



OPEN ACCESS

EDITED BY

Saadullah Farooq Abbasi,
University of Birmingham, United Kingdom

REVIEWED BY

Muhammad Farrukh Qureshi,
Namal College, Pakistan
Hafza Ayesha Siddiq,
Fudan University, China

*CORRESPONDENCE

Jie Zhou
✉ sxuj_zhou@163.com

RECEIVED 12 January 2025

ACCEPTED 06 March 2025

PUBLISHED 09 April 2025

CITATION

Jiang Z, Hu K, Qu J, Bian Z, Yu D and
Zhou J (2025) Recognition of MI-EEG signals
using extended-LSR-based inductive transfer
learning.
Front. Neuroinform. 19:1559335.
doi: 10.3389/fninf.2025.1559335

COPYRIGHT

© 2025 Jiang, Hu, Qu, Bian, Yu and Zhou.
This is an open-access article distributed
under the terms of the [Creative Commons
Attribution License \(CC BY\)](https://creativecommons.org/licenses/by/4.0/). The use,
distribution or reproduction in other forums is
permitted, provided the original author(s) and
the copyright owner(s) are credited and that
the original publication in this journal is cited,
in accordance with accepted academic
practice. No use, distribution or reproduction
is permitted which does not comply with
these terms.

Recognition of MI-EEG signals using extended-LSR-based inductive transfer learning

Zhibin Jiang^{1,2}, Keli Hu^{1,3}, Jia Qu⁴, Zekang Bian^{5,6}, Donghua Yu^{1,2}
and Jie Zhou^{1,2*}

¹Department of Computer Science and Engineering, Shaoxing University, Shaoxing, China, ²Institute of Artificial Intelligence, Shaoxing University, Shaoxing, China, ³Information Technology R&D Innovation Center of Peking University, Shaoxing, China, ⁴Department of Computer Science and Artificial Intelligence, Changzhou University, Changzhou, China, ⁵Department of AI & Computer Science, Jiangnan University, Wuxi, China, ⁶Department of Taihu Jiangsu Key Construction Lab of IoT Application Technologies, Wuxi, China

Introduction: Motor imagery electroencephalographic (MI-EEG) signal recognition is used in various brain–computer interface (BCI) systems. In most existing BCI systems, this identification relies on classification algorithms. However, generally, a large amount of subject-specific labeled training data is required to reliably calibrate the classification algorithm for each new subject. To address this challenge, an effective strategy is to integrate transfer learning into the construction of intelligent models, allowing knowledge to be transferred from the source domain to enhance the performance of models trained in the target domain. Although transfer learning has been implemented in EEG signal recognition, many existing methods are designed specifically for certain intelligent models, limiting their application and generalization.

Methods: To broaden application and generalization, an extended-LSR-based inductive transfer learning method is proposed to facilitate transfer learning across various classical intelligent models, including neural networks, Takagi-SugenoKang (TSK) fuzzy systems, and kernel methods.

Results and discussion: The proposed method not only promotes the transfer of valuable knowledge from the source domain to improve learning performance in the target domain when target domain training data are insufficient but also enhances application and generalization by incorporating multiple classic base models. The experimental results demonstrate the effectiveness of the proposed method in MI-EEG signal recognition.

KEYWORDS

motor imagery, EEG, brain–computer interface, LSR, inductive transfer learning

1 Introduction

A brain–computer interface (BCI) is a technology that establishes connections between the brain and external devices, facilitating information exchange between them (Edelman et al., 2024). BCIs collect and analyze electrical signals generated by brain activity, transforming these signals into instructions that can be used to control external devices such as computers, prosthetics, and wheelchairs. As such, BCIs can assist, enhance, and repair human sensory and motor functions, improving human–computer interaction capabilities. BCIs do not rely on the peripheral nervous system or muscles, providing a new method for people who have lost their mobility due to illness or disability to communicate with the external environment and operate devices. BCIs not only open new possibilities for people with disabilities but also

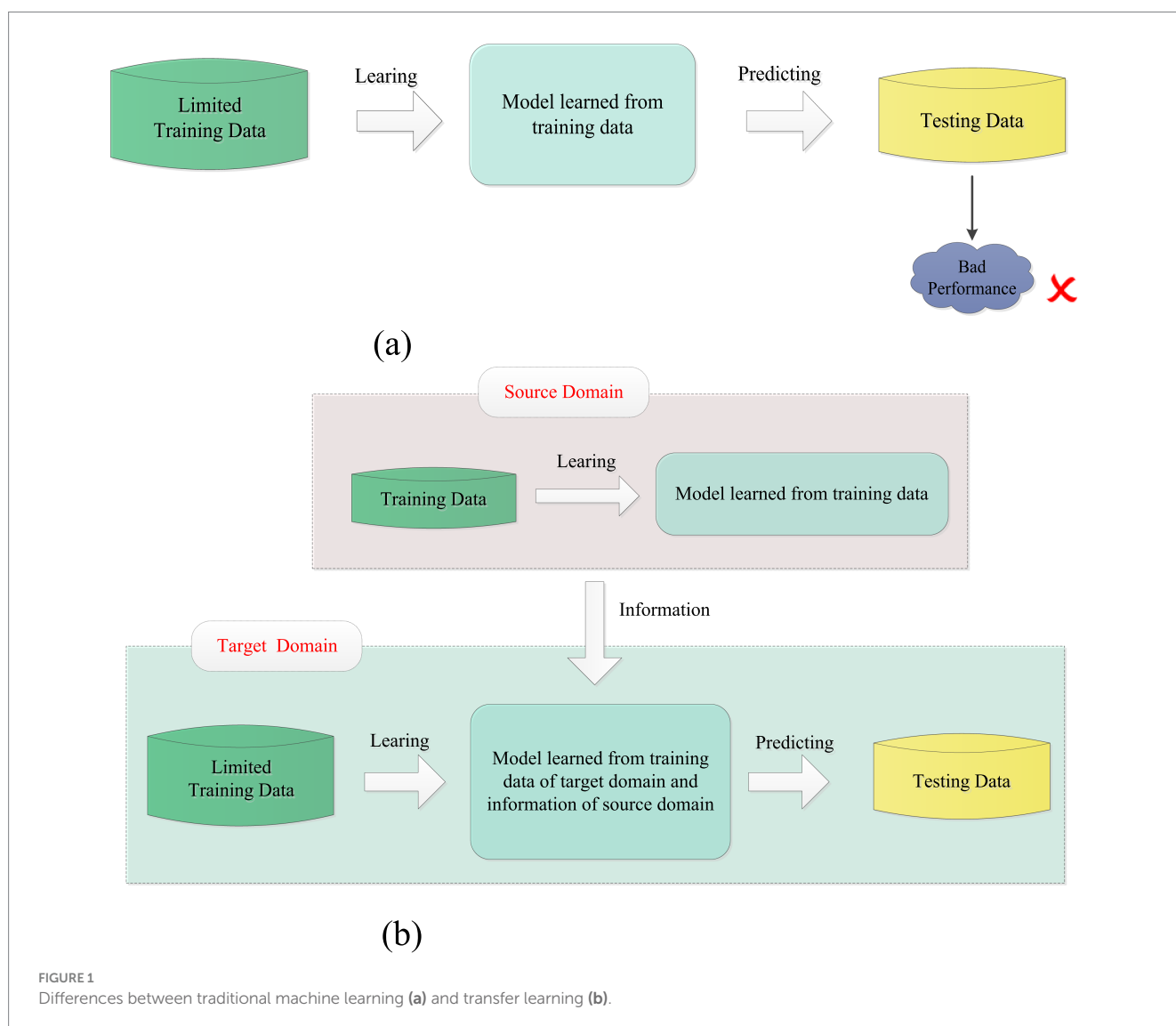
advance our understanding of the brain, ushering in a new era of human–computer interaction.

1.1 Motivation

Motor imagery electroencephalographic (MI-EEG) (Mohammadi et al., 2022) signal recognition is an important mechanism for brain-computer interfaces (BCIs). Moreover, with the advancement of machine learning, numerous classification methods based on machine learning have been proposed for MI-EEG signal recognition in the literature (Abbas et al., 2021; Ko et al., 2021; Zhang et al., 2024; Ghumman et al., 2021; Cover and Hart, 1967; Aldea et al., 2014; Kohavi, 1996; Wang and Zhang, 2016; Fisher, 1936; Li et al., 2022; Bennett and Demiriz, 1999; Fouad et al., 2020; Siddiqa et al., 2024; Siddiqa et al., 2023; Qureshi et al., 2022; Qureshi et al., 2023), including neural networks (NNs) (Abbas et al., 2021; Ko et al., 2021), fuzzy logic systems (FLSs) (Zhang et al., 2024; Ghumman et al., 2021), k-nearest neighbors (kNNs) (Cover and Hart, 1967; Aldea et al., 2014), naïve Bayes (NB) (Kohavi, 1996; Wang and Zhang, 2016), linear discriminant analysis (LDA) (Fisher, 1936; Li et al.,

2022), support vector machines (SVMs) (Bennett and Demiriz, 1999; Fouad et al., 2020), and more. Although these methods have demonstrated varying degrees of success, they typically require a large amount of subject-specific training data to adjust their parameters. However, this data acquisition process can be time-consuming and not user-friendly. When calibration data is insufficient, the classification performance of these algorithms can significantly deteriorate. As highlighted in BCI Competition III (Blankertz et al., 2006), “a challenge is that more expectations of training a model with a good classification accuracy are becoming urgent in the case that only a small number of training samples are available.” Therefore, it is essential to develop advanced machine-learning methods for MI-EEG that perform effectively with small calibration datasets.

Transfer learning is a promising method for addressing the above problem. It can be used to transfer useful information from related scenes (i.e., source domains) to the current scene (i.e., target domain), which typically has limited training data (Pan and Yang, 2010). As a result, transfer learning is particularly effective in improving classification performance during the early stages of model training when there is not enough subject-specific training data. Figure 1 shows the differences



between traditional machine learning and transfer learning. Since its introduction in 1995, transfer learning has been successfully applied in classification, clustering, and regression, with classification being the most extensively researched area. Some representative studies can be found in Zhang et al. (2022), Jiang et al. (2019), Xie et al. (2018), Pan et al. (2011), Wan et al. (2021), Li et al. (2019). Existing transfer learning methods can be categorized into three types: inductive transfer learning methods (Zhang et al., 2022; Jiang et al., 2019), which consider both supervised source and target domains; transductive transfer learning methods (Xie et al., 2018; Pan et al., 2011), which involve supervised source domains and unsupervised target domains; and unsupervised transfer learning methods, which account for both unsupervised source and target domains (Wan et al., 2021; Li et al., 2019). In MI-EEG signal recognition, when labeled MI-EEG samples in the target domain are insufficient, inductive transfer learning methods naturally become the preferred choice. Furthermore, since MI-EEG signals involve personal privacy information, inspired by Jiang et al. (2019), we investigate a knowledge-based inductive transfer learning method to ensure security without directly utilizing samples from the source domain.

Inductive transfer learning has recently attracted widespread attention and demonstrated strong performance in MI-EEG signal recognition. However, most existing inductive transfer learning methods are tailored to specific base models, rendering them inapplicable to other base models. As a result, they demonstrate poor performance in terms of application and generalization. To address this limitation, we propose an extended-LSR-based inductive transfer learning framework (ELSR-TL) that integrates neural networks, Takagi-Sugeno-Kang (TSK) fuzzy systems, and kernel methods. Figure 2 shows the framework of ELSR-TL.

1.2 Contributions

The main contributions of this study can be highlighted as follows:

- 1 ELSR-TL has an inductive transfer learning mechanism that can be used to transfer useful knowledge from the source domain to enhance learning performance in the target domain when the training data in the target domain are insufficient.
- 2 ELSR-TL enhances LSR by integrating multiple classic base models, such as neural networks, TSK fuzzy systems, and kernel methods. As such, ELSR-TL is not only suited for a specific model but also demonstrates improved applicability and generalization.
- 3 Experimental studies were conducted to validate the applicability of the proposed method for MI-EEG signal identification.

The remainder of this paper is organized as follows: Section II describes related work, including studies on existing MI-EEG feature extraction and pattern recognition methods. Section III details the proposed extended-LSR-based inductive transfer learning method. Section IV provides the experimental results and analysis. Finally, Section VI presents the conclusions drawn.

2 Backgrounds

This section states the backgrounds underlying the proposed MI-EEG recognition method. It describes the datasets used to evaluate

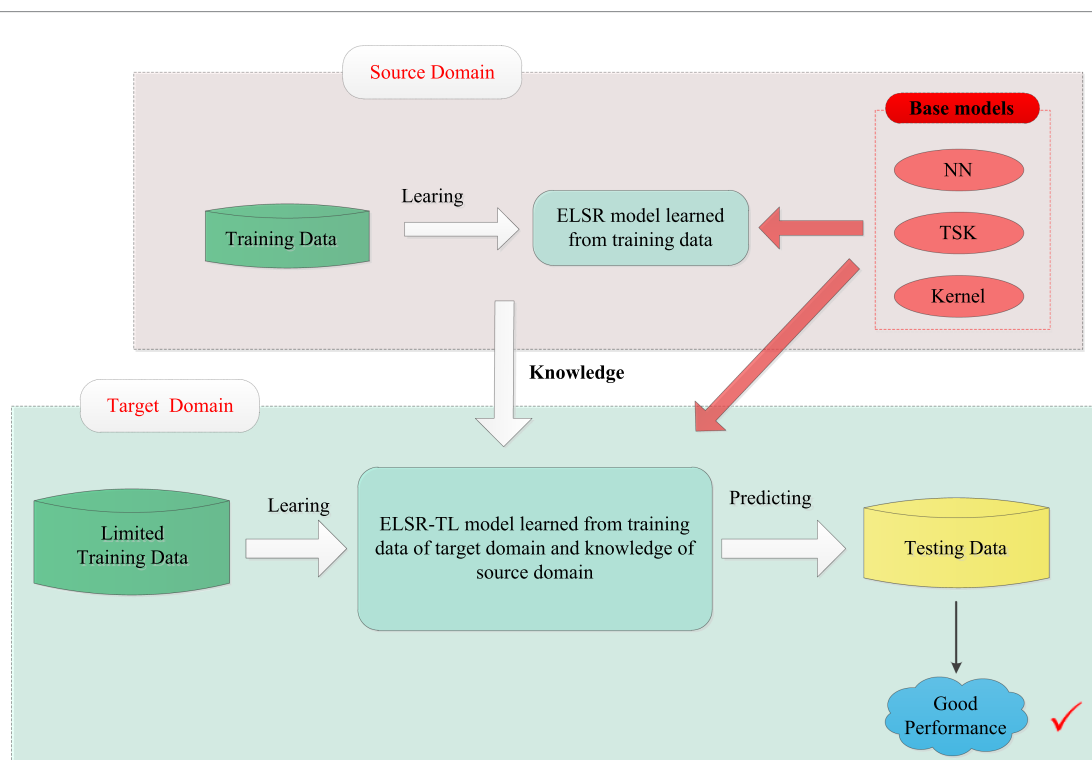


FIGURE 2
Framework of ELSR-TL.

the method and reviews several classical feature extraction and pattern recognition methods.

2.1 Datasets

We used BCI Competition Data Set IVa, provided by Fraunhofer FIRST and Charité University Medicine Berlin. A detailed description of this dataset can be found in (Blankertz et al., 2006).

This MI-EEG dataset contains five subsets corresponding to five healthy testers (aa, al, av, aw, and ay). Each subset contains 280 EEG trials, which have 128 electrodes and a trial length of 3.5 s. Each subset was partitioned into a training set and a test set, as shown in Figure 3. Figure 4 shows the representative MI-EEG signals in the five subsets.

2.2 Feature extraction methods

EEG signals are complex, nonlinear, and non-stationary. Effective feature extraction is critical to pattern recognition performance. Some of the most representative feature extraction methods have been proposed to manage raw MI-EEG signals. Typically, feature extraction methods can be classified into four main categories: time-domain analysis, frequency-domain analysis, time-frequency analysis, and space-domain analysis.

In time domain analysis, EEG signal features are analyzed in the time domain. Characteristics of the waveforms, such as mean, variance, amplitude, and kurtosis, can be used to extract features of MI-EEG signals (Greene et al., 2008).

In frequency domain analysis, the features of EEG signals are analyzed by investigating the relationship between their frequency and energy. The short-time Fourier transform (Schafer and Rabiner, 1973) is a classical power spectrum analysis method, and adaptive autoregression (Pfurtscheller et al., 1998) is an improved frequency domain analysis method.

In time-frequency analysis (Blanco et al., 1997), the features of EEG signals are extracted using the joint distribution information of

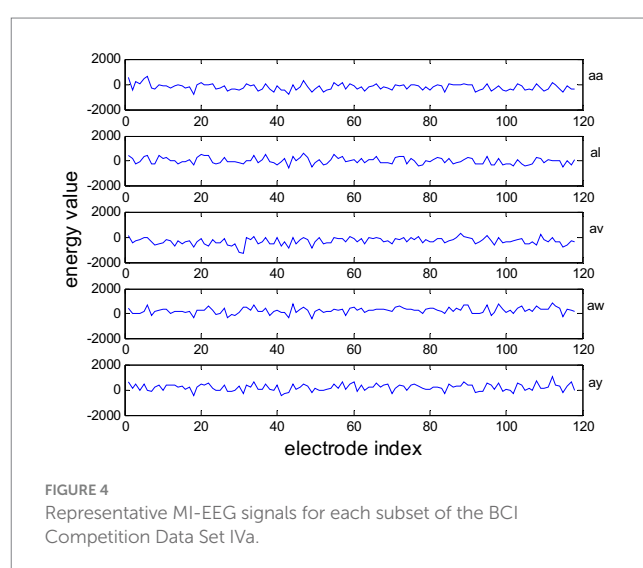
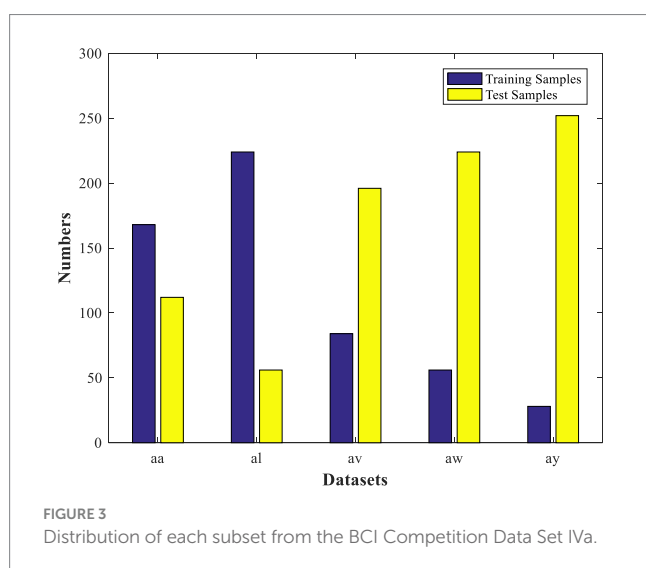
the time and frequency domains. Wavelet transform analysis (Antonini et al., 1992) is the most representative method in this category.

In space-domain analysis, the features of EEG signals are extracted by analyzing the electrical activity of neurons in different brain spaces. Common spatial pattern (CSP) (Lotte and Guan, 2011) is a commonly used method in this category. In this method, labeled trials are used to produce a transformation that maximizes the variance of one class while minimizing the variance of the other.

2.3 Pattern recognition methods

Pattern recognition utilizes the extracted EEG features for classification. Some of the most representative pattern recognition methods include the following: (1) NNs (Abbas et al., 2021; Ko et al., 2021), which simulate the mechanism of the human nervous system. Feedforward NNs are the most commonly used in EEG classification. (2) FLSs (Zhang et al., 2024; Ghuman et al., 2021), which emulate the human reasoning process and excel at managing numerical and linguistic uncertainties. (3) kNNs (Cover and Hart, 1967; Aldea et al., 2014), which determine the class of a new sample by considering its k nearest neighbors. (4) NB (Kohavi, 1996; Wang and Zhang, 2016), a simple and efficient classification algorithm based on probability. By utilizing known conditional probability and *a priori* probability, NB calculates the posterior probability of each class and assigns the test sample to the class with the highest a posteriori probability. (5) LDA (Fisher, 1936; Li et al., 2022), which applies the Fisher criterion to find the optimal projective vector that maximizes the largest scatter between classes while minimizing the scatter within each class. (6) SVMs (Bennett and Demiriz, 1999; Fouad et al., 2020), which aim to maximize the margins between different classes.

Although existing MI-EEG classification methods have demonstrated their effectiveness in various applications, they all require a substantial amount of subject-specific training data. In



practice, such training data may not be easy to obtain, and the classification accuracy of existing methods may drop significantly. To address this challenge, we use an inductive transfer learning-based MI-EEG classification method.

3 Extended-LSR-based inductive transfer learning

In this section, we provide a detailed description of the proposed extended-LSR-based inductive transfer learning (ELSR-TL) method. First, we extend LSR (Naseem et al., 2010) to its extended version, ELSR, by merging neural networks, TSK fuzzy systems, and kernel methods. Then, we develop the proposed ELSR-TL. Finally, we present the learning algorithm and theoretical analysis of ELSR-TL.

3.1 ELSR

3.1.1 Objective function of ELSR

ELSR is an extension of the basic LSR (Naseem et al., 2010). Given n d -dimensional samples $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$, where $\mathbf{x}_i \in \mathbb{R}^d$, $y_i \in \{-1; +1\}$, the objective function of ELSR can be expressed as follows:

$$\min_{\mathbf{w}} \frac{1}{2} \|\mathbf{X}_\rho \mathbf{w} - \mathbf{y}\|^2 + \frac{\lambda}{2} \|\mathbf{w}\|^2 \quad (1)$$

where the matrix $\mathbf{X}_\rho = [\rho(\mathbf{x}_1), \dots, \rho(\mathbf{x}_N)]^T \in \mathbb{R}^{N \times d_\rho}$ denotes all the given training samples, and $\rho(\mathbf{x}) \in \mathbb{R}^{d_\rho}$ is the hidden mapping function in the hidden mapping space. $\mathbf{w}$ represents the mapping matrix, λ is the given regularization parameter, and $\mathbf{y}$ is the corresponding label matrix.

The decision-making function of ELSR can be expressed as follows:

$$y = f(\mathbf{x}) = \rho(\mathbf{x})^T \mathbf{w} \quad (2)$$

Using different mapping functions $\rho(\mathbf{x})$, we can integrate multiple models, such as neural networks, TSK fuzzy systems, and kernel methods, into the proposed ELSR framework. In other words, ELSR can be developed for different base models, which improves its generalization and adaptability. We will describe its relationships with several base models next.

3.1.1.1 The relationship between ELSR and feedforward NNs

A multiple hidden layer feedforward network (MHFN) has an input layer, M hidden layers, and an output layer. The multiple hidden layers can be treated as a single complex hidden layer, allowing the overall activation function of these hidden layers to be represented by a single complex function. Therefore, an MHFN can be viewed as a generalized single hidden layer feedforward network (SHFN) with a more complex activation

function. The output of a generalized SHFN can be expressed as follows:

$$y = f(\mathbf{x}) = \sum_{i=1}^{N_M} g_i(\mathbf{x}, \theta_i)^T \mathbf{w}_i \quad (3)$$

where N_M is the number of nodes in the last hidden layer of an MHFN. As demonstrated in Huang et al. (2006), if the activation function $g_i(\mathbf{x}, \theta_i)$ is piecewise continuous, then the hidden nodes can be randomly generated independently of the training data, and the corresponding NN still maintains its universal approximation capability. Let the hidden mapping function $\rho(\mathbf{x})$ as Equation 4:

$$\rho(\mathbf{x}) = [g_1(\mathbf{x}, \theta_1), \dots, g_{N_M}(\mathbf{x}, \theta_{N_M})]^T \quad (4)$$

Then, Equation 3 can be expressed as follows:

$$y = f(\mathbf{x}) = \rho(\mathbf{x})^T \mathbf{w} \quad (5)$$

Comparing Equation 5 with Equation 2, we can see that Equation 5 is a special case of Equation 2, so Equation 1 can be used to optimize the corresponding MHFN.

3.1.1.2 The relationship between ELSR and TSK fuzzy systems

The Takagi–Sugeno–Kang fuzzy system (Gu et al., 2024; Bian et al., 2024) is the most widely used FLS due to its simplicity and flexibility. The rules in a TSK fuzzy system are typically represented as Equation 6:

TSK Fuzzy Rule R^k :

$$\begin{aligned} \text{IF } x_1 \text{ is } A_1^k \wedge x_2 \text{ is } A_2^k \wedge \dots \wedge x_d \text{ is } A_d^k \\ \text{Then } f^k(\mathbf{x}) = p_0^k + p_1^k x_1 + \dots + p_d^k x_d \quad k = 1, \dots, K \end{aligned} \quad (6)$$

Here, A_i^k is a fuzzy set for the i th input variable in the k th rule, K is the number of fuzzy rules, and $\wedge$ is a fuzzy conjunction operator. The output of the TSK fuzzy system is computed as Equation 7:

$$y = f(\mathbf{x}) = \sum_{k=1}^K \frac{\mu^k(\mathbf{x})}{\sum_{k'=1}^K \mu^{k'}(\mathbf{x})} \cdot f^k(\mathbf{x}) = \sum_{k=1}^K \mu^{-k}(\mathbf{x}) \cdot f^k(\mathbf{x}) \quad (7)$$

where $\mu^k(\mathbf{x})$ is the firing level of Rule R^k , and $\mu^{-k}(\mathbf{x})$ is the normalized $\mu^k(\mathbf{x})$, i.e., Equation 8:

$$\mu^k(\mathbf{x}) = \prod_{i=1}^d \mu_{A_i^k}(x_i) \quad (8a)$$

$$\mu^k(\mathbf{x}) = \mu^k(\mathbf{x}) / \sum_{k'=1}^K \mu^{k'}(\mathbf{x}) \quad (8b)$$

The parameters of the antecedent fuzzy sets are usually derived from clustering. The output of the TSK fuzzy system can subsequently be expressed as as Equations 9, 10:

$$y = f(\mathbf{x}) = \rho(\mathbf{x})^T \mathbf{w} \quad (9)$$

where

$$\rho(\mathbf{x}) = \left[\begin{pmatrix} -1 \\ \mathbf{x} \end{pmatrix}^T, \begin{pmatrix} -2 \\ \mathbf{x} \end{pmatrix}^T, \dots, \begin{pmatrix} -K \\ \mathbf{x} \end{pmatrix}^T \right]^T \quad (10a)$$

$$\mathbf{x}_e = \mu^k(\mathbf{x}) \mathbf{x}_e \quad (10b)$$

$$\mathbf{x}_e = (1, \mathbf{x}^T)^T \quad (10c)$$

$$\mathbf{w} = \left[(\mathbf{p}^1)^T, (\mathbf{p}^2)^T, \dots, (\mathbf{p}^K)^T \right]^T \quad (10d)$$

$$\mathbf{p}^k = (p_0^k, p_1^k, \dots, p_d^k)^T \quad (10e)$$

Equation 9 suggests that training the TSK fuzzy system can also be treated as a special case of ELSR, and thus, it can be addressed using Equation 1.

3.1.1.3 The relationship between ELSR and kernel methods

A kernel linear regression model is expressed as follows:

$$y = \rho(\mathbf{x})^T \mathbf{w} \quad (11)$$

The hidden mapping $\rho(\mathbf{x})$ can be viewed as a kernel function; thus, Equation 11 can also be solved using Equation 1. In this case, ELSR also corresponds to the classical kernel ridge regression (Saunders et al., 1998).

3.1.2 Solution of ELSR

Depending on the condition of the hidden mapping, the objective function of ELSR in Equation 1 can be efficiently solved in various ways $\rho(\mathbf{x})$. Here, we discuss the different cases as follows:

Case 1: $\rho(\mathbf{x})$ is known: In this case, we can obtain explicit values of the data $\rho(\mathbf{x})$ in the hidden mapping space.

Let $J(\mathbf{w}) = \min_{\mathbf{w}} \frac{1}{2} \|\mathbf{X}_{\rho} \mathbf{w} - \mathbf{y}\|^2 + \frac{\lambda}{2} \|\mathbf{w}\|^2$; according to the optimization theory (Qu et al., 2023a; Qu et al., 2023b), the solution for the model parameter $\mathbf{w}$ can then be obtained by taking the derivatives of Equation 1 and equating them to zero. That is,

$$\begin{aligned} \frac{\partial J(\mathbf{w})}{\partial \mathbf{w}} &= \mathbf{0} \\ \Rightarrow \mathbf{X}_{\rho}^T \mathbf{X}_{\rho} \mathbf{w} - \mathbf{X}_{\rho}^T \mathbf{y} + \lambda \mathbf{w} &= \mathbf{0} \\ \Rightarrow \mathbf{w} &= (\mathbf{X}_{\rho}^T \mathbf{X}_{\rho} + \lambda \mathbf{I}_{d_{\rho}})^{-1} \mathbf{X}_{\rho}^T \mathbf{y} \end{aligned} \quad (12)$$

The final decision function $f(\mathbf{x})$ can then be expressed as Equation 13:

$$y = f(\mathbf{x}) = \rho(\mathbf{x})^T \mathbf{w} \quad (13)$$

with $\mathbf{w}$ obtained in Equation 12.

Case 2: $\rho(\mathbf{x})$ is unknown: In this case, the explicit formulation of the data $\rho(\mathbf{x})$ in the hidden mapping space cannot be obtained, meaning that $\mathbf{w}$ cannot be specified explicitly. Therefore, the kernel trick is necessary to determine the final decision function $f(\mathbf{x})$. Although introducing the kernel trick into the solution strategy in Equation 12 is challenging, Equation 14, identity can be adopted to address this issue:

$$(\mathbf{P}^{-1} + \mathbf{Q}^T \mathbf{U}^{-1} \mathbf{Q})^{-1} \mathbf{Q}^T \mathbf{U}^{-1} = \mathbf{P} \mathbf{Q}^T (\mathbf{Q} \mathbf{P} \mathbf{Q}^T + \mathbf{U})^{-1} \quad (14)$$

In Equation 14, $\mathbf{P}$, $\mathbf{Q}$, and $\mathbf{U}$ are three matrices. Let $\mathbf{P} = \frac{1}{\lambda} \mathbf{I}_{d_{\rho}}$, $\mathbf{Q} = \mathbf{X}_{\rho}$, and $\mathbf{U} = \mathbf{I}_N$. With the identity of Equation 14, the solution in Equation 12 can then be expressed as follows:

$$\mathbf{w} = (\mathbf{X}_{\rho}^T \mathbf{X}_{\rho} + \lambda \mathbf{I}_{d_{\rho}})^{-1} \mathbf{X}_{\rho}^T \mathbf{y} = \mathbf{X}_{\rho}^T (\mathbf{X}_{\rho} \mathbf{X}_{\rho}^T + \lambda \mathbf{I}_N)^{-1} \mathbf{y} \quad (15)$$

Define a Mercer kernel matrix as Equation 16:

$$\Omega = \mathbf{X}_{\rho} \mathbf{X}_{\rho}^T \in \mathbb{R}^{N \times N}, \quad (16)$$

where $\Omega_{i,j} = \rho(\mathbf{x}_i)^T \rho(\mathbf{x}_j) = K(\mathbf{x}_i, \mathbf{x}_j)$, and $K(\cdot)$ is a kernel function.

The final decision function $f(\mathbf{x})$ can then be expressed as follows:

$$\begin{aligned} y &= f(\mathbf{x}) = \rho(\mathbf{x})^T \mathbf{w} = \rho(\mathbf{x})^T \mathbf{X}_{\rho}^T (\mathbf{X}_{\rho} \mathbf{X}_{\rho}^T + \lambda \mathbf{I}_N)^{-1} \mathbf{y} \\ \mathbf{y} &= \begin{bmatrix} K(\mathbf{x}, \mathbf{x}_1) \\ \vdots \\ K(\mathbf{x}, \mathbf{x}_N) \end{bmatrix} (\Omega + \lambda \mathbf{I}_N)^{-1} \mathbf{y} \end{aligned} \quad (17)$$

3.2 ELSR-TL

3.2.1 Objective function of ELSR-TL

ELSR-TL integrates transfer learning and ELSR. Its objective function can be expressed as follows:

$$\min_{\mathbf{w}_t} \frac{1}{2} \|\mathbf{X}_{\rho,t} \mathbf{w}_t - \mathbf{y}\|^2 + \frac{\lambda}{2} \|\mathbf{w}_t\|^2 + \frac{\beta}{2} \|\mathbf{w}_t - \mathbf{w}_s\|^2 \quad (18)$$

where $\mathbf{X}_{\rho,t} \in \mathbb{R}^{N_t \times d_\rho}$ represents N_t training samples of d_ρ

dimensions in the target domain. $\mathbf{w}_t$ and $\mathbf{w}_s$ represent the mapping matrices of the target domain and source domain, respectively. λ and β are the given regularization parameters, and $\mathbf{y}$ is the corresponding label matrix of the target domain.

In Equation 18, the first two terms are inherited directly from ELSR for learning from the target domain data, while the third term is used to leverage knowledge from the source domain. In other words, ELSR-TL generalizes ELSR from the perspective of transfer learning.

Moreover, as a regularization parameter, β can be used to adjust the role of transfer learning. When β is large, it indicates that transfer learning has a significant impact, indicating that the knowledge obtained from the source domain has a significant positive effect on the target domain. In contrast, when β is very small, it indicates that its role in learning of the target domain is relatively small. In extreme cases, when $\beta = 0$, it means that β has no effect on the learning of the target domain. In other words, we can control the effectiveness of transfer learning by making adjustments, thus effectively avoiding negative transfer.

3.2.2 Solution of ELSR-TL

ELSR-TL is solved differently in different scenarios:

Case 1: $\rho(\mathbf{x})$ is known: In this case, we can obtain explicit values of the data $\rho(\mathbf{x})$ in the hidden mapping space. The solution for the model parameter $\mathbf{w}_t$ can then be obtained in a similar form as that shown in Equation 12, that is

$$\begin{aligned} \frac{\partial J(\mathbf{w}_t)}{\partial \mathbf{w}_t} &= \mathbf{0} \\ \Rightarrow \mathbf{X}_{\rho,t}^T \mathbf{X}_{\rho,t} \mathbf{w}_t - \mathbf{X}_{\rho,t}^T \mathbf{y} + \lambda \mathbf{w}_t + \beta (\mathbf{w}_t - \mathbf{w}_s) &= \mathbf{0} \\ \Rightarrow \mathbf{w}_t &= \left(\mathbf{X}_{\rho,t}^T \mathbf{X}_{\rho,t} + (\lambda + \beta) \mathbf{I}_{d_\rho} \right)^{-1} \left(\mathbf{X}_{\rho,t}^T \mathbf{y} + \beta \mathbf{w}_s \right) \end{aligned} \quad (19)$$

The final output of the proposed ELSR-TL is expressed as follows:

$$y = f(\mathbf{x}) = \rho(\mathbf{x})^T \mathbf{w}_t \quad (20)$$

with $\mathbf{w}_t$ obtained in Equation 19.

Case 2: $\rho(\mathbf{x})$ is unknown: In this case, the explicit formulation of the data $\rho(\mathbf{x})$ in the hidden mapping space cannot be obtained, and thus, $\mathbf{w}_t$ cannot be specified explicitly. Similar to the form shown in Equation 17, the output of the proposed ELSR-TL can be calculated

using the kernel trick. From Equation 15, we know that $\mathbf{w}_s$ can be expressed as Equation 21:

$$\mathbf{w}_s = \mathbf{X}_{\rho,s}^T \left(\mathbf{X}_{\rho,s} \mathbf{X}_{\rho,s}^T + \lambda_s \mathbf{I}_{N_s} \right)^{-1} \mathbf{y}_s \quad (21)$$

Here, $\mathbf{w}_s$ is the parameter of ELSR in the source domain. For a similar scenario, let

$$\alpha_s = \left(\mathbf{X}_{\rho,s} \mathbf{X}_{\rho,s}^T + \lambda_s \mathbf{I}_{N_s} \right)^{-1} \mathbf{y}_s \quad (22)$$

A Mercer kernel matrix is defined, and Equation 22 can then be re-expressed as Equation 23:

$$\alpha_s = \left(\Omega_s + \lambda_s \mathbf{I}_{N_s} \right)^{-1} \mathbf{y}_s \quad (23)$$

where $\Omega_s = \mathbf{X}_{\rho,s} \mathbf{X}_{\rho,s}^T = \left[K(\mathbf{x}_{i,s}, \mathbf{x}_{j,s}) \right]_{N_s \times N_s}$, in which $K(\mathbf{x}_{i,s}, \mathbf{x}_{j,s})$ is the kernel function.

$\mathbf{w}_s$ can then be written as follows:

$$\mathbf{w}_s = \mathbf{X}_{\rho,s}^T \alpha_s \quad (24)$$

From Equation 19, we obtain the following:

$$\begin{aligned} \mathbf{w}_t &= \frac{\beta}{\lambda + \beta} \mathbf{w}_s \\ &+ \frac{1}{\lambda + \beta} \mathbf{X}_{\rho,t}^T \left(\mathbf{X}_{\rho,t} \mathbf{X}_{\rho,t}^T + (\lambda + \beta) \mathbf{I}_{N_t} \right)^{-1} \left(\mathbf{y} - \frac{\beta}{\lambda + \beta} \mathbf{X}_{\rho,t} \mathbf{w}_s \right) \end{aligned} \quad (25)$$

Substituting Equation 24 into Equation 25 and defining a Mercer kernel matrix, the equation above can then be rewritten as follows:

$$\begin{aligned} \mathbf{w}_t &= \frac{\beta}{\lambda + \beta} \mathbf{w}_s + \frac{1}{\lambda + \beta} \mathbf{X}_{\rho,t}^T \\ &\left(\mathbf{X}_{\rho,t} \mathbf{X}_{\rho,t}^T + (\lambda + \beta) \mathbf{I}_{N_t} \right)^{-1} \left(\mathbf{y} - \frac{\beta}{\lambda + \beta} \mathbf{X}_{\rho,t} \mathbf{w}_s \right) \\ &= \frac{\beta}{\lambda + \beta} \mathbf{X}_{\rho,s}^T \alpha_s + \frac{1}{\lambda + \beta} \mathbf{X}_{\rho,t}^T \\ &\left(\Omega_t + (\lambda + \beta) \mathbf{I}_{N_t} \right)^{-1} \left(\mathbf{y} - \frac{\beta}{\lambda + \beta} \Omega_{t,s} \alpha_s \right) \end{aligned} \quad (26)$$

where $\Omega_t = \mathbf{X}_{\rho,t} \mathbf{X}_{\rho,t}^T = \left[K(\mathbf{x}_{i,t}, \mathbf{x}_{j,t}) \right]_{N_t \times N_t}$,

$\Omega_{t,s} = \mathbf{X}_{\rho,t} \mathbf{X}_{\rho,s}^T = \left[K(\mathbf{x}_{i,t}, \mathbf{x}_{j,s}) \right]_{N_t \times N_s}$

Finally, by using $\mathbf{w}_t$ obtained in Equation 26, the decision function of the proposed ELSR-TL can be expressed as follows:

$$\begin{aligned}
f(\mathbf{x}) &= \rho(\mathbf{x})^T \mathbf{w}_t \\
&= \rho(\mathbf{x})^T \left[\frac{\beta}{\lambda + \beta} \mathbf{X}_{\rho,s}^T \alpha_s + \frac{1}{\lambda + \beta} \mathbf{X}_{\rho,t}^T \left(\Omega_t + (\lambda + \beta) \mathbf{I}_{N_t} \right)^{-1} \left(\mathbf{y} - \frac{\beta}{\lambda + \beta} \Omega_{t,s} \alpha_s \right) \right] \\
&= \frac{\beta}{\lambda + \beta} \begin{bmatrix} K(\mathbf{x}_{1,s}, \mathbf{x}) \\ \vdots \\ K(\mathbf{x}_{N_s,s}, \mathbf{x}) \end{bmatrix}^T \alpha_s + \frac{1}{\lambda + \beta} \begin{bmatrix} K(\mathbf{x}_{1,t}, \mathbf{x}) \\ \vdots \\ K(\mathbf{x}_{N_t,t}, \mathbf{x}) \end{bmatrix}^T \left(\Omega_t + (\lambda + \beta) \mathbf{I}_{N_t} \right)^{-1} \left(\mathbf{y} - \frac{\beta}{\lambda + \beta} \Omega_{t,s} \alpha_s \right)
\end{aligned} \quad (27)$$

3.3 Learning algorithm of ELSR-TL

Considering the above discussion, we summarize the learning algorithm of ELSR-TL in Algorithm 1. Below, we provide some remarks on ELSR-TL.

3.3.1 Remark 1

For the proposed ELSR-TL, if the hidden mapping is known and the amount of training data exceeds the dimensionality of the hidden mapping features (i.e., $(N_t \gg d_{\rho,t})$), obtaining the solution using Equation 25 is more efficient than that using Equation 19, due to the computational complexity of matrix; otherwise, Equation 19 is more efficient.

3.3.2 Remark 2

When the hidden mapping is known, only the knowledge $\mathbf{w}_s$ is used for transfer learning, and the data in the source domain are not required. This means that the proposed method provides good privacy protection. However, if the hidden feature mapping is unknown, the data in the source are also required, as shown in Equations 26, 27, to effectively implement transfer learning. In this case, the proposed method can no longer protect the privacy of the data in the source domain.

3.3.3 Computational Complexity

In this section, we discussed the computational complexity of Algorithm 1 as follows:

When the hidden mapping is known, the complexity of computing step 1 is about $O(d_{\rho,s}^3 + d_{\rho,s}^2 N_s)$, where $d_{\rho,s}$ is the dimension of samples and N_s is the number of samples in the source domain. The complexity of computing the target domain model parameters $\mathbf{w}_t$ in step 2 is about $O(d_{\rho,t}^3 + d_{\rho,t}^2 N_t)$, where $d_{\rho,t}$ is the dimension of samples and N_t is the number of samples in the target domain. In this case, the computational complexity of Algorithm 1 is about $O(d_{\rho,s}^3 + d_{\rho,s}^2 N_s + d_{\rho,t}^3 + d_{\rho,t}^2 N_t)$. When the hidden mapping is unknown, the computational complexity of Algorithm 1 is about $O(N_s^3 + N_t^3)$.

4 Experiments

In this section, we adopted a real MI-EEG dataset to evaluate the performance of the proposed ELSR-TL method. Moreover, we compared it with seven non-transfer learning methods—LSR (Naseem et al., 2010), KNN (Cover and Hart, 1967), SVM (Bennett and Demiriz, 1999), NB (Kohavi, 1996), CNN (Zhang et al., 2019), ELSR (NN), ELSR (TSK), and ELSR (Ker)—alongside two transfer learning methods—Au-SVM (Wu and Dietterich, 2004) and Tr-Adaboost (Dai et al., 2007). The comparison was conducted in terms of both average classification accuracy and standard deviation for 10 runs. The details of the experimental settings and the MI-EEG recognition results are provided as follows.

Input: The set of training samples in target domain $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{N_t}] \in \mathbb{R}^{N_t \times d}$, and their corresponding class labels

$y_i \in \{-1, +1\} (i = 1, 2, \dots, N_t)$, where $\mathbf{x}_i \in \mathbb{R}^d (i = 1, 2, \dots, N_t)$.

Output: The mapping $\mathbf{w}_t$ matrix for target domain.

Procedure:

Case 1: The hidden mapping function is known

Step 1: Obtain the source domain knowledge, i.e., the model parameters $\mathbf{w}_s$ and the related parameters of the hidden nodes of the source domain.

Step 2: Calculate the target domain model parameters $\mathbf{w}_t$ by using (19).

Step 3: Use (20) to calculate the output of the test set.

Case 2: The hidden mapping function is unknown

Step 1: Obtain the knowledge and data from the source domain, i.e., α_s , and $D_s = \{\mathbf{x}_{i,s}\}$.

Step 2: Use (27) to calculate the output of the test set.

ALGORITHM 1
The ELSR-TL.

4.1 Data preparation and feature extraction

4.1.1 Configurations of source and target domains

To match the transfer learning task, we constructed 20 different transfer learning datasets by subject-to-subject transferring. Table 1 shows the 20 different configurations of source and target domains. All source domains have the same number of training data, but the target domains do not. Please note that in our experiments, non-transfer learning methods are only used on the target domain.

4.1.2 Feature extraction

As mentioned in the background section, effective feature extraction is critical to pattern recognition performance. Based on (Lotte and Guan, 2011), we primarily used the Tikhonov regularization-based common spatial pattern (TR-CSP) (Lotte and Guan, 2011) for feature extraction. Furthermore, we conducted simple experiments using two other feature extraction methods, namely Composite CSP (C-CSP) and Filter Bank CSP (FB-CSP), to compare with TR-CSP. The three feature extraction methods are briefly introduced as follows:

- 1 TR-CSP: It introduces a quadratic regularization into the CSP objective function and replaces the feature matrix of the new data with the prior knowledge matrix. This regularization prefers filters with smaller norms, reducing the influence of noise.

- 2 C-CSP: It aims to perform subject-to-subject transfer by regularizing the covariance matrices using data from other subjects. Within the framework of this study, it relies only on the β hyperparameter and defines the generic covariance matrices according to the covariance matrices of other subjects.

FB-CSP: This is a feature extraction method used for motor imagery classification in BCI. It improves the accuracy of motion imagery classification by combining CSP and filter bank techniques, optimizing the subject-specific frequency band for CSP.

We extracted features from the time segment between 0.5 and 2.5 s after the cue instructing the subject to perform MI. Each trial is bandpass filtered in the 8–30 Hz range using a fifth-order Butterworth filter. For TR-CSP, we applied three pairs of filters, as recommended in (Lotte and Guan, 2011). Some examples of features extracted from subset aa are shown in Figure 5.

4.2 Adopted methods and parameters settings

All the adopted methods (without CNN) are listed in Table 2. Based on the guidelines in (Jiang et al., 2019; Xie et al., 2018; Zhang X. et al., 2023) and our experiments, we employ a grid search strategy to identify the appropriate parameters for all the adopted methods. Table 2 also includes a list of the grid search ranges for each parameter related to all the adopted methods.

4.3 Performance indices

The classification accuracy defined Equation 28 is used to evaluate the performances of different methods:

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{28}$$

TABLE 1 Settings of the source domain and target domain.

Source domain		Target domain		
Datasets	Size	Datasets	Size	
			Training	Test
al	280	aa	168	112
av	280	aa		
aw	280	aa		
ay	280	aa		
aa	280	al	224	56
av	280	al		
aw	280	al		
ay	280	al		
aa	280	av	86	196
al	280	av		
aw	280	av		
ay	280	av		
aa	280	aw	56	224
al	280	aw		
av	280	aw		
ay	280	aw		
aa	280	ay	28	252
al	280	ay		
av	280	ay		
aw	280	ay		

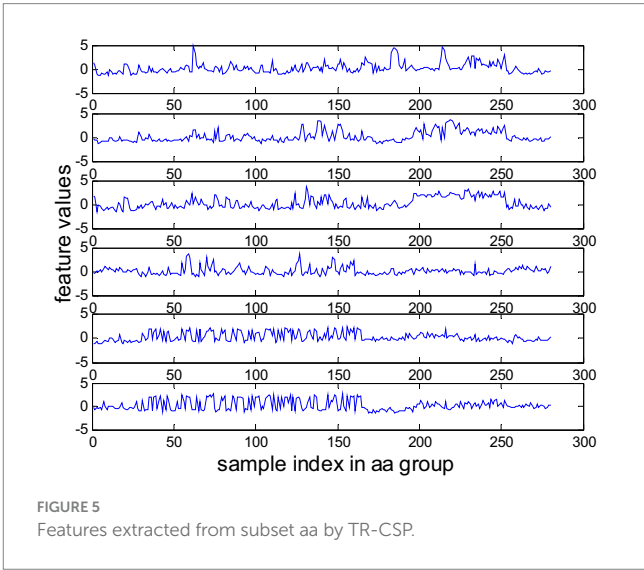


TABLE 2 The parameter setting of different methods.

Methods	Parameter settings for grid search
LSR (Naseem et al., 2010): Learns a linear regression model by using each class of training samples with the ℓ_2 -norm regularization.	The regularization parameter $\lambda \in \{10^{-6}, 10^{-5}, \dots, 10^5, 10^6\}$
kNN (Cover and Hart, 1967): A classical supervised learning model and has been widely used for classification and regression analysis.	The number of nearest points: $k \in \{1, 3, 5, 7, 9\}$.
SVM (Bennett and Demiriz, 1999): A classical classification method based on kernel trick and margin maximization.	The tradeoff parameter $C \in \{2^{-6}, 2^{-5}, \dots, 2^0, 2^1, \dots, 2^5, 2^6\}$; the width in the Gaussian kernel function $\sigma \in \{2^{-6}, 2^{-5}, \dots, 2^0, 2^1, \dots, 2^5, 2^6\}$
NB (Kohavi, 1996): A classification method based on Bayes' theorem and independent assumption of feature conditions.	The tradeoff parameter $\alpha \in \{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^0\}$
ELSR (NN): Applying the proposed ELSR for signal layer neural networks.	The number of hidden nodes $P \in \{10, 20, 30, 40, 50, 75, 100, 150, 200\}$, the parameters of the sigmoid function: $\kappa \in \{2^{-6}, 2^{-5}, \dots, 2^0, 2^1, \dots, 2^5, 2^6\}$, and the regularization parameters: $\lambda \in \{10^{-6}, 10^{-5}, \dots, 10^5, 10^6\}$
ELSR (TSK): Applying the proposed ELSR for TSK.	The number of fuzzy rules: $M \in \{5, 10, 15, 20, 25, 30, 40, 50, 80, 100\}$; the regularization parameter: $\tau \in \{10^{-6}, 10^{-5}, \dots, 10^5, 10^6\}$, the regularization parameters: $\lambda \in \{10^{-6}, 10^{-5}, \dots, 10^5, 10^6\}$.
ELSR (Ker): Applying the proposed ELSR for the kernel method.	The width in the Gaussian kernel function $\sigma \in \{2^{-6}, 2^{-5}, \dots, 2^0, 2^1, \dots, 2^5, 2^6\}$ and the regularization parameters: $\lambda \in \{10^{-6}, 10^{-5}, \dots, 10^5, 10^6\}$.
Au-SVM (Wu and Dietterich, 2004): an inductive transfer learning method based on the linear programming support vector machine with the RBF-type kernel function by using the auxiliary data.	The tradeoff parameter $C \in \{2^{-6}, 2^{-5}, \dots, 2^0, 2^1, \dots, 2^5, 2^6\}$; the width in RBF kernel function $\sigma \in \{2^{-6}, 2^{-5}, \dots, 2^0, 2^1, \dots, 2^5, 2^6\}$
Tr-Adaboost (Dai et al., 2007): an inductive transfer learning method based on the LS-SVM learner with the RBF-type kernel function for classification.	The tradeoff parameter $C \in \{2^{-6}, 2^{-5}, \dots, 2^0, 2^1, \dots, 2^5, 2^6\}$; the width in RBF kernel function $\sigma \in \{2^{-6}, 2^{-5}, \dots, 2^0, 2^1, \dots, 2^5, 2^6\}$
ELSR-TL (NN): Applying the proposed method for transfer learning of signal layer neural networks.	The number of the hidden nodes: $P \in \{10, 20, 30, 40, 50, 75, 100, 150, 200\}$, the parameters of sigmoid function: $\kappa \in \{2^{-6}, 2^{-5}, \dots, 2^0, 2^1, \dots, 2^5, 2^6\}$, the regularization parameters: $\lambda \in \{10^{-6}, 10^{-5}, \dots, 10^5, 10^6\}$, $\lambda \in \{10^{-6}, 10^{-5}, \dots, 10^5, 10^6\}$.
ELSR-TL (TSK): Applying the proposed method for transfer learning of TSK.	The number of fuzzy rules: $M \in \{5, 10, 15, 20, 25, 30, 40, 50, 80, 100\}$, the regularization parameters: $\tau \in \{10^{-6}, 10^{-5}, \dots, 10^5, 10^6\}$, $\lambda \in \{10^{-6}, 10^{-5}, \dots, 10^5, 10^6\}$, $\lambda \in \{10^{-6}, 10^{-5}, \dots, 10^5, 10^6\}$.
ELSR-TL (Ker): Applying the proposed method for transfer learning of the kernel method.	The width in the Gaussian kernel function $\sigma \in \{2^{-6}, 2^{-5}, \dots, 2^0, 2^1, \dots, 2^5, 2^6\}$, the regularization parameters: $\lambda \in \{10^{-6}, 10^{-5}, \dots, 10^5, 10^6\}$, $\lambda \in \{10^{-6}, 10^{-5}, \dots, 10^5, 10^6\}$.

4.4 Results and discussions

In all the experiments, each comparison method is implemented for 10 runs to report the average classification performance. The experimental results are shown in Tables 3, 4 and Figure 6. Please note that the feature extraction method used for these results is TR-CSP. We can make the following observations:

- i In general, the performance of the proposed ELSR-TL-based methods significantly surpasses that of the other methods used, whether they are non-transfer learning methods such as LSR, kNN, SVM, NB, CNN, and ELSR-based methods, or transfer learning methods such as Au-SVM and Tr-Adaboost. This provides experimental evidence that ELSR-TL effectively

enhances MI-EEG recognition through knowledge transfer from the source domain to the target domain.

- ii Comparing the performances of the seven non-transfer learning methods, we can see that the performance of ELSR (TSK) is the best, while the performance of NB is inferior. Moreover, each method obtains significant performance differences on different datasets. Specifically, seven non-transfer learning methods obtain the best performance on dataset *al* but poor performance on datasets *aw* and *ay*. This is because these methods require a large amount of training samples to achieve satisfactory performance, while their performance decreases when there are few training samples.
- iii Table 4 shows the performances of five transfer learning methods, showing that ELSR-TL (TSK) performs the best while Tr-Adaboost performs the worst. Furthermore, each method achieves similar

TABLE 3 Classification accuracies of the non-transfer learning method.

Datasets	LSR	kNN	SVM	NB	CNN	ELSR (NN)	ELSR (TSK)	ELSR (Ker)
aa	0.6673 (0.0133)	0.5982 (0.0148)	0.6518 (0.0071)	0.6696 (0.0101)	0.6041 (0.0136)	0.6664 (0.0106)	0.6693 (0.0110)	0.6708 (0.0107)
al	1(0)	1(0)	0.9821 (0.0031)	1(0)	0.5450 (0.0029)	1(0)	1(0)	1(0)
av	0.5416 (0.0115)	0.5663 (0.0132)	0.5561 (0.0064)	0.5510 (0.0124)	0.5459 (0.0028)	0.5612 (0.0127)	0.5658 (0.0122)	0.5508 (0.0103)
aw	0.7122 (0.0139)	0.7277 (0.0129)	0.7143 (0.0158)	0.7009 (0.0125)	0.5556 (0.0072)	0.7188 (0.0103)	0.7366 (0.0109)	0.7054 (0.0128)
ay	0.7019 (0.0142)	0.7302 (0.0172)	0.7698 (0.0145)	0.5873 (0.0129)	0.5137 (0.0107)	0.7143 (0.0120)	0.7063 (0.0117)	0.7262 (0.0114)
Avg.Acc	0.7246	0.7245	0.7348	0.7018	0.5527	0.7321	0.7356	0.7306
Avg.Std	0.0106	0.0116	0.0094	0.0096	0.0074	0.0091	0.0092	0.0090

The best results are highlighted in this table.

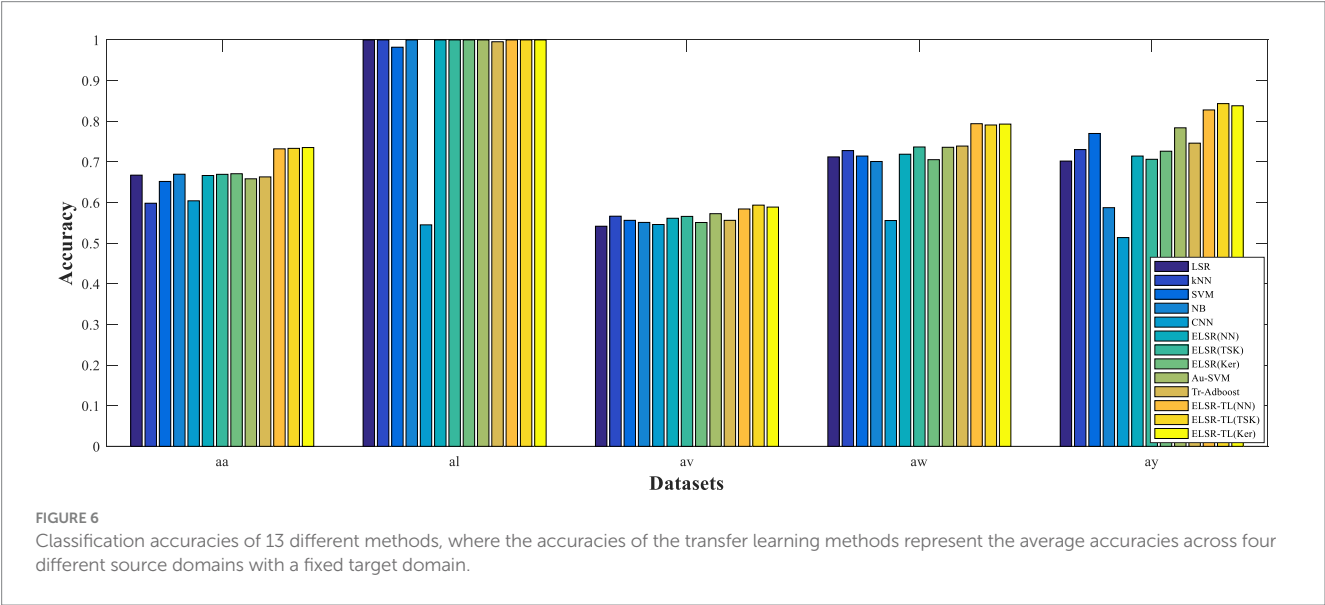


TABLE 4 Average accuracies of four different source domains for the transfer learning methods.

Target domain	Au-SVM	Tr-Adaboost	ELSR-TL (NN)	ELSR-TL (TSK)	ELSR-TL (Ker)
aa	0.6583(0.0165)	0.6630(0.0242)	0.7320(0.0116)	0.7332(0.0108)	0.7352(0.0108)
al	1(0)	0.9955(0.0002)	1(0)	1(0)	1(0)
av	0.5725(0.0180)	0.5561(0.0146)	0.5840(0.0101)	0.5935 (0.0114)	0.5887(0.0113)
aw	0.7359(0.0179)	0.7389(0.0117)	0.7939 (0.0108)	0.7907(0.0117)	0.7929(0.0100)
ay	0.7838(0.0184)	0.7460(0.0173)	0.8277(0.0111)	0.8433(0.0110)	0.8380(0.0110)
Avg.Acc	0.7501	0.7399	0.7875	0.7921	0.7910
Avg.Std	0.0142	0.0136	0.0087	0.0090	0.0086

The best results are highlighted in this table.

performances for each target subject, regardless of the auxiliary subject chosen as the source domain. Additionally, for a fixed configuration of the source and target domains, the three ELSR-TL-based methods yield similar classification accuracies.

iv Comparing the performances of transfer learning methods (i.e., Au-SVM and ELSR-TL-based methods) with their corresponding

non-transfer learning methods (i.e., SVM and ELSR-based methods), we can see that the transfer learning methods outperform the others. Therefore, transfer learning strategies are effective for MI-EEG signal recognition. Impressively, even the least effective transfer learning methods still perform better than or are comparable to the non-transfer learning methods.

- v When datasets *aa*, *al*, and *av* are used in the target domain, the performance improvements achieved by the ELSR-TL-based methods are not very significant. This is because the target domain contains sufficient data to train a good model; therefore, the knowledge from the source domain is not critical. However, when datasets *aw* and *ay* are used in the target domain, the ELSR-TL-based methods significantly outperform the other methods due to the limited training data available in the target domain.
- vi To visually compare the performances of all methods, Figure 6 illustrates the performance of each method across all datasets for visual comparison. Please note that, for transfer learning methods, we report the average performance of four different source domains with fixed target domains in Figure 6. For each dataset, the ELSR-TL-based methods achieve either the best accuracy or performance comparable to that of the other methods.

In summary, we show that ELSR-TL-based methods can outperform other methods, especially when the number of training samples in the target domain is limited.

4.5 On different feature extraction

In this section, we compare the effectiveness of three feature extraction methods. Figure 7 illustrates the classification results of

these methods when using the same classification method. Specifically, we use ELSR-TL (TSK) as the classification method.

4.6 On running time

In this section, we compared the average running times of all adopted methods (without CNN) over ten trials. Table 5 lists the average time (in seconds) for each method across all datasets. It is evident that LSR has the shortest computational time. However, among all transfer learning methods, the computational time of ELSR-TL (NN) is less than that of the other two transfer learning methods. Nevertheless, the running time of the proposed ELSR-TL is still not particularly small. Therefore, determining how to accelerate the proposed method for large-scale data remains an open problem that we should explore in the future.

4.7 Statistical analysis

A nonparametric Friedman test (Zhang Y. et al., 2023) is used to validate whether the performance differences among different algorithms are statistically significant. This test uses the rankings of different algorithms in multiple comparisons. First, we calculate the sum ranking and average ranking of the accuracy of each algorithm (without CNN), as shown in Table 6, and find the best one. We then perform *post hoc* hypothesis testing.

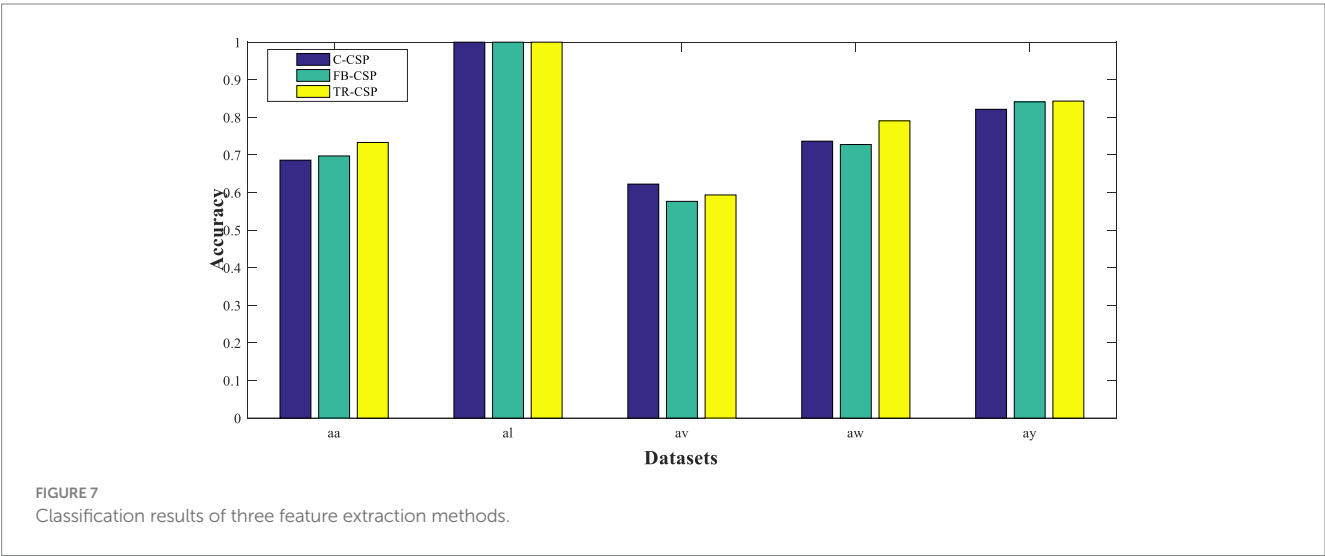


TABLE 5 Running time (Seconds) for all adopted methods on all datasets.

Datasets	LSR	kNN	SVM	NB	ELSR(NN)	ELSR(TSK)	ELSR(Ker)	Au-SVM	Tr-Adaboost	ELSR-TL (NN)	ELSR-TL (TSK)	ELSR-TL (Ker)
aa	0.2715	0.3014	83.9542	0.3058	84.5401	85.0113	86.7844	823.7881	325.6131	269.3713	274.3649	298.0787
al	0.2646	0.3001	83.0776	0.3022	84.5932	84.9688	85.0619	819.3875	325.2478	269.0109	271.3828	293.1613
av	0.2503	0.2953	82.5644	0.2759	82.9663	83.6533	83.9803	818.7702	310.2496	260.4783	270.5006	289.3647
aw	0.2382	0.2922	82.1311	0.2801	80.6762	81.4226	82.2949	803.2313	298.1549	252.3586	266.1136	279.5312
ay	0.2344	0.2887	79.6846	0.2794	78.4760	78.6811	79.1756	801.4153	296.9221	249.3428	258.9078	268.5438

TABLE 6 Rankings of the 12 algorithms (Friedman test).

Algorithm	aa	al	av	aw	ay	Sum Ranking	Average Ranking
LSR	7	5.5	12	10	11	45.5	9.1
kNN	12	5.5	5	7	7	36.5	7.3
SVM	11	12	8.5	9	5	45.5	9.1
NB	5	5.5	10	12	12	44.5	8.9
ELSR(NN)	8	5.5	7	8	9	37.5	7.5
ELSR(TSK)	6	5.5	6	5	10	32.5	6.5
ELSR(Ker)	4	5.5	11	11	8	39.5	7.9
Au-SVM	10	5.5	4	6	4	29.5	5.9
Tr-Adaboost	9	11	8.5	4	6	38.5	7.7
ELSR-TL(NN)	3	5.5	3	1	3	15.5	3.1
ELSR-TL(TSK)	2	5.5	1	3	1	12.5	2.5
ELSR-TL(Ker)	1	5.5	2	2	2	12.5	2.5

TABLE 7 The *post-hoc* comparison for $\alpha = 0.05$ (Friedman).

i	Algorithm	z	p	Holm	Hypothesis
11	LSR	2.894291	0.0038	0.004545	Reject
10	SVM	2.894291	0.0038	0.005	Reject
9	NB	2.806586	0.005007	0.005556	Reject
8	ELSR(Ker)	2.368057	0.017882	0.00625	Reject
7	Tr-Adaboost	2.280351	0.022587	0.007143	Accept
6	ELSR(NN)	2.192645	0.028333	0.008333	Accept
5	kNN	2.104939	0.035297	0.01	Accept
4	ELSR(TSK)	1.754116	0.079411	0.0125	Accept
3	Au-SVM	1.490999	0.135962	0.016667	Accept
2	ELSR-TL(NN)	0.263117	0.79246	0.025	Accept
1	ELSR-TL(Ker)	0	1	0.05	Accept

The Friedman test statistics are as Equation 29:

$$Q = \frac{12}{kn(n+1)} \sum_{i=1}^k \left(R_i - \frac{k(n+1)^2}{2} \right) = \frac{12}{kn(n+1)} \sum_{i=1}^k R_i^2 - 3k(n+1) \quad (29)$$

where R_i is the sum ranking of each algorithm, n is the number of algorithms, and k is the number of datasets.

From Table 6, we have $Q = 26.25$, and the corresponding p -value is 0.005964. This suggests that the performance differences among the 12 methods are statistically significant, with ELSR-TL (TSK) performing the best. To further evaluate the performance differences between ELSR-TL (TSK) and the other 11 methods, we also conduct post hoc multiple comparison tests:

$$z = \frac{|\bar{R}_0 - \bar{R}_i|}{SE} \quad \text{with} \quad SE = \sqrt{\frac{n^*(n+1)}{6*k}}$$

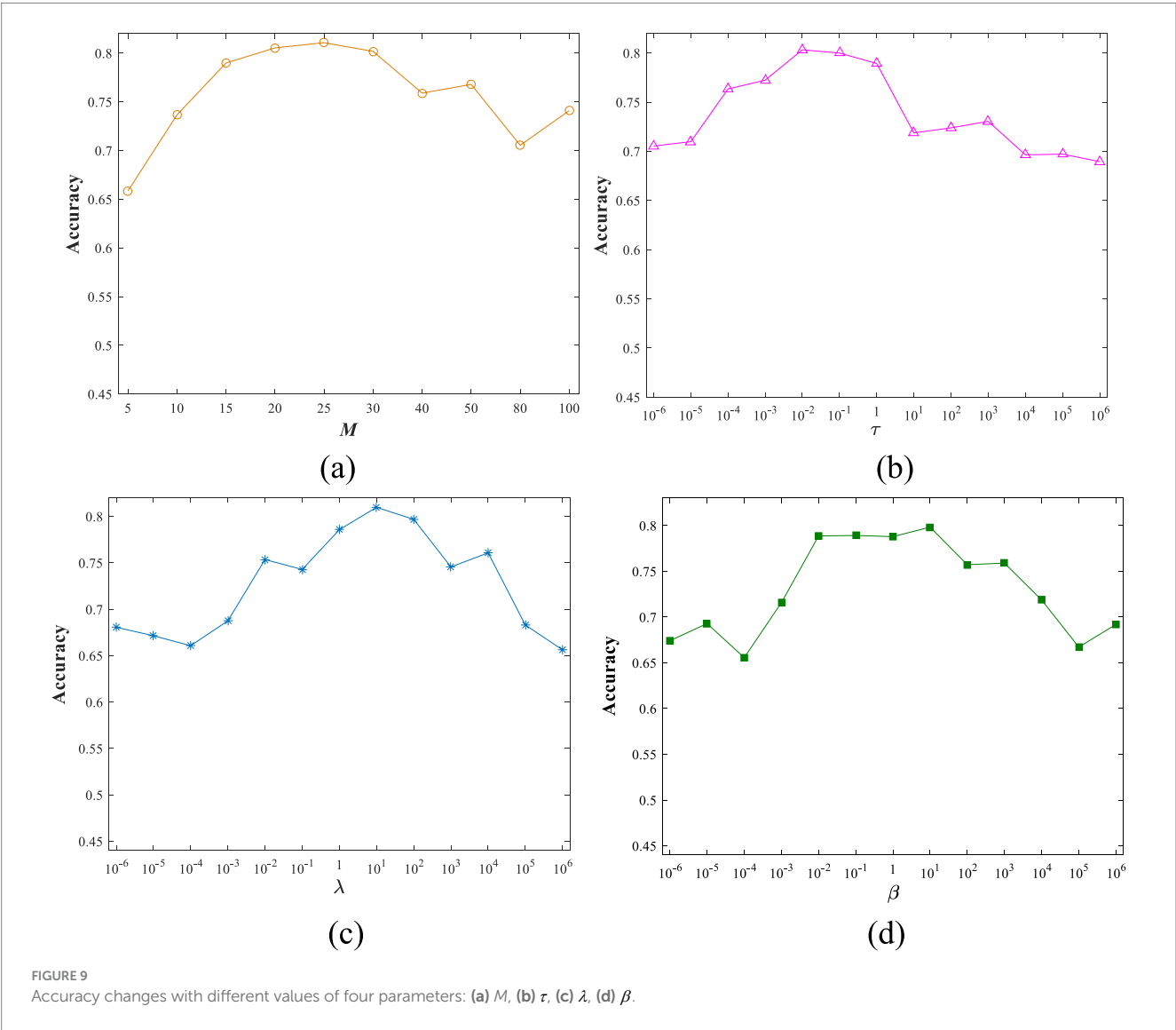
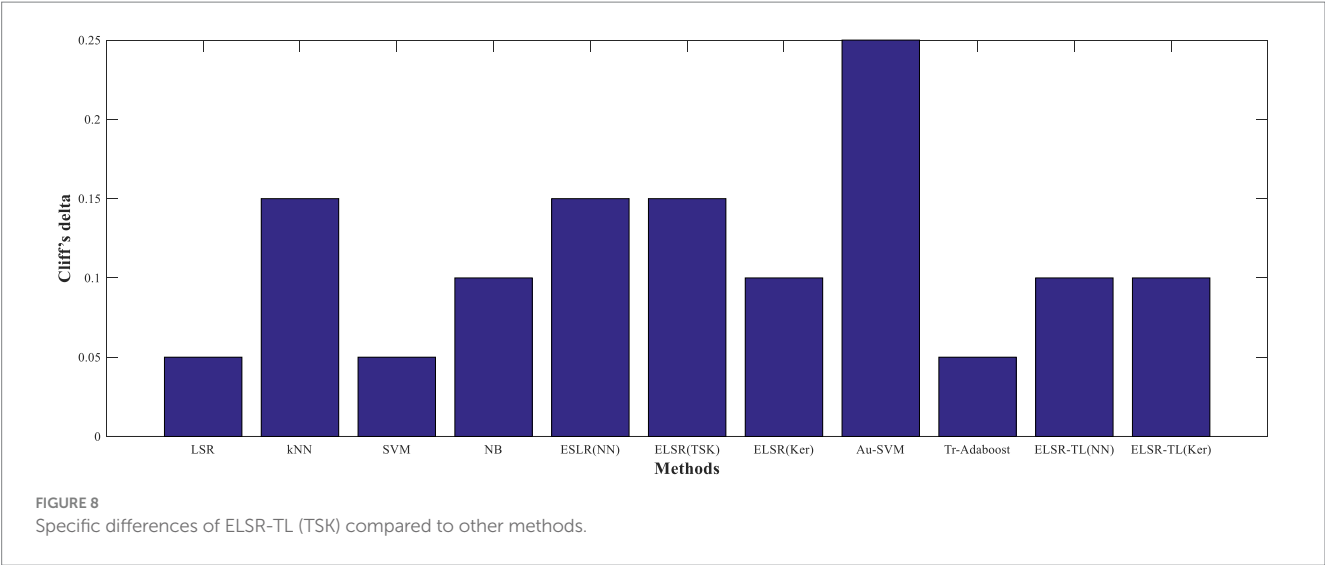
where z is subject to the standard normal distribution that will be used further to calculate the value of P .

Table 7 shows the *post hoc* comparison results for $\alpha = 0.05$ (Friedman). The null hypothesis is rejected when $p \leq 0.00625$ because $p \leq Holm$. In summary, we conclude that there are significant performance differences between ELSR-TL (TSK) and other methods, confirming that transfer learning is effective in boosting classification accuracy.

In addition, Figure 8 shows the specific differences between ELSR-TL (TSK) and other methods. Clearly, it is consistent with the above conclusion that there are significant performance differences between ELSR-TL (TSK) and the other methods.

4.8 Sensitivity analysis

We also conduct experiments to study the sensitivity of ELSR-TL to various parameters. Below, we use the AW dataset as an example of sensitivity analysis. Figure 9 illustrates how accuracy varies with different values of four parameters while the others remain fixed, based on the grid search detailed in section 4.2. Please note that, due to the limitations of this paper, we only use ELSR-TL (TSK) for the sensitivity analysis.



4.9 Limitations

Although the proposed ELSR-TL demonstrates effectiveness in these experiments, it still has some limitations. For example, there are four hyperparameters in the proposed method, and the hyperparameter optimization procedure based on grid searching and cross-validation is computationally expensive. The running time of the proposed method is significant, especially compared to traditional simple methods, making it unsuitable for real-time scenarios. The proposed ELSR-TL operates offline and cannot be used in online scenarios. The datasets used in this paper contain only five subjects on a small scale, and the effectiveness of the proposed method needs to be validated on more extensive and larger datasets in future studies. In addition, it is also worth further investigating how to provide more theoretical justifications for knowledge transfer and how to avoid negative transfer. We primarily focused on binary MI tasks in this study; therefore, exploring how to extend the proposed ELSR-TL to multi-class MI tasks, multimodal integration, and cross-dataset transfers is worth studying.

5 Conclusion

In this study, an extended LSR-based inductive transfer learning method was proposed to facilitate transfer learning for several classical intelligent models, including neural networks, TSK fuzzy systems, and kernel methods. We applied this method to MI-EEG signal recognition in BCIs. ELSR-TL provides three distinctive advantages: (1) It features an inductive transfer learning mechanism that allows for the transfer of useful knowledge from the source domain to enhance learning performance in the target domain when the training data in the target domain are insufficient. (2) It enhances application and generalization by extending LSR while integrating multiple classic base models such as neural networks, TSK fuzzy systems, and kernel methods. (3) It uses knowledge extracted from the source domain to train the classification model in the target domain, ensuring security for MI-EEG signal recognition. Experimental studies indicate the effectiveness of the proposed method in MI-EEG signal recognition. Although the proposed ELSR-TL demonstrates effectiveness in these experiments, there is still room for further research. For example, the hyperparameter optimization procedure based on grid searching and cross-validation is computationally expensive, so future research should focus on addressing this issue. The proposed ELSR-TL operates offline and cannot be applied in real-time scenarios. The datasets used in this study are relatively small in scale; thus, the effectiveness of the proposed method needs validation on more extensive datasets in future studies. Additionally, it is also worth further examining how to provide more theoretical justifications for knowledge transfer to avoid negative transfer. While this study primarily focuses on binary MI tasks, extending the proposed ELSR-TL to multi-class MI tasks, multimodal integration, and cross-dataset transfers is also worth studying.

References

- Abbas, A., Sutter, D., Zoufal, C., Lucchi, A., Figalli, A., and Woerner, S. (2021). The power of quantum neural networks. *Nat. Comput. Sci.* 1, 403–409. doi: 10.1038/s43588-021-00084-1
- Aldea, R., Fira, M., and Lazăr, A. (2014). Classifications of motor imagery tasks using k-nearest neighbors. In *12th symposium on Neural Network applications in electrical engineering (NEUREL)*, IEEE, 115–120.
- Antonini, M., Barlaud, M., Mathieu, P., and Daubechies, I. (1992). Image coding using wavelet transform. *IEEE Trans. Image Process.* 1, 205–220. doi: 10.1109/83.136597
- Bennett, K., and Demiriz, A. (1999). Semi-supervised support vector machines. *Adv. Neural Inf. Proces. Syst.* 11, 368–374.
- Bian, Z., Qu, J., Zhou, J., Jiang, Z., and Wang, S. (2024). Weighted adaptively ensemble clustering method based on fuzzy co-association matrix. *Inform. Fusion* 103:102099. doi: 10.1016/j.inffus.2023.102099
- Blanco, S., Kochen, S., Rosso, O., and Salgado, P. (1997). Applying time-frequency analysis to seizure EEG activity. *Engin. Med. Biol. Magazine IEEE* 16, 64–71. doi: 10.1109/51.566156

Data availability statement

Publicly available datasets were analyzed in this study. This data can be found at: https://www.bbc.de/competition/iii/desc_IVa.html.

Author contributions

ZJ: Writing – original draft, Writing – review & editing. KH: Formal analysis, Writing – review & editing. JQ: Methodology, Validation, Visualization, Writing – review & editing. ZB: Data curation, Methodology, Validation, Visualization, Writing – review & editing. DY: Methodology, Validation, Visualization, Writing – review & editing. JZ: Conceptualization, Methodology, Supervision, Writing – review & editing.

Funding

The author(s) declare that financial support was received for the research and/or publication of this article. This study was partially supported by the National Natural Science Foundation of China through Grants 62206177, 62106145, 62002227, and 62306126; from the Zhejiang Provincial Natural Science Foundation of China under Grants LY23F020007, LQ22F020024, LTY22F020003, and LZ24F020006; and from the Natural Science Foundation of Jiangsu Province through Grant BK20220621.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Generative AI statement

The authors declare that no Gen AI was used in the creation of this manuscript.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

- Blankertz, B., Müller, K., Krusienski, D., Schalk, G., Wolpaw, J. R., Schlögl, A., et al. (2006). The BCI competition III: validating alternative approaches to actual BCI problems. *IEEE Trans. Neural Syst. Rehabil. Eng.* 14, 153–159. doi: 10.1109/TNSRE.2006.875642
- Cover, T., and Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Trans. Inf. Theory* 13, 21–27. doi: 10.1109/TIT.1967.1053964
- Dai, W., Yang, Q., Xue, G., and Yu, Y., “Boosting for transfer learning,” in Proc. 24th Int. Conf. Mach. Learning, (2007), pp. 193–200.
- Edelman, B. J., Zhang, S., Schalk, G., Brunner, P., Muller-Putz, G., Guan, C., et al. (2024). Non-invasive brain-computer interfaces: state of the art and trends. *IEEE Rev. Biomed. Eng.* 18, 26–49. doi: 10.1109/RBME.2024.3449790
- Fisher, R. (1936). The use of multiple measurements in taxonomic problems. *Ann. Eugenics* 7, 179–188. doi: 10.1111/j.1469-1809.1936.tb02137.x
- Fouad, I. A., Labib, F. E. Z. M., Mabrouk, M. S., Sharawy, A. A., and Sayed, A. Y. (2020). Improving the performance of P300 BCI system using different methods. *Network Model. Analysis Health Inform. Bioinform.* 9, 1–13. doi: 10.1007/s13721-020-00268-1
- Ghumman, M. K., Singh, S., Singh, N., and Jindal, B. (2021). Optimization of parameters for improving the performance of P300 BCI system using different methods. *J. Reliable Intelligent Environ.* 7, 145–156. doi: 10.1007/s40860-020-00117-y
- Greene, B. R., Faul, S., Marnane, W. P., Lightbody, G., Korotchikova, I., and Boylan, G. B. (2008). A comparison of quantitative EEG features for neonatal seizure detection. *Clin. Neurophysiol.* 119, 1248–1261. doi: 10.1016/j.clinph.2008.02.001
- Gu, S., Chou, Y., Zhou, J., Jiang, Z., and Lu, M. (2024). Takagi-sugeno-Kang fuzzy clustering by direct fuzzy inference on fuzzy rules. *IEEE Transact. Emerg. Topics Comput. Intelligence* 8, 1264–1279. doi: 10.1109/TETCI.2023.3336537
- Huang, G., Chen, L., and Siew, C. (2006). Universal approximation using incremental constructive feedforward networks with random hidden nodes. *IEEE Trans. Neural Netw.* 17, 879–892. doi: 10.1109/TNN.2006.875977
- Jiang, Z., Chung, F. L., and Wang, S. (2019). Recognition of multiclass epileptic EEG signals based on knowledge and label space inductive transfer. *IEEE Trans. Neural Syst. Rehabil. Eng.* 27, 630–642. doi: 10.1109/TNSRE.2019.2904708
- Ko, W., Jeon, E., Jeong, S., and Suk, H.-I. (2021). Multi-scale neural network for EEG representation learning in BCI. *IEEE Comput. Intell. Mag.* 16, 31–45. doi: 10.1109/MCI.2021.3061875
- Kohavi, R. (1996). Scaling up the accuracy of naive-Bayes classifiers: a decision-tree hybrid. *KDD*. 96, 202–207.
- Li, S., Jin, J., Daly, I., Liu, L., and Cichocki, A. (2022). Feature selection method based on Menger curvature and LDA theory for a P300 brain-computer interface. *J. Neural Eng.* 18:066050. doi: 10.1088/1741-2552/ac42b4
- Li, J., Qiu, S., Shen, Y. Y., Liu, C. L., and He, H. (2019). Multisource transfer learning for cross-subject EEG emotion recognition. *IEEE Transact. Cybernet.* 50, 3281–3293. doi: 10.1109/TCYB.2019.2904052
- Lotte, F., and Guan, C. (2011). Regularizing common spatial patterns to improve BCI designs: unified theory and new algorithms. *IEEE Trans. Biomed. Eng.* 58, 355–362. doi: 10.1109/TBME.2010.2082539
- Mohammadi, E., Daneshmand, P. G., and Khorzoghi, S. M. S. M. (2022). Electroencephalography-based brain-computer interface motor imagery classification. *J. Med. Signals Sensors* 12, 40–47. doi: 10.4103/jmss.JMSS_74_20
- Naseem, I., Togneri, R., and Bennamoun, M. (2010). Linear regression for face recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* 32, 2106–2112. doi: 10.1109/TPAMI.2010.128
- Pan, S., Tsang, I., Kwok, J., and Yang, Q. (2011). Domain adaptation via transfer component analysis. *IEEE Trans. Neural Netw.* 22, 199–210. doi: 10.1109/TNN.2010.2091281
- Pan, S., and Yang, Q. (2010). A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.* 22, 1345–1359.
- Pfurtscheller, G., Neuper, C., Schlögl, A., and Lugger, K. (1998). Separability of EEG signals recorded during right and left motor imagery using adaptive autoregressive parameters. *IEEE Trans. Rehabil. Eng.* 6, 316–325. doi: 10.1109/86.712230
- Qu, J., Song, Z., Cheng, X., Jiang, Z., and Zhou, J. (2023a). A new integrated framework for the identification of potential virus-drug associations. *Front. Microbiol.* 14:1179414. doi: 10.3389/fmicb.2023.1179414
- Qu, J., Song, Z., Cheng, X., Jiang, Z., and Zhou, J. (2023b). Neighborhood-based inference and restricted Boltzmann machine for small molecule-miRNA associations prediction. *PeerJ* 11:e15889. doi: 10.7717/peerj.15889
- Qureshi, M. F., Mushtaq, Z., Rehman, M. Z. U., and Kamavuako, E. N. (2022). Spectral image-based multiday surface electromyography classification of hand motions using CNN for human-computer interaction. *IEEE Sensors J.* 22, 20676–20683. doi: 10.1109/JSEN.2022.3204121
- Qureshi, M. F., Mushtaq, Z., Rehman, M. Z. U., and Kamavuako, E. N. (2023). E2cnn: an efficient concatenated cnn for classification of surface emg extracted from upper limb. *IEEE Sensors J.* 23, 8989–8996. doi: 10.1109/JSEN.2023.3255408
- Saunders, C., Gammernan, A., and Vovk, V., “Ridge regression learning algorithm in dual variables,” (ICML-1998) Proceedings of the 15th International Conference on Machine Learning. Morgan Kaufmann, pp. 515–521 (1998)
- Schafer, R., and Rabiner, L. (1973). Design and simulation of a speech analysis-synthesis system based on short-time Fourier analysis. *IEEE Trans. Audio Electroacoust.* 21, 165–174. doi: 10.1109/TAU.1973.1162474
- Siddiqua, H. A., Irfan, M., Abbasi, S. F., and Chen, W. (2023). Electroencephalography (EEG) based neonatal sleep staging and detection using various classification algorithms. *Comput. Materials Continua* 77, 1759–1778. doi: 10.32604/cmc.2023.041970
- Siddiqua, H. A., Tang, Z., Xu, Y., Wang, L., Irfan, M., Abbasi, S. F., et al. (2024). Single-Channel EEG data analysis using a multi-branch CNN for neonatal sleep staging. *IEEE Access* 12, 29910–29925. doi: 10.1109/ACCESS.2024.3365570
- Wan, Z., Yang, R., Huang, M., Zeng, N., and Liu, X. (2021). A review on transfer learning in EEG signal analysis. *Neurocomputing* 421, 1–14. doi: 10.1016/j.neucom.2020.09.017
- Wang, H., and Zhang, Y. (2016). Detection of motor imagery EEG signals employing Naive Bayes based learning process. *Measurement* 86, 148–158. doi: 10.1016/j.measurement.2016.02.059
- Wu, P., and Dietterich, T. G., “Improving SVM accuracy by training on auxiliary data sources,” in Proc. 21st Int. Conf. Mach. Learning, (2004)
- Xie, L., Deng, Z., Xu, P., Choi, K. S., and Wang, S. (2018). Generalized hidden-mapping transductive transfer learning for recognition of epileptic electroencephalogram signals. *IEEE Transact. Cybernet.* 49, 2200–2214. doi: 10.1109/TCYB.2018.2821764
- Zhang, X., Dong, S., Shen, Q., Zhou, J., and Min, J. (2023). Deep extreme learning machine with knowledge augmentation for EEG seizure signal recognition. *Front. Neuroinform.* 17:1205529. doi: 10.3389/fninf.2023.1205529
- Zhang, J., Li, Y., Liu, B., Chen, H., Zhou, J., Yu, H., et al. (2024). A dynamic broad TSK fuzzy classifier based on iterative learning on progressively rebalanced data. *Inf. Sci.* 677, 120976.
- Zhang, Y., Wang, G., Huang, X., and Ding, W. (2023). TSK fuzzy system fusion at sensitivity-ensemble-level for imbalanced data classification. *Inform. Fusion* 92, 350–362. doi: 10.1016/j.inffus.2022.12.014
- Zhang, Y., Xia, K., Jiang, Y., Qian, P., Cai, W., Qiu, C., et al. (2022). Multi-modality Fusion & Inductive Knowledge Transfer Underlying non-Sparse Multi-Kernel Learning and distribution adaption. *IEEE/ACM Trans. Comput. Biol. Bioinform.* 20, 2387–2397. doi: 10.1109/TCBB.2022.3142748
- Zhang, L., Song, J., Gao, A., Chen, J., Bao, C., and Ma, K. “Be your own teacher: Improve the performance of convolutional neural networks via self distillation,” In Proc. IEEE/CVF Int. Conf. Comput. Vis., Seoul, Korea (South), (2019), pp. 3713–3722.