energy ratio of the current IMF to the total signal energy falls below a certain threshold.
Residue Size	Ends the decomposition when the residue becomes a monotonic function or its energy falls below a certain level, indicating no further IMFs can be extracted.

Table 2. EMD Stopping Criteria

I will be using a stopping criterion based on an MSE threshold - the Cauchy-type convergence - as I found it the most appropriate for guaranteeing a fair comparison of performance between EMD and ICA. When employing this criterion, EMD's iterative process continues until the change between successive iterations of an IMF falls below a predefined MSE threshold. This approach guarantees that each IMF is a true representation of the signal's characteristics and not a byproduct of noise or other distortions. By setting a clear quantitative measure (the MSE threshold) for when an IMF is considered complete, EMD's output becomes more consistent, which is essential for a fair comparison with ICA. It avoids the extraction of spurious IMFs that could arise from noise, ensuring that the comparison between the algorithms focuses on their ability to reconstruct the original signal and isolate artifacts effectively.

Of course, it is not without its drawbacks. It's know to be sensitive to noise, as small fluctuations might be amplified, leading to a premature stop [19]. However, this concern is mitigated in the context of this research as a synthetically generated dataset with uniform noise levels, from -7 to -1dB, will be applied consistently across all trials. This consistency in noise exposure ensures that the criterion's noise sensitivity does not skew the comparison between EMD and ICA outcomes.

The criterion's issue lies however in the fact that if the convergence threshold is not set appropriately, it could result in either overfitting, where noise and signal features are conflated, or underfitting, where not all the significant oscillatory modes are captured [19]. In this research, I will be using a threshold of 10-5 as recommended by previous papers for analyses employing simulated EEG data[19].

To perform EMD, I will be using PyEMD, a Python package that provides algorithms for performing EMD and its variations.

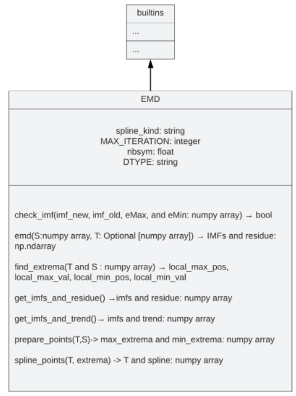

Fig. 5. UML Diagram of the EMD class of the PyEMD Library

Figure 5 reveals that the existing PyEMD library lacks a feature for specifying a Cauchy-type convergence threshold. Consequently, I developed a custom script, guided by the process outlined in Figure 4, which integrates the desired stopping criterion with the PyEMD library's functionality.

Algorithm 1 Pseudocode of EMD with Cauchy Convergence Stopping Criterion

```
emd = EMD()                                              ▷ EMD Class
residue = signal
all_imfs = []
while MSE < toleranceorlast_imfisnotempty do
    imfs ← residue        ▷ residue obtained using emd() method from EMD class
    if no more IMFs then
        break
    end if
    current_imf = last imf from imfs
    Add current_imf to all_imfs list
    last_imf = current_imf
    residue -= current_imf
end while
```

Figure 6 shows a sample resulting IMFs of the EMD algorithm.

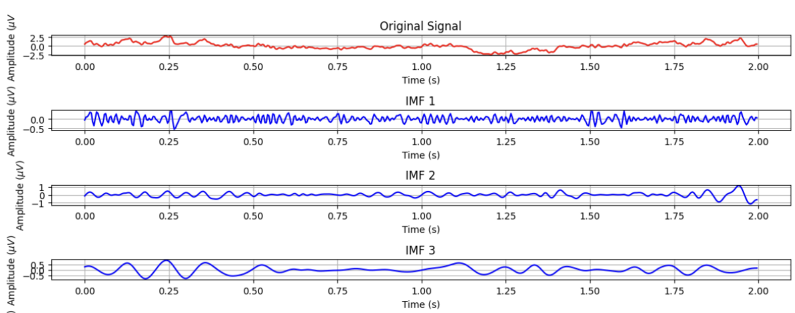

Fig. 6. IMFs obtained after performing EMD algorithm on a contaminated signal sample

2.4 Independent Component Analysis

Independent Component Analysis (ICA) is a computational method for separating a multivariate signal into additive, independent non-Gaussian signals, or components. ICA is based on the assumption that the observed multivariate data are linear mixtures of unknown latent variables or source signals, and the mixing system is also unknown [9]. The technique is aimed at recovering the latent variables, assumed to be non-Gaussian and statistically independent, from the observed mixtures. ICA is widely used for blind source separation tasks, where the original signals are unknown and not directly observable[13]. This makes ICA particularly powerful for applications such as extracting meaningful signals from background noise [5, 8].

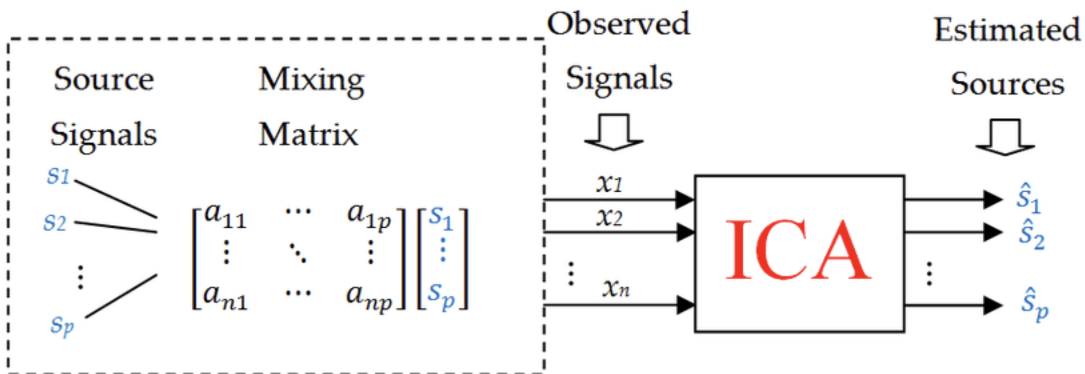

Fig. 7. Block Diagram of the Independent Component Analysis Algorithm

In this study, I will employ FastICA, a widely recognized algorithm for Independent Component Analysis, known for its robustness in separating non-Gaussian independent components from multivariate signals. To ensure a coherent and fair comparison with Empirical Mode Decomposition (EMD), which utilizes a cauchy convergence (MSE-based) threshold, we have implemented a Mean Squared Error (MSE) threshold as the stopping criterion for FastICA. This choice of stopping criterion is pivotal in aligning the termination conditions of the two algorithms, thereby facilitating a more equitable assessment of their performance. Utilizing MSE as a threshold allows us to establish a quantifiable and objective point at which FastICA has achieved a satisfactory level of accuracy, akin to the qualitative convergence achieved in EMD. This parallel in stopping criteria is crucial to ensure that any observed differences in the performance outcomes of FastICA and EMD can be attributed to their inherent algorithmic characteristics, rather than disparities in their termination conditions.

Investigating the parameters of a library like FastICA before comparison with EMD is crucial as it ensures a thorough understanding of how the algorithm behaves and can be optimized for specific data characteristics. This exploration is essential to ensure fair and accurate comparisons, as different parameter settings can significantly impact the algorithm's performance, effectiveness, and applicability to the data at hand.

Parameter	Description	Default Value	Other Options	Implications of Changing Default Value
n_components	Number of components to use. If None, all are used.	None	Any integer	Affects the number of independent components extracted; setting a specific number can reduce dimensionality or target specific signals.
algorithm	Algorithm to use for FastICA.	'parallel'	'deflation'	Different algorithms may converge differently and have impacts on performance and accuracy of component separation.
whiten	Whitening strategy to use.	'warn'	'arbitrary-variance', 'unit-variance', False	Changes in whitening can affect preprocessing and thereby influence the independence and scaling of components.
fun	The functional form of the G function used in the approximation to neg-entropy.	'logcosh'	'exp', 'cube', callable	Different functions can lead to varying results in non-Gaussian component extraction, impacting the final ICA outcome.
fun_args	Arguments to send to the functional form.	None	Dictionary of arguments	Customizing function arguments allows fine-tuning of the algorithm for specific types of data or desired outcomes.
max_iter	Maximum number of iterations during fit.	200	Any positive integer	Increasing iterations can improve convergence at the cost of computational time, while reducing may speed up the algorithm but risk poor convergence.
tol	Tolerance at which the un-mixing matrix is considered to have converged.	0.0001	Any positive scalar	Adjusting tolerance affects the convergence criterion; lower values can lead to more precise but potentially slower convergence.
w_init	Initial un-mixing array. If None, an array drawn from a normal distribution is used.	None	Array-like of shape (n_components, n_components)	Providing a specific initial un-mixing matrix can influence convergence and is useful when prior knowledge about the signal is available.
whiten_solver	Solver to use for whitening.	'svd'	'eigh'	Choice of solver can impact computational efficiency, especially for large datasets, with 'eigh' being more memory efficient in certain cases.
random_state	Seed used by the random number generator for reproducibility.	None	int, RandomState instance	Setting a seed ensures reproducibility in results, crucial for comparative studies and consistent algorithm behavior across runs.

Table 3. FastICA parameters and implications

I will be utilizing FastICA's default parameters as this approach maintains objectivity by focusing on the inherent differences between the algorithms, rather than outcomes influenced by specialized configurations. Similar to PyEMD, FastICA does not have a parameter that allows for an MSE-based stopping criterion. To utilize this stopping criterion in the implementation of ICA, I created the script and made it in the same structure as for the EMD implementation.

Algorithm 2 Pseudocode of ICA with Cauchy Convergence Stopping Criterion

```
icaa = ICA()                                              ▷ ICA Class
last_components=None
while MSE < toleranceorlast_componentsisnotemptylast do
    ica.max_iter = i + 1                                  ▷ Incremental increase in iterations
    components = ica.fit_transform(signal)
    last_components = components
end while
```

2.5 Comparisons

The algorithms will be compared using Normalized Mean-Squared Error (NMSE), Root Mean Squared Error (RMSE), time complexity, and space complexity on the same computer system.

Time complexity and space complexity measure the computational efficiency of an algorithm in terms of the time and memory requirements respectively, to execute the algorithm as a function of the input size. These metrics provide insights into the algorithm's scalability and efficiency, particularly in resource-constrained environments, and find a balance between computational performance and resource usage.

To accurately assess the time and space complexity of the ICA and EMD algorithms, effective memory management is essential, particularly through garbage collection in Python. This process automatically frees up memory from objects that are no longer needed, a critical step for precise memory profiling. I'll utilize the memory profiler and time library to measure these metrics.

The approach also involves isolating the algorithm by starting a timer at the algorithm's initiation and stopping it right after its completion, ensuring that both time and memory measurements are strictly associated with the algorithm's operation, free from any interference from the rest of the program. Consistency in hardware and software environments is also key to ensuring the accuracy and reliability of the results. For a realistic and comprehensive analysis, the data used includes a range of random Signal-to-Noise Ratios (SNRs), reflective of the variety encountered in real-world EEG data, thus providing a robust basis for comparing the efficiencies of ICA and EMD.

In order to compare the denoising quality of EMD and ICA, I will use RMSE and NMSE as means of comparison. Before doing so, I collect the denoised signals from each algorithms.

Applying ICA would decompose the contaminated signal into several independent components. Similarly, EMD would decompose the signal into IMFs. Thus, to obtain the actual denoised signal, I will utilize Kurtosis-based detection, a statistical measure that describes the "tailedness" of the probability distribution of a real-valued random variable. In the context of EEG, components with high kurtosis are often associated with artifacts.Components with kurtosis significantly higher than that of the original signal can be considered as noise. Kurtosis is calculated using Formula 5 where X_i is a data point in the presumed denoised signal (the ICA component or the IMF), μ is the mean of the dataset, σ is the standard deviation, and N is the number of data points.

ICA decomposes the contaminated EEG signal into multiple independent components, while EMD breaks it down into various Intrinsic Mode Functions (IMFs). To isolate the actual denoised signal, I will employ kurtosis-based detection. This statistical method assesses the "tailedness" of a distribution of real-valued random variables[12]. In EEG analysis, high kurtosis in components typically indicates the presence of artifacts. Therefore, any component—whether an ICA component or an IMF—exhibiting a kurtosis significantly greater than the original signal's can be flagged as noise[12]. The kurtosis is computed as per Formula 5 where X_i represents each data point in the component or IMF under analysis, μ is the mean of these datapoints, σ is the standard deviation, and N is the number of data points.

$$\text{Kurtosis} = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{X_i - \mu}{\sigma}\right)^4 \tag{5}$$

With the actual non-noise component identified from the several IMFs and components from EMD and ICA respectively, the next step is to recombine these selected components to reconstruct the 'denoised' signal from EMD and ICA. Comparison with RMSE and NMSE can then occur.

The RMSE is the measure of standard deviation of evaluated deviation [20]where Y_i is the actual value and $\hat{Y}_i$ is the predicted value.

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{N}(Y_i - \hat{Y}_i)^2}{N}} \tag{6}$$

On the other hand, NMSE calculates the mean squared error between estimated and actual values, normalized by the observed data's variance. It is given by equation 7 where

$$\text{RMSE} = \frac{\sum_{i=1}^{N}(Y_i - \hat{Y}_i)^2}{S^2} \tag{7}$$

The rationale behind selecting RMSE lies in its ability to provide error magnitudes in the same units as the data, which is critical for a clear and direct interpretation of the algorithms' performance. RMSE will offer an intuitive understanding of the average error per sample introduced by each algorithm[2]. Moreover, NMSE is chosen for its advantage in standardizing the error, which allows for a fair and meaningful comparison across different Signal-to-Noise Ratios (SNRs) ranging from -7 to -1. This normalization is crucial, as it accounts for the varying levels of noise in the data, ensuring that the performance evaluation is not skewed by the inherent differences in SNR levels[6].

Before I arrived at the decision of using RMSE and NMSE, I considered other metrics like MSE. While MSE is a straightforward metric, it does not offer the same interpretability as RMSE. While MSE gives a measure of the average squared error, its units make it less intuitive to understand how large an error is on average. RMSE, by taking the square root of MSE, brings the error metric back to the original scale of the data, making it easier to understand and interpret. MAE, though effective in averaging out all errors equally, does not emphasize larger errors as RMSE does, which can be pivotal in applications where larger deviations are more consequential. Furthermore, the choice of NMSE over other normalized metrics like Relative Mean Squared Error (RNMSE) is driven by the need to maintain a consistent scale of comparison across different noise levels in our dataset, making our analysis robust and reliable.

3 RESULTS & DISCUSSION

3.1 Time Complexity

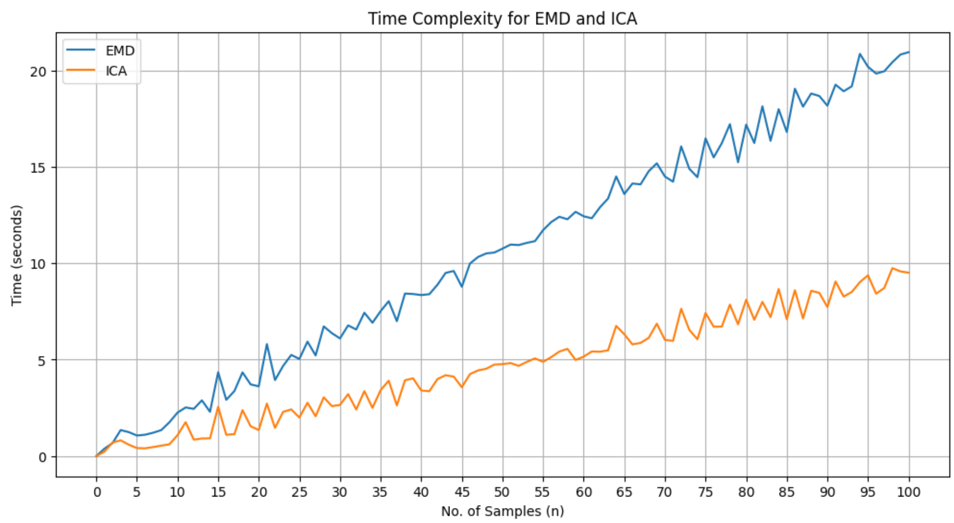

Fig. 8. Time Complexity graph comparing EMD and ICA over up to 100 samples

The blue line representing EMD shows a gradual increase in time with the number of samples. This suggests that the time complexity of EMD is likely to be linear or quasi-linear, $O(n)$ or $O(n \log n)$, with respect to the number of samples, which is typical for EMD because it processes each IMF sequentially. However, the rate of increase is not constant; there are regions where the slope is steeper. This could be due to EMD's sensitivity to signal characteristics such as noise, intermittency, or the presence of different frequency components that may require more iterations to resolve.

The orange line representing ICA demonstrates a more consistent but less steep increase, indicating a lower time required compared to EMD for the same number of samples. ICA's curve is smoother, which could be indicative of the algorithm's more consistent performance across different signal types or less sensitivity to the aforementioned signal characteristics.

Another potential cause for ICA's better performance in terms of time complexity is FastICA's structure. FastICA involves only simple operations such as matrix multiplication and non-linear functions applied element-wise. These operations are well-optimized in most numerical computing environments and thus can be scaled efficiently with increasing data size. FastICA is also known for its rapid convergence compared to other ICA algorithms[1]. It employs a fixed-point iteration scheme that often requires fewer iterations to converge, making it faster for large datasets.

EMD is an adaptive algorithm designed to decompose non-stationary and non-linear time series into a number of IMFs. It is an iterative process that can be computationally intensive, especially as the signal becomes more complex or contains more modes. The differences in the fundamental approaches of these algorithms might explain why EMD shows a higher time complexity in this graph. EMD's iterative sifting process can become more time-consuming as the number of IMFs increases with more samples. In contrast, FastICA optimizes a fixed number of components, possibly resulting in a more predictable increase in computation time with the number of samples.

3.2 Space Complexity

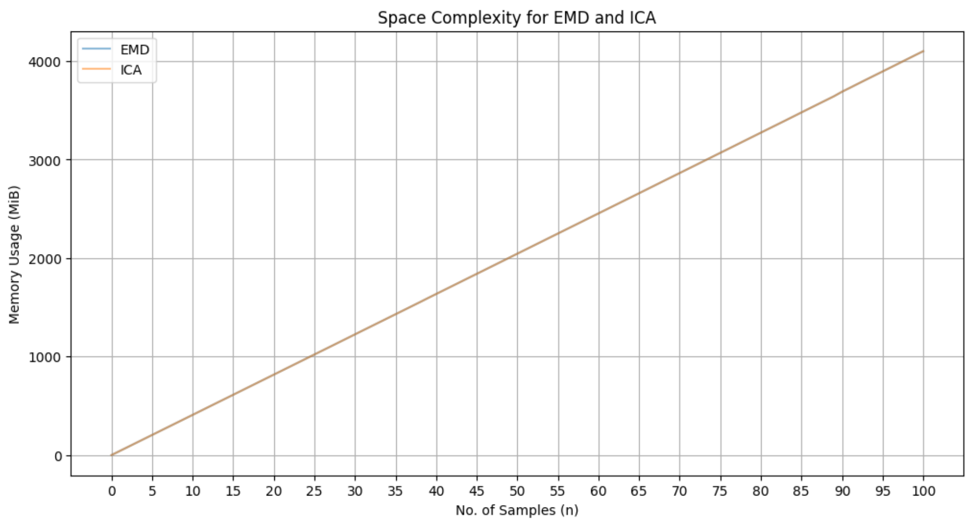

Fig. 9. Space Complexity graph comparing EMD and ICA over up to 100 samples

I expected the space complexity of ICA to be a function of the number of components and samples, which, in a straightforward implementation, would result in linear space complexity with respect to the number of samples. For EMD, I expected to have a space requirement that grows with the number of IMFs to be stored, which should increase with the number of samples as each IMF and residue must be stored in memory.

The overlay of the lines for EMD and ICA suggests an identical increase in memory usage for both algorithms, which is unusual given their different computational structures and mechanisms. This could imply that the memory measurement is not capturing the actual space complexity of the algorithms but rather some common overhead shared between them. Both algorithms used are also part of a larger library that has a fixed memory overhead and this overhead might have dominated the measured space complexity as well.

3.3 RMSE and NMSE

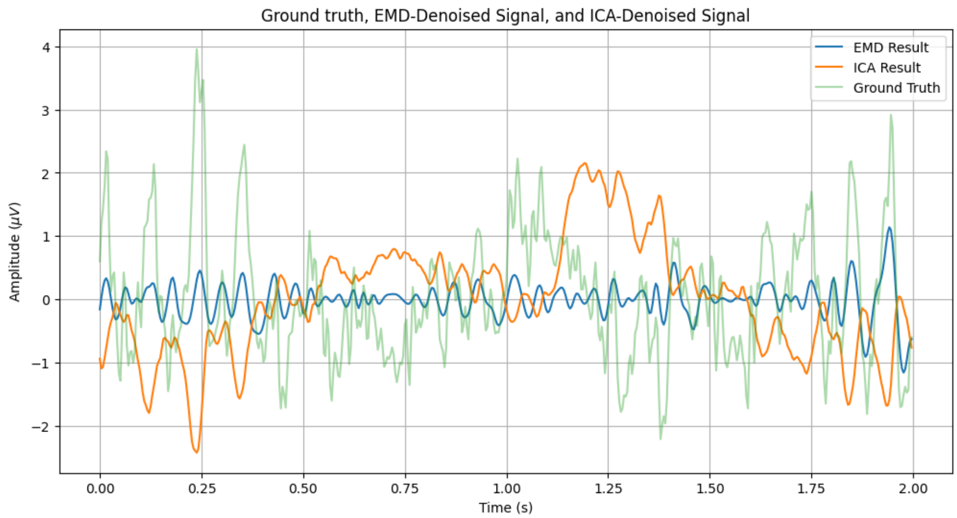

Fig. 10. Graph comparing the EMD-denoised signal, ICA-denoised signal, and the ground truth signal. SNR = −2

The EMD denoised signal (blue) seems to follow the general trend of the ground truth but with some notable deviations, particularly where the ground truth has sharp peaks or troughs. EMD works by decomposing a signal into its inherent oscillatory modes, which means it might sometimes reconstruct the signal without these sharp features if they are not consistent across all IMFs or if they are treated as noise.

The ICA denoised signal (orange) also follows the general trend but deviates from the ground truth in a different manner compared to the EMD result. It appears to smooth out the signal more than EMD, which suggests that it may be considering some of the quick changes in the signal as non-independent components or noise.

It's important to note that the contaminated signal denoised for comparison shown in Figure 10. has an SNR of -2. Different algorithms may perform variably across different SNR levels, and what works for this noise level might not be optimal for a different SNR, or vice versa.

Thus, I will compare both of the algorithms' performance quantitatively based on RMSE and NMSE.

SNR	RMSE		NMSE	
	EMD	ICA	EMD	ICA
-7	0.9443735949	1.405560046	0.8918414867	1.975599044
-6	0.9283421793	1.422007686	0.8618192019	2.022105859
-5	0.9102225753	1.441454982	0.8285051366	2.077792466
-4	0.8903412103	1.46253804	0.7927074708	2.139017519
-3	0.8702658478	1.486548252	0.7573626459	2.209825704
-2	0.8508360165	1.507848469	0.723921927	2.273607005
-1	0.834108529	1.518918735	0.6957370382	2.307114124

Table 4. Average RMSE and NMSE values of EMD and ICA in 5 trials across 7 varied SNRs

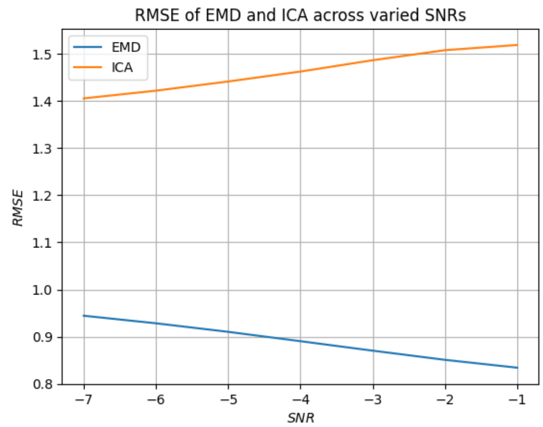

Fig. 11. Graph comparing the EMD-denoised signal, ICA-denoised signal, and the ground truth signal.

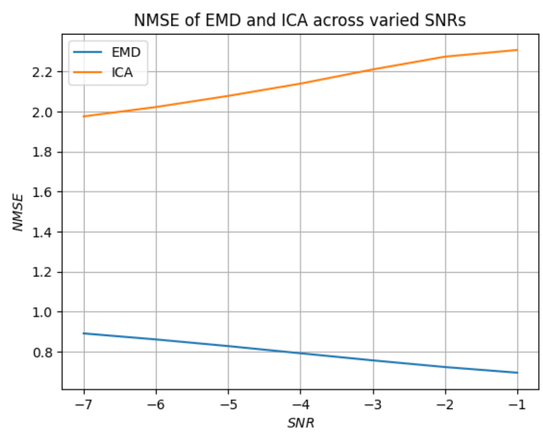

Fig. 12. Graph comparing the EMD-denoised signal, ICA-denoised signal, and the ground truth signal.

For EMD, both the RMSE and NMSE decrease as the SNR increases. This trend aligns with the expectation that EMD, which is inherently sensitive to noise, performs better with higher SNR levels. The algorithm decomposes signals into Intrinsic Mode Functions (IMFs) through an iterative sifting process. When the SNR is low, the noise can interfere with this process, causing EMD to potentially misidentify noise as part of the signal, which can lead to inaccurate decomposition. As the noise level decreases, EMD can more accurately identify and separate the true signal components, improving its accuracy as evidenced by the lower RMSE and NMSE.

ICA, on the other hand, exhibits distinct behaviors in RMSE and NMSE trends. The RMSE remains relatively flat, indicating that the performance of ICA is stable across different SNR levels. This stability suggests that ICA's ability to separate the signal from the noise is not significantly affected by changes in SNR, likely due to its reliance on the statistical independence of signal components rather than their time-frequency characteristics. Conversely, the NMSE for ICA increases with SNR, which is counterintuitive as we would typically expect better performance with cleaner data. This unexpected trend in NMSE might suggest that the assumptions of independence may not hold as the SNR increases or perhaps that ICA is overfitting to noise at lower SNRs, but this overfitting is less pronounced at higher SNRs.

Comparatively, EMD shows more sensitivity to SNR levels than ICA. While EMD benefits from a higher SNR, with a clear improvement in both RMSE and NMSE, ICA's performance in terms of RMSE does not appear to be influenced by SNR, but its NMSE performance raises questions about its robustness or the assumption of statistical independence between signal and noise.

Both methods are expected to perform better as the SNR increases, but the graph shows that EMD benefits more from increased SNR compared to ICA. This might be because, at higher SNRs, the signal's structure becomes clearer, aiding EMD's sifting process to accurately extract the signal's IMFs. ICA's flat RMSE line suggests that it may be a more robust choice in scenarios where SNR varies or cannot be controlled. It's important to note, however, that a flat RMSE does not necessarily mean better performance; it means consistent performance across the tested SNR range. The actual RMSE value might still be higher than desired. However, ICA's increasing NMSE with SNR requires further investigation to understand the underlying cause. It is crucial to consider the specific characteristics of the signals and noise, as well as the parameter settings of each algorithm, when interpreting these results and choosing the appropriate method for a given application.

4 CONCLUSION

In this manuscript, I explored the denoising capabilities of two algorithms, Independent Component Analysis (ICA) and Empirical Mode Decomposition (EMD), within the context of EEG signal processing. The research aimed to discern how a stochastic algorithm (ICA) compares to a deterministic iterative algorithm (EMD) in mitigating artifacts in simulated EEG time series data, particularly focusing on ocular artifacts.

The findings of this research indicate that ICA requires less additional processing time as the dataset size increases, which could makeit a more suitable choice in scenarios involving large-scale data analysis and where time efficiency is a critical factor. It's important to note however that due to the constraints of my experimental setup, with it being done on a MacBook Air with limited capacity, the results for time complexity only reflects up to 100 samples. This limitation represents a significant shortcoming, as the results may not extrapolate reliably to larger datasets. With this, an evaluation of the algorithms with a more extensive collection of data samples is recommended to ensure the robustness and generalizability of the observed trends.

In terms of space complexity, the observed data showed an identical trajectory of space complexity for both ICA and EMD, which deviates from expectations given their distinct computational architectures. This observation indicates a potential limitation in our experimental methodology, suggesting that the memory profiling may not have effectively distinguished between the algorithm's memory usage and the overhead associated with the encompassing library utilized for both algorithms's. Future investigations should consider a more granular approach to memory profiling, ensuring a more precise evaluation of their scalability and efficiency.

When comparing the two, EMD's sensitivity to SNR variations is evident, with a performance that scales positively with higher SNRs. ICA's robustness, as shown by the RMSE, offers an advantage in scenarios with fluctuating SNR levels. Nevertheless, the increasing NMSE with SNR raises critical questions about its efficacy under varying conditions, which may impact its practical deployment in EEG denoising tasks.

The comparative analysis provided in this study sheds light on the nuanced behaviors of these algorithms under different noise conditions, highlighting the need for careful consideration of the denoising context. It also underscores the importance of parameter optimization and the understanding of signal and noise characteristics for the successful application of these methods.

As the field of EEG denoising research advances, the development of standardized evaluation protocols and the refinement of denoising techniques will be crucial. The work contributes to this ongoing effort, providing insights that could inform future algorithm selection and optimization for EEG artifact removal.

REFERENCES

[1] L. Albera, A. Kachenoura, P. Comon, A. Karfoul, F. Wendling, L. Senhadji, and I. Merlet. 2012. ICA-Based EEG denoising: a comparative analysis of fifteen methods. *Bulletin of the Polish Academy of Sciences: Technical Sciences* 60, 3 (Dec. 2012), 407–418. https://doi.org/10.2478/v10175-012-0052-3

[2] Zaid Abdi Alkareem Alyasseri, Ahamad Tajudin Khader, Mohammed Azmi Al-Betar, Ammar Kamal Abasi, and Sharif Naser Makhadmeh. 2020. EEG Signals Denoising Using Optimal Wavelet Transform Hybridized With Efficient Metaheuristic Methods. *IEEE Access* 8 (2020), 10584–10605. https://doi.org/10.1109/ACCESS.2019.2962658

[3] Andrea Biasiucci, Benedetta Franceschiello, and Micah M. Murray. 2019. Electroencephalography. *Current Biology* 29, 3 (Feb. 2019), R80–R85. https://doi.org/10.1016/j.cub.2018.11.052

[4] Chorlian, David B., and Howard L. Cohen. 1995. Measuring Electrical Activity of the Brain. *RESEARCH WORLD, vol. 19, no. 4*.

[5] Pierre Comon. 2008. Independent Component Analysis. *Higher Order Statistics*.

[6] Harender and R. K. Sharma. 2017. EEG signal denoising based on wavelet transform. In *2017 International conference of Electronics, Communication and Aerospace Technology (ICECA)*, Vol. 1. 758–761. https://doi.org/10.1109/ICECA.2017.8203645

[7] Norden E. Huang, Zheng Shen, Steven R. Long, Manli C. Wu, Hsing H. Shih, Quanan Zheng, Nai-Chyuan Yen, Chi Chao Tung, and Henry H. Liu. 1998. The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences* 454, 1971 (March 1998), 903–995. https://doi.org/10.1098/rspa.1998.0193

[8] A. Hyvärinen and E. Oja. 2000. Independent component analysis: algorithms and applications. *Neural Networks* 13, 4-5 (June 2000), 411–430. https://doi.org/10.1016/S0893-6080(00)00026-5

[9] Herault J. and Jutten C. 1986. Space or time adaptive signal processing by neural network models. In *AIP Conference Proceedings*, Vol. 151. 206–211. https://doi.org/10.1063/1.36258

[10] Xiao Jiang, Gui-Bin Bian, and Zean Tian. 2019. Removal of Artifacts from EEG Signals: A Review. *Sensors* 19, 5 (Feb. 2019), 987. https://doi.org/10.3390/s19050987

[11] Tzyy-Ping Jung, Colin Humphries, Te-Won Lee, Scott Makeig, Martin McKeown, Vicente Iragui, and Terrence J Sejnowski. 1997. Extended ICA Removes Artifacts from Electroencephalographic Recordings. In *Advances in Neural Information Processing Systems*, M. Jordan, M. Kearns, and S. Solla (Eds.), Vol. 10. MIT Press. https://proceedings.neurips.cc/paper_files/paper/1997/file/674bfc5f6b72706fb769f5e93667bd23-Paper.pdf

[12] M.A. Klados, C. Bratsas, C. Frantzidis, C.L. Papadelis, P.D. Bamidis, and Nicolas Pallikarakis. 2010. A Kurtosis-Based Automatic System Using Naïve Bayesian Classifier to Identify ICA Components Contaminated by EOG or ECG Artifacts. In *XII Mediterranean Conference on Medical and Biological Engineering and Computing 2010*. 49–52.

[13] Ganesh Naik and Dinesh Kumar. 2011. An Overview of Independent Component Analysis and Its Applications. *Informatica* 35 (01 2011), 63–81.

[14] Soheyl Noachtar and Jan Rémi. 2009. The role of EEG in epilepsy: A critical review. *Epilepsy & Behavior* 15, 1 (May 2009), 22–33. https://doi.org/10.1016/j.yebeh.2009.02.035

[15] G. Pfurtscheller and C. Neuper. 2001. Motor imagery and direct brain-computer communication. *Proc. IEEE* 89, 7 (July 2001), 1123–1134. https://doi.org/10.1109/5.939829

[16] Parthana Sarma and Shovan Barma. 2020. Review on Stimuli Presentation for Affect Analysis Based on EEG. *IEEE Access* 8 (2020), 51991–52009. https://doi.org/10.1109/ACCESS.2020.2980893

[17] A. Schlögl, C. Keinrath, D. Zimmermann, R. Scherer, R. Leeb, and G. Pfurtscheller. 2007. A fully automated correction method of EOG artifacts in EEG recordings. *Clinical Neurophysiology* 118, 1 (Jan. 2007), 98–104. https://doi.org/10.1016/j.clinph.2006.09.003

[18] Monika Sheoran, Sanjeev Kumar, and Seema Chawla. 2015. Methods of denoising of electroencephalogram signal: a review. *International Journal of Biomedical Engineering and Technology* 18, 4 (2015), 385. https://doi.org/10.1504/IJBET.2015.071012

[19] Catherine M. Sweeney-Reed, Slawomir J. Nasuto, Marcus F. Vieira, and Adriano O. Andrade. 2018. Empirical Mode Decomposition and its Extensions Applied to EEG Analysis: A Review. *Advances in Data Science and Adaptive Analysis* 10, 02 (April 2018), 1840001. https://doi.org/10.1142/S2424922X18400016

[20] Priya AG Varshini, Anitha K Kumari, Janani D, and Soundariya S. 2021. Comparative analysis of Machine learning and Deep learning algorithms for Software Effort Estimation. *Journal of Physics: Conference Series* 1767, 12–19. Issue 1. https://doi.org/10.1088/1742-6596/1767/1/012019

[21] Gang Wang, Chaolin Teng, Kuo Li, Zhonglin Zhang, and Xiangguo Yan. 2016. The Removal of EOG Artifacts From EEG Signals Using Independent Component Analysis and Multivariate Empirical Mode Decomposition. *IEEE Journal of Biomedical and Health Informatics* 20, 5 (Sept. 2016), 1301–1308. https://doi.org/10.1109/JBHI.2015.2450196

[22] H. Zhang, M. Zhao, C. Wei, D. Mantini, Z. Li, and Q. Liu. 2021. EEGdenoiseNet: a benchmark dataset for deep learning solutions of EEG denoising. *Journal of Neural Engineering* 18 (Oct. 2021). Issue 5. https://doi.org/10.1088/1741-2552/ac2bf8