IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 15 January 2024, accepted 16 February 2024, date of publication 27 February 2024, date of current version 12 March 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3371259

RESEARCH ARTICLE

Bias-Boosted ELM for Knowledge Transfer in Brain Emotional Learning for Time Series Forecasting

SUTHASINEE IAMSA-AT, PUNYAPHOL HORATA, AND KHAMRON SUNAT, (Associate Member, IEEE)
Department of Computer Science, College of Computing, Khon Kaen University, Khon Kaen 400002, Thailand
Corresponding author: Punyaphol Horata (punhor1@kku.ac.th)

This work was supported by the Graduate Education of Computer and Information Science Interdisciplinary Research Grant from the Department of Computer Science, College of Computing, Khon Kaen University, Thailand.

ABSTRACT This paper presents the Bias-Boosted Extreme Learning Machine guided Brain Emotional Learning (B2ELM-BEL) model, a significant advancement in chaotic time series prediction that effectively incorporates knowledge transfer learning. Integrating traditional Brain Emotional Learning (BEL) with the novel Biased-ELM method, the B2ELM-BEL introduces a bias term into the output weights of Extreme Learning Machines (ELM). This addition enhances the model's predictive accuracy, proving particularly beneficial in configurations with a minimal number of hidden nodes. Our evaluation of the B2ELM-BEL model across various datasets, including complex chaotic time-series benchmarks and real-world scenarios, demonstrates its superior performance over several BEL models. It achieves lower mean RMSE, MAE, and SMAPE values, and exhibits enhanced generalizability and efficiency. The findings indicate that while the single hidden node variant of B2ELM-BEL is suitable for simpler tasks, the multi-node version is more adept at handling challenging environments. This highlights the necessity of tailoring the model to the complexity of the specific dataset being analyzed.

INDEX TERMS Chaotic time-series forecasting, brain emotional learning (BEL), extreme learning machine (ELM), transfer learning, bias-boosted prediction models, machine learning robustness.

I. INTRODUCTION

Many real-world systems exhibit dynamic properties that evolve and exhibit chaotic characteristics. The term "chaos" suggests that these systems are susceptible to minor changes in their initial conditions, rendering their behavior largely unpredictable. Chaotic time-series problems manifest these complexities and hold significant importance in various domains. They find extensive applications in areas such as wind power prediction using chaotic time series [1], financial time series forecasting [2], [3], and weather forecasting [4].

Efforts have been made to employ machine learning and artificial neural networks (ANN) to address time-series prediction challenges. For example, Wang et al. [5] introduced an improved extreme learning machine for the online sequential prediction of multivariate time series. Duduku et al. [6] presented a hybrid approach that combines temporal convolutional networks and recurrent network methods, yielding promising results.

Recent advancements in transfer learning (TL) or knowledge transfer (KT) have significantly impacted the field of machine learning, particularly in tasks related to recognition and classification, as evidenced by the works of [7], [8], and [9]. This approach empowers the improvement of predictions in target tasks by leveraging insights from previously established source domains. Notably, Obst et al. [10] have contributed a theoretical framework that harnesses TL for linear model predictions.

Among the techniques in this domain, the Extreme Learning Machine (ELM), a prominent artificial neural network (ANN) method, stands out. ELM is characterized by its random initialization of weights between inputs and

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

hidden layers, as well as its analytical determination of output weights through a generalized inverse operation. These features expedite the training phase and ensure robust generalization performance, distinguishing ELM from traditional neural networks. In the context of knowledge transfer, ELM has proven to be an effective method, enhancing the adaptability of machine learning models, a subject extensively reviewed by Salaken et al. [11].

The Brain Emotional Learning (BEL) models are computational representations inspired by the emotional processes of the mammalian brain, simulating signal propagation in emotional brain structures. Key components include the amygdala (AMYG), which preserves emotional memories and interfaces with both the orbitofrontal cortex (OFC) and thalamus. A notable innovation in this field is the emotional backpropagation (EmBP) learning algorithm, which integrates emotional variables, updated by anxiety and confidence parameters, into machine learning processes [12]. EmBP mimics emotional learning within the amygdala, a pivotal region in emotion processing [13].

These models find practical application in the brain-emotional-learning-based intelligent controller (BELBIC), inspired by the computational model of the mammalian limbic system. BELBIC, analogous to a linear-quadratic regulator (LQR) and designed for omni-directional robots, maximizes a quadratic reward function [14]. Furthermore, the Adaptive Decayed Brain Emotional Learning (ADBEL) network has demonstrated effectiveness in online time series forecasting, with its predictive capabilities further enhanced when integrated with a neo-fuzzy network [15]. This integration focuses on the OFC segment of the network and employs three membership functions for neo-fuzzy neurons in online prediction tasks [16].

An evolution in BEL models is marked by the integration of interval knowledge, allowing these models to better handle uncertainty and imprecision during the learning process [17]. In dynamic system control, the BELBIC Controller, inspired by emotional learning principles akin to those in the human brain, shows promise and has been compared to conventional feedback control systems such as the Proportional-Integral (PI) controller within specific applications, shedding light on the advantages and limitations of these innovative control strategies [18].

Another intriguing development combines emotional learning with the optimization of DVRs operations, optimizing dual objectives to potentially enhance DVR performance and reliability in mitigating voltage fluctuations [19]. Additionally, an adaptive control system has been conceptualized, merging elements from a typical BEL network and a self-organizing radial basis function network, with a focus on mobile robot control [20].

The BEL approach is gaining traction across various applications, including earthquake prediction, where emotional impact plays a significant role in understanding and preparing for seismic events [21]. Novel control strategies that integrate BEL with Adaptive Model Predictive Control [22] have been developed for induction motor drives, while methods combining Recursive Terminal Sliding-Mode Control (RTSMC) with a Double Hidden Layer Fuzzy Emotional Recurrent Neural Network (DHL-FERNN) have improved robustness and adaptability in controlling nonlinear systems [23]. However, emotional learning-based controllers, including the nonparametric ELBC, encounter challenges such as computational complexity and lack of robustness, especially in uncertain environments [24]. Recent proposals aim to enhance learning accuracy and convergence rates through dopamine-inspired methods and stochastic learning on high-dimensional datasets [25]. Additionally, a bionic memristive circuit, capable of replicating emotional learning, that can generate a spectrum of emotions through valence and arousal signals, has been designed [26].

While significant progress has been made in brain emotional learning (BEL) methods, challenges persist when dealing with intricate chaotic time series like the Henon problem. The existing BEL-IK model utilizes the 'max' function as the emotional neuron, emphasizing localized emotion influenced by the present context and its potential modulation by experiences. As a result, we aim to find a straightforward machine-learning model that facilitates effective knowledge transfer.

To address these challenges, this paper introduces an enhanced version of the existing BEL-IK model, leveraging transfer learning with Biased-ELM as the source of knowledge to enhance BEL-IK. Our contributions encompass two key aspects: the introduction of a Biased-ELM method that augments traditional ELM models by incorporating a bias term into the output weights and the introduction of an improved BEL-IK approach that selects the optimal Biased-ELM from multiple candidates based on error minimization.

In summary, we present the Bias-Boosted Extreme Learning Machine guided Brain Emotional Learning (BBELM-BEL or B2ELM-BEL) model. In this model, Biased-ELM serves as both an enhancer for traditional ELM models and a source of essential knowledge for the improved BEL-IK model. These contributions collectively enhance the BEL-IK approach's overall performance, particularly in handling complex chaotic time series data.

The paper is organized into the five following sections: Section II lays out the background knowledge on transfer learning, ELM, BEL, and the improved BEL version (BEL-IK) [17]. Section III details the newly proposed methods and algorithms. Section IV describes experiments that compare the performance of the new method against other BEL methods. The final section concludes the paper and outlines future research directions.

II. RELATED WORKS

A. KNOWLEDGE TRANSFER LEARNING

Knowledge transfer among animals often occurs in nature. This is exemplified by Namibian desert-dwelling elephants passing their unique survival skills and knowledge to subsequent generations [9], [27]. These elephants have adapted to the Namib desert's high-temperature environment, acquiring abilities not necessarily inherited through DNA [28]. Another example can be observed in humans, where experience and knowledge in one domain can enhance learning performance in another. For instance, those skilled at playing the guitar may learn to play another string instrument, such as the violin, more quickly.

Inspired by natural knowledge transfer, researchers have broadly applied this concept specifically in the machine learning field, where it has shown considerable promise. A general definition of transfer learning [8], [9] involves a domain $\mathcal{D} = \{\mathcal{X}, P(X)\}$, consisting of a feature space $\mathcal{X}$ and a marginal distribution $P(X)$, where $X = \{x_1, \ldots, x_n\} \in \mathcal{X}$. Additionally, it involves a task denoted as $\mathcal{T} = \{\mathcal{Y}, f(\cdot)\}$, comprising a label space $\mathcal{Y}$ and an objective predictive function $f(\cdot)$.

The goal of transfer learning is to improve the target prediction function $f_T(\cdot)$ in a target domain $\mathcal{D}_T = \{(x_{T_1}, y_{T_1}), \ldots, (x_{T_{n_T}}, y_{S_{n_T}})\}$ by leveraging knowledge from a source domain $\mathcal{D}_S = \{(x_{S_1}, y_{S_1}), \ldots, (x_{S_{n_S}}, y_{S_{n_S}})\}\}$. Here $x_{S_i} \in \mathcal{X}_S$ and $y_{S_i} \in \mathcal{Y}_S$ represent the corresponding target or class labels of $\mathcal{D}_S$, and $\mathcal{D}_S \subset \mathcal{D}_T$.

This paper explores the application of ELM models in predicting time series data. Our approach involves using a specific subset of data $\mathcal{D}_S$, which we have defined as the knowledge transfer learning space. This subset comprises elements $\{\mathbf{x}_i\}$, with i ranging from 1 to N_S, and is selected from a larger set of the training data $\mathcal{D}_T$. By training the ELM models using the $\mathcal{D}_S$ subset, our objective is to enhance the prediction accuracy of the BEL model, specifically in the context of time series analysis.

B. EXTREME LEARNING MACHINE

The ELM was proposed by Huang et al. [29]. It is a learning algorithm for single-layer feedforward neural networks (SLFNs) that offers several key concepts and advantages:

1) Random Input Weight Initialization: ELM initializes the weights connecting the input layer to the hidden layer randomly and does not adjust them during training, which differentiates it from traditional neural network learning algorithms that use iterative tuning [29].
2) Analytic Output Weights Determination: The weights between the hidden layer and the output layer are determined analytically by a simple generalized inverse operation (such as Moore-Penrose pseudoinverse), which is a form of linear system solution. This avoids the need for iterative fine-tuning, see Remark 5 in [29].
3) Fast Learning Speed: Because of its noniterative weight determination, ELM can train much faster than traditional neural networks that use backpropagation [29].
4) Generalization Performance: Despite the randomness in weight initialization, ELMs can achieve good generalization performance. They often require fewer hidden nodes than traditional feedforward networks to achieve comparable or even superior performance [29].
5) Ease of Use: ELMs are straightforward to implement and use, with fewer hyperparameters to tune compared to other learning methods, making them accessible to practitioners with limited machine learning expertise [29].
6) Universal Approximation Capability: ELMs have been proven to have universal approximation capability under certain conditions, meaning they can approximate any continuous function given a sufficient number of hidden nodes [30].
7) Online and Sequential Learning: ELMs can be adapted for online and sequential learning scenarios where data comes in a stream or batches, and the model needs to update its parameters on the fly [31].
8) Versatility: ELMs have been successfully applied to a wide range of tasks, including regression, classification, clustering, and feature learning [29], [32].

These concepts make ELM a powerful tool for various machine learning tasks, particularly where speed and ease of use are important considerations.

The ELM model consists of input weights $\mathbf{w}_j$, biases b_j, a number of hidden nodes $\#Hn$, an activation function $g(x)$, and output weights β', all of which operate on SLFNs. Let $\mathbf{X} = \{(\mathbf{x}_i, \mathbf{t}_i)\}_{i=1}^{N}$ represent a set of N samples of d dimensional vector. Each $(\mathbf{x}_i, \mathbf{t}_i)$ denotes the order pair of (the i-th input, the i-th target), where $\mathbf{x}_i = [x_{i1}, \ldots, x_{id}]^T \in \mathbb{R}^{N\times d}$ and $\mathbf{t}_i \in \mathbf{T} \subset \mathbb{R}^{N\times 1}$.

The output of ELM for the i-th input sample is given by

$$\mathbf{o}_i = \sum_{j=1}^{\#Hn} \beta'_j g\left(\mathbf{w}'_j \cdot \mathbf{x}_i + b_j\right), \tag{1}$$

where $\mathbf{w}'_j$ and b_j are a randomly connecting weight vector of the j-th hidden node, and b_j is its associated input bias. Both are randomly initialized and fixed. The vector $\beta' = \left[\beta'_1, \ldots, \beta'_{\#Hn}\right]^T$ represents the output weight.

For the training data of size N, we have N equations that can be written in the matrix form as

$$\mathbf{H}\beta' = \mathbf{T} \tag{2}$$

where $\mathbf{H}$ is the hidden output matrix, $N \times \#Hn$. Each of the i-th rows of $\mathbf{H}$ is written as $\mathbf{h}_i = \left[g\left(\mathbf{w}'_1 \cdot \mathbf{x}_i + b_1\right), \ldots, g\left(\mathbf{w}'_{\#Hn} \cdot \mathbf{x}_i + b_{\#Hn}\right)\right]$. For computing β', it is solved by

$$\beta' = \mathbf{H}^{\dagger}\mathbf{T}, \tag{3}$$

where $\mathbf{H}^{\dagger}$ is the Moore–Penrose generalized inverse of matrix $\mathbf{H}$ according to the smallest training error as

$$\min_{\beta} \|\mathbf{T} - \mathbf{H}\beta'\| \tag{4}$$

where $\#Hn \ll N$.

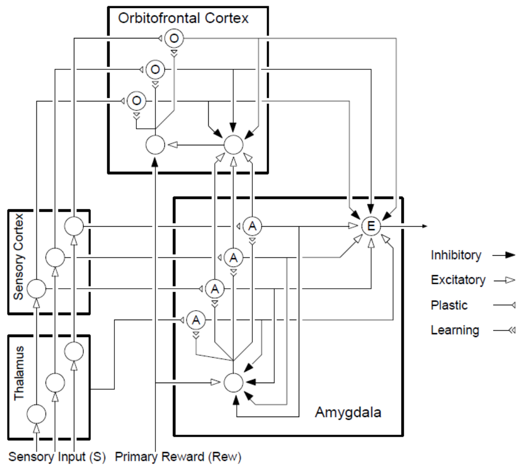


FIGURE 1. **The structure of the amygdala and orbitofrontal cortext model [13].**

C. BRAIN EMOTIONAL LEARNING

Morén et al. [13] proposed a computational model of emotional learning in the amygdala (MBEL). The model comprises two main parts: one representing the amygdala and the other representing the orbitofrontal cortex. The function of the amygdala is to respond to an input stimulus $\mathbf{S}_i$ that carries an emotional charge and neutral stimuli. At the same time, the role of the orbitofrontal cortex is to make decisions that will inhibit inappropriate responses from the amygdala.

From Fig. 1, for the stimulus $\mathbf{S}_i \in \mathbb{R}^d$, $A_j \in \mathbb{R}$ represents the input at the j-th dimension of amygdala, while $O_j \in \mathbb{R}$ represents the input at the j-th dimension of the orbitofrontal cortex, which are computed as

$$A_j = S_{ij}V_j, \tag{5}$$
$$O_j = S_{ij}W_j, \tag{6}$$

where $\mathbf{V} = [V_1, \ldots, V_d]$ and $\mathbf{W} = [W_1, \ldots, W_d]$ are the connecting weight vectors of the amygdala and orbitofrontal cortex, respectively. For a special input of the amygdala, it has an additional input or an expanded feature, which is computed as

$$A_{d+1} = \max(S_{i1}, \ldots, S_{id}). \tag{7}$$

For the MBEL, A_{d+1} is directly produced by the thalamus and fed to the amygdala. It is different from other parts of $\mathbf{S}_i$ that are obtained from the sensory cortex part. The approximated values of MBEL are written as

$$E = (\sum_{j=1}^{d} A_j + A_{d+1}) - \sum_{j=1}^{d} O_j, \tag{8}$$

where $(\sum_{j=1}^{d} A_j) + A_{d+1}$ is the output of the amygdala that reacts to the sensory input $\mathbf{S}_i$ while $\sum_{j=1}^{d} O_j$ inhibits inappropriate reactions for $\mathbf{S}_i$.

The updating of connecting weight V_j, iinvolves the difference between the reinforcer (Rew) and the activation of $\mathbf{A}$, which is computed as

$$\Delta V_j = \varepsilon(S_{ij} \times \max(0, Rew - \sum_{k=1}^{d} A_k)) \tag{9}$$

where ε is the learning rate of the model, Rew is a reward value and $\sum_{k=1}^{d} A_k$ is the summation of $\mathbf{A}$ at the previous period. The authors of this model have claimed that adjusting the weight rule of $\mathbf{V}$ is a monotonic increase [13].

Adjusting the j-th component of $\mathbf{W}$, is written as

$$\Delta W_j = \alpha(S_{ij} \times (\sum_{k=1}^{d} \mathbf{O}_k - Rew)), \tag{10}$$

where α is the learning rate.

D. BRAIN EMOTIONAL LEARNING BASED ON INTERVAL KNOWLEDGE (BEL-IK)

Sharafi et al. [17] have proposed an improved brain emotional learning model based on interval knowledge (BEL-IK) to enhance the MBEL model. It was reported that the performance of BEL-IK is better than that of MBEL. The strengths of BEL-IK lie in its efficiency, quick training and testing, compact size, simplicity, and ease of computation. BEL-IK, designed using the rough neural network concept, is based on interval knowledge (IK). For BEL-IK, connection weight vectors V_j of the amygdala are divided into two parts: V_j^{upper} and V_j^{lower}, while weight vectors W_j of the orbitofrontal cortex are modified to W_j^{upper} and W_j^{lower}.

Each orbitofrontal cortex node is divided into two parts for the structure of a BEL-IK neural network. The first and second are written as the following

$$O^{upper} = \sum_{j=1}^{d} S_j \times W_j^{uppper}, \tag{11}$$
$$O^{lower} = \sum_{j=1}^{d} S_j \times W_j^{lower}. \tag{12}$$

Each amygdala node is composed of two parts that are written as

$$A^{upper} = \sum_{j=1}^{d} S_j \times V_j^{upper}, \tag{13}$$
$$A^{lower} = \sum_{j=1}^{d} S_j \times V_j^{lower}. \tag{14}$$

The orbitofrontal cortex output is computed as follows

$$O^{ob} = \mu_1 O^{upper} + \mu_2 O^{lower}, \tag{15}$$

while the amygdala output is written as

$$O^{am} = \sigma_1 A^{upper} + \sigma_2 A^{lower}, \tag{16}$$

where μ_1 and μ_2 are weights of the orbitofrontal cortex output, σ_1 and σ_2 are weights of the amygdala output.

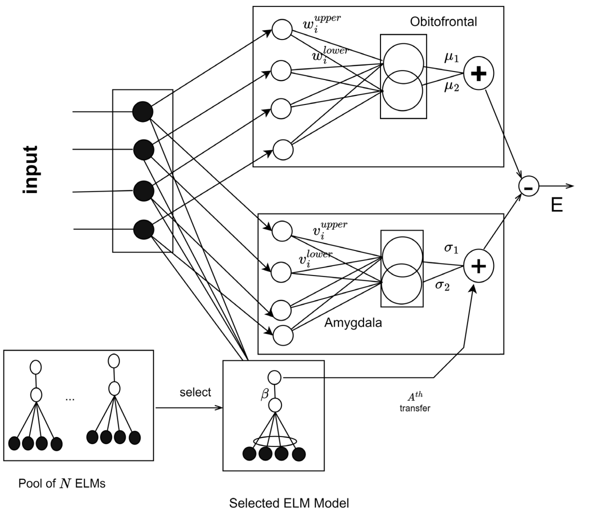


FIGURE 2. The structure of the Extreme Learning Machine guided Brain Emotional Learning model (ELM-BEL). There is no output bias term in the model.

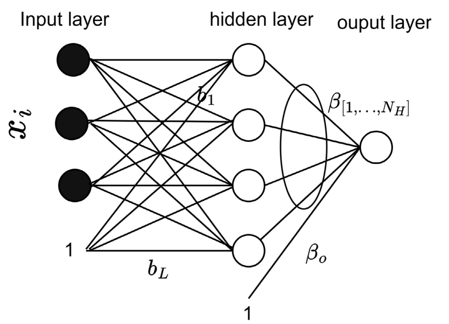


FIGURE 3. The Biased-ELM model features a single-hidden layer feedforward network with an output bias term sourced from the B2BELM-BEL's bias, affecting neuron activations and the network's decision-making process.

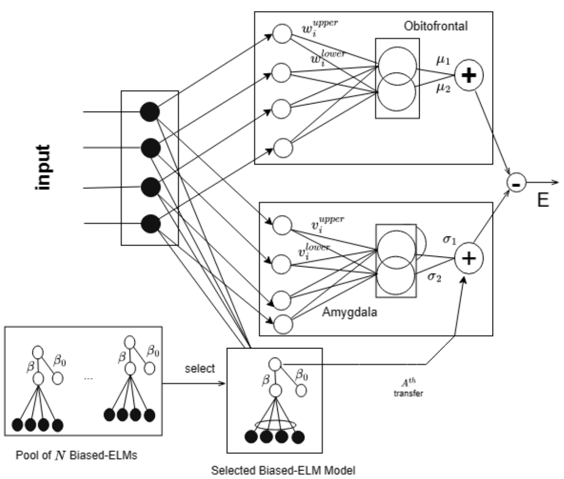


FIGURE 4. The proposed BBELM-BEL or B2ELM-BEL model includes a built-in bias term. This model builds upon the BEL and the Biased-ELM methods. The Biased-ELM is a single-hidden layer feedforward network, with the output bias coming directly from B2ELM-BEL's integrated bias.

The output of BEL-IK is computed as

$$E = (O^{am} + A_{d+1}) - O^{ob}. \tag{17}$$

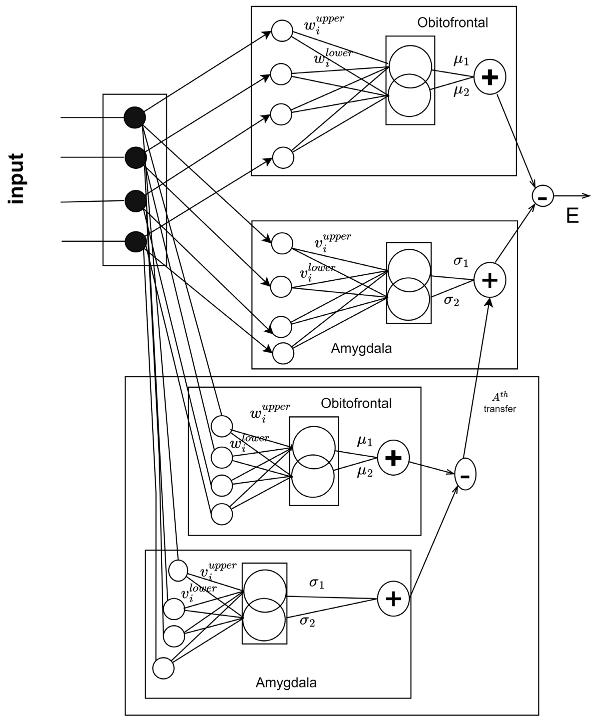


FIGURE 5. The structure of the BEL-BEL model.

Algorithm 1 Training of Biased Extreme Learning Machines

(1) call the TrainBiasedELM procedure to train and find a feed forward neural network giving the minimum error.
(2) call the FFELM function to get the ELM-guided expended feature.
procedure TrainBiasedELM($\mathbf{X}_S$,$\mathbf{Y}_S$,N_H,N_E)
1. create a set of ELM neural networks size N_E.
2. find the minimum error ELM neural network.
  for $i \leftarrow 1$ to N_E **do**
    $\mathbf{w}_i' \in \mathbb{R}^{d+1 \times N_H} \leftarrow 2 \times rand \in \mathbb{R}^{d+1\times N_H} - 0.5$
    $\mathbf{H} \leftarrow \left[\mathbf{h}_1, \ldots, \mathbf{h}_{N_S}\right]^T, \mathbf{H}_i \leftarrow \left[\mathbf{1}_{N_H\times 1}, \mathbf{H}\right]$
    $\beta_i \leftarrow \mathbf{H}^{\dagger}\mathbf{Y}_S, \hat{\mathbf{Y}}_i \leftarrow \mathbf{H}_i\beta_i$
    $Error_i$ gets from (30)
  end for
  find $BELM_{best}$ using (31)
  $\mathbf{W}_{BELM} \leftarrow \mathbf{W}'_{i_{best}}, \beta_{BELM} \leftarrow \beta_{i_{best}}$
return $\mathbf{W}_{BELM}, \beta_{BELM}$
end procedure

The updating of μ_1, μ_2, σ_1 and σ_2 at time t, they are computed as

$$\mu_1(t) = \mu_1(t-1) + \eta_1 \times e \times (-1) \times (1) \times (O^{upper}) \tag{18}$$
$$\mu_2(t) = \mu_2(t-1) + \eta_2 \times e \times (-1) \times (1) \times (O^{lower}) \tag{19}$$
$$\sigma_1(t) = \sigma_1(t-1) + \eta_3 \times e \times (-1) \times (1) \times (A^{upper}) \tag{20}$$
$$\sigma_2(t) = \sigma_2(t-1) + \eta_4 \times e \times (-1) \times (1) \times (A^{lower}), \tag{21}$$

where e is an approximating error, $e = \mathbf{t}_i - E$ when $\mathbf{t}_i \in \mathbf{T}$ is the $i-th$ target vector, η_1, η_2, η_3 and η_4 are learning rates.

Algorithm 2 The Bias-Boosted Extreme Learning Machine guided Brain Emotional Learning(B2ELM-BEL)

procedure B2ELMBEL($\mathbf{X}$,$\mathbf{Y}$,η_A,η_O,ω,Rew,$\mathbf{X}_S$,$\mathbf{Y}_S$,N_H,N_E)
  $\mathbf{a}_{ELM}$,β_{ELM} $\leftarrow$ TrainBiasedELM($\mathbf{X}_S$,$\mathbf{Y}_S$,N_H,N_E)
  $\mathbf{V}$, $\mathbf{W}$ $\leftarrow$ *rand* ▷ uniform random weights in [−1,1]
  $pMSE \leftarrow realMax$ ▷ *realMax* is a very large MSE.
  $loopEps \leftarrow 1e-8$ ▷ a small value for loop breaking.
  for $k \leftarrow 1$ to N_e **do** ▷ N_e is the number of epochs
    for $i \leftarrow 1$ to N_T **do** ▷ N_T is the training size
      $\hat{\mathbf{Y}}_i$ $\leftarrow$ B2ELMBEL_Test($\mathbf{X}_i$, $\mathbf{V}$, $\mathbf{W}$,$\mathbf{W}_{BELM}$,β_{BELM})
      compute E_i^{ob} and E_i^{am} from (33), (34), respectively.
      $e_i \leftarrow \left(\mathbf{Y}_i - \hat{\mathbf{Y}}_i\right)$
      for $j \leftarrow 1$ to size($\mathbf{V}$) **do**
        $\Delta\mathbf{V} \leftarrow e_i\epsilon_1 \left(\mathbf{X}_{ij}\max(0, Rew - \sum E_i^{am})\right)$
        $\Delta\mathbf{W} \leftarrow e_i\epsilon_2 \left(\mathbf{X}_{ij}(\sum \left(E_i^{ob}\right) - \omega Rew)\right)$
        $\mathbf{V}_j \leftarrow \mathbf{V}_j + \Delta\mathbf{V}$, $\mathbf{W}_j \leftarrow \mathbf{W}_j + \Delta\mathbf{W}$
      end for
    end for
    if $|pMSE - cMSE| \leq loopEps$ **then**
      $cloop \leftarrow cloop + 1$
      if $cloop \geq LoopLimit$ **then**
        break
      end if
    end if
    $pMSE \leftarrow cMSE$ ▷ cMSE is the current validated mean square error.
  end for
return $\mathbf{V}$, $\mathbf{W}$, $\mathbf{W}_{BELM}$, β_{BELM}
end procedure
function B2ELMBEL_Test($\mathbf{X}$, $\mathbf{V}$, $\mathbf{W}$,$\mathbf{W_{BELM}}$,β_{BELM})
  for $i \leftarrow 1$ to size($\mathbf{X}$) **do**
    $A_i^{th} \leftarrow$ FFELM($\mathbf{W}_{BELM}$,β_{BELM},$\mathbf{X}_i$)
    compute E_i^{ob} and E_i^{am} from (33), (34), respectively
    $E_i \leftarrow \left(E_i^{am} + A_i^{th}\right) - E_i^{ob}$), $\hat{\mathbf{Y}}_i \leftarrow E_i$
  end for
return $\hat{\mathbf{Y}}$
end function
function FFELM($\mathbf{W}_{BELM}$,β_{BELM},$\mathbf{X}$)
  create a $\mathbf{H}$ matrix according to (28)
  $\mathbf{H} \leftarrow \left[\mathbf{1}_{N_H\times 1}, \mathbf{H}\right]$, $\hat{\mathbf{Y}} \leftarrow \mathbf{H}\beta_{BELM}$
return $\hat{\mathbf{Y}}$
end function

V_j^{upper} and W_j^{upper} are updated according to (9) and (10), respectively.

III. PROPOSED METHODS

A. THE BIAS-BOOSTED EXTREME LEARNING MACHINE GUIDED BRAIN EMOTIONAL LEARNING (BBELM-BEL OR B2ELM-BEL)

In this section, we propose a new method for enhancing or structuring the Brain Emotional Learning processes based on functionalities derived from the Biased-ELM. The following pieces of literature have inspired this method:

- Khashman [12] delivered a thorough philosophical review encompassing emotions, their significance in AI, and previous emotion models. A modified variant of the backpropagation (BP) learning algorithm, referred to as the EmBP learning algorithm, was introduced for weight update in the generalized EmBP-based neural network topology. This algorithm focused on updating the weight of emotional neurons. Architecturally, it incorporates emotional neurons as an extra input dimension within every node, spanning both the hidden and output layers. Comparing the BEL-IK with the generalized EmBP-based neural network topology, the 'max' function of BEL-IK performs as the emotional neurons.

We view emotion as localized information, influenced by the present context rather than encompassing the

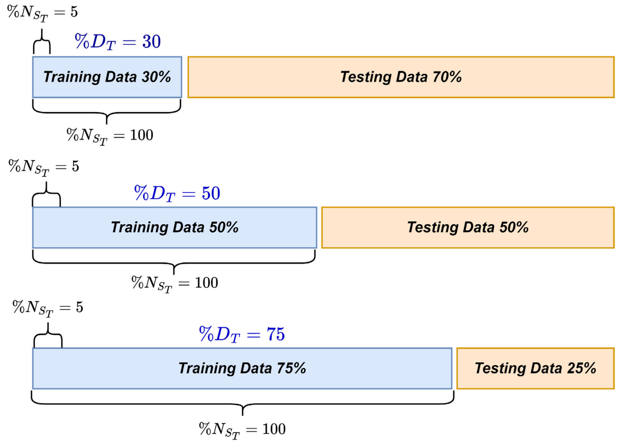


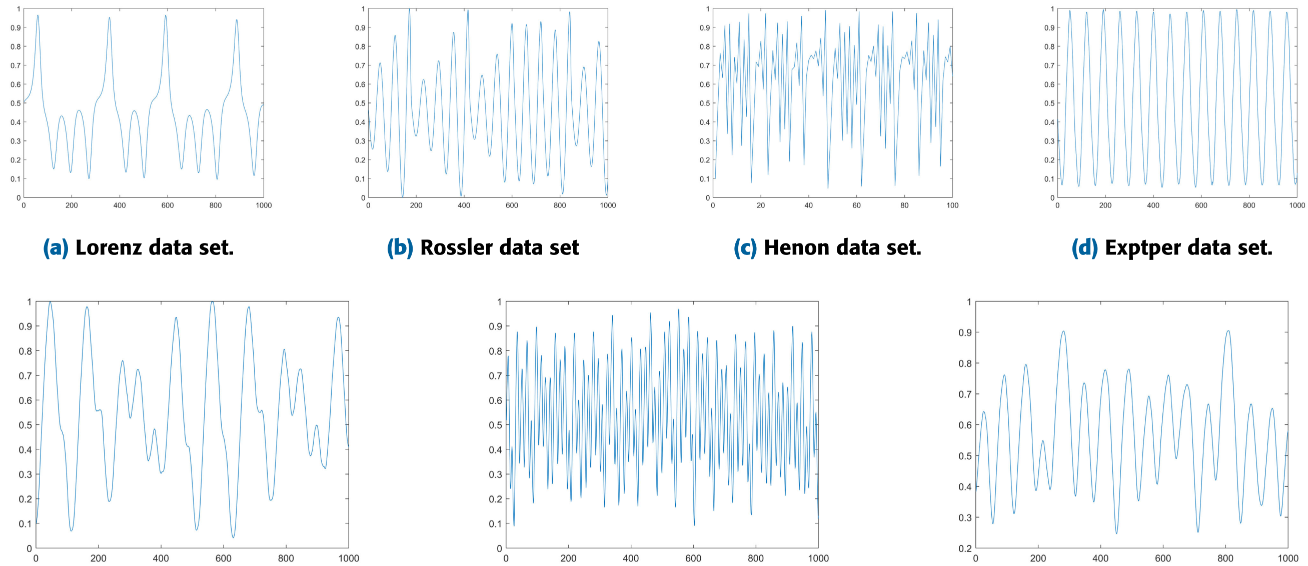
FIGURE 6. Data division in ELM-BEL and B2ELM-BEL: Illustrated with three training data proportions (30%, 50%, 75%). When 100% of the training data is used ($\%N_{S_T} = 100$), it exemplifies complete knowledge transfer learning.

TABLE 1. Percentage-based data division for chaotic time series: D_T and D_S ($\%N_{S_T}$) sizes. If $\%N_{S_T} = 100$, then $N_{S_T} = N_T$.

Problem	Total sample size	Number of Samples Determined by $\%D_T$ and $\%N_{S_T}$					
		$\%D_T$=30		$\%D_T$=50		$\%D_T$=75	
		$\%N_{S_T}$		$\%N_{S_T}$ t		$\%N_{S_T}$	
		5	100	5	100	5	100
EXPTCHAO, EXPTPER, EXPTQP2, EXPTQP3, HENON, LORENZ	16384	246	4915	410	8192	614	12288
ROSSLER	8192	123	2458	205	4096	307	6144

TABLE 2. Consolidated experimental settings.

Parameter	Value / Description
Learning Rate, ϵ_1 and ϵ_2, (All Methods)	0.0001
Rew and ω (All Methods)	2
Number of Epochs (All Methods)	200
Runs	30
Pool Size for A^{th} (All Methods)	50
Hidden Nodes (B2ELM-BEL, ELM-BEL)	{1, 5}
$\%D_T$ (BEL-IK, BEL-BEL, B2ELM-BEL, ELM-BEL)	{30, 50, 100}
$\%N_{S_T}$ (B2ELM-BEL, ELM-BEL)	{5, 100}
$\%N_{S_T}$ (BEL-BEL)	100
Activation Function (B2ELM-BEL, ELM-BEL)	Sigmoid
Evaluation Metrics	RMSE, MAE, SMAPE
Dept size for all experiments of chaotic time series benchmarking	2



(a) Lorenz data set. (b) Rossler data set (c) Henon data set. (d) Exptper data set.

(e) Exptqp2 data set. (f) Exptqp3 data set. (g) Exptchao data set.

FIGURE 7. Examples of seven chaotic time series benchmark data sets.

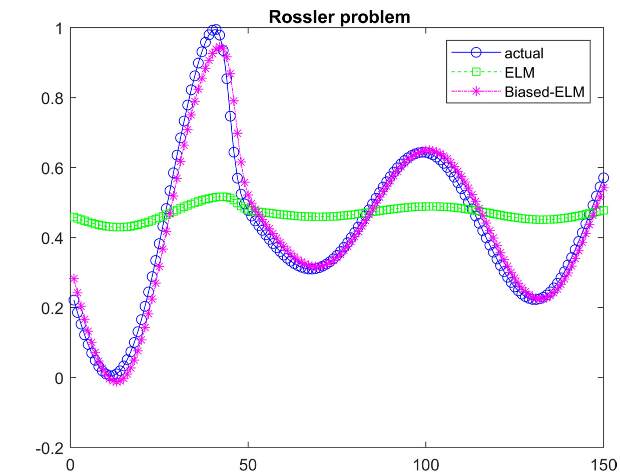


FIGURE 8. Comparison preliminary predicting of Rossler problem, between ELM and Biased-ELM, averaged from 30-runs and trained using ($\%D_T = 75$ and $\#Hn = 1$).

entire experience. Nonetheless, it's important to note that emotions can be modulated by one's experiences [33] and these experiences can also be transferred [34]. Consequently, we are in search of a straightforward machine learning model for facilitating knowledge transfer.

- Vincent and Bengio [35] showed that certain dictionary functions can help reduce errors. Specifically, if the output of these functions aligns closely with the current residue, the overall error goes down.
- Kwok and colleagues [36] found that tweaking the Projection Pursuit Learning (PPL) algorithm with a bias term better represents complex data. This change also helps the algorithm converge faster to solutions.

For approximation in the regression problem, PPL is formulated in the form of the projection pursuit regression (PPR) as

$$f_{PPR}(\mathbf{x}) = \sum_{k=1}^{\tilde{n}} g_k(\mathbf{a_k} \cdot \mathbf{x}), \tag{22}$$

where $\mathbf{x} \in \mathcal{R}^d$ is the input vector having the dimension d, $\mathbf{a}_k$ is the projection vector with $\|\mathbf{a_k}\| = 1$ and g_k is a transfer function or $\{g_1, \ldots, g_{\tilde{R}}\}$ are called smoothers, from a statistics perspective, $\tilde{R}$ called the order. The behavior of the smoothers in PPL are parametric functions that are written in linear combinations of Hermite functions as

$$g(\tilde{z}) = \sum_{r=1}^{\tilde{R}} \rho_r h_r(\tilde{z}), \tag{23}$$

where $\tilde{z} = \mathbf{a} \cdot \mathbf{x}$, $h_r(\tilde{z})$ is a Hermite function having orthogonormal property and defined by

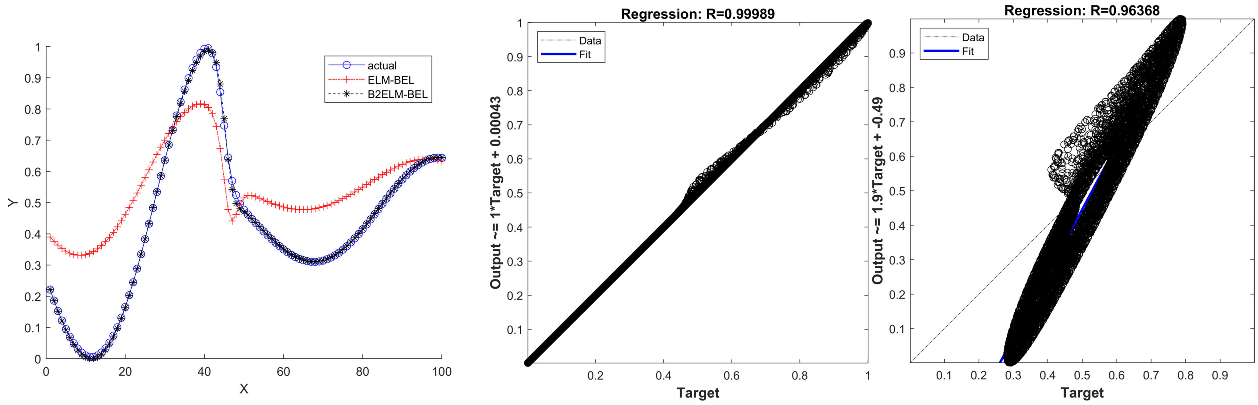
$$h_r(\tilde{z}) = (r!)^{-\frac{1}{2}} \pi^{\frac{1}{4}} 2^{-\frac{(r-1)}{2}} \tilde{H}_r(\tilde{z})\phi(\tilde{z}), \tag{24}$$

where each $\tilde{H}_r(\tilde{z})$ is a Hermite polynomial. It is created in the recursive style as

$$\begin{aligned}
&\tilde{H}_0(\tilde{z}) = 1 \\
&\tilde{H}_1(\tilde{z}) = 2\tilde{z} \\
&\tilde{H}_r(\tilde{z}) = 2[\tilde{z}\tilde{H}_{r-1}(\tilde{z}) - (r-1)\tilde{H}_{r-2}(\tilde{z})] \\
&\quad r = 2, 3, 4, \ldots
\end{aligned}$$

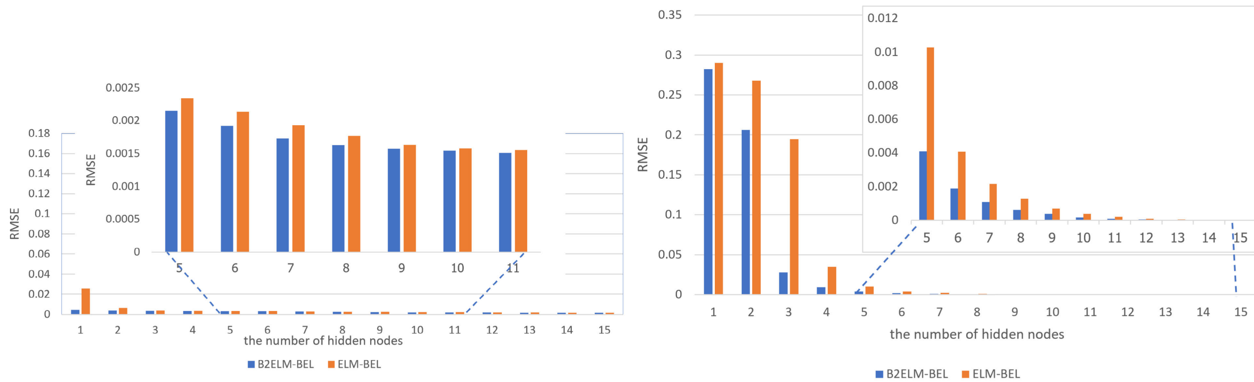
and $\phi(\tilde{z})$ is the weighting function

$$\phi(\tilde{z}) = \frac{1}{\sqrt{2\pi}} e^{-\frac{\tilde{z}^2}{2}}.$$



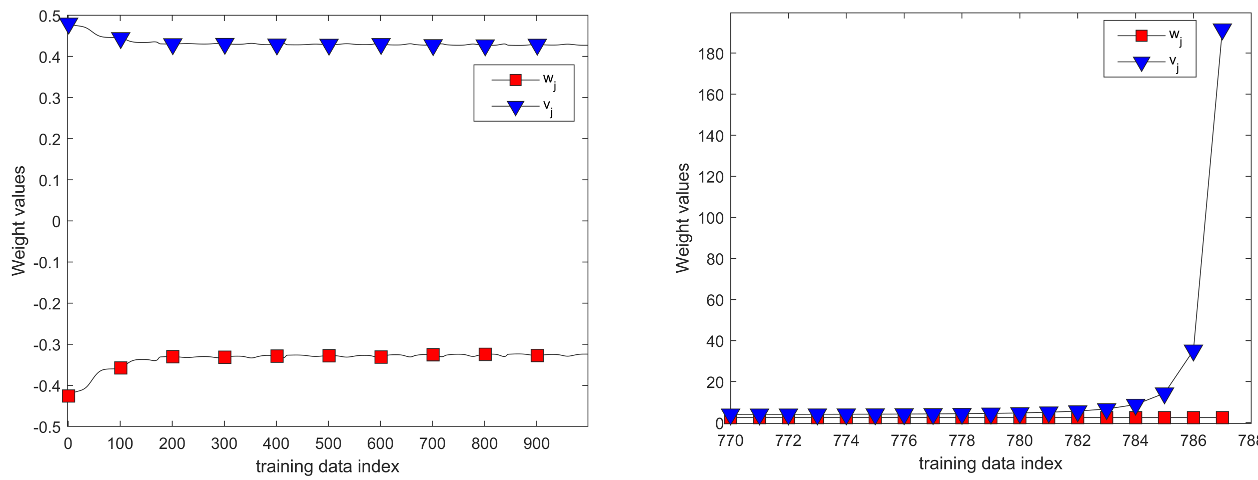
(a) Rossler: Comparison of model prediction. **(b)** Rossler: B2ELM-BEL prediction vs. Actual value **(c)** Rossler: ELM-BEL prediction vs. Actual value.

FIGURE 9. Example of prediction comparisons using scatter plots with the regression lines of B2ELM-BEL and ELM-BEL for the Rossier problem. Note that they were trained using ($\%D_T = 75$, $\%N_{S_T} = 5$ and $\#Hn = 1$).



(a) Rossler data set. **(b)** Henon data set.

FIGURE 10. Comparison testing estimation of B2ELM-BEL and ELM-BEL methods when the number of hidden nodes was varied in the range [1-15] by showing testing averaged RMSE from 30 runs, ($\%D_T = 75$, $\%N_{S_T} = 5$).

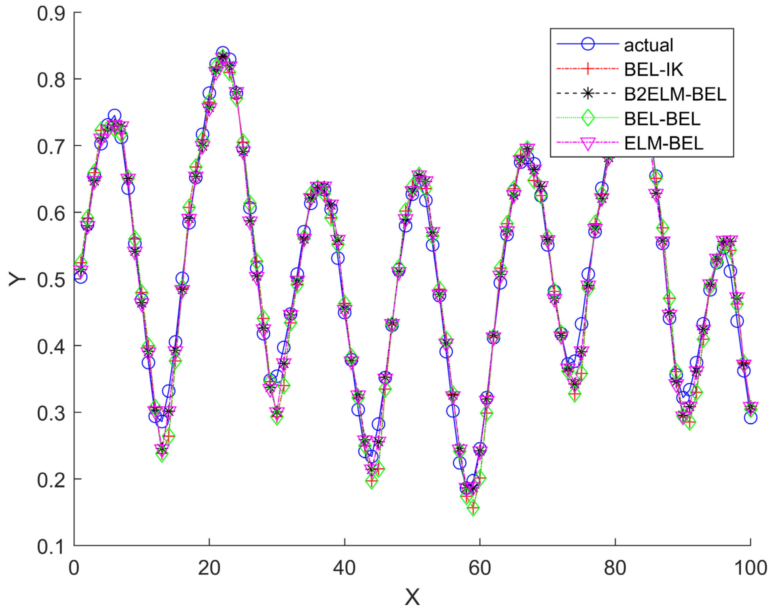


(a) Proper Weights. **(b)** Improper Weights.

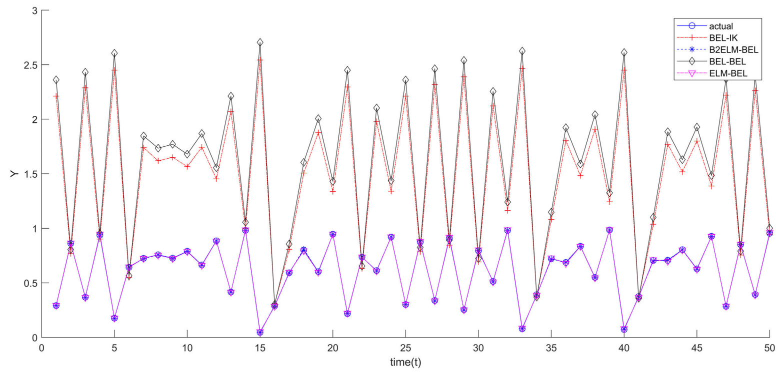
FIGURE 11. Distinguishing the variances in weight: Proper versus Improper BEL models.

To formulate PPL by PPR in a feedforward neural networking structure having a hidden layer and output layer, the output of the neural network with the n hidden nodes is written as

$$f_n(\mathbf{x}) = \sum_{k=1}^{\tilde{n}} \tilde{\beta}_k g_k(\mathbf{a_k} \cdot \mathbf{x}). \tag{25}$$



(a) Exptqp3: Comparison of model prediction.



(b) Henon: Comparison of model prediction.

FIGURE 12. **Comparative one-step predictions by various methods for the Exptqp3 and Henon datasets, illustrating the performance of ELM-BEL and B2ELM-BEL with 5 hidden nodes trained using (%D_T = 30 and %N_{S_T} = 5).**

However, Kwok and colleagues claimed that PPL has a major problem involving convergence because a neural network, as in (25) with the fixed order $\tilde{R}$ will degrade both training and testing. In addition, they reported that both $\tilde{R}$ and the number $\tilde{n}$ of hidden nodes affected the generalization performance. To address the problem, in the case of fixed $\tilde{R}$, and still be able to be a universal approximation, they suggested adding a bias term to the parameter of each transfer function. Therefore, the equation (25) is modified to be

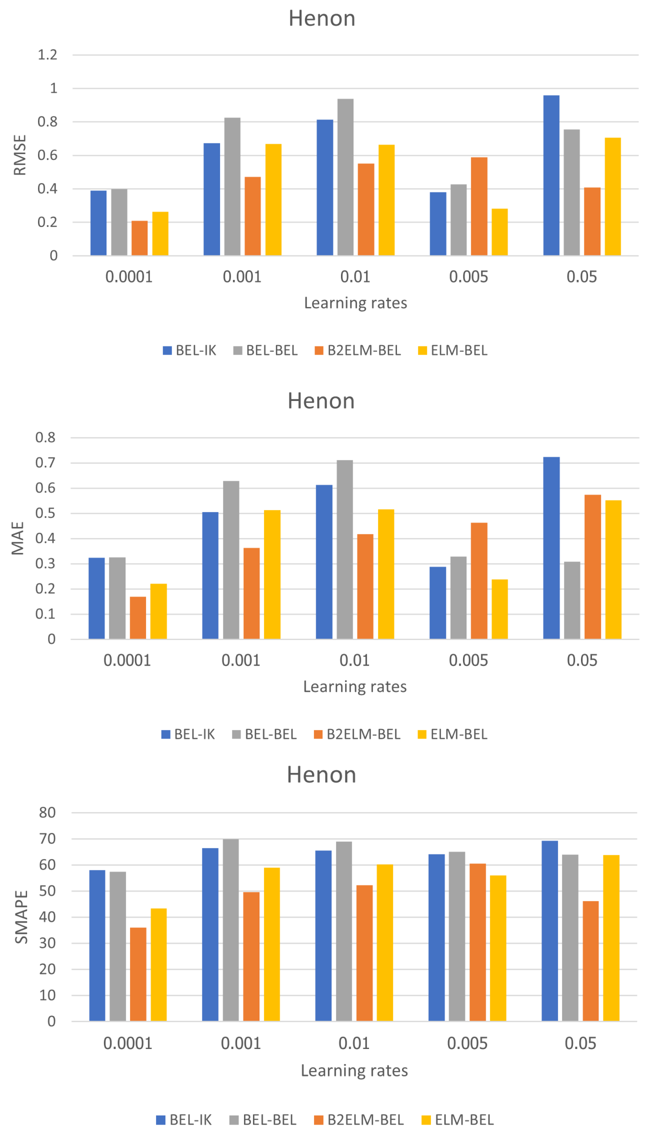
$$f_n(\mathbf{x}) = \sum_{k=1}^{\tilde{n}} \tilde{\beta}_k g_k(\mathbf{a_k} \cdot \mathbf{x} + \tilde{\theta}_k), \tag{26}$$

where $\tilde{\beta}_k$ are the output weights, $\tilde{n}$ is the number of hidden units and $\tilde{\theta}_k$ is a bias value of the k-th hidden node. PPL can be constructed by adding one hidden node into the network as

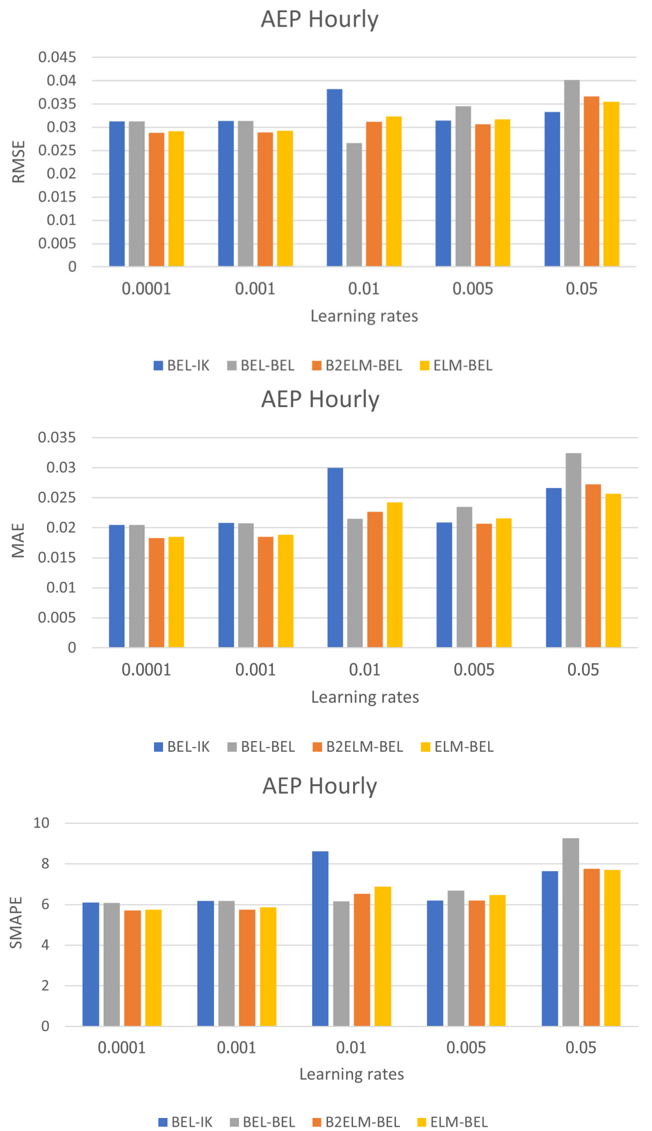
$$f_n(\mathbf{x}) = \sum_{k=1}^{\tilde{n}} \tilde{\beta}_k g_k(\mathbf{a_k} \cdot \mathbf{x} + \tilde{\theta}_k) + \tilde{\beta}_{\tilde{n}+1} g_{\tilde{n}+1}(\mathbf{a}_{\tilde{n}+1} \cdot \mathbf{x} + \tilde{\theta}_{n+1}). \tag{27}$$

Based on our findings, incorporating ELM principles is expected to enhance the performance of the BEL model in predicting time-series data. To this end, we have created a novel model that replaces the 'max' function in the existing BEL-IK framework with ELM. This integration is executed without incorporating a bias term. This results in the ELM-BEL model, as shown in Fig. 2. Moreover, introducing an output bias term into our ELM-BEL model could further boost its predictive accuracy.

To identify the optimal bias value, we integrated this term into the output layer of the ELM, utilizing the



(a) Henon dataset.



(b) AEP hourly dataset.

FIGURE 13. Comparative analysis of preliminary test predictions for the Henon and AEP hourly problems, focusing on the variation in learning rates. This comparison involves the BEL-IK, BEL-BEL, B2ELM-BEL, and ELM-BEL methods over 10 runs. Each method is denoted with a shorthand that includes the respective learning rate values: $\%D_T = 50$, $\%N_{S_T} = 100$, and $\#Hn = 1$ (specifically in the case of B2ELM-BEL and ELM-BEL).

pseudo-inversion method as detailed in equation (3). Notably, the original ELM, as described in [29], does not include this bias term. Consequently, we have developed an enhanced version, termed the Biased-ELM. The architecture of the Biased-ELM is depicted in Fig. 3 where β_0 is the added bias term.

Additionally, we have devised a further advanced model named the Bias-Boosted Extreme Learning Machine guided Brain Emotional Learning (BBELM-BEL or B2ELM-BEL), with its structure illustrated in Fig. 4. The design of the B2ELM-BEL structure draws inspiration from the work proposed in [12].

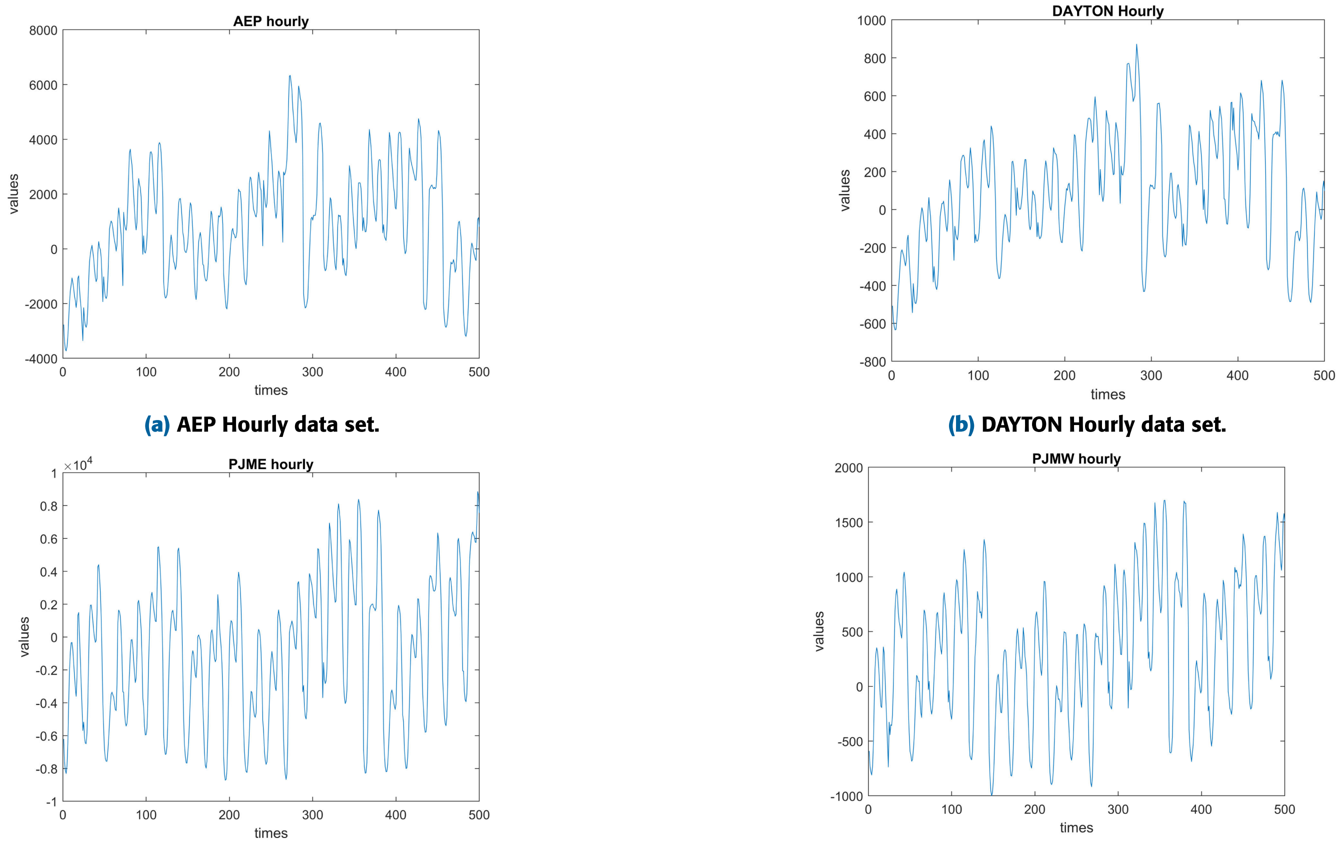
We anticipate that the inclusion of the output bias term will enable the B2ELM-BEL model to outperform other models in predicting time-series data, particularly when the number of hidden nodes in the ELM is limited.

IV. TRAINING PROCESS OF B2ELM-BEL

A. PHASE 1: DEVELOPMENT OF BIASED-ELM MODELS

In the first phase, we developed a set of N_{ELM} Biased-ELM models by training with the dataset $\mathcal{D}_S$. The primary goal at this stage was to select a Biased-ELM model that effectively determined the right thalamus stimulus for directing the amygdala's response.

For training the Biased-ELM algorithm with N_H hidden nodes, we considered the dataset $\mathcal{D}_S = \left\{\left(\mathbf{x}_i^S, \mathbf{y}_i^S\right)\right\}$ and the randomized weight vector $\mathbf{w}_j' = \left[b_j, w_{j1}', \ldots, w_{jN_H}'\right]^T$



(a) AEP Hourly data set.

(b) DAYTON Hourly data set.

(c) PJM East Region (PJME) Hourly data set.

(d) PJM West Region (PJMW) Hourly data set.

FIGURE 14. Example of data in the context of real-world energy consumption data, $\%D_T = 75$.

for each hidden node $j = 1, \ldots, N_H$. The feature set $\mathbf{X}_s = \{\mathbf{x}_1^s, \ldots, \mathbf{x}_{N_S}^S\}$ and the corresponding target $\mathbf{Y}_S = \{\mathbf{y}_1^s, \ldots, \mathbf{y}_{N_S}^s\}$ were used for a transfer learning perspective. The hidden output matrix $\mathbf{H}$ is computed as follows:

$$\mathbf{h}_i = \left[1, g(w'_{j1} \cdot [\mathbf{1}, x_i^s]_i^s), \cdots, g(w'_{jN_H} \cdot [\mathbf{1}, x_i^s]_i^s)\right]_{1\times(N_H+1)}, \tag{28}$$

where $g(\cdot)$ is an activation function, such as the sigmoid function.

The predicted target $\hat{\mathbf{Y}}_i$ of the i-th input in the Biased-ELM model is obtained as follows:

$$\hat{\mathbf{Y}}_i = \beta_0 + \mathbf{H}\beta_{[1,\ldots,N_H]}, \tag{29}$$

where $\min_{\beta_{[1,\ldots,N_H]},\beta_0} \|\mathbf{Y}_i - (\beta_0 + \mathbf{H}\beta_{[1,\ldots,N_H]})\|$ which $\beta_{[1,\ldots,N_H]}$, β_0 are estimated using the least squares approach. This approach involves computing the pseudo-inverse matrix, which utilizes experience-transformed information via $\mathbf{H}$ and is supervised by the actual target $\mathbf{Y}_i$ of the i-th Biased-ELM model, thereby guiding their optimized values. If every Biased-ELM model utilizes the same training data, then $\mathbf{Y}_i$ equals$\mathbf{Y}_S$. The output weights β, including bias terms, are computed similarly to β' in the ELM model but with different dimensions. The prediction error for each model is evaluated using:

$$Error_j = \sum_i^{N_S} (\mathbf{Y}_i - \hat{\mathbf{Y}}_i)^2, \tag{30}$$

and the model with the minimum error $BELM_{best}$ is selected from the following condition:

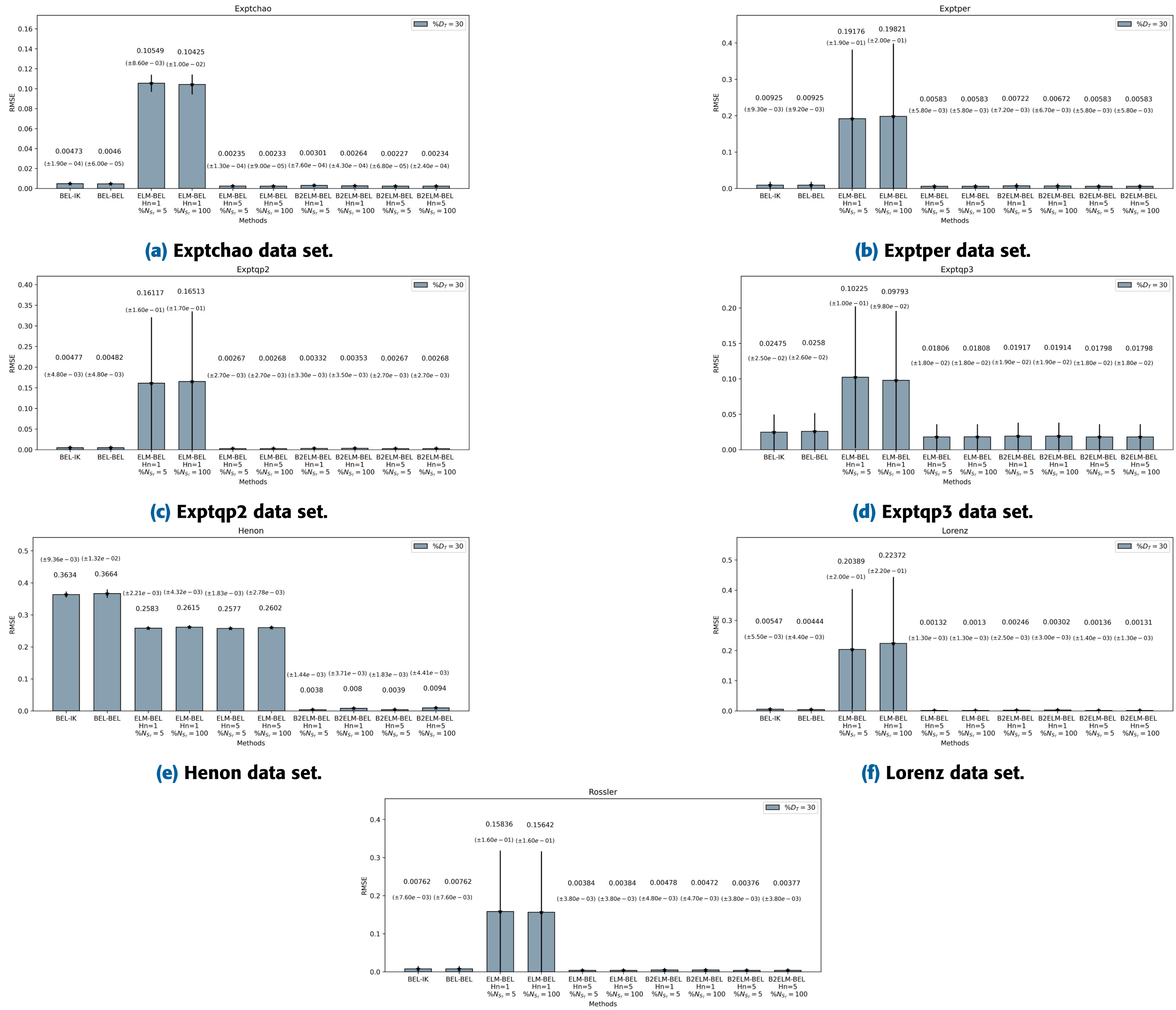
$$\min(Error_1, \ldots, Error_{N_E}), \tag{31}$$

where $j = 1, \ldots, N_e$, N_e is the pool size or the number of ELM-based models.

B. PHASE 2: TRAINING OF THE BRAIN EMOTIONAL LEARNING MODEL

In the second phase, the BEL model is trained using the dataset $\mathcal{D}_T$, influenced by the thalamus stimulus selected in the first phase. This training is guided by the chosen Biased-ELM model, ensuring accurate tuning to the specific thalamus stimulus, leading to effective learning and response patterns.

The selected Biased-ELM, as outlined in Algorithm 1, yields two essential elements: $\mathbf{W}_{BELM}$ and β_{BELM}. These are used to construct the *FFELM* function. *FFELM* plays a key role in transferring knowledge, transforming $\mathbf{x}_i$ into A^{th},



(a) Exptchao data set. **(b)** Exptper data set. **(c)** Exptqp2 data set. **(d)** Exptqp3 data set. **(e)** Henon data set. **(f)** Lorenz data set. **(g)** Rossler data set.

FIGURE 15. Comparison of testing RMSE and SD for time series benchmark data. $\%D_T = 30$.

which is expressed as:

$$A_i^{th} = FFELM\left(\mathbf{W}_{BELM}, \beta_{BELM}, \mathbf{x}_i\right). \quad (32)$$

A_i^{th} serves as a guideline or knowledge transfer from experience, influencing the amygdala's reaction automatically. Additionally, $FFELM$ is pivotal in predicting $\hat{\mathbf{y}}$ for a given sample $\mathbf{x_i}$.

The training and predicting with the BEL Model are detailed in Algorithm 2, this phase focuses on using $\mathcal{D}_T$, a set of training samples for the target domain. The output of the orbitofrontal cortex and the amygdala are computed as:

$$E_i^{ob} = \sum_{j=1}^{d} x_{ij} \times w_j, \quad (33)$$

$$E_i^{am} = \sum_{j=1}^{d} x_{ij} \times v_j. \quad (34)$$

Here, d represents the dimension number of $\mathbf{x}_i$, with weight vectors $\mathbf{w} = [w_1, \ldots, w_d]^T$ and $\mathbf{v} = [v_1, \ldots, v_d]^T$ for the orbitofrontal cortex and amygdala, respectively. The output for sample $\mathbf{x}_i$ in B2ELM-BEL is:

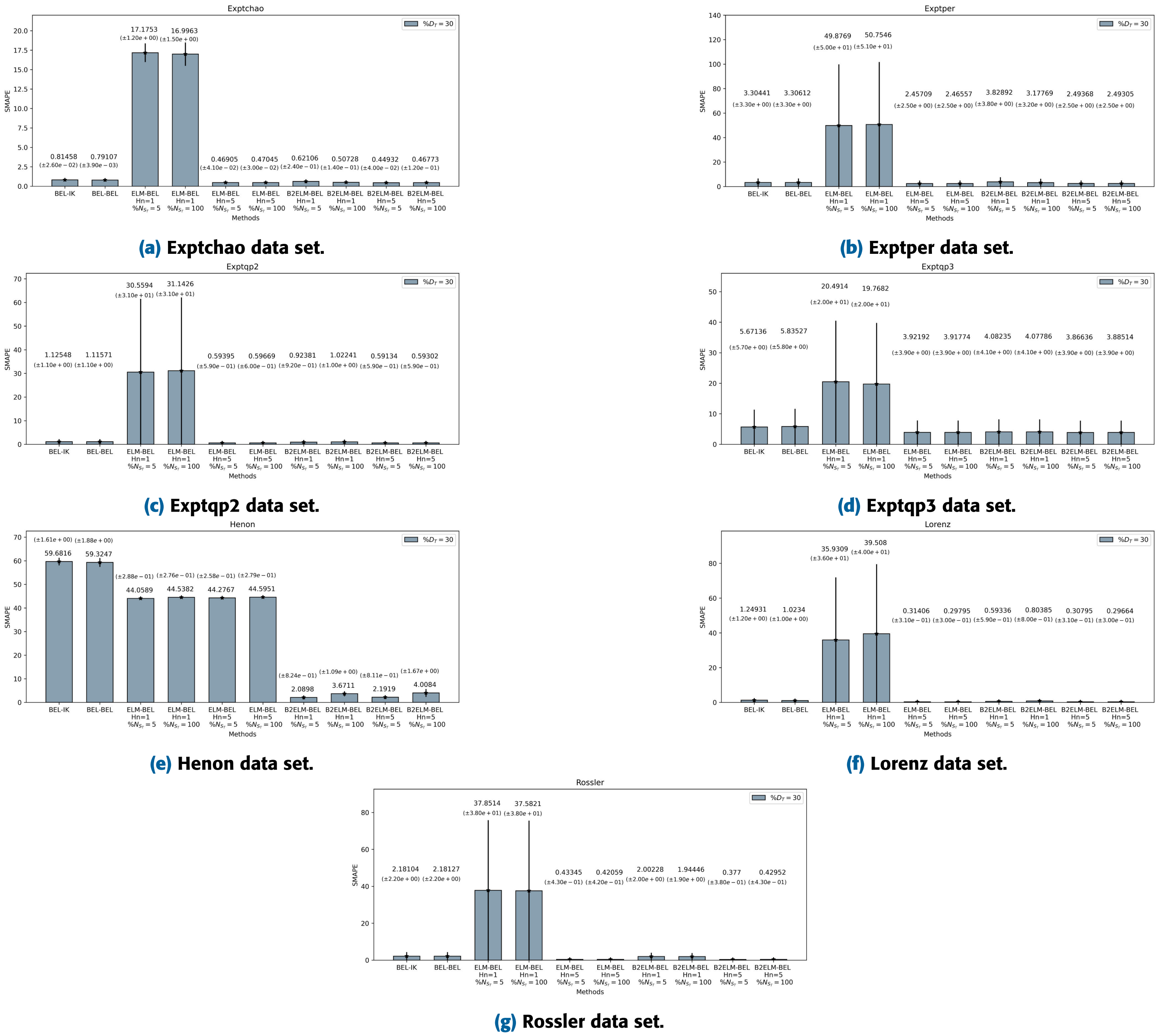
$$E_i = (E_i^{am} + A_i^{th}) - E_i^{ob}. \quad (35)$$

Here, A_i^{th} is produced using (32).

The updates of $\mathbf{w}$ and $\mathbf{v}$ follow BEL methodologies by Morén [13] and BEL-IK. The updating formulas are:

$$\Delta \mathbf{v}_j = error_i \times \epsilon_1 \times (x_{ij} \times \max(0, Rew - E_i^{am})) \quad (36)$$

$$\mathbf{v}_j(t) = \mathbf{v}_j(t-1) + \Delta \mathbf{v}_j \quad (37)$$



(a) Exptchao data set. **(b) Exptper data set.** **(c) Exptqp2 data set.** **(d) Exptqp3 data set.** **(e) Henon data set.** **(f) Lorenz data set.** **(g) Rossler data set.**

FIGURE 16. Comparison of testing SMAPE and SD for time series benchmark data. $\%D_T = 30$.

$$\Delta \mathbf{w}_j = error_i \times \epsilon_2 \times (x_{ij} \times (E_i^{ob} - \omega \times Rew)) \tag{38}$$

$$\mathbf{w}_j(t) = \mathbf{w}_j(t-1) + \Delta \mathbf{w}_j, \tag{39}$$

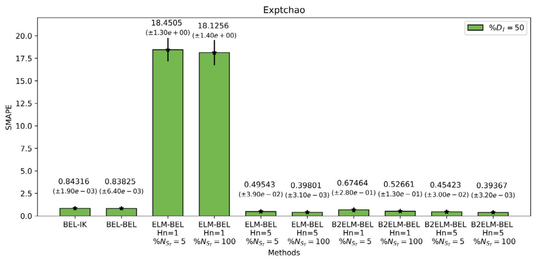
where $j = \{1, \ldots, d\}$ and $error_i = \mathbf{y}_i - \hat{\mathbf{y}}_i$ is the prediction error for time series input $\mathbf{x}_i$, in the case of if $d = 2$ then $j = 1$ is *lower* and $j = 2$ is *upper*. In these formulas, *Rew* denotes the reinforcer value, with ϵ_1 and ϵ_2 as learning rates for $\mathbf{v}$ and $\mathbf{w}$, respectively and ω adjusting the value of *Rew* in the orbitofrontal cortex.

C. THE BRAIN EMOTIONAL LEARNING GUIDED BRAIN EMOTIONAL LEARNING (BEL-BEL)

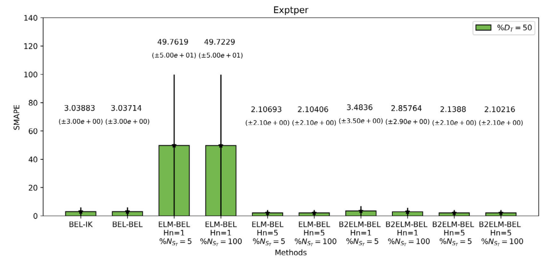
Shifting our focus from the ELM to explore the potential of BEL in knowledge transfer, we investigated the capability of BEL to function independently in this capacity. This exploration led to the development of a model we've termed the Brain Emotional Learning guided Brain Emotional Learning (BEL-BEL). This model echoes the structural complexity of the B2ELM-BEL, employing a two-stage training process.

In the initial stage, we utilized a dataset $\mathcal{D}_S$ to train a BEL-IK model. This model is designed to approximate the stimuli in the thalamus. Following this, the second stage involves training another BEL-IK model using a different dataset $\mathcal{D}_T$. This stage is distinct,in that it operates under the guidance of the initial BEL-IK model, fostering a deeper and more nuanced knowledge transfer.

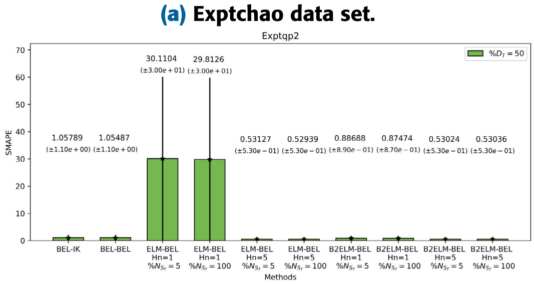
Fig. 5 illustrates the BEL-BEL model's design and main features, providing an visual guide to its structure and how it supports knowledge transfer.



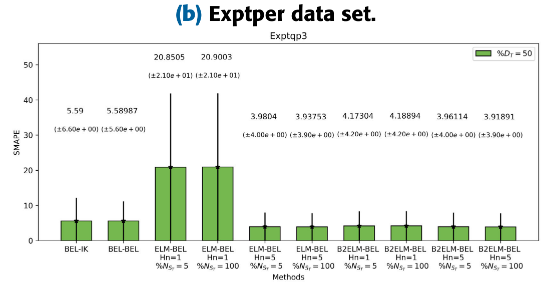
(a) Exptchao data set.



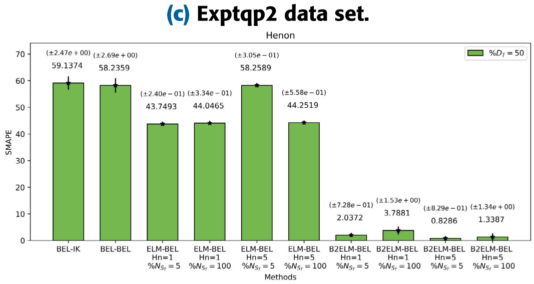
(b) Exptper data set.



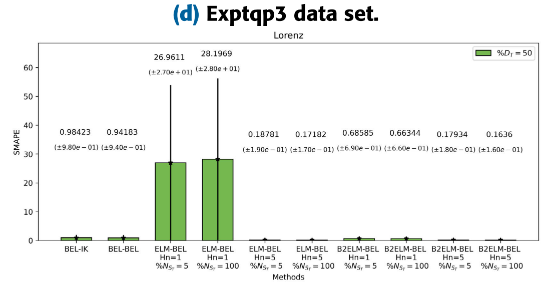
(c) Exptqp2 data set.



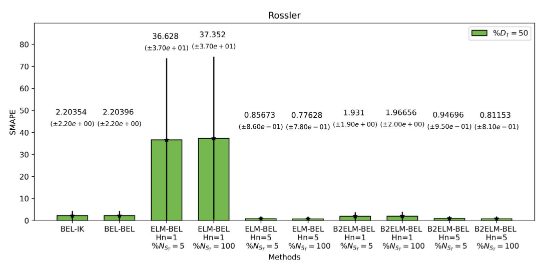
(d) Exptqp3 data set.



(e) Henon data set.



(f) Lorenz data set.



(g) Rossler data set.

FIGURE 17. Comparison of testing SMAPE and SD for time series benchmark data. %D_T = 50.

D. COMPLEXITY ANALYSIS

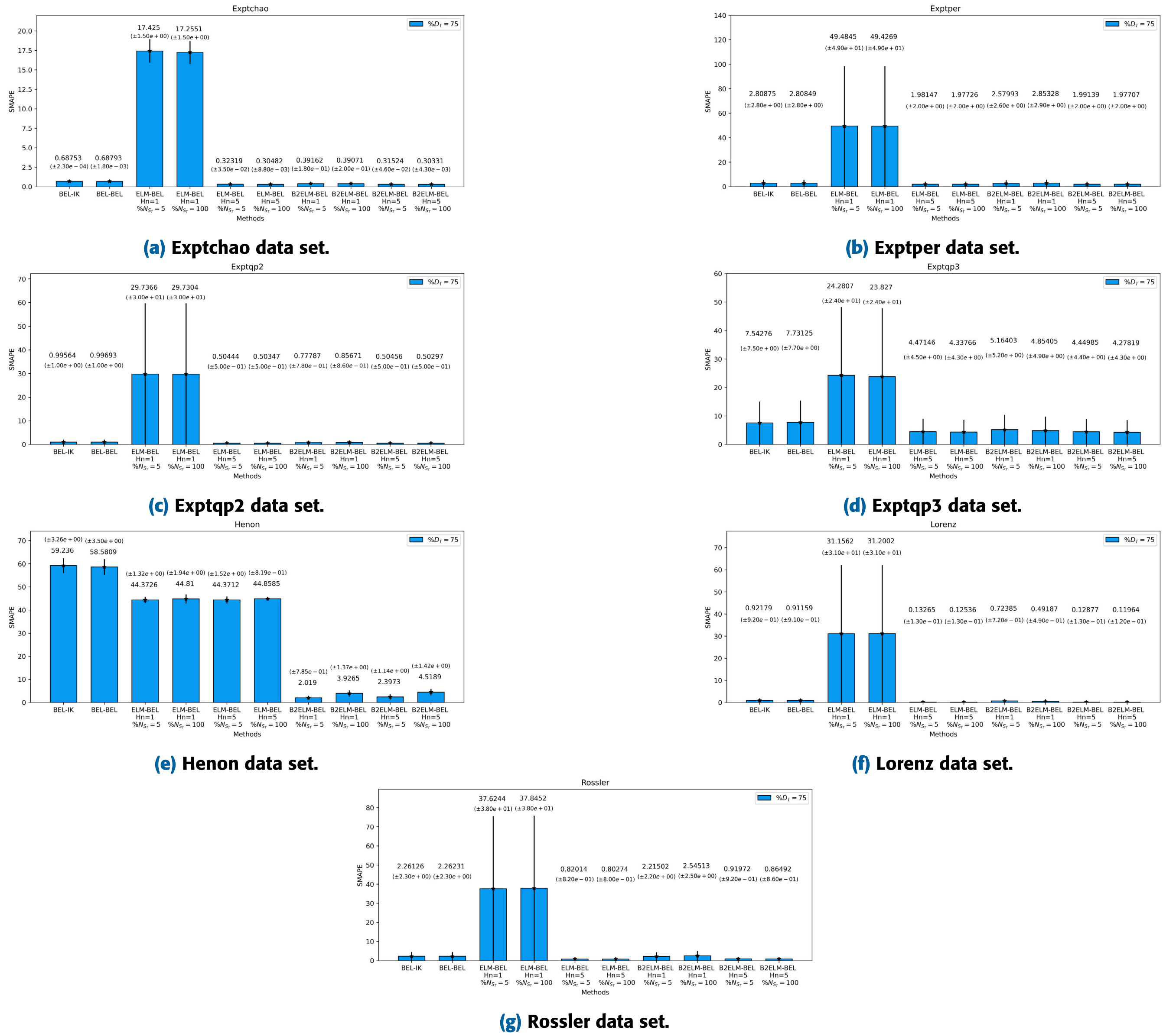
The complexity analysis of B2ELM-BEL based on the floating point operations per second (FLOPS), from [37] claimed that the number of FLOPS required to solve the least squares equation $\mathbf{Ax} = \mathbf{b}$ is $mn^2 + 2n^3$ where $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{b} \in \mathbb{R}^m$. Therefore, the complexity of Algorithm 1 is approximated as $O\left(N_E\left(N_S\left(N_H+1\right)^2+2\left(N_H+1\right)^3\right)\right)$ FLOPS where N_E is the number of Biased-ELM models to be trained, N_S is the training source sample size, which is very small when compared with the training sample size of BEL, $N_s \ll N_T$, N_H is the number of hidden nodes. For the complexity of Algorithm 2, the number of FLOPS in each part is approximated as

- for computing the B2ELMBEL_Test function to get A_i^{th}, the number of flops is $N_S\left(N_H+1\right)cd + 2d$ where c is the number of dimension of Y_i, $\mathbf{Y} \in \mathbb{R}^{N_T \times c}$, d is the number of features in $\mathbf{x}_i \in \mathbb{R}^d$.
- for computing E_i^{ob} and E_i^{am}, the number of FLOPS is $4d$.
- for computing e_i, the number of FLOPS is c
- for computing $\mathbf{W}_j$ and $\mathbf{V}_j$, the number of FLOPS is $10d$

The FLOPS total in Algorithm 2 is approximated as

$$O\left(N_e N_T\left(t_{FFELM} + 14d + c\right) + t_{BELM}\right)$$

where $t_{FFELM} = N_S\left(N_H+1\right)cd + 2d$, $t_{BELM} = N_E\left(N_S\left(N_H+1\right)^2 + 2\left(N_H+1\right)^3\right)$, N_e is the number of epochs. In the case of ELM without a biased term in Algorithm 1 and 2, the modified version of B2ELM-BEL is called ELM-BEL. The number of FLOPS of ELM-BEL is



(a) Exptchao data set. (b) Exptper data set. (c) Exptqp2 data set. (d) Exptqp3 data set. (e) Henon data set. (f) Lorenz data set. (g) Rossler data set.

FIGURE 18. Comparison of testing SMAPE and SD for time series benchmark data. %D_T = 75.

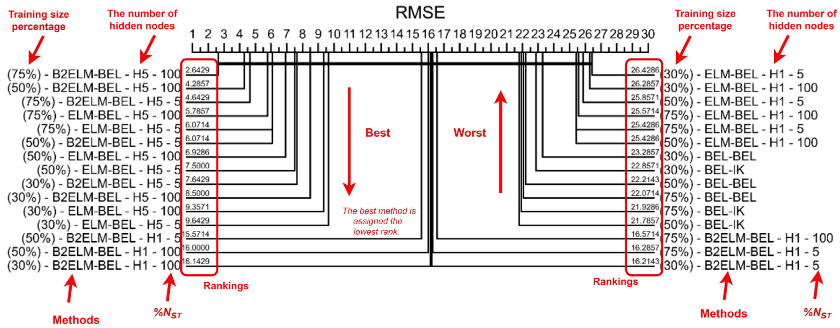
approximated as

$$O\left(N_e N_T\left(t_{FELM}+14d+c\right)+t_{ELM}\right)$$

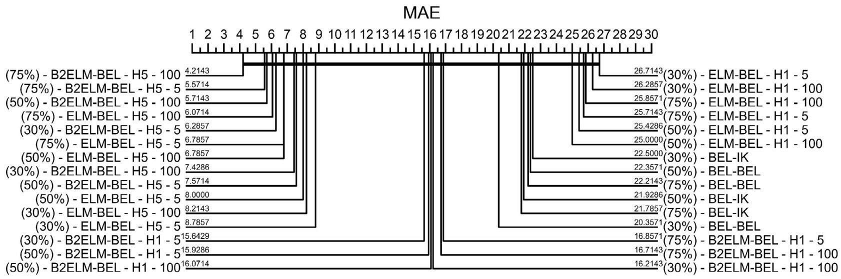
where $t_{FELM} = N_S Lcd + 2d$, $t_{ELM} = N_E\left(N_S N_H^2 + 2N_H^3\right)$. The number of FLOPS for BEL-IK training is approximated as $O\left(N_e N_T\left(14d+2c\right)\right)$ which is faster than of B2ELM-BEL. If N_T is very large, then t_{BELM}/N_T might be close to zero, therefore the proportion of FLOPS between BEL-IK and B2ELM-BEL can be transformed from $\frac{N_e N_T(14d+2c)}{N_e N_T(N_s(N_H+1)cd+2d+14d+c)+N_E\left(N_S(N_H+1)^2+2(N_H+1)^3\right)}$ to be $\frac{14d+2c}{N_s(N_H+1)cd+16d+c}$. If we set values to c and d as $c = 1$, $d = 2$ then the proportion is modified as $\frac{15}{N_s(N_H+1)+17}$. That mean training time scale of B2ELM-BEL is greater than of BEL-IK, which depends on the number of the source size N_s and the number of hidden nodes. However, usually N_s and N_H sizes usually tend to be small values.

V. ADVANCED BEL MODELING TECHNIQUES: EXPERIMENTAL DESIGN AND TIME SERIES FORECASTING

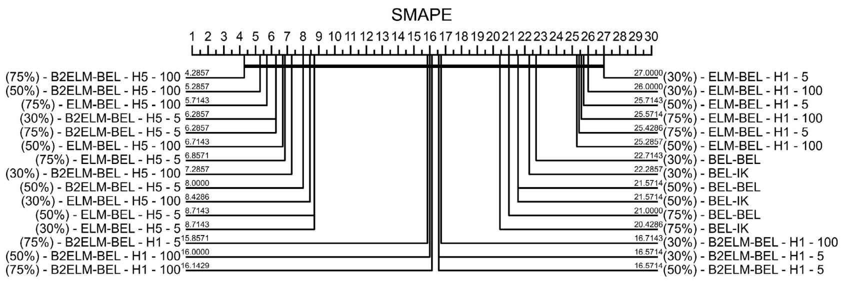
This section explains the experimental design of B2ELM-BEL and ELM-BEL and assesses their performance compared to BEL-IK and BEL-BEL. All experiments presented in this paper were conducted using MATLAB 2022a on a PC equipped with 8GB of RAM and running the Windows 10 operating system. The data sets analyzed in this paper are divided into two categories: seven chaotic time series benchmark data sets and four real-world data sets.



(a) critical difference diagram based on RMSE.



(b) critical difference diagram based on MAE.



(c) critical difference diagram based on SMAPE.

FIGURE 19. Comparative ranking of methods using a critical difference diagram based on RMSE, MAE, and SMAPE metrics across seven time series benchmark datasets.

A. NAMING SCHEME AND EVALUATION METRICS

This section outlines the naming scheme to be used for the short-term identification of comparative methods and describes the evaluation metrics employed.

1) EXPLANATION OF THE NAMING SCHEME USING THE SPECIFIED FORMAT

When determining the size of D_T, the naming for BEL-BEL and BEL-IK is straightforward since N_{S_T} (which equals the size of D_T) is fixed and involves no variation in hidden nodes. In contrast, for ELM-BEL and B2ELM-BEL, both N_{S_T} and the number of hidden nodes vary, resulting in more complex naming.

- Training Size Percentage ($\%D_T$): The dataset portion used for training.
- Method Short Name: Identifies the algorithm, like B2ELM-BEL.
- Number of Hidden Nodes: Indicates hidden nodes in the network.
- Transfer Learning Size Percentage ($\%N_{S_T}$): The training set part used for transfer learning.

For example, "(50%)-B2ELM-BEL-H1-5" means:

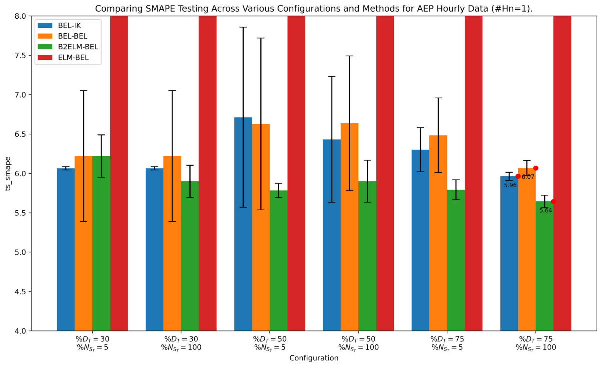
- "50%": Half of the dataset is for training ($\%\mathcal{D}_T = 50$).
- "B2ELM-BEL": The algorithm used.
- "H1": One hidden node.
- "5": 5% of D_T for transfer learning ($\%N_{S_T} = 5$).

This concise naming offers quick insights into the model setup, useful for analysis and comparison. $\%D_T$ can be omitted if already mentioned.

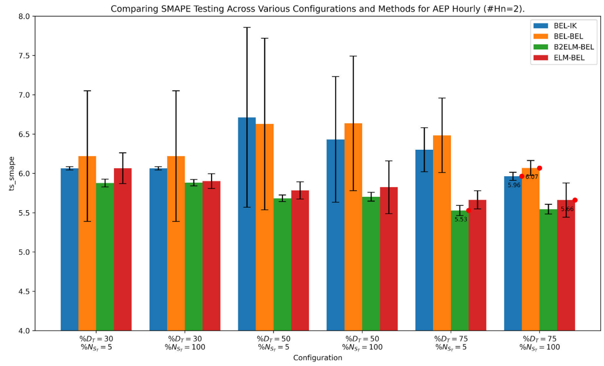
2) UNDERSTANDING THE TRAINING SIZE IN TRANSFER LEARNING MODELS

When working with transfer learning models, it's crucial to determine the size of the knowledge transfer learning set that will be utilized. This is represented by N_S in our calculations as follows:

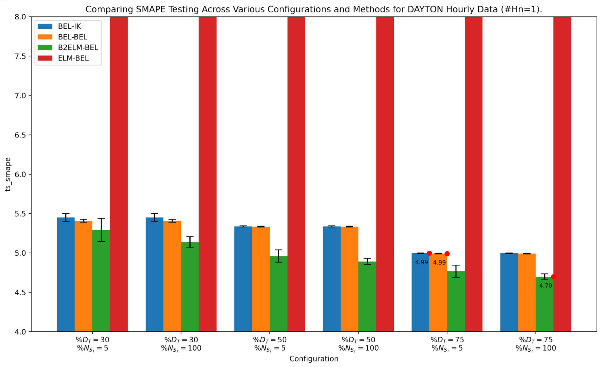
$$N_S = \%N_{S_T} \times \frac{N_T}{100}. \tag{40}$$



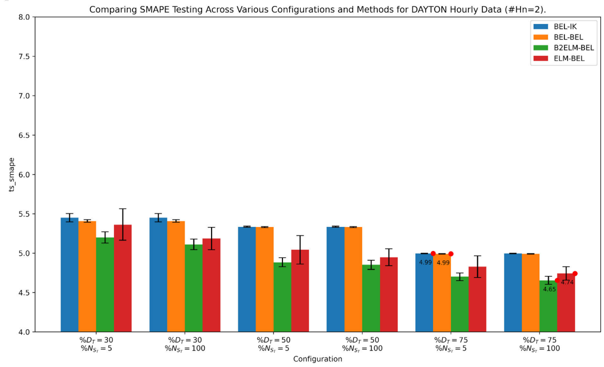
(a) **AEP Hourly Data #Hn=1: SMAPE-Based Comparative Analysis with Error Bars.**



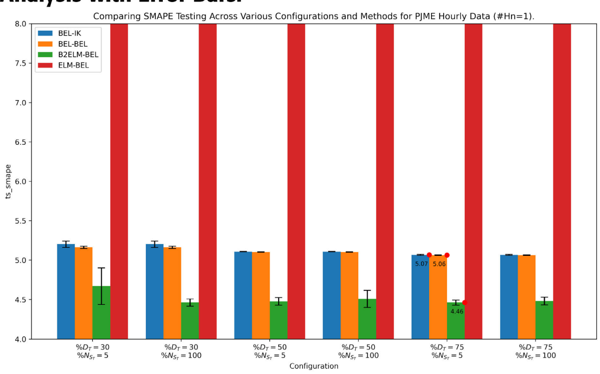
(b) **AEP Hourly Data #Hn=2: SMAPE-Based Comparative Analysis with Error Bars.**



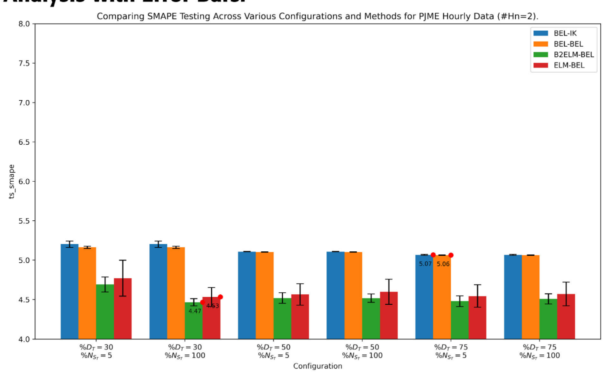
(c) **DAYTON Hourly Data #Hn=1: SMAPE-Based Comparative Analysis with Error Bars.**



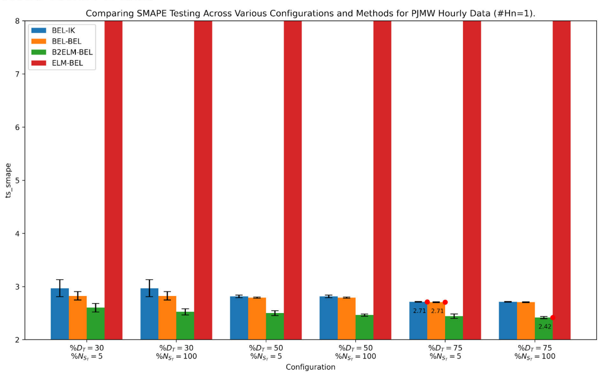
(d) **DAYTON Hourly Data #Hn=2: SMAPE-Based Comparative Analysis with Error Bars.**



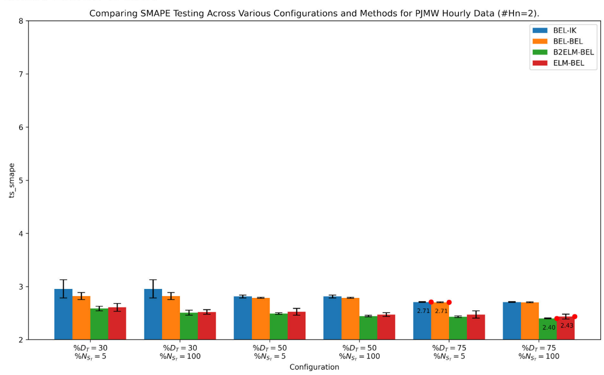
(e) **PJME Hourly #Hn=1: SMAPE-Based Comparative Analysis with Error Bars.**



(f) **PJME Hourly #Hn=2: SMAPE-Based Comparative Analysis with Error Bars.**



(g) **PJMW Hourly #Hn=1: SMAPE-Based Comparative Analysis with Error Bars.**



(h) **PJMW Hourly #Hn=2: SMAPE-Based Comparative Analysis with Error Bars.**

FIGURE 20. **Comparative analysis of different methods using testing SMAPE metrics across real-world data sets.**

TABLE 3. Training and testing time (Seconds) for Exptchao, Exptper, Exptqp2, and Exptqp3 problems.

Problems	Algorithms	#Hn	$\%N_{S_T}$	Training time (Training size%) 30	50	75	Testing time (Training size%) 30	50	75
Exptchao	BEL-IK	-	-	0.08977	0.16062	0.21923	0.00054	0.00042	0.00020
	BEL-BEL	-	-	1.48490	1.94297	1.93212	0.01148	0.00616	0.00188
	B2ELM-BEL	1	5	2.79411	3.66457	4.12663	0.03106	0.01741	0.00656
	ELM-BEL	1	5	3.32276	4.96905	7.65482	0.03499	0.02145	0.01170
	B2ELM-BEL	1	100	3.11045	3.64776	3.96081	0.03283	0.01732	0.00612
	ELM-BEL	1	100	3.16834	4.56649	7.10493	0.03420	0.02015	0.01111
	B2ELM-BEL	5	5	4.87923	5.05676	5.57796	0.05489	0.02363	0.00910
	ELM-BEL	5	5	3.57750	3.76084	3.54184	0.03943	0.01745	0.00571
	B2ELM-BEL	5	100	4.11794	5.49286	6.09121	0.04549	0.02667	0.00979
	ELM-BEL	5	100	3.13131	3.73619	4.72888	0.03349	0.01758	0.00744
Exptper	BEL-IK	-	-	0.08903	0.16391	0.21979	0.00059	0.00044	0.00021
	BEL-BEL	-	-	0.87243	1.32870	1.32912	0.00562	0.00329	0.00097
	B2ELM-BEL	1	5	1.59074	1.49204	1.96876	0.01761	0.00706	0.00313
	ELM-BEL	1	5	3.55835	5.12761	8.17898	0.03770	0.02222	0.01260
	B2ELM-BEL	1	100	1.53087	1.34454	1.65369	0.01576	0.00615	0.00253
	ELM-BEL	1	100	3.38670	4.58060	7.49973	0.03604	0.02036	0.01165
	B2ELM-BEL	5	5	1.73952	2.07233	2.10436	0.01954	0.00960	0.00341
	ELM-BEL	5	5	1.25688	1.44190	1.48336	0.01399	0.00666	0.00238
	B2ELM-BEL	5	100	1.85969	2.39426	2.29552	0.02048	0.01131	0.00350
	ELM-BEL	5	100	1.19513	1.80399	1.64739	0.01274	0.00824	0.00248
Exptqp2	BEL-IK	-	-	0.08869	0.15860	0.21974	0.00054	0.00042	0.00020
	BEL-BEL	-	-	1.34214	1.63798	1.97852	0.01013	0.00489	0.00194
	B2ELM-BEL	1	5	3.06439	3.19797	3.84294	0.03397	0.01525	0.00612
	ELM-BEL	1	5	3.53659	4.95811	7.87320	0.03750	0.02150	0.01207
	B2ELM-BEL	1	100	2.63771	2.86634	3.98288	0.02747	0.01352	0.00617
	ELM-BEL	1	100	3.35496	4.58077	7.15069	0.03562	0.02021	0.01114
	B2ELM-BEL	5	5	4.18294	4.50505	5.21291	0.04706	0.02103	0.00846
	ELM-BEL	5	5	3.29503	2.84061	3.68316	0.03690	0.01317	0.00589
	B2ELM-BEL	5	100	4.27865	4.88055	5.57408	0.04720	0.02339	0.00890
	ELM-BEL	5	100	3.21067	3.28199	4.41500	0.03433	0.01521	0.00693
Exptqp3	BEL-IK	-	-	0.08965	0.15917	0.21859	0.00053	0.00041	0.00021
	BEL-BEL	-	-	0.57714	0.86170	1.68343	0.00274	0.00105	0.00149
	B2ELM-BEL	1	5	0.65434	0.91742	4.12100	0.00722	0.00435	0.00668
	ELM-BEL	1	5	1.27577	2.91776	7.22995	0.01352	0.01270	0.01132
	B2ELM-BEL	1	100	0.54432	1.05804	2.05054	0.00532	0.00484	0.00322
	ELM-BEL	1	100	1.06418	2.22087	6.50390	0.01109	0.00980	0.01037
	B2ELM-BEL	5	5	0.70271	0.81980	1.20267	0.00774	0.00373	0.00196
	ELM-BEL	5	5	0.55412	0.54192	0.79402	0.00603	0.00248	0.00128
	B2ELM-BEL	5	100	0.66585	0.96719	1.39416	0.00723	0.00421	0.00202
	ELM-BEL	5	100	0.46606	0.73552	1.01367	0.00490	0.00308	0.00147

where:

- N_T: This is the total number of training data samples in the models. It represents the full set of data available for training purposes.

- $\%N_{S_T}$: This percentage indicates how much of the total training data (represented by N_T) is being used for transfer learning. If $\%N_{S_T}$=100, then $N_{S_T} = N_T$.

Example: Using the last row of Table 1 for $\%D_T$=75. Consider the B2ELM-BEL model, where the total training data (N_T) consists of 6144 samples. If we set $\%N_{S_T}$ to 100%, it means we are using the entire training dataset for transfer learning. Therefore, in this case, the number of data samples for transfer learning (N_S) is also 6144. The model name will be (75%)-B2ELM-BEL-Hn-100, where n represents the number of hidden nodes in the model.

By adopting this approach, it is hoped that the configuration of an experiment is straightforward and allow for the accurate calculation of the specific amount of original training data used in any given transfer learning mode. Alternatively, refer to Fig. 6 for more details.

3) EVALUATION METRICS

For every experiment conducted, it was important to evaluate how well the algorithm performed. In this paper, we used three key measures for this evaluation: the root mean square error rate (RMSE), the mean absolute error (MAE), and the symmetric mean absolute percentage error (SMAPE). Their formulas are written as follows:

$$RMSE = \sqrt{\left(\frac{1}{N}\sum_{i}^{N}\left(Y_i - \hat{Y}_i\right)^2\right)}, \tag{41}$$

TABLE 4. Training and testing time(seconds) for Henon, Lorenz and Rossler problems.

Problems	Algorithms	#Hn	$\%N_{S_T}$	Training time: Training size% 30	50	75	Testing time: Training size% 30	50	75
Henon	BEL-IK	-	-	0.08856	0.16329	0.22007	0.00055	0.00041	0.00021
	BEL-BEL	-	-	1.70468	2.78654	2.87752	0.01367	0.00977	0.00324
	B2ELM-BEL	1	5	4.54430	6.60529	10.74830	0.05007	0.03037	0.01645
	ELM-BEL	1	5	3.59623	5.32434	8.48035	0.03753	0.02226	0.01242
	B2ELM-BEL	1	100	4.40317	5.81173	10.12130	0.04658	0.02731	0.01510
	ELM-BEL	1	100	3.35371	4.62318	7.64607	0.03668	0.02007	0.01149
	B2ELM-BEL	5	5	0.21854	0.31573	0.42763	0.00216	0.00136	0.00066
	ELM-BEL	5	5	0.15724	0.21324	0.30966	0.00151	0.00092	0.00047
	B2ELM-BEL	5	100	0.19163	0.44442	0.64842	0.00195	0.00155	0.00076
	ELM-BEL	5	100	0.13520	0.35759	0.80941	0.00132	0.00111	0.00109
Lorenz	BEL-IK	-	-	0.09086	0.16009	0.23394	0.00054	0.00042	0.00024
	BEL-BEL	-	-	1.34129	2.33765	2.43341	0.01075	0.00765	0.00273
	B2ELM-BEL	1	5	3.34015	3.88932	4.67059	0.03701	0.01823	0.00709
	ELM-BEL	1	5	3.41466	5.08320	8.24560	0.03814	0.02366	0.01293
	B2ELM-BEL	1	100	3.58373	4.15853	4.29212	0.03782	0.02067	0.00684
	ELM-BEL	1	100	3.34996	4.50230	7.31107	0.03482	0.02081	0.01139
	B2ELM-BEL	5	5	4.85492	6.20452	7.15430	0.05364	0.02999	0.01148
	ELM-BEL	5	5	3.76063	4.39932	4.92854	0.04108	0.02111	0.00779
	B2ELM-BEL	5	100	4.31749	6.67176	7.26416	0.04785	0.03204	0.01147
	ELM-BEL	5	100	3.24533	5.27185	5.58114	0.03561	0.02525	0.00882
Rossler	BEL-IK	-	-	0.04513	0.08081	0.11269	0.00027	0.00021	0.00010
	BEL-BEL	-	-	0.59879	0.77000	0.82124	0.00432	0.00210	0.00074
	B2ELM-BEL	1	5	1.41739	1.31880	1.65435	0.01543	0.00626	0.00257
	ELM-BEL	1	5	1.83676	2.59841	3.06824	0.01854	0.01122	0.00459
	B2ELM-BEL	1	100	1.31084	1.18567	1.47615	0.01316	0.00551	0.00224
	ELM-BEL	1	100	1.77265	2.26041	2.72117	0.01788	0.00995	0.00414
	B2ELM-BEL	5	5	1.95728	1.90522	1.93301	0.02148	0.00886	0.00316
	ELM-BEL	5	5	1.35189	1.23989	1.49349	0.01471	0.00574	0.00242
	B2ELM-BEL	5	100	1.96140	2.25045	2.25167	0.02158	0.01055	0.00357
	ELM-BEL	5	100	1.27369	1.48429	1.58453	0.01351	0.00684	0.00246

$$MAE = \frac{1}{N}\sum_{i}^{N} |Y_i - \hat{Y}_i|, \tag{42}$$

$$SMAPE = \frac{1}{N}\sum_{i}^{N} \frac{|Y_i - \hat{Y}_i|}{\left(|Y_i| + |\hat{Y}_i|\right)/2}, \tag{43}$$

where Y_i, $\hat{Y}_i$ are an actual target value and a predicted value of the i-th sample.

4) ALGORITHM PERFORMANCE COMPARATIVE ANALYSIS

The results of the experiment are presented in a series of tables. These tables contain data for analyzing the performance of various machine learning models, with a primary focus on ELM-related models. We examined two versions of these models: one with a single hidden node and another with multiple hidden nodes, specifically tailored for the time series benchmark dataset. The objective was to evaluate the models' effectiveness in predicting complex chaotic time series data.

Table 11 functions as a comprehensive summary of testing performance metrics, denoted by Total Score and Position. The methodology for calculating the Total Score and Position can be explained as follows:

- Total Score: A lower Total Score indicates better performance. The Total Score for each method is the cumulative sum of ranks obtained from three different percentages of the training size ($\%D_T = 30$, $\%D_T = 50$, and $\%D_T = 75$) across different metrics (RMSEs, MAEs, and SMAPEs).
  Formula:

$$\text{Total Score} = \text{Rank}_{\%D_T=30} + \text{Rank}_{\%D_T=50} + \text{Rank}_{\%D_T=75}. \tag{44}$$

  In this formula, $\text{Rank}_{\%D_T=30} + \text{Rank}_{\%D_T=50} + \text{Rank}_{\%D_T=75}$ represent the ranks of the method for each specified training size percentage, which are typically averaged across the metrics RMSEs, MAEs, and SMAPEs. Based on the experimental results, the ranks for each method, varying by #Hn, $\%N_{S_T}$, metrics and $\%D_T$, are shown in Tables 12, 13, and 14.
- Position: A lower Position indicates better performance. The Position of each method is determined based on the lowest Total Score, as follows:
  Let $S = \{s_1, s_2, \ldots, s_n\}$ be the set of Total Scores for n methods, where each s_i is the Total Score of the i-th method.

TABLE 5. Comparison of testing performance metrics for Exptchao, Exptper, Exptqp2 and Exptqp3 problems where #Hn, $\%N_{S_T}$ is the number of hidden nodes and the training size percentage for specifying the transfer learning size, respectively.

Problems	#Hn	$\%N_{S_T}$	Testing evaluation metrics								
			RMSE Training size percentage			MAE Training size percentage			SMAPE Training size percentage		
Methods			30	50	75	30	50	75	30	50	75
Exptchao											
BEL-IK	-	-	0.00473	0.00480	0.00445	0.00376	0.00382	0.00361	0.81458	0.84316	0.68753
BEL-BEL	-	-	0.00460	0.00479	0.00445	0.00367	0.00381	0.00362	0.79107	0.83825	0.68793
B2ELM-BEL	1	5	0.00301	0.00292	0.00248	0.00232	0.00224	0.00196	0.62106	0.67464	0.39162
ELM-BEL	1	5	0.10549	0.11358	0.11161	0.08645	0.09211	0.09051	17.17530	18.45050	17.42500
B2ELM-BEL	1	100	0.00264	0.00256	0.00250	0.00208	0.00203	0.00197	0.50728	0.52661	0.39071
ELM-BEL	1	100	0.10425	0.11103	0.11035	0.08544	0.09011	0.08946	16.99630	18.12560	17.25510
B2ELM-BEL	5	5	0.00227	0.00221	0.00209	0.00179	0.00175	0.00166	0.44932	0.45423	0.31524
ELM-BEL	5	5	0.00235	0.00230	0.00211	0.00186	0.00180	0.00168	0.46905	0.49543	0.32319
B2ELM-BEL	5	100	0.00234	0.00212	0.00206	0.00182	0.00168	0.00163	0.46773	0.39367	0.30331
ELM-BEL	5	100	0.00233	0.00213	0.00207	0.00184	0.00168	0.00164	0.47045	0.39801	0.30482
Exptper											
BEL-IK	-	-	0.00925	0.00906	0.00915	0.00745	0.00733	0.00738	3.30441	3.03883	2.80875
BEL-BEL	-	-	0.00925	0.00907	0.00915	0.00745	0.00733	0.00738	3.30612	3.03714	2.80849
B2ELM-BEL	1	5	0.00722	0.00732	0.00654	0.00578	0.00588	0.00520	3.82892	3.48360	2.57993
ELM-BEL	1	5	0.19176	0.19601	0.19484	0.16536	0.16483	0.16835	49.87690	49.76190	49.48450
B2ELM-BEL	1	100	0.00672	0.00669	0.00685	0.00535	0.00534	0.00547	3.17769	2.85764	2.85328
ELM-BEL	1	100	0.19821	0.19564	0.19470	0.17111	0.16464	0.16794	50.75460	49.72290	49.42690
B2ELM-BEL	5	5	0.00583	0.00580	0.00574	0.00459	0.00456	0.00450	2.49368	2.13880	1.99139
ELM-BEL	5	5	0.00583	0.00580	0.00574	0.00459	0.00456	0.00450	2.45709	2.10693	1.98147
B2ELM-BEL	5	100	0.00583	0.00580	0.00574	0.00459	0.00457	0.00451	2.49305	2.10216	1.97707
ELM-BEL	5	100	0.00583	0.00581	0.00574	0.00459	0.00457	0.00451	2.46557	2.10406	1.97726
Exptqp2											
BEL-IK	-	-	0.00477	0.00518	0.00467	0.00385	0.00417	0.00376	1.12548	1.05789	0.99564
BEL-BEL			0.00482	0.00518	0.00467	0.00386	0.00417	0.00376	1.11571	1.05487	0.99693
B2ELM-BEL	1	5	0.00332	0.00331	0.00308	0.00261	0.00261	0.00244	0.92381	0.88688	0.77787
ELM-BEL	1	5	0.16117	0.16057	0.15177	0.13255	0.13201	0.12629	30.55940	30.11040	29.73660
B2ELM-BEL	1	100	0.00353	0.00341	0.00327	0.00276	0.00269	0.00258	1.02241	0.87474	0.85671
ELM-BEL	1	100	0.16513	0.15858	0.15168	0.13590	0.13037	0.12619	31.14260	29.81260	29.73040
B2ELM-BEL	5	5	0.00267	0.00262	0.00254	0.00209	0.00206	0.00199	0.59134	0.53024	0.50456
ELM-BEL	5	5	0.00267	0.00262	0.00254	0.00210	0.00206	0.00200	0.59395	0.53127	0.50444
B2ELM-BEL	5	100	0.00268	0.00262	0.00253	0.00210	0.00206	0.00199	0.59302	0.53036	0.50297
ELM-BEL	5	100	0.00268	0.00261	0.00254	0.00210	0.00205	0.00200	0.59669	0.52939	0.50347
Exptqp3											
BEL-IK	-	-	0.02475	0.02443	0.03430	0.01892	0.01830	0.02849	5.67136	5.59000	7.54276
BEL-BEL	-	-	0.02580	0.02443	0.03573	0.01996	0.01830	0.02970	5.83527	5.58987	7.73125
B2ELM-BEL	1	5	0.01917	0.01967	0.02358	0.01486	0.01529	0.01911	4.08235	4.17304	5.16403
ELM-BEL	1	5	0.10225	0.10463	0.11829	0.08512	0.08730	0.09750	20.49140	20.85050	24.28070
B2ELM-BEL	1	100	0.01914	0.01987	0.02176	0.01480	0.01541	0.01751	4.07786	4.18894	4.85405
ELM-BEL	1	100	0.09793	0.10494	0.11524	0.08154	0.08759	0.09512	19.76820	20.90030	23.82700
B2ELM-BEL	5	5	0.01798	0.01834	0.01975	0.01394	0.01431	0.01573	3.86636	3.96114	4.44985
ELM-BEL	5	5	0.01806	0.01841	0.01975	0.01399	0.01433	0.01569	3.92192	3.98040	4.47146
B2ELM-BEL	5	100	0.01798	0.01809	0.01894	0.01394	0.01408	0.01488	3.88514	3.91891	4.27819
ELM-BEL	5	100	0.01808	0.01820	0.01917	0.01399	0.01415	0.01504	3.91774	3.93753	4.33766

- Sort S in ascending order to get $S' = \{s'_1, s'_2, \ldots, s'_n\}$, where s'_1 is the smallest score, s'_2 is the second smallest, and so on.
- Assign a position to each method based on its rank in S'. The position function $P(i)$ for the i-th method in the original set S is given by:

$$P(i) = \text{the index of } s_i \text{ in } S'. \tag{45}$$

This means that $P(i)$ returns the position of the i-th method's score in the sorted list.

B. EXPERIMENTAL SETUP

This experimental phase is designed to assess how different training sample sizes affect the forecast accuracy of the B2ELM-BEL and ELM-BEL models. It employs seven unique chaotic time series datasets. The total sample sizes of these datasets are listed in Table 1, which originates from [38]. For preprocessing the data from the seven chaotic time series datasets, we rescaled each dataset to fit within the range [0,1]. We then prepared the features of each sample using a sliding window with a size of 2. After rescaling, the

TABLE 6. Comparison of testing performance metrics for Henon, Lorenz and Rossler problems where #Hn, $\%N_{S_T}$ is the number of hidden nodes and the training size percentage for specifying the transfer learning size, respectively.

Problems	#Hn	$\%N_{S_T}$	Testing evaluation metrics								
			RMSE Training size percentage			MAE Training size percentage			SMAPE Training size percentage		
Methods			30	50	75	30	50	75	30	50	75
Henon											
BEL-IK	-	-	0.3634	0.3798	0.4373	0.3166	0.3209	0.3465	59.6816	59.1374	59.2360
BEL-BEL	-	-	0.3664	0.3878	0.4828	0.3173	0.3230	0.3722	59.3247	58.2359	58.5809
B2ELM-BEL	1	5	0.2583	0.2603	0.2824	0.2212	0.2213	0.2300	44.0589	43.7493	44.3726
ELM-BEL	1	5	0.2615	0.2623	0.2902	0.2245	0.2233	0.2355	44.5382	44.0465	44.8100
B2ELM-BEL	1	100	0.2577	0.3918	0.2802	0.2221	0.3253	0.2299	44.2767	58.2589	44.3712
ELM-BEL	1	100	0.2602	0.2674	0.2849	0.2243	0.2258	0.2335	44.5951	44.2519	44.8585
B2ELM-BEL	5	5	0.0038	0.0039	0.0041	0.0028	0.0029	0.0031	2.0898	2.0372	2.0190
ELM-BEL	5	5	0.0080	0.0095	0.0103	0.0063	0.0073	0.0079	3.6711	3.7881	3.9265
B2ELM-BEL	5	100	0.0039	0.0021	0.0039	0.0030	0.0016	0.0031	2.1919	0.8286	2.3973
ELM-BEL	5	100	0.0094	0.0041	0.0093	0.0075	0.0031	0.0075	4.0084	1.3387	4.5189
Lorenz											
BEL-IK	-	-	0.00547	0.00404	0.00386	0.00470	0.00330	0.00311	1.24931	0.98423	0.92179
BEL-BEL	-	-	0.00444	0.00391	0.00384	0.00374	0.00313	0.00306	1.02340	0.94183	0.91159
B2ELM-BEL	1	5	0.00246	0.00234	0.00271	0.00197	0.00190	0.00232	0.59336	0.68585	0.72385
ELM-BEL	1	5	0.20389	0.13584	0.14889	0.16897	0.11188	0.12810	35.93090	26.96110	31.15620
B2ELM-BEL	1	100	0.00302	0.00223	0.00195	0.00233	0.00183	0.00162	0.80385	0.66344	0.49187
ELM-BEL	1	100	0.22372	0.14487	0.14918	0.18530	0.11930	0.12839	39.50800	28.19690	31.20020
B2ELM-BEL	5	5	0.00136	0.00067	0.00059	0.00108	0.00053	0.00045	0.30795	0.17934	0.12877
ELM-BEL	5	5	0.00132	0.00068	0.00060	0.00103	0.00053	0.00047	0.31406	0.18781	0.13265
B2ELM-BEL	5	100	0.00131	0.00064	0.00056	0.00104	0.00049	0.00044	0.29664	0.16360	0.11964
ELM-BEL	5	100	0.00130	0.00068	0.00060	0.00102	0.00051	0.00047	0.29795	0.17182	0.12536
Rossler											
BEL-IK	-	-	0.00762	0.00758	0.00770	0.00579	0.00581	0.00583	2.18104	2.20354	2.26126
BEL-BEL	-	-	0.00762	0.00758	0.00768	0.00579	0.00582	0.00583	2.18127	2.20396	2.26231
B2ELM-BEL	1	5	0.00478	0.00453	0.00482	0.00266	0.00264	0.00278	2.00228	1.93100	2.21502
ELM-BEL	1	5	0.15836	0.14984	0.15407	0.13439	0.12572	0.12859	37.85140	36.62800	37.62440
B2ELM-BEL	1	100	0.00472	0.00455	0.00491	0.00272	0.00272	0.00294	1.94446	1.96656	2.54513
ELM-BEL	1	100	0.15642	0.15481	0.15548	0.13280	0.12984	0.12986	37.58210	37.35200	37.84520
B2ELM-BEL	5	5	0.00376	0.00300	0.00300	0.00132	0.00201	0.00186	0.37700	0.94696	0.91972
ELM-BEL	5	5	0.00384	0.00336	0.00335	0.00142	0.00196	0.00176	0.43345	0.85673	0.82014
B2ELM-BEL	5	100	0.00377	0.00291	0.00295	0.00139	0.00171	0.00171	0.42952	0.81153	0.86492
ELM-BEL	5	100	0.00384	0.00329	0.00334	0.00142	0.00170	0.00165	0.42059	0.77628	0.80274

samples were divided into training and testing sets according to the percentage designated for training, denoted as $\%D_T$. For B2ELM-BEL and ELM-BEL, the size of their transfer learning is defined according to $\%N_{S_T}$. Fig. 7 displays example graphs for each of these seven datasets.

In these experiments, both models were trained using a single or five hidden neurons (#Hn $= \{1, 5\}$). Each set of experiments was repeated over 30 different runs.

The training regimen of the B2ELM-BEL model includes adjusting the $\%N_{S_T}$ parameter, which represents the proportion of the training data allocated for the knowledge transfer learning size. This process is detailed in Table 1. For instance, in our approach, 30% of the total data (2458 samples) was assigned for D_T, and 5% of D_T (specifically the first 123 samples) was used as the knowledge transfer learning size ($\%N_{S_T} = 5$) for D_S, as shown in the last row of Table 1. This example illustrates the application of $\%N_{S_T}$ in the model's training process, highlighting its role in the allocation of data for knowledge transfer learning.

Meanwhile, the ELM-BEL's training mirrors that of the B2ELM-BEL but differentiates by employing an ELM devoid of a bias component at the output.

The foundational BEL-IK model was also benchmarked against its augmented counterpart, the BEL-Guided BEL (BEL-BEL). The BEL-BEL is an advanced iteration of BEL-IK, leveraging the latter to facilitate knowledge transfer that modulates the thalamic stimulus A_i^{th}. In this paper, the knowledge transfer learning data for BEL-BEL was the same as the training data.

C. ASSESSING MODEL PARAMETERS IN PREDICTIVE TASKS: A COMPARATIVE STUDY WITH ROSSLER, HENON, AND AEP HOURLY DATASETS

This subsection explores the optimization of BEL models, focusing on the impact of the number of hidden nodes and learning rate adjustments on prediction accuracy. The goal is to systematically demonstrate their influence on the

TABLE 7. Comparison of testing performance metric standard deviations (SD) for Exptchao, Exptper, Exptqp2 and Exptqp3 problems where #Hn, $\%N_{S_T}$ is the number of hidden nodes and the training size percentage for specifying the transfer learning size, respectively.

Problems	#Hn	$\%N_{S_T}$	Standard deviations (SD) of testing metric								
			RMSE Training size percentage			MAE Training size percentage			SMAPE Training size percentage		
Methods			30	50	75	30	50	75	30	50	75
Exptchao											
BEL-IK	-	-	1.9E-04	1.5E-05	5.6E-06	1.3E-04	9.5E-06	2.8E-06	2.6E-02	1.9E-03	2.3E-04
BEL-BEL	-	-	6.0E-05	2.1E-05	2.3E-05	2.8E-05	1.9E-05	1.0E-05	3.9E-03	6.4E-03	1.8E-03
B2ELM-BEL	1	5	7.6E-04	9.3E-04	5.1E-04	5.6E-04	6.7E-04	3.9E-04	2.4E-01	2.8E-01	1.8E-01
ELM-BEL	1	5	8.6E-03	9.8E-03	1.1E-02	7.1E-03	8.0E-03	9.1E-03	1.2E+00	1.3E+00	1.5E+00
B2ELM-BEL	1	100	4.3E-04	5.2E-04	5.7E-04	3.3E-04	4.4E-04	3.9E-04	1.4E-01	1.3E-01	2.0E-01
ELM-BEL	1	100	1.0E-02	1.0E-02	1.1E-02	8.4E-03	8.2E-03	9.1E-03	1.5E+00	1.4E+00	1.5E+00
B2ELM-BEL	5	5	6.8E-05	5.1E-05	5.0E-05	4.3E-05	3.3E-05	2.6E-05	4.0E-02	3.0E-02	4.6E-02
ELM-BEL	5	5	1.3E-04	7.9E-05	4.5E-05	1.0E-04	4.3E-05	3.1E-05	4.1E-02	3.9E-02	3.5E-02
B2ELM-BEL	5	100	2.4E-04	1.5E-05	1.4E-05	1.2E-04	1.3E-05	1.0E-05	1.2E-01	3.2E-03	4.3E-03
ELM-BEL	5	100	9.0E-05	1.1E-05	1.3E-05	6.7E-05	9.7E-06	1.0E-05	3.0E-02	3.1E-03	8.8E-03
Exptper											
BEL-IK	-	-	9.3E-03	9.1E-03	9.1E-03	7.5E-03	7.3E-03	7.4E-03	3.3E+00	3.0E+00	2.8E+00
BEL-BEL	-	-	9.2E-03	9.1E-03	9.1E-03	7.5E-03	7.3E-03	7.4E-03	3.3E+00	3.0E+00	2.8E+00
B2ELM-BEL	1	5	7.2E-03	7.3E-03	6.5E-03	5.8E-03	5.9E-03	5.2E-03	3.8E+00	3.5E+00	2.6E+00
ELM-BEL	1	5	1.9E-01	2.0E-01	1.9E-01	1.7E-01	1.6E-01	1.7E-01	5.0E+01	5.0E+01	4.9E+01
B2ELM-BEL	1	100	6.7E-03	6.7E-03	6.8E-03	5.3E-03	5.3E-03	5.5E-03	3.2E+00	2.9E+00	2.9E+00
ELM-BEL	1	100	2.0E-01	2.0E-01	1.9E-01	1.7E-01	1.6E-01	1.7E-01	5.1E+01	5.0E+01	4.9E+01
B2ELM-BEL	5	5	5.8E-03	5.8E-03	5.7E-03	4.6E-03	4.6E-03	4.5E-03	2.5E+00	2.1E+00	2.0E+00
ELM-BEL	5	5	5.8E-03	5.8E-03	5.7E-03	4.6E-03	4.6E-03	4.5E-03	2.5E+00	2.1E+00	2.0E+00
B2ELM-BEL	5	100	5.8E-03	5.8E-03	5.7E-03	4.6E-03	4.6E-03	4.5E-03	2.5E+00	2.1E+00	2.0E+00
ELM-BEL	5	100	5.8E-03	5.8E-03	5.7E-03	4.6E-03	4.6E-03	4.5E-03	2.5E+00	2.1E+00	2.0E+00
Exptqp2											
BEL-IK	-	-	4.8E-03	5.2E-03	4.7E-03	3.8E-03	4.2E-03	3.8E-03	1.1E+00	1.1E+00	1.0E+00
BEL-BEL	-	-	4.8E-03	5.2E-03	4.7E-03	3.9E-03	4.2E-03	3.8E-03	1.1E+00	1.1E+00	1.0E+00
B2ELM-BEL	1	5	3.3E-03	3.3E-03	3.1E-03	2.6E-03	2.6E-03	2.4E-03	9.2E-01	8.9E-01	7.8E-01
ELM-BEL	1	5	1.6E-01	1.6E-01	1.5E-01	1.3E-01	1.3E-01	1.3E-01	3.1E+01	3.0E+01	3.0E+01
B2ELM-BEL	1	100	3.5E-03	3.4E-03	3.3E-03	2.8E-03	2.7E-03	2.6E-03	1.0E+00	8.7E-01	8.6E-01
ELM-BEL	1	100	1.7E-01	1.6E-01	1.5E-01	1.4E-01	1.3E-01	1.3E-01	3.1E+01	3.0E+01	3.0E+01
B2ELM-BEL	5	5	2.7E-03	2.6E-03	2.5E-03	2.1E-03	2.1E-03	2.0E-03	5.9E-01	5.3E-01	5.0E-01
ELM-BEL	5	5	2.7E-03	2.6E-03	2.5E-03	2.1E-03	2.1E-03	2.0E-03	5.9E-01	5.3E-01	5.0E-01
B2ELM-BEL	5	100	2.7E-03	2.6E-03	2.5E-03	2.1E-03	2.1E-03	2.0E-03	5.9E-01	5.3E-01	5.0E-01
ELM-BEL	5	100	2.7E-03	2.6E-03	2.5E-03	2.1E-03	2.1E-03	2.0E-03	6.0E-01	5.3E-01	5.0E-01
Exptqp3											
BEL-IK	-	-	2.5E-02	3.0E-02	3.4E-02	1.9E-02	2.5E-02	2.8E-02	5.7E+00	6.6E+00	7.5E+00
BEL-BEL	-	-	2.6E-02	2.4E-02	3.6E-02	2.0E-02	1.8E-02	3.0E-02	5.8E+00	5.6E+00	7.7E+00
B2ELM-BEL	1	5	1.9E-02	2.0E-02	2.4E-02	1.5E-02	1.5E-02	1.9E-02	4.1E+00	4.2E+00	5.2E+00
ELM-BEL	1	5	1.0E-01	1.0E-01	1.2E-01	8.5E-02	8.7E-02	9.7E-02	2.0E+01	2.1E+01	2.4E+01
B2ELM-BEL	1	100	1.9E-02	2.0E-02	2.2E-02	1.5E-02	1.5E-02	1.8E-02	4.1E+00	4.2E+00	4.9E+00
ELM-BEL	1	100	9.8E-02	1.0E-01	1.2E-01	8.2E-02	8.8E-02	9.5E-02	2.0E+01	2.1E+01	2.4E+01
B2ELM-BEL	5	5	1.8E-02	1.8E-02	2.0E-02	1.4E-02	1.4E-02	1.6E-02	3.9E+00	4.0E+00	4.4E+00
ELM-BEL	5	5	1.8E-02	1.8E-02	2.0E-02	1.4E-02	1.4E-02	1.6E-02	3.9E+00	4.0E+00	4.5E+00
B2ELM-BEL	5	100	1.8E-02	1.8E-02	1.9E-02	1.4E-02	1.4E-02	1.5E-02	3.9E+00	3.9E+00	4.3E+00
ELM-BEL	5	100	1.8E-02	1.8E-02	1.9E-02	1.4E-02	1.4E-02	1.5E-02	3.9E+00	3.9E+00	4.3E+00

efficiency and effectiveness of BEL models in forecasting tasks.

Various configurations of BEL models, including both ELM and Biased-ELM, are employed across three datasets: Rossler, Exptqp3, and Henon. The Rossler and Exptqp3 datasets serve as a baseline for prediction accuracy, while the Henon dataset presents a more complex challenge.

In the study of the Rossler problem, a comparative analysis of the Biased-ELM and ELM models was conducted, each undergoing 30 tests with a single hidden node. The Biased-ELM model notably outperformed the ELM model, achieving a lower average testing RMSE (refer to Fig. 8).

The number of hidden nodes greatly affects performance, particularly when there are fewer hidden nodes. This was true of the ELM-BEL model, which faced more difficulties compared to the B2ELM-BEL model. However, as the number of hidden nodes increases, the performance of both models tends to converge. This is illustrated in Fig. 10.

In analyzing the complex Henon problem, significant performance differences were observed between the

TABLE 8. Comparison of testing performance metric standard deviations (SD) for Henon, Lorenz and Rossler problems where #Hn, $\%N_{S_T}$ is the number of hidden nodes and the training size percentage for specifying the transfer learning size, respectively.

Problems	#Hn	$\%N_{S_T}$	Standard deviations (SD) of testing metric								
			RMSE Training size percentage			MAE Training size percentage			SMAPE Training size percentage		
Methods			30	50	75	30	50	75	30	50	75
Henon											
BEL-IK	-	-	9.36E-03	4.34E-02	1.01E-01	1.69E-03	1.63E-02	4.99E-02	1.61E+00	2.47E+00	3.26E+00
BEL-BEL	-	-	1.32E-02	4.01E-02	1.46E-01	2.74E-03	1.47E-02	8.44E-02	1.88E+00	2.69E+00	3.50E+00
B2ELM-BEL	1	5	2.21E-03	7.73E-03	3.61E-02	1.44E-03	2.77E-03	2.03E-02	2.88E-01	2.40E-01	1.32E+00
ELM-BEL	1	5	4.32E-03	1.02E-02	5.04E-02	1.72E-03	3.95E-03	2.96E-02	2.76E-01	3.34E-01	1.94E+00
B2ELM-BEL	1	100	1.83E-03	1.32E-02	3.74E-02	1.40E-03	5.16E-03	2.19E-02	2.58E-01	3.05E-01	1.52E+00
ELM-BEL	1	100	2.78E-03	1.88E-02	3.44E-02	1.71E-03	8.10E-03	1.60E-02	2.79E-01	5.58E-01	8.19E-01
B2ELM-BEL	5	5	1.44E-03	1.83E-03	2.07E-03	1.06E-03	1.30E-03	1.53E-03	8.24E-01	7.28E-01	7.85E-01
ELM-BEL	5	5	3.71E-03	4.58E-03	4.16E-03	2.96E-03	3.45E-03	3.05E-03	1.09E+00	1.53E+00	1.37E+00
B2ELM-BEL	5	100	1.83E-03	2.06E-03	1.71E-03	1.33E-03	1.57E-03	1.41E-03	8.11E-01	8.29E-01	1.14E+00
ELM-BEL	5	100	4.41E-03	4.15E-03	3.68E-03	3.59E-03	3.09E-03	2.93E-03	1.67E+00	1.34E+00	1.42E+00
Lorenz											
BEL-IK	-	-	5.5E-03	4.0E-03	3.9E-03	4.7E-03	3.3E-03	3.1E-03	1.2E+00	9.8E-01	9.2E-01
BEL-BEL	-	-	4.4E-03	3.9E-03	3.8E-03	3.7E-03	3.1E-03	3.1E-03	1.0E+00	9.4E-01	9.1E-01
B2ELM-BEL	1	5	2.5E-03	2.3E-03	2.7E-03	2.0E-03	1.9E-03	2.3E-03	5.9E-01	6.9E-01	7.2E-01
ELM-BEL	1	5	2.0E-01	1.4E-01	1.5E-01	1.7E-01	1.1E-01	1.3E-01	3.6E+01	2.7E+01	3.1E+01
B2ELM-BEL	1	100	3.0E-03	2.2E-03	2.0E-03	2.3E-03	1.8E-03	1.6E-03	8.0E-01	6.6E-01	4.9E-01
ELM-BEL	1	100	2.2E-01	1.4E-01	1.5E-01	1.9E-01	1.2E-01	1.3E-01	4.0E+01	2.8E+01	3.1E+01
B2ELM-BEL	5	5	1.4E-03	6.7E-04	5.9E-04	1.1E-03	5.3E-04	4.5E-04	3.1E-01	1.8E-01	1.3E-01
ELM-BEL	5	5	1.3E-03	6.8E-04	6.0E-04	1.0E-03	5.2E-04	4.7E-04	3.1E-01	1.9E-01	1.3E-01
B2ELM-BEL	5	100	1.3E-03	6.4E-04	5.6E-04	1.0E-03	4.9E-04	4.4E-04	3.0E-01	1.6E-01	1.2E-01
ELM-BEL	5	100	1.3E-03	6.8E-04	6.0E-04	1.0E-03	5.1E-04	4.7E-04	3.0E-01	1.7E-01	1.3E-01
Rossler											
BEL-IK	-	-	7.6E-03	7.6E-03	7.7E-03	5.8E-03	5.8E-03	5.8E-03	2.2E+00	2.2E+00	2.3E+00
BEL-BEL	-	-	7.6E-03	7.6E-03	7.7E-03	5.8E-03	5.8E-03	5.8E-03	2.2E+00	2.2E+00	2.3E+00
B2ELM-BEL	1	5	4.8E-03	4.5E-03	4.8E-03	2.7E-03	2.6E-03	2.8E-03	2.0E+00	1.9E+00	2.2E+00
ELM-BEL	1	5	1.6E-01	1.5E-01	1.5E-01	1.3E-01	1.3E-01	1.3E-01	3.8E+01	3.7E+01	3.8E+01
B2ELM-BEL	1	100	4.7E-03	4.6E-03	4.9E-03	2.7E-03	2.7E-03	2.9E-03	1.9E+00	2.0E+00	2.5E+00
ELM-BEL	1	100	1.6E-01	1.5E-01	1.6E-01	1.3E-01	1.3E-01	1.3E-01	3.8E+01	3.7E+01	3.8E+01
B2ELM-BEL	5	5	3.8E-03	3.0E-03	3.0E-03	1.3E-03	2.0E-03	1.9E-03	3.8E-01	9.5E-01	9.2E-01
ELM-BEL	5	5	3.8E-03	3.4E-03	3.3E-03	1.4E-03	2.0E-03	1.8E-03	4.3E-01	8.6E-01	8.2E-01
B2ELM-BEL	5	100	3.8E-03	2.9E-03	2.9E-03	1.4E-03	1.7E-03	1.7E-03	4.3E-01	8.1E-01	8.6E-01
ELM-BEL	5	100	3.8E-03	3.3E-03	3.3E-03	1.4E-03	1.7E-03	1.6E-03	4.2E-01	7.8E-01	8.0E-01

B2ELM-BEL and ELM-BEL models, especially with less than five hidden nodes. Their alignment with actual data was notably better than other BEL models (refer to Figs 12(a-b)).

When setting the learning rate for training BEL models, choosing an improper value can sometimes lead to undesirable weights **w** and **v**, as shown in Fig. 11 The experiments identified 0.0001 as a suitable learning rate. This rate not only yielded desirable weights **w** and **v** but also produced the best results across multiple metrics for all tested models, as shown in Fig. 13.

This study underscores the vital importance of hidden nodes and learning rates in optimizing BEL models. For reference, the specific parameter settings employed in our experiments are detailed in Table 2. These settings will be applied in the deployment of the models.

D. REAL-WORLD DATA

The BEL models were further tested using four real-world datasets from Kaggle [39]: (1) American Electric Power (AEP), (2) Dayton Power and Light Company (DAYTON), (3) PJM Interconnection LLC East Region (PJME), and (4) PJM Interconnection LLC West Region (PJMW). These datasets represent actual energy capacities in Megawatts (MW) and are briefly described as follows:

- **AEP Hourly**: This dataset, with 121,273 samples, presents hourly electrical power capacity data in megawatts from American Electric Power (AEP) between 2004 and 2018. It reflects AEP's total power generation capacity, crucial for evaluating their ability to meet energy demands.
- **DAYTON Hourly**: With a sample size of 121,275 samples spanning from 2004 and 2018, this dataset provides data of the electrical capacity, measured in megawatts, associated with the Dayton Power and Light Company. The dataset likely includes values recorded over a specific period. This would allow for the analysis of the company's power generation or consumption patterns and provide a better understanding of the

TABLE 9. Comparative testing performance metrics for AEP Hourly and DAYTON hourly datasets using parameters, $\%D_T$, #Hn and $\%N_{S_T}$. Note: In each cell, the upper row represents the metric value, and the lower row indicates the standard deviation.

Problems	#Hn	$\%N_{S_T}$	Testing evaluation metrics and Standard deviations (SD)								
			RMSE Training size percentage ($\%D_T$)			MAE Training size percentage ($\%D_T$)			SMAPE Training size percentage ($\%D_T$)		
Methods			30	50	75	30	50	75	30	50	75
AEP Hourly											
BEL-IK	-	-	0.03155	0.03420	0.03190	0.02035	0.02342	0.02146	6.06508	6.71404	6.30225
			1.51E-04	5.68E-03	2.23E-03	5.99E-05	5.53E-03	1.61E-03	2.25E-02	1.14E+00	2.79E-01
BEL-BEL	-	-	0.03238	0.03285	0.03199	0.02112	0.02216	0.02144	6.22022	6.45553	6.28389
			3.89E-03	4.51E-03	2.35E-03	3.94E-03	4.38E-03	1.83E-03	8.31E-01	8.97E-01	3.28E-01
B2ELM-BEL	1	5	0.02986	0.02931	0.02886	0.01977	0.01903	0.01916	6.21987	5.78361	5.79272
			5.44E-04	3.44E-04	6.45E-04	7.61E-04	4.66E-04	7.14E-04	2.69E-01	8.93E-02	1.28E-01
ELM-BEL	1	5	0.10636	0.03002	0.11394	0.09118	0.01973	0.08889	26.73370	5.98974	24.42520
			5.77E-03	8.82E-04	5.77E-03	5.10E-03	1.00E-03	4.47E-03	1.16E+00	2.24E-01	1.16E+00
B2ELM-BEL	1	100	0.02966	0.02967	0.02821	0.01943	0.01934	0.01834	6.08100	5.90077	5.64393
			4.18E-04	1.02E-03	3.18E-04	5.84E-04	1.22E-03	3.63E-04	2.04E-01	2.67E-01	7.76E-02
ELM-BEL	1	100	0.11853	0.12054	0.11509	0.10170	0.09548	0.09249	29.07200	26.28640	25.76540
			7.21E-03	6.88E-03	6.61E-03	6.30E-03	5.38E-03	5.28E-03	1.37E+00	1.37E+00	1.31E+00
B2ELM-BEL	2	5	0.02907	0.02887	0.02777	0.01877	0.01832	0.01775	5.87679	5.68427	5.52980
			1.26E-04	9.66E-05	8.49E-05	1.20E-04	6.80E-05	7.80E-05	4.99E-02	4.16E-02	6.38E-02
ELM-BEL	2	5	0.02955	0.02913	0.02803	0.01939	0.01858	0.01808	6.06598	24.315	5.66359
			5.67E-04	2.92E-04	2.48E-04	7.12E-04	3.65E-04	3.47E-04	1.96E-01	1.17E+00	1.15E-01
B2ELM-BEL	2	100	0.02894	0.02880	0.02771	0.01864	0.01828	0.01773	5.88162	5.70341	5.54565
			5.10E-05	5.41E-05	6.14E-05	5.71E-05	4.39E-05	6.52E-05	4.12E-02	5.56E-02	6.09E-02
ELM-BEL	2	100	0.02923	0.02920	0.02809	0.01885	0.01864	0.01812	5.90193	5.82493	5.66072
			2.49E-04	7.26E-04	5.24E-04	3.11E-04	8.98E-04	6.65E-04	9.36E-02	3.36E-01	2.17E-01
DAYTON Hourly											
BEL-IK	-	-	0.02873	0.02850	0.02813	0.01888	0.01888	0.01813	5.45151	5.33815	4.99379
			9.76E-05	9.10E-06	6.06E-07	1.44E-04	2.64E-05	1.40E-05	4.94E-02	7.24E-03	3.93E-03
BEL-BEL	-	-	0.02867	0.02851	0.02813	0.01876	0.01887	0.01812	5.40947	5.33542	4.98907
			3.62E-05	9.73E-06	5.56E-07	5.59E-05	2.04E-05	1.10E-05	1.84E-02	5.85E-03	3.10E-03
B2ELM-BEL	1	5	0.02665	0.02637	0.02609	0.01756	0.01693	0.01683	5.29155	4.95818	4.76548
			2.93E-04	1.99E-04	2.11E-04	4.77E-04	2.72E-04	2.98E-04	1.49E-01	7.80E-02	7.68E-02
ELM-BEL	1	5	0.08820	0.08786	0.08516	0.07293	0.07216	0.06982	20.82310	20.1092	18.80700
			5.43E-03	5.61E-03	5.33E-03	4.61E-03	4.72E-03	4.45E-03	1.10E+00	1.13E+00	1.06E+00
B2ELM-BEL	1	100	0.02646	0.02627	0.02600	0.01713	0.01671	0.01659	5.13448	4.89033	4.69667
			1.57E-04	1.48E-04	1.51E-04	2.21E-04	1.34E-04	1.45E-04	7.06E-02	4.05E-02	3.74E-02
ELM-BEL	1	100	0.09630	0.09110	0.08931	0.07966	0.07482	0.07323	22.4116	20.7367	19.60620
			6.46E-03	5.98E-03	5.83E-03	5.46E-03	5.03E-03	4.86E-03	1.28E+00	1.18736	1.14E+00
B2ELM-BEL	2	5	0.02647	0.02619	0.02585	0.01733	0.01677	0.01660	5.19875	4.88316	4.69964
			1.49E-04	1.16E-04	1.21E-04	1.44E-04	6.27E-05	9.00E-05	7.36E-02	5.74E-02	4.85E-02
ELM-BEL	2	5	0.02702	0.02665	0.02638	0.01790	0.01723	0.01710	5.36323	5.04072	4.82707
			8.53E-04	6.63E-04	6.71E-04	9.25E-04	7.66E-04	6.92E-04	2.01E-01	0.179818	1.38E-01
B2ELM-BEL	2	100	0.02626	0.02607	0.02580	0.01700	0.01660	0.01645	5.10937	4.85064	4.65335
			1.01E-04	9.22E-05	8.97E-05	1.28E-04	7.31E-05	8.03E-05	6.68E-02	5.77E-02	5.18E-02
ELM-BEL	2	100	0.02664	0.02646	0.02617	0.01727	0.01689	0.01675	5.18472	4.94651	4.74097
			2.79E-04	2.64E-04	2.58E-04	3.36E-04	3.25E-04	2.88E-04	1.43E-01	1.07E-01	8.48E-02

company's contributions to the power grid; potentially identifying trends or patterns in its energy-related activities.

- **PJME Hourly**: The dataset covers 2002 to 2018, and consists of 145,366 records of hourly power consumption in the PJM Interconnection LLC East Region, measured in megawatts (MW). Originating from PJM, a regional transmission organization in the Eastern United States, this dataset reflects the electric transmission system's capacity across several states and jurisdictions, including Delaware, Illinois, Indiana, Kentucky, Maryland, Michigan, New Jersey, North Carolina, Ohio, Pennsylvania, Tennessee, Virginia, West Virginia, and the District of Columbia.
- **PJMW Hourly**: The dataset spans from 2002 to 2018, and comprises 143,206 samples of hourly power consumption in the PJM Interconnection LLC West Region (PJMW). Measured in megawatts (MW), this data is sourced from PJM, a regional transmission organization in the Eastern Interconnection grid of the United States. Covering specific states and areas in the West Region, the dataset's scope and availability may have varied over the years due to regional changes.

We preprocessed these datasets by first removing the best straight-fit line from each using the 'detrend' function in

TABLE 10. Comparative testing performance metrics for PJME Hourly and PJMW hourly datasets using parameters, $\%D_T$, #Hn and $\%N_{S_T}$. Note: In each cell, the upper row represents the metric value, and the lower row indicates the standard deviation.

Problems	#Hn	$\%N_{S_T}$	Testing evaluation metrics and Standard deviations (SD)								
			RMSE Training size percentage ($\%D_T$)			MAE Training size percentage ($\%D_T$)			SMAPE Training size percentage ($\%D_T$)		
Methods			30	50	75	30	50	75	30	50	75
PJME Hourly											
BEL-IK	-	-	0.02500	0.02478	0.02475	0.01734	0.01679	0.01658	5.20271	5.10697	5.06589
			1.20E-04	5.39E-06	1.86E-06	1.40E-04	1.22E-05	1.77E-05	4.05E-02	3.34E-03	5.84E-03
BEL-BEL	-	-	0.02490	0.02477	0.02476	0.01719	0.01677	0.01657	5.16203	5.10176	5.06312
			4.00E-05	1.84E-06	2.05E-06	5.80E-05	1.08E-05	1.26E-05	1.45E-02	2.95E-03	4.54E-03
B2ELM-BEL	1	5	0.02281	0.02259	0.02274	0.01532	0.01458	0.01441	4.66937	4.47803	4.46179
			5.81E-04	2.35E-04	1.69E-04	7.67E-04	2.40E-04	1.34E-04	2.31E-01	4.82E-02	3.15E-02
ELM-BEL	1	5	0.07982	0.08990	0.09387	0.06519	0.07242	0.07710	19.19150	20.96900	22.37840
			4.51E-03	5.46E-03	5.91E-03	3.73E-03	4.41E-03	4.88E-03	9.53E-01	1.12E+00	1.24E+00
B2ELM-BEL	1	100	0.02249	0.02263	0.02278	0.01468	0.01468	0.01448	4.46227	4.50919	4.48246
			2.08E-04	3.46E-04	2.04E-04	2.22E-04	4.16E-04	2.05E-04	4.65E-02	1.08E-01	5.02E-02
ELM-BEL	1	100	0.08899	0.09281	0.09268	0.07265	0.07477	0.07614	21.07420	21.56220	22.13460
			5.55E-03	5.87E-03	5.84E-03	4.57E-03	4.74E-03	4.82E-03	1.14E+00	1.20E+00	1.22E+00
B2ELM-BEL	2	5	0.02288	0.02238	0.02255	0.01527	0.01451	0.01439	4.69195	4.52042	4.47955
			3.51E-04	1.25E-04	1.10E-04	3.31E-04	1.31E-04	1.04E-04	9.45E-02	6.77E-02	6.90E-02
ELM-BEL	2	5	0.02333	0.02279	0.02293	0.01579	0.01481	0.01462	4.77078	4.56557	4.54513
			1.12E-03	3.99E-04	2.85E-04	1.06E-03	3.77E-04	3.26E-04	2.27E-01	1.36E-01	1.44E-01
B2ELM-BEL	2	100	0.02228	0.02236	0.02257	0.01455	0.01454	0.01446	4.46585	4.51806	4.50856
			1.03E-04	1.03E-04	1.07E-04	1.15E-04	7.11E-05	9.01E-05	4.61E-02	5.39E-02	6.50E-02
ELM-BEL	2	100	0.02265	0.02282	0.02294	0.01486	0.01490	0.01470	4.53363	4.59834	4.57162
			3.35E-04	3.82E-04	2.80E-04	3.33E-04	3.72E-04	3.25E-04	1.20E-01	1.60E-01	1.50E-01
PJMW Hourly											
BEL-IK	-	-	0.02330	0.02253	0.02299	0.01597	0.01529	0.01503	2.96996	2.81499	2.71233
			8.78E-04	1.16E-04	1.51E-05	8.20E-04	1.31E-04	3.72E-05	1.60E-01	2.55E-02	7.95E-03
BEL-BEL	-	-	0.02266	0.02248	0.02300	0.01525	0.01516	0.01501	2.82661	2.79114	2.70714
			3.87E-04	3.70E-05	1.47E-05	3.94E-04	5.19E-05	3.29E-05	7.88E-02	1.03E-02	7.09E-03
B2ELM-BEL	1	5	0.02084	0.02064	0.02124	0.01397	0.01360	0.01356	2.60179	2.50007	2.44156
			3.12E-04	2.10E-04	2.03E-04	4.53E-04	2.76E-04	2.73E-04	8.03E-02	4.63E-02	4.18E-02
ELM-BEL	1	5	0.06594	0.06607	0.06955	0.05395	0.05367	0.05602	9.90054	9.69519	9.86733
			5.97E-03	5.96E-03	6.33E-03	4.94E-03	4.89E-03	5.15E-03	8.74E-01	8.62E-01	8.93E-01
B2ELM-BEL	1	100	0.02061	0.02053	0.02115	0.01356	0.01338	0.01340	2.52425	2.46161	2.41513
			2.58E-04	8.49E-05	9.84E-05	3.43E-04	1.34E-04	1.41E-04	5.88E-02	2.05E-02	2.00E-02
ELM-BEL	1	100	0.06662	0.06599	0.06874	0.05451	0.05361	0.05536	10.00080	9.68410	9.75183
			5.99E-03	5.94E-03	6.17E-03	4.95E-03	4.87E-03	5.02E-03	8.76E-01	8.59E-01	8.71E-01
B2ELM-BEL	2	5	0.02077	0.02054	0.02114	0.01387	0.01352	0.01349	2.58506	2.49176	2.43238
			2.73E-04	6.33E-05	4.51E-05	2.44E-04	6.85E-05	3.86E-05	4.38E-02	1.49E-02	1.50E-02
ELM-BEL	2	5	0.02097	0.02082	0.02147	0.01407	0.01376	0.01378	2.60767	2.52471	2.47302
			4.62E-04	3.84E-04	4.68E-04	4.66E-04	3.93E-04	4.65E-04	7.36E-02	6.28E-02	6.84E-02
B2ELM-BEL	2	100	0.02054	0.02042	0.02106	0.01346	0.01329	0.01331	2.50757	2.44593	2.39951
			2.73E-04	5.96E-05	4.07E-05	2.58E-04	6.93E-05	3.31E-05	4.57E-02	1.34E-02	7.37E-03
ELM-BEL	2	100	0.02063	0.02062	0.02130	0.01357	0.01346	0.01355	2.52117	2.47345	2.43476
			2.30E-04	2.12E-04	3.12E-04	2.61E-04	2.34E-04	3.39E-04	4.30E-02	3.44E-02	4.53E-02

MATLAB version R2022a. Then, we rescaled the datasets to fit within the range [0,1]. Finally, we slide the data with a specified window size of 2. The examples of each dataset are illustrated in Fig. 14.

Each ELM-based model was designed with a simplified architecture featuring two specific hidden nodes, denoted as #Hn=1 and #Hn=2. The experiments were conducted using a training size, a percentage of knowledge transfer learning, and other essential parameters consistent with those listed in Table 2.

E. EXPERIMENTAL RESULTS

In a comprehensive analysis of various BEL models applied to time series data, a fascinating pattern emerged, particularly when comparing B2ELM-BEL, ELM-BEL, BEL-BEL and BEL-IK. The study, which scrutinized the performance across chaotic time series benchmarking and real-world energy consumption datasets, brought to light a detailed picture of how these models stack up against each other, offering valuable insights for practitioners in selecting the appropriate model based on the dataset and desired outcomes.

Central to this evaluation were the testing performance metrics, namely Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Symmetric Mean Absolute Percentage Error (SMAPE), alongside their standard deviations (SD). These metrics are pivotal in gauging the accuracy and consistency of the models, with lower values indicating superior performance and reliability.

TABLE 11. Summary of testing performance metrics.

Data Category	Method	#Hn	$\%N_{S_T}$	Total Score (Lower is better)	Position (Lower is better)
Chaotic Time Series Benchmarking	BEL-IK	-	-	484	7
	BEL-BEL	-	-	489	8
	B2ELM-BEL	1	5	362	5
	ELM-BEL	1	5	570	10
	B2ELM-BEL	1	100	372	6
	ELM-BEL	1	100	558	9
	B2ELM-BEL	5	5	153.5	2
	ELM-BEL	5	5	204.5	4
	B2ELM-BEL	5	100	109	1
	ELM-BEL	5	100	163	3
Real-World Energy Consumption	BEL-IK	-	-	279.5	8
	BEL-BEL	-	-	261.5	7
	B2ELM-BEL	1	5	163	5
	ELM-BEL	1	5	326	9
	B2ELM-BEL	1	100	112	3
	ELM-BEL	1	100	351	10
	B2ELM-BEL	2	5	96.5	2
	ELM-BEL	2	5	195	6
	B2ELM-BEL	2	100	50.5	1
	ELM-BEL	2	100	145	4

1) SEVEN CHAOTIC TIME SERIES BENCHMARK EXPERIMENTAL RESULTS

In the context of time scales for comparative methods, Tables 3-4 show that BEL-IK has the shortest training and testing times. Following BEL-IK, the BEL-BEL model ranks second in speed, while ELM-BEL is the third fastest, and B2ELM-BEL is the slowest. The extended training time for B2ELM-BEL and ELM-BEL, in comparison to the others, can be attributed to their requirement to train N_e Biased-ELM models. This process is necessary to select the optimal Biased-ELM model for transfer knowledge in BEL. In terms of testing times, ELM-BEL outperforms B2ELM-BEL. This difference arises because the B2ELM-BEL model includes a biased term, which impacts its testing speed.

The experimental results for the chaotic time series benchmarks are extensively documented in Tables 5-6. These tables provide a thorough analysis of RMSEs, MAEs, and SMAPEs for each testing evaluation metric. To better clarify the assessment of testing performance across various problems and methodologies, Standard Deviations (SDs) for the testing evaluation metrics are presented in Table 7 and Table 8. These tables provide additional insights into the variability of standard deviations (SDs) across different scenarios and experimental setups. Moreover, tables mentioned provide a report on testing evaluation metrics. These tables furnish specific information regarding the number of hidden nodes (#Hn) and the training size percentage ($\%N_{S_T}$). Notably, #Hn represents the number of hidden nodes, while $\%N_{S_T}$ denotes the training size percentage used to specify the transfer learning size.

Utilizing seven chaotic time-series datasets, Fig. 19 features the Critical Difference (CD) diagram, where the average ranks of various methods are depicted on the x-axis. In this diagram, lower ranks indicate superior performance, while higher ranks point to less favorable outcomes. The CD diagram distinctly illustrates that the B2ELM-BEL model with five hidden nodes (#Hn=5) surpasses other models in its category in terms of overall predictive performance. Notably, the position of the B2ELM-BEL with one hidden node (#Hn=1) in the middle of the CD diagram implies its moderate effectiveness based on the chosen performance metrics, without exhibiting extreme performance in either direction. However, in direct comparison to the baseline BEL-IK method, the B2ELM-BEL model with one hidden node #Hn=1) demonstrates superior performance in all metrics, including RMSE, MAE, and SMAPE.

Table 11 shows a comprehensive summary of testing performance metrics, denoted by Total Score and Position. In the chaotic time series benchmarking datasets, the performance of various models displayed distinct variations. The standard BEL-IK model ranked seventh with a total score of 484, while its variant, BEL-BEL, placed slightly lower at eighth with a score of 489.

The experimental results from Figs. 15e, 16e, 17e, and 18e demonstrate that the B2ELM-BEL model, when equipped with five hidden nodes (#Hn=5), significantly outperforms its counterpart with only one hidden node (#Hn=1), particularly in complex scenarios such as the Henon problem. This is especially noticeable in the Henon case, where the one-hidden-node model shows notably weaker performance, highlighting its limitations in handling intricate challenges. Consequently, although the B2ELM-BEL model with one hidden node is functional for simpler tasks, it falls short in more complex situations, underscoring the need to increase the number of hidden nodes beyond one.

In conclusion, the B2ELM-BEL model with one hidden node (#Hn=1) and a 5% transfer learning size ($\%N_{S_T}$) secured fifth position with a score of 362. This model excelled further with five hidden nodes (#Hn=5) and the same transfer

TABLE 12. Performance rankings of machine learning methods for Exptchao, Exptper and Exptqp2 chaotic time-series.

Problems	Methods	#Hn	$\%N_{S_T}$	RMSE $\%D_T$ 30	RMSE $\%D_T$ 50	RMSE $\%D_T$ 75	MAE $\%D_T$ 30	MAE $\%D_T$ 50	MAE $\%D_T$ 75	SMAPE $\%D_T$ 30	SMAPE $\%D_T$ 50	SMAPE $\%D_T$ 75
Exptchao	BEL-IK	-	-	8	8	7.5	8	8	7	8	8	7
	BEL-BEL	-	-	7	7	7.5	7	7	8	7	7	8
	B2ELM-BEL	1	5	6	6	5	6	6	5	6	6	6
	ELM-BEL	1	5	10	10	10	10	10	10	10	10	10
	B2ELM-BEL	1	100	5	5	6	5	5	6	5	5	5
	ELM-BEL	1	100	9	9	9	9	9	9	9	9	9
	B2ELM-BEL	5	5	1	3	3	1	3	3	1	3	3
	ELM-BEL	5	5	4	4	4	4	4	3	4	4	4
	B2ELM-BEL	5	100	3	1	1	2	1.5	1	2	1	1
	ELM-BEL	5	100	2	2	2	3	1.5	2	4	2	2
Exptper	BEL-IK	-	-	7.5	7	7.5	7.5	7.5	7.5	6	7	7
	BEL-BEL	-	-	7.5	8	7.5	7.5	7.5	7.5	7	6	6
	B2ELM-BEL	1	5	6	6	5	6	6	5	8	8	5
	ELM-BEL	1	5	9	10	10	9	10	10	9	10	10
	B2ELM-BEL	1	100	5	5	6	5	5	6	5	5	8
	ELM-BEL	1	100	10	9	9	10	9	9	10	9	9
	B2ELM-BEL	5	5	2.5	2	2.5	2.5	1.5	1.5	4	4	4
	ELM-BEL	5	5	2.5	2	2.5	2.5	1.5	1.5	1	3	3
	B2ELM-BEL	5	100	2.5	2	2.5	2.5	3.5	3.5	3	1	1
	ELM-BEL	5	100	2.5	4	2.5	2.5	3.5	3.5	2	2	2
Exptqp2	BEL-IK	-	-	7	7.5	7.5	7	7.5	7.5	8	8	7
	BEL-BEL	-	-	8	7.5	7.5	8	7.5	7.5	7	7	8
	B2ELM-BEL	1	5	5	5	5	5	5	5	6	5	5
	ELM-BEL	1	5	9	10	10	9	10	10	9	10	10
	B2ELM-BEL	1	100	6	6	6	6	6	6	6	5	6
	ELM-BEL	1	100	10	9	9	10	9	9	10	9	9
	B2ELM-BEL	5	5	1.5	3	3	1	3	1.5	1	2	4
	ELM-BEL	5	5	1.5	3	3	3	3	3.5	3	4	3
	B2ELM-BEL	5	100	3.5	3	1	3	3	1.5	2	3	1
	ELM-BEL	5	100	3.5	1	3	3	1	3.5	4	1	2
Exptqp3	BEL-IK	-	-	7	7.5	7	7	7.5	7	7	8	7
	BEL-BEL	-	-	8	7.5	8	8	7.5	8	8	7	8
	B2ELM-BEL	1	5	6	5	6	6	5	6	6	5	6
	ELM-BEL	1	5	10	9	10	10	9	10	10	9	10
	B2ELM-BEL	1	100	5	6	5	5	6	5	5	6	5
	ELM-BEL	1	100	9	10	9	9	10	9	9	10	9
	B2ELM-BEL	5	5	1.5	3	3.5	1.5	3	4	1	3	3
	ELM-BEL	5	5	3	4	3.5	3.5	4	3	4	4	4
	B2ELM-BEL	5	100	1.5	1	1	1.5	1	1	2	1	1
	ELM-BEL	5	100	4	2	2	3.5	2	2	3	2	2

learning size, achieving the first runner-up position with a score of 153.5.

The ELM-BEL model, on the other hand, trailed in the bottom, securing the tenth position with a score of 570 when set at one hidden node (#Hn=1). However, it reached its best ranking, third place, with a score of 163 when enhanced to five hidden nodes (#Hn=5) and a 100% transfer learning size ($\%N_{S_T}$). These findings underscore the importance of a higher number of hidden nodes for effectively managing challenging datasets, and demonstrate that a single-node ELM-BEL model is not viable in such contexts.

2) REAL-WORLD DATA EXPERIMENTAL RESULTS

The experimental results based on the real-world data, meticulously detailed in Tables 9-10, are complemented by summaries of RMSEs, MAEs, SMAPEs and SDs of each testing evaluation metrics. Figure. 20 displays the performance, summarized by SMAPEs and SDs.

Table 11 offers a detailed overview of the testing performance metrics, encompassing both Total Score and Position. The ranking results are not much different in the real-world energy consumption datasets. Here, BEL-IK and BEL-BEL occupied the lower tiers, ranking eighth and seventh, with scores of 279.5 and 261.5, respectively. B2ELM-BEL emerged as a clear front-runner, especially with 2 #Hn and 5 $\%N_{S_T}$, achieving the first runner-up position with a score of 50.5. It consistently outperformed BEL-IK in various combinations of $\%\mathcal{D}_T$, $\%N_{S_T}$, and #Hn, underlining its superior predictive accuracy and reliability across different datasets. ELM-BEL, although displaying inconsistency,

TABLE 13. Performance rankings of machine learning methods for Henon, Lorenz and Rossler chaotic time-series.

Problems	Methods	#Hn	$\%N_{S_T}$	Testing Performance Metrics Ranking at Different Training size percentages($\%D_T$)								
				RMSE $\%D_T$			MAE $\%D_T$			SMAPE $\%D_T$		
				30	50	75	30	50	75	30	50	75
Henon	BEL-IK	-	-	9	9	9	9	9	9	9	9	9
	BEL-BEL	-	-	10	10	10	10	10	10	10	10	10
	B2ELM-BEL	1	5	6	7	7	6	7	7	6	7	7
	ELM-BEL	1	5	7	6	5	7	6	5	7	6	5
	B2ELM-BEL	1	100	8	8	8	8	8	8	8	8	8
	ELM-BEL	1	100	5	5	6	5	5	6	5	5	6
	B2ELM-BEL	5	5	1	2	2	1	2	2	1	2	2
	ELM-BEL	5	5	4	3	3	4	3	3	3	3	3
	B2ELM-BEL	5	100	2	1	1	2	1	1	2	1	1
	ELM-BEL	5	100	3	4	4	3	4	4	4	4	4
Lorenz	BEL-IK	-	-	8	8	8	8	8	8	8	8	8
	BEL-BEL	-	-	7	7	7	7	7	7	7	7	7
	B2ELM-BEL	1	5	5	6	6	5	6	6	5	6	6
	ELM-BEL	1	5	9	9	9	9	9	9	9	9	9
	B2ELM-BEL	1	100	6	5	5	6	5	5	6	5	5
	ELM-BEL	1	100	10	10	10	10	10	10	10	10	10
	B2ELM-BEL	5	5	4	2	2	4	3.5	2	3	3	3
	ELM-BEL	5	5	3	3.5	3.5	2	3.5	3.5	4	4	4
	B2ELM-BEL	5	100	2	1	1	3	1	1	1	1	1
	ELM-BEL	5	100	1	3.5	3.5	1	2	3.5	2	2	2
Rossler	BEL-IK	-	-	7.5	7.5	8	7.5	7	7.5	7	7	6
	BEL-BEL	-	-	7.5	7.5	7	7.5	8	7.5	8	8	7
	B2ELM-BEL	1	5	6	5	5	5	5	6	5	5	
	ELM-BEL	1	5	10	9	9	10	9	9	10	9	9
	B2ELM-BEL	1	100	5	6	6	6	6	6	5	6	8
	ELM-BEL	1	100	9	10	10	9	10	10	9	10	10
	B2ELM-BEL	5	5	1	2	2	1	4	4	1	4	4
	ELM-BEL	5	5	3.5	4	4	3.5	3	3	4	3	2
	B2ELM-BEL	5	100	2	1	1	2	2	2	3	2	3
	ELM-BEL	5	100	3.5	3	3	3.5	1	1	2	1	1

showed potential in specific settings, particularly with 2 #Hn and 100 $\%N_{S_T}$, where it ranked fourth with a score of 145.

From these results, it is evident that B2ELM-BEL, with its varying configurations, demonstrated a significant advantage over BEL-IK across diverse datasets, consistently recording lower RMSE, MAE, SMAPE, and SD. This underlines its superior predictive accuracy and performance under these settings. ELM-BEL's performance, while varied, suggested its effectiveness under certain conditions, particularly with higher $\%N_{S_T}$ and #Hn.

In contrast, the BEL-BEL model, when compared with BEL-IK, did not show a marked statistical advantage. The performance metrics across datasets for these models were closely matched, suggesting similar levels of accuracy and reliability. The choice between BEL-BEL and BEL-IK would likely hinge on factors beyond just the basic error metrics, such as computational demands or specific application requirements.

VI. CONCLUSION

Our extensive analysis of time series forecasting has revealed insights regarding the performance of various predictive models, with a particular emphasis on the BEL models. This conclusion synthesizes the key findings of our study, offering a comprehensive view of the capabilities and potential applications of these models within the broader context of machine learning and predictive analysis.

The paper introduces the Bias-Boosted Extreme Learning Machine-Guided Brain Emotional Learning (B2ELM-BEL) model, marking significant advancements in chaotic time series prediction and machine learning. A notable contribution of this research is the innovative integration of the existing BEL-IK model with the ELM method. This integration effectively replaces 'emotion' with 'experience,' where the ELM functions as a knowledge transfer unit. Such a unique combination signifies a considerable advancement, offering a novel perspective in approaching chaotic time series prediction.

Another key aspect of this research is the introduction of a bias term into ELM output weights. This modification is a minor tweak and a fundamental enhancement that significantly improves the model's predictive accuracy. It distinguishes the B2ELM-BEL model from traditional

TABLE 14. **Performance rankings of machine learning methods for real-world energy consumption at different training sizes.**

Problems	Methods	#Hn	$\%N_{S_T}$	Testing Performance Metrics Ranking at Different Training size percentages($\%D_T$)								
				RMSE $\%D_T$			MAE $\%D_T$			SMAPE $\%D_T$		
				30	50	75	30	50	75	30	50	75
AEP Hourly	BEL-IK	-	-	7	9	7	7	9	8	4	8	8
	BEL-BEL	-	-	8	8	8	8	8	7	8	7	7
	B2ELM-BEL	1	5	6	5	6	6	5	6	7	3	6
	ELM-BEL	1	5	9	7	9	9	7	9	9	6	9
	B2ELM-BEL	1	100	5	6	5	5	6	5	6	5	3
	ELM-BEL	1	100	10	10	10	10	10	10	10	10	10
	B2ELM-BEL	2	5	2	2	2	2	2	2	1	1	1
	ELM-BEL	2	5	4	3	3	4	3	3	5	9	5
	B2ELM-BEL	2	100	1	1	1	1	1	1	2	2	2
	ELM-BEL	2	100	3	4	4	3	4	4	3	4	4
Dayton Hourly	BEL-IK	-	-	8	7	7.5	8	8	8	8	8	8
	BEL-BEL	-	-	7	8	7.5	7	7	7	7	7	7
	B2ELM-BEL	1	5	5	4	4	5	5	5	5	5	5
	ELM-BEL	1	5	9	9	9	9	9	9	9	9	9
	B2ELM-BEL	1	100	2	3	3	2	2	2	2	3	2
	ELM-BEL	1	100	10	10	10	10	10	10	10	10	10
	B2ELM-BEL	2	5	3	2	2	4	3	3	4	2	3
	ELM-BEL	2	5	6	6	6	6	6	6	6	6	6
	B2ELM-BEL	2	100	1	1	1	1	1	1	1	1	1
	ELM-BEL	2	100	4	5	5	3	4	4	3	4	4
PJME Hourly	BEL-IK	-	-	8	8	7	8	8	8	8	8	8
	BEL-BEL	-	-	7	7	8	7	7	7	7	7	7
	B2ELM-BEL	1	5	4	3	3	5	3	2	4	1	1
	ELM-BEL	1	5	9	9	10	9	9	10	9	9	10
	B2ELM-BEL	1	100	2	4	4	2	4	4	1	2	3
	ELM-BEL	1	100	10	10	9	10	10	9	10	10	9
	B2ELM-BEL	2	5	5	2	1	4	1	1	5	4	2
	ELM-BEL	2	5	6	5	5	6	5	5	6	5	5
	B2ELM-BEL	2	100	1	1	2	1	2	3	2	3	4
	ELM-BEL	2	100	3	6	6	3	6	6	3	6	6
PJMW Hourly	BEL-IK	-	-	8	8	7	8	8	8	8	8	8
	BEL-BEL	-	-	7	7	8	7	7	7	7	7	7
	B2ELM-BEL	1	5	5	5	4	5	5	5	5	5	5
	ELM-BEL	1	5	9	10	10	9	10	10	9	10	10
	B2ELM-BEL	1	100	2	1	3	2	2	2	3	2	2
	ELM-BEL	1	100	10	9	9	10	9	9	10	9	9
	B2ELM-BEL	2	5	4	2.5	2	4	4	3	4	4	3
	ELM-BEL	2	5	6	6	6	6	6	6	6	6	6
	B2ELM-BEL	2	100	1	2.5	1	1	1	1	1	1	1
	ELM-BEL	2	100	3	4	5	3	3	4	2	3	4

approaches, adding a unique feature that bolsters its performance.

Compared to BEL-IK, the B2ELM-BEL model consistently demonstrates superior accuracy, as indicated by its lower RMSE, MAE, SMAPE, and SD scores. For instance, in the chaotic time series benchmarking, B2ELM-BEL notably outperformed BEL-IK, showcasing its robustness in handling complex predictive tasks. The B2ELM-BEL model's effectiveness, even with fewer hidden nodes and a lower percentage of knowledge transfer data size, represents a significant advantage, particularly in scenarios requiring computational efficiency.

This attribute reflects the model's sophisticated design, enabling it to maintain high-performance levels despite reduced computational complexity. Our evaluations across various benchmarks and real-world scenarios affirm the model's versatility and robustness. B2ELM-BEL has consistently proven its superiority over other BEL models, making it a preferred choice for diverse forecasting applications.

The performance of the ELM-BEL model, which operates without a bias term in the ELM, varies but demonstrates effectiveness in specific settings. This is particularly evident when the model is configured with higher $\%N_{S_T}$ and #Hn, highlighting its adaptability and potential effectiveness in certain scenarios. This finding suggests that ELM-BEL could be highly effective for specific applications, necessitating a tailored approach based on the particular requirements of the task.

The comparison between the BEL-BEL and BEL-IK models reveals no distinct statistical advantage. With similar performance levels, BEL-BEL requires more computational effort than BEL-IK.

Looking ahead, the success of B2ELM-BEL opens numerous avenues for further research and development. Potential focus areas include enhancing the model's algorithmic efficiency, exploring its applicability to a broader spectrum of data types, and conducting comparative studies with other advanced models. These endeavors aim to amplify the model's capabilities and extend its utility in the rapidly evolving landscape of data science and machine learning.

In summary, the B2ELM-BEL model, with its established strengths in accuracy and adaptability, emerges as a potent tool in the field of chaotic time series prediction. Its potential for further development and varied applications holds great promise for advancing methodologies in time series forecasting.

APPENDIX

TABLES

See Tables 12–14.

REFERENCES

[1] C. Ai, S. He, X. Fan, and W. Wang, "Chaotic time series wind power prediction method based on OVMD-PE and improved multi-objective state transition algorithm," *Energy*, vol. 278, Sep. 2023, Art. no. 127695.

[2] Y. Tang, Z. Song, Y. Zhu, H. Yuan, M. Hou, J. Ji, C. Tang, and J. Li, "A survey on machine learning models for financial time series forecasting," *Neurocomputing*, vol. 512, pp. 363–380, Nov. 2022.

[3] D. M. Durairaj and B. H. K. Mohan, "A convolutional neural network based approach to financial time series prediction," *Neural Comput. Appl.*, vol. 34, no. 16, pp. 13319–13337, Aug. 2022.

[4] V. Pérez-Muñuzuri, "Forecasting of chaotic cloud absorption time series for meteorological and plume dispersion modeling," *J. Appl. Meteorol.*, vol. 37, no. 11, pp. 1430–1443, Nov. 1998.

[5] X. Wang and M. Han, "Improved extreme learning machine for multivariate time series online sequential prediction," *Eng. Appl. Artif. Intell.*, vol. 40, pp. 28–36, Apr. 2015.

[6] H. V. Dudukcu, M. Taskiran, Z. G. Cam Taskiran, and T. Yildirim, "Temporal convolutional networks with RNN approach for chaotic time series prediction," *Appl. Soft Comput.*, vol. 133, Jan. 2023, Art. no. 109945.

[7] B. Li and S. Rangarajan, "A conceptual study of transfer learning with linear models for data-driven property prediction," *Comput. Chem. Eng.*, vol. 157, Jan. 2022, Art. no. 107599.

[8] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[9] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," 2019, *arXiv:1911.02685*.

[10] D. Obst, B. Ghattas, S. Claudel, J. Cugliari, Y. Goude, and G. Oppenheim, "Improved linear regression prediction by transfer learning," *Comput. Statist. Data Anal.*, vol. 174, Oct. 2022, Art. no. 107499.

[11] S. M. Salaken, A. Khosravi, T. Nguyen, and S. Nahavandi, "Extreme learning machine based transfer learning algorithms: A survey," *Neurocomputing*, vol. 267, pp. 516–524, Dec. 2017.

[12] A. Khashman, "A modified backpropagation learning algorithm with added emotional coefficients," *IEEE Trans. Neural Netw.*, vol. 19, no. 11, pp. 1896–1909, Nov. 2008.

[13] J. Morn and C. Balkenius, "A computational model of emotional learning in the amygdala," in *Proc. 6th Int. Conf. Simulation Adapt. Behav. (From Animals Animats)*, 2000, pp. 1–9.

[14] M. A. Sharbafi, C. Lucas, and R. Daneshvar, "Motion control of omni-directional three-wheel robots by brain-emotional-learning-based intelligent controller," *IEEE Trans. Syst. Man, Cybern. C, Appl. Rev.*, vol. 40, no. 6, pp. 630–638, Nov. 2010, doi: 10.1109/TSMCC.2010.2049104.

[15] E. Lotfi and M.-R. Akbarzadeh-T., "Adaptive brain emotional decayed learning for online prediction of geomagnetic activity indices," *Neurocomputing*, vol. 126, pp. 188–196, Feb. 2014, doi: 10.1016/j.neucom.2013.02.040.

[16] H. S. A. Milad, U. Farooq, M. E. El-Hawary, and M. U. Asad, "Neo-fuzzy integrated adaptive decayed brain emotional learning network for online time series prediction," *IEEE Access*, vol. 5, pp. 1037–1049, 2017.

[17] Y. Sharafi, S. Setayeshi, and A. Falahiazar, "An improved model of brain emotional learning algorithm based on interval knowledge," *J. Math. Comput. Sci.*, vol. 14, no. 1, pp. 42–53, Jan. 2015.

[18] M. Sharma and A. Kumar, "Performance comparison of brain emotional learning-based intelligent controller (BELBIC) and PI controller for continually stirred tank heater (CSTH)," in *Computational Advancement in Communication Circuits and Systems*, vol. 335, 2015, pp. 293–301.

[19] M.-H. Khooban and R. Javidan, "A novel control strategy for DVR: Optimal bi-objective structure emotional learning," *Int. J. Electr. Power Energy Syst.*, vol. 83, pp. 259–269, Dec. 2016.

[20] Q. Wu, C.-M. Lin, W. Fang, F. Chao, L. Yang, C. Shang, and C. Zhou, "Self-organizing brain emotional learning controller network for intelligent control system of mobile robots," *IEEE Access*, vol. 6, pp. 59096–59108, 2018.

[21] S. H. Fakhrmoosavy, S. Setayeshi, and A. Sharifi, "A modified brain emotional learning model for earthquake magnitude and fear prediction," *Eng. With Comput.*, vol. 34, no. 2, pp. 261–276, Apr. 2018.

[22] M. Affan and R. Uddin, "Brain emotional learning and adaptive model predictive controller for induction motor drive: A new cascaded vector control topology," *Int. J. Control, Autom. Syst.*, vol. 19, no. 9, pp. 3122–3135, Sep. 2021.

[23] C. Jia, D. Kong, and L. Du, "Recursive terminal sliding-mode control method for nonlinear system based on double hidden layer fuzzy emotional recurrent neural network," *IEEE Access*, vol. 10, pp. 118012–118023, 2022.

[24] B. Debnath and S. J. Mija, "Adaptive emotional-learning-based controller: A practical design approach for helicopters with variable speed rotors," *IEEE Trans. Ind. Informat.*, vol. 18, no. 2, pp. 1132–1141, Feb. 2022.

[25] M. A. Zare, R. Boostani, M. Mohammadi, and S. Kouchaki, "A dopamine based adaptive emotional neural network," *IEEE Access*, vol. 10, pp. 109460–109475, 2022.

[26] Z. Wang, X. Wang, and Z. Zeng, "Memristive circuit design of brain-like emotional learning and generation," *IEEE Trans. Cybern.*, vol. 53, no. 1, pp. 222–235, Jan. 2023.

[27] C. Woese. *Desert Elephants Pass on Knowledgenot Mutationsto Survive*. [Online]. Available: https://phys.org/news/2016-08-elephants-knowledgenot-mutationsto-survive.html

[28] Y. Ishida, P. J. Van Coeverden de Groot, K. E. A. Leggett, A. S. Putnam, V. E. Fox, J. Lai, P. T. Boag, N. J. Georgiadis, and A. L. Roca, "Genetic connectivity across marginal habitats: The elephants of the Namib desert," *Ecol. Evol.*, vol. 6, no. 17, pp. 6189–6201, Sep. 2016, doi: 10.1002/ece3.2352.

[29] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, nos. 1–3, pp. 489–501, Dec. 2006.

[30] G.-B. Huang, L. Chen, and C.-K. Siew, "Universal approximation using incremental constructive feedforward networks with random hidden nodes," *IEEE Trans. Neural Netw.*, vol. 17, no. 4, pp. 879–892, Jul. 2006.

[31] G. B. Huang, N. Y. Liang, H. J. Rong, P. Saratchandran, and N. Sundararajan, "On-line sequential extreme learning machine," in *Proc. IASTED Int. Conf. Comput. Intell.*, Calgary, AB, Canada, Jul. 2005, pp. 232–237.

[32] J. Wang, S. Lu, S.-H. Wang, and Y.-D. Zhang, "A review on extreme learning machine," *Multimedia Tools Appl.*, vol. 81, no. 29, pp. 41611–41660, Dec. 2022, doi: 10.1007/s11042-021-11007-7.

[33] D. M. Isaacowitz, K. M. Livingstone, and V. L. Castro, "Aging and emotions: Experience, regulation, and perception," *Current Opinion Psychol.*, vol. 17, pp. 79–83, Oct. 2017.

[34] A. S. Morris, J. S. Silk, L. Steinberg, S. S. Myers, and L. R. Robinson, "The role of the family context in the development of emotion regulation," *Social Develop.*, vol. 16, no. 2, pp. 361–388, May 2007, doi: 10.1111/j.1467-9507.2007.00389.x.

[35] P. Vincent and Y. Bengio, "Kernel matching pursuit," *Mach. Learn.*, vol. 48, pp. 169–191, 2002.

[36] T.-Y. Kwok and D.-Y. Yeung, "Use of bias term in projection pursuit learning improves approximation and convergence properties," *IEEE Trans. Neural Netw.*, vol. 7, no. 5, pp. 1168–1183, Sep. 1996.
[37] G. Golub and C. Van Loan, *Matrix Computations*. Baltimore, MD, USA: JHU Press, 2013.
[38] E. Weeks. (1976). *My Adventures in Chaotic Time Series Analysis*. [Online]. Available: https://physics.emory.edu/faculty/weeks/research/tseries0a.html
[39] R. Mulla. (2019). *Hourly Energy Consumption*. [Online]. Available: https://www.kaggle.com/datasets/robikscube/hourly-energy-consumption/

SUTHASINEE IAMSA-AT received the B.S. degree in information technology from the Suranaree University of Technology, Nakhon Ratchasima, Thailand, and the M.S. degree in computer science from Khon Kaen University, Thailand, where she is currently pursuing the Ph.D. degree in information technology with the Advanced Smart Computing Laboratory. Her research interests include machine learning, artificial intelligence, neural networks, pattern recognition, and intelligent information processing.

PUNYAPHOL HORATA received the B.S. degree in mathematics from Khon Kaen University, Thailand, the M.S. degree in computer science from Chulalongkorn University, Thailand, and the Ph.D. degree in computer science from Khon Kaen University. He is currently an Associate Professor with Khon Kaen University. His research interests include machine learning, soft computing, and pattern recognition.

KHAMRON SUNAT (Associate Member, IEEE) received the degree in chemical engineering, the M.S. degree in computational science, and the Ph.D. degree in computer science from Chulalongkorn University, in 1989, 1998, and 2004, respectively. He is currently an Assistant Professor with the Department of Computer Science, Khon Kaen University, Thailand, and joined the Research Group in Intelligence System and Machine Learning. His research interests include neural networks, pattern recognition, computer vision, soft computing, fuzzy systems, evolutionary computing and learning, and optical computing.