Expert Systems With Applications 251 (2024) 123993

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Explaining the predictions of kernel SVM models for neuroimaging data analysis

Mengqi Zhang, Matthias Treder, David Marshall, Yuhua Li *

School of Computer Science and Informatics, Cardiff University, Cardiff, UK

ARTICLE INFO

Keywords:
Support vector machines
Activation pattern
Neuroimaging
EEG

ABSTRACT

Machine learning methods have shown great performance in many areas, including neuroimaging data analysis. However, model performance is only one objective in neuroimaging analysis. Gaining insight from the data is also critical in this field, such as identifying regions where detected signals are relevant to cognitive and diagnostic tasks. To fulfill this need, enabling the explainability of a model's decision-making process is critical. Predictions of complex machine learning models are notoriously difficult to explain. This limits the use of complex models like kernel support vector machines (SVM) in neuroimaging analysis. Recently, several permutation-based methods have been developed to explain these complex models. However, the explanation results are affected by class-irrelevant features like suppressor variables and high background noise variables. This problem may also happen when explaining linear models. One possible reason is that the permutation process will produce unrealistic data instances when features are not independent, e.g. correlated. These unrealistic data instances will influence the explanation results. In neuroimaging analysis, the activation pattern, the estimated weight of the assumed generative model corresponding to the current classifier, is used to deal with this problem for linear models. This method does not rely on a permutation process but rather on the available data information. In this paper, we propose a novel method of Explanation through Activation Pattern (EAP) to explain the SVM models with different types of kernels for neuroimaging data analysis. Our method can generate a global feature importance score by estimating the activation pattern of kernel SVM models. We evaluate our method against three popular methods on both simulation datasets and a publicly available EEG/MEG dataset on visual tasks. The experimental results demonstrate that the proposed EAP method can provide explanations with low computational cost and is less affected by class-irrelevant features than the other three methods. In the experiment using the MEG/EEG dataset of visual tasks, the proposed EAP method provides agreement results with the brain's electrical activity patterns reported in the literature on the visual tasks EEG/MEG data and is significantly faster than the other explanation methods.

1. Introduction

Modern machine learning models are capable of advance and complex performance, especially with respect to prediction and classification. However, concerns have been raised about how these models make a decision. Complex models make predictions through a highly nonlinear data processing process, which is not transparent, since these models can be regarded as 'black-box' models. Due to this characteristic, the use of these models is limited in some fields where model explainability is equally essential as model performance. Neuroimage analysis is one of these fields. In this field, machine learning models can help to decode highly noisy signals like electroencephalogram (EEG) and magnetoencephalography (MEG). However, gaining insight related to the studied tasks is equally, if not more, important than model performance. Explaining the prediction from a model can help researchers understand the mechanism of the concerned cognitive tasks, for example, determining the brain region or specific channels of the brain cognitive process being studied. To fulfill this need, although nonlinear models may extract more information, linear models are preferred (Dima, Perry, Messaritaki, Zhang, & Singh, 2018; Sharma et al., 2022; Wang, Tian, Zhang, & Hu, 2022) due to their predictions being relatively easier to explain.

In recent years, eXplainable Artificial Intelligence (XAI) has become a topic of great interest with many techniques developed that can help to explain complex models. Several attempts have been made in the neuroimaging field (Lawhern et al., 2018; Sturm, Lapuschkin, Samek, & Müller, 2016). Although most efforts have focused on deep

* Corresponding author.
E-mail addresses: ZhangM55@cardiff.ac.uk (M. Zhang), trederm@cardiff.ac.uk (M. Treder), MarshallAD@cardiff.ac.uk (D. Marshall), LiY180@cardiff.ac.uk (Y. Li).

https://doi.org/10.1016/j.eswa.2024.123993
Received 17 August 2023; Received in revised form 17 March 2024; Accepted 14 April 2024
Available online 16 April 2024

learning models because of their promising performance in many tasks, traditional models can also benefit from these XAI techniques. The support vector machine (SVM) is still one of the most popular models in neuroimaging analysis. Usually, the amount of available neuroimaging data, for training and testing, is small. Compared with deep learning models, these models require less data and are easy to implement. Therefore, they are still prominently used within this field.

With respect to the need for model explanation, although nonlinear SVM models can extract more information and show better performance, linear SVM models are preferred and wildly used in this field (Lotte et al., 2018). One way to explain nonlinear SVM models is to measure the feature importance within the underlying data/model. These methods are model-agnostic XAI methods, such as local interpretable model-agnostic explanations (LIME) (Ribeiro, Singh, & Guestrin, 2016) and kernel Shapley additive explanations (SHAP) (Lundberg & Lee, 2017), that can characterize feature behaviours by permuting feature input. While these model-agnostic approaches have shown good model explanation power for many applications, their results can be affected by class-irrelevant features, such as suppressor variables (Krus & Wilkinson, 1986; Wilming, Budding, Müller, & Haufe, 2022) and background information (Budding, Eitel, Ritter, & Haufe, 2021). As their works indicated, even though class-irrelevant features are intentionally designed in a straightforward way, these features can still be assigned high importance scores. This can cause misunderstandings about the features and the current classification tasks. It is important to note that this problem may appear in both linear and nonlinear cases. One potential reason is that the permutation process assumes that features are independent, when some features are not independent, the newly produced instances may fall into data space that does not exist in the original dataset. These instances affect the calculation and cause bias to the final results.

To deal with this problem, Haufe et al. (2014) proposed a method for the linear models. When explaining a linear classifier, the first intuition is using model weights. However, as mentioned above, this sometimes results in a misleading explanation (Hebart & Baker, 2018; Kriegeskorte & Douglas, 2019), for example, where the only goal of a classifier is to gain better performance. To achieve this goal, a classifier will extract the interested signal and suppress the non-interested signal of the data. Extraction and suppression steps may eliminate non-interesting features, which can cause misinterpretation of the results. In Haufe et al. (2014), observations are assumed to be generated from a generative model associated with the current classifier. The weights of this generative model, which is called the activation pattern, can reflect those class-related features while suppressing those class-irrelevant features since the generative model directly reflects the generation process of observed data. Furthermore, this method relies on available data information rather than permutation processes, which does not suffer from the problem of the unrealistic data instances that the permutation process produces.

We adopt this notion and apply it to kernel SVM models. The idea of kernel SVM models is that data can be mapped from input space into higher dimensional kernel space. Those data that cannot be linearly separated in input space can be separated by a hyperplane in this kernel space. We can construct the high-dimensional activation pattern in kernel space by using this hyperplane. The feature importance can be measured by estimating the input space version of this high-dimensional activation pattern. This kind of result is less affected by class-irrelevant features.

This paper aims to address the aforementioned drawbacks of existing methods to propose an explanation method for kernel SVM with a focus on neuroimaging data analysis. The novel contributions of the paper can be summarized as follows:

- This paper proposes a novel activation pattern-based method for explaining kernel SVM in neuroimaging analysis tasks. We call this method: Explanation through Activation Pattern (EAP).
- It generates explanation results more robust to noise variables by constructing patterns instead of directly permuting features.
- It produces a faster explanation than other state-of-the-art explanation methods.
- It presents an extensive experimental evaluation using simulated data and real EEG/MEG data, Wakeman & Henson dataset (Wakeman & Henson, 2015), showing the explanation results consistently better agree with patterns of brain activities than the benchmark methods.

This paper proceeds as follows. The next section provides a brief review of the related works. The linear activation pattern is also introduced in this section. Section 3 presents the EAP method for explaining the predictions of kernel SVM models using activation patterns. Section 4 details the evaluation experiments, including the datasets used, the experiment setups and the selected representative explanation methods for comparison. The experiment results and discussion are presented in Section 5. The final section provides an overall conclusion of this paper.

2. Related works

Various methods have been proposed to make complex models transparent to users from different directions. For example, Bach et al. (2015) proposed a method of measuring feature contribution based on Taylor decomposition. Greenwell, Boehmke, and McCarthy (2018) developed a framework based on partial dependence plots (Friedman, 2001) for measuring interactions between 2 features. Some works try to explain models from an example-based view, such as finding representative samples (Kim, Khanna, & Koyejo, 2016) of a classifier and adversarial samples (Goodfellow, Shlens, & Szegedy, 2014; Szegedy et al., 2013) which can cheat the model. Recently, much effort has been made to focus on deep learning methods (Bučková, Brunovský, Bareš, & Hlinka, 2020; Guerrero-Gómez-Olmedo, Salmeron, & Kuchkovsky, 2020; Kim & Ye, 2020; Van Putten, Olbrich, & Arns, 2018). When applying XAI methods on deep learning models, their results may be influenced by class-irrelevant features such as large background noise (Budding et al., 2021), and suppressor variable (Wilming et al., 2022).

Deep learning models have some advantages when applied in neuroimaging analysis tasks, such as requiring fewer preprocessing steps. Despite the impressive improvement of deep learning models in recent years, such as in the Motor Imagery (MI) field (Al-Saegh, Dawwd, & Abdul-Jabbar, 2021), these models have yet to significantly improve performance in some tasks (Lotte et al., 2018). It has been observed that the SVM model generally still achieves considerable performance and is still the most frequently used method even in recent years (Pahuja, Veer, et al., 2022; Saeidi et al., 2021). This can be attributed to the simplicity and low computation cost of the SVM model (Moctezuma & Molinas, 2020), which makes it widely used in many tasks (Savadkoohi, Oladunni, & Thompson, 2020; Wen & Aris, 2022; Zhang et al., 2018). There is still an urgent need to explain these models.

One approach to non-linear SVM explanation uses rule-extraction (Barakat & Bradley, 2010), which is also applied to neural networks (Hailesilassie, 2016). Their goal is to learn a set of rules to mimic the output of the original model. They use EEG channels and associated values to represent the extracted rules. The rules are transparent and straightforward. However, as the number of rules increases and the length of each rule becomes longer, it can lead to the loss of interpretability and make it difficult for human interpretation (Minh, Wang, Li, & Nguyen, 2022).

Another potential approach is using gradient-based methods like sensitivity analysis. Rasmussen, Madsen, Lund, and Hansen (2011) has applied sensitivity analysis to identify important voxels in functional magnetic resonance (fMRI) data. One problem with this method is that when the number of features increases, the results may be unstable, and the difference between the summarized gradient score of each feature may be minimal. This kind of problem makes it hard to interpret the results. Additionally, gradient-based methods may suffer from the influence of class-irrelevant variables (Wilming et al., 2022).

Recently, several model-agnostic explanation methods have been proposed (Fisher, Rudin, & Dominici, 2019; Lundberg & Lee, 2017; Ribeiro et al., 2016) which enlarge the toolbox for neuroimaging analysis. For example, permutation importance (PI) is applied to a memory-related study (Valentin, Harkotte, & Popov, 2020), which aims to highlight important frequency bands and channels from EEG signals. The idea of this model-agnostic method is to make explanations by perturbing inputs. Feature contributions are calculated in different strategies based on the model output. However, some of these methods have efficiency problems. Usually, these methods need a large amount of sampling process, i.e. repeating permute one feature to estimate the model prediction changes. For some feature importance calculation strategies, the computational cost increases significantly as the number of features increases. This efficiency problem will limit the use of these explanation methods in some scenarios, such as searchlight analysis (Kriegeskorte, Goebel, & Bandettini, 2006), in which multiple classifiers will be trained along time points. In this case, the computational cost for some model-agnostic methods is impracticable. Furthermore, recent research suggests that class-irrelevant features like suppressor variables may influence these explanation methods (Budding et al., 2021; Wilming et al., 2022). These class-irrelevant features may provide side information like noise reduction but do not provide study-related information.

3. Methods

3.1. Linear activation pattern

To address the problem caused by class-irrelevant variables, Haufe et al. (2014) proposed a method for linear model cases in neuroimaging analysis tasks by constructing the activation pattern. In their work, the m-dimensional observed data $\mathbf{X} \in \mathcal{R}^{n\times m}$ are assumed to be generated by some k-dimensional latent factors $\mathbf{L} \in \mathcal{R}^{n\times k}$ using specific patterns $\mathbf{W}_{pattern} \in \mathcal{R}^{m\times k}$ and model weights $\mathbf{W}_{linear} \in \mathcal{R}^{m\times k}$ which is:

Generative model: $\mathbf{X} = \mathbf{L}\mathbf{W}_{pattern}^{T} + \epsilon$

Discriminative model: $\mathbf{L} = \mathbf{X}\,\mathbf{W}_{linear}$,

where the ϵ represents the noise variables, and the discriminative model represents the classifier. The latent variables could be a brain process under study or different classes interested in analysis. The generative model shows how observations are generated by class-related latent variables, while its weights $\mathbf{W}_{pattern}$, called activation pattern, directly indicate variables related to class-related latent variables. This activation pattern is less affected by class-irrelevant variables than classifier weights.

This activation pattern for linear models can be reconstructed once we have the associated classifier. By assuming the latent variables are independent, and noise variables ϵ are uncorrelated with latent factors. We can obtain this reconstructed activation pattern as follows:

$$\mathbf{W}_{pattern} = \Sigma_{\mathbf{X}}\mathbf{W}_{linear}\Sigma_{L}^{-1},$$

where $\Sigma_{\mathbf{X}}$ and Σ_{L} represent the covariance matrix of observations $\mathbf{X}$ and latent variables $\mathbf{L}$.

If only one latent variable exists, i.e., k = 1 and $\mathbf{W}_{pattern}$, $\mathbf{W}_{linear}$ become an m-dimensional vector, such as in the binary classification case, the Σ_{L} is a constant, i.e., the variance of this latent variable, that can be ignored if our goal is to find important features. Therefore, the pattern can be simplified as follows:

$$\mathbf{W}_{pattern} \propto \Sigma_{\mathbf{X}}\mathbf{W}_{linear}.$$

The linear activation pattern combines classification information and data variance information. Various statistical techniques, such as factor analysis, are also available to discover latent components in data with associated patterns and measure the variance contribution of observed variables. These methods aim to obtain independent latent variables and better explain the data. Although the latent variables have the potential to contribute to the classification task, this is not guaranteed. The linear activation pattern is particularly suited for classification cases since it restricts the latent variable associated with the classification task.

We adapt the notion of linear pattern to kernel SVM models. The proposed method contains two steps: the first is to estimate the coefficient of the activation pattern in kernel space, and the second is to map the activation pattern back into input space.

3.2. Construct activation pattern in kernel space

To separate the data, SVM models aim to find a hyperplane, represented using data samples and associated coefficients. This idea also works for kernel-based SVM models, which map data into high-dimensional space to make the linearly inseparable data in input space separable in the high-dimensional kernel space. We can then construct an activation pattern in this kernel space.

Usually, the hyperplane of an SVM model can be written as a linear combination of support vectors and associated coefficients. For the kernel-based SVM model, assuming the dimension of kernel space is q which we do not know, this hyperplane $\mathbf{W}_{kernel} \in \mathcal{R}^{q}$ can be represented as $\mathbf{W}_{kernel} = \Sigma_i \alpha_i \Phi(\mathbf{x}_i)$, where the $\Phi(\mathbf{x}_i) \in \mathcal{R}^{q}$ represents the kernel space support vector. And α_i represents the associated coefficient. For the convenience of representing equations, we will rewrite them in matrix format.

Assume $\mathbf{S}^{T} = [\Phi(\mathbf{x}_1), \Phi(\mathbf{x}_2), \dots, \Phi(\mathbf{x}_s)]$ represents the mapped support vector, i.e., $\mathbf{S} \in \mathcal{R}^{s\times q}$, and the number of support vectors is $\mathbf{s}$, and α is the associated coefficient vector. The hyperplane can be shown as follows:

$$\mathbf{W}_{kernel} = \mathbf{S}^{T}\alpha \tag{1}$$

The covariance matrix of mapped data can also be represented in this matrix format, which is as follows:

$$\begin{aligned}\Sigma_{\phi(\mathbf{x})} &= \frac{1}{n}\mathbf{F}^{T}\mathbf{H}\,(\mathbf{F}^{T}\mathbf{H})^{T}\\ &= \frac{1}{n}\mathbf{F}^{T}\mathbf{H}\mathbf{H}^{T}\mathbf{F}\\ &= \frac{1}{n}\mathbf{F}^{T}\mathbf{H}\mathbf{F}\end{aligned} \tag{2}$$

where the $\mathbf{F} \in \mathcal{R}^{n\times q}$ and $\mathbf{F}^{T} = [\Phi(\mathbf{x}_1), \Phi(\mathbf{x}_2), \dots, \Phi(\mathbf{x}_n)]$ represents the mapped data matrix with the number of samples is $\mathbf{n}$. We do not know whether the mapped data is centered, which is required when calculating the covariance. So we introduce the centering matrix $\mathbf{H}$, which is $\mathbf{H} = \mathbf{I_n} - \frac{1}{\mathbf{n}}\mathbf{1_n}$, where the $\mathbf{I}$ is $\mathbf{n}$ dimensional identity matrix and $\mathbf{1_n}$ is the n-by-n matrix of all 1. $\mathbf{F}^{T}\mathbf{H}$ can be seen as the sample minus mean step while calculating variance/covariance. Centering matrix has a good property which $\mathbf{H}\mathbf{H}^{T} = \mathbf{H}$. This has been applied in Eq. (2). One thing to note is that usually, the data will be scaled before training the SVM model. The data used here is the actual data used for training and testing, i.e. the scaled data if using the scaler. Combining the equations above, the pattern in kernel space $\mathbf{W}_{kernel-pattern} \in \mathcal{R}^{q}$ can be constructed as:

$$\begin{aligned}\mathbf{W}_{kernel-pattern} &= \Sigma_{\phi(\mathbf{x})}\,\mathbf{W}_{kernel}\\ &= \frac{1}{n}\mathbf{F}^{T}\mathbf{H}\mathbf{F}\mathbf{S}^{T}\alpha\end{aligned} \tag{3}$$

where $\mathbf{F}\mathbf{S}^{T} = [k(\mathbf{x}_i, \mathbf{x}_j)]_{n*s}$ is a $(n * s)$ dimensional matrix, which can be seen as the inner product between each mapped data sample and each support vectors. This inner product can be calculated using the kernel function used in the classifier, which is represented as $k(\mathbf{x}_i, \mathbf{x}_j)$.

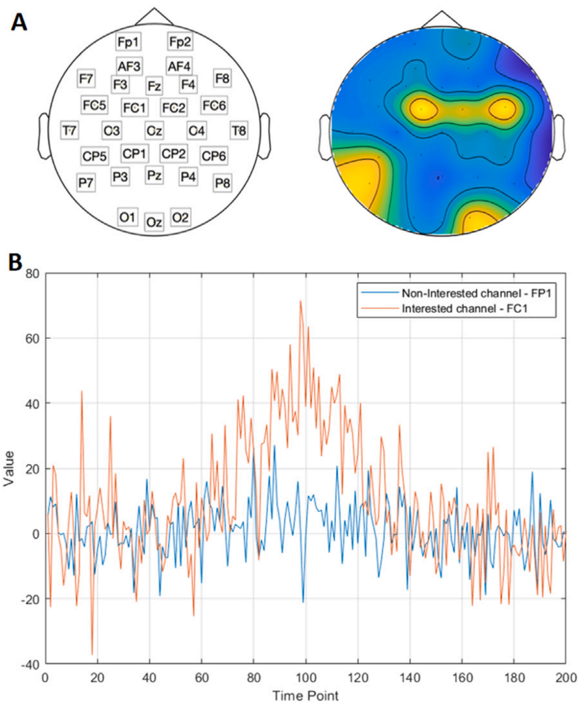

Fig. 1. Figure A show the channel position of the simulation data and an example channel weights of a simulation dataset used in experiment 1. The bright color indicates a higher weight than the dark color. Figure B shows some signals of interested and non-interested signals. The right line chart shows the averaged event-related potential for the 2 channels: FP1, which is selected as an interested channel, and FC1, which is not interested.

We should note that the current pattern $\mathbf{W}_{kernel-pattern}$ is in high-dimensional space, which we do not know specifically. However, this is still a combination of mapped data $\mathbf{F}$. The combination coefficient vector $\mathbf{P} \in \mathcal{R}^n$ can easily calculate:

$$\text{Kernel Pattern Coefficient: } \mathbf{P} = \frac{1}{n}\mathbf{HFS}^T\alpha \tag{4}$$

and $\mathbf{W}_{kernel-pattern}$ is rewritten as:

$$\mathbf{W}_{kernel-pattern} = \mathbf{F}^T\mathbf{P} \tag{5}$$

3.3. Mapping method: fixed point iteration

For the kernel-based method, there is no need to specify how a data point is mapped into kernel space. This mapping process is not required in the later calculation. For training and making a prediction, a kernel-based SVM model only uses the inner product of two mapped samples, which is defined by the kernel function $\mathbf{k}(\mathbf{x}_1, \mathbf{x}_2)$. As mentioned above, the reconstructed activation is in the kernel space, and it is hard to calculate the input space results of a known kernel space vector. This problem can be solved by applying a pre-image technique (Honeine & Richard, 2011; Kwok & Tsang, 2004). These methods can find input space results of a kernel target, based on some constraints such as minimizing the MSE between the results and kernel target, which have been used for kernel PCA denoising (Mika et al., 1998).

With the constructed kernel activation pattern above, the pre-image method can estimate the associated result $\mathbf{x}^*$ in input space by minimizing the mean squared error (MSE) distance between $\mathbf{x}^*$ and the target $\mathbf{W}_{kernel-pattern}$ in kernel space, which is

$$\arg\min_{\mathbf{x}^*} \text{MSE}(\mathbf{x}^*) = ||\mathbf{W}_{kernel-pattern} - \phi(\mathbf{x}^*)||^2$$

Furthermore, the mean squared distance is as follows:

$$\begin{aligned}\text{MSE}(\mathbf{x}^*) &= ||\,\mathbf{W}_{kernel-pattern} - \phi(\mathbf{x}^*)\,||^2\\ &= (\mathbf{F}^T\mathbf{P} - \phi(\mathbf{x}^*))^T(\mathbf{F}^T\mathbf{P} - \phi(\mathbf{x}^*))\\ &= \mathbf{P}^T\mathbf{F}\mathbf{F}^T\mathbf{P} - \phi(\mathbf{x}^*)^T\mathbf{F}^T\mathbf{P} - \mathbf{P}^T\mathbf{F}\phi(\mathbf{x}^*) + \phi(\mathbf{x}^*)^T\phi(\mathbf{x}^*)\end{aligned}$$

where $\mathbf{P}$ represent the kernel pattern coefficient as shown in Eq. (4); $\phi(\mathbf{x}^*)$ represent the mapped result $\mathbf{x}^*$ in kernel space. Note that $\mathbf{P}^T\mathbf{F}\mathbf{F}^T\mathbf{P}$ is clearly a fixed constant, i.e., the inner product of the activation pattern. This will not change once the classifier is trained. The two middle fractions, $\phi(\mathbf{x}^*)^T\mathbf{F}^T\mathbf{P}$ and $\mathbf{P}^T\mathbf{F}\phi(\mathbf{x}^*)$ is the same. $\phi(\mathbf{x}^*)^T\mathbf{F}^T$ and $\mathbf{F}\phi(\mathbf{x}^*)$ represent the inner product between the mapped result $\phi(\mathbf{x}^*)$ and each of the mapped data samples. After multiplying the coefficient vector $\mathbf{P}$, whether $\phi(\mathbf{x}^*)^T\mathbf{F}^T\mathbf{P}$ or $\mathbf{P}^T\mathbf{F}\phi(\mathbf{x}^*)$ can be regarded as the sum of the inner product multiplied by the corresponding coefficient. For convenience, these 2 middle parts are rewritten as a sum-up format later as $\phi(\mathbf{x}^*)^T\mathbf{F}^T\mathbf{P} = \mathbf{P}^T\mathbf{F}\phi(\mathbf{x}^*) = \Sigma_i^n \mathbf{p}_i k(\mathbf{x}^*, \mathbf{x}_i)$. The last fraction, $\phi(\mathbf{x}^*)^T\phi(\mathbf{x}^*)$ represents the inner product of the mapped result, which can be calculated using kernel function as $k(\mathbf{x}^*, \mathbf{x}^*)$.

The function of minimizing the MSE error can be simplified as follows:

$$\arg\min_{\mathbf{x}^*} \text{MSE}(\mathbf{x}^*) = k(\mathbf{x}^*, \mathbf{x}^*) - 2\Sigma_i^n \mathbf{p}_i k(\mathbf{x}^*, \mathbf{x}_i) \tag{6}$$

As mentioned before, we do not need to know the exact mapping process, just like we can classify samples using kernel SVM models without needing to know the exact mapping process.

A variety of random searching methods can solve this problem. Nevertheless, the large search space of these methods requires many computation costs. Here we use the fixed-point iteration method to search the results. Compared with random searching methods, the fixed-point iteration method limited the search space, which decreased the computation cost. Furthermore, the result of this method will have the same scale as the input vectors have (Honeine & Richard, 2011), which makes the results more straightforward when using the pattern to explain the model.

The overall goal is the same for different kernels, as shown in Eq. (6). By setting the derivative of Eq. (6) for $\mathbf{x}^*$ to zero, we can easily obtain a fixed-point iteration format. The exact fixed-point iteration format equation is different depending on the kernel function used.

When using RBF kernel function

$$k(\mathbf{x}_1, \mathbf{x}_2) = \exp\frac{||\mathbf{x}_1 - \mathbf{x}_2||^2}{\gamma},$$

the fixed-point iteration format becomes

$$\begin{aligned}\text{MSE}(\mathbf{x}^*) &= \exp\frac{||\mathbf{x}^* - \mathbf{x}^*||^2}{\gamma} - 2\Sigma_i^n \mathbf{p}_i \exp\frac{||\mathbf{x}_i - \mathbf{x}^*||^2}{\gamma}\\ &= 1 - 2\Sigma_i^n \mathbf{p}_i \exp\frac{||\mathbf{x}_i - \mathbf{x}^*||^2}{\gamma}\end{aligned}$$

By setting the derivative for $\mathbf{x}^*$ to zero, we can obtain a fixed-point iteration method:

$$\mathbf{x}_{t+1} = \frac{\Sigma_i^n \mathbf{p}_i k(\mathbf{x}_i, \mathbf{x}^*)\,\mathbf{x}_i}{\Sigma_i^n \mathbf{p}_i k(\mathbf{x}_i, \mathbf{x}^*)} \tag{7}$$

Similarly for polynomial kernel, which the kernel function is $k(\mathbf{x}_1, \mathbf{x}_2) = (\mathbf{x}_1^T\mathbf{x}_2 + \mathbf{c})^d$ where the $\mathbf{d}$ is the degree of kernel and $\mathbf{c}$ is constant term. The fixed-point iteration format is as follows:

$$\mathbf{x}_{t+1} = \frac{\Sigma_i^n \mathbf{p}_i(\mathbf{x}_i^T\mathbf{x}_t + \mathbf{c})^{d-1}\,\mathbf{x}_i}{(\mathbf{x}_t^T\mathbf{x}_t + \mathbf{c})^{d-1}} \tag{8}$$

And for sigmoid kernel, which the kernel function is $k(\mathbf{x}_1, \mathbf{x}_2) = tanh(\mathbf{x}_1^T\mathbf{x}_2 + \mathbf{c})$ where the $\mathbf{c}$ is constant term. The fixed-point iteration format is:

$$\mathbf{x}_{t+1} = \frac{\Sigma_i^n \mathbf{p}_i(1 - tanh^2(\mathbf{x}_i^T\mathbf{x}_t + \mathbf{c}))\,\mathbf{x}_i}{1 - tanh^2(\mathbf{x}_t^T\mathbf{x}_t + \mathbf{c})} \tag{9}$$

3.4. Our proposed explanation algorithm

Section 3.2 above derives the use of activation pattern to explain the prediction of nonlinear SVM models. This sub-section presents the proposed method into an algorithm as shown below.

The algorithm can be seen as two parts: calculate the coefficient of the estimated pattern in kernel space and map back this pattern into input space. The pattern is estimated in kernel space and using support vectors in kernel space. The first step is to calculate its associated coefficient. The second step is an optimizing procedure.

Algorithm 1: Algorithm for EAP

```
Input:
    • NI: Maximum iteration number
    • $\alpha$: Coefficients of the support vectors
    • $\mathbf{S}^T = [\mathbf{x}_1, \ldots, \mathbf{x}_\mathbf{s}]$: Support vectors
    • $\mathbf{X}^T = [\mathbf{x}_1, \ldots, \mathbf{x}_\mathbf{n}]$: Training set
    • mch: The minimum changes
Output:
    • $\mathbf{x}^*$: Explanation vector
// Calculate coefficient of pattern in kernel space
Calculate centre matrix $\mathbf{H} = \mathbf{I}_n - \frac{1}{n}\mathbf{1}_n$. ;
for $i = 1{:}n$ do
    for $j = 1{:}s$ do
        Calculate the element $\mathbf{FS}^T_{ij} = \mathbf{k}(\mathbf{x}_i, \mathbf{x}_j)$. The kernel function depends on the kernel used in the SVM model. ;
        j++ ;
    i++ ;
Calculate the coefficient of estimated pattern $\mathbf{P} = \frac{1}{n}\mathbf{HFS}^T\alpha$. ;
// Mapping the estimated pattern into input space
$\mathbf{t} \leftarrow 0$ ;
$\mathbf{x}^*_0 \leftarrow$ initialised from standard normal distribution;
$\mathbf{diff}_{t-1} \leftarrow 1$ /*difference between $\mathbf{x}^*_{t-1}$ and $\mathbf{x}^*_t$/;
$\mathbf{diff}_{t-2} \leftarrow 1$ /*difference between $\mathbf{x}^*_{t-2}$ and $\mathbf{x}^*_t$/;
while $(\mathbf{t} < \mathbf{NI})$ AND (All $|\mathbf{diff}_{t-i}| > \mathbf{mch}$) do
    Update $\mathbf{x}^*_t$ with equation (7) or (8) or (9), depending on the kernel type. ;
    Update the $\mathbf{diff}_{t-1} \leftarrow \frac{|\mathrm{MSE}(\mathbf{x}^*_{t-1}) - \mathrm{MSE}(\mathbf{x}^*_t)|}{\mathrm{MSE}(\mathbf{x}^*_{t-1})}$ ;
    Update the $\mathbf{diff}_{t-2} \leftarrow \frac{|\mathrm{MSE}(\mathbf{x}^*_{t-2}) - \mathrm{MSE}(\mathbf{x}^*_t)|}{\mathrm{MSE}(\mathbf{x}^*_{t-2})}$
    $\mathbf{t} \leftarrow \mathbf{t} + 1$ ;
```

Based on the results of multiple experiments, in our experiments, the initialization of $\mathbf{x}^*_0$ is initialized from the standard normal distribution, and the number of iterations is set to 1000. Another stop iteration strategy is the MSE changes between the results $\mathbf{x}^*_t$, $\mathbf{x}^*_{t-1}$ and $\mathbf{x}^*_{t-2}$. Any of these changes smaller than the threshold will cause the iteration to stop. The threshold of these changes is set to 0.001 empirically based on experiments. Besides, this threshold is reasonable; if the changes are below this standard, the changes do not significantly influence the final results.

The initialized starting result $\mathbf{x}_0$ may affect the fixed-point iteration method. In our experiments, the starting results are initialized from the standard normal distribution because of the standard scaler. Multiple runs using different starting results may be needed. Other optimization methods like the momentum-based method may also help in severe cases.

4. Experiments

This section presents three experiments to evaluate the EAP method against three popular explanation methods. The first two experiments are performed on two simulated electroencephalography (EEG) datasets. Experiment 3 is performed on a publicly available EEG/MEG dataset, Wakeman & Henson dataset (Wakeman & Henson, 2015), on a visual face perception task. SVM classification models using RBF, polynomial and sigmoid kernel are trained on each dataset. EAP and three benchmark explanation methods are then used to explain the predictions of those trained models.

4.1. Datasets

4.1.1. Simulation dataset

The overall simulation dataset setting is similar for experiments 1 and 2. The main difference is the interested channel selection strategies. The simulated EEG signal is Event-Related Potential (ERP) like simulations, which can be seen as the epoched and preprocessed real EEG signal. The simulation datasets are simulated using a MATLAB toolbox, MVPA-Light (Treder, 2020). Each dataset has 1000 samples divided into two classes. Each sample has 30 channels and 200 time points. For each sample, one peak is simulated.

The difference in simulation datasets for experiment 1 and 2 is how to set the channel weights. Channel weights decide the signal changes of each channel in the peak. Interested channels will have larger weights than other non-interested channels, which means that those channels with large weights will react significantly despite the noise. The channel weights reflected the spatial information of simulation datasets, which can be seen as the ground truth. The goal is to recover this information with the help of explanation methods.

Experiment 1. The channel weights are initialized with random numbers sampled from the standard normal distribution. Then six interested channels are randomly selected. Additional weight is added to each of the interested channels. This additional weight contains a fixed and random number sampled from standard uniformed distribution. This can ensure that the additional weight is significantly larger than the initialized value while maintaining a slight difference between different channels. Samples are divided into two classes based on average peak values. Both classes have roughly the same mean value during peak, while one class contains more samples close to the mean peak value and another contains more extreme cases, then adds strong Gaussian noise. An example is shown in Fig. 1.

Experiment 2. The simulation dataset in experiment 2 contains 3 parts: **signal**, **distractor**, and **noise**. The weights of the **signal**, and **distractor**, or called the patterns, are shown in Fig. 3A. Two classes are divided depending on **signal**, and like in experiment 1, the average signal of the two classes is roughly the same. One class sets the signal weights as $1 \times \mathit{signal\ pattern}$ and $-1 \times \mathit{signal\ pattern}$, while another class sets as $0.5 \times \mathit{signal\ pattern}$ and $-0.5 \times \mathit{signal\ pattern}$. The **distractor**, and **noise** both represent the non-interested signals, while the **distractor** show overlapped signals. To mimic the signal changes, for each sample, the distractor weights multiply a random number from the standard normal distribution. This distractor signal is irrelevant to the class label and will weaken the interested signal. This is a simulation of those class-irrelevant variables like suppressor variables. The **noise** acts as strong background Gaussian noise. The proportion of the signal combination is: $\mathbf{X} = 0.25 \times \mathbf{signal} + 0.25 \times \mathbf{distractor} + 0.5 \times \mathbf{noise}$.

4.1.2. Real dataset

The Wakeman & Henson dataset used in this experiment is an MEG/EEG dataset of visual face perception tasks (Wakeman & Henson, 2015): in this dataset, participants are asked to see pictures of famous faces, unfamiliar faces, and scrambled faces during the recording period. Signals are measured using Elekta Neuromag Vectorview 306

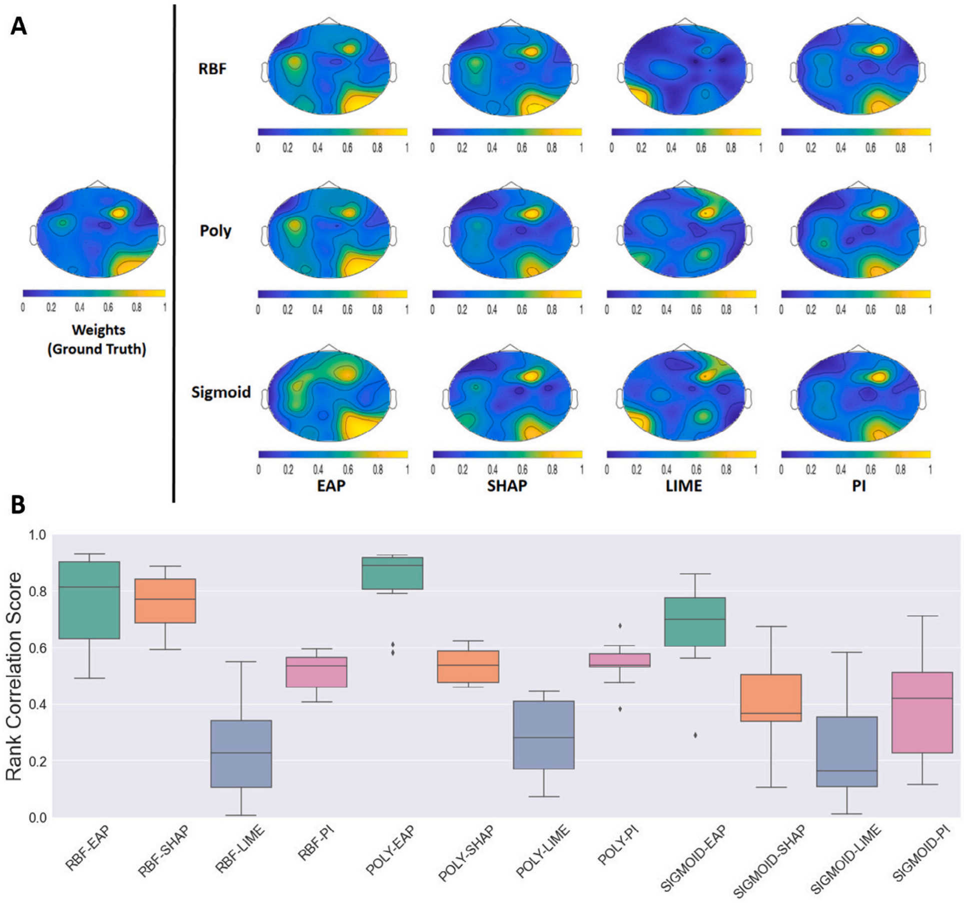

Fig. 2. Figure A shows the explanation results of one randomly picked simulation dataset. The left topograph in Figure A shows the actual channel weights, which is the ground truth of this simulation dataset. For the convenience of comparison, all results are scaled from 0 to 1 before drawing topographies. Figure B shows the rank correlation score. Since our goal is to highlight the important channels, the correlation scores are calculated between the absolute value of explanation results and the absolute value of actual weights.

system in a light magnetically shielded room. Sixteen participants are involved in the dataset. The number of trials in each of the 3 classes is around 290. In our experiment, only signals of the famous faces task and scrambled face task are selected in this study.

Preprocessing. Irrelevant channels, such as ECG and EOG channels, are removed first. Then, to highlight the interested signal and reduce the noise, a bandpass filter is applied between 1 Hz to 40 Hz with windowed-sinc Finite Impulse Response (FIR) filters. The EEG data is re-referenced using the average reference method, i.e., subtracted by the average signal of all channels. This step will help highlight the interested signals. To save computation costs, data are then down-sampled into 220 Hz. Since many trials are logged continuously, trials are segmented based on the event file provided by the dataset. This step ensures that the time points in each sample are aligned relative to the condition, i.e., seeing the picture. Each trial maintains 0.5 s before participants see pictures and 1 s after. Baseline correlation is applied based on a time window from 0.5 s to 0 s before the stimulus. 70 EEG and 102 magnetometer channels are then selected as classification features. We directly use sensor-level data, so the explanation results should also reflect spatial information at the sensor level. All preprocessing tasks are carried out using Fieldtrip and MVPA-Light toolbox (Treder, 2020) on MATLAB.

Unlike the simulation dataset, several peaks are detected in this dataset. This means that several reactions are logged, and we should identify them. Two interested time intervals are identified based on local minima in the Global Field Power (Skrandies, 1990), a method that quantifies the amount of activity at each time point. The local minima of Global Field Power are similar for EEG and MEG data. The selected two time intervals, 80–135 ms and 145–190 ms, are identified for use in this experiment.

4.2. Classification

All experiments use the mean of each channel within selected time intervals as the classification feature. For experiment 1 and 2, the selected time interval is the peak area. For experiment 3, the selected time intervals are the two selected time intervals.

RBF kernel, Polynomial kernel, and sigmoid kernel are performed separately. For each simulation dataset/participant, the classifier's hyperparameters are optimized based on 5-fold cross-validation under the same hyperparameter searching range. Then build classifiers and implement several explanation methods.

4.3. Selected explanation methods

The EAP method is evaluated against three selected state-of-the-art model agnostic explanation methods: permutation importance, local interpretable model agnostic explanation (LIME), and Shapley additive explanations (SHAP). All these methods are implemented using the default parameter setting in their code package.

Permutation importance (PI) is firstly proposed by Breiman (2001), to measure the feature contribution of the random forest model. Fisher et al. (2019) introduces a model-agnostic framework for permutation importance called model reliance. The idea of this approach is to assign an importance score for each feature. Importance scores are calculated based on measuring the changes in model function results after shuffling one feature value while fixing other feature values.

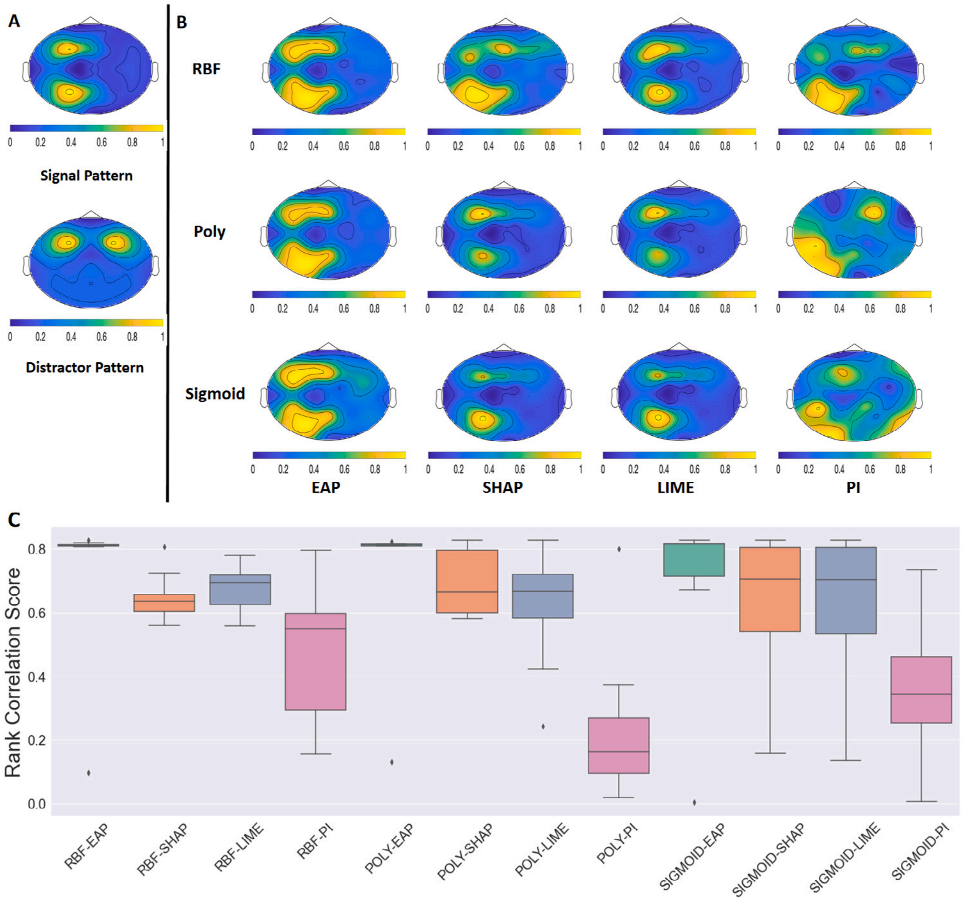

Fig. 3. The left topograph in Figure A shows the channel weights of the signal pattern, which is the ground truth, and the channel weights of the distractor pattern. Since the distractor pattern is irrelevant to the 2 classes, the explanation methods are expected to highlight the channels shown in the signal pattern and not highlight the channels shown in the distractor pattern. The right topographies in Figure A shows the average results of the explanation results. For the convenience of comparison, all results take absolute value and then scaled from 0 to 1. Under the use of different kernels, the EAP method is less influenced by class-irrelevant variables than the other three methods. Figure B shows the rank correlation score between the explanation results and the absolute value of actual weights. The performance of the EAP method in different kernel settings is more stable than the other three methods.

This method provides a global view of explaining current models. An advantage of this method is that it has a low computation cost.

The local interpretable model-agnostic explanation (LIME) (Ribeiro et al., 2016) is an instance-based model-agnostic explanation method. LIME first samples around a single instance and weighs new samples by their similarity compared with the associated instance. Then learn a linear surrogate model for this instance, trying to approximate the properties near this instance. The weight of this surrogate model is the explanation score assigned for this instance. It explains the current instance by approximating its feature gradient. In our experiment, all samples in the training set are explained.

The Shapley value (Shapley, 1952), which comes from game theory, is a method that can measure feature influence for a model in a real-world setting. Shapley additive explanations (SHAP) (Lundberg & Lee, 2017) is a model-agnostic framework that can approximate Shapley value for a single instance, which is increasingly popular. Like LIME, all training samples are explained to make a global explanation.

The explanation results may be in different scales, and our goal is to find the most important features. For the convenience of comparison, all explanation results will take absolute value and be scaled between 0–1 using the min–max scaling method.

5. Results analysis and discussion

Explainable machine learning can be categorized as global and local methods depending on their scope of explainability. Global methods provide an overall explanation of model behavior on features in the data collectively affecting the prediction, local methods explain why the model makes a certain prediction for an instance. For those methods in the evaluation, the EAP method and PI provide a global explanation, and LIME and SHAP provide a local explanation for each instance. To make the evaluation results comparable, the results in this section are presented in a global explanation scope, i.e., the results explain the overall feature importance to the prediction. We follow the commonly used aggregation approach for LIME and SHAP to yield a global explanation using averaged absolute results. For the EAP method and PI, the results are also using the absolute value of their results.

All results are shown and compared in two formats: topography and ranked correlation score. Topography is a kind of figure that map the channel signals to the associated position of the head. For experiment 1, the topographies only show an example of the experiment. For experiment 2 and Wakeman & Henson dataset, since different dataset has the same task, i.e. in experiment 2, all datasets have the same channel weights and all Wakeman & Henson datasets have the same visual task, topographies are shown the averaged results. For the convenience of comparison, all results will be rescaled between 0 to 1 by a min–max scaler. For the simulation datasets, since we have the generation channel weights, we can calculate the rank correlation score to compare the explanation results. Our goal is to highlight the important channels, so all methods will use the absolute value. Since different methods may have different scales, all the results are rescaled between 0 to 1 using a min–max scaler after taking the absolute value. Then rank correlation scores are calculated between the results and the absolute value of

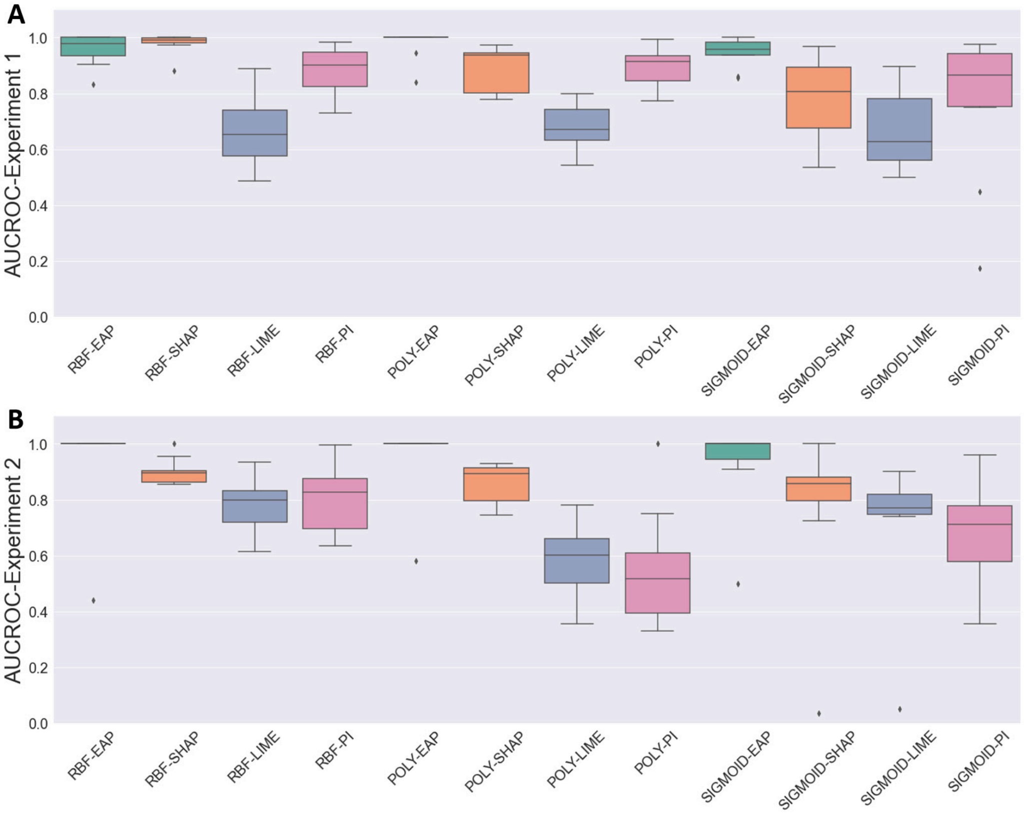

Fig. 4. Area Under the Curve of the Receiver Operating Characteristic curve (AUCROC) for comparing explanation results to the ground truth. The channel weights value is replaced by a binary set of 0, 1 as the ground truth. Figure A shows the AUCROC results for experiment 1, while Figure B shows the results for experiment 2. For most experiments, EAP methods show lower variance and outperformed others.

generation channel weights, the randomly selected channel weights in experiment 1, and the signal pattern in experiment 2. To measure our results from the precision perspective, the area under the curve of the receiver operating characteristic curve (AUCROC) is involved. Here, the ground truth is a binary set calculated based on channel weights.

5.1. Simulation: Experiment 1

The goal is to test whether these explanation methods can pick important features out correctly. Results are shown in Fig. 2. As one example shown in 2A, all methods can highlight the important channels while the importance score varies. As shown in Fig. 2B, the EAP method outperformed the other three methods in experiment 1. SHAP can also obtain relatively good results, but this method requires much more runtime to obtain global explanation results. The AUCROC results, shown in Fig. 4A indicate the same results.

5.2. Simulation: Experiment 2

Fig. 3 shows the results of experiment 2. Fig. 3B shows the average value of the absolute explanation results. Fig. 3C gives the rank correlation score between the absolute explanation results and the absolute value of the signal pattern. The distractor pattern can be seen as a large noise that is irrelevant to the 2 classes. All explanation methods are expected to highlight the features that contain class-related information which are shown in the signal pattern. Furthermore, features shown in the distractor pattern should not be highlighted in the explanation results. As shown in Fig. 3B, from the mean value view of all explanation results, all methods can roughly pick out the important channels as shown in the signal pattern. The result of the EAP method may involve more channels compared to the signal pattern. However, the other three methods are influenced more by the distractor signal. The results displayed in Fig. 4B also provide evidence to support this from the AUCROC perspective. In experiment 2, the EAP method shows more stable results than the other three methods. Fig. 5 gives the results of one simulation dataset, which indicate similar results as in Fig. 3.

5.3. Real data: experiment 3

The results of EEG data are shown in Fig. 6 and MEG in Fig. 7. All explanation results are averaged among sixteen participants first, then rescaled to 0 to 1.

The two selected time intervals reflect the previous study, which is consistent with the two components of the brain cognitive process in visual tasks: the P100 and N170. These two components have been reported in many previous studies (Boutros et al., 1997; Kropotov, 2016). Unlike simulation datasets, we do not have the exact ground truth of Wakeman & Henson dataset. Using similar time intervals allows us to verify the explanation results in this study against the reported results of previous vision task studies.

For the explanation results of EEG data (Fig. 6), in the first time interval, which can be seen as the P100 component, channels at all occipital areas and inferior occipitotemporal area, which is the back head area, are highlighted by EAP method. In contrast, other methods highlight the channels at the center occipital areas. For the P100 component, signal changes are found in most channels in the occipital area and some in the inferior occipitotemporal area (Herrmann, Ehlis, Ellgring, & Fallgatter, 2005; Negrini, Brkić, Pizzamiglio, Premoli, & Rivolta, 2017) in the previous study. In the second time interval, all methods gave a high score to the channels located in the right back head, which is the occipitotemporal area in the right hemisphere, and related weak scores to the channels in the left hemisphere. However,

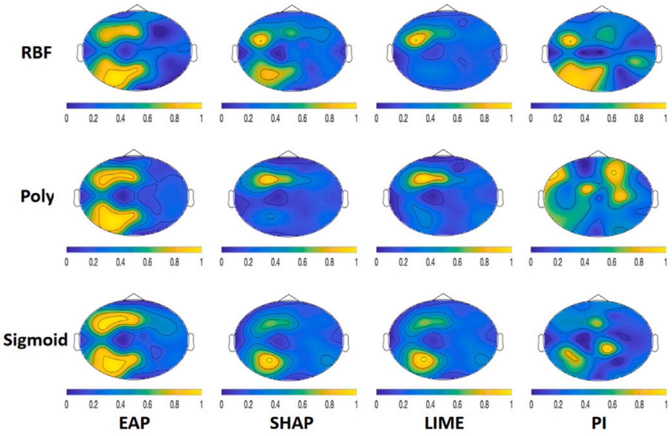

Fig. 5. This figure shows an example in experiment 2. Topographies of explanation results for different kernels are listed. For the convenience of comparison, all results are scaled from 0 to 1.

the other three methods also highlight the channels in the center back head, the center occipital area. The second time interval can be seen as the N170 component. In previous findings (Bentin, Allison, Puce, Perez, & McCarthy, 1996), the signals for the face are more significant than non-face stimulus. These signal differences can be found in channels located in the occipitotemporal area in both hemispheres, the left and right back head in the topography, while not find significant signal differences in channels at center occipital area (Negrini et al., 2017; Rossion & Jacques, 2008). Furthermore, compared with non-face stimulus, channels at the right occipitotemporal area will detect a larger difference with face stimulus (Eger, Jedynak, Iwaki, & Skrandies, 2003; Maurer, Rossion, & McCandliss, 2008; Wang et al., 2019). As a result, the EAP results are more consistent with the previous finding in both time intervals.

For the explanation results of MEG data (Fig. 7), all methods highlight the channels located at the midline occipital area and right occipitoparietal area in the first time interval. However, the EAP results of poly and sigmoid kernel experiments also highlighted channels in the left occipitotemporal and right occipital areas. In the RBF and poly kernel experiment, SHAP, LIME, and PI also highlight the left occipitotemporal area. In previous findings, the P100 component, which is usually called M100 in MEG, some studies report the signal difference between face and non-face stimulus is found in the occipitotemporal of both hemispheres (Liu, Harris, & Kanwisher, 2002; Xu, Liu, & Kanwisher, 2005). In contrast, the signal differences are not significant. However, in Tanskanen, Näsänen, Montez, Päällysaho, and Hari (2005) channels located in the midline occipital area, locate the front-mid position of the back head. Some studies (Halgren, Raij, Marinkovic, Jousmäki, & Hari, 2000; Susac, Ilmoniemi, Pihko, Nurminen, & Supek, 2009) support this finding but also report that channel level signal differences are found in channels located at the right occipitotemporal area while not the significant signal difference in channels located at the left occipitotemporal area.

For the results in the second time interval, SHAP, LIME, and PI give similar results, highlighting both the midline occipital and right occipitoparietal areas. While the EAP method highlights the channels located in the right temporal area, left occipitotemporal area, and right occipital area. In the previous study, signal differences are found in channels in the inferior occipitotemporal area (Liu et al., 2002; Xu et al., 2005). While some study also reports signal differences are also found in channels located in the inferior parietal and the middle temporal area (Lu et al., 1991; Susac et al., 2009). A recent study (Tadel et al., 2019) analyzed the same dataset as we used, which reported the signal difference in channels located at the right temporal area, right occipital area, and left frontal area at 151 ms, and channels located at the right temporal area and left occipitotemporal area on 202 ms. Some MEG source analysis report similar results in the view of source position analysis (Meeren, de Gelder, Ahlfors, Hämäläinen, & Hadjikhani, 2013; Takeda, Suzuki, Kawato, & Yamashita, 2019). In previous studies, channel-level signal differences are usually found in both hemispheres. In general, the results of EAP result are more consistent with previous findings.

5.3.1. *Empirical computational cost*

This sub-section provides an empirical comparison of computational cost between the EAP method and the three benchmark methods. The experiment is carried out on both the simulation dataset and Wakeman & Henson dataset. The run time is measured over a desktop with i7 9700k CPU with 32 GB RAM. The operating system is Ubuntu 20.04. Table 1 provides a summary of the run time of the methods.

Table 1 shows that SHAP is the slowest and our EAP is the fastest. Overall, EAP is a few times faster than PI and several orders of magnitude faster than LIME and SHAP. SHAP requires a much longer run time than LIME for assigning explanations to all the samples due to the difference in permutation procedure. Moreover, the number of features also increases the run time of LIME and SHAP since a greater number of features makes the permutation procedure exponentially less efficient. In contrast, PI and our EAP provide a global explanation, which is made through the model itself rather than through samples. The computation cost of PI and EAP is much lower than LIME and SHAP.

Although LIME and SHAP require a long run time, they have an advantage in providing an explanation for each instance, which makes them more flexible in understanding the model at an instance level. For cases of obtaining a global understanding, however, the excessively long run time of LIME and SHAP hinders their application, and EAP is clearly preferred.

6. Conclusion

In neuroimaging analysis tasks, understanding what the model has learned is as important as the prediction accuracy of the learned model. This paper presents the EAP explanation method for nonlinear kernel-based SVM models in the analysis of neuroimaging data. This method can be adapted to several different kernels. The EAP method provides an importance score for each variable based on the reconstruction of an activation pattern, a widely used technique in this field for explaining linear models in neuroimaging analysis tasks. It can highlight variables with information in concern and is less affected by noise and class-irrelevant variables. The EAP method is evaluated against three state-of-art explanation methods: PI, LIME, and SHAP explanation methods on simulated data and Wakeman & Henson dataset. Experimental results show that the EAP method can successfully identify interested features of simulated data in SVM models with different kernels. The results also indicate the stability of the EAP method: the results of the EAP method from different kernels are given similar and correct results. In experiment 3, the EAP method provides consistent agreement with the human understanding of the brain's electrical activity in response to stimuli on the real EEG/MEG data. In addition, the proposed EAP method is significantly faster than other methods, multiple orders of magnitude faster than SHAP.

The experiment results indicate that the other three methods have been influenced more by class-irrelevant variables than the EAP method, which is potentially caused by the permutation process when facing feature-dependent cases. One possible explanation, from the data variance perspective, is that the total variance of the permutation process is larger compared to the original data. However, this excess variance can increase the probability of impossible instances outside of the original data space when permuted. The EAP method considered the data variance information, which was less affected by this case. Another problem we encountered is that the result scales produced by different methods varied. For the convenience of comparison, we

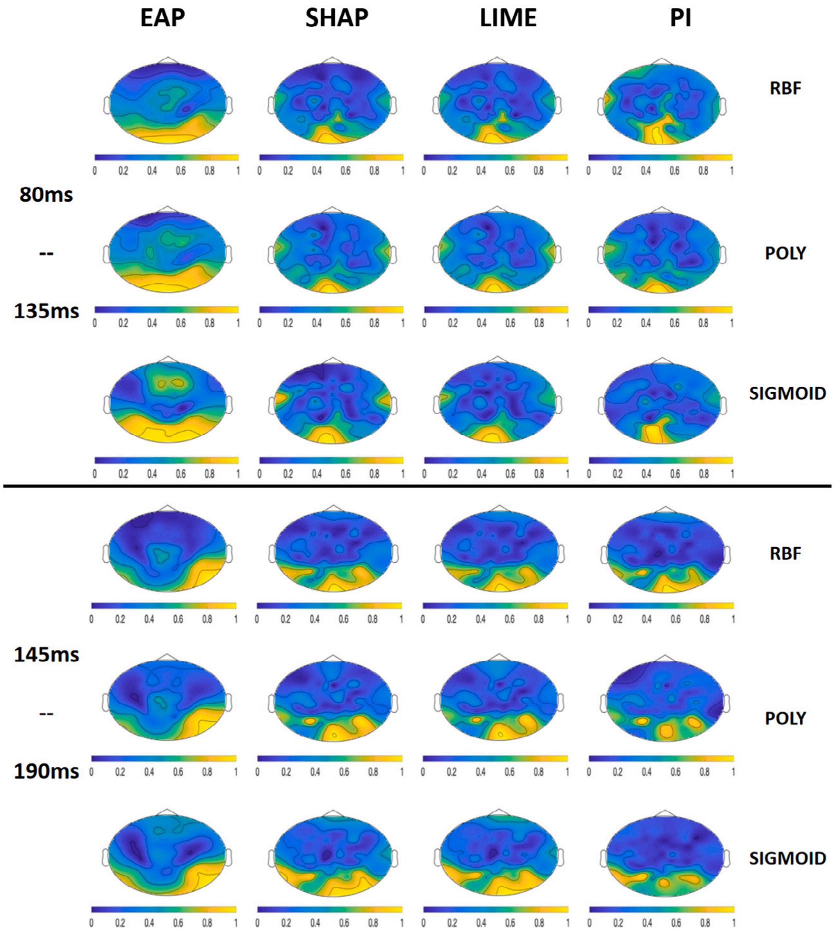

Fig. 6. Topographies of explanation results for EEG datasets. The results are averaged value of explanation results among 16 participants. For the convenience of comparison, all results are scaled from 0 to 1 before drawing topographies.

Table 1
Average run time (in seconds) of all methods. EEG-first (EEG-1) time interval and MEG-first (MEG-1) time interval refers to the time range of 80–135 ms, and EEG-second (EEG-2) time interval and MEG-second (EEG-2) time interval refers to the time range of 145–190ms.

	RBF				Polynomial				Sigmoid			
	EAP	PI	LIME	SHAP	EAP	PI	LIME	SHAP	EAP	PI	LIME	SHAP
Simulation 1	2.24	4.03	75.79	18 414.00	0.21	1.89	52.73	7132.36	1.78	6.44	141.16	36 286.68
Simulation 2	2.52	11.65	158.74	31 651.03	0.38	2.38	69.163	13 258.65	1.18	5.38	91.97	21 544.58
EEG-1	2.30	10.93	128.89	10 942.02	0.30	3.63	73.66	6596.95	0.45	5.713	120.55	8098.00
EEG-2	3.32	9.18	125.97	9946.59	0.26	3.56	67.57	5834.21	0.45	5.83	118.61	8798.29
MEG-1	2.31	9.55	115.75	10 614.00	0.26	5.94	88.87	7471.90	0.37	8.73	153.06	9998.29
MEG-2	2.66	13.15	101.67	9639.90	0.22	4.56	80.47	5729.55	0.30	6.78	166.87	12 361.26

applied the min–max method to normalize all results. However, a more robust normalization method, like (Giudici & Raffinetti, 2021), would be advantageous for more consistent and reliable results. Furthermore, we suggest that statistical tests are developed and used for comparing model performance, e.g., using the tests in DeLong, DeLong, and Clarke-Pearson (1988), Giudici and Raffinetti (2024), Sun and Xu (2014). These tests will enable us to rigorously analyze and compare the differences between the results obtained from different methods.

Rigorously verifying the explanation results is still challenging, and lacking ground truth is still the biggest problem in XAI research (Saeed & Omlin, 2023). This poses a significant obstacle when validating explanation results using real-world data sets. Investigating benchmark datasets that compare XAI methods is a pivotal research problem we aim to address in the future.

In the future, we are interested in expanding its scope of explainability. Currently, the EAP method can only explain a model's prediction at a global level, which may limit its application for cases where instance-level explainability is required. In nonlinear classification cases, heterogeneous problems may occur, i.e., a single class may contain several different patterns. In this case, explanations for a single instance or a small sample group may be preferred. So, exploring the possibility of the EAP method for local explanation will be an investigation in the future.

CRediT authorship contribution statement

Mengqi Zhang: Conceptualization, Methodology, Software, Data curation, Validation, Investigation, Writing – original draft. **Matthias Treder:** Conceptualization, Writing – review & editing, Supervision. **David Marshall:** Writing – review & editing, Supervision. **Yuhua Li:** Writing – review & editing, Supervision.

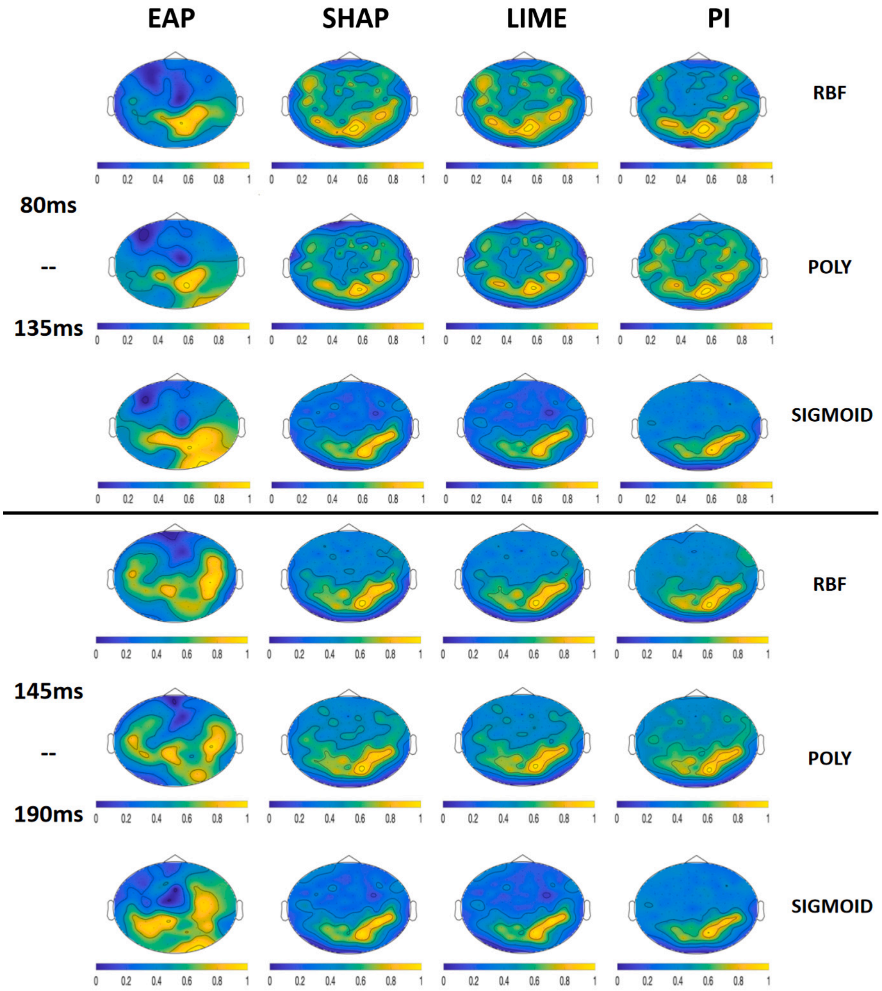

Fig. 7. Topographies of explanation results for MEG datasets. The results are averaged value of explanation results among 16 participants. For the convenience of comparison, all results are scaled from 0 to 1 before drawing topographies.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Mengqi Zhang reports financial support was provided by China Scholarship Council (CSC).

Data availability

The authors do not have permission to share data.

Funding acknowledgment

This research was supported by the China Scholarship Council (CSC) under Grant ID 202009370051.

References

Al-Saegh, A., Dawwd, S. A., & Abdul-Jabbar, J. M. (2021). Deep learning for motor imagery EEG-based classification: A review. *Biomedical Signal Processing and Control*, *63*, Article 102172.

Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R., & Samek, W. (2015). On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS One*, *10*(7), 1–46.

Barakat, N., & Bradley, A. P. (2010). Rule extraction from support vector machines: A review. *Neurocomputing*, *74*(1), 178–190, Artificial Brains.

Bentin, S., Allison, T., Puce, A., Perez, E., & McCarthy, G. (1996). Electrophysiological studies of face perception in humans. *Journal of Cognitive Neuroscience*, *8*(6), 551–565.

Boutros, N., Nasrallah, H., Leighty, R., Torello, M., Tueting, P., & Olson, S. (1997). Auditory evoked potentials, clinical vs. research applications. *Psychiatry Research*, *69*(2–3), 183–195.

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32.

Bučková, B., Brunovský, M., Bareš, M., & Hlinka, J. (2020). Predicting sex from EEG: Validity and generalizability of deep-learning-based interpretable classifier. *Frontiers in Neuroscience*, *14*.

Budding, C., Eitel, F., Ritter, K., & Haufe, S. (2021). Evaluating saliency methods on artificial data with different background types. arXiv preprint arXiv:2112.04882.

DeLong, E. R., DeLong, D. M., & Clarke-Pearson, D. L. (1988). Comparing the areas under two or more correlated receiver operating characteristic curves: a nonparametric approach. *Biometrics*, 837–845.

Dima, D. C., Perry, G., Messaritaki, E., Zhang, J., & Singh, K. D. (2018). Spatiotemporal dynamics in human visual cortex rapidly encode the emotional content of faces. *Human Brain Mapping*, *39*(10), 3993–4006.

Eger, E., Jedynak, A., Iwaki, T., & Skrandies, W. (2003). Rapid extraction of emotional expression: evidence from evoked potential fields during brief presentation of face stimuli. *Neuropsychologia*, *41*(7), 808–817.

Fisher, A., Rudin, C., & Dominici, F. (2019). All models are wrong, but many are useful: learning a variable's importance by studying an entire class of prediction models simultaneously. *Journal of Machine Learning Research*, *20*(177), 1–81.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, 1189–1232.

Giudici, P., & Raffinetti, E. (2021). Shapley-Lorenz explainable artificial intelligence. *Expert Systems with Applications*, *167*, Article 114104.

Giudici, P., & Raffinetti, E. (2024). RGA: a unified measure of predictive accuracy. *Advances in Data Analysis and Classification*, 1–27.

Goodfellow, I. J., Shlens, J., & Szegedy, C. (2014). Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572.

Greenwell, B. M., Boehmke, B. C., & McCarthy, A. J. (2018). A simple and effective model-based variable importance measure. arXiv preprint arXiv:1805.04755.

Guerrero-Gómez-Olmedo, R., Salmeron, J. L., & Kuchkovsky, C. (2020). LRP-Based path relevances for global explanation of deep architectures. *Neurocomputing*, *381*, 252–260.

Hailesilassie, T. (2016). Rule extraction algorithm for deep neural networks: A review. arXiv preprint arXiv:1610.05267.

Halgren, E., Raij, T., Marinkovic, K., Jousmäki, V., & Hari, R. (2000). Cognitive response profile of the human fusiform face area as determined by MEG. *Cerebral Cortex*, *10*(1), 69–81.

Haufe, S., Meinecke, F., Görgen, K., Dähne, S., Haynes, J.-D., Blankertz, B., et al. (2014). On the interpretation of weight vectors of linear models in multivariate neuroimaging. *NeuroImage*, *87*, 96–110.

Hebart, M. N., & Baker, C. I. (2018). Deconstructing multivariate decoding for the study of brain function. *NeuroImage*, *180*, 4–18, New advances in encoding and decoding of brain signals.

Herrmann, M., Ehlis, A.-C., Ellgring, H., & Fallgatter, A. (2005). Early stages (P100) of face perception in humans as measured with event-related potentials (ERPs). *Journal of Neural Transmission*, *112*(8), 1073–1081.

Honeine, P., & Richard, C. (2011). Preimage problem in kernel-based machine learning. *IEEE Signal Processing Magazine*, *28*(2), 77–88.

Kim, B., Khanna, R., & Koyejo, O. O. (2016). Examples are not enough, learn to criticize! criticism for interpretability. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, & R. Garnett (Eds.), *Advances in neural information processing systems*: *vol. 29*, Curran Associates, Inc..

Kim, B.-H., & Ye, J. C. (2020). Understanding graph isomorphism network for rs-fMRI functional connectivity analysis. *Frontiers in Neuroscience*, *14*.

Kriegeskorte, N., & Douglas, P. K. (2019). Interpreting encoding and decoding models. *Current Opinion in Neurobiology*, *55*, 167–179, Machine Learning, Big Data, and Neuroscience.

Kriegeskorte, N., Goebel, R., & Bandettini, P. (2006). Information-based functional brain mapping. *Proceedings of the National Academy of Sciences*, *103*(10), 3863–3868.

Kropotov, J. (2016). Chapter 3.1-Sensory systems and attention modulation. In *Functional neuromarkers for psychiatry* (pp. 140–142). Boston, Massachusetts, USA: Academic Press.

Krus, D. J., & Wilkinson, S. M. (1986). Demonstration of properties of a suppressor variable. *Behavior Research Methods, Instruments, & Computers*, *18*, 21–24.

Kwok, J.-Y., & Tsang, I.-H. (2004). The pre-image problem in kernel methods. *IEEE Transactions on Neural Networks*, *15*(6), 1517–1525.

Lawhern, V. J., Solon, A. J., Waytowich, N. R., Gordon, S. M., Hung, C. P., & Lance, B. J. (2018). EEGNet: a compact convolutional neural network for EEG-based brain–computer interfaces. *Journal of Neural Engineering*, *15*(5), Article 056013.

Liu, J., Harris, A., & Kanwisher, N. (2002). Stages of processing in face perception: an MEG study. *Nature Neuroscience*, *5*(9), 910–916.

Lotte, F., Bougrain, L., Cichocki, A., Clerc, M., Congedo, M., Rakotomamonjy, A., et al. (2018). A review of classification algorithms for EEG-based brain–computer interfaces: a 10 year update. *Journal of Neural Engineering*, *15*(3), Article 031005.

Lu, S., Hämäläinen, M., Hari, R., Ilmoniemi, R., Lounasmaa, O., Sams, M., et al. (1991). Seeing faces activates three separate areas outside the occipital visual cortex in man. *Neuroscience*, *43*(2–3), 287–290.

Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in neural information processing systems*: *vol. 30*, Curran Associates, Inc..

Maurer, U., Rossion, B., & McCandliss, B. (2008). Category specificity in early perception: face and word N170 responses differ in both lateralization and habituation properties. *Frontiers in Human Neuroscience*, *2*.

Meeren, H. K., de Gelder, B., Ahlfors, S. P., Hämäläinen, M. S., & Hadjikhani, N. (2013). Different cortical dynamics in face and body perception: an MEG study. *PLoS One*, *8*(9), Article e71408.

Mika, S., Schölkopf, B., Smola, A., Müller, K.-R., Scholz, M., & Rätsch, G. (1998). Kernel PCA and de-noising in feature spaces. In M. Kearns, S. Solla, & D. Cohn (Eds.), *Advances in neural information processing systems*: *vol. 11*, MIT Press.

Minh, D., Wang, H. X., Li, Y. F., & Nguyen, T. N. (2022). Explainable artificial intelligence: a comprehensive review. *Artificial Intelligence Review*, 1–66.

Moctezuma, L. A., & Molinas, M. (2020). Classification of low-density EEG for epileptic seizures by energy and fractal features based on EMD. *Journal of Biomedical Research*, *34*(3), 180.

Negrini, M., Brkić, D., Pizzamiglio, S., Premoli, I., & Rivolta, D. (2017). Neurophysiological correlates of featural and spacing processing for face and non-face stimuli. *Frontiers in Psychology*, *8*.

Pahuja, S., Veer, K., et al. (2022). Recent approaches on classification and feature extraction of EEG signal: A review. *Robotica*, *40*(1), 77–101.

Rasmussen, P. M., Madsen, K. H., Lund, T. E., & Hansen, L. K. (2011). Visualization of nonlinear kernel models in neuroimaging by sensitivity maps. *NeuroImage*, *55*(3), 1120–1131.

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1135–1144). New York, NY, USA: Association for Computing Machinery.

Rossion, B., & Jacques, C. (2008). Does physical interstimulus variance account for early electrophysiological face sensitive responses in the human brain? Ten lessons on the N170. *NeuroImage*, *39*(4), 1959–1979.

Saeed, W., & Omlin, C. (2023). Explainable AI (XAI): A systematic meta-survey of current challenges and future opportunities. *Knowledge-Based Systems*, *263*, Article 110273.

Saeidi, M., Karwowski, W., Farahani, F. V., Fiok, K., Taiar, R., Hancock, P., et al. (2021). Neural decoding of EEG signals with machine learning: A systematic review. *Brain Sciences*, *11*(11), 1525.

Savadkoohi, M., Oladunni, T., & Thompson, L. (2020). A machine learning approach to epileptic seizure prediction using Electroencephalogram (EEG) signal. *Biocybernetics and Biomedical Engineering*, *40*(3), 1328–1341.

Shapley, L. S. (1952). *A value for N-person games*. Santa Monica, CA: RAND Corporation.

Sharma, L. D., Bohat, V. K., Habib, M., Ala'M, A.-Z., Faris, H., & Aljarah, I. (2022). Evolutionary inspired approach for mental stress detection using EEG signal. *Expert Systems with Applications*, *197*, Article 116634.

Skrandies, W. (1990). Global field power and topographic similarity. *Brain Topography*, *3*(1), 137–141.

Sturm, I., Lapuschkin, S., Samek, W., & Müller, K.-R. (2016). Interpretable deep neural networks for single-trial EEG classification. *Journal of Neuroscience Methods*, *274*, 141–145.

Sun, X., & Xu, W. (2014). Fast implementation of DeLong's algorithm for comparing the areas under correlated receiver operating characteristic curves. *IEEE Signal Processing Letters*, *21*(11), 1389–1393.

Susac, A., Ilmoniemi, R. J., Pihko, E., Nurminen, J., & Supek, S. (2009). Early dissociation of face and object processing: A magnetoencephalographic study. *Human Brain Mapping*, *30*(3), 917–927.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., et al. (2013). Intriguing properties of neural networks. arXiv.

Tadel, F., Bock, E., Niso, G., Mosher, J. C., Cousineau, M., Pantazis, D., et al. (2019). MEG/EEG group analysis with brainstorm. *Frontiers in Neuroscience*, 76.

Takeda, Y., Suzuki, K., Kawato, M., & Yamashita, O. (2019). MEG source imaging and group analysis using VBMEG. *Frontiers in Neuroscience*, *13*, 241.

Tanskanen, T., Näsänen, R., Montez, T., Päällysaho, J., & Hari, R. (2005). Face recognition and cortical responses show similar sensitivity to noise spatial frequency. *Cerebral Cortex*, *15*(5), 526–534.

Treder, M. S. (2020). MVPA-Light: A classification and regression toolbox for multi-dimensional data. *Frontiers in Neuroscience*, *14*.

Valentin, S., Harkotte, M., & Popov, T. (2020). Interpreting neural decoding models using grouped model reliance. *PLoS Computational Biology*, *16*(1), 1–17.

Van Putten, M. J., Olbrich, S., & Arns, M. (2018). Predicting sex from brain rhythms with deep learning. *Scientific Reports*, *8*(1), 1–7.

Wakeman, D. G., & Henson, R. N. (2015). A multi-subject, multi-modal human neuroimaging dataset. *Scientific Data*, *2*(1), 1–10.

Wang, Y., Huang, H., Yang, H., Xu, J., Mo, S., Lai, H., et al. (2019). Influence of EEG references on N170 component in human facial recognition. *Frontiers in Neuroscience*, *13*, 705.

Wang, F., Tian, Y.-C., Zhang, X., & Hu, F. (2022). An ensemble of Xgboost models for detecting disorders of consciousness in brain injuries through EEG connectivity. *Expert Systems with Applications*, *198*, Article 116778.

Wen, T. Y., & Aris, S. A. M. (2022). Hybrid approach of eeg stress level classification using k-means clustering and support vector machine. *IEEE Access*, *10*, 18370–18379.

Wilming, R., Budding, C., Müller, K.-R., & Haufe, S. (2022). Scrutinizing XAI using linear ground-truth data with suppressor variables. *Machine Learning*, 1–21.

Xu, Y., Liu, J., & Kanwisher, N. (2005). The M170 is selective for faces, not for expertise. *Neuropsychologia*, *43*(4), 588–597.

Zhang, Y., Wang, Y., Zhou, G., Jin, J., Wang, B., Wang, X., et al. (2018). Multi-kernel extreme learning machine for EEG classification in brain-computer interfaces. *Expert Systems with Applications*, *96*, 302–310.