Neural Computing and Applications (2021) 33:3169–3182
https://doi.org/10.1007/s00521-020-05179-w

ORIGINAL ARTICLE

Temporal representation learning for time series classification

Yupeng Hu[1] · Peng Zhan[4] · Yang Xu[2] · Jia Zhao[3] · Yujun Li[3] · Xueqing Li[4]

Received: 9 March 2020 / Accepted: 11 July 2020 / Published online: 21 July 2020

Abstract
Recent years have witnessed the exponential growth of time series data as the popularity of sensing devices and development of IoT techniques; time series classification has been considered as one of the most challenging studies in time series data mining, attracting great interest over the last two decades. According to the empirical evidences, temporal representation learning-based time series classification has more superiority of accuracy, efficiency and interpretability as compared to hundreds of existing time series classification methods. However, due to the high time complexity of feature process, the performance of these methods has been severely restricted. In this paper, we first presented an efficient shapelet transformation method to improve the overall efficiency of time series classification, and then, we further developed a novel enhanced recurrent neural network model for deep representation learning to further improve the classification accuracy. Experimental results on typical real-world datasets have justified the superiority of our models over several shallow and deep representation learning competitors.

Keywords Machine learning · Recurrent neural network · Deep representation learning · Turning points evaluation · Time series classification

1 Introduction

Nowadays, time series classification (TSC) has been attracting great interest over the past decade. Recent empirical evidence has demonstrated the advantages of shapelet-based TSC methods in terms of accuracy, efficiency and interpretability [19, 24]. Shapelet, the specific representative subsequence in a certain original time series, does play a pivotal role in TSC. In order to make the presentation of shapelet more intuitive, a representative TSC problem is shown in Fig. 1.

As shown in Fig. 1, there are two types of time series: one is the GUN time series including the shapelet with green solid line and the other is the NoGUN time series containing the shapelet with red solid line. With the help of the above shapelets, whether the boy has a gun in his hand can be classified by many off-the-shelf general classification methods, such as C4.5, 1NN, Native Bayes and Rot Forest. Moreover, the efficiency of the shapelets-based TSC method is more than one order of magnitude faster than the traditional TSC methods based on the entire time series. Last but not least, the appropriate shapelets can make the final classification result more interpretable. As shown in Fig. 1, the shapelet of Gun time series has a clear upward fluctuation, indicating the action of inserting the gun into the holster. Similarly, the shapelet of NoGun time series has an obvious downward fluctuation, which could be explained by the inertia phenomenon "overshoot" introduced by Lexiang and Eamonn [19].

Although shapelets do have obvious advantages in TSC, there is still a challenge on selecting these representative subsequences (shapelets) from the specific time series and even the whole dataset effectively and efficiently.

✉ Yujun Li
liyujun@sdu.edu.cn

Yupeng Hu
huyupeng@sdu.edu.cn

[1] School of Computer Science and Technology, Shandong University, Qingdao 266237, Shandong, China

[2] School of Information Science and Engineering, Shandong Normal University, Jinan 250358, Shandong, China

[3] School of Information Science and Engineering, Shandong University, Qingdao 266237, Shandong, China

[4] School of Software, Shandong University, Jinan 250101, Shandong, China

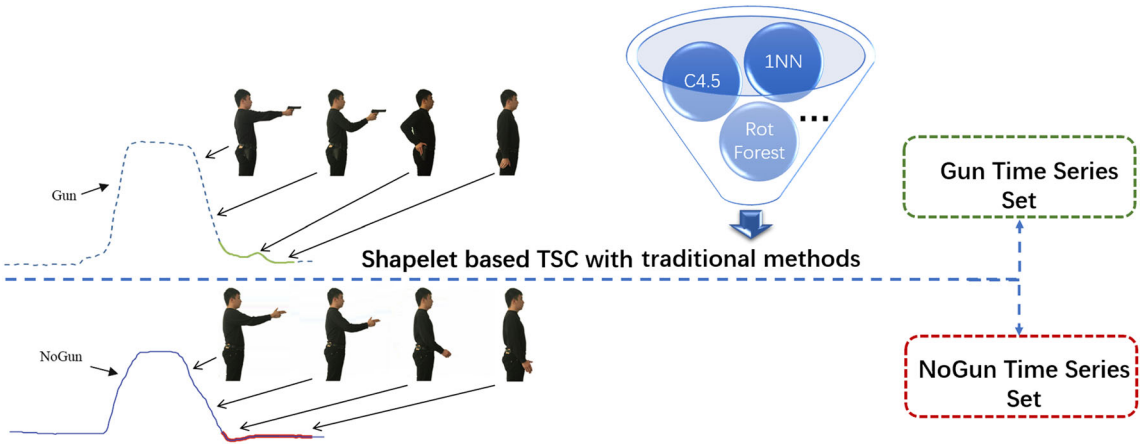


Fig. 1 Shapelet-based time series classification on GunPoint

In view of the superiority of shapelets, scholars have done much work on the shapelet-based TSC methods. Three recent evolutionary TSC algorithms based on efficient shapelets selection are listed as follows.

- *Fast shapelets (FS)*, which employs an extensional decision tree mechanism for accelerating the process of shapelets discovery [26].
- *Shapelet transformation (ST)*, which integrates various classification strategies (SVM, random forest, etc.) into a classification voting mechanism for TSC [18], i.e., ST can incorporate extensive off-the-shelf classification strategies for improving the final TSC accuracy.
- *Learned shapelets (LS)*, which adopts a heuristic gradient descent strategy for searching appropriate shapelets rather than enumerating all possible candidates [11]. Similar with FS, the learning process could not be separated from the entire classification process, both of which are different from ST.

According to the above introduction, we further compare the above three typical TSC methods. The corresponding analyses results are listed in Table 1. According to this table, we have the following findings: (1) the accuracy of FS is lower than these of ST and LS, whose accuracy are basically same introduced by Anthony et al. [3]. (2) The efficiency of FS, LS and ST decreases in order [1]. (3) ST can be integrated by some ensemble algorithms: Collective Of Transformation Ensembles (COTE) [1], Hierarchical Vote COTE (Hive-COTE) [15], etc. However, neither LS nor FS have been adopted by some kinds of ensemble TSC methods. And (4) Although ST has better accuracy, the efficiency of ST is slower than other methods, which would definitely obstruct the popularity of ST for more extensive application scenarios.

Motivated by the above analysis, in this paper, we proposed a novel shapelet transformation method to improve the efficiency of ST. Different from previous acceleration strategies, our algorithm not only focus on reducing the time complexity of evaluation, but also cutting down the number of shapelet candidates as much as possible. The primal shapelets selection strategy in ST is replaced by our temporal feature-based shapelet generation to form a novel efficient shapelet transformation (EST) for TSC to greatly improve the TSC efficiency, while retaining the corresponding classification accuracy in the same level of ST.

More importantly, due to the fact that all the above methods represent the temporal feature in non-deep learning manner for TSC, which are named as shallow representation learning methods in this paper. Accordingly, we presented a novel deep representation learning method for TSC. In particular, we designed a TempOral

Table 1 Comprehensive comparison on FS, ST and LS

Method	Train time[a]	Train space	Accuracy (ranks)[b]	Combinableness
FS	$O(nm^2)$	$O(nm^2)$	2.7419	No
ST	$O(n^2m^4)$	$O(kn)$	1.8548	COTE, Hive-COTE
LS	$O(en^2m^2k^2)$	$O(n^2m^2k^2)$	1.4032	No

[a]In this table, n is the number of time series, m is the length of time series, k is number of shapelets, e is the maximum number of iterations

[b]The average ranks are taken from Anthony et al. [3]

Representation RecurrEnt Neural NeTwork, dubbed as TORRENT, for TSC. TORRENT is an enhanced bidirection LSTM model, which jointly considers the local feature and the corresponding important context information to complete more accurate representation learning for significantly promoting TSC accuracy. The main contributions of this work are threefold:

1. we presented a novel efficient shapelet transformation (EST), which utilizes turning points (TPs) to identify the main temporal features and the overall trend of the sequence for representation learning, while improving the efficiency of TSC.
2. we developed a TempOral Representation RecurrEnt Neural NeTwork (TORRENT) for temporal feature representation learning from scratch. TORRENT can leverage crucial context information to strengthen the feature representation, while boosting the accuracy of TSC.
3. Extensive empirical results on a large number of benchmark time series datasets not only demonstrate EST is more efficient than the main stream shallow representation learning methods, but also verify TORRENT has higher classification accuracy than other deep representation learning methods.

The remainder of this paper is structured as follows. Firstly, an overview of related work is provided in Sect. 2. Secondly, our two TSC methods EST and TORRENT are introduced in details in Sect. 3. Thirdly, comparison experiment results and analyses are presented in Sect. 4. Finally, our conclusions are given in Sect. 5.

2 Related work

2.1 Shallow representation learning methods for TSC

Temporal feature, especially shapelet, based representation learning has been formally proposed for TSC by Lexiang and Eamonn [19]. After that, a large number of shallow representation methods have been proposed, such as: FS and its variants [20, 22, 26], ST and its variants [14, 18], LS and its variants [11, 21]. Considering ST is not only one of the best TSC algorithms in terms of classification accuracy, but also has a broader range of application scenarios, we want to utilize some appropriate acceleration strategies to improve the shapelet selection efficiency of ST, while ensuring the corresponding classification accuracy is in the same level with that of ST. Lexiang and Eamonn [20] proposed two early abandon and entropy pruning-based optimal methods for time series classification . Almost at the same time, Mueen et al. utilized a search space pruning strategy to accelerate the shapelet generation based on the triangle inequality. Considering the above three methods neglect to reduce redundant shapelet candidates, Grabocka et al. [12] presented a scalable shapelet discovery (SD) method to utilize clustering-based pruning strategy for shapelet candidate reduction. Analogously, Isak et al. [13] and Cun et al. [7] separately adopted random shapelet forests and temporal important point evaluation [31] strategies for drastically reducing the redundant number of candidates. Yupeng et al. [32] proposed multi-resolution representation method for shapelet generation and further combined existing general classification models, i.e., rotation forest, support vector machines, etc.[4, 5] for TSC.

2.2 Deep representation learning methods for TSC

Although much progress has been made by the above methods, they all complete TSC in some kinds of shallow representation manners, i.e., they ignored the possible contribution of deep neural networks (DNNs) for TSC, especially DNNs have already achieved great success in representation learning. To the best of our knowledge, there do exist works based on DNNs for TSC; however, these works also have their own problems. Zhiguang et al. [34] separately adopted a multilayer perceptron (MLP), fully convolutional neural network (FCN) and residual network (ResNet) to complete temporal feature representation learning and further TSC. However, relying on a single deep learning model alone, it is impossible to obtain a relatively satisfactory TSC effect. Sangdi and George [25] developed a group-constrained convolutional recurrent neural network (GCRNN), which combines a convolutional network model with a recurrent network component for temporal feature embedding-based TSC. Subsequently, Fazle et al. [10] utilized FCN network [6] as feature representation sub-module for feature encoding and adopted bidirectional long short-term memory network (Bi-LSTM) for TSC.

Although promising classification accuracy has been achieved, these deep representation learning methods do ignore the crucial temporal trends and corresponding contextual information in the original time series, thus fail to achieve more comprehensive feature representation and accurate classification. Therefore, in this paper, we present a novel temporal representation recurrent neural network to leverage crucial context information to strengthen the feature representation, while boosting the accuracy of TSC.

3 Temporal feature-based classification models

In this section, we first give the relevant definitions, and then, we describe the turning points-based shapelet selection strategy for EST. Subsequently, we present deep representation learning-based TORRENT in details.

3.1 Problem formulation

Given a time series dataset D, containing N time series, expressed as

$$D = \{(T_1, L_1), (T_1, L_1), \ldots, (T_i, L_i), \ldots, (T_N, L_N)\} \ 1 \le i \le N \tag{1}$$

where T_i denotes the ith time series of D, expressed as

$$T_i = \{v_1^i, v_2^i, \ldots, v_j^i, \ldots, v_m^i\} \quad 1 \le j \le m \tag{2}$$

where v_j^i denotes the jth point of T_i. Besides, L_i denotes the corresponding one-hot label vector. Without loss of generality, supposing there are K classes of D, L_i is a vector with K element, as

$$L_i = \{l_1^i, l_2^i, \ldots, l_k^i, \ldots, l_K^i\} \quad 1 \le k \le K \tag{3}$$

where each element $l_k^i \in \{0, 1\}$. If $l_k^i = 1$ indicates T_i belongs to the kth class and 0 otherwise.

Definition 1 TSC aims to learn a classifier $\mathcal{C}$ on D to predict the specific class label L_i of given time series T_i, i.e., mapping all the inputs into corresponding class label-based probability distributions.

$$L_i \leftarrow \mathcal{C}(T_i). \tag{4}$$

Empirical evidence implied that instead of the entire T_i, a certain subsequence S_j^i of T_i has distinct temporal significance to represent a certain class L_i can be named as shapelet in Yupeng et al. [32]. Accordingly, it is advisable for us to identify the significant parts according to their own temporal features. The temporal features are constructed by a sequence of data points and each point actually has the different influence on the variation trend [29]. In our paper, we focus on some data points indicating the changing trend of time series, dubbed as turning points (TPs).

Definition 2 For a certain time series T_i. If v_k meets one of the following two inequations, it can be defined as TP in T_i.

$$\begin{aligned} &v_{k-1} < v_k > v_{k+1} \ or \ v_{k-1} < v_k = v_{k+1} \ or \ v_{k-1} = v_k < v_{k+1} \\ &v_{k-1} > v_k < v_{k+1} \ or \ v_{k-1} > v_k = v_{k+1} \ or \ v_{k-1} = v_k > v_{k+1}. \end{aligned} \tag{5}$$

According to Definition 2, the main temporal features in the whole time series could be identified completely. More concretely, an instance for TPs identification on T_i of "Symbols" from TSC benchmark dataset [2] has been given and is shown in Fig. 2.

As shown in Fig. 2a, with the help of Definition 2, 31 TPs in T_i have been identified completely. However, it is obvious that most of TPs are distributed at the peaks and troughs of T_i, i.e., TPs do reflect corresponding importance temporal trends. Therefore, TPs should be evaluated and sorted orderly. After the jth turning point TP_j has been identified, its importance can be evaluated based on the sum of fitting errors, named $\mathrm{Sum}_{\mathrm{FE}}$, between its two adjacent TPs $(\mathrm{TP}_{j-1}, \mathrm{TP}_{j+1})$. Specially, the evaluation of TP_j could be described as follows. Firstly, the left adjacent TP_{j-1} and right adjacent TP_{j+1} would form a straight line named TPLine. Secondly, the fitting error of each raw data point v_k, in the range of $(\mathrm{TP}_{j-1}, \mathrm{TP}_{j+1})$, would be calculated, respectively, which could be obtained by the vertical distance [30] from v_k to TPLine. The value and timestamp of TP_{j-1}, TP_j and TP_{j+1} could be denoted as $(v_{\mathrm{tp}_{j-1}}, t_{\mathrm{tp}_{j-1}})$, $(v_{\mathrm{tp}_j}, t_{\mathrm{tp}_j})$ and $(v_{\mathrm{tp}_{j+1}}, t_{\mathrm{tp}_{j+1}})$, respectively. Subsequently, the fitting error of v_k could be obtained through vertical distance calculating from v_k to TPLine, named FE_k in Eq. (6). Finally, the importance of TP_j, named ITP_j, could be measured by the accumulated fitting errors in the range of $(t_{\mathrm{tp}_{j-1}}, t_{\mathrm{tp}_{j+1}})$, as shown in Eq. (7).

In particular, the importance evaluation instance of v_{115} could be divided into the following three steps in Fig. 2b. (1) The left adjacent v_{63} and right adjacent v_{154} would form a straight line as a green dotted line in Fig. 2b. (2) The fitting error of each raw data point v_k, FE_k, in the range of (63, 154), would be calculated by (6). And (3) the importance of v_{115}, can be calculated as $\mathrm{ITP}_{115} = \Sigma_{k=63}^{154}(\mathrm{FE}_k)$ by (7).

$$\mathrm{FE}_k = \left| v_{\mathrm{tp}_{j-1}} + \frac{(k - t_{\mathrm{tp}_{j-1}}) * (v_{\mathrm{tp}_{j+1}} - v_{\mathrm{tp}_{j-1}})}{(t_{\mathrm{tp}_{j+1}} - t_{\mathrm{tp}_{j-1}})} - v_k \right| \tag{6}$$
$$t_{\mathrm{tp}_{j-1}} < k < t_{\mathrm{tp}_{j+1}}$$

$$\mathrm{ITP}_j = \Sigma_{k=t_{\mathrm{tp}_{j-1}}}^{t_{\mathrm{tp}_{j+1}}}(\mathrm{FE}_k). \tag{7}$$

According to the above instance, the importance of all the TPs in T_i could be evaluated completely. There is still one thing should be noted that the beginning point of T_i would be used to evaluate the first TP, similarly the ending point

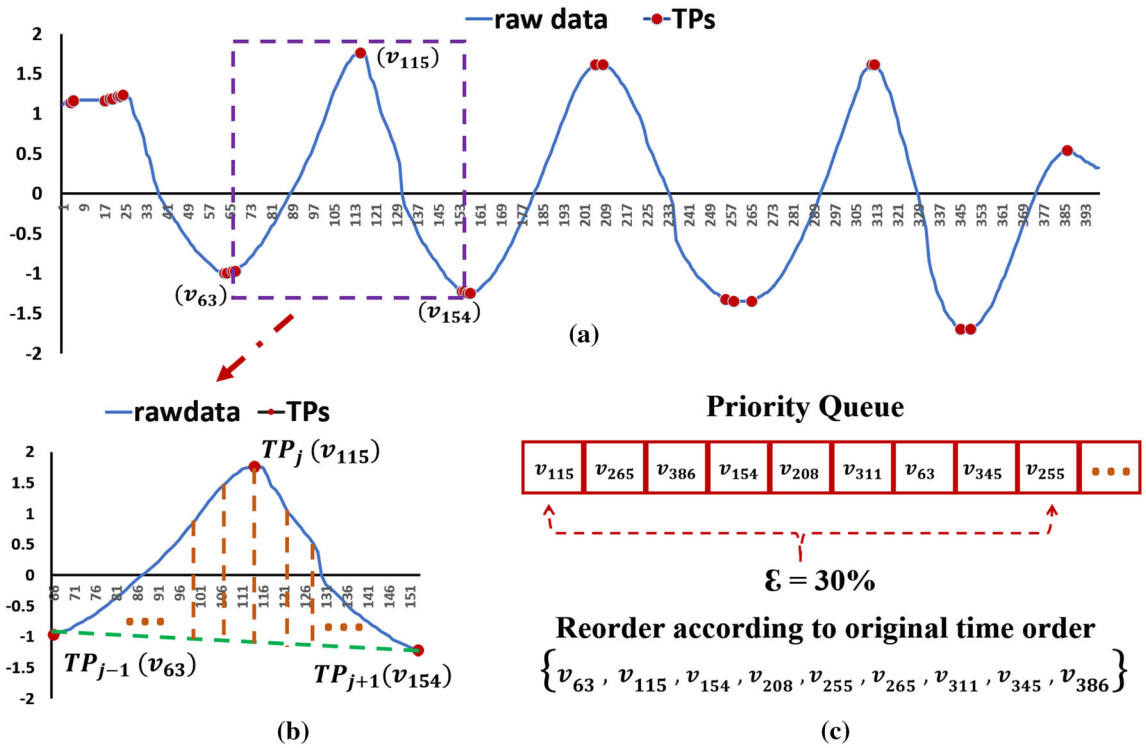


Fig. 2 TP identification on TS_i

of T_i would be used to evaluate the last TP in Fig. 2a. The TPLine in the first TP v_2 evaluation is connected from v_1 (the beginning point of T_i) to v_4 (right adjacent TP of v_2) and the TPLine in the last TP v_{386} evaluation is connected from v_{349} (left adjacent TP of v_{386}) to v_{398} (the ending point of T_i).

After the importance of all TPs in a certain time series has been evaluated completely, TPs could be stored in a priority queue according to the descending order of their own importance, as shown in Fig. 2c. Subsequently, a certain number of TPs, based on the specific data compression ratio, would be selected for shapelets generation and the corresponding definition is listed as follows.

Definition 3 After Num TPs in T_i have already been sorted in the descending order of their importance, the current data compression ratio (DCR), denoted as ε can be calculated in (8).

$$\text{CurNum} = \lfloor \text{Num} * \varepsilon \rfloor \quad 0 \leq \varepsilon \leq 100\%, \tag{8}$$

where CurNum refers to the current number of TPs.

According to Definition 3, supposing the current DCR (ε) is set as 30%, the 9 ($\lfloor 31 * 30\% \rfloor$) TPs in T_{16} have been selected and rearranged based on their original temporal order in Fig. 2c.

In what follows, two temporal feature-based representation learning models for TSC are introduced one by one.

3.2 Efficient shapelet transformation for TSC

The main processing steps of ST can be summarized as: (1) shapelet quality measurement setting, (2) shapelet generation and selection and (3) data transformation, respectively. In this paper, we proposed an efficient ST method (EST) to only replace the second step of original ST and remain the other two steps for efficient TSC.

Having obtained all the TPs in a certain time series T_i, we use them to produce the initial shapelet candidates. The subsequence could be selected as shapelet candidate with the following two requirements: (1) the begin point should be TP or the beginning point of T_i. (2) the end point should be TP or the ending point of T_i. According to these requirements, two subsequences in T_i should be selected as shapelet candidates, which describe the obvious temporal trends of T_i.

Specially, considering that the total number of data points for producing shapelets is 11 (9 TPs and 2 endpoints) in Fig. 2, the number of shapelet candidates is no more than 110 (11*10) by ESS, which is much less than that of the original T_i 158,006 (398*397) with 398 data points. Analogously, the number of corresponding shapelets generated from the entire D can be reduced dramatically. Subsequently, we can use information gain [20] to select a certain number (L) of candidates as final shapelets for further data transformation and classification. The

subsequent comparison experiments between our EST and other competitors are introduced in Sect. 4.

In light of this, assuming that the number of T_i in D is n, the length of T_i remains m, the average selected number of TPs in T_i is p, the time complexity of final shapelet selection could be analyzed as follows: (1) considering the number of TPs is no more than the length of time series m, the time complexity of TPs identification is $O(m)$ and the time complexity of TPs evaluation on T_i and the entire D is no worse than $O(p * m)$ and $O(n * p * m)$, respectively. (2) The time complexity of shapelet candidates generation in each T_i is $O(p^2)$, and the overall time complexity of D is no worse than $O(n * p^2)$. And (3) measuring the quality of one candidate requires the corresponding comparisons with n time series in D, whose time complexity is $O(n * p^2 * n * m * m)$. Hence, the overall time complexity of final shapelets selection is $O(n * p * m + n * p^2 * n * m * m)$, i.e., $O(n^2 * p^2 * m^2)$, which is less than the original time complexity $O(n^2 * m^4)$ of ST.

3.3 Deep representation learning for TSC

According to the above analysis, shapelets do reflect the main temporal features of a certain time series. Moreover, TP identification and evaluation strategies can be utilized adaptively for producing the corresponding shapelets on the whole datasets. Consequently, in this paper, we develop a TORRENT, for TSC. The framework of our proposed model is illustrated in Fig. 3.

Our proposed TORRENT model comprises the following three components: (1) temporal trend extraction, which can simultaneously acquire the basic and important trends in the given time series T_i; (2) temporal feature encoding, which can encode the discriminative temporal trends into corresponding features; (3) comprehensive representation learning for predicting the appropriate class label for T_i.

3.3.1 Temporal trend extraction

Given a certain time series T_i with l TPs, we first conduct TP identification and evaluation through Definition. 2 to sort l TPs according to the descending order of their own importance. And then we select CurNum TPs $(l * \varepsilon + 2)$ including 2 endpoints (v_1^j, v_m^j) in T_i based on a predefined DCR ε. As shown in Fig. 3, there are 6 TPs to identify the corresponding important temporal trends in T_i. Although TPs do reflect the main temporal information of T_i based on the corresponding time series domain knowledge, considering the different types of time series, it is relatively one-sided to rely on human experience-based shapelets for TSC, in other words, the widely varying latent features in time series are also important criteria for TSC. Consequently, we utilize sliding window with length λ and overlap rate α [33], to subdivide the entire T_i into $CurNum + 1$ sequences, expressed as $S_i = \{s_1^i, s_2^i, \ldots, s_j^i, \ldots, s_M^i\}$, where s_j^i denotes the jth segment of T_i.

3.3.2 Temporal feature encoding

To model time series T_i, we proposed a novel bidirectional enhanced LSTM (Bi-ELSTM) to encode the temporal features of S_i. Although traditional Bi-LSTM is able to memorize the main temporal information of S_i for T_i encoding, it is insufficient to capture crucial temporal trends in long time series. To tackle the aforementioned problem, it is necessary to heighten the memorization of crucial temporal trends with contextual information. Specially, if s_j^i of S_i contains TP_k, the segments $\widetilde{\mathbf{s}}_j^*$, involving TP_{k-1} and TP_{k+1}, in S_i are considered as the corresponding contextual information of s_j^i, Bidirectional Enhanced LSTM (Bi-ELSTM) combines s_j^i and $\widetilde{\mathbf{s}}_j^*$ together for feature encoding.

In Fig. 3, S_i, expressed as the set of column vectors, is to input into Bi-ELSTM according to its own temporal order. Moreover, the segments s_{j-1}^i and s_{j+1}^i of s_j^i, containing TP_{k-1} and TP_{k+1}, respectively, are also considered as the corresponding pre-context and post-context features to input into Bi-ELSTM for feature encoding. Specially, if the pre-context and post-context information of the specific s_j^i involve multiple segments, we adopted mean pooling-based dimension reduction strategy [28, 35] to obtain the comprehensive context representation of s_j^i. The detail of ELSTM unit is also shown in Fig. 3.

Formally, we formulate the forward ELSTM as follows,

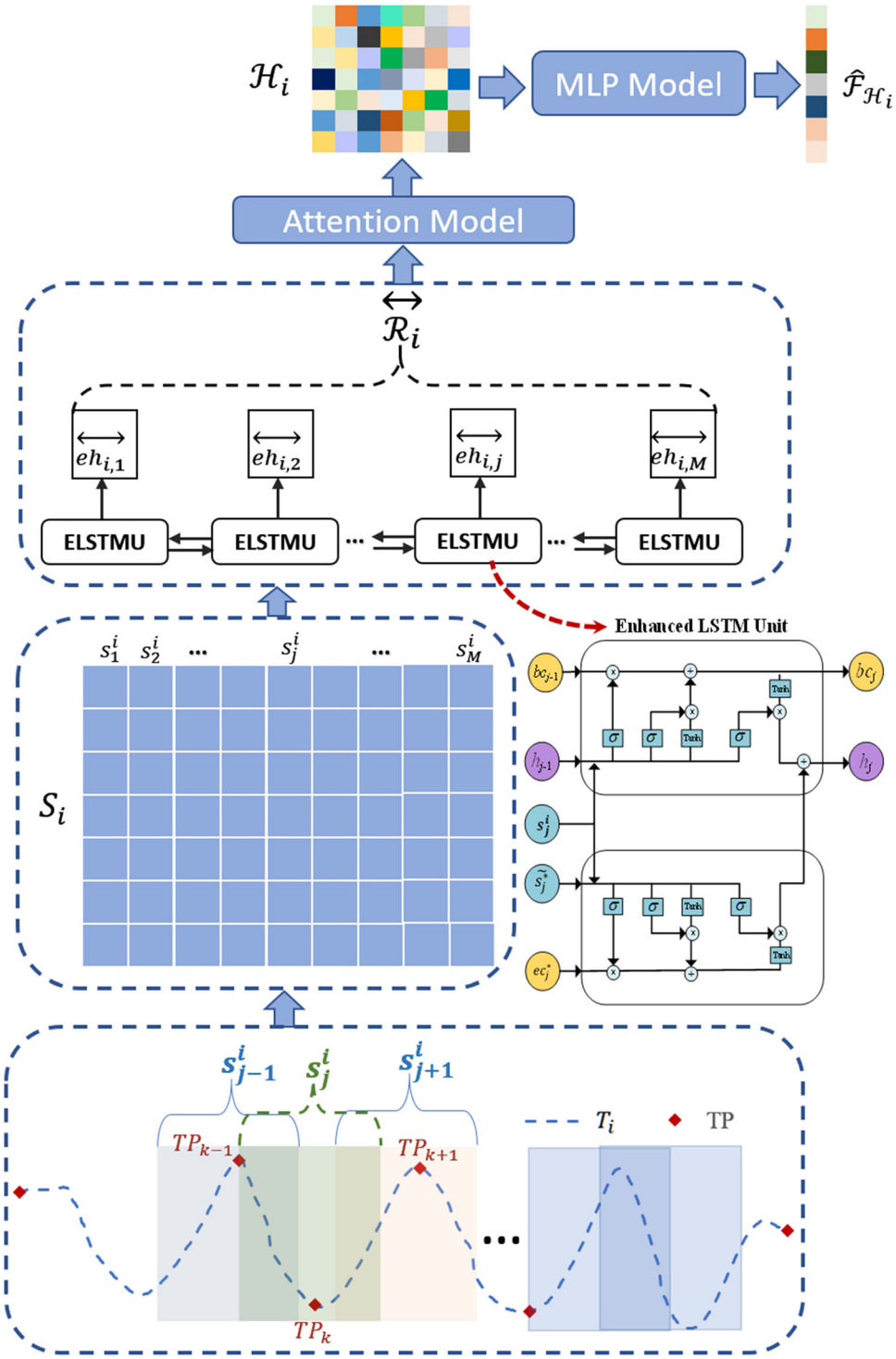


Fig. 3 Schematic illustration of our proposed TORRENT

$$
\begin{cases}
\overrightarrow{\mathbf{fi}}_j = \sigma(\mathbf{W}_{is}\mathbf{s}_j^i + \mathbf{W}_{ih}\overrightarrow{\mathbf{h}}_{j-1} + \mathbf{b}_i), \\
\overrightarrow{\mathbf{ff}}_j = \sigma(\mathbf{W}_{fs}\mathbf{s}_j^i + \mathbf{W}_{fh}\overrightarrow{\mathbf{h}}_{j-1} + \mathbf{b}_f), \\
\overrightarrow{\mathbf{fo}}_j = \sigma(\mathbf{W}_{os}\mathbf{s}_j^i + \mathbf{W}_{oh}\overrightarrow{\mathbf{h}}_{j-1} + \mathbf{b}_{out}), \\
\overrightarrow{\mathbf{fu}}_j = tanh(\mathbf{W}_{us}\mathbf{s}_j^i + \mathbf{W}_{uh}\overrightarrow{\mathbf{h}}_{j-1} + \mathbf{b}_u), \\
\overrightarrow{\mathbf{bc}}_j = \overrightarrow{\mathbf{fi}}_j \odot \overrightarrow{\mathbf{fu}}_j + \overrightarrow{\mathbf{ff}}_j \odot \overrightarrow{\mathbf{bc}}_{j-1}, \\
\overrightarrow{\mathbf{ei}}_j^* = \sigma(\mathbf{W}_{is}^*\widetilde{\mathbf{s}}_j^* + \mathbf{W}_{ih}^*\overrightarrow{\mathbf{h}}^* + \mathbf{b}_i^*), \\
\overrightarrow{\mathbf{ef}}_j^* = \sigma(\mathbf{W}_{fx}^*\widetilde{\mathbf{s}}_j^* + \mathbf{W}_{fh}^*\overrightarrow{\mathbf{h}}^* + \mathbf{b}_f^*), \\
\overrightarrow{\mathbf{eo}}_j^* = \sigma(\mathbf{W}_{os}^*\widetilde{\mathbf{s}}_j^* + \mathbf{W}_{oh}^*\overrightarrow{\mathbf{h}}^* + \mathbf{b}_{out}^*), \\
\overrightarrow{\mathbf{eu}}_j^* = tanh(\mathbf{W}_{us}^*\widetilde{\mathbf{s}}_j^* + \mathbf{W}_{uh}^*\overrightarrow{\mathbf{h}}^* + \mathbf{b}_u^*), \\
\overrightarrow{\mathbf{ec}}_j^* = \overrightarrow{\mathbf{ei}}_j^* \odot \overrightarrow{\mathbf{eu}}_j^* + \overrightarrow{\mathbf{ef}}_j^* \odot \overrightarrow{\mathbf{ec}}^*, \\
\overrightarrow{\mathbf{eh}}_j = \overrightarrow{\mathbf{bo}}_j \odot tanh(\overrightarrow{\mathbf{bc}}_j) + \overrightarrow{\mathbf{eo}}_j^* \odot tanh(\overrightarrow{\mathbf{ec}}_j^*),
\end{cases} \tag{9}
$$

where s_j^i is the temporal segment produced by sliding window at the time step j, $\overrightarrow{fi}_j$, $\overrightarrow{ff}_j$, $\overrightarrow{fo}_j$, $\overrightarrow{h}_{j-1}$ denote the jth forward feature embedding of input gate, forget gate, output gate and memory cell state, respectively. $\overrightarrow{bc}_{j-1}$ refers to jth the basic hidden state feature. σ denotes the logistic sigmoid function; and $\odot$ denotes element wise multiplication. Moreover, $\overrightarrow{eh}^*$ and $\overrightarrow{ec}^*$ are the enhanced memory cell state and hidden state at the time step j, combining with the current context information. Subsequently, we obtain forward representations $\overrightarrow{\mathcal{R}}_i = [\overrightarrow{eh}_{i,1}, \overrightarrow{eh}_{i,2}, \ldots, \overrightarrow{eh}_{i,j}, \ldots, \overrightarrow{eh}_{i,M}]$. Similarly, we employ backward ELSTM on S_i for the corresponding feature encoding, $\overleftarrow{\mathcal{R}}_i = [\overleftarrow{eh}_{i,1}, \overleftarrow{eh}_{i,2}, \ldots, \overleftarrow{eh}_{i,j}, \ldots, \overleftarrow{eh}_{i,M}]$.

Subsequently, we concatenate the forward $\overrightarrow{\mathcal{R}}_i$ and backward $\overleftarrow{\mathcal{R}}_i$ together to obtain the bidirectional representation, as $\overleftrightarrow{\mathcal{R}}_i = [\overleftrightarrow{eh}_{i,1}, \ldots, \overleftrightarrow{eh}_{i,j}, \ldots, \overleftrightarrow{eh}_{i,M}] \in \mathbb{R}^{d_h \times M}$, where d_h is the dimension of each hidden state in R_i. Obviously, our ELSTM layer can simultaneously leverage crucial context information to enhance the memorization of temporal trends and further strengthen the feature encoding.

3.3.3 Temporal representation learning

We utilize attention strategy to further obtain the improved representation of $\overleftrightarrow{\mathcal{R}}_i$ based on different attentive scores. Specifically, the last hidden state $\overleftrightarrow{eh}_{i,M}$ is selected to measure with each encoder hidden state $\overleftrightarrow{eh}_{i,j} \in \overleftrightarrow{\mathcal{R}}_i$ for attentive score calculation, expressed as

$$
r_j^i = \tanh(W_{rh}\overleftrightarrow{eh}_{i,j} + U_{rh}\overleftrightarrow{eh}_{i,M}) \tag{10}
$$

where W_{rh} and U_{rh} are parameters. Subsequently, all the attentive scores r_j^is are normalized to further form attention-based representation for $\overleftrightarrow{\mathcal{R}}_i$ by weighted sum in Eq. 11:

$$
\begin{aligned}
\gamma_{i,j} &= \frac{\exp(r_j^i)}{\sum_{j=1}^{M}\exp(r_j^i)}, \\
\widetilde{h_{i,j}} &= \sum_{t=1}^{M}\gamma_{i,j}\overleftrightarrow{eh}_{i,j},
\end{aligned} \tag{11}
$$

where $\widetilde{h_{i,j}}$ is the attention-based representation of jth encoder hidden state. Consequently, the attention-based representation on $\overleftrightarrow{\mathcal{R}}_i$ can be expressed as $\mathcal{H}_i = [\widetilde{h_{i,1}}, \ldots, \widetilde{h_{i,j}}, \ldots, \widetilde{h_{i,M}}]$.

Thereafter, we feed $\mathcal{H}_i$ into a multi-layer perception (MLP) network with Z layers to complete the classification on T_i, as follows,

$$
\begin{cases}
\mathcal{F}_{\mathcal{H}_i}^1 = \sigma_r^1(\mathbf{W}_r^1\mathcal{H}_i + \mathbf{g}_r^1), \\
\vdots \\
\mathcal{F}_{\mathcal{H}_i}^z = \sigma_r^z(\mathbf{W}_r^z\mathcal{F}_{\mathcal{H}_i}^{z-1} + \mathbf{g}_r^z), \\
\vdots \\
\mathcal{F}_{\mathcal{H}_i}^Z = \sigma_r^Z(\mathbf{W}_r^Z\mathcal{F}_{\mathcal{H}_i}^{Z-1} + \mathbf{g}_r^Z),
\end{cases} \tag{12}
$$

where $\mathbf{W}_r^z$, $\mathbf{g}_r^z$ and $\mathcal{F}_{\mathcal{R}_i}^z$, respectively, denote the weight matrix, bias vector and output of the zth hidden layers, σ_r^z is the Randomized Leaky Rectified Linear Units function [16], and the output of the Zth layer of MLP $\mathcal{F}_{\mathcal{H}_i}^Z$ is the final representation of T_i, expressed as $\widehat{\mathcal{F}}_{\mathcal{H}_i}$ with K elements. Subsequently, we use softmax to normalize $\widehat{\mathcal{F}}_{\mathcal{H}_i}$ into corresponding class probability distribution $P(K|T_i)$.

$$
P(K|T_i) = \text{softmax}(\widehat{\mathcal{F}}_{\mathcal{H}_i}), \tag{13}
$$

where K refers to the number of class label, and $P(K|T_i)$ represents the probability distribution of classifying the ith time series (T_i) into K class labels.

Considering our goal aims to maximize the classification prediction probability on D, we optimized the negative log-likelihood loss function as follows:

$$
K(\theta) = -\frac{1}{N}\sum_{i=1}^{N}\log L_i P(K|T_i), \tag{14}
$$

where L_i is the 0-1 vector of K class labels on T_i; N denotes the total number of time series in dataset D; θ is the parameter of TORRENT.

4 Experiments and evaluation

We have conducted a set of experiments to evaluate the classification performance of our EST and TORRENT compared to other baseline competitors. We begin with the experimental settings and then analyze the corresponding experimental results.

4.1 Experimental settings

We conduct extensive comparison experiments on 10 typical datasets [2] and 3 our collected network traffic flow time series datasets of Shandong University, dubbed as Inflow, Outflow and Totalflow. Moreover, in the following comparison experiments, on the one hand, as for EST, the number of shapelets (L) is set at the half number of time series (N) in D, i.e., $L = N/2$, which is exactly the same as original ST. The ε in our experiments is set as 30% initially and the corresponding analysis on the varying ε is also given later. On the other hand, as for TORRENT, the hidden state size and drop ratio [8, 17] are set to 200 and 0.8, respectively. The learning rate is set to 0.001. Moreover, we empirically set the maximum number of epochs as 1000 to ensure the convergence. Besides, all the deep learning comparison experiments are conducted over a computer equipped with Ubuntu 16.04.6 LTS, Intel Xeon CPU E5-2620, 128 GB Memory and NVIDIA TITAN Xp GPU.

4.2 Comparison on shallow representation learning for TSC

In order to fully evaluate the performance of our methods, 4 evolutionary time series classification algorithms based on shapelets mentioned above: ST [18], COTE [1], LS [11], FS [26] have been chosen as the baseline methods. Moreover, there are 4 shapelet-based TSC acceleration algorithms: SD [12], RS [27], gRSF [13], SALSA-R [9] have also been selected as the baseline methods in our experiments.

4.2.1 Comparison on classification accuracy

In the following shallow learning comparison experiments, the accuracy results of gRSF are obtained by utilizing the default parameters as well as the open source tools shared by Isak et al. [13], that of SALSA-R are obtained by sampling 30% subsequences as candidates for classification. The experimental accuracy results are shown in Table 2.

Obviously, the classification accuracy of EST is better than other 4 acceleration methods. As for the comparison with 3 primal TSC methods, due to the shapelet selection only rely on TP evaluation, some latent features of the given time series may not be captured for TSC. The accuracy of EST is lower than ST and significantly higher than LS and FS. Moreover, considering that the average accuracy difference between EST and ST is less than 0.05, the accuracy of ST and EST is basically in the same level.

4.2.2 Comparison experiments on shapelets generation efficiency

According to the above analysis, the main difference between EST and ST exists in shapelet generation and selection; therefore, the specific processing efficiency of them is further analyzed. Besides, according to the above accuracy rankings, 2 methods with relatively high accuracy (LS, gRSF) are selected to further analyze the efficiency of shapelets selection. Considering FS is the fastest in three primal methods mentioned in Table 1, the running time for shapelets selection in FS is also concerned in our comparison experiments. Moreover, due to the fact that the running time for the completed classification by the primal methods is too long, e.g., it takes more than 7 days to complete the entire classification on the above datasets by ST, LS, etc. Accordingly, in this subsection, the comparison experiments on running time only focus on the elapsed time for shapelets generation and selection. last but not least, due to the selection process in gRSF is addressed in parallel, the CPU Time would be considered as the running time in gRSF to complete shapelets selection. The results of running time on shapelets selection are shown in Table 3.

According to Table 3, it is no difficult to find that EST is not only significantly faster than the acceleration method gRSF by nearly 30-fold, but also faster than FS and ST by one and three orders of magnitude, respectively.

In consequence, based on the above analyses on accuracy and running time, we can conclude that the EST can greatly improve the efficiency of the shapelet transform-based TSC methods, while retaining relatively high classification accuracy.

4.2.3 Effect of ε on EST

To further analysis the different DCR effect on the running time and accuracy of EST, ε would be varied from 10 to 50% on the two datasets (Inflow, Outflow), the corresponding results are shown in Fig. 4.

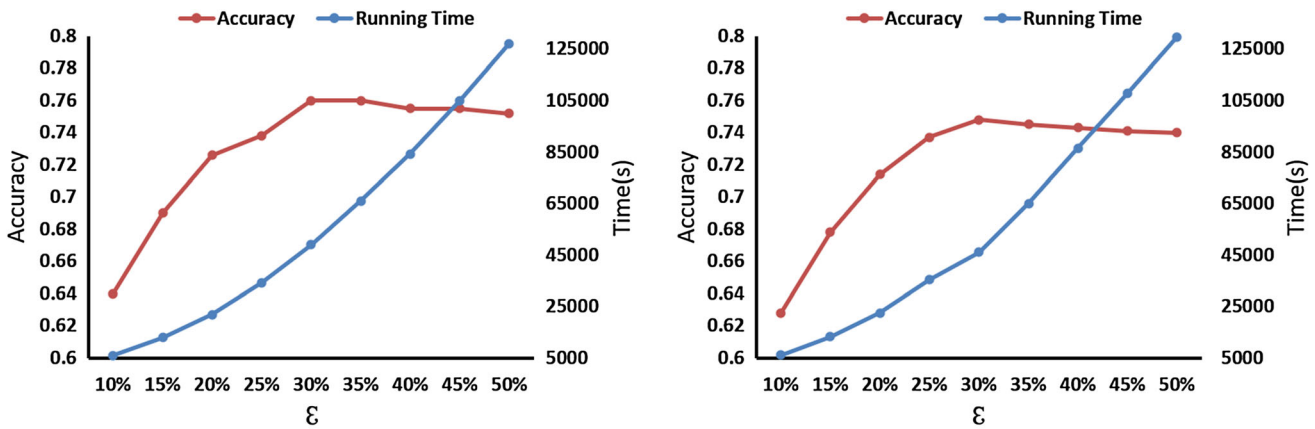
As ε increases from 10 to 50%, we can find that the running time of EST continues to rise in Fig. 4. The reason is that the increase in ε means more TPs are chose for shapelets selection, which undoubtedly increase the corresponding time overhead. As for accuracy, It can also be

Table 2 Shallow representation learning comparison on classification accuracy

Dataset	Primal methods			Acceleration methods				
	ST	LS	FS	gRSF	SALSA-R	SD	RS	EST
ChlorineConcentration	0.682	0.586	0.566	0.658	0.671	0.553	0.572	0.717
Coffee	0.995	0.998	0.917	0.964	0.960	0.961	0.769	1.000
DiatomSizeReduction	0.911	0.927	0.873	0.779	0.769	0.896	0.774	0.869
Inflow	0.778	0.727	0.551	0.736	0.722	0.581	0.521	0.767
ItalyPowerDemand	0.953	0.960	0.917	0.944	0.951	0.920	0.924	0.955
Light7	0.724	0.765	0.644	0.726	0.695	0.652	0.635	0.678
MedImgs	0.691	0.704	0.609	0.697	0.686	0.676	0.529	0.699
MoteStrain	0.882	0.876	0.793	0.952	0.854	0.783	0.815	0.904
Outflow	0.766	0.713	0.564	0.743	0.731	0.578	0.530	0.748
Symbols	0.862	0.919	0.908	0.755	0.864	0.865	0.795	0.941
Totalflow	0.779	0.738	0.560	0.739	0.728	0.586	0.527	0.759
Trace	1.000	0.996	0.998	1.000	1.000	0.965	0.934	0.980
TwoLeadECG	0.984	0.994	0.920	0.991	0.958	0.867	0.914	0.982
Ranking	1	3	7	4	5	6	8	2

Table 3 Running time on shapelet selection (*s*)

Dataset	ST	LS	FS	gRSF	EST
ChlorineConcentration	21,758.6	6172.5	170.7	618.6	2.9
Coffee	530.8	230.6	6.2	5.8	0.1
DiatomSizeReduction	310.9	769.1	7.0	11.5	0.2
Light7	13,569.8	14,465.9	102.9	113.4	9.2
MoteStrain	2.1	20.4	0.3	1.7	0.01
Symbols	4386.1	4623.7	35.7	38.2	4.1
Trace	6832.7	5251.31	48.2	77.8	15.8
Average time	6770.14	4504.79	53.00	123.86	4.62



(a) Effect of ε on Inflow. **(b)** Effect of ε on Outflow.

Fig. 4 Parameter analyses on ε of EST

found in Fig. 4 that the corresponding accuracies on both of the datasets have all experienced continuous rise, remained stable and slight decline, attributed to the following reasons. In the beginning, due to the relatively small ε, few TPs (relatively few data features) are used for shapelets selection and subsequent classification, so the accuracy of EST is also comparatively low. Along with the continues increase in ε, the increasing numbers of TPs (more temporal features) continuously improve the classification accuracy in a relatively high level. Subsequently,

as ε increases further, more but less important TPs are chosen to bury the main temporal feature into plenty of the trivial details and incur the corresponding decline of accuracy. Finally, according to the verification on the above experimental results, ε is set within the range of 30–40% is relatively reasonable. To better balance efficiency and accuracy, ε is set to 30% in the comparison experiments.

4.3 Comparison on deep representation learning for TSC

Analogously, we selected 4 shallow learning methods: ST in Jon et al. [18], COTE in Anthony et al. [1], LS in Grabocka et al. [11] and FS in Thanawin and Eamonn [26] as well as 3 deep learning methods: MLP [34], GCRNN [25] and LSTM-FCN [10] as the baselines in our comparison experiments. The corresponding results are shown in Table 4.

4.3.1 Comparison on classification accuracy

As shown in Table 4, the classification accuracy of COTE is higher than other 3 shallow learning methods (ST, LS and FS) but slightly lower than MLP. Moreover, we can find the classification accuracy of GCRNN basically better than COTE, which verifies that the fully convolutional network based on deep representation learning is indeed effective in improving the accuracy of classification. However, regarding GCRNN neglects the intrinsic temporal features of time series, its accuracy is lower than that of LSTM-FCN. Finally, compared to other baselines, TORRENT can not only learn the local temporal features in a certain time series, but also leverage crucial context information to enhance the memorization of temporal trends, while improving the accuracy of TSC.

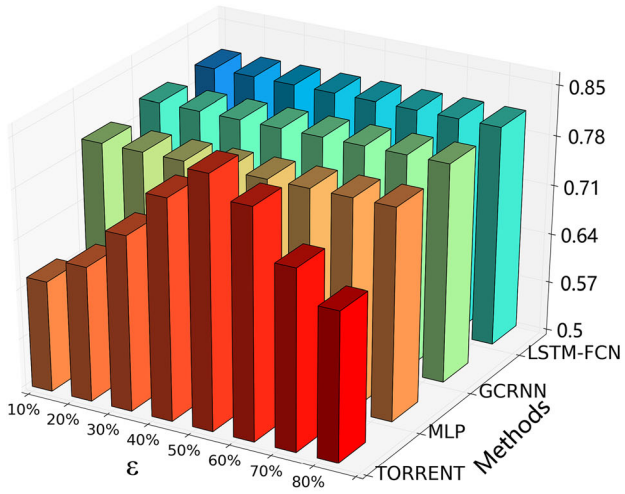
4.3.2 Effect of ε and λ for TORRENT

Subsequently, to further analysis the different ε and λ effects on the corresponding accuracy of TORRENT, we first set λ to 20% and vary ε from 10 to 50% on the two datasets (Inflow, Outflow), and then, we set ε to 30% and vary λ from 10 to 50% on the same datasets. Although the other three deep representation learning methods are not affected by parameters ε and λ, GCRNN adopts the CNN based feature embedding in accordance with the original temporal order [25], LSTM-FCN comprehensively processes the entire time series in 1 time step [10]. We still added them together as constant references to clearly display the changing trends of TORRENT.

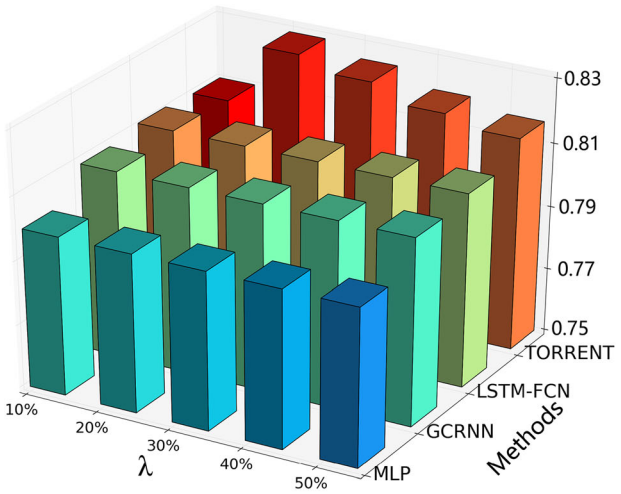
In Fig. 5, with the increase in ε (from 10 to 80%), in the beginning, TORRENT cannot capture main temporal features for representation learning and subsequent classifying based on relatively few TPs, so the accuracy of TORRENT is also comparatively low. Along with the continues increase in ε, the increasing numbers of TPs (more data features) sustained improve the classification accuracy in a relatively high level. Subsequently, as ε increases further, more but less important TPs would be chosen to add some noise information and several trivial temporal fluctuations, which undoubtedly incur the corresponding decline of accuracy. Analogously, with the increase in λ (from 20 to 60%), the corresponding accuracy results of TORRENT on these two datasets have all experienced corresponding rising, maintenance and declining. Compared to the above

Table 4 Deep representation learning comparison on classification accuracy

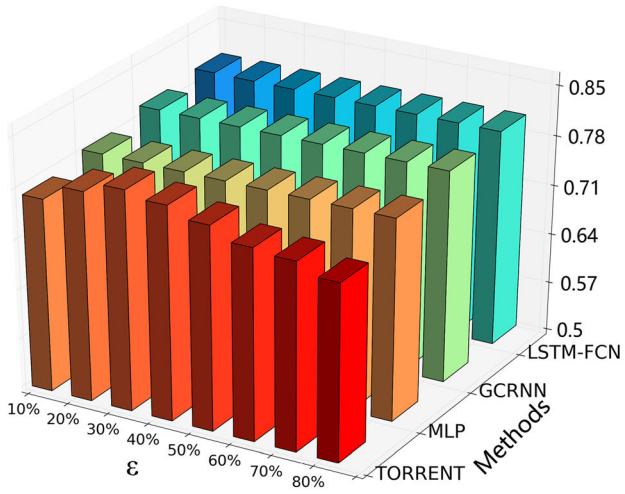
Dataset	Shallow learning methods				Deep learning methods			
	COTE	ST	LS	FS	MLP	GCRNN	LSTM-FCN	TORRENT
ChlorineConcentration	0.736	0.682	0.586	0.566	0.867	0.859	0.839	0.875
Coffee	1.000	0.995	0.995	0.917	1.000	1.000	1.000	0.999
DiatomSizeReduction	0.929	0.911	0.927	0.873	0.967	0.965	0.967	0.969
Inflow	0.801	0.778	0.727	0.551	0.800	0.809	0.811	0.829
ItalyPowerDemand	0.970	0.953	0.960	0.917	0.969	0.966	0.963	0.971
Light7	0.799	0.724	0.765	0.644	0.863	0.865	0.835	0.868
MedImgs	0.785	0.691	0.704	0.609	0.792	0.803	0.801	0.799
MoteStrain	0.902	0.882	0.876	0.793	0.952	0.949	0.939	0.957
Outflow	0.793	0.766	0.713	0.564	0.785	0.799	0.805	0.810
Symbols	0.953	0.862	0.919	0.908	0.967	0.955	0.984	0.981
Totalflow	0.787	0.779	0.738	0.560	0.792	0.803	0.811	0.808
Trace	1.000	1.000	0.996	0.998	1.000	1.000	0.989	1.000
TwoLeadECG	0.983	0.984	0.994	0.920	1.000	0.983	0.999	0.999
Ranking	5	6	7	8	4	3	2	1



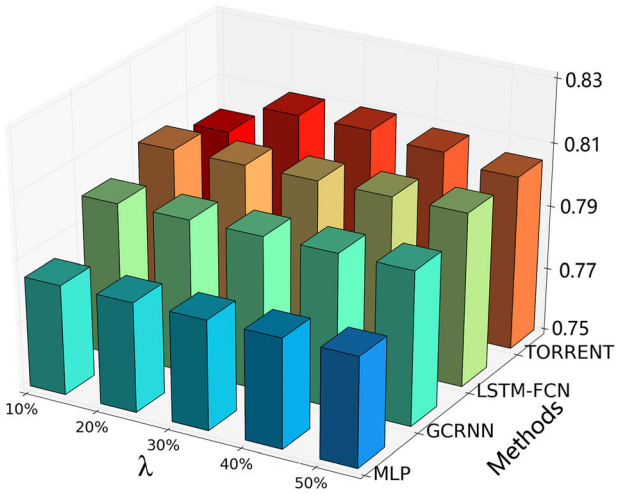
(a) Accuracy w.r.t ε on Inflow.



(b) Accuracy w.r.t λ on Inflow.



(c) Accuracy w.r.t ε on Outflow.



(d) Accuracy w.r.t λ on Outflow.

Fig. 5 Parameter analyses on ε and λ of TORRENT

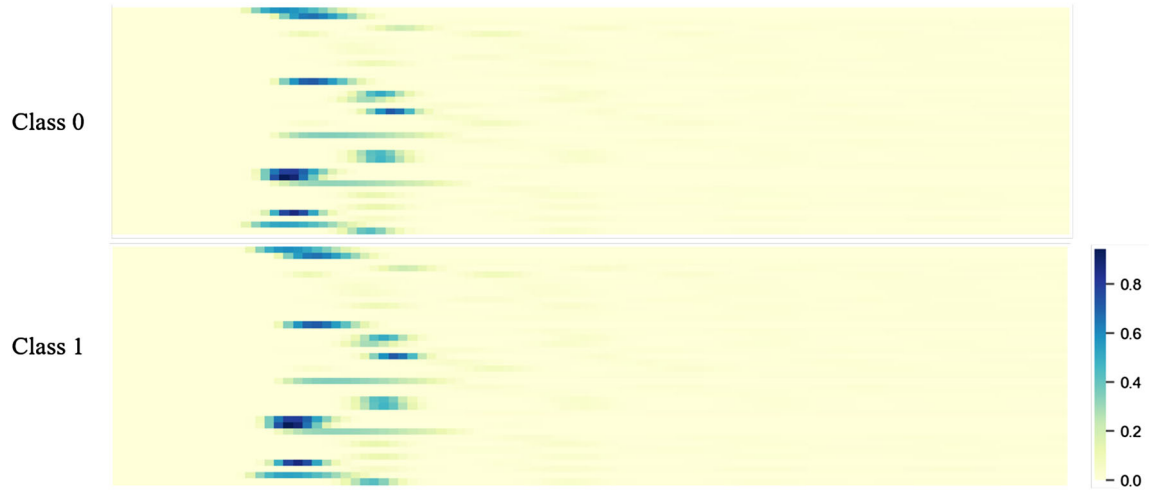
results on λ, the changes on ε are significantly dramatic, which reveals that important temporal trends have more pronounced effects on classification accuracy.

4.3.3 Attention visualization

In this subsection, we visualize attention mechanism of TORRENT for TSC. Accordingly, we randomly select the attention scores-based classification results on Inflow and Totalflow dataset. The color depth reflects to the corresponding attentive weights in Eq. 10, concretely, deeper color indicates higher attentive score and vice versa.

As shown in Fig. 6, two interesting observations can be found. For one thing, in the same dataset, the attention scores on time series belonging to the identical classification set are relatively close, while the attention scores on time series from different classification sets differ greatly. This demonstrates that our proposed TORRENT can effectively capture and learn the corresponding hidden temporal features from different time series for TSC. For another, in the same class set, segments of the given time series with higher attention weights (darker colors) are also relatively concentrated, which not only verifies TORRENT can automatically learn important temporal features in the given time series, but also shows the rationality of traditional shapelet-based TSC methods, i.e., shapelet instead of the entire time series can reflect the main feature of class membership to some extent.

(a) Attention visualization on Inflow.



(b) Attention visualization on Totalflow.

Fig. 6 Attention visualization of TORRENT for TSC

5 Conclusion

In this paper, we propose a novel shapelet transformation model EST, which utilizes turning points-based representation learning, to promote the efficiency of TSC. Moreover, we develop a deep representation learning network TORRENT, which can leverage crucial context information to strengthen the capability of representation learning, while boosting the accuracy of TSC. The extensive experimental results demonstrate the necessity and correctness of our proposed models. Based upon this study, our future work will be carried out along 3 promising directions: (1) we plan to integrate EST into incremental learning strategy for streaming time series classification, (2) we intend to utilize the latent feature encoding of TORRENT for time series anomaly detection [23], and (3) we desire to adopt TORRENT as a useful sub-model for the audio representation learning for multimedia data mining.

Acknowledgements The authors would like to thank the anonymous reviewers and the editors for their insightful comments and suggestions, which are greatly helpful for improving the quality of this paper. This work is supported by the National Natural Science Foundation of China, Nos.: 61772310, 61702300, 61702302, 61802231; the Key Research and Development Program of China, Nos.: 2017YFC0803400, 2018YFC0831000; the project of CERNET Innovation (NGII20190109); and the project of Qingdao Postdoctoral Applied Research.

Compliance with ethical standards

Conflict of interest All authors declare that they have no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

Informed consent Informed consent was obtained from all individual participants included in the study.

References

1. Anthony B, Jason L, Jon H, Aaron B (2015) Time-series classification with cote: the collective of transformation-based ensembles. IEEE Trans Knowl Data Eng 27:2522–2535
2. Anthony B, Jason L, William V, Eamonn K (2016) The uea & ucr time series classification repository. www.timeseriesclassification.com
3. Anthony B, Jason L, Aaron B, James L, Eamonn K (2017) The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. Data Min Knowl Disc 31:606–660
4. Chaoran C, Jun M, Tao L, Zhumin C, Shuaiqiang W (2015) Improving image annotation via ranking-oriented neighbor search and learning-based keyword propagation. J Assoc Inf Sci Technol 66:82–98
5. Chaoran C, Jialie S, Liqiang N, Richang H, Jun M (2017) Augmented collaborative filtering for sparseness reduction in personalized poi recommendation. ACM Trans Intell Syst Technol 8:1–23
6. Chaoran C, Huihui L, Tao L, Liqiang N, Lei Z, Yilong Y (2019) Distribution-oriented aesthetics assessment with semantic-aware hybrid network. IEEE Trans Multim 21:1209–1220
7. Cun J, Chao Z, Shijun L, Chenglei Y, Li P, Lei W, Xiangxu M (2019) A fast shapelet selection algorithm for time series classification. Comput Netw 148:231–240
8. Dan S, Lei Z, Yikun L, Jingjing L, Xiushan N (2019) Robust structured graph clustering. IEEE Trans Neural Netw Learn Syst. https://doi.org/10.1109/TNNLS.2019.2955209
9. Daniel G, Danny H, Lior R (2015) Fast and space-efficient shapelets-based time-series classification. Intell Data Anal 19:953–981
10. Fazle K, Somshubra M, Houshang D (2019) Insights into lstm fully convolutional networks for time series classification. IEEE Access 7:67718–67725
11. Grabocka J, Schilling N, Wistuba M, Schmidt-Thieme L (2014) Learning time-series shapelets. In: International conference on Knowledge discovery and data mining, pp 392–401
12. Grabocka J, Wistuba M, Schmidt-Thieme L (2016) Fast classification of univariate and multivariate time series through shapelet discovery. Knowl Inf Syst 49:429–454
13. Isak K, Panagiotis P, Henrik B (2016) Generalized random shapelet forests. Data Min Knowl Disc 30:1053–1085
14. Jason L, Luke MD, Jon H, Anthony B (2012) A shapelet transform for time series classification. In: International conference on Knowledge discovery and data mining, pp 289–297
15. Jason L, Sarah T, Anthony B (2016) Hive-cote: the hierarchical vote collective of transformation-based ensembles for time series classification. In: International conference on data mining
16. Jingjing L, Ke L, Zi H, Lei Z, Hengtao S (2019) Heterogeneous domain adaptation through progressive alignment. IEEE Trans Neural Netw 30:1381–1391
17. Jingjing L, Ke L, Zi H, Lei Z, Hengtao S (2019) Transfer independently together: a generalized framework for domain adaptation. IEEE Trans Cybern 49:2144–2155
18. Jon H, Jason L, Edgaras B, James M, Anthony B (2014) Classification of time series by shapelet transformation. Data Min Knowl Disc 28:851–881
19. Lexiang Y, Eamonn K (2009) Time series shapelets: a new primitive for data mining. In: International conference on Knowledge discovery and data mining, pp 947–956
20. Lexiang Y, Eamonn K (2011) Time series shapelets: a novel technique that allows accurate, interpretable and fast classification. Data Min Knowl Disc 22:149–182
21. Mit S, Josif G, Nicolas S, Martin W, Lars S (2016) Learning DTW-shapelets for time-series classification. In: International conference on data science
22. Mueen A, Keogh E, Young N (2011) Logical-shapelets: an expressive primitive for time series classification. In: International conference on Knowledge discovery and data mining, pp 1154–1162
23. Qi Z, Yupeng H, Cun J, Peng Z, Xueqing L (2018) Edge computing application: real-time anomaly detection algorithm for sensing data. J Comput Res Dev 55:524–536
24. Qing H, Zhi D, Fuzhen Z, Tianfeng S, Zhongzhi S (2012) Fast time series classification based on infrequent shapelets. In: International conference on machine learning and applications. IEEE, pp 215–219
25. Sangdi L, George CR (2018) GCRNN: group-constrained convolutional recurrent neural network. IEEE Trans Neural Netw 29:4709–4718
26. Thanawin R, Eamonn K (2013) Fast shapelets: a scalable algorithm for discovering time series shapelets. In: International conference on data mining, pp 668–676
27. Xavier R, Maria R, Walid E, Marcin D (2015) Random-shapelet: an algorithm for fast shapelet discovery. In: International conference on data science and advanced analytics, pp 1–10
28. Yudong H, Lei Z, Zhiyong C, Jingjing L, Xiaobai L (2020) Discrete optimal graph clustering. IEEE Trans Cybern 50:1697–1710
29. Yupeng H, Cun J, Ming J, Xueqing L (2016) A k-motifs discovery approach for large time-series data analysis. In: Asia-pacific web conference, pp 492–496
30. Yupeng H, Cun J, Ming J, Yiming D, Shuo K, Xueqing L (2016) A continuous segmentation algorithm for streaming time series. In: International conference on collaborative computing: networking, applications and worksharing, pp 140–151
31. Yupeng H, Cun J, Qingke Z, Lin C, Peng Z, Xueqing L (2019) A novel multi-resolution representation for time series sensor data analysis. Soft Comput: 1–26
32. Yupeng H, Peiyuan G, Peng Z, Yiming D, Xueqing L (2019b) A novel segmentation and representation approach for streaming time series. IEEE Access 7:184423–184437
33. Yupeng H, Pengjie R, Wei L, Peng Z, Xueqing L (2019c) Multi-resolution representation with recurrent neural networks application for streaming time series in iot. Comput Netw 152:114–132
34. Zhiguang W, Weizhong Y, Tim O (2017) Time series classification from scratch with deep neural networks: a strong baseline. In: International joint conference on neural networks, pp 1578–1585
35. Zhiyong C, Xiaojun C, Lei Z, Catherine Rose K, Mohan SK (2019) MMALFM: explainable recommendation by leveraging reviews and images. ACM Trans Inf Syst 37:16:1–16:28