Reservoir Computing Combined with Optimizable Quantile-Based Subsampling for Small-Sample Time-Series Classification

Ziqiang Li[1*], Yun Liu[2], and Gouhei Tanaka[1,3]
[1]Department of Computer Science, Graduate School of Engineering, Nagoya Institute of Technology, Nagoya, 466-8555, Japan
Emails: ziqiang-li@nitech.ac.jp*, tanaka.gouhei@nitech.ac.jp
[2]Faculty of Information and Human Sciences, Kyoto Institute of Technology, Kyoto, 606-8585, Japan.
Email: liuyun@kit.ac.jp
[3]International Research Center for Neurointelligence, The University of Tokyo, Tokyo, 113-0033, Japan.

***Abstract*—Rapidly training reliable machine learning models from limited data remains highly desirable in real-world time-series classification (TSC) applications. Reservoir Computing (RC), originating from dynamical system modeling and characterized by fixed high-dimensional nonlinear projections with lightweight training, has emerged as a promising paradigm for small-sample TSC. In RC-based TSC models, a significant challenge lies in how the classifier can effectively utilize the rich temporal representations generated by the reservoir to produce accurate classification results. In this paper, we propose a novel subsampling strategy for reservoir states, termed Quantile-based Positive Proportion (QPP). Unlike existing subsampling strategies, QPP transforms continuous reservoir representations into binary representations using optimizable quantile thresholds and constructs features by computing the proportion of positive activations. Experiments are conducted on 13 small-sample TSC benchmark datasets from the UCR archive using two representative RC models. Comparative results against existing subsampling strategies show that the proposed QPP consistently achieves superior average rankings across datasets, which highlights the high effectiveness and generalization capability of QPP with respect to RC models in dealing with small-sample TSC tasks.**

***Index Terms*—Machine learning, Recurrent neural network, Time-series processing, Temporal pooling**

I. Introduction

Time-series classification (TSC) [1] has become a fundamental task in machine learning, enabling the automated analysis of sequential data across various domains, including speech recognition [2], physiological signal identification [3], and sensor-based activity recognition [4]. The goal of TSC is to learn discriminative temporal representations from training data that can effectively distinguish temporal patterns across categories.

In many practical applications, however, the available number of training samples per class is extremely limited, leading to the so-called small-sample TSC problem [5]. Such situations frequently arise in medical diagnosis, fault detection, and environmental monitoring, where data collection is costly or time-consuming. The scarcity of training examples poses a severe challenge for traditional data-hungry neural network models, which rely heavily on large-scale data to generalize effectively.

Recurrent Neural Networks (RNNs) and their variants, such as Long Short-Term Memory (LSTM) [6] and Gated Recurrent Units (GRU) [7], are classical frameworks for modeling temporal dependencies in TSC tasks. Nevertheless, fully trained RNNs often suffer from instability, gradient vanishing/explosion, and overfitting when trained on small datasets. Their dependence on extensive parameter optimization and backpropagation through time (BPTT) further limits their effectiveness and generalization in edge computing scenarios.

Reservoir Computing (RC) provides a compelling alternative to traditional RNN training [8]. By fixing the recurrent weights in the reservoir and training only a simple readout layer, RC eliminates the need for gradient-based optimization while preserving the ability to model complex temporal dynamics. This approach offers several advantages, including training efficiency, structural simplicity, and potential for neuromorphic or physical implementations [9]. Among its prominent variants, the Echo State Network (ESN) [10] and the recently proposed Euler State Network (EuSN) [11] have demonstrated significant effectiveness in TSC tasks due to their rich dynamical behaviors and powerful temporal representation ability.

Typically, RC-based TSC models either exploit the entire reservoir state sequence or adopt subsampling strategies to construct the final feature representation. When all states are utilized, a winner-take-all (WTA) strategy is commonly employed [12]; however, this approach substantially increases computational cost and may introduce redundancy in long sequences. In contrast, subsampling can effectively reduce training cost and has therefore been widely adopted in both RNN and RC-based architectures. Nevertheless, conventional subsampling schemes, such as selecting or aggregating the maximum, mean, or last state, rely solely on sample-wise (local) information, without incorporating global information from the entire training set into the feature construction process. As a result, the extracted features often exhibit limited

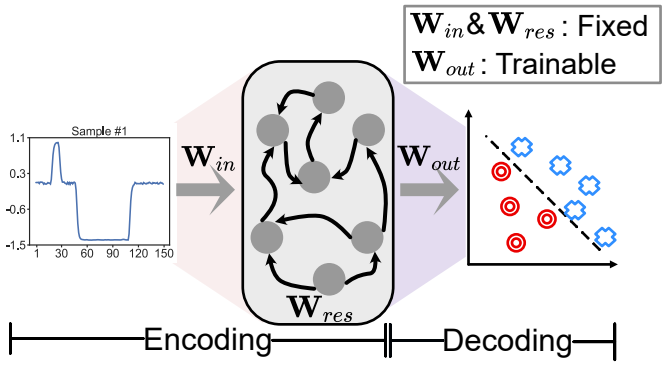

Fig. 1. The basic workflow for RC models in the process of TSC.

discriminability, which constrains the overall classification performance.

Recently, quantile-based subsampling has been proposed and successfully applied in Deep Set Networks for 3D point cloud classification [13]. By preserving representative features from the whole training set, rather than relying on a few extreme or average values from each sample itself, quantile-based methods can achieve relatively higher representation ability.

Inspired by these findings, we propose a quantile-based subsampling strategy, termed Quantile-based Positive Proportion (QPP), for RC frameworks in TSC. Specifically, QPP can be integrated into two representative architectures, including ESN and EuSN. To validate the effectiveness, we conduct comprehensive evaluations across 13 small-sample TSC datasets. Experimental results show that QPP outperforms classical strategies in terms of mean classification accuracy ranking. These findings demonstrate the strong compatibility between QPP and RC models, establishing our proposal as an effective and generalizable strategy in dealing with small-sample TSC tasks with respect to RC architectures.

The rest of this paper is organized as follows: The preliminaries about the TSC and two representative RC methods for TSC are introduced in Section II. Our proposal is described in Section III. The details of experiments are presented in Section IV. The conclusion is given in Section V.

II. Preliminaries

For clarity, we use lowercase bold letters and uppercase bold letters to denote vectors and matrices, respectively.

We denote a TSC dataset with N_S sample-label pairs as $\left\{\mathbf{U}^{(n)}, \mathbf{y}^{(n)}\right\}_{n=1}^{N_S}$, where $\mathbf{U}^{(n)}$ and $\mathbf{y}^{(n)}$ represent the n-th time-series sample and the corresponding N_C-dimensional bipolar encoding label defined as a 1-of-N_C label vector in $\{-1, 1\}$ coding scheme, respectively. The n-th time-series sample with time length N_T can be expanded as $\mathbf{U}^{(n)} = \left[\mathbf{u}_1^{(n)}, ..., \mathbf{u}_{N_T}^{(n)}\right] \in \mathbb{R}^{N_U \times N_T}$, where $\mathbf{u}_t^{(n)} \in \mathbb{R}^{N_U}$ is the corresponding N_U-dimensional input vector at time t. The basic workflow of RC models for TSC is illustrated in Fig. 1. We can see that the whole computation process is composed of the encoding and decoding processes [14], [15].

In the encoding process, two predetermined weight matrices, $\mathbf{W}_{in} \in \mathbb{R}^{N_R \times N_U}$ and $\mathbf{W}_{res} \in \mathbb{R}^{N_R \times N_R}$, are engaged in calculating the high-dimensional reservoir state matrix $\mathbf{X}^{(n)} \in \mathbb{R}^{N_R \times N_T}$ corresponding to the n-th sample, where N_R is the size of the reservoir. In the decoding process, the reservoir state matrix is classified into a desired category by a simple linear transformation with a trainable readout weight matrix $\mathbf{W}_{out} \in \mathbb{R}^{N_C \times N_R}$. The corresponding training process can be formulated by minimizing the cost function as follows:

$$\hat{\mathbf{W}}_{out} = \underset{\mathbf{W}_{out}}{\arg\min} \frac{1}{N_S^{tr}} \sum_{n=1}^{N_S^{tr}} g\left(\mathbf{y}^{(n)}, \hat{\mathbf{y}}^{(n)}\right), \tag{1}$$

where N_S^{tr} is the number of training samples and the symbol $g(\cdot)$ is a certain loss function such as mean squared error (MSE). The vector $\hat{\mathbf{y}}^{(n)} \in \mathbb{R}^{N_C}$ is the predicted output that can be calculated as follows:

$$\hat{\mathbf{y}}^{(n)} = \mathbf{W}_{out}\tilde{\mathbf{x}}^{(n)}, \tag{2}$$

where $\tilde{\mathbf{x}}^{(n)} \in \mathbb{R}^{N_R}$ is a subsampled reservoir state corresponding to the n-th sample by applying sub-sampling functions (such as max and mean) on $\mathbf{X}^{(n)}$ over the time-series sample of length N_T. The matrix $\mathbf{W}_{out}$ can be calculated in a closed-form expression of the ridge regression as follows:

$$\mathbf{W}_{out} = \mathbf{Y}\tilde{\mathbf{X}}^{\top}\left(\tilde{\mathbf{X}}\tilde{\mathbf{X}}^{\top} + \lambda\mathbf{I}\right)^{-1}, \tag{3}$$

where $\mathbf{Y} \in \mathbb{R}^{N_C \times N_S}$ denotes the label matrix and $\tilde{\mathbf{X}} \in \mathbb{R}^{N_R \times N_S}$ represents the subsampled reservoir state matrix. Here, $\mathbf{I} \in \mathbb{R}^{N_R \times N_R}$ is the identity matrix, and λ is the regularization coefficient.

In this work, we adopt the encoding mechanisms of ESN and EuSN to examine the effectiveness of different subsampling strategies. The ESN represents a reservoir with short-term memory encoding, while the EuSN represents one with long-term memory encoding, enabling a comprehensive and balanced evaluation setting. The encoding process of ESN can be written as follows:

$$\mathbf{x}_t^{(n)} = (1-\alpha)\,\mathbf{x}_{t-1}^{(n)} + \alpha\tanh\left(\mathbf{W}_{in}\mathbf{u}_t^{(n)} + \mathbf{W}_{res}\mathbf{x}_{t-1}^{(n)}\right), \tag{4}$$

where the parameter α means the leaking rate. The element values of $\mathbf{W}_{in}$ are randomly selected from a uniform distribution with a range of $[-\theta, \theta]$ where θ is the corresponding input scaler. The reservoir weight matrix $\mathbf{W}_{res}$ is initialized as a sparse random matrix with values drawn from a uniform distribution with a range of $[-1, 1]$, after which it is rescaled to have a desired spectral radius ρ. The encoding process of EuSN can be written as follows:

$$\mathbf{x}_t^{(n)} = \mathbf{x}_{t-1}^{(n)} + \alpha\tanh\left(\mathbf{W}_{in}\mathbf{u}_t^{(n)} + \mathbf{W}_{res}\mathbf{x}_{t-1}^{(n)}\right), \tag{5}$$

where $\mathbf{W}_{res}$ in the EuSN is set as follows:

$$\mathbf{W}_{res} = \mathbf{W}_x - \gamma\mathbf{I}. \tag{6}$$

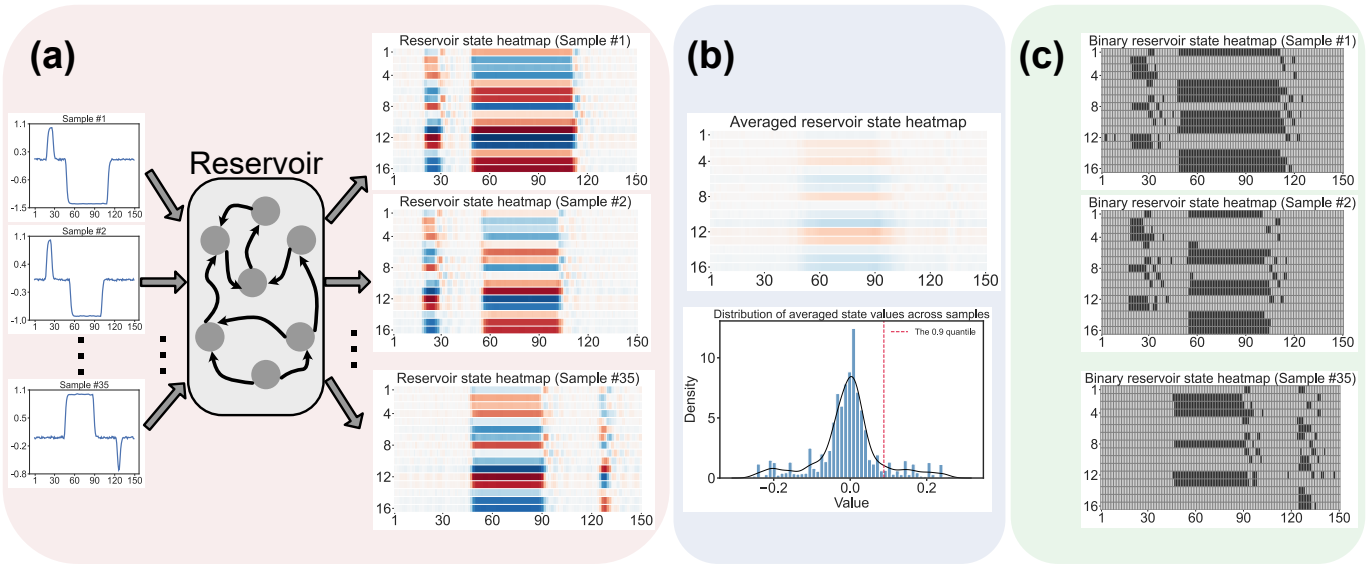

Fig. 2. Three key procedures for TSC using RC-based models integrated with the proposed QPP strategy. (a) Extracting reservoir states corresponding to the training samples. (b) Determining quantile-based thresholds from the averaged reservoir state computed over all training samples. (c) Transforming continuous reservoir-state values into binary representations.

The parameter γ is a diffusion coefficient whose value is a small positive value, and the matrix $\mathbf{W}_x$ is an antisymmetric matrix that satisfies the following condition [11]:

$$\mathbf{W}_x = -\mathbf{W}_x^\top, \tag{7}$$

where $\mathbf{W}_x$ can be calculated as $\mathbf{W}_x = \mathbf{W}_\epsilon - \mathbf{W}_\epsilon^\top$ and the element values of $\mathbf{W}_\epsilon$ are randomly chosen from a uniform distribution with a range of $[-\epsilon, \epsilon]$ with a scaling coefficient ϵ.

III. The Proposed Method

Unlike classical subsampling methods, such as max, last, and mean subsampling strategies, which compute a subsampled vector solely from reservoir states obtained from each time series, QPP generates subsampled features by counting the ratio of values that exceed a quantile-based threshold derived from the global mean reservoir state. In Fig. 2, we illustrate how QPP operates on the reservoir states produced by the encoding stage. During the training phase, based on the reservoir states as shown in Fig. 2(a), we calculate a mean reservoir state matrix $\bar{\mathbf{X}} \in \mathbb{R}^{N_R \times N_T}$ as shown in Fig. 2(b), which can be formulated as follows:

$$\bar{\mathbf{X}} = \frac{1}{N_S^{tr}} \sum_{n=1}^{N_S^{tr}} \mathbf{X}^{(n)}. \tag{8}$$

Based on $\bar{\mathbf{X}}$, we calculate the binary reservoir state matrix $\mathbf{B}^{(n)} \in \mathbb{R}^{N_R \times N_T}$ for $\mathbf{X}^{(n)}$ as follows:

$$\mathbf{B}_{i,j}^{(n)} = \begin{cases} 1, & \text{if } \mathbf{X}_{i,j}^{(n)} > Q_\tau\big(\text{vec}(\bar{\mathbf{X}})\big), \\ 0, & \text{otherwise}, \end{cases} \tag{9}$$

where the subscript (i, j) of a matrix indicates its (i, j) element (e.g., $\mathbf{B}_{i,j}^{(n)}$ and $\mathbf{X}_{i,j}^{(n)}$). The function vec$(\cdot)$ denotes the vectorization operator that flattens a matrix into a vector. The function $Q_\tau(\cdot)$ represents the τ-quantile function, which outputs the value of the input element corresponding to the τ-quantile. Depending on the training strategy of the decoder, τ can be fine-tuned using various approaches, such as heuristic algorithms or backpropagation-based algorithms. The subsampled reservoir state vector by QPP can be calculated as follows:

$$\tilde{\mathbf{x}}^{(n)} = \frac{1}{N_T} \sum_{j=1}^{N_T} \mathbf{B}_{:,j}^{(n)}. \tag{10}$$

By flexibly combining subsampled vectors obtained from $m \in \mathbb{Z}^+$ different quantile functions (i.e., $\{Q_{\tau_1}(\cdot), ..., Q_{\tau_m}(\cdot)\}$), QPP constructs concatenated temporal features that significantly enhance the computational ability of RC models in dealing with TSC tasks (See Fig. 3 in Section IV-E).

IV. Experiments

A. Dataset details

To comprehensively evaluate the effectiveness of our proposed subsampling strategy under small-sample conditions, we conduct experiments on a subset of the UCR2018 time series classification archive [16]. As shown in Table I, these datasets are characterized by the number of training samples per class being less than 10 ($N_S^{tr}/N_C < 10$), making them challenging benchmarks for assessing the computational ability of reservoir-based models on few-training-sample scenarios. The number of classes varies from 3 to 52, and the sequence lengths vary considerably from 270 to 2000 time steps. This diversity allows our evaluation to include both short and long sequences, simple and complex temporal patterns, and tasks with low and high class cardinality.

TABLE I
OVERVIEW OF UCR2018 DATASETS CHARACTERIZED BY FEWER THAN TEN TRAINING SAMPLES PER CLASS.

	#Train	#Test	#Classes	#Timesteps
Beef	30	30	5	470
DiatomSizeReduction	16	306	4	345
FaceFour	24	88	4	350
FiftyWords	450	455	50	270
InsectEPGSmallTrain	17	249	3	601
Mallat	55	2345	8	1024
OliveOil	30	30	4	570
Phoneme	214	1896	39	1024
PigAirwayPressure	104	208	52	2000
PigArtPressure	104	208	52	2000
PigCVP	104	208	52	2000
Rock	20	50	4	2844
Symbols	25	995	6	398

B. Subsampling strategies for comparison

To verify the effectiveness of our proposed method, we compare it with representative statistical subsampling strategies commonly used in reservoir computing models.

- **Max**: This strategy selects the maximum activation over the temporal dimension for each reservoir unit. It captures the most salient maximal response of the reservoir dynamics and is often used to highlight strong activation patterns in the sequence.
- **Mean**: This strategy computes the average activation across N_T timesteps. It produces a smoothed global summary of the reservoir state, reducing local fluctuations while preserving overall temporal trends.
- **Last**: This strategy takes the final reservoir state of the sequence. It relies on the fading memory property of the reservoir, assuming that information from earlier time steps is implicitly preserved within the reservoir state of the last timestep.

Since our proposed strategy is regarded as an optimizable subsampling strategy for RC models, to examine whether the proposed method outperforms other optimizable subsampling strategies, we include the following backpropagation-based strategy in the comparison:

- **Attn**: The attention-based subsampling strategy is considered a promising information aggregation approach with optimizable coefficients. Its main idea is to learn an importance coefficient for each time step and to generate a weighted combination of reservoir states across all timesteps, allowing the model to emphasize informative reservoir states while suppressing irrelevant ones [17]. The weighted combination of the reservoir state corresponding to the n-th sample can be calculated as follows:

$$\tilde{\mathbf{x}}^{(n)} = \sum_{t=1}^{N_T} c_t \mathbf{x}_t^{(n)}, \tag{11}$$

where c_t is the coefficient at time t. This coefficient can be computed as follows:

$$c_t = \frac{\exp\left(\mathbf{w}^\top \mathbf{x}_t\right)}{\sum_{k=1}^{N_T} \exp\left(\mathbf{w}^\top \mathbf{x}_k\right)}, \tag{12}$$

where $\mathbf{w} \in \mathbb{R}^{N_R}$ is a learnable weight vector.

In addition, we set an aggregated strategy as a counterpart of the above-mentioned subsampling strategies:

- **WTA**: This strategy maps each reservoir state at every timestep to a bipolar target vector via a linear regression decoder. The outputs are then summed across timesteps, and the class with the maximum accumulated score is chosen as the final prediction [18], effectively transforming local inference signals into a global decision. The predicted output corresponding to the n-th sample for the WTA strategy can be calculated as follows:

$$\hat{\mathbf{y}}^{(n)} = \sum_{t=1}^{N_T} \hat{\mathbf{y}}_t^{(n)}, \tag{13}$$

where $\hat{\mathbf{y}}_t^{(n)} \in \mathbb{R}^{N_C}$ is the local predicted output at time t corresponding to the n-th sample.

C. Parameter settings

To comprehensively evaluate the effectiveness of the proposed subsampling strategy against existing baselines, we employed two representative RC models introduced in Section II: ESN and EuSN, which respectively characterize short-term and long-term memory dynamics. These two models are used as testbeds throughout all experiments. To ensure fair and reliable comparisons, we performed hyperparameter optimization for both ESN and EuSN using the Particle Swarm Optimization (PSO) method as adopted in Ref. [19]. For initial settings of PSO, the number of particles and iterations were fixed at 80 and 30, respectively. For the parameters of the PSO algorithm, we set both the cognitive coefficient and the social coefficient to 2. The inertia weight ω^j at the j-th iteration is updated as follows:

$$\omega^j = \omega_{max} - (\omega_{max} - \omega_{min}) \times j / N_J, \tag{14}$$

where the maximal weight ω_{max} and the minimal weight ω_{min} were set to 0.9 and 0.6, respectively. The symbol N_J represents the number of iterations.

We show the searching ranges of each hyperparameter for the ESN and the EuSN in Table II. We searched for the optimal reservoir size in the range of $[32, 64, 128]$ for the two RC models on each dataset.

TABLE II
THE RANGES OF HYPERPARAMETERS SEARCHED BY THE PSO ALGORITHM FOR TWO RC MODELS.

	Symbol	Range
The ESN-based encoder	α	$[0, 0.99]$
	ρ	$[0.6, 1.3]$
	θ	$[0.05, 5]$
The EuSN-based encoder	α	$[10^{-5}, 1]$
	γ	$[10^{-5}, 1]$
	θ	$[10^{-3}, 10^{+1}]$
	ϵ	$[10^{-3}, 10^{+1}]$
The decoder	λ	$[5 * 10^{-5}, 5 * 10^{-1}]$

For the attention-based strategy, we employed a dense layer to transform the weighted combination of the reservoir states

into the desired category. We used the cross-entropy function to evaluate the difference between the true labels and model predictions. The Stochastic Gradient Descent (SGD) [20] was leveraged for training $\mathbf{w}$ in (12). We set the number of training epochs to 20 and the learning rate to 0.1.

For the proposed QPP, we varied m from 1 to 4 for each candidate reservoir size. The corresponding $\{\tau_m\}_{m=1}^{4}$ are searched by PSO with the searching range $(0, 1)$. We hereby use an open set to avoid the occurrence of an all-zero or all-one $\mathbf{B}^{(n)}$ in (9).

D. Metrics

We used the classification accuracy as adopted in Ref. [11] across the whole evaluation, which can be formulated as follows:

$$\text{Accuracy} = \frac{N_{\text{correct}}}{N_S^{te}} \times 100\%, \tag{15}$$

where N_{correct} denotes the number of correctly classified test samples and N_S^{te} denotes the total number of test samples. All reported numerical results are obtained by averaging over ten trials using different random seeds for both ESN and EuSN.

E. Results

We show the 5-fold cross-validation performances for the two RC models with the proposed QPP on the 13 datasets in Fig. 3. By observing two aspects, the variation performance with respect to the reservoir size N_R and the number of quantiles m, we can observe that the best validation performance of each dataset for both two RC models generally occurs with a relatively larger reservoir size ($N_R = 128$), except for the Phoneme dataset in Fig. 3(b). In addition, the highest validation accuracy for each dataset is achieved when $m > 1$ for both ESN and EuSN. These observations indicate that QPP allows richer temporal features associated with larger reservoirs to be retained, while also enhancing the representation ability of two RC models for classification tasks by concatenating subsampled features corresponding to different quantile values.

We report the test performances of ESN and EuSN under the parameter settings that achieve the best validation performance in Tables III and IV, respectively. As shown in Table III, ESN with the QPP strategy outperforms the other competing strategies on 11 datasets, except for InsectEPGSmallTrain and Mallat datasets. Similarly, Table IV shows that EuSN with QPP achieves superior performance on 11 datasets compared with the other strategies, except for InsectEPSSmallTrain and Rock datasets. In particular, in terms of the average classification accuracies on PigAirwayPressure, PigArtPressure, and PigCVP datasets, both two RC models combined with the QPP strategy perform markedly better than those using other strategies. This suggests that QPP is more effective than other strategies in enabling RC models to achieve improved classification performance when only an extremely small number of training samples per class are available.

To statistically compare the performance of different strategies combined with two RC models, we employed the Friedman test with the Nemenyi post-hoc test at a significance level of 0.05 [21]. The comparison results for ESN and EuSN are visualized using Critical Difference (CD) diagrams as shown in Figs. 4(a) and (b), respectively, which are specifically designed to illustrate the mean rank of each strategy across all evaluated datasets and to indicate whether the observed differences are statistically significant. As observed from Figs. 4(a) and (b), QPP significantly outperforms Attn., Mean, Last, and WTA strategies on both ESN and EuSN models. Although the difference between QPP and Max does not exceed the CD threshold, QPP consistently attains a better mean rank than Max in both CD diagrams. These results reflect that the QPP strategy has superior overall performance across the entire evaluation and high robustness with respect to RC model variations. Among the remaining strategies, the Max strategy achieves the second-best overall performance on the two RC models. For the attention-based strategy, although it explicitly fits training-set information, its overall performance is less competitive than QPP and Max, possibly due to overfitting in small-sample settings, where the attention-based subsampling strategy may fail to generalize effectively.

F. Computational complexity analysis

As introduced in Section II, the entire procedure of RC-based models for dealing with TSC tasks can be divided into two processes, including encoding and decoding processes. In the training phase, all tested models share the same encoding process, which can be formulated as

$$O\left(N_S^{tr} N_T \left(N_R N_U + N_R^2\right)\right). \tag{16}$$

We separate the decoding phase into subsampling and readout training stages. For the subsampling stage, the strategies of Max and Mean share the same complexity as $O\left(N_S^{tr} N_R N_T\right)$, the strategy Last costs $O\left(N_S^{tr} N_R\right)$. Assuming that the training epoch of the attention-based subsampling strategy is E, the corresponding complexity is $O\left(E N_S^{tr} N_R N_T\right)$. The proposed strategy QPP costs $O\left(m N_S^{tr} N_R N_T\right)$.

For the readout training stage, the complexities of the readout training phase corresponding to the subsampling strategies Max, Mean, and Last cost $O\left(N_S^{tr} N_R^2 + N_R^3\right)$. The complexity of fine-tuning the linear classifier corresponding to the strategy Attn. is $O\left(E N_S^{tr} N_R N_C\right)$. The complexity of training the readout layer for the strategy WTA is $O\left(N_S^{tr} N_T N_R^2 + N_R^3\right)$. For our proposed strategy, the corresponding complexity cost of training readout is $O\left(m^2 N_S^{tr} N_R^2 + m^3 N_R^3\right)$.

Compared with Max, Mean, and Last, QPP introduces only a moderate additional cost in the subsampling stage due to its linear dependence on the number of quantile levels m. Although the readout complexity increases polynomially with m due to the expanded feature dimension, this overhead remains limited since m is typically small. In contrast to Attn, QPP avoids iterative optimization and thus eliminates the multi-epoch training overhead. Compared with WTA, QPP adopts a compress-then-classify strategy, which keeps the

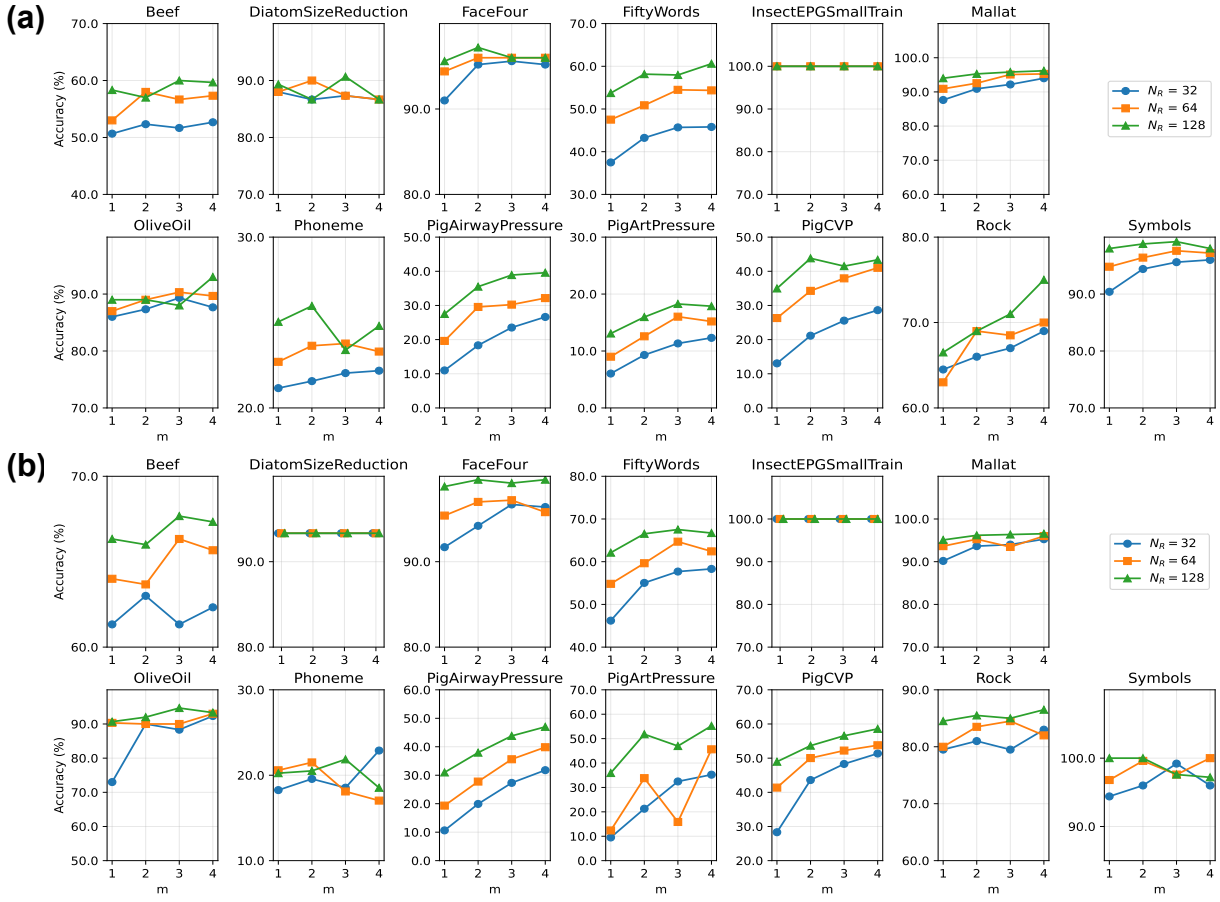

Fig. 3. The 5-fold cross-validation performances of the two RC models with the proposed QPP under different values of m. (a) ESN and (b) EuSN.

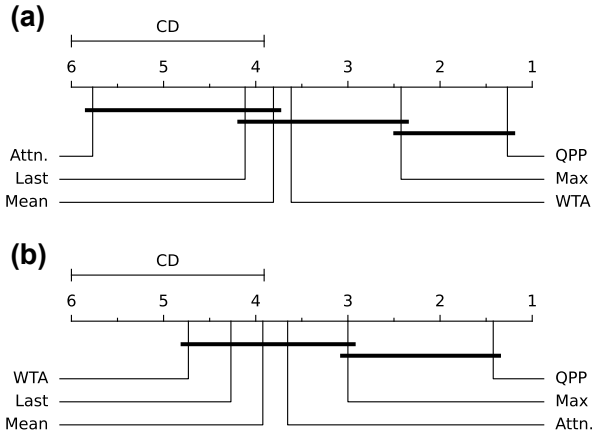

Fig. 4. Critical distance diagrams for the two RC models with six evaluated strategies. (a) ESN and (b) EuSN.

readout complexity independent of the sequence length N_T and avoids redundant timestep-wise output computation. Overall, the proposed QPP achieves an effective balance between computational efficiency and representational capacity, making it a scalable subsampling strategy for RC-based models in small-sample TSC tasks.

V. Conclusion

In this paper, we proposed a quantile-based subsampling strategy that is compatible with two representative RC models for TSC. The proposed strategy leverages optimizable quantiles to determine optimal threshold values from the averaged reservoir states generated from training samples, thereby enhancing the discriminative ability of features used for the final classification in RC-based models. The corresponding experimental results demonstrated that the proposed strategy outperforms the other evaluated strategies in terms of average classification performance ranking, validating the effectiveness of the proposed approach. From a computational perspective, QPP introduces only moderate overhead while avoiding iterative optimization and redundant timestep-wise computation, resulting in a scalable and efficient solution. Overall, the proposed method achieves a favorable balance between efficiency and performance.

Acknowledgment

This work was partly supported by JSPS KAKENHI Grant Numbers JP23K28154 (GT), JP25H00451 (GT), JST Moonshot R&D Grant Number JPMJMS2021 (GT), and JST CREST Grant Number JPMJCR24R2 (GT).

TABLE III
THE MEAN ACCURACIES AND THE CORRESPONDING STANDARD DEVIATIONS FOR ESN WITH SIX STRATEGIES ON 13 DATASETS.

	QPP	Max	Last	WTA	Mean	Attn.
Beef	**76.00±(6.11)**	61.67±(5.82)	57.33±(5.12)	62.67±(4.16)	54.33±(7.46)	30.00±(4.71)
DiatomSizeReduction	**93.95±(1.93)**	88.89±(1.50)	87.65±(0.41)	84.25±(2.93)	70.62±(0.45)	30.07±(0.00)
FaceFour	**90.00±(2.08)**	85.45±(5.72)	53.30±(2.41)	86.36±(3.13)	73.64±(7.69)	35.80±(8.79)
FiftyWords	**67.69±(1.25)**	66.22±(1.49)	63.34±(1.06)	47.52±(1.41)	52.37±(2.11)	15.56±(2.28)
InsectEPGSmallTrain	100.00±(0.00)	100.00±(0.00)	100.00±(0.00)	100.00±(0.00)	100.00±(0.00)	100.00±(0.00)
Mallat	92.63±(0.62)	**92.92±(1.43)**	65.63±(5.44)	78.08±(1.02)	78.85±(0.72)	12.02±(1.34)
OliveOil	**79.33±(1.33)**	70.67±(9.64)	50.67±(8.92)	40.00±(0.00)	53.33±(0.00)	40.00±(0.00)
Phoneme	**26.29±(0.68)**	22.50±(0.52)	13.24±(0.18)	15.93±(0.53)	16.51±(0.62)	12.67±(1.55)
PigAirwayPressure	**73.56±(7.02)**	20.53±(2.78)	3.61±(0.44)	36.01±(2.94)	17.12±(1.41)	1.73±(0.58)
PigArtPressure	**54.04±(3.13)**	15.53±(2.26)	2.50±(0.36)	18.61±(2.03)	5.00±(1.43)	1.83±(0.29)
PigCVP	**90.38±(2.97)**	64.81±(5.12)	6.54±(0.78)	53.08±(3.04)	53.51±(3.57)	2.98±(1.05)
Rock	**56.60±(3.69)**	52.20±(5.83)	53.40±(3.58)	37.60±(2.33)	37.20±(2.71)	29.20±(4.40)
Symbols	**94.80±(1.05)**	94.68±(1.15)	77.29±(4.45)	71.72±(1.56)	79.37±(4.09)	25.72±(5.32)

TABLE IV
THE MEAN ACCURACY AND THE CORRESPONDING STANDARD DEVIATION FOR EUSN WITH SIX STRATEGIES ON 13 DATASETS.

	QPP	Max	Last	WTA	Mean	Attn.
Beef	**76.33±(3.14)**	55.33±(5.62)	25.00±(6.87)	55.67±(5.39)	20.00±(4.22)	53.33±(4.47)
DiatomSizeReduction	**96.47±(0.73)**	85.49±(0.72)	92.29±(0.72)	74.71±(7.30)	80.85±(0.66)	91.34±(2.98)
FaceFour	**94.09±(1.22)**	80.34±(3.97)	26.25±(4.14)	33.07±(0.94)	40.57±(0.89)	85.91±(6.35)
FiftyWords	**73.16±(0.98)**	45.93±(1.48)	18.40±(2.60)	21.38±(0.44)	13.63±(2.10)	43.52±(2.57)
InsectEPGSmallTrain	100.00±(0.00)	98.39±(1.24)	100.00±(0.00)	100.00±(0.00)	95.90±(5.25)	100.00±(0.00)
Mallat	**93.66±(0.57)**	55.72±(3.18)	84.00±(2.55)	23.81±(3.36)	84.60±(1.96)	63.36±(3.99)
OliveOil	**81.67±(3.42)**	71.00±(4.23)	43.00±(3.14)	40.33±(3.14)	44.00±(4.67)	55.00±(10.98)
Phoneme	**17.70±(0.26)**	16.22±(0.55)	7.00±(0.33)	7.46±(0.74)	8.84±(0.55)	11.11±(0.34)
PigAirwayPressure	**87.21±(1.85)**	13.65±(1.45)	5.82±(1.04)	1.88±(0.85)	10.82±(0.94)	3.41±(1.04)
PigArtPressure	**97.69±(0.71)**	20.53±(2.81)	7.55±(11.24)	3.51±(1.08)	3.75±(0.91)	2.40±(0.91)
PigCVP	**95.34±(0.86)**	3.32±(1.66)	3.85±(4.24)	1.87±(1.15)	4.71±(1.37)	2.69±(1.12)
Rock	46.40±(5.85)	**53.40±(7.43)**	39.80±(4.24)	51.00±(5.16)	50.4±(5.12)	47.80±(5.47)
Symbols	**96.58±(0.61)**	66.54±(2.82)	51.98±(5.39)	49.32±(1.13)	79.23±(2.86)	75.69±(5.49)

REFERENCES

[1] J. Faouzi, "Time series classification: A review of algorithms and implementations," *Machine Learning (Emerging Trends and Applications)*, 2022.

[2] H. Kheddar, M. Hemis, and Y. Himeur, "Automatic speech recognition using advanced deep learning approaches: A survey," *Information Fusion*, vol. 109, p. 102422, 2024.

[3] Z. Li, K. Fujiwara, and G. Tanaka, "An echo state network-based method for identity recognition with continuous blood pressure data," in *International Conference on Artificial Neural Networks*. Springer, 2023, pp. 13–25.

[4] A. Saha, S. Rajak, J. Saha, and C. Chowdhury, "A survey of machine learning and meta-heuristics approaches for sensor-based human activity recognition systems," *Journal of Ambient Intelligence and Humanized Computing*, vol. 15, no. 1, pp. 29–56, 2024.

[5] J.-J. Liu, J.-P. Yao, Z. Wang, Z.-Y. Wang, and L. Huang, "Small sample time series classification based on data augmentation and semi-supervised learning," *Information Technology and Control*, vol. 53, no. 2, pp. 470–491, 2024.

[6] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[7] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *arXiv preprint arXiv:1409.1259*, 2014.

[8] M. Cucchi, S. Abreu, G. Ciccone, D. Brunner, and H. Kleemann, "Hands-on reservoir computing: a tutorial for practical implementation," *Neuromorphic Computing and Engineering*, vol. 2, no. 3, p. 032002, 2022.

[9] G. Tanaka, T. Yamane, J. B. Héroux, R. Nakane, N. Kanazawa, S. Takeda, H. Numata, D. Nakano, and A. Hirose, "Recent advances in physical reservoir computing: A review," *Neural Networks*, vol. 115, pp. 100–123, 2019.

[10] M. Lukoševičius, "A practical guide to applying echo state networks," in *Neural Networks: Tricks of the Trade: Second Edition*. Springer, 2012, pp. 659–686.

[11] C. Gallicchio, "Euler state networks: Non-dissipative reservoir computing," *Neurocomputing*, vol. 579, p. 127411, 2024.

[12] A. Rodan and P. Tino, "Minimum complexity echo state network," *IEEE Transactions on Neural Networks*, vol. 22, no. 1, pp. 131–144, 2010.

[13] Z. Chen, X. Zhu, D. Su, and J. C. Chuang, "Stacking deep set networks and pooling by quantiles," in *Forty-first International Conference on Machine Learning*, 2024.

[14] Z. Li and G. Tanaka, "Multi-reservoir echo state networks with sequence resampling for nonlinear time-series prediction," *Neurocomputing*, vol. 467, pp. 115–129, 2022.

[15] Z. Li, Y. Liu, and G. Tanaka, "Multi-reservoir echo state networks with hodrick–prescott filter for nonlinear time-series prediction," *Applied Soft Computing*, vol. 135, p. 110021, 2023.

[16] H. A. Dau, A. Bagnall, K. Kamgar, C.-C. M. Yeh, Y. Zhu, S. Gharghabi, C. A. Ratanamahatana, and E. Keogh, "The UCR time series archive," *IEEE/CAA Journal of Automatica Sinica*, vol. 6, no. 6, pp. 1293–1305, 2019.

[17] M. J. Er, Y. Zhang, N. Wang, and M. Pratama, "Attention pooling-based convolutional neural network for sentence modelling," *Information Sciences*, vol. 373, pp. 388–403, 2016.

[18] M. D. Skowronski and J. G. Harris, "Automatic speech recognition using a predictive echo state network classifier," *Neural Networks*, vol. 20, no. 3, pp. 414–423, 2007.

[19] Z. Li, R. S. Fong, K. Fujiwara, K. Aihara, and G. Tanaka, "Structuring multiple simple cycle reservoirs with particle swarm optimization," in *2025 International Joint Conference on Neural Networks (IJCNN)*, 2025, pp. 1–9.

[20] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proceedings of COMPSTAT'2010: 19th International Conference on Computational StatisticsParis France, August 22-27, 2010 Keynote, Invited and Contributed Papers*. Springer, 2010, pp. 177–186.

[21] M. Hollander, D. A. Wolfe, and E. Chicken, *Nonparametric Statistical Methods*. John Wiley & Sons, 2013.