A Dynamic Ensemble and Replaying Model for Online Marine Sensor Data Prediction

Xiang Li[1,2] ✉, Xi Fu[1,2], Congqi Lin[3], Xiangkai Wang[1,2], Yuhang Zhang[1,2], Hao Wang[1,2], Zhigang Zhao[1,2] ✉, Meihong Yang[1,2], and Yinglong Wang[1,2]

[1] Key Laboratory of Computing Power Network and Information Security, Ministry of Education, Shandong Computer Science Center (National Supercomputer Center in Jinan), Qilu University of Technology (Shandong Academy of Sciences), Jinan, China.
[2] Shandong Provincial Key Laboratory of Computing Power Internet and Service Computing, Shandong Fundamental Research Center for Computer Science, Jinan, China.
[3] Johns Hopkins University, Baltimore, USA
{xiangli,zhaozhg}@sdas.org

Abstract. Deep learning excels in time-series data mining, yet offline-trained models often degrade when faced with dynamic marine observation data. To address this, we propose a brain-inspired online learning and replay framework for efficient marine time-series data prediction. The proposed framework tackles concept drift not only by updating the parameters of its internal modules but also by employing an attention mechanism to adaptively assign importance to these modules, and incorporating a neuroscience-inspired memory replay mechanism for reinforcing past knowledge. Unlike traditional deep learning models reliant on extensive historical data, our framework enables cold-start learning and inference, making it ideal for environmental monitoring stations with limited data where offline models struggle to generalize. We further introduce the first marine data prediction benchmark dataset MarineDrift-1.0, covering key marine environmental indicators with natural conecpt drift. Experiments on this dataset demonstrate the model's superior performance over state-of-the-art methods. Notably, the framework is model-independent, allows seamless integration with various models, delivering strong results even with simple architectures.

Keywords: ocean · time series · concept drift · online deep learning · variational auto-encoders

1 Introduction

In recent years, the rapid expansion of Internet of Things (IoT) systems has driven the real-time generation of massive time-series data from sensors and devices, particularly in environmental monitoring such as ocean observation. Ocean buoys equipped with multi-parameter sensors continuously collect critical marine data including water temperature, salinity, wind speed, and pollution levels. Forecasting such data provides technical support for monitoring dynamic

marine ecosystem evolutions, early environmental risk warnings, and scientific decision-making.

Though Deep learning (DL) models have demonstrated effectiveness in time series data forecasting[1,2]. These models require sufficient training data and assume static input-output relationships. In contrast, marine sensor time series data often exhibits fluctuating patterns with evolving underlying distributions (Concept Drift or Distribution Shift) [3,4,5]. As shown in Figure 1, a marine sensor time series consists of segments with distinct distributions. The drift disrupts the assumption of stationarity, Thus current Deep Learning approaches face the challenge of **Stability-Plasticity Dilemma** [6]. As data evolves unpredictably, static forecasting models experience a decline in performance.

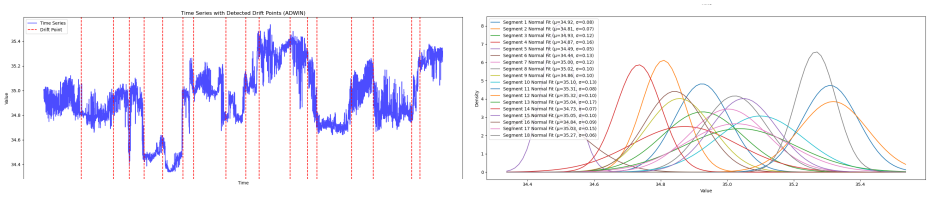

Fig. 1. Conceptual drift occurs in a marine time-series data stream, which can be divided into 18 distinct segments with ADWIN algorithm, each exhibiting a different distribution.

This challenge has sparked extensive research into methods for addressing concept drift in time series analysis [7]. There are two distinct but interconnected approaches: Online Learning (**active mode**) and Incremental Learning (**lazy mode**). Online learning relies on scalable and efficient algorithms to sequentially process training instances from a data stream, one by one, to learn the model [8,9,10]. On the other hand, incremental learning updates the model when a batch of data instances arrives[11]. In this paper, we focus methods in the context of active mode. There is still room for improvement in active mode methods specific to marine time series data, the motivation of this work is as follows:

- **Motivation 1 (Enhancing Model Adaptability):** Existing online models primarily adjust hidden layer parameters while overlooking structural adaptation. Given numerous marine variables and monitoring scenarios, customized model designs are impractical, necessitating self-evolving architectures that autonomously balance model capacity with environmental demands.
- **Motivation 2 (Catastrophic Forgetting Prevention):** MLP-based models inherently suffer from catastrophic forgetting [12], where new knowledge acquisition overwrites previous patterns [13]. Many existing approaches rely on storing and ensembling historical models to incorporate past knowledge. However [14], storing intermediate models can quickly become unmanageable on resource-constrained marine observation devices.
- **Motivation 3 (Cold Start Learning):** In under-monitored environments, the lack of sufficient historical data hampers the development of deep learning (DL)-

based online learning models, as they typically require pre-existing data to warm up the model, followed by online learning and inference on the data stream (e.g., FSNet [15] and OneNet [16]). We prefer a 'Cold Start' learning paradigm that can be applied directly to the stream, quickly adapting to target time series data without initial training.

To address these challenges, we propose a **B**rain-inspired **R**eplay **A**daptive **I**ncremental **N**etwork built on a **V**ariational **A**utoencoder (BRAIN-VAE). The model integrates two neuro-inspired strategies: A modular attention mechanism that emulates functional specialization in brain regions, where distinct neural modules dynamically reconfigure their contributions for specific temporal patterns, akin to how visual and auditory cortices process separate patterns. A generative memory replay system that mitigates catastrophic forgetting by synthesizing pseudo-experiences of historical patterns through latent space generation, mirroring hippocampal-neocortical interactions in memory consolidation. Furthermore, we introduce the **MarineDrift-1.0** dataset—the first open-source dataset specifically designed for ocean data mining with natural concept drift. Experimental results on this dataset show BRAIN-VAE significantly outperforms all baselines in cold-start settings.

2 Backgrounds and Related Works

2.1 Learning on Time Series with Concept Drift

Concept drift is prevalent in sensor time - series data due to unpredictable factors like environmental changes and pollutant accumulation. Some prior studies have tackled this issue from the Domain Adaptation perspective in Transfer Learning, e.g., the ADARNN model [17]. However, these methods are limited to offline training of pre-available time series data. Online Learning methods include the Dynamically Scalable Network (DEN) [18], Deep Evolutionary Denoising Autoencoder (DEVDAN) [19] and Neural Networks with Dynamically Evolved Capacity (NADINE) [12], which evolves the structure dynamically and adjust the representational capacity; the HSN-LSTM [20], the first to embed an Adaptive Hybrid Spike (AHS) module in LSTM for stream-based prediction in concept-drift environments; and the FSNet [15] and OneNet [16] models employ a dynamic adaptive mechanism and enhance the effectiveness of online learning through a complementary learning paradigm, representing the most advanced methods in the current field of time-series online learning.

2.2 Catastropical Forgetting and Brain Memory Replay

Deep neural networks (DNNs) excel in diverse tasks but suffer from catastrophic forgetting during sequential learning, losing previously acquired knowledge [13]. For example, LSTM/RNN models experience exponential decay of long-term memory, degrading performance as critical temporal patterns are rapidly forgotten [21]. Conventional replay-based approaches mitigate forgetting by ensembling

past learned models [14], but these methods are computationally inefficient and impractical for real-time IoT systems due to storage constraints. In contrast, the human brain employs scalable Memory Replay during sleep/awake SWR events, where the hippocampus coordinates neocortical reactivation to stabilize memories [22,23]. The Complementary Learning Systems (CLS) Theory posits that this process consolidates experiences via hippocampal-neocortical interactions [24]. **Notably, brain replay abstract representations of learned patterns rather than raw data [25], suggesting no need for full historical data storage**. Inspired by this mechanism, we propose a generative replay neural network for marine time-series forecasting. Our model enables efficient continuous learning in resource-constrained IoT environments.

3 Methodology

3.1 Problem Definition

Online learning algorithms feature instantaneous forecasting and learning. The model iteratively updates parameters using per-data-point loss and gradients. Let $X = x_0, x_1, ..., x_t, ..., x_T$ denote real-time time-series data from a sensor, where T is unbounded. Under the online setting, data arrives sequentially: for forecasting at time t, only historical data $x_0, x_1, ..., x_{t-1}$ are available. The model generates predictions $\hat{x}^t$ using this past window, then updates parameters with the loss computed from the observed x^t. For multi-step forecasting with horizon $h > 1$, the target sequence $\{\hat{x}_t, \hat{x}_{t+1}, ..., \hat{x}_{t+h}\}$ is predicted, and loss is computed once the entire window is observed.

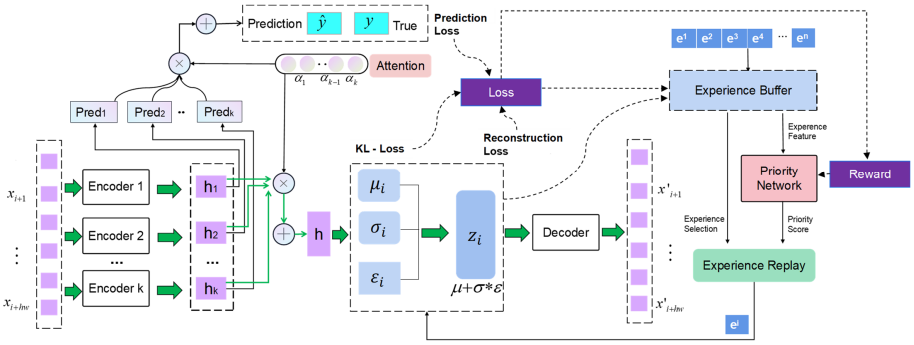

Fig. 2. The proposed online learning framework for dynamic marien time series data forecasting.

3.2 Overall Structure of the BRAIN-VAE

The BRAIN-VAE model integrates several key functions, including VAE-based generative learning mechanism, attention mechanisms, online learning mechanism, and brain-inspired replay mechanism. Accordingly, the BRAIN-VAE model is structured into three pathways, are shown in Figure 2:

- The **VAE-based encoder-decoder pathway** for capturing latent representations of time series data and generating reconstructions.
- The **Attention-based forecasting fusion pathway**, which dynamically weights the contributions of different layers for real-time prediction.
- The **Brain-inspired memory replay pathway**, which implement a generative replay mechanism to retain previously learned experience and prevent catastrophic forgetting

3.3 VAE based Encoder-Decoder Pathway

The BRAIN-VAE model is built upon the Variational Autoencoder (VAE) structure, a generative model that is capable of learning distribution parameters (mean and variance) of latent factors and generating data based on learned distributions.

- **Encoder**: Given an input sequence $x = \{x_i, x_{i+1}, \ldots, x_T\}$, the encoder outputs the mean (μ) and variance (σ^2) of the latent variables. These latent variables z are sampled from a Gaussian distribution:

$$\begin{aligned} h &= \sum_{k=1}^{K} \alpha_k h_k \\ \mu_z &= f^{\mu}_{linear}(h) \\ \sigma_z &= f^{\sigma}_{linear}(h) \end{aligned} \tag{1}$$

  Let h denote the concatenated hidden representations derived from the internal encoders. The attention weight α for the concatenation operation is computed through the proposed attention-based forecasting fusion pathway, which will be elaborated in the subsequent section.
- **Reparameterization Trick**: To enable backpropagation during training, the reparameterization trick is employed. Instead of directly sampling z from $\mathcal{N}(\mu_z, \sigma_z^2)$, a random noise vector ε is sampled from a standard normal distribution, and z is sampled by:

$$z = \mu_z + \sigma_z \cdot \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, 1) \tag{2}$$

  This reparameterization allows the latent space to remain differentiable, thus facilitating gradient-based optimization.
- **Decoder**: The decoder reconstructs the input data by generating x' from the sampled latent variables:

$$x' = f_{decoder}(z) \tag{3}$$

  The reconstruction loss is minimized to ensure that the latent space captures meaningful information.

3.4 Attention-based Forecasting Fusion Pathway

Dynamic representational capacity adjustment is vital for time-series forecasting under concept drift [26,19]. Specifically, we aim to develop a model capable of performing efficient inference with a 'cold start' fashion while dynamically adapting its representational capacity to the evolving stream. We realize it through a complementary inference pathway integrating internal encoders' information and dynamically balancing contributions via an attention mechanism.

The BRAIN-VAE model supports integrating multiple time-series encoders as internal modules, thereby enhancing the learning capability of temporal representations. The BRAIN-VAE model synthesizes its forecast by assigning weighted importance to the predictions generated at each module. Modules contributing less relevant information to the current forecast receive lower weights, resulting in a final prediction that aggregates the forecasts from all internal modules as a weighted sum, formulated as:

$$\begin{aligned} h_k &= f^k_{encoder}(x) \\ y^k_{pred} &= f^k_{pred}(h_i) \\ y_{pred} &= \sum_{k=1}^{K} \alpha_i y^k_{pred} \end{aligned} \tag{4}$$

, $f_{encoder}(*)$ and $f_{pred}(*)$ transforms the original sequence x into intermediate representation and localized forecasts, and h_i represents the hidden representation of the ith module. The weight α_i is recalculated dynamically through the following re-weighting mechanism. To update the model iteratively, we incorporate the re-weighting operation, which is implemented based on an feed forward neural network based Attention Network, formulated as:

$$\alpha_k = f_{attention}(h_k) \tag{5}$$

This dynamic attention mechanism adaptively adjusts module contributions via softmax-normalized weights, quantifying their importance in capturing time-varying patterns. Unlike fixed-weight architectures, this enables rapid cold-start convergence while allowing modules to dynamically specialize based on input characteristics, enhancing adaptability to evolving data distributions and improving prediction accuracy.

3.5 Brain-inspired Memory Replay Pathway

Drawing on cognitive neuroscience, we incorporate a brain-inspired replay mechanism into the BRAIN-VAE model. Instead of replaying or generating the original raw data, it internally replays high-level hidden representations. This concept stems from the discovery in human brain hippocampal memory reactivation process that reactivating abstract representations tied to past experiences. Important components related to the replay strategy is described as follows:

Experience Replay Buffer Define the replay buffer $\mathcal{D} = \{e_i\}_{i=1}^{T}$ where each stored experience e_i contains:

- Experience index $i \in \mathbb{R}$
- Latent mean $\mu_z^{(i)} \in \mathbb{R}^{Z_{\dim}}$
- Latent variance $\sigma_z^{(i)} \in \mathbb{R}^{Z_{\dim}}$
- loss $\mathcal{L}_{BRAIN\_VAE}^{(i)}$

Priority Scoring Function Define the priority network $\theta_{priority} : \mathbb{R}^{d_\phi} \rightarrow \mathbb{R}$ with input feature vector, the feature vector is built based on the stored experience:

$$\phi^{(i)} = \left[\mu_z^{(i)}, \sigma_z^{(i)}, \mu_z^{(t)}, \sigma_z^{(t)}, \mathcal{L}_{BRAIN\_VAE}^{(i)}, \mathcal{L}_{BRAIN\_VAE}^{(t)}, \Delta d^{(i,t)}\right] \tag{6}$$

The temporal difference is calculated by $\Delta d^{(i,t)} = t - i$, it reflects the influence of the past experience in time i to data in current time t. Priority score (expected reward) is calculated on the feature:

$$R_{expected}^{(i)} = \theta_{priority}(\phi^{(i)}) \tag{7}$$

Selection Strategy The z selection and replay policy is as follows:

$$P_{select}(e_i) = \begin{cases} 1, & \text{if } i = \arg\max_j R_{expected}^{(j)} \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

Inspired by reinforcement learning, we introduce the reward-based optimization strategy that leverages the experience feature $\phi^{(i)}$ to compute a performance-based reward signal. This reward quantifies the prediction performance enhancement achieved through experience replay of the selected experience e_i This strategy ensures that the model consolidates both new and previously learned information to enhance online prediction with a low-cost fashion.

3.6 Loss functions and Parameter Optimization

For the BRAIN-VAE, the most important learning objective is to minimize the forecast bias, hence we need first define the forecast bias in objective function. The **Forecast Loss** function include the forecast loss of each module and the ensemble forecast loss, as follows:

$$\begin{aligned} \mathcal{L}_{P_j} &= \sum_{h=1}^{H} MSE\left(y_{j,h}{}', y_{j,h}\right) \\ \mathcal{L}_P &= \sum_{h=1}^{H} MSE\left(\sum_{k=1}^{K} \alpha_k y'_{k,h}, y_{k,h}\right) \end{aligned} \tag{9}$$

In addition to the loss function of forecast, the training of the BRAIN-VAE model also needs the guidance of two other loss functions. One of them is the Kullback-Leibler divergence loss (KL loss), which serves to measure the difference between the posterior distribution $q(z|x)$ and the standard normal distribution $p(z)$. The **KL-loss** is defined as:

$$\mathcal{L}_{KL} = -0.5 * \sum_{m=1}^{Z_{\text{dim}}} (1 + \log(\sigma_m^2) - \mu_m^2 - \sigma_m^2) \tag{10}$$

The other is **Reconstruction loss**, which can be L1 or L2 Loss, as follows:

$$\mathcal{L}_{Rec} = MSE(x, x') \tag{11}$$

Finally, in order to optimize the BRAIN-VAE model, we need to minimize the above four losses simultaneously, namely:

$$\mathcal{L}_{BRAIN\_VAE} = \sum_{k=1}^{K} \mathcal{L}_{P_j} + \mathcal{L}_P + \mathcal{L}_{KL} + \mathcal{L}_{Rec} \tag{12}$$

For the Priority Network, we introduce the reward-based optimization strategy, the reward signal is based on prediction improvement. The larger the loss decrease after replay, the higher the reward assigned, as follows:

$$R_{actual}^{(i)} = \underbrace{\mathcal{L}_{\text{BRAIN\_VAE}}^{(i)}}_{\text{pre-replay}} - \underbrace{\mathcal{L}_{\text{BRAIN\_VAE}}^{(i)}}_{\text{post-replay}} \tag{13}$$

The Priority Network loss **Priority Loss** is as follows:

$$\mathcal{L}_{\text{priority}}^{i} = \left(R_{expected}^{(i)} - R_{actual}^{(i)} \right)^2 \tag{14}$$

The The training process is illustrated as follows:

Algorithm 1 Optimization Procedure for BRAIN-VAE Model

Initialize main model θ_{main} and priority network θ_{priority}
for each time step $t = 1$ to T **do**
  Generate prediction $\mathbf{y}_t$ and latent $\mathbf{z}_t$
  Compute loss $\mathcal{L}_{BRAIN-VAE}^{t}$ before replay and store experience e_t in $\mathcal{D}$
  if t mod $K = 0$ **and** $|\mathcal{D}| > 0$ **then**
    Build feature vector on $\mathcal{D}$ and compute priority scores (expected reward) for the stored experiences $R_{expected}^{(i)} = \theta_{priority}(\phi^{(i)})$
    Select experience e_i from $\mathcal{D}$ with top priority score
    Replay the selected experience and compute loss of the replayed experience, and update the main model θ_{main}.
    Recompute the loss $\mathcal{L}_{BRAIN-VAE}^{t}$ for current time t after replay, get the actual reward (prediction improvement) $R_{actual}^{(t)}$
    Compute the reward loss of the Priority Network, and update the the model θ_{priority};
  end if
end for

In practice, we can set the replay frequency $K = 1$, and at the end of each round of forecast, the model adjusts itself based on the instantaneous loss before the coming rounds. The error derivatives of BRAIN-VAE model are backpropagated to each of the internal module to adjust the corresponding parameters.

4 Experiments

4.1 Experimental Data, Metrics and Environment

The MarineDrift-1.0 dataset is constructed using in-situ near-real-time marine observation data from buoy networks under the MO category of the Copernicus Marine Service[4].This dataset currently contains six critical physical and biogeochemical parameters essential for monitoring marine environmental dynamics: sea temperature (TEMP), salinity (PSAL), dissolved oxygen concentration (DOX1), turbidity (TUR4), chlorophyll-a concentration (CPHL), and horizontal wind speed (WSPD). To ensure data representativeness, we implement a two-stage concept drift detection framework. First, the ADWIN (Adaptive Windowing) algorithm dynamically identifies concept drift phase through adaptive time window adjustments. Second, the Wasserstein distance metric quantifies the magnitude of detected drifts, providing an interpretable measure of distributional divergence between sequential data segments.

Table 1. Statistics Information of Selected Experimental Data from MarinShift-1.0

Dataset	Domain	# Time Series	Min Length	Mean Length	Max Length	Total Observations	Forecast Horizon
CPHL	BGC	5	6376	46391.6	70000	231958	{1,24,48}
DOX1	BGC	5	14524	38249.4	79046	191247	{1,24,48}
PSAL	Physical oceanography	5	11701	17179.8	28987	85899	{1,24,48}
TEMP	Physical oceanography	5	12796	19321.2	34361	96606	{1,24,48}
TUR4	BGC	5	4398	9214.8	15542	46074	{1,24,48}
WSPD	Meteorological	5	8045	34417	75595	172085	{1,24,48}

In this work, for experimental validation, we curated challenging subsets from the MarineDrift-1.0 dataset exhibiting substantial distributional shifts. These subsets provide a rigorous testbed for evaluating model adaptation capabilities under realistic marine environmental non-stationarity conditions. Each time series was split into warm-up (30%) and online-inference (70%) phases. The warm-up data serves to train traditional offline models and several online learning models that need warm-up phase, also used to normalize the online-inference data. The online-inference data is used for performance validation across all methods.

[4] `https://data.marine.copernicus.eu/product/INSITU_GLO_PHYBGCWAV_DISCRETE_MYNRT_013_030/files?subdataset=cmems_obs-ins_glo_phybgcwav_mynrt_na_irr_202311--ext--history`

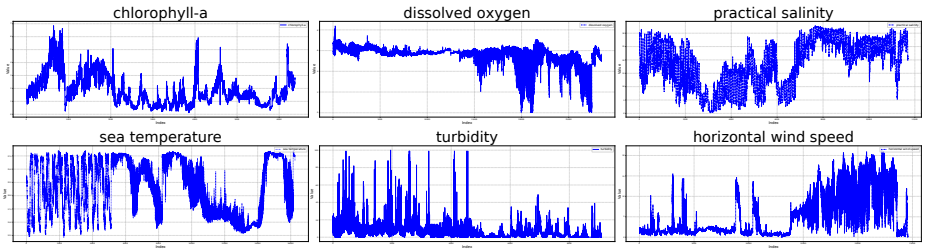

Fig. 3. Representative examples of concept drift in 6 key types of marine observational time series data.

4.2 Experimental Settings

Settings for BRAIN-VAE: The BRAIN-VAE model employs a three-module architecture, each adopts the encoder structure of TS2Vec, as follows:

Module 1 (Global Feature Extraction Module): This module is dedicated to extracting global contextual information from the entire historical time series window. By encoding the input sequence as a holistic entity, it aims to capture long-term dependencies and macroscopic patterns that span the entire segment, such as trends and seasonality. The module synthesizes and refines feature representations across all time steps to generate a single, fixed-dimensional context vector. This vector is subsequently fed into a linear regressor to produce predictions from a macro-level perspective.

Module 2 (Local Feature Extraction Module): This module focuses on capturing local patterns within the original univariate time series. It employs a TS2Vec framework with an input dimensionality of 1 to process the raw univariate time series, leveraging hierarchical dilated convolution operations to extract multi-scale temporal features. For prediction, this module exclusively extracts and utilizes the output feature vector from the final time step of the sequence. Finally, this feature vector is mapped to the prediction space via a linear regressor.

Module 3 (Temporal Feature Enhancement Module): This module integrates additonal 7-dimensional timestamp features (minute, hour, day of week, day of month, day of year, month, and week of year) and concatenates them with the original time series data to form an 8-dimensional input. Significantly improving the model's ability to capture periodic, seasonal, event-driven patterns, and complex temporal dependencies across scales in ocean.

The dimensionality of latent variable z is set to 8. In terms of the Attention Network and Priority Network, we both set up 2 hidden layers with 50 neurons in each layer.

Settings for Baselines: Considering that BRAIN-VAE is fundamentally a deep learning model, we compared it with several representative deep learn-

ing models for time series data prediction. This includes models without online learning capabilities, as well as models designed for online learning. The detailed settings for the baselines are shown in Table 2.

Offline Learning Models includes: 1) **RNN-LSTM** [27]; 2) **Transformer** [28] and its variants 3) **Informer** [29], 4) **FEDFormer** [30], and 5) **Autoformer** [31]; 6) **NBeats** [32], the M-Competition 2020 champion; Pre-trained time series model 7) **PatchTST** [33]; and 8) **AdaRNN** [34], a domain adaptation framework employing Temporal Distribution Matching.

Online Learning Models includes: 1) **HSN-LSTM** [20], which integrates an Adaptive Hybrid Spike module and dual attention mechanisms into LSTM for concept drift-aware stream prediction; 2) **NADINE** [12], a dynamic neural network that evolves its architecture by pruning/growing hidden units/layers based on drift detection; 3) **FSNet** [15], inspired by Complementary Learning Systems theory to combine fast learning with slow adaptation via temporal pattern memory; and 4) **OneNet** [16], which dynamically combines temporal/cross-variable dependency models using reinforcement learning within an online convex programming framework. FSNet and OneNet are state-of-the-art (SOTA) models that serve as strong baselines in online time-series forecasting.

Settings for Ablation Study: To evaluate the brain replay mechanism's impact, we designed three variants: **BRAIN-VAE-RER** (random experience replay), **BRAIN-VAE-WRE** (without experience replay), and **BRAIN-VAE-PER** (priority experience replay via PriorityNetwork that we adopt). Ablation experiments were also conducted to assess encoder architecture sensitivity by substituting the original three Ts2Vec-based encoders with TCN (**BRAIN-VAE-TCN**). Additionally, we tested the model's performance without attention mechanisms, referred to as **BRAIN-VAE-woAttn**.

4.3 Experimental Results and Discussion

We conducted a comparative evaluation of the BRAIN-VAE model against baseline methods. As shown in Table 3 and Figure 4, the BRAIN-VAE model not only excels in prediction accuracy but also demonstrates remarkable stability across different prediction horizons and diverse data types, showcasing a clear advantage over the baseline methods.. These results highlight the model's robustness in handling marine time series data, particularly in addressing distributional shifts. Notably, our approach was rigorously validated under cold-start settings without any warm-up data. In contrast, all other methods, including state-of-the-art baselines such as FSNet and OneNet, rely on warm-up data for initialization. Furthermore, our analysis reveals that FSNet exhibits less robustness compared to OneNet and, in some cases, fails to outperform even offline methods.

In the ablation study, as illustrated in Figure 5, the choice of replay mechanism significantly impacts the performance of the BRAIN-VAE model. Specifically, the BRAIN-VAE-PER variant, which employs Prioritized Experience Replay, performs better than other models in the vast majority of cases. Mean-

Table 2. Parameter Settings for Comparison Methods

Offline Methods	
RNN-LSTM	dimension of LSTM hidden layer: 200 number of LSTM layers: 2 use of bias in LSTM: True
NBeats	stack types: {H-1: generic}, {H-24,H-48: [trend, seasonality]} number of blocks per stack: 3 hidden layer units: 256
{Transformer Informer Autoformer FEDformer}	model dimension: 512 feedforward network dimension: 2048 number of attention heads: 8 number of encoder layers: 3 number of decoder layers: 1 attention factor: 3
PatchTST	model dimension: 128 feedforward network dimension: 256 number of encoder layers: 3 number of attention heads: 8 patch length: 16 stride: 8
AdaRNN	RNN hidden layer dimension: 64 number of RNN layers: 2 number of domains: 2 data model: TDC distribution distance function: adversarial distance
Online Methods	
HSN-LSTM	membrane potential time constant, τu: 64 spike time constant, τa: 32 initial threshold, $c0$: 4e-2 dynamic range control parameter, $d0$: 1.8 time step, dt: 0.01(10ms)
NADINE	network type: stacked initial hidden layers: 1 stabilization period: 20 anomaly threshold 1: chi2inv(0.99, 168) chi2inv(0.999, 168) forgetting factor: 0.98 drift detection error thresholds: {0.001 (drift), 0.005 (warning)}
FSNet	feed-forward network dimension: 2048 model hidden layer dimension: 512 number of decoder layers: 1 number of encoder layers: 2 number of attention heads: 8 sparse attention factor: 5 test batch size: 1 online learning model: full training method: fsnet
OneNet	model dimension: 32 feed-forward network dimension: 128 number of attention heads: 8 number of encoder layers: 2 number of decoder layers: 1 sparse attention factor: 5 test batch size: 1 online learning model: full

Table 3. Performance of the BRAIN-VAE model and baseline models.

Model	CPHL						DOX1					
	H=1		H=24		H=48		H=1		H=24		H=48	
	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
BRAIN-VAE	**0.1200**	**0.0348**	**0.2752**	**0.1303**	**0.3561**	**0.1744**	**0.1284**	0.2431	**0.3233**	**0.1835**	**0.4143**	**0.2493**
RNN-LSTM	1.2129	0.7007	1.2791	0.7697	1.2873	0.7759	1.5160	1.0581	1.7229	1.2775	1.8434	1.3923
NBeats	0.3128	0.1357	0.5346	0.2595	0.6535	0.3253	0.3593	0.2442	0.6490	0.3887	0.7792	0.5021
Transformer	0.7835	0.3957	1.0258	0.5666	1.0278	0.5759	0.7265	0.4107	1.0493	0.6573	1.2220	0.8219
Informer	0.8494	0.4298	0.9944	0.5613	1.0740	0.6297	0.8116	0.4744	1.1784	0.7544	1.2083	0.8098
FEDformer	0.2812	0.1259	0.5834	0.3167	0.6682	0.3679	0.3048	0.1818	0.6862	0.4198	0.7637	0.4753
Autoformer	0.3950	0.1915	0.6145	0.3434	0.7136	0.3996	0.4002	0.2570	0.7143	0.4603	0.7865	0.5178
PatchTST	0.2743	0.1150	0.5068	0.2558	0.6132	0.3176	0.2712	0.1387	0.6224	0.3659	0.7106	0.4298
AdaRNN	0.6563	0.4481	0.7884	0.5313	1.0645	0.6720	0.3310	0.2046	0.9604	0.7031	0.9843	0.7004
HSN-LSTM	0.6525	0.3446	0.8595	0.4563	0.9336	0.5106	0.6184	0.3243	0.8875	0.5427	1.0505	0.6711
NADINE	0.4007	0.1741	0.6803	0.3809	0.8394	0.4884	0.4465	0.1847	1.2152	0.6640	1.6422	0.9133
FSNet	0.4020	0.2028	0.3435	0.1777	0.3909	0.2031	0.6235	0.4459	0.5898	0.3069	0.5625	0.3499
OneNet	0.2343	0.0938	0.4560	0.2317	0.4958	0.2581	0.2165	**0.1099**	0.5497	0.3331	0.5721	0.3712

Model	PSAL						TEMP					
	H=1		H=24		H=48		H=1		H=24		H=48	
	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
BRAIN-VAE	**0.2953**	**0.1340**	**0.9480**	**0.5856**	**1.2242**	**0.8045**	**0.2198**	**0.0994**	**0.6287**	**0.3842**	**0.8412**	**0.5137**
RNN-LSTM	4.2336	3.4063	4.8702	4.0146	5.5061	4.6823	3.2687	2.4213	4.2192	3.1841	4.4146	3.3858
NBeats	0.8107	0.5186	1.8068	1.1749	2.2441	1.5172	1.0030	0.6786	1.5496	0.9989	2.1463	1.4210
Transformer	0.8027	0.5094	2.4351	1.7783	3.3364	2.4657	1.2431	0.7707	2.7991	1.8921	3.4034	2.4428
Informer	1.3949	0.9856	2.4642	1.7923	3.1687	2.3294	1.7742	1.2145	2.8586	2.0476	3.2066	2.3762
FEDformer	0.6396	0.3796	2.2741	1.6161	2.5236	1.8353	0.8266	0.5282	1.7587	1.2064	2.0894	1.4110
Autoformer	1.2059	0.8755	2.3049	1.6566	2.9015	2.1718	1.1957	0.8272	1.8325	1.3267	2.0443	1.5037
PatchTST	0.6136	0.3350	1.8009	1.1721	2.3252	1.5422	0.7651	0.4506	1.5439	0.9190	1.8213	1.0751
AdaRNN	1.0281	0.8156	4.0677	3.3787	3.9977	3.2982	1.0847	0.8079	2.8720	2.3254	3.3872	2.5916
HSN-LSTM	0.9187	0.5960	2.0921	1.5183	2.6806	2.0383	1.3076	0.8061	2.3727	1.6801	2.8312	2.0698
NADINE	2.0536	0.5998	5.3457	2.9008	6.8524	3.9764	1.2894	0.5368	3.4520	1.6377	5.2622	2.6760
FSNet	1.6107	1.0467	4.5993	2.0053	2.5862	1.4668	1.6133	0.9453	2.2440	1.4468	2.4191	1.6479
OneNet	0.5730	0.3291	1.7088	1.2038	2.0060	1.4595	0.6825	0.4075	1.2452	0.8421	1.6971	1.2028

Model	TUR4						WSPD					
	H=1		H=24		H=48		H=1		H=24		H=48	
	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
BRAIN-VAE	**1.3968**	**0.5463**	**2.8633**	**1.5436**	**3.0698**	**1.7291**	**0.6496**	**0.3076**	**1.5595**	**0.9558**	1.7178	1.1927
RNN-LSTM	7.0597	5.5112	7.2619	5.6497	7.6757	6.1001	4.5227	3.1496	4.9202	3.4894	5.0434	3.6070
NBeats	3.9913	2.2494	4.8533	2.7216	5.1332	3.0728	1.9122	1.1723	2.9384	2.0149	3.6390	2.5035
Transformer	5.1282	3.7572	5.9702	4.3753	6.6822	4.9547	2.6666	1.6509	3.7313	2.5430	4.3578	2.9860
Informer	5.0593	3.4409	6.4531	4.8229	7.1578	5.5133	2.8657	1.8281	4.1323	2.8662	4.3691	3.0360
FEDformer	4.3692	2.7313	5.1605	3.3677	5.6446	3.7936	1.8921	1.2258	2.7489	1.9040	2.9713	2.0751
Autoformer	4.2407	2.5758	4.9667	3.1035	5.2695	3.3313	2.1136	1.3943	2.6700	1.8512	2.9566	2.0761
PatchTST	3.9400	2.0615	4.5397	2.4823	4.9003	2.7783	1.8384	1.1825	2.5421	1.7481	2.8270	1.9613
AdaRNN	4.7166	3.0292	6.4782	4.8479	6.1856	4.4350	1.8935	1.2347	3.4822	2.3841	3.9063	2.7208
HSN-LSTM	4.4713	2.9312	6.0312	4.3518	6.5432	4.9223	2.2278	1.3779	3.3015	2.2822	3.8957	2.7278
NADINE	4.7241	2.6707	5.6897	3.4842	6.6388	4.4067	2.3424	1.5615	3.5149	2.4989	3.8332	2.7577
FSNet	4.0991	2.1153	4.8031	2.9947	5.4131	3.7947	1.9539	1.2961	1.7099	1.1518	**1.7038**	**1.1465**
OneNet	3.2020	2.1984	3.9796	2.6236	4.4182	3.1360	1.4301	0.8046	2.6530	1.7941	2.9708	1.9418

[1] We conducted 2340 experiments in total, and it is not feasible to present them all specifically. All data presented above are the average results across time-series instances of each marine data type. The detailed tables can be obtained by contacting the author..

while, the BRAIN-VAE-RER variant, which employs a random experience selected replay mechanism, performs notably worse than even those models that do not utilize any replay mechanism at all. This comparison strongly demonstrates that the strategy of strategically selecting samples for replay is effective and helps the model to learn better. Additionally, it was observed that removing the attention-based fusion module (BRAIN-VAE-woAttn) results in a noticeable decline in predictive performance, underscoring the critical role of the attention layer in the model's effectiveness. The BRAIN-VAE model is designed to seamlessly integrate any type of time series model as its internal module; even when incorporating relatively simple structures such as TCN (Temporal Convolutional Network), the model achieves competitive performance, showcasing its flexibility and robustness in handling diverse time series data.

To gain deeper insights into the importance reweighting behavior and the dynamics of the latent variable, we conducted a case study using the PSAL-3 dataset. As shown in Figure 6, the attention weights across the three modules exhibit dynamic adjustments rather than remaining static throughout the online learning process. Notably, a significant change in weight allocation occurs during periods of data distribution shift. Specifically, Module-1 tends to contribute less in high-frequency data segments, while BRAIN-VAE assigns higher weights to Module-2, which is better suited for capturing and adapting to high-frequency patterns. To further investigate the behavior of the latent variable z, we selected several time points to extract its distribution parameters and plotted the corresponding probability density. The results demonstrate that the distribution of z undergoes continuous changes over time, reflecting its dynamic adaptation to the evolving patterns in the time series.

4.4 Performance and Deployment Considerations

The operational deployment of online learning models in large-scale systems requires careful consideration of computational efficiency and scalability. Our implementation shows that edge devices in the network have limited computational capacity, making them unable to support complex AI algorithms—especially in high-throughput environments. Therefore, our deployment architecture adopts a compute-offloading paradigm: sensor data skips local processing and is directly streamed to a centralized computing facility, the National Supercomputing Center in Jinan. As a core partner in the ocean observation network, the center handles data collection, processing, and analysis. Specifically, we design a supercomputing-parallelized architecture to handle massive concurrent data streams, distributing high-throughput intelligent computations across thousands of cores. Each stream is assigned to an individual core for end-to-end online learning and inference, addressing the inability of edge devices to perform AI model inference services in high-throughput scenarios. This centralized, high-performance framework meets BRAIN-VAE's computational needs and provides a robust, scalable solution for deploying advanced AI in marine monitoring systems.

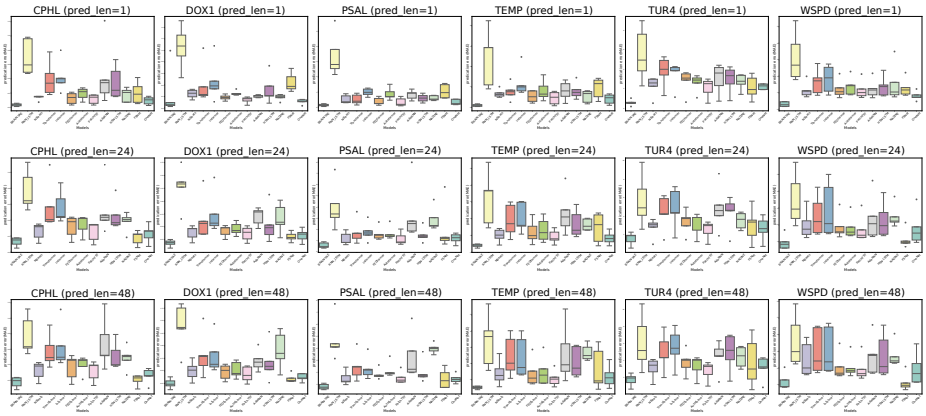

Fig. 4. A box plot comparison of model performance across the experimental datasets, presenting results in a consistent order from left to right: BRAIN-VAE, RNN-LSTM, NBeats, Transformer, Informer, FEDformer, Autoformer, PatchTST, AdaRNN, HSN-LSTM, NADINE, FSNet, and OneNet.

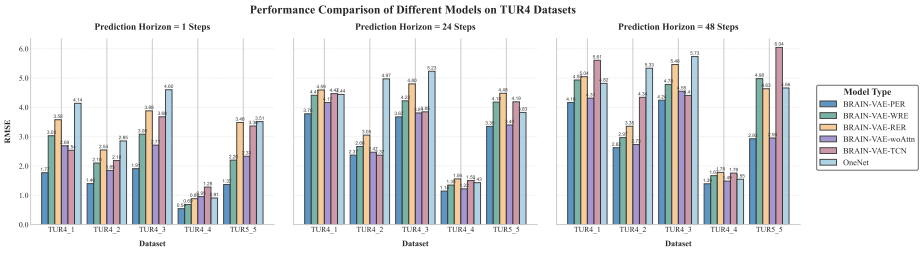

Fig. 5. Performance comparison of BRAIN-VAE variants on 5 time series in TUR4 datasets across prediction horizons (1, 24, 48 steps) using RMSE.

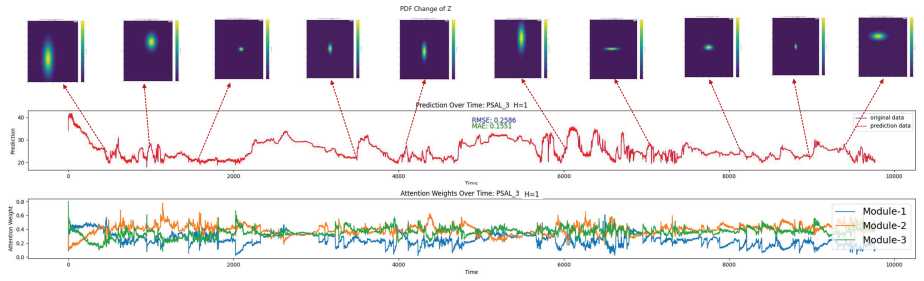

Fig. 6. An illustration of the probability density of z (the lantent representation space is set as 2) change across Prediction Steps for PSAL-3 data.

5 Conclusion

In this work, we propose BRAIN-VAE, an online deep learning model tailored for marine observation data forecasting. BRAIN-VAE integrates a variational autoencoder backbone with an attention mechanism to effectively fuse information from multiple modules, enabling robust generalization capabilities for handling marine data with concept drift. Notably, BRAIN-VAE operates efficiently in cold-start scenarios without the need for pre-training or storing historical models. Instead, it employs a generative replay mechanism that reconstructs hidden representations of past data distributions, akin to a 'hippocampus', to serve as pseudo-replay data. This allows the model to 'review' past experiences most relevant to the current patterns, making it particularly well-suited for long-term, real-time processing of marine observation data streams. Additionally, we introduce MarineDrift-1.0, the first dataset specifically designed to study concept drift in marine observation data. This dataset provides a valuable resource for evaluating forecasting models in marine environments. To promote further research and reproducibility, we make the source code and MarineDrift-1.0 dataset publicly available at: `https://github.com/muzixiang/BRAIN-VAE`

Despite the model's strong performance, several promising enhancement directions exist. The current univariate BRAIN-VAE framework, featuring a modular architecture that decouples generative memory replay, multi-module dynamic ensembling, and VAE-based representation learning, can naturally scale to multivariate scenarios. Future work will integrate multivariate encoders and cross-attention layers to model inter-variable dependencies—requiring no infrastructure modifications. Additionally, the model lacks systematic sensitivity analysis of hyperparameters, e.g., the replay frequency and the number of replayed experiences. Furthermore, we will quantify the model's performance in catastrophic forgetting scenarios by introducing a dedicated dataset and evaluating retention of historical knowledge. Finally, we aim to enrich the MarineDrift dataset with complex periodic patterns and multivariate samples.

Acknowledgments.

This work was supported by the Major Innovation Project for the Integration of Science, Education, and Industry of Qilu University of Technology (Shandong Academy of Sciences) (2024GH24, 2023JBZ02, 2023HYZX01), the Jinan '20 New Colleges and Universities' Funded Project (202333043), the Key Research and Development Program of Shandong Province (Major Scientific and Technological Innovation Project) (2024CXGC010111), and the Shandong Province Taishan Scholar Climbing Program Project (NO.tspd20240814).

References

1. Torres, J.F., Hadjout, D., Sebaa, A., Martínez-Álvarez, F., Troncoso, A.: Deep learning for time series forecasting: a survey. Big Data **9**(1), 3–21 (2021)

2. Wen, Q., Zhou, T., Zhang, C., Chen, W., Ma, Z., Yan, J., Sun, L.: Transformers in time series: A survey. arXiv preprint arXiv:2202.07125 (2022)
3. Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., Bouchachia, A.: A survey on concept drift adaptation. ACM computing surveys (CSUR) **46**(4), 1–37 (2014)
4. Agrahari, S., Singh, A.K.: Concept drift detection in data stream mining: A literature review. Journal of King Saud University-Computer and Information Sciences **34**(10), 9523–9540 (2022)
5. Nguyen, N.T., Heldal, R., Pelliccione, P.: Concept-drift-adaptive anomaly detector for marine sensor data streams. Internet of Things p. 101414 (2024)
6. Grossberg, S., Grossberg, S.: How does a brain build a cognitive code? Studies of mind and brain: Neural principles of learning, perception, development, cognition, and motor control pp. 1–52 (1982)
7. Yuan, L., Li, H., Xia, B., Gao, C., Liu, M., Yuan, W., You, X.: Recent advances in concept drift adaptation methods for deep learning. In: IJCAI. pp. 5654–5661 (2022)
8. Hoi, S.C., Sahoo, D., Lu, J., Zhao, P.: Online learning: A comprehensive survey. Neurocomputing **459**, 249–289 (2021)
9. Sahoo, D., Pham, Q., Lu, J., Hoi, S.C.: Online deep learning: learning deep neural networks on the fly. In: Proceedings of the 27th International Joint Conference on Artificial Intelligence. pp. 2660–2666 (2018)
10. Zhang, S.s., Liu, J.w., Zuo, X., Lu, R.k., Lian, S.m.: Online deep learning based on auto-encoder. Applied Intelligence **51**(8), 5420–5439 (2021)
11. Lu, Y., Cheung, Y.m., Tang, Y.Y.: Dynamic weighted majority for incremental learning of imbalanced data streams with concept drift. In: IJCAI. pp. 2393–2399 (2017)
12. Pratama, M., Za'in, C., Ashfahani, A., Ong, Y.S., Ding, W.: Automatic construction of multi-layer perceptron network from streaming examples. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management. pp. 1171–1180 (2019)
13. French, R.M.: Catastrophic forgetting in connectionist networks. Trends in cognitive sciences **3**(4), 128–135 (1999)
14. Wang, H., Li, M., Yue, X.: Inclstm: incremental ensemble lstm model towards time series data. Computers & Electrical Engineering **92**, 107156 (2021)
15. Pham, Q., Liu, C., Sahoo, D., Hoi, S.C.: Learning fast and slow for online time series forecasting. arXiv preprint arXiv:2202.11672 (2022)
16. Wen, Q., Chen, W., Sun, L., Zhang, Z., Wang, L., Jin, R., Tan, T., et al.: Onenet: Enhancing time series forecasting models under concept drift by online ensembling. Advances in Neural Information Processing Systems **36** (2024)
17. Du, Y., Wang, J., Feng, W., Pan, S., Qin, T., Xu, R., Wang, C.: Adarnn. In: Proceedings of the 30th ACM International Conference on Information &; Knowledge Management (Oct 2021)
18. Yoon, J., Yang, E., Lee, J., Hwang, S.J.: Lifelong learning with dynamically expandable networks. arXiv preprint arXiv:1708.01547 (2017)
19. Ashfahani, A., Pratama, M., Lughofer, E., Ong, Y.S.: Devdan: Deep evolving denoising autoencoder. Neurocomputing **390**, 297–314 (2020)
20. Zheng, W., Zhao, P., Chen, G., Zhou, H., Tian, Y.: A hybrid spiking neurons embedded lstm network for multivariate time series learning under concept-drift environment. IEEE Transactions on Knowledge and Data Engineering p. 1–1 (Jan 2022)

21. Zhao, J., Huang, F., Lv, J., Duan, Y., Qin, Z., Li, G., Tian, G.: Do rnn and lstm have long memory? In: International Conference on Machine Learning. pp. 11365–11375. PMLR (2020)
22. Ji, D., Wilson, M.A.: Coordinated memory replay in the visual cortex and hippocampus during sleep. Nature neuroscience **10**(1), 100–107 (2007)
23. Tambini, A., Davachi, L.: Awake reactivation of prior experiences consolidates memories and biases cognition. Trends in cognitive sciences **23**(10), 876–890 (2019)
24. Kumaran, D., Hassabis, D., McClelland, J.L.: What learning systems do intelligent agents need? complementary learning systems theory updated. Trends in cognitive sciences **20**(7), 512–534 (2016)
25. Pezzulo, G., Van der Meer, M.A., Lansink, C.S., Pennartz, C.M.: Internally generated sequences in learning and executing goal-directed behavior. Trends in cognitive sciences **18**(12), 647–657 (2014)
26. Gama, J., Rodrigues, P.P., Spinosa, E., Carvalho, A.: Knowledge discovery from data streams. In: Web Intelligence and Security, pp. 125–138. IOS Press (2010)
27. Hu, Y., Yan, L., Hang, T., Feng, J.: Stream-flow forecasting of small rivers based on lstm. arXiv preprint arXiv:2001.05681 (2020)
28. Li, S., Jin, X., Xuan, Y., Zhou, X., Chen, W., Wang, Y.X., Yan, X.: Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. Advances in neural information processing systems **32** (2019)
29. Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., Zhang, W.: Informer: Beyond efficient transformer for long sequence time-series forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 35, pp. 11106–11115 (2021)
30. Zhou, T., Ma, Z., Wen, Q., Wang, X., Sun, L., Jin, R.: Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In: International conference on machine learning. pp. 27268–27286. PMLR (2022)
31. Wu, H., Xu, J., Wang, J., Long, M.: Autoformer: Decomposition transformers with Auto-Correlation for long-term series forecasting. In: Advances in Neural Information Processing Systems (2021)
32. Oreshkin, B.N., Carpov, D., Chapados, N., Bengio, Y.: N-beats: Neural basis expansion analysis for interpretable time series forecasting. arXiv preprint arXiv:1905.10437 (2019)
33. Nie, Y., Nguyen, N.H., Sinthong, P., Kalagnanam, J.: A time series is worth 64 words: Long-term forecasting with transformers (2022)
34. Du, Y., Wang, J., Feng, W., Pan, S., Qin, T., Xu, R., Wang, C.: Adarnn: Adaptive learning and forecasting of time series. In: Proceedings of the 30th ACM International Conference on Information & Knowledge Management. pp. 402–411 (2021)