IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 7 September 2023, accepted 27 September 2023, date of publication 5 October 2023, date of current version 11 October 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3322375

RESEARCH ARTICLE

ELM-Based Discriminant Auto-Encoder and Multi-Kernel Fusion for Radar Specific Emitter Identification

YA SHI[1], CHENYI WANG[1], GUANGFU GE[2], FEIXIANG LIU[1], AND LELE ZHANG[1]

[1]School of Information and Control Engineering, Xi'an University of Architecture and Technology, Xi'an 710055, China
[2]Nanjing Research Institute of Electronic Engineering, Nanjing 210007, China

Corresponding author: Ya Shi (shiyaworld@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61803293 and Grant 61803294, and in part by the Special Scientific Research Plan of Education Department of Shaanxi Province under Grant 20JK0721.

ABSTRACT Radar specific emitter identification (SEI) distinguishes different radar emitters, which is the research hotspot in the fields of electronic countermeasures and intelligence reconnaissance. To enhance the identification accuracy of the real radar SEI system with limited training data, we propose the multi-kernel extreme learning machine-based discriminant auto-encoder (MK-ELM-DAE) method by combining representation learning and multi-kernel fusion in this paper. Firstly, ELM-DAE is applied to each primary feature of radar signals to extract the more discriminative low-dimensional feature representations. With the features extracted by ELM-DAE, the linear discriminant ratio-based two stage multiple kernel ELM algorithm is then employed to conduct the multi-feature fusion. The most important module of MK-ELM-DAE is ELM-DAE, which is an effective supervised dimensionality reduction method for representation learning. ELM-DAE incorporates the label information into ELM auto-encoder by introducing a supersized regularization term, so it is more suitable for classification tasks. Specifically, we use the ambiguity function (AF) to extract primary features, and subsequently design an AF-based MK-ELM-DAE method for radar SEI. Experiments show that our method has significant advantages in accuracy and testing efficiency.

INDEX TERMS Discriminant auto-encoder, extreme learning machine, multi-kernel fusion, radar specific emitter identification.

I. INTRODUCTION

Radar specific emitter identification (SEI) began in the mid-1960s [1]. SEI aims at distinguishing different radar or communication emitter individuals, which may even be identical emitters from the same production line [1], [2]. Therefore, SEI is a more challenging task in comparison to signal modulation recognition [3], waveform recognition [4], and type identification [5]. As a branch of SEI, radar SEI can uniquely identify different radar emitters, which plays a significant role in electronic countermeasures, intelligence reconnaissance, and some civilian applications.

A typical radar SEI system consists of two key modules namely feature extraction and classifier design. Intrinsically, what makes the radar SEI task feasible is the unintentional modulation on pulse (UMOP) caused by the manufacturing differences of the radar transmitters [2]. Hence, to capture the UMOP information, various UMOP features have been extracted in the time domain or transform domains of radar signals. It is generally believed that the UMOP features carry all the information about the individual differences. In the time domain, the extracted UMOP features are instantaneous frequency curve [2], [6], [7], the instantaneous phase curve [7], [8], [9], and the instantaneous amplitude curve [2], [9]. In the frequency domain, the features include the Fourier spectrum [8], the Fourier spectrum asymmetry characteristic [9], and bispectrum [10], [11], [12]. In the joint time-frequency domain, the commonly used features include the features based on variational mode decomposition [2], ambiguity function (AF) [12], [13], and other time-frequency

The associate editor coordinating the review of this manuscript and approving it for publication was Chengpeng Hao.

transforms. With these UMOP features, hundreds of classifiers [14] can be employed to accomplish the identification task.

To further improve the accuracy of radar SEI, researchers have employed various artificial intelligence (AI) techniques [15] to enhance the discriminability of the UMOP features and/or perform multi-feature fusion. As an important branch of AI, deep learning (DL) methods have demonstrated their powerful ability in extracting high-level feature representations in recent years [16], [17]. Therefore, some researchers have made efforts to learn more discriminative deep features of radar signals by putting the primary UMOP features into the DL models [9], [10], [11], [12]. In [9], a 1D convolutional neural network (1D-CNN) was designed to extract deep features from the UMOP features, such as the amplitude curve, phase curve, and spectrum asymmetry feature. In [10], the bispectrum-radon transform was followed by a hybrid deep model, which combines denoising auto-encoders (DAE) and deep belief networks (DBN). Chen et al. [11] proposed an adversarial shared-private CNN (ASP-CNN) to extract the shared features and private features from the bispectrum. In [12], the primary features are the integral bispectrum and the slice of AF (AF slice for short), which were input into the deep residual network (ResNet). Moreover, multi-feature fusion offers another effective way of improving accuracy. Liu [9] proposed a deep ensemble learning method to fuse three kinds of UMOP features. Based on multiple kernel learning (MKL) [18], [19], [20] and extreme learning machine (ELM) [21], [22], an efficient two-stage multiple kernel ELM (TSMKELM) algorithm was developed to fuse AF slices of radar signals [13]. In the framework of MKL-based multi-feature fusion, multiple kernels are first constructed upon different feature representations, and then they are linearly or nonlinearly combined by MKL.

Indeed, deep features usually can better capture individual differences. However, most DL algorithms require heavy computation and large training samples (e.g., [9], [10], [11], [12]). For a real-time radar SEI system, lightweight learning methods are preferred. Furthermore, in non-cooperative scenarios, the number of available signals is often limited. Thus, traditional non-deep feature learning methods are still worth studying. The aforementioned ELM naturally draws our attention. ELM is a fast learning mechanism for single-hidden-layer feedforward networks (SLFNs) [21]. We can further derive the nonlinear kernel ELM (KELM) via kernel trick. ELM and KELM have various applications for their good performance and simple analytical calculation way [22]. Inspired by DL, ELM has been extended to multilayer models. Kasun et al. [23] proposed an ELM auto-encoder (ELM-AE) with the output equal to the input, which uses the same solving method as ELM. By stacking ELM-AE, multilayer ELM (ML-ELM) can be easily created [23]. ML-ELM does not require fine-tuning, so it is time-efficient. Tang et al. [24] proposed a hierarchical ELM algorithm by stacking ELM sparse auto-encoder (ELM-SAE). The unsupervised ELM-AE, ELM-SAE, and their multilayer versions are not optimal for classification problems. Thus, Du et al. [25] proposed a supervised ELM-AE (SELM-AE), which not only minimizes the reconstruction error and the intraclass distance but also maximizes the interclass distance in the destination feature space. ELM-AE and its variants mentioned above can be considered as the feature extractors or the dimensionality reduction (DR) tools for representation learning. Compared to DL networks, the ELM-AE-based feature learning models have a significant advantage in terms of faster training speed [26], so they are more suitable for the radar SEI task.

Inspired by the work of [13] and [25], this paper proposes a novel radar SEI approach called multi-kernel ELM-based discriminant auto-encoder (MK-ELM-DAE), which simultaneously considers representation learning and feature fusion to improve the performance of radar SEI. In the representation learning stage, different UMOP features are input into our proposed ELM-DAE algorithm, which is an effective and efficient supervised feature extraction algorithm. In the feature fusion stage, an improved TSMKELM [13] algorithm is designed to fuse the features extracted by ELM-DAE. Therefore, the proposed MK-ELM-DAE method aims not only to enhance the discriminability of UMOP features but also to perform multi-feature fusion. The experiments on two real radar datasets demonstrate that our method outperforms the compared methods in terms of identification accuracy and test time. In particular, our method remains effective even when the number of training samples is extremely limited.

The novelty and contributions of this paper are:

1) We introduce a non-deep feature learning algorithm called ELM-DAE, which is built by adding a supervised regularizer into ELM-AE. ELM-DAE is simple but effective. It can extract more discriminative features than ELM-AE. Compared to the currently popular deep representation learning methods, ELM-DAE is much easier to solve, making it significantly more efficient. Additionally, ELM-DAE is suitable for scenarios where the number of training samples is limited.
2) To conduct multi-feature fusion, we propose a modified TSMKELM algorithm that calculates the kernel combination weights using the linear discriminant ratio (LDR) criterion instead of the kernel discriminant ratio (KDR) criterion [13]. The computation of LDR is faster than that of KDR. Moreover, the calculation of kernel combination weights and the subsequent KELM classifier are independent, leading to a fast two-stage MKL method. Therefore, both the feature learning stage and feature fusion stage of our method are highly efficient.
3) We specifically consider the AF slices as different UMOP features that need to be fused, similar to [13]. To select the most useful AF slices in the AF plane of radar signals, we initially calculate the LDR for each AF slice and subsequently discard the less valuable slices

with low LDRs. Therefore, all useful information in the AF plane is utilized to achieve improved accuracy.

The remainder of this paper is organized as follows. Section II provides a brief review of related work, including ELM, ELM-AE, and MKL. Section III describes the details of our proposed ELM-DAE and MK-ELM-DAE algorithms. In Section IV, we present the experimental setup, results, and analysis. Finally, Section V concludes the paper.

II. RELATED WORK

In this section, we will provide a brief overview of ELM, ELM-AE, and MKL as a foundation for the next section.

A. ELM AND ITS VARIANTS

Different from traditional SLFNs, the hidden-node parameters of ELM are randomly generated rather than trained [21]. The following are the details of ELM. Given the N training data $\{(\mathbf{x}_i \in \mathbb{R}^d, y_i \in \mathbb{R})\}_{i=1}^N$, where d is the data dimension and y_i is the label of $\mathbf{x}_i$. Firstly, the labels are converted to one-hot label vectors $\{\mathbf{t}_i \in \mathbb{R}^U\}_{i=1}^N$, where U is the number of classes. Assume that there are Q hidden nodes and the activation function is $\mathrm{g}(\cdot)$. Let $\{(\mathbf{a}_q \in \mathbb{R}^d, b_q \in \mathbb{R})\}_{q=1}^Q$ denote the randomly generated input weights and hidden biases. The output of the q-th hidden node is $h_q(\mathbf{x}_i) = \mathrm{g}(\mathbf{x}_i, \mathbf{a}_q, b_q)$, so the hidden layer's output is $\mathbf{h}(\mathbf{x}_i) = [h_1(\mathbf{x}_i), \ldots, h_Q(\mathbf{x}_i)] \in \mathbb{R}^{1\times Q}$. We use $\boldsymbol{\alpha}_q \in \mathbb{R}^U$ as the weight vector between the q-th hidden node and the output layer, so the output weight matrix is $\boldsymbol{\alpha} = [\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_Q]^{\mathrm{T}} \in \mathbb{R}^{Q\times U}$. When $\mathbf{x}_i$ is input into ELM, the output is $\mathbf{h}(\mathbf{x}_i)\boldsymbol{\alpha} \in \mathbb{R}^{1\times U}$. From an optimization perspective, the objective function of ELM is

$$\min_{\boldsymbol{\alpha}} \ \frac{1}{2}\|\boldsymbol{\alpha}\|_{\mathrm{F}}^2 + \frac{C}{2}\sum\nolimits_{i=1}^N \|\boldsymbol{\xi}_i\|_2^2, \tag{1}$$

where $\|\cdot\|_{\mathrm{F}}$ denotes Frobenius norm, $\|\cdot\|_2$ denotes L2 norm, $C > 0$ is the user-defined trade-off parameter, and $\boldsymbol{\xi}_i^{\mathrm{T}} = \mathbf{t}_i^{\mathrm{T}} - \mathbf{h}(\mathbf{x}_i)\boldsymbol{\alpha}$ is the training error vector for the input $\mathbf{x}_i$. Let $\mathbf{H} = [\mathbf{h}(\mathbf{x}_1)^{\mathrm{T}}, \ldots, \mathbf{h}(\mathbf{x}_N)^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{N\times Q}$ and $\mathbf{T} = [\mathbf{t}_1, \ldots, \mathbf{t}_N]^{\mathrm{T}} \in \mathbb{R}^{N\times U}$ be the hidden-layer output matrix and label matrix, respectively. Equation (1) can be rewritten as

$$\min_{\boldsymbol{\alpha}} \ \frac{1}{2}\|\boldsymbol{\alpha}\|_{\mathrm{F}}^2 + \frac{C}{2}\|\mathbf{T} - \mathbf{H}\boldsymbol{\alpha}\|_{\mathrm{F}}^2. \tag{2}$$

The matrix $\boldsymbol{\alpha}$ has a closed-form solution without iteration, which is what makes ELM much more efficient. In theory, ELM has universal approximation capability and can classify any disjoint regions [21]. Therefore, ELM has been widely adopted in various application fields due to its superior training speed, accuracy, and generalization [22]. Based on the theory of kernel machines [27], KELM can be derived by setting $\mathbf{h}(\mathbf{x}_i)$ as an implicit mapping [21]. In comparison to the well-known support vector machine (SVM) [27], both ELM and KELM algorithms can achieve comparable or even superior classification performance.

To enhance the classification accuracy, many variants of ELM have emerged. Here, we provide a brief introduction to some relevant variants. Peng et al. [28] proposed a discriminative graph regularized ELM (GELM) algorithm for face recognition. The objective function of GELM is

$$\min_{\boldsymbol{\alpha}} \ \|\mathbf{T} - \mathbf{H}\boldsymbol{\alpha}\|_{\mathrm{F}}^2 + \frac{\|\boldsymbol{\alpha}\|_{\mathrm{F}}^2}{\lambda_1} + \lambda_2 \mathrm{tr}(\boldsymbol{\alpha}^{\mathrm{T}}\mathbf{H}^{\mathrm{T}}\mathbf{L}_{\mathrm{GELM}}\mathbf{H}\boldsymbol{\alpha}), \tag{3}$$

where $\mathrm{tr}(\cdot)$ denotes the matrix trace, $\mathbf{L}_{\mathrm{GELM}}$ is the graph Laplacian matrix constructed upon the labels of training samples, λ_1 and λ_2 are trade-off hyperparameters. In [29], a discriminative manifold ELM (DMELM) was proposed for image and signal classification. The objective function of DMELM is obtained by replacing $\mathbf{L}_{\mathrm{GELM}}$ in (3) with $\mathbf{L}_{\mathrm{DMELM}} = (\mathbf{L}_b^{-1/2})^{\mathrm{T}}\mathbf{L}_w\mathbf{L}_b^{-1/2}$, where $\mathbf{L}_w$ and $\mathbf{L}_b$ represent the within-class and between-class local neighborhood graph Laplacian matrices, respectively. Iosifidis et al. [30] proposed a graph embedded ELM (GEELM) algorithm, the objective function of which is given by

$$\min_{\boldsymbol{\alpha}} \ \frac{1}{2}\|\boldsymbol{\alpha}\|_{\mathrm{F}}^2 + \frac{C}{2}\|\mathbf{T} - \mathbf{H}\boldsymbol{\alpha}\|_{\mathrm{F}}^2 + \frac{\gamma}{2}\mathrm{tr}(\boldsymbol{\alpha}^{\mathrm{T}}\mathbf{S}_p^{\dagger}\mathbf{S}_i\boldsymbol{\alpha}), \tag{4}$$

where γ is the trade-off parameter, † denotes generalized pseudoinverse, $\mathbf{S}_p$ and $\mathbf{S}_i$ refer to penalty and intrinsic scatter matrices, respectively. Under the graph embedding framework [31], different forms of $\mathbf{S}_p$ and $\mathbf{S}_i$ can be defined using the set $\{\mathbf{h}(\mathbf{x}_i)\}_{i=1}^N$. For the multi-label classification (MLC) problem, Rezaei-Ravari et al. [32] established two regularized MLC methods: one is the regularized MLC via feature manifold learning (RMLFM), and the other is the regularized MLC via dual-manifold learning (RMLDM). The objective function of RMLFM is

$$\min_{\boldsymbol{\alpha}} \ \|\mathbf{T} - \mathbf{H}\boldsymbol{\alpha}\|_{\mathrm{F}}^2 + \frac{\delta}{2}\|\boldsymbol{\alpha}\|_{21} + \frac{\theta}{2}\mathrm{tr}(\boldsymbol{\alpha}^{\mathrm{T}}\mathbf{L}_{\mathrm{LLE}}\boldsymbol{\alpha}), \tag{5}$$

where $\|\cdot\|_{21}$ is L21 norm, $\mathbf{L}_{\mathrm{LLE}} \in \mathbb{R}^{Q\times Q}$ is the graph Laplacian matrix of locally linear embedding (LLE) [31], δ and θ are hyperparameters. The objective function of RMLDM is

$$\min_{\boldsymbol{\alpha}} \ J_{\mathrm{RMLFM}} + \frac{\mu}{2}\mathrm{tr}(\boldsymbol{\alpha}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{L}'_{\mathrm{LLE}}\mathbf{X}\boldsymbol{\alpha}), \tag{6}$$

where $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]^{\mathrm{T}} \in \mathbb{R}^{N\times d}$ denotes the training data matrix, J_{RMLFM} is the three terms of (5), $\mathbf{L}'_{\mathrm{LLE}} \in \mathbb{R}^{N\times N}$ is similar to $\mathbf{L}_{\mathrm{LLE}}$, and μ is a hyperparameter. $\mathbf{L}_{\mathrm{LLE}}$ is built upon each column (feature) of $\mathbf{H}$, whereas $\mathbf{L}'_{\mathrm{LLE}}$ is built upon each row (sample) of $\mathbf{X}$. It can be seen that (6) is valid when Q is equal to d. This condition limits the performance of RMLDM to some extent. The RMLFM and RMLDM algorithms are naturally suitable for traditional (single-label) classification problems. In summary, these variants incorporate one or two regularization terms to ELM, taking into account the label and/or data structural information of the training data. Similar to ELM, all these algorithms have closed-form solutions. More variants can be found in [22].

B. ELM-AE AND ITS VARIANTS

To perform feature extraction or representation learning, ELM has been modified into ELM-AE by simply replacing the target output $\mathbf{t}_i$ with the input $\mathbf{x}_i$. Thus, ELM-AE aims to approximate the input data at the output layer [26]. In ELM-AE, the randomly generated input weights and biases are orthogonalized. We use $\boldsymbol{\beta} \in \mathbb{R}^{Q\times d}$ as the weight matrix between the hidden layer and the output layer, so the output of ELM-AE is $\mathbf{h}(\mathbf{x}_i)\boldsymbol{\beta}$, which is the reconstruction result of $\mathbf{x}_i$. The objective function of ELM-AE is

$$\min_{\boldsymbol{\beta}} \frac{1}{2}\|\boldsymbol{\beta}\|_{\mathrm{F}}^2 + \frac{C}{2}\|\mathbf{X} - \mathbf{H}\boldsymbol{\beta}\|_{\mathrm{F}}^2. \tag{7}$$

By solving (7), we have

$$\boldsymbol{\beta} = \begin{cases} \mathbf{H}^{\mathrm{T}}\left(\mathbf{I}_N/C + \mathbf{H}\mathbf{H}^{\mathrm{T}}\right)^{-1}\mathbf{X} & N < Q \\ \left(\mathbf{I}_Q/C + \mathbf{H}^{\mathrm{T}}\mathbf{H}\right)^{-1}\mathbf{H}^{\mathrm{T}}\mathbf{X} & N \geq Q \end{cases}, \tag{8}$$

where $\mathbf{I}_N$ and $\mathbf{I}_Q$ are the N-order and Q-order identity matrices respectively. With $\boldsymbol{\beta}$, the learned features are computed as $\{\boldsymbol{\beta}\mathbf{x}_i \in \mathbb{R}^Q\}_{i=1}^N$. When $Q < d$ is satisfied, $\boldsymbol{\beta}$ can be seen as a low-dimensional projection matrix. By utilizing network stacking technique initially proposed by Hinton et al. [33], ML-ELM [23] can be easily constructed. Wong et al. [34] introduced the KELM-AE algorithm, which is a kernelized version of ELM-AE. However, ELM-AE is unsupervised, so it is not optimal for classification tasks.

To make ELM-AE more adaptable to classification tasks, several supervised variants have been developed. In theory, regardless of effectiveness, the supervised versions of ELM-AE can be derived by directly adding supervised regularizers to ELM-AE. Therefore, by replacing the variables $\mathbf{T}$ and $\boldsymbol{\alpha}$ with $\mathbf{X}$ and $\boldsymbol{\beta}$, respectively, in GELM [28], DMELM [29], GEELM [30], RMLFM [32], and RMLDM [32], five new algorithms can be obtained. In fact, RMLFM and RMLDM have been extended to their supervised counterparts of ELM-AE in [32]. To facilitate the discussion in the subsequent section of this paper, we will refer to these five algorithms as GELM-AE, DMELM-AE, GEELM-AE, RMLFM-AE, and RMLDM-AE, respectively. However, the above strategy imposes regularization constraints on $\boldsymbol{\beta}$ or $\mathbf{H}\boldsymbol{\beta}$ (reconstructed samples) rather than on the desired low-dimensional samples $\{\boldsymbol{\beta}\mathbf{x}_i\}_{i=1}^N$. As a result, the discriminative ability of the learned features may be limited. In [25], Du et al. introduced the SELM-AE algorithm whose objective function is

$$\min_{\boldsymbol{\beta}} J_{\text{ELM-AE}} + \frac{\nu}{2}\mathrm{tr}(\boldsymbol{\beta}\mathbf{X}^{\mathrm{T}}\mathbf{L}_w\mathbf{X}\boldsymbol{\beta}^{\mathrm{T}}) - \frac{\tau}{2}\mathrm{tr}(\boldsymbol{\beta}\mathbf{X}^{\mathrm{T}}\mathbf{L}_b\mathbf{X}\boldsymbol{\beta}^{\mathrm{T}}), \tag{9}$$

where $J_{\text{ELM-AE}}$ refers to the objective function of ELM-AE, ν and τ are hyperparameters. SELM-AE employs the trace difference instead of the trace quotient used by DMELM. Wang et al. [35] proposed another supervised ELM-AE algorithm called within-class scatter information-based AE (WSI-AE) for one-class classification. WSI-AE minimizes the reconstruction error as well as the within-class scatter of the encoded features. The objective function of WSI-AE is

$$\min_{\boldsymbol{\beta}} \frac{C}{2}\|\mathbf{X} - \mathbf{H}\boldsymbol{\beta}\|_{\mathrm{F}}^2 + \frac{1}{2}\mathrm{tr}(\boldsymbol{\beta}\mathbf{S}_w\boldsymbol{\beta}^{\mathrm{T}}), \tag{10}$$

where $\mathbf{S}_w$ refers to the within-class scatter matrix. Yang et al. [36] proposed a graph embedding-based DR framework with ELM-AE, which reconstructs all samples according to a supervised graph matrix. As these variants are essentially built upon the ELM-AE algorithm, it is straightforward to extend them into multilayer networks using stacking techniques.

C. MKL

MKL was originally proposed to deal with the kernel learning or selection problem in kernel machines [18]. For instance, Rezaei-Ravari et al. [37] proposed the composite kernel-based KELM-AE algorithm, which linearly combines linear, polynomial, Sigmoid, and Gaussian kernels to bypass the process of kernel selection. If the base kernels of MKL are constructed upon different feature representations, MKL will be a powerful multi-feature fusion tool [13], [19], [20]. Therefore, as a complementary approach to decision fusion [38] and traditional feature fusion [38], [39], MKL offers a new paradigm for information fusion. For radar SEI, multifarious features can be extracted from radar signals. Thus, MKL has been introduced into the field of radar SEI to improve the identification accuracy. To maintain conciseness, in this paper, we refer to the concept of MKL-based multi-feature fusion as multi-kernel fusion. Suppose there are R types of features, and the corresponding base kernels are $\{K_r\}_{r=1}^R$, which can be the same or different. The Gaussian kernel is frequently used and will be employed in our experiments. The combined kernel is usually computed by

$$K_{\text{combine}} = \sum\nolimits_{r=1}^{R} \theta_r K_r; \quad s.t. \sum\nolimits_{r=1}^{R} \theta_r = 1, \tag{11}$$

where θ_r is the kernel combination weight for K_r. SVM is the most commonly used base learner (kernel machine) in MKL. Therefore, multi-kernel SVMs (MKSVMs) are widely employed. In terms of training methodology, lots of methods have been devised to learn the kernel weights and the structural parameters of base learners [19]. In general, these methods can be categorized into two groups: one-stage methods and two-stage methods. The two-stage methods are usually more efficient than the one-stage ones. Certainly, besides the linear combination illustrated in (11), one can also engage in nonlinear combinations (see [19] for details).

Due to the inherent complexity in solving SVM, the solution methods for MKSVMs are also relatively complex. By utilizing ELM as the base learner, Liu et al. [40] proposed the multi-kernel ELM (MKELM) algorithm, which is solved by an alternating optimization strategy. Thanks to the closed-form analytical solution method for ELM, each iteration of MKELM can be efficiently completed. To further accelerate MKELM, the TSMKELM algorithm was developed and successfully applied to radar SEI [13]. In TSMKELM, the KDR criterion is adopted to evaluate the importance of each feature representation, which is subsequently employed to calculate the weights for kernel combination. Once the combined kernel is obtained, KELM is directly employed for classification purposes. In [41], a combination method of canonical correlation analysis-based feature fusion and multi-kernel fusion was proposed for radar SEI. In summary, multi-kernel fusion plays a significant role in radar SEI task.

III. PROPOSED METHOD

In this section, we first present the motivation behind our method. Next, we propose ELM-DAE for supervised feature learning. Finally, we introduce the MK-ELM-DAE method.

A. MOTIVATION

For a practical radar SEI system, it demands higher requirements for both accuracy and efficiency. As stated earlier, to improve accuracy, two effective strategies are enhancing the discriminability of UMOP features and conducting multi-feature fusion. Undoubtedly, the popular DL excels at feature representation learning, enabling the extraction of high-level features [9], [10], [11], [12]. However, most DL models require a substantial amount of data and training time. Table 1 presents details of the DL models and datasets used in [9], [10], [11], and [12]. These datasets were collected in a collaborative setting, thereby ensuring a large volume of data required for training DL models. In practice, it is highly probable to encounter non-cooperative radar emitters, resulting in a scarcity of training samples. For the small sample size problem, DL models often suffer from severe overfitting and fail to perform effectively. Therefore, we continue to explore solutions within the traditional non-deep learning paradigm to overcome the challenge of limited training samples for radar SEI. Our proposed framework is illustrated in Fig. 1. Firstly, primary features are extracted from the original patterns, such as signals or images. Next, these features are enhanced using the non-deep feature learning module. Subsequently, the enhanced features are fused using the multi-kernel fusion module. Finally, the classification results are obtained. Our framework simultaneously leverages feature learning and feature fusion, aiming to achieve promising results. It should be noted that this framework is applicable to general pattern classification tasks and is not limited to radar SEI task. In addition, we have also noticed that there have been studies on DL models for handling data scarcity [42], [43], [44]. For instance, data augmentation [43], [44] is a commonly used and effective technique. We will explore in this direction in our future work.

Looking back to Fig. 1, the most important components of our framework are the non-deep feature learning module and multi-kernel fusion module. With a focus on training speed, we have designed specific algorithm for each module based on ELM. In the feature learning module, a supervised ELM-AE algorithm called ELM-DAE is proposed. To reduce the number of user-defined trade-off hyperparameters, ELM-DAE adds only one discriminant regularization term to ELM-AE. In the feature fusion module, an improved TSMKELM algorithm is developed. The efficiency of two-stage MKL methods is relatively high because the determination of kernel combination weights and structural parameters of base learners are not entangled with each other. The resultant method is named MK-ELM-DAE.

TABLE 1. DL models and the datasets used in four related methods.

Methods	[9]	[10]	[11]	[12]
DL Model	1D-CNN	DAE & DBN	ASP-CNN	ResNet
# Classes	20	6	8	4
# Samples	5829	≈17000	480000	84000
Cooperative?[1]	Yes	Yes	Yes	Yes
Real?[2]	Yes	Yes	Yes	No

[1] Are the radar emitters cooperative (Yes) or non-cooperative (No)?
[2] Are the datasets real (Yes) or simulated by computer (No)?

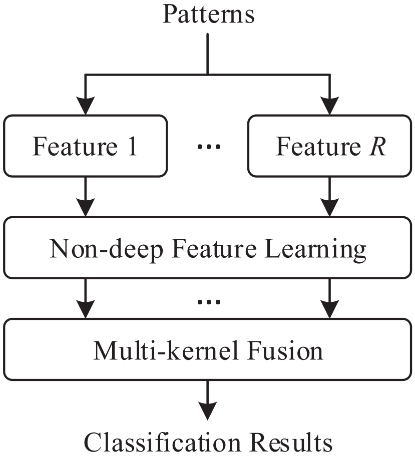


FIGURE 1. The proposed framework for general pattern classification tasks (including radar SEI task). *R* refers to the number of feature sets.

B. ELM-DAE

Similar to the aforementioned supervised variants of ELM-AE, in order to overcome the shortcoming of ELM-AE, we also incorporate the label information into the proposed ELM-DAE algorithm. The network structure of ELM-DAE is the same as that of ELM-AE. The objective function of ELM-DAE is composed of three terms: the norm constraint on the output weight matrix (i.e., the projection matrix), the reconstruction error, and the neighborhood structure of the data. The difference between ELM-DAE and ELM-AE lies in the last term, which is constructed according to the label information. We employ the idea of supervised locality preserving projection (SLPP) [45], [46] to realize the last term. Denote by $\mathbf{z}_i = \boldsymbol{\beta}\mathbf{x}_i$ the projected low-dimensional representation of $\mathbf{x}_i$. If $\mathbf{x}_i$ and $\mathbf{x}_j$ are close then $\mathbf{z}_i$ and $\mathbf{z}_j$ are close as well [45]. Consequently, ELM-DAE minimizes the following objective function:

$$J(\boldsymbol{\beta}) = \frac{1}{2}\|\boldsymbol{\beta}\|_F^2 + \frac{C}{2}\|\mathbf{X} - \mathbf{H}\boldsymbol{\beta}\|_F^2 + \frac{\lambda}{4}\sum_{i,j=1}^{N}\|\mathbf{z}_i - \mathbf{z}_j\|_2^2 W_{ij}, \tag{12}$$

where $\lambda > 0$ is the trade-off parameter and W_{ij} represents the adjacent relationship between $\mathbf{x}_i$ and $\mathbf{x}_j$. Note that the coefficients $1/2$ and $1/4$ in (12) are included for the purpose of subsequent simplification. There are several different manners of defining W_{ij}, and we will discuss them later. By substituting $\mathbf{z}_i = \boldsymbol{\beta}\mathbf{x}_i$ to the last term of (12), we get

$$\begin{aligned}
&\frac{\lambda}{4}\sum_{i,j=1}^{N}\|\mathbf{z}_i - \mathbf{z}_j\|_2^2 W_{ij} \\
&\quad= \frac{\lambda}{4}\sum_{i,j=1}^{N}\|\boldsymbol{\beta}\mathbf{x}_i - \boldsymbol{\beta}\mathbf{x}_j\|_2^2 W_{ij} \\
&\quad= \frac{\lambda}{4}\mathrm{tr}\left\{\boldsymbol{\beta}\left[\sum_{i,j=1}^{N}\left(\mathbf{x}_i - \mathbf{x}_j\right)\left(\mathbf{x}_i - \mathbf{x}_j\right)^{\mathrm{T}} W_{ij}\right]\boldsymbol{\beta}^{\mathrm{T}}\right\} \\
&\quad= \frac{\lambda}{2}\mathrm{tr}\left[\boldsymbol{\beta}\mathbf{X}^{\mathrm{T}}\left(\mathbf{D} - \mathbf{W}\right)\mathbf{X}\boldsymbol{\beta}^{\mathrm{T}}\right],
\end{aligned} \tag{13}$$

where $\mathbf{W} = \left[W_{ij}\right]_{i,j=1}^{N}$ is the adjacency matrix, and $\mathbf{D}$ is the diagonal matrix whose entries are column or row sum of $\mathbf{W}$ (for the symmetry of $\mathbf{W}$). Let $\mathbf{L} = \mathbf{D} - \mathbf{W}$, (12) can be expressed as follows:

$$J\left(\boldsymbol{\beta}\right) = \frac{1}{2}|\boldsymbol{\beta}|_{\mathrm{F}}^2 + \frac{C}{2}|\mathbf{X} - \mathbf{H}\boldsymbol{\beta}|_{\mathrm{F}}^2 + \frac{\lambda}{2}\mathrm{tr}\left(\boldsymbol{\beta}\mathbf{X}^{\mathrm{T}}\mathbf{L}\mathbf{X}\boldsymbol{\beta}^{\mathrm{T}}\right). \tag{14}$$

To find the optimal solution $\boldsymbol{\beta}$, we take the derivative of $J\left(\boldsymbol{\beta}\right)$ with respect to $\boldsymbol{\beta}$ as follows:

$$J'(\boldsymbol{\beta}) = \boldsymbol{\beta} - C\mathbf{H}^{\mathrm{T}}\left(\mathbf{X} - \mathbf{H}\boldsymbol{\beta}\right) + \lambda\boldsymbol{\beta}\mathbf{X}^{\mathrm{T}}\mathbf{L}\mathbf{X}. \tag{15}$$

Let the above derivative $J'(\boldsymbol{\beta})$ be equal to zero, we can derive the following matrix equation

$$\left(\mathbf{I}_Q + C\mathbf{H}^{\mathrm{T}}\mathbf{H}\right)\boldsymbol{\beta} + \boldsymbol{\beta}\left(\lambda\mathbf{X}^{\mathrm{T}}\mathbf{L}\mathbf{X}\right) = C\mathbf{H}^{\mathrm{T}}\mathbf{X}. \tag{16}$$

Equation (16) is the Sylvester equation (SE) [25], [47]. Since the matrices $\mathbf{I}_Q + C\mathbf{H}^{\mathrm{T}}\mathbf{H} \triangleq \mathbf{A}$ and $\lambda\mathbf{X}^{\mathrm{T}}\mathbf{L}\mathbf{X} \triangleq \mathbf{B}$ are positive semi-definite, (16) has a unique solution. The SE can be solved by using the classical Bartels-Stewart algorithm [47]. This algorithm utilizes a QR algorithm to decompose matrices $\mathbf{A}$ and $\mathbf{B}$ into Schur forms, and then solves the resultant triangular system through back-substitution. Since $\mathbf{A} \in \mathbb{R}^{Q\times Q}$ and $\mathbf{B} \in \mathbb{R}^{d\times d}$ are not very large, the Bartels-Stewart algorithm is efficient. Similar to [25], we use the MATLAB built-in function sylvester($\cdot$) to solve (16). However, training DL models is often time-consuming. It depends on various factors such as network structure, learning rate, batch size, and hardware resources (e.g., CPU or GPU).

Now we introduce the methods of constructing $\mathbf{W}$. In [31], various definitions of $\mathbf{W}$ have been summarized. Firstly, we need to find the neighbors of each training samples from the same class or different classes. Using graph theory, we put edges between neighboring samples. Next, weights should be assigned to these edges. The commonly used methods include simple-minded 0-1 weighting and heat kernel weighting [45]. In our algorithm, $\mathbf{W}$ is constructed in a simplest manner without any parameter. In detail, if $y_i = y_j = u$ (i.e., x_i and x_j belong to the u-th class) then $W_{ij} = 1/n_u$ (n_u is the number of training samples in the u-th class), otherwise we have $W_{ij} = 0$. Thus far, the supervised label information has been added to ELM-DAE. If we want ELM-DAE to be suitable for data with local distribution characteristics, it is necessary to set the number of neighboring points, which is an additional parameter that needs to be tuned. In order to avoid manually defining $\mathbf{W}$, in future work, we will employ the adaptive strategy to make ELM-DAE suitable for classification problems with complex data distributions [48].

In the remaining part of this section, we will compare our ELM-DAE with the related supervised ELM-AE algorithms discussed in Section II-B. As the RMLDM-AE algorithm can only perform equal-dimensional feature transformation, we will exclude it from further discussion.

- ***ELM-DAE vs. GELM-AE and RMLFM-AE:*** They add only one regularization term into ELM-AE. These terms are $\mathrm{tr}(\boldsymbol{\beta}\mathbf{X}^{\mathrm{T}}\mathbf{L}\mathbf{X}\boldsymbol{\beta}^{\mathrm{T}})$, $\mathrm{tr}(\boldsymbol{\beta}^{\mathrm{T}}\mathbf{H}^{\mathrm{T}}\mathbf{L}_{\mathrm{GELM}}\mathbf{H}\boldsymbol{\beta})$, and $\mathrm{tr}(\boldsymbol{\beta}^{\mathrm{T}}\mathbf{L}_{\mathrm{LLE}}\boldsymbol{\beta})$, respectively. Therefore, the number of trade-off hyperparameters is two, which is less than that of SELM-AE. $\mathbf{L}$ is computed in the same way as $\mathbf{L}_{\mathrm{GELM}}$ without any additional parameters. However, as mentioned earlier, ELM-DAE applies the discriminant regularizer directly to the desired low-dimensional features, whereas GELM-AE and RMLFM-AE do not. Therefore, from the perspective of feature learning, our method should be more effective.
- ***ELM-DAE vs. DMELM-AE and GEELM-AE:*** Their regularization terms are as follows: $\mathrm{tr}(\boldsymbol{\beta}X^{\mathrm{T}}\mathbf{L}\mathbf{X}\boldsymbol{\beta}^{\mathrm{T}})$, $\mathrm{tr}\{\boldsymbol{\beta}^{\mathrm{T}}\mathbf{H}^{\mathrm{T}}(\mathbf{L}_b^{-1/2})^{\mathrm{T}}\mathbf{L}_w\mathbf{L}_b^{-1/2}\mathbf{H}\boldsymbol{\beta}\}$, and $\mathrm{tr}(\boldsymbol{\beta}^{\mathrm{T}}\mathbf{S}_p^{\dagger}\mathbf{S}_i\boldsymbol{\beta})$. It can be seen that DMELM-AE and GEELM-AE contain more discriminative information compared to ELM-DAE. However, the four matrices $\mathbf{L}_w$, $\mathbf{L}_b$, $\mathbf{S}_p$, and $\mathbf{S}_i$ are constructed using graph theory, which needs to tune additional parameters, such as the number of neighboring points [31]. Moreover, DMELM-AE and GEELM-AE also do not impose the discriminant regularization terms on the desired features.
- ***ELM-DAE vs. SELM-AE:*** Our algorithm is directly inspired by SELM-AE. In terms of the number of parameters to be tuned, SELM-AE has four hyperparameters (Q, C, ν, and τ), whereas ELM-DAE has three (Q, C, and λ). Therefore, it is easier to tune our ELM-DAE algorithm, which is crucial for a real-time system. However, regardless of the number of parameters, it must be acknowledged that SELM-AE contains more discriminant information compared to our algorithm. Nevertheless, we have the subsequent feature fusion module to further enhance the performance.
- ***ELM-DAE vs. WSI-AE:*** WSI-AE is initially proposed for one-class classification, but it can be easily extended to solve two-class or multi-class classification tasks. WSI-AE has only one trade-off parameter. Compared to ELM-AE and ELM-DAE, the term $\|\boldsymbol{\beta}\|_{\mathrm{F}}^2$ is missed.

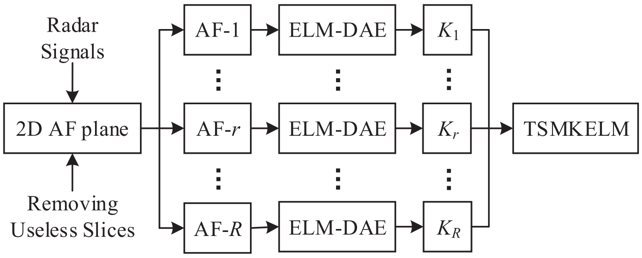


FIGURE 2. The proposed AF-based MK-ELM-DAE method for radar SEI.

In the context of classification problems, this term is important as it is used to control the model complexity.

In summary, the main advantages of ELM-DAE lie in its simple objective function, fewer parameters, and efficient solving method. Moreover, we can also observe its effectiveness in the experimental section. Therefore, we believe that the ELM-DAE algorithm is well-suited for practical systems.

C. MK-ELM-DAE

The MK-ELM-DAE algorithm is obtained by restricting the feature learning module of Fig. 1 to be ELM-DAE and the feature fusion module of Fig. 1 to be the two-stage MKL algorithm. The features are first enhanced by ELM-DAE and then fused by MKL. For the conventional methods, the features are often from the different domains of radar signals. In our paper, an AF-based MK-ELM-DAE fusion method is tailored for radar SEI. The AF-based time-frequency features are popular UMOP features [12], [13]. The performance of each Doppler shift slice of AF has been studied in [13], which shows that there exist lots of different informative AF slices in the 2D AF plane. So the AF slices can be considered as the primary multi-feature representations. Our method is shown in Fig. 2, where R denotes the number of AF slices, AF-r represents the r-th AF slice, and K_r is the corresponding kernel of AF-r. The main steps of MK-ELM-DAE for radar SEI are summarized as follows:

1) For each AF slice, ELM-DAE is first applied to extract the more discriminative feature representation.
2) With the enhanced new features, base kernels are constructed for the subsequent multi-kernel fusion.
3) Based on these multiple kernels, the fast LDR-based TSMKELM algorithm (see [13] for details) is used as the classifier to finish the classification task.

For a radar signal, we can usually obtain a great many AF slices from the 2D AF plane. To ensure the efficiency and effectiveness of our method, it is necessary to filter out the less useful AF slices before the first step. Here we adopt the linear discriminant analysis-based LDR criterion [31], which is more efficient than KDR. Assume that the AF slices are $\left\{\mathbf{x}_k^{u,v} \mid u = 1, \ldots, U; v = 1, \ldots, n_u\right\}_{k=1}^{K}$, where $\mathbf{x}_k^{u,v}$ denotes the v-th sample of the u-th class for the k-th AF slice, K is the total number of AF slices, U is the number of radar emitters, and n_u is the number of samples in the u-th class. The definition of LDR for the k-th AF slice is

$$\mathrm{LDR}_{\text{AF-}k} = \frac{\sum_{u=1}^{U}\sum_{v=1}^{n_u}\left(\mathbf{x}_k^{u,v} - \mathbf{m}_k^u\right)^{\mathrm{T}}\left(\mathbf{x}_k^{u,v} - \mathbf{m}_k^u\right)}{\sum_{u=1}^{U} n_u \left(\mathbf{m}_k^u - \mathbf{m}_k\right)^{\mathrm{T}}\left(\mathbf{m}_k^u - \mathbf{m}_k\right)}, \quad (17)$$

where $\mathbf{m}_k^u$ and $\mathbf{m}_k$ are the mean vector of the u-th class and the global mean vector, respectively. LDR can measure the class separability of each AF slice. Sorting these LDRs from largest to smallest, we retain the top R largest LDRs and the corresponding slices $\{\text{AF-}1 \ldots, \text{AF-}r, \ldots, \text{AF-}R\}$ $(R \ll K)$. Then, the retained AF slices are fused by MK-ELM-DAE (see Fig. 2). In the TSMKELM algorithm, the LDRs are utilized as the multi-kernel fusion weights. Specifically, the kernel combination weights and the combined kernel are:

$$\theta_r = \mathrm{LDR}_{\text{AF-}r}, \ K_{\text{combine}} = \sum_{r=1}^{R} \theta_r K_r; \ \ r = 1, \ldots, R. \quad (18)$$

Finally, the KELM classifier is used with K_{combine}. Algorithm 1 gives the pseudo-code of the proposed AF-based MK-ELM-DAE method for radar SEI. In summary, our method has the following advantages:

Algorithm 1 MK-ELM-DAE Method for Radar SEI

```
Given:
  The $d$-dimensional AF slices of radar signals, the number of AF slices to be reserved $R$, the number of hidden nodes $Q$, the parameters $C$ and $\lambda$, the activation function $g(\cdot)$.
Steps:
  (a) Compute LDR for each AF slice by (17) and then sort these LDRs in descending order;
  (b) Keep the top $R$ largest LDRs and their corresponding AF slices $\{\text{AF-}1 \ldots, \text{AF-}r, \ldots, \text{AF-}R\}$;
  (c) For each retained AF slice:
      (c.1) Generate $\{(\mathbf{a}_q \in \mathbb{R}^d, b_q \in \mathbb{R})\}_{q=1}^{Q}$ randomly and then orthogonalize them;
      (c.2) Calculate the hidden-layer output matrix $\mathbf{H}$, the adjacency matrix $\mathbf{W}$, and the matrix $\mathbf{L}$;
      (c.3) Optimize the projection matrix $\boldsymbol{\beta} \in \mathbb{R}^{Q \times d}$ by (12) and compute it by (16);
      (c.4) Project the input data ($d$-dimensional AF slices) into the $Q$-dimensional space ($Q < d$);
  (d) Construct the multiple base kernels $\{K_1, \ldots, K_R\}$ for the new features and then compute the combined kernel with the kernel weights defined by LDR;
  (e) Use the combined kernel and KELM to finish the classification task.
```

1) Compared to most DL methods, ELM-DAE is much more efficient for feature learning. We can see from (16) that ELM-DAE is much easier to solve. ELM-DAE is inspired by the earliest "auto-encoder" [33] and has considered the label information, so it is also effective, which will be validated in the experiments.
2) The MK-ELM-DAE method uses the fast TSMKELM algorithm [13] to conduct multi-kernel fusion. Hence,

TABLE 2. Basic information about the experimental datasets.

Name	# Classes	# Samples	Signal Length	# AF Slices (K)
Data-1	20	1000	1100	550
Data-2	30	3000	512	256

both the feature learning stage and the classification stage of our method are efficient.

3) The simple LDR criterion, which is designed to keep the informative AF slices from the AF plane, not only helps to reduce the training time but also can be regarded as the kernel weights for the TSMKELM algorithm.

It is worth noting that our proposed method can be applied to other classification applications as well. Fig. 1 presents a general framework for classification tasks, whereas Fig. 2 specifically illustrates an AF-based multi-kernel fusion framework for radar SEI. The LDR criterion can be used as a fixed rule to calculate the kernel combination weights [19].

IV. EXPERIMENTS

In this section, we present the experimental details, including datasets, experimental settings, and experimental results. We use the laptop equipped with Intel Core i7 CPU, 20G memory, and MATLAB R2020b platform to do the experiments.

A. DATASETS

Table 2 presents detailed information on the experimental datasets, which are collected from a real scenario. By comparing Table 1 and Table 2, we can observe that the quantity of our experimental data is relatively small. The typical samples of Data-1 and Data-2 are shown in Fig. 3 and Fig. 4, respectively. For Data-1, there are totally 550 (i.e., $K = 550$) AF slices and the dimension of each AF slice is 550. For Data-2, the number of AF slices is 256 (i.e., $K = 256$) and the dimension of each AF slice is 256. Thus, it is indeed necessary to remove the less important AF slices. Otherwise, too many slices (i.e., too large K) will lead to inefficiency.

B. EXPERIMENTAL SETTINGS

We use the single best AF slice feature and multiple AF slices features to validate our method. Each slice is normalized to have unit length. To save time, only ten AF slices (i.e., AF-1 to AF-10) are taken as the searching space to obtain the best AF slice. For the single best AF slice, the compared methods are ELM, KELM, and ELM-DAE. Our ELM-DAE is just for feature learning, so it is followed by KELM to perform classification. For multiple AF slices, KDR-TSMKELM [13] and MK-ELM-DAE are compared. For each dataset, 6%, 8%, 10%, 20%, and 50% of the samples per class are respectively selected for training and the rest samples are used for testing. We generate ten different training/test set partitions to avoid the bias caused by the random selection of training samples, and thus the average recognition rate and standard deviation are reported for fair comparison.

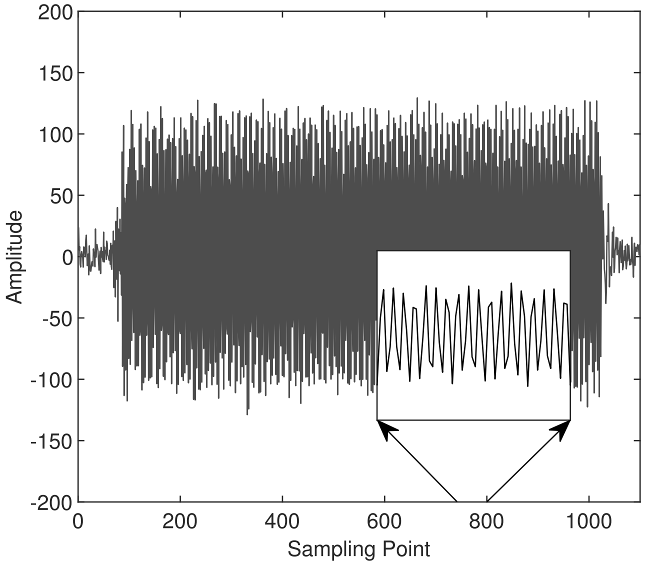


FIGURE 3. A typical sample of Data-1.

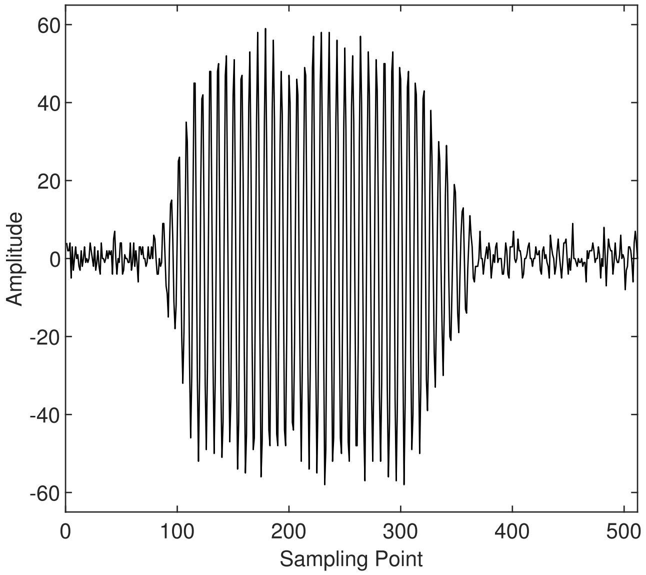


FIGURE 4. A typical sample of Data-2.

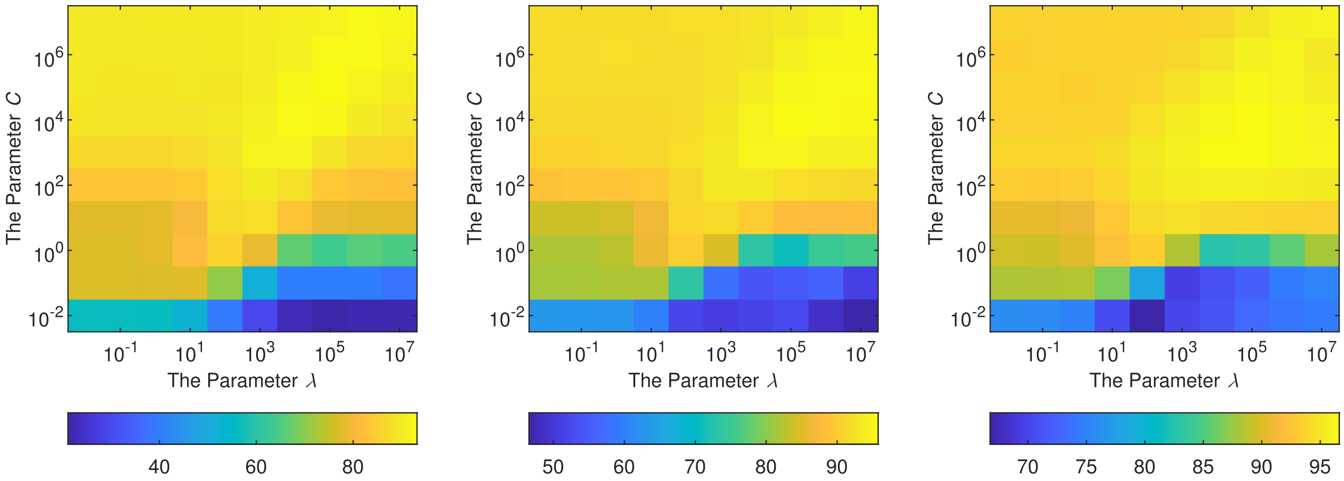
There are some hyperparameters to be specified for each method. For KELM, the Gaussian kernel is employed and its parameter is set as the mean of pairwise distances of training samples. The parameter C of KELM is set as 100. As we will see from the experimental results, KELM is robust with the above empirical parameter settings. Thus, KELM can be viewed as a parameter-free algorithm to some extent. For simplicity, the best AF slice is found according to the testing accuracy of KELM, and then ELM and ELM-DAE are directly applied to the best AF slice determined by KELM. For ELM, we set the activation function as the Sigmoid function and the number of hidden

TABLE 3. The identification results of Data-1 (Mean ± standard deviation, %). Best results are marked in bold.

Feature(s)	Method	6%-train	8%-train	10%-train	20%-train	50%-train
single best AF slice	KELM	86.17 ± 1.67	88.73 ± 1.21	90.22 ± 1.92	94.99 ± 0.50	97.74 ± 0.49
	ELM	85.20 ± 1.79	86.86 ± 1.35	89.62 ± 1.26	94.98 ± 0.77	96.68 ± 0.74
	ELM-DAE	89.19 ± 1.83	92.09 ± 1.69	93.19 ± 1.68	96.96 ± 0.24	97.78 ± 0.39
multiple AF slices (AF-1 to AF-10)	KDR-TSMKELM	89.21 ± 1.04	91.16 ± 0.89	91.87 ± 1.04	96.14 ± 0.37	97.92 ± 0.58
	MK-ELM-DAE	93.29 ± 1.39	95.09 ± 1.23	95.78 ± 0.85	97.54 ± 0.20	98.28 ± 0.61
multiple AF slices ($R = 40$)	KDR-TSMKELM	91.10 ± 1.17	92.32 ± 0.84	93.66 ± 1.12	97.01 ± 0.48	98.10 ± 0.58
	MK-ELM-DAE	$\mathbf{94.90 \pm 1.20}$	$\mathbf{96.22 \pm 0.80}$	$\mathbf{96.62 \pm 0.75}$	$\mathbf{98.10 \pm 0.00}$	$\mathbf{98.44 \pm 0.41}$

TABLE 4. The identification results of Data-2 (Mean ± standard deviation, %). Best results are marked in bold.

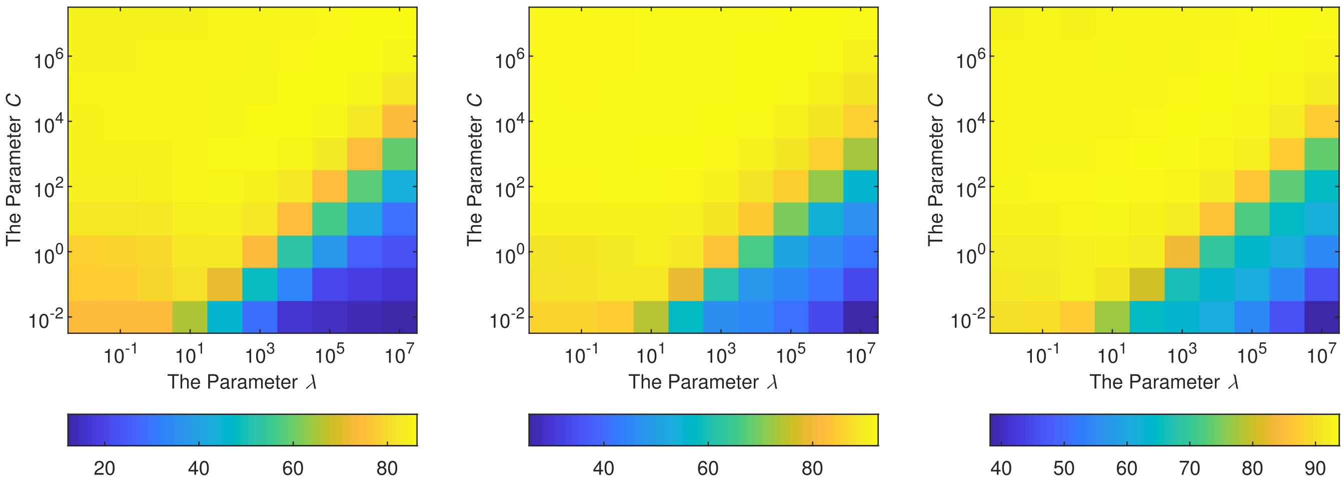
Feature(s)	Method	6%-train	8%-train	10%-train	20%-train	50%-train
single best AF slice	KELM	80.27 ± 1.31	83.20 ± 1.14	85.02 ± 0.62	89.74 ± 0.34	93.25 ± 0.56
	ELM	76.64 ± 1.67	79.23 ± 1.27	81.27 ± 0.64	86.56 ± 0.61	92.34 ± 0.75
	ELM-DAE	82.43 ± 1.58	84.96 ± 1.06	86.29 ± 0.57	90.04 ± 0.34	93.43 ± 0.62
multiple AF slices (AF-1 to AF-10)	KDR-TSMKELM	88.60 ± 1.23	91.04 ± 0.44	92.14 ± 0.56	95.02 ± 0.63	97.03 ± 0.38
	MK-ELM-DAE	89.41 ± 1.30	91.63 ± 0.36	92.55 ± 0.49	95.13 ± 0.53	97.21 ± 0.32
multiple AF slices ($R = 40$)	KDR-TSMKELM	89.50 ± 1.45	91.78 ± 0.53	93.07 ± 0.47	95.45 ± 0.30	97.37 ± 0.30
	MK-ELM-DAE	$\mathbf{90.67 \pm 1.37}$	$\mathbf{92.57 \pm 0.51}$	$\mathbf{93.77 \pm 0.43}$	$\mathbf{95.73 \pm 0.26}$	$\mathbf{97.47 \pm 0.40}$



(a) ELM-DAE
(b) MK-ELM-DAE (AF-1 to AF-10)
(c) MK-ELM-DAE ($R = 40$)

FIGURE 5. The performance heatmaps with the variation of C and λ for Data-1.

nodes Q as 5000 from experience. The performance of ELM with Sigmoid activation function is not sensitive to Q, so only the trade-off parameter C needs to be tuned [21]. We choose C from $\{10^{-2}, \ldots, 10^{7}\}$. We also use the Sigmoid activation function for ELM-DAE. The parameters Q, C, and λ of ELM-DAE need to be tuned. We first set Q as 200 ($Q < d$), and then C and λ are selected from $\{10^{-2}, \ldots, 10^{7}\}$. The parameter selection is conducted by the grid search strategy. After the optimal C and λ are determined, we will alter the value of Q in the range of $\{50 : 50 : 500\}$ for Data-1 and



(a) ELM-DAE (b) MK-ELM-DAE (AF-1 to AF-10) (c) MK-ELM-DAE ($R = 40$)

FIGURE 6. The performance heatmaps with the variation of C and λ for Data-2.

{50 : 50 : 250} for Data-2. For multi-kernel fusion methods, there are two settings: 1) the ten AF slices (AF-1 to AF-10) are fused; 2) the AF slices corresponding to the top R largest KDRs or LDRs are fused. Considering the training time cost, we fix R to 40 for the two datasets. Additionally, we will discuss the relationship between the accuracy and R.

It should be mentioned that our training samples are scarce in the small sample size setting. For instance, in Data-1, the number of training samples per class is 3 (6%), 4 (8%), 5 (10%), 10 (20%), and 25 (50%), respectively. In non-cooperative scenarios, it is likely to intercept such a small number of signals. Even with the simple 1D-CNN model proposed in [9] (two convolutional layers, one max-pooling layer, one dropout layer, and two fully-connected layers), it is difficult to directly train the model with such scarce samples. Therefore, we have not conducted experimental comparisons between our algorithm with DL models. Our code is available at: https://github.com/yyashi/MK-ELM-DAE.

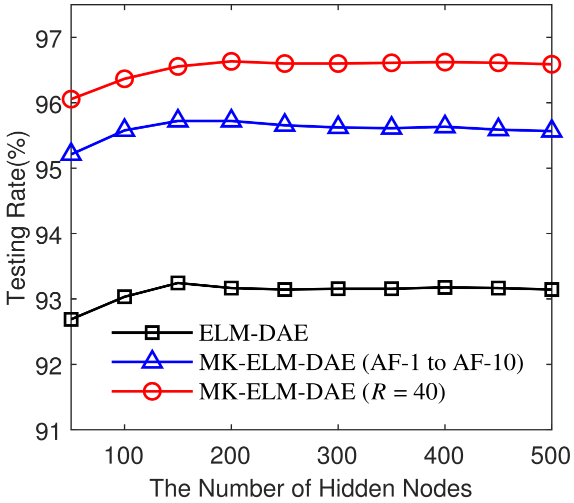


FIGURE 7. The testing rate of Data-1 with varying values of Q.

C. EXPERIMENTAL RESULTS

When Q is equal to 200 and R is fixed as 40, the average recognition accuracies of Data-1 and Data-2 are presented in Table 3 and Table 4, respectively. Taking 10%-train as an example, Fig. 5 and Fig. 6 show the performance heatmaps of ELM-DAE and MK-ELM-DAE for each dataset with varying parameter C and parameter λ. Based on these results, we can draw the following conclusions:

- For the single best AF slice feature, ELM-DAE has good performance especially when the number of training samples is small. For Data-1, the accuracy gaps between the best ELM-DAE and the second-best KELM are 3.02%, 3.36%, 2.97%, 1.97% and 0.04%. For Data-2, the accuracy gaps are 2.16%, 1.76%, 1.27%, 0.30%, and 0.18%. ELM-DAE can indeed improve the discriminability of the primary features, so the effectiveness of ELM-DAE has been testified.
- Since the discriminability of each primary feature has been enhanced by ELM-DAE, our proposed MK-ELM-DAE method exhibits superior performance compared to KDR-TSMKELM, particularly on Data-1. For example, when the top R AF slices are fused, the performance gaps between MK-ELM-DAE and KDR-TSMKELM are 3.80%, 3.90%, 2.96%, 1.09%, and 0.34%. We can see that both ELM-DAE and MK-ELM-DAE are helpful for the small training set problem. Besides, using more valuable information in the AF plane is better than using a few slices (e.g., AF-1 to AF-10), which will be discussed later by varying the number of AF slices.
- Fig. 5 and Fig. 6 demonstrate that the performance of ELM-DAE and MK-ELM-DAE is robust across a wide range of parameter values for C and λ. Therefore, our methods have a lighter burden of parameter selection.

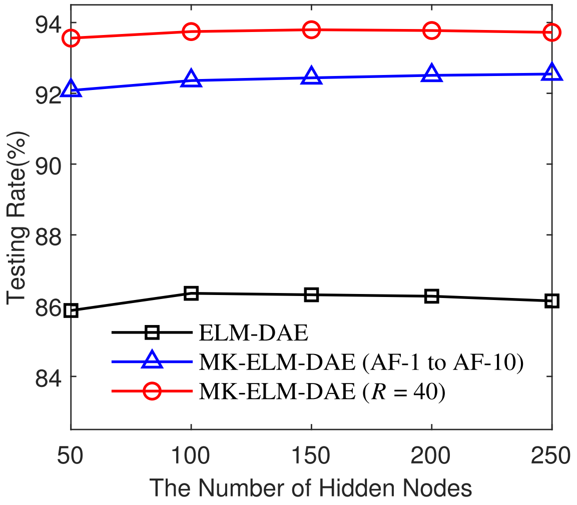


FIGURE 8. The testing rate of Data-2 with varying values of Q.

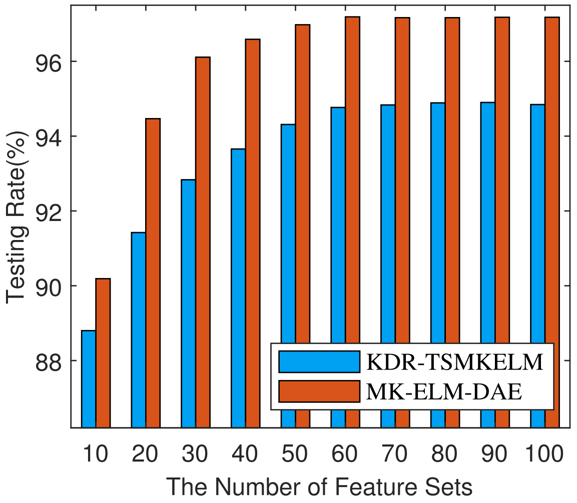


FIGURE 9. The testing rate of Data-1 with varying values of R.

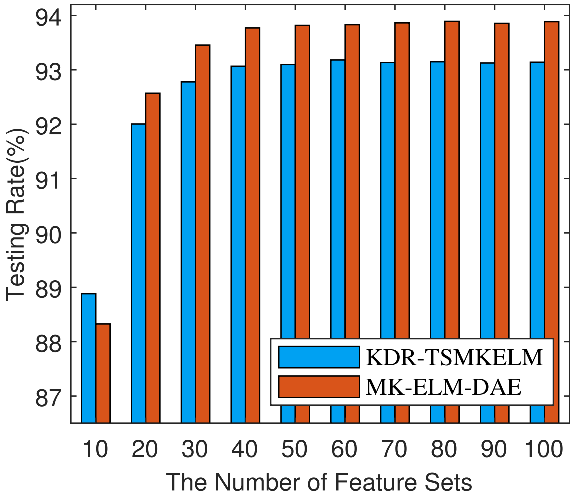


FIGURE 10. The testing rate of Data-2 with varying values of R.

TABLE 5. Running time comparison of KDR-TSMKELM and MK-ELM-DAE on Data-1 and Data-2 (seconds).

Dataset		KDR-TSMKELM		MK-ELM-DAE	
		Training	Test	Training	Test
Data-1	6%-train	0.664	0.039	7.530	0.011
	8%-train	0.826	0.048	7.551	0.011
	10%-train	0.989	0.055	7.523	0.011
	20%-train	1.778	0.082	7.604	0.014
	50%-train	6.232	0.204	10.458	0.017
Data-2	6%-train	0.603	0.059	2.151	0.012
	8%-train	0.795	0.073	2.427	0.013
	10%-train	1.001	0.083	2.642	0.014
	20%-train	2.869	0.152	4.555	0.018
	50%-train	13.113	0.328	13.591	0.061

Furthermore, we will discuss the impact of Q and R on the recognition accuracy. Considering the case of 10%-train of Data-1 and Data-2, Fig. 7 and Fig. 8 illustrate the relationship between the testing rate and the number of hidden nodes Q. Fig. 9 and Fig. 10 show the relationship between the testing rate and the number of feature sets R. In our experiments, we can see that ELM-DAE and MK-ELM-DAE consistently perform well when Q is greater than 100. As R increases, the performance of KDR-TSMKELM and MK-ELM-DAE is improved and tends to be stable. However, the larger R means the slower training speed. Consequently, it is important to select a suitable value of R to keep the balance between performance and training time.

Finally, we present a comparison of the training time and test time (per testing sample) for the two multi-kernel fusion methods in Table 5 ($R = 40$). The time-consuming part of the MK-ELM-DAE method is the ELM-DAE modules. Since ELM-DAE needs to be applied to every selected AF slice, the training time cost of MK-ELM-DAE is heavier than that of KDR-TSMKELM. However, the testing efficiency of MK-ELM-DAE is greatly higher than that of KDR-TSMKELM, which is benefited from the DR ability of ELM-DAE. It is worth noting that the number of feature sets R are usually not too large in general pattern classification tasks [20], so the training time of MK-ELM-DAE is acceptable.

In summary, our proposed MK-ELM-DAE algorithm exhibits outstanding effectiveness and efficiency. MK-ELM-DAE combines the benefits of both feature learning and multi-kernel fusion, so it is superior to the compared methods when applied to the radar SEI task. As the small sample size problem is prevalent in real-world application scenarios, this paper emphasizes the utilization of traditional non-deep DR methods for feature learning, rather than relying on DL models. With the development of data-limited DL methods [42], [43], [44], we will investigate the problem of representation learning for radar signals under small sample size condition in our future work. Nevertheless, MK-ELM-DAE still has a time advantage over the DL-based methods (see Table 5).

V. CONCLUSION

Based on representation learning and multi-kernel fusion, we propose the MK-ELM-DAE method for radar SEI. For radar signals, the AF slices are used as the primary UMOP features, and then a LDR criterion is devised to keep the more informative slices. With the selected AF slices, our proposed supervised DR algorithm, ELM-DAE, is first employed to extract the more discriminative feature representations. Then, the fast two stage multi-kernel ELM algorithm is utilized to do classification. Experimental results show that our method can improve the identification accuracy particularly when the training sample size is small. Furthermore, the training speed of our method is fast. In the future work, to further enhance the performance of MK-ELM-DAE, we will incorporate richer discriminant information to ELM-DAE. We can also leverage the stacking technique to construct a multilayer ELM-DAE network, which will explore deeper and more discriminative representations. Moreover, a multi-kernel multilayer ELM-DAE algorithm can be derived.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers and the associate editor for their helpful comments and suggestions, which greatly contributed to the enhancement of both the technical content and the quality of this article.

REFERENCES

[1] K. I. Talbot, P. R. Duley, and M. H. Hyatt, "Specific emitter identification and verification," *Technol. Rev. J.*, vol. 11, no. 1, pp. 113–133, Jan. 2003.

[2] G. Gok, Y. K. Alp, and O. Arikan, "A new method for specific emitter identification with results on real radar measurements," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 3335–3346, 2020.

[3] F. Zhang, C. Luo, J. Xu, Y. Luo, and F.-C. Zheng, "Deep learning based automatic modulation recognition: Models, datasets, and challenges," *Digit. Signal Process.*, vol. 129, Sep. 2022, Art. no. 103650.

[4] Z. Huang, Z. Ma, and G. Huang, "Radar waveform recognition based on multiple autocorrelation images," *IEEE Access*, vol. 7, pp. 98653–98668, 2019.

[5] X. Wu, X. Hou, Q. Huang, and Q. Bu, "Radar emitter identification algorithm based on fuzzy set theory," *Command. Inf. Syst. Technol.*, vol. 9, no. 2, pp. 55–59, Apr. 2018.

[6] L. E. Langley, "Specific emitter identification (SEI) and classical parameter fusion technology," in *Proc. WESCON*, San Francisco, CA, USA, Sep. 1993, pp. 377–381.

[7] H. Ye, Z. Liu, and W. Jiang, "Comparison of unintentional frequency and phase modulation features for specific emitter identification," *Electron. Lett.*, vol. 48, no. 14, pp. 875–877, Jul. 2012.

[8] L. Sun, X. Wang, and Z. Huang, "Unintentional modulation evaluation in time domain and frequency domain," *Chin. J. Aeronaut.*, vol. 35, no. 4, pp. 376–389, Apr. 2022.

[9] Z.-M. Liu, "Multi-feature fusion for specific emitter identification via deep ensemble learning," *Digit. Signal Process.*, vol. 110, Mar. 2021, Art. no. 102939.

[10] Y. Zhou, X. Wang, Y. Chen, and Y. Tian, "Specific emitter identification via bispectrum-radon transform and hybrid deep model," *Math. Probl. Eng.*, vol. 2020, p. 7646527, Jan. 2020.

[11] P. Chen, Y. Guo, G. Li, and J. Wan, "Adversarial shared-private networks for specific emitter identification," *Electron. Lett.*, vol. 56, no. 6, pp. 296–299, Mar. 2020.

[12] T. Wan, H. Ji, W. Xiong, B. Tang, X. Fang, and L. Zhang, "Deep learning-based specific emitter identification using integral bispectrum and the slice of ambiguity function," *Signal, Image Video Process.*, vol. 16, no. 7, pp. 2009–2017, Feb. 2022.

[13] Y. Shi, M. Yuan, J. Ren, and S. Xu, "Specific radar emitter identification based on two stage multiple kernel extreme learning machine," *Electron. Lett.*, vol. 57, no. 18, pp. 699–701, Aug. 2021.

[14] M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim, "Do we need hundreds of classifiers to solve real world classification problems?" *J. Mach. Learn. Res.*, vol. 15, pp. 3133–3181, Jan. 2014.

[15] Z. Xiong, W. Lyu, W. Wu, J. Xu, and X. Li, "Application and development of artificial intelligence technology for intelligence reconnaissance field," *Command. Inf. Syst. Technol.*, vol. 10, no. 5, pp. 8–13, Oct. 2019.

[16] A. Shrestha and A. Mahmood, "Review of deep learning algorithms and architectures," *IEEE Access*, vol. 7, pp. 53040–53065, 2019.

[17] S. Sengupta, S. Basak, P. Saikia, S. Paul, V. Tsalavoutis, F. Atiah, V. Ravi, and A. Peters, "A review of deep learning with special emphasis on architectures, applications and recent trends," *Knowl.-Based Syst.*, vol. 194, Apr. 2020, Art. no. 105596.

[18] G. R. G. Lanckriet, N. Cristianini, P. Bartlett, L. E. Ghaoui, and M. I. Jordan, "Learning the kernel matrix with semideflnite programming," *J. Mach. Learn. Res.*, vol. 5, pp. 27–72, Dec. 2004.

[19] M. Gönen and E. Alpaydın, "Multiple kernel learning algorithms," *J. Mach. Learn. Res.*, vol. 12, pp. 2211–2268, Feb. 2011.

[20] S. S. Bucak, R. Jin, and A. K. Jain, "Multiple kernel learning for visual object recognition: A review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 7, pp. 1354–1369, Jul. 2014.

[21] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Trans. Syst. Man, Cybern. B, Cybern.*, vol. 42, no. 2, pp. 513–529, Apr. 2012.

[22] J. Wang, S. Lu, S. H. Wang, and Y. D. Zhang, "A review on extreme learning machine," *Multimedia Tools Appl.*, vol. 81, pp. 41611–41660, May 2021.

[23] L. L. C. Kasun, H. Zhou, G. B. Huang, and C. M. Vong, "Representational learning with extreme learning machine for big data," *IEEE Intell. Syst.*, vol. 28, no. 6, pp. 31–34, Nov./Dec. 2013.

[24] J. Tang, C. Deng, and G.-B. Huang, "Extreme learning machine for multilayer perceptron," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 4, pp. 809–821, Apr. 2016.

[25] J. Du, C.-M. Vong, C. Chen, P. Liu, and Z. Liu, "Supervised extreme learning machine-based auto-encoder for discriminative feature learning," *IEEE Access*, vol. 8, pp. 11700–11709, 2020.

[26] J. Zhang, Y. Li, W. Xiao, and Z. Zhang, "Non-iterative and fast deep learning: Multilayer extreme learning machines," *J. Franklin Inst.*, vol. 357, no. 13, pp. 8925–8955, Sep. 2020.

[27] T. Hofmann, B. Schölkopf, and A. J. Smola, "Kernel methods in machine learning," *Ann. Statist.*, vol. 36, no. 3, pp. 1171–1220, Jun. 2008.

[28] Y. Peng, S. Wang, X. Long, and B.-L. Lu, "Discriminative graph regularized extreme learning machine and its application to face recognition," *Neurocomputing*, vol. 149, pp. 340–353, Feb. 2015.

[29] Y. Peng and B.-L. Lu, "Discriminative manifold extreme learning machine and applications to image and EEG signal classification," *Neurocomputing*, vol. 174, pp. 265–277, Jan. 2016.

[30] A. Iosifidis, A. Tefas, and I. Pitas, "Graph embedded extreme learning machine," *IEEE Trans. Cybern.*, vol. 46, no. 1, pp. 311–324, Jan. 2016.

[31] S. Yan, D. Xu, B. Zhang, H.-J. Zhang, Q. Yang, and S. Lin, "Graph embedding and extensions: A general framework for dimensionality reduction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 1, pp. 40–51, Jan. 2007.

[32] M. Rezaei-Ravari, M. Eftekhari, and F. Saberi-Movahed, "Regularizing extreme learning machine by dual locally linear embedding manifold learning for training multi-label neural network classifiers," *Eng. Appl. Artif. Intell.*, vol. 97, Jan. 2021, Art. no. 104062.

[33] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, Jul. 2006.

[34] C. M. Wong, C. M. Vong, P. K. Wong, and J. Cao, "Kernel-based multilayer extreme learning machines for representation learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 3, pp. 757–762, Mar. 2018.

[35] T. Wang, J. Cao, X. Lai, and Q. M. J. Wu, "Hierarchical one-class classifier with within-class scatter-based autoencoders," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 8, pp. 3770–3776, Aug. 2021.

[36] L. Yang, S. Song, S. Li, Y. Chen, and G. Huang, "Graph embedding-based dimension reduction with extreme learning machine," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 51, no. 7, pp. 4262–4273, Jul. 2021.

[37] M. Rezaei-Ravari, M. Eftekhari, and F. Saberi-Movahed, "ML-CK-ELM: An efficient multi-layer extreme learning machine using combined kernels for multi-label classification," *Scientia Iranica*, vol. 27, no. 6, pp. 3005–3018, Nov. 2020.

[38] U. G. Mangai, S. Samanta, S. Das, and P. R. Chowdhury, "A survey of decision fusion and feature fusion strategies for pattern classification," *IETE Tech. Rev.*, vol. 27, no. 4, pp. 293–307, Sep. 2010.

[39] L. Gao, L. Qi, E. Chen, and L. Guan, "Discriminative multiple canonical correlation analysis for information fusion," *IEEE Trans. Image Process.*, vol. 27, no. 4, pp. 1951–1965, Apr. 2018.

[40] X. Liu, L. Wang, G.-B. Huang, J. Zhang, and J. Yin, "Multiple kernel extreme learning machine," *Neurocomputing*, vol. 149, pp. 253–264, Feb. 2015.

[41] Y. Shi and H. Ji, "Kernel canonical correlation analysis for specific radar emitter identification," *Electron. Lett.*, vol. 50, no. 18, pp. 1318–1320, Aug. 2014.

[42] L. Alzubaidi, J. Bai, A. Al-Sabaawi, J. Santamaría, A. S. Albahri, B. S. N. Al-Dabbagh, M. A. Fadhel, M. Manoufali, J. Zhang, A. H. Al-Timemy, Y. Duan, A. Abdullah, L. Farhan, Y. Lu, A. Gupta, F. Albu, A. Abbosh, and Y. Gu, "A survey on deep learning tools dealing with data scarcity: Definitions, challenges, solutions, tips, and applications," *J. Big Data*, vol. 10, no. 1, p. 46, Apr. 2023.

[43] G. Iglesias, E. Talavera, Á. González-Prieto, A. Mozo, and S. Gómez-Canaval, "Data augmentation techniques in time series domain: A survey and taxonomy," *Neural Comput. Appl.*, vol. 35, no. 14, pp. 10123–10145, Mar. 2023.

[44] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *J. Big Data*, vol. 6, p. 60, Dec. 2019.

[45] X. He and P. Niyogi, "Locality preserving projections," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Vancouver, BC, Canada, 2004, pp. 153–160.

[46] Z. Zheng, F. Yang, W. Tan, J. Jia, and J. Yang, "Gabor feature-based face recognition using supervised locality preserving projection," *Signal Process.*, vol. 87, no. 10, pp. 2473–2483, Oct. 2007.

[47] R. H. Bartels and G. W. Stewart, "Solution of the matrix equation AX+XB=C [F4]," *Commun. ACM*, vol. 15, no. 9, pp. 820–826, Sep. 1972.

[48] X. Li, Q. Wang, F. Nie, and M. Chen, "Locality adaptive discriminant analysis framework," *IEEE Trans. Cybern.*, vol. 52, no. 8, pp. 7291–7302, Aug. 2022.

YA SHI received the B.S. degree in electronic and information engineering and the M.S. and Ph.D. degrees in pattern recognition and intelligent system from Xidian University, Xi'an, China, in 2008, 2011, and 2015, respectively. Since 2016, she has been a Lecturer with the School of Information and Control Engineering, Xi'an University of Architecture and Technology. Her research interests include machine learning, computer vision, and signal recognition.

CHENYI WANG received the B.S. degree in information and computing science from the Xi'an University of Finance and Economics, Xi'an, China, in 2021. Currently, he is pursuing the M.S. degree with the School of Information and Control Engineering, Xi'an University of Architecture and Technology. His current research interests include deep learning and object detection.

GUANGFU GE received the B.S. and M.S. degrees from the School of Electronic Engineering, Xidian University, Xi'an, China, in 2008 and 2011, respectively. From 2011 to 2016, he was with Nanjing Corad Electronic Equipment Company Ltd. Currently, he is with the Nanjing Research Institute of Electronic Engineering. His research interests include army command information systems, cloud computing, and artificial intelligence.

FEIXIANG LIU received the B.S. degree in electrical engineering and automation from Luoyang Normal University, Luoyang, China, in 2022. Currently, he is pursuing the M.S. degree with the School of Information and Control Engineering, Xi'an University of Architecture and Technology. His current research interests include deep learning and object detection.

LELE ZHANG received the B.S. degree in network engineering from Taiyuan University, Taiyuan, China, in 2021. Currently, he is pursuing the M.S. degree with the School of Information and Control Engineering, Xi'an University of Architecture and Technology. His current research interests include deep learning and time series analysis.