Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Multivariate time series classification with crucial timestamps guidance

Da Zhang [a,b], Junyu Gao [a,b], Xuelong Li [c,b,a,*]

[a] School of Artificial Intelligence, OPtics and ElectroNics (iOPEN), Northwestern Polytechnical University, Xi'an 710072, China
[b] Key Laboratory of Intelligent Interaction and Applications, Ministry of Industry and Information Technology, Xi'an 710072, China
[c] Institute of Artificial Intelligence (TeleAI), China Telecom, China

ARTICLE INFO

Keywords:
Multivariate time series classification
Feature representation
Gaussian transformer encoder
Data-driven reconstruction

ABSTRACT

Transformer-based deep learning methods have significantly facilitated multivariate time series classification (MTSC) tasks. However, due to the inherent operation of self-attention mechanism, most existing methods tend to overlook the internal local features and temporal invariance of time series, potentially resulting in a limited understanding of the representation and context information within the model. In contrast to global features, local features demonstrate greater specificity and detail, thereby being more conducive to capture essential texture information and local structures of time series. To ameliorate these problems, we propose CTNet, a novel network that enhances the time series representation learning by reconstructing Crucial Timestamps, aiming to improve the ability to address MTSC tasks. Specifically, we introduce a novel Transformer encoder that incorporates a highly effective Gaussian-prior mechanism to accurately capture local dependencies. Additionally, we present a data-driven mask strategy to boost model's capability of representation learning by reconstructing crucial timestamps. During the reconstruction process, we employ context-aware positional encoding to augment the temporal invariance of the model. Extensive experiments conducted on 30 accessible UEA datasets validate the superiority of CTNet compared to previous competitive methods. Furthermore, ablation studies and visualization analyses are conducted to confirm the effectiveness of the proposed model.

1. Introduction

Multivariate time series are significant data types prevalent across various domains (Wu et al., 2023), which are longitudinally acquired sequences of events, each consisting of observations recorded by multiple attributes (Cheng et al., 2023). Comprehensive analysis of multivariate time series data can facilitate decision-making in numerous intelligent applications, including predicting meteorological factors in weather forecasting (Bi et al., 2023), monitoring medical conditions in medical clinics (Phyo et al., 2022), detecting anomalies in Computer Network Traffic (Protic & Stankovic, 2023), emergency response in freight fires (Tian et al., 2023), and identifying trajectories in intelligent transportation (Zeng et al., 2021). The task of MTSC, which is a fundamental problem in the field of time series analysis (Jastrzebska et al., 2021), has garnered considerable attention owing to its substantial practical value (Li, 2022).

Over the past decades, researchers have dedicated significant efforts to addressing the MTSC problem. Traditional MTSC methods (Ma et al., 2020) can be broadly classified into two primary categories: distance-based methods and feature-based methods. The former distance-based methods classify time series using Dynamic Time Warping (DTW) (Langfu et al., 2023) or Support Vector Machine (SVM) algorithm (Vos et al., 2022). Nevertheless, distanced-based methods often incur exorbitant computational costs when measuring the distances among given sequences. In contrast, feature-based methods exhibit higher efficiency (O'Reilly et al., 2017) but necessitate extensive feature engineering efforts that significantly depend on the expertise of domain specialists.

With the advancements of deep learning (Zhang et al., 2024), researchers have taken up exploring more expressive features to improve classification capabilities (Chen et al., 2022; Karim et al., 2019; Korytkowski et al., 2020). Thanks to the capacity of these methods to autonomously learn discriminative features pertaining to time series in an end-to-end manner, the requirements for manual feature engineering have markedly diminished, with Transformer (Vaswani et al., 2017) emerging as the predominant approach (Korban et al., 2023). Despite the remarkable contributions of Transformer to multivariate time series classification, most recent state-of-the-art (SOTA) models still encounter several challenges: (1) Inadequate local feature representation (Du et al., 2023): Existing deep learning models, including those based on Transformer architectures, have demonstrated substantial success in capturing global dependencies within time series data. However, these models often fall short in effectively capturing local temporal features (Li et al., 2019), thereby potentially impairing the understanding

* Corresponding author.
E-mail addresses: dazhang@mail.nwpu.edu.cn (D. Zhang), gjy3035@gamil.com (J. Gao), xuelong_li@ieee.org (X. Li).

https://doi.org/10.1016/j.eswa.2024.124591
Received 24 January 2024; Received in revised form 14 June 2024; Accepted 22 June 2024
Available online 27 June 2024

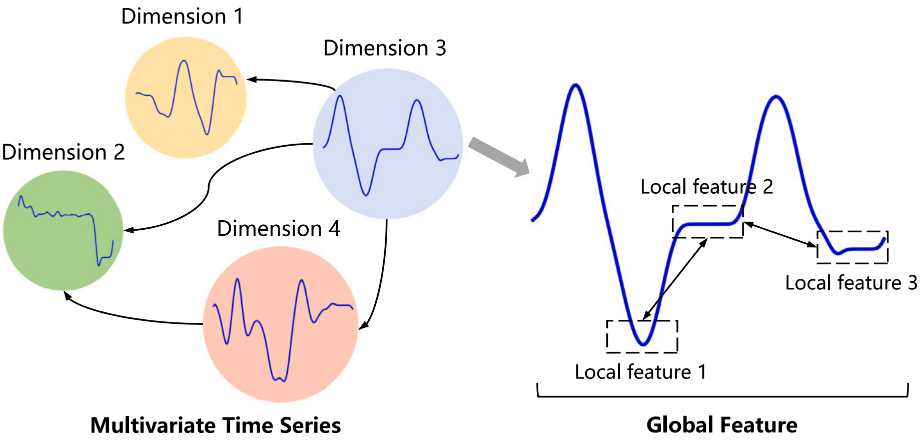

Fig. 1. Different features of multivariate time series. Left: Lines with arrows represent the correlations among different dimensions. Right: The local features are contained within the dashed box, among which the interrelationship compose the global features.

of the representation and context within the model (Dusmanu et al., 2019; Tang et al., 2020). In contrast to global features, these local representations demonstrate greater specificity and detail (Fig. 1), thus being more conducive to capture essential texture information and local structures of time series. (2) Lack of temporal invariance (Cheng et al., 2023): The self-attention mechanism in Vanilla Transformer is permutation-invariant (Vaswani et al., 2017), which is misaligned with the inherent temporal characteristics of time series, hence its temporal-invariance capability of encoder is awfully attenuated (Lohit et al., 2019; Oh et al., 2018). Consequently, this characteristic poses challenges in encoding multivariate time series since it results in a loss of unique temporal order information. This oversight can significantly impact the model's ability to generalize across diverse temporal dynamics and accurately classify time series data. Despite the existence of previous groundbreaking research on Transformer-based approaches, the aforementioned problems are still under investigation in the MTSC task (Wen et al., 2022).

To ameliorate these intractable problems, we propose CTNet, an approach for facilitating the representation learning of time series by reconstructing crucial timestamps, which aims to enhance the performance of MTSC. Specifically, we introduce a novel Transformer encoder that incorporates an enhanced self-attention mechanism by augmenting the effective Gaussian prior probability. This design highlights the contributions of adjacent tokens to the central token, thereby more efficaciously capturing the local dependencies in multivariate time series. Additionally, a data-driven mask strategy is presented to improve the model's capability in learning representations through the reconstruction of crucial timestamps. This scheme involves alternate training between classification task and reconstruction task, with shared parameters in each epoch, accordingly simplifying the network while boosting its overall performance simultaneously. In order to maintain the positional information of the multivariate time series during encoding, we put forward a context-aware positional encoding (PE) to amplify the temporal invariance of the model throughout the reconstruction process.

For the purpose of evaluating the effectiveness of CTNet, we conduct extensive experiments on 30 publicly accessible datasets sourced from the UEA archive (Bagnall et al., 2018). Our CTNet outperforms the current SOTA method by 3.05% in terms of average accuracy, and extensive experiments unambiguously demonstrate that our model exhibits robust classification capabilities. Moreover, various ablation experiments and visualization analyses are carried out to substantiate the effectiveness of distinct components within the proposed method. In summary, our contributions can be summarized as follows:

- We introduce a novel Transformer encoder with an efficient Gaussian-prior mechanism. By emphasizing adjacent tokens, this method can capture the latent local dependencies for time series more efficaciously, thereby enabling a more nuanced representation of time series data.
- A novel data-driven mask strategy is proposed. This strategy, which leverages the reconstruction of Crucial Timestamps, facilitates a more efficient learning process by guiding the network to concentrate on the most informative parts of time series.
- We employ a context-aware positional encoding to preserve crucial temporal information, which not only preserves positional information for different sequences, but also enhances the model's temporal invariance.
- Extensive experiments on 30 publicly available datasets are conducted to demonstrate CTNet's superiority over existing SOTA methods, highlighting its robustness and versatility across a wide range of MTSC tasks.

2. Related works

2.1. Traditional methods for MTSC

Time series classification is an extensively investigated subject in data mining field (Huang & Deng, 2023). Traditional approaches can be categorized into distance-based methods and feature-based methods. Distance-based methods integrate distance or similarity metrics into conventional classifiers such as K-means, K-Nearest Neighbor (KNN) and SVM for MTSC. Borlea et al. (2022) proposed a method to enhance the quality of clusters produced by the K-means algorithm by post-processing these clusters with a supervised learning algorithm. Langfu et al. (2023) employed modified DTW algorithm to oversample samples from satellite time series and integrated it with KNN for cluster analysis to enhance the accuracy of anomaly detection. Gong et al. (2018) introduced a multi-objective learning algorithm for time series approximation and classification, which combined distance metric learning with label information to improve MTSC performance. El Amouri et al. (2023) extended learning DTW-preserving shapelets and presented constrained DTW-preserving shapelets to direct representation learning for time series clustering. However, these methods are computationally intensive because they require measuring the distance between each pair of time series (Hu et al., 2019). On the other hand, feature-based methods focus on extracting discriminative representations from raw data for accurate classification. Traditional approaches have extracted various time series characteristics (such as periodicity, trend, steepness) and salient subsequences like Shapelets (Hills et al., 2014). O'Reilly et al. (2017) introduced an approach that learns representations from time series using autoregressive kernels and then applies similarity metrics based on the learned representations. Pei et al. (2017) presented a hidden-unit logic network for MTSC, which employs binary

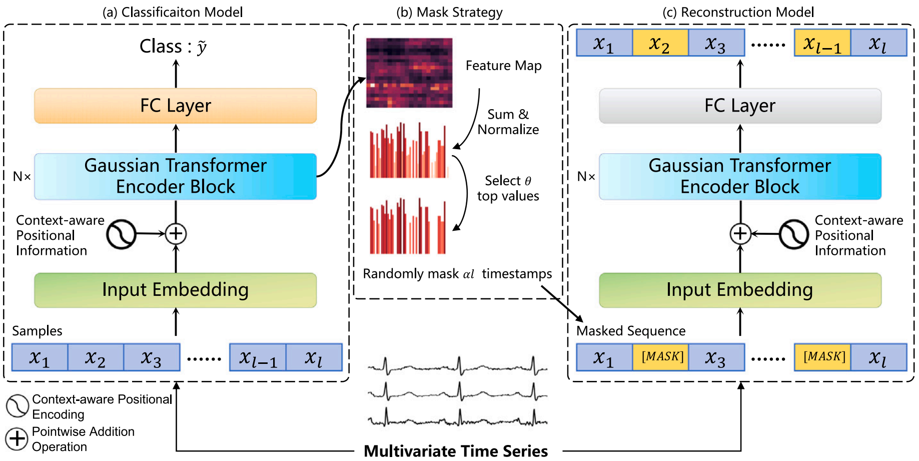
(a) Classificaiton Model
(b) Mask Strategy
(c) Reconstruction Model
Class : $\tilde{y}$
FC Layer
N×
Gaussian Transformer Encoder Block
Context-aware Positional Information
Input Embedding
Samples
x_1 x_2 x_3 ...... x_{l-1} x_l
Feature Map
Sum & Normalize
Select θ top values
Randomly mask αl timestamps
FC Layer
N×
Gaussian Transformer Encoder Block
Context-aware Positional Information
Input Embedding
Masked Sequence
x_1 [MASK] x_3 ...... [MASK] x_l
Context-aware Positional Encoding
Pointwise Addition Operation
Multivariate Time Series

Fig. 2. Overall framework of CTNet for MTSC. (a) Classification Model: Input sequences are firstly normalized before feed into an embedding layer with context-aware positional encoding, and then are send to N-Layer Gaussian Transformer Encoders and Fully Connected (FC) Layer; (b) Mask Strategy: Extracted features of each time-series data are aggregated into feature maps. According to these scores, we decide which timestamps are crucial and perform mask strategy; (c) Reconstruction Model: Masked sequences are reconstructed by N-Layer Gaussian Transformer Encoders. Note that the two FC layers are different and that the Gaussian Transformer Encoder Block is parameter shared.

random hidden units to effectively capture the underlying structure and temporal dependence. Fulcher (2018) summarized the range of feature-based representations for time series which allowed us to understand the properties of specific time series. Nevertheless, feature-based methods heavily depend on domain expert knowledge and necessitate substantial effort in feature engineering and preprocessing (Wang et al., 2017), making such methods difficult to transfer to other time series datasets.

2.2. Deep learning methods for MTSC

With renaissance of deep learning, researchers have started exploring more expressive features to enhance classification performance. Convolutional neural networks (CNNs), as the most popular method, have been proven effectiveness and efficiency in sequence modeling (Ismail Fawaz et al., 2019). Tang et al. (2020) proposed the Omni-Scale Block and incorporated it into the 1D convolution to obtain an adaptive acceptance domain for different datasets, which greatly enhanced the performance of MTSC. Dempster et al. (2020) achieved advanced classification accuracy on UCR time series datasets (Dau et al., 2019) using a simple linear classifier with random convolutional kernal. Furthermore, there exist several classical CNN-based approaches, such as InceptionTime (Fawaz et al., 2020), HiveCOTE (Middlehurst et al., 2021), MultiRocket (Tan et al., 2022), which strive for SOTA performance by integrating diverse time-series representations. However, despite the effectiveness of CNNs, we contend that their classification performance is hindered by their inability to capture global contextual information and their tendency to overlook the temporal relationships present in high-dimensional sequences, which are crucial for time series data smoothing (Lohit et al., 2019).

Designed for sequence modeling, Transformer networks have achieved significant success in the field of natural language processing (NLP) (Devlin et al., 2018; Vaswani et al., 2017). Due to the capacity to capture long-term dependencies and interactions, works based on Transformer have been extensively explored by researchers for various time series analysis tasks, like time series prediction (Zhou et al., 2021), regression (Chowdhury et al., 2022) and anomaly detection (Wang et al., 2024; Yu et al., 2023). However, recent advancements in time series classification with Transformer architecture are still at the early stages and primarily focused on multivariate time series classification (Zerveas et al., 2021). For instance, Ruß wurm and Körner (2020) investigated Transformer into time series classification of original optical satellite images and demonstrated superior performance compared to recurrent neural networks (RNNs) and CNNs. Another approach proposed by Liu et al. (2021), GTN, employed a two-tower Transformer encoder and each branch extracts time-step or channel attention respectively. In order to effectively fuse the features of two towers, the model utilized learned weighted connections (also known as gating). Furthermore, Zerveas et al. (2021) proposed a Transformer-based framework for MTS representation learning, which incorporates an unsupervised pre-training scheme. By fine-tuning on different datasets, the model can be well adapted to different tasks. Although these existing works have shown promising results, they overlook latent local features and temporal invariance of time series, hence the goal of this study is to ameliorate these limitations and improve MTSC task performance.

3. Proposed framework

In this section, we present the development of CTNet. We first introduce the problem definition in Section 3.1. Afterwards, the Gaussian Transformer Encoder with vivid illustrations will be introduced in Section 3.2. Subsequently, we elaborate on the designed data-driven mask strategy in Section 3.3 and context-aware positional encoding in Section 3.4. Fig. 2 illustrates the overall framework.

3.1. Problem definition

(1) Multivariate Time-Series Classification Definition: MTSC is the task of predicting the categorical label associated with a multivariate time series instance. Each instance consists of multiple synchronized time series, each representing a distinct variable observed over time. The goal is to learn a mapping from the multidimensional temporal input space to a set of discrete class labels, based on the intrinsic patterns and dependencies within the data.

(2) Multivariate Time-Series Training Set Definition: A training set $\mathbf{D} = \{(\mathbf{X}_1, y_1), \ldots, (\mathbf{X}_n, y_n)\}$ in MTSC comprises a collection of n labeled instances, where each instance $(\mathbf{X}_i, y_i)$ consists of a multivariate time series $\mathbf{X}_i$ and its corresponding class label y_i. The multivariate time series $\mathbf{X}_i \in \mathbb{R}^{l \times m}$ is composed of m synchronized sequences observed over l time steps, reflecting the multidimensional nature of the data. The class label $y_i \in \{1, \ldots, k\}$ indicates the category to which the instance belongs, out of k possible classes. Note that, in our case, all instances have the same number of time steps in a MTS dataset.

(3) *Crucial Timestamps Definition:* Just as in the field of NLP, where certain words hold significance, and in images, where specific pixel blocks are crucial, time series data also encompasses Crucial Timestamps. In this study, we refer to timestamps with attention weights exceeding a certain threshold in feature maps as “Crucial Timestamps”. We provide a detailed explanation in the subsequent Section 3.3.

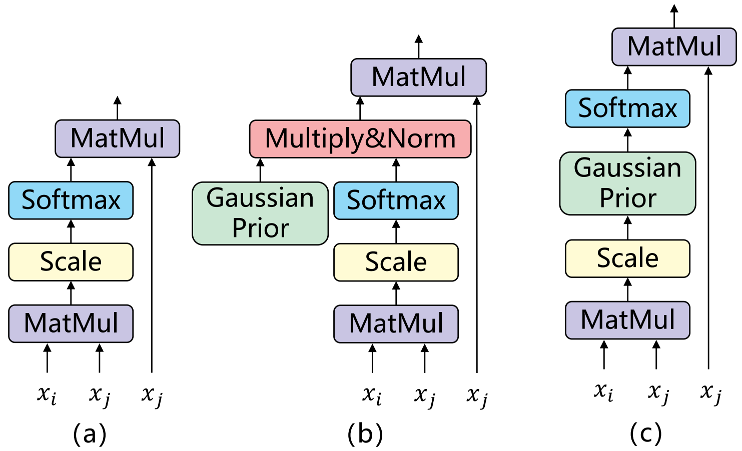

Fig. 3. Illustration of vanilla self-attention and our gaussian variants. (a) Original dot-product self-attention. (b) Gaussian prior self-attention. (c) Gaussian prior self-attention after simplification.

3.2. Gaussian transformer encoder

While the Vanilla Transformer is proficient at modeling global context, it tends to treat tokens at various distances almost equally. However, adjacent tokens practically contribute more significantly to the features of the central token (Guo et al., 2019). Consequently, traditional Transformer models, while effective at capturing global dependencies, often overlook the nuanced local dynamics crucial for accurate time series classification. In order to enhance the model's capability for extracting local features in time series data, we introduce a novel Transformer Encoder with effective Gaussian prior probability. At the same time, for the sake of simplicity, we maintain the rest of the structure the same as the Vanilla Transformer encoder.

Assuming that x_i represents a token in time series $\mathbf{X}$, the ordinary dot-product self-attention mechanism is depicted in Fig. 3(a). Self-attention is a kind of soft-align mechanism, and it employs compatibility function calculated by Softmax to <u>s</u>oft-<u>a</u>lign between x_i and other token x_j from $\mathbf{X}$, i.e.:

$$SA_{i,j} = Softmax(x_i \cdot x_j), \tag{1}$$

where SA represents soft-align function. Then we sum all weight values for the score of x_i,

$$Score_{x_i} = \sum_j SA_{i,j} x_j. \tag{2}$$

In order to enhance modeling of local structure for temporal sequences, we attenuate the significance of distant tokens relative to x_i while elevating the importance of adjacent tokens. We firstly hypothesize that contributions of tokens at varying distances to the semantic features associated with x_i conform to a normal distribution. The normal distribution is selected because it is challenging to quantify the semantic significance of one token relative to another and comparative experiments with different decay mechanisms (Brookes, 1968) have demonstrated the superior performance of the Gaussian hypothesis. Therefore, we employ a standard normal distribution with the variance of $\sigma^2 = 1/(2\pi)$ and the probability density function are $\psi(v) = e^{-\pi v^2}$, where v is the distance between different tokens. After that, $\psi(v_{i,j})$ is inserted into Eq. (1) and add normalization to correct the importance of tokens (Fig. 3(b)):

$$\begin{aligned} Score_{x_i} &= \sum_j \frac{\psi(v_{i,j}) SA_{i,j}}{\sum_k \psi(v_{i,k}) SA_{i,k}} x_j \\ &= \sum_j \frac{e^{-v_{i,j}^2 + x_i \cdot x_j}}{\sum_k e^{-v_{i,k}^2 + x_i \cdot x_k}} x_j \\ &= \sum_j Softmax(-v_{i,j}^2 + x_i \cdot x_j) x_j. \end{aligned} \tag{3}$$

Since under normal circumstances, the variance does not coincidentally equal $1/(2\pi)$, we employ a modified Gaussian prior. Specially, we introduce a learnable linear factor ω and a bias term β, enhancing the salience of the local feature in general scenario (Eq. (4)).

$$Score_{x_i} = \sum_j Softmax(-|\omega v_{i,j}^2 + \beta| + x_i \cdot x_j) x_j. \tag{4}$$

3.3. Data-driven mask and reconstruction

To further enhance the modeling of local features within time series data and to improve the model's generalization across diverse time series datasets, we explore the concept of data reconstruction in MAE (He et al., 2022) to enhance the model's representational capacity. Representation learning through data reconstruction has been investigated in the field of NLP (Devlin et al., 2018) and time series (Zerveas et al., 2021). Data reconstruction in time series Transformer typically involves the random masking of timestamps followed by subsequent reconstruction. Nevertheless, it is important to acknowledge that distinct timestamps within time series data hold varying degrees of significance in the context of representation learning. For example, temporal intervals characterized by frequent fluctuations in time series often encapsulate richer information compared to flat temporal intervals. Consequently, as illustrated in Fig. 2(b), we devise a novel data-driven masking strategy to judiciously mask the input data $\mathbf{X}$ and then employ reconstruction, which can obviously enhance the model's capacity for representing learning of time series. This inclusion of a Data-Driven Reconstruction structure is motivated by the aim to enrich the model's ability to learn discriminative and meaningful features from crucial timestamps, thereby developing a deeper understanding of temporal sequences for more accurate classification outcomes.

In order to clearly elucidate the notion of Crucial Timestamps, we introduce the self-attention weight derived from the Gaussian Transformer Encoder. This weight signifies the allocation of weights to each x_i for computing the representation score. Then we aggregate all attention weights generated by each layer of Gaussian Transformer Encoder to get the feature map $M \in \mathbb{R}^{l \times l}$ of the input temporal sequences. M_{ij} represents the attention weight assigned to x_j during updating x_i, in which $i, j \in \{1, 2, \dots, l\}$ and $\sum_{j=1}^{l} M_{ij} = 1$. Ultimately, $\tau_i \in \mathbb{R}^l$ represents the normalized aggregated attention weight of timestamp x_i (Note that τ is different from $Score$ cause it is the aggregation of each attention layer's $Score$ in encoder block):

$$\tau_i = \frac{\sum_{i=1}^{l} M_{ij}}{\sum_{j=1}^{l} \sum_{i=1}^{l} M_{ij}}. \tag{5}$$

After this, we define the importance of each different timestamps according to the corresponding value in τ. The higher τ_i value is, the more crucial timestamp x_i is. Then a regularization parameter $\theta (0 < \theta < 1)$ is introduced to control the range of crucial timestamps. That is, θ is the threshold to determine which timestamps are important.

Given that model may tend to memorize specific timestamps with in the samples, which could lead to overfitting, we take measures to guarantee that, for each sample within every epoch, the model will randomly select a diverse set of crucial timestamps. Thus we select αl timestamps ($0 < \alpha < \theta$ is mask ratio and l is sequence length) with high attention-weight value in the whole sequence $\mathbf{X}$ and mask them for the subsequent reconstruction. Finally, αl crucial timestamps are selected randomly from $\mathbf{X}$ following by $\tau' = \theta\tau$ to generate the mask λ ($\lambda = \{0, 1\}$). When $\lambda_i = 1$, x_i will be masked, otherwise not. To ensure the precision of the reconstructed sequence, we employ the Mean Square Error (MSE) to calculate the loss between the predicted $\tilde{\mathbf{X}}$ and ground truth $\mathbf{X}$:

$$\mathcal{L}_{rec} = \frac{\sum_{i=1}^{l} \lambda_i \|\tilde{x}_i - x_i\|_2^2}{m \sum_{i=1}^{l} \lambda_i}. \tag{6}$$

For classification, the prediction y is subjected to the softmax operation to generate the probability distribution ρ across k class categories. Subsequently, we employ the cross-entropy to calculate classification loss:

$$\mathcal{L}_{cls} = \sum_{i=1}^{k} y_i log(\rho_i). \tag{7}$$

The primary goal of our learning process is to minimize the classification error across a multivariate time series dataset. Algorithm 1 briefly outlined training of CTNet.

Algorithm 1 Training CTNet.

Input: $\mathbf{X} = \{x_1, x_2, \cdots, x_{l-1}, x_l\}$; $y \in \{1, \cdots, k\}$.
Hyper-parameters: Random sample rate α; Control of crucial range θ.

```
Model = Gaussian Transformer Encoder()
$\tau = 0$ # Initial settings or randomly initialized
while training do
for $x_i$ in $\mathbf{X} = \{x_1, x_2, \cdots, x_{l-1}, x_l\}$ do
    $Score_{x_i} = \sum_j Softmax(-|\omega v_{i,j}^2 + \beta| + x_i \cdot x_j)x_j$ # Eq. (4), Calculate scores of timestamps
    $\mathbf{M} \sim$ Feature map, all attention weights generated by each layer of Gaussian Transformer encoder
    $\tau_i = \sum_{i=1}^{l} M_{ij} / \sum_{j=1}^{l}\sum_{i=1}^{l} M_{ij}$ # Eq. (5), Sum scores and get $\tau$
end for
$\tau' = top\ \theta l$ # Select top values from $\tau$
$\lambda \sim$ Boolean vector, randomly sample $\alpha l$ timestamps according to $\tau'$
$\tilde{\mathbf{X}}$, $\tilde{y}$, $\mathbf{M}$ = Model.train($\mathbf{X}$, $\tau$)
Update Model
end while
```

3.4. Context-aware positional encoding

Transformer, a feed-forward architecture, is notably insensitive to the input order (Pal et al., 2023), yet its success partly hinges on the use of positional encoding operations (Vaswani et al., 2017). This encoding is pivotal due to the self-attention mechanism's ability to preserve the sequential attributes of input data, allowing Transformers to handle both absolute and relative forms of positional information. Absolute encoding provides a comprehensive view of temporal sequences by assigning unique positional codes to each timestamp, offering extensive temporal context. However, this method can dilute the temporal invariance critical to time series analysis—defined as the model's capacity to recognize patterns irrespective of their occurrence in the sequence. This quality is essential for the model to generalize well across variations in time, such as shifts or dilations, by accurately identifying and classifying consistent temporal patterns, thereby enhancing its predictive accuracy and robustness.

Conversely, while relative encoding can address the limitations of absolute encoding by focusing on the relationships between timestamps, it may neglect essential absolute temporal information. Such information is crucial for tasks where the timing of specific events within a sequence is key to accurate classification, like identifying particular patterns in ECG signals indicative of health conditions. Our novel approach, the context-aware positional encoding, aims to merge the strengths of both encoding types. It seeks to ensure temporal invariance by recognizing pattern consistency while retaining the capability to utilize absolute temporal cues crucial for certain classification tasks. By emphasizing context information around adjacent sequences, this encoding strategy enhances the model's ability to capture the dynamics of time series data, where each timestamp's significance is potentially augmented by its neighbors, without losing sight of the overarching chronological framework of the series.

Specifically, we investigate and conclude that context information representing adjacent local sequences is sufficient (Chu et al., 2021). We adopt a $1D$ convolution operation with zero padding ($kernel\ size \geq 3$) as context-aware positional encoding for multivariate time series (Mohamed et al., 2019). The advantage of this approach lies in its sensitivity to positional information while maintaining translation equivalence, and its natural ability to handle different temporal sequences. Furthermore, the zero-padding operation can to some extent encode the absolute position information in the sequence (Islam et al., 2020).

4. Experiments

In this section, extensive experiments on 30 publicly available multivariate time-series datasets are conducted to verify our method. Firstly, we describe the datasets from UEA archive and outline experimental settings. Subsequently, we make comparisons with different SOTA methods. Ultimately, we demonstrate the effectiveness of the CTNet through ablation study and visualization analysis. Experiments are carried out 5 times on NVIDIA GeForce RTX 3090 platform equipped with 24G video memory and averaged as final results.

4.1. Experimental setup

4.1.1. Datasets

For our experiments, we select 30 public datasets from the renowned UEA Multivariate Time Series Classification Archive (Bagnall et al., 2018) to conduct MTSC task. Factually, the UEA Archive has become the predominant benchmark for MTSC in contemporary research. These datasets come from a variety of fields, such as Human Activity, Audio Spectra, Electrocardiogram (ECG), Electroencephalogram (EEG), etc., and are sampled at different frequencies by a variety of sensors. Hence, each dataset exhibits unique characteristics in terms of sample size, length, and class distribution. Among the raw data, both the training sets and test sets have undergone thorough processing, thus requiring no further manipulation to ensure equitable comparison. Table 1 provides a comprehensive overview of the detailed information contained within these datasets.

4.1.2. Compared baselines

For a comprehensive assessment, we select the following competitive baseline methods. There are both CNN-based methods and Transformer-based methods, and DTW is a distance-based method.

- **TS2Vec**[1] (Yue et al., 2022) aggregates representations of arbitrary timestamps through hierarchical contrastive learning.

[1] https://github.com/yuezhihan/ts2vec

Table 1
Properties of all datasets.

Datasets	Train	Test	Dimensions	Length	Classes	Task	Type
ArticularyWordRecognition	300	275	9	144	25	Speech Recognition	Motion
AtrialFibrillation	15	15	2	640	3	ECG Classification	ECG
BasicMotions	40	40	6	100	4	Human Action Recognition	HAR
CharacterTrajectories	1422	1436	3	182	20	Primitive Extraction	Motion
Cricket	108	72	6	1197	12	Simulation Signal Recognition	HAR
DuckDuckGeese	50	50	1345	270	5	Species Classification	Audio
EigenWorms	128	131	6	17 984	5	Species Recognition	Motion
Epilepsy	137	138	3	206	4	Human Action Recognition	HAR
ERing	30	270	4	65	6	Finger Posture Recognition	HAR
EthanolConcentration	261	263	3	1751	4	Spectral Classification	Other
FaceDetection	5890	3524	144	62	2	EEG Classification	EEG
FingerMovements	316	100	28	50	2	Finger Movements Recognition	EEG
HandMovementDirection	160	74	10	400	4	Wrist Movements Recognition	EEG
Handwriting	150	850	3	152	26	Motion Recognition	HAR
Heartbeat	204	205	61	405	2	Heart Sound Recordings	Audio
InsectWingbeat	30 000	20 000	200	30	10	Insect Classification	Audio
JapaneseVowels	270	370	12	29	9	Predict Speakers	Audio
Libras	180	180	2	45	15	Hand Movements Classification	HAR
LSST	2459	2466	6	36	14	Astronomical Series Classification	Other
MotorImagery	278	100	64	3000	2	EEG Classification	EEG
NATOPS	180	180	24	51	6	Arm Movements Classification	HAR
PEMS-SF	267	173	963	144	7	Traffic Flow Classification	MISC
PenDigits	7494	3498	2	8	10	Handwritten Digit Classification	Motion
PhonemeSpectra	3315	3353	11	217	39	Phonetic Classification	Sound
RacketSports	151	152	6	30	4	Human Action Recognition	HAR
SelfRegulationSCP1	268	293	6	896	2	EEG Classification	EEG
SelfRegulationSCP2	200	180	7	1152	2	EEG Classification	EEG
SpokenArabicDigits	6599	2199	13	93	10	Speech Recognition	Speech
StandWalkJump	12	15	4	2500	3	ECG Classification	ECG
UWaveGestureLibrary	120	320	3	315	8	Posture Recognition	HAR

- **TapNet**[2] (Zhang et al., 2020) combines multi-layer CNN to design a random grouping permutation to learn low-dimensional features in different time series.
- **MICOS**[3] (Hao et al., 2023) proposes a hybrid supervised contrast loss, and on this basis introduces a contrast learning framework for MTSC.
- **TNC**[4] (Tonekaboni et al., 2021) employs local smoothness of the signal generation process to define temporal neighborhoods.
- **DKN** (Xiao et al., 2024) proposes a densely knowledge-aware network to enable dense mutual supervision between lower-and higher-level semantic information.
- **Formertime**[5] (Cheng et al., 2023) proposes a hierarchical multi-scale network to learn multilevel representations of multivariate time series.
- **FEAT** (Kim et al., 2023) flexibly learns unique patterns specific to features while dynamically changing temporal patterns.
- **TS-TCC**[6] (Eldele et al., 2021) employs time-series representation learning via temporal and contextual contrasting with different data augmentations.
- **TST**[7] (Zerveas et al., 2021) provides an unsupervised pre-training programme for multivariate time series.

Different datasets are standardized for each experiment and we provide the same results based on reported papers with published baselines. Accuracy is the most common classification metric, representing the proportion of correctly classified samples. We leverage accuracy as the metric for the remaining datasets. In Eq. (8), *TP* and *TN* are the test result that correctly indicates the positive and negative samples, and *FN* and *FP* denote the test result that wrongly indicates the positive and negative samples, correspondingly. The higher the accuracy, the better the classification performance of the model.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{8}$$

4.1.3. Implementation details

During the training phase, our setup closely mirrors the original Transformer model in terms of the number of encoders and attention heads to maintain consistency with established benchmarks. We selected an initial learning rate of 0.001 and utilized the AdamW optimizer, a decision based on its proven effectiveness in similar contexts. We set the number of training epochs to 300 based on a combination of experience and experimental results and save the best-performing checkpoints during training and used these checkpoints for evaluation on the test set. We also implement several measures like using data augmentation and L2 regularization to prevent overfitting. The hyperparameters θ and α were meticulously chosen as 0.5 and 0.15, respectively, after careful experimentation to optimize our model's performance. For a comprehensive and equitable comparison with alternative methods, we adhered to configurations provided in their official implementations, making adjustments only as recommended for optimal performance on our test datasets. To ensure reliability and fairness in our comparative analysis, all models were trained to their peak performance, with each experiment conducted five times to account for variability, thus allowing us to present the average outcomes as a robust measure of each strategy's efficacy. We will make our code available at https://github.com/zhangda1018/CTNet.

4.2. Experimental results

4.2.1. Classification performance evaluation

Table 2 shows the accuracy results of different models. The overall accuracy of CTNet according to Table 2 is the best of all comparison methods. Our CTNet performs best on 14 datasets, while the next best baseline DKN and MICOS perform best on 8 and 6 datasets, respectively. CTNet's average accuracy across all datasets is 3.1 percentage points higher than MICOS. CTNet's closest competing baselines are

[2] https://github.com/kdd2019-tapnet/tapnet
[3] https://github.com/haoshileii/micos
[4] https://github.com/sanatonek/TNC_representation_learning
[5] https://github.com/Mingyue-Cheng/FormerTime
[6] https://github.com/emadeldeen24/TS-TCC
[7] https://github.com/gzerveas/mvts_transformer

Table 2
Multivariate time-series classification results in terms of average accuracy and average rank. Note that as there is no official result of InsectWingbeat from DTW in the UEA archive, we excluded it when computing the average accuracy and rank. The best and second-best results are highlighted in **bold** and underlined, respectively.

Datasets	TS2Vec	TNC	TS-TCC	TST	TapNet	MICOS	DKN	Formertime	FEAT	CTNet(Ours)
ArticularyWordRecognition	0.987	0.973	0.953	0.977	0.987	0.990	**0.993**	0.984	0.991	0.987
AtrialFibrillation	0.200	0.133	0.267	0.067	0.333	0.333	0.467	0.600	0.293	**1.000**
BasicMotions	0.975	0.975	**1.000**	0.975	**1.000**	**1.000**	**1.000**	1.000	**1.000**	**1.000**
CharacterTrajectories	0.995	0.967	0.985	0.975	**0.997**	0.994	0.986	0.991	0.993	0.992
Cricket	0.972	0.958	0.917	**1.000**	0.958	**1.000**	0.951	0.981	0.969	**1.000**
DuckDuckGeese	0.680	0.460	0.380	0.620	0.575	0.740	0.560	0.600	0.564	**0.760**
EigenWorms	0.847	0.840	0.779	0.748	0.489	**0.901**	0.628	0.618	0.811	0.840
Epilepsy	0.964	0.957	0.957	0.949	0.971	0.971	**0.979**	0.964	0.948	0.964
ERing	0.874	0.852	0.904	0.874	0.133	0.941	0.933	0.904	0.896	**0.944**
EthanolConcentration	0.308	0.297	0.285	0.262	0.323	0.247	0.372	**0.485**	0.322	0.312
FaceDetection	0.501	0.536	0.544	0.534	0.556	0.523	0.631	**0.687**	0.530	0.646
FingerMovements	0.480	0.470	0.460	0.560	0.530	0.570	0.600	0.618	0.488	**0.670**
HandMovementDirection	0.338	0.324	0.243	0.243	0.378	0.649	**0.662**	0.567	0.378	0.567
Handwriting	0.515	0.249	0.498	0.225	0.357	**0.621**	0.231	0.214	0.542	0.399
Heartbeat	0.683	0.746	0.751	0.746	0.751	**0.766**	0.765	0.761	0.746	**0.766**
InsectWingbeat	0.466	**0.469**	0.264	0.105	0.208	0.218	0.362	0.227	0.462	0.322
JapaneseVowels	0.984	0.978	0.930	0.978	0.965	0.989	0.930	0.964	0.983	**0.991**
Libras	0.867	0.817	0.822	0.656	0.850	0.889	0.900	0.889	0.889	**0.950**
LSST	0.537	0.595	0.474	0.408	0.568	0.667	0.347	0.543	0.548	**0.681**
MotorImagery	0.510	0.500	0.610	0.500	0.590	0.500	0.620	**0.632**	0.562	0.590
NATOPS	0.928	0.911	0.822	0.850	0.939	**0.967**	0.872	0.961	0.921	0.939
PEMS-SF	0.682	0.699	0.734	0.740	0.751	0.809	**0.948**	0.774	0.874	0.855
PenDigits	0.989	0.979	0.974	0.560	0.980	0.981	0.930	0.981	0.989	**0.990**
PhonemeSpectra	0.233	0.207	0.252	0.085	0.175	0.276	**0.525**	0.147	0.216	0.133
RacketSports	0.855	0.776	0.816	0.809	0.868	0.941	0.879	0.842	0.888	**0.947**
SelfRegulationSCP1	0.812	0.799	0.823	0.754	0.652	0.799	**0.913**	0.887	0.852	0.852
SelfRegulationSCP2	0.578	0.550	0.533	0.550	0.550	0.578	**0.600**	0.592	0.562	0.578
SpokenArabicDigits	0.988	0.934	0.970	0.923	0.983	0.981	0.963	0.992	0.986	**0.995**
StandWalkJump	0.467	0.400	0.333	0.267	0.400	0.533	0.533	0.533	0.533	**0.667**
UWaveGestureLibrary	0.906	0.759	0.753	0.575	0.894	0.891	0.897	0.888	**0.929**	0.847
Total Best Acc.	0	1	1	1	2	6	8	3	2	**14**
Avg. Acc.	0.704	0.670	0.668	0.617	0.657	0.742	0.732	0.727	0.722	**0.773**
Avg. Rank	5.42	7.47	7.32	8.25	5.78	3.87	4.53	4.50	4.73	**3.13**

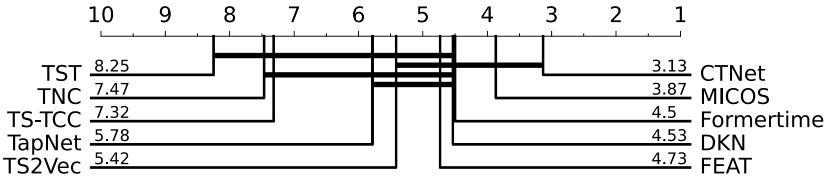

Fig. 4. Critical Difference (CD) diagram of different methods on 30 multivariate time series datasets. The smaller the value, the better the classification result.

MICOS and DKN, but CTNet still outperforms both of them on 13 datasets but loses to them on 9 and 10 datasets respectively. CTNet's average accuracy ranks first, 0.74-point below the second-best MICOS. Formertime and DKN rank third and fourth, with average scores 1.37 and 1.40 points higher than CTNet.

The extensive use of diverse datasets and baselines makes it unlikely for a single model to outperform all other methods on each dataset. For example, DKN's "Total Best Acc." (8) rank second, but its average accuracy and average rank are lower than MICOS's. However, CTNet performs relatively well on all of the evaluated metrics, not only having the highest "Total Best Acc." (14) and "Avg. Acc" (0.773), but also ranking first, which means that for datasets where CTNet performs poorly, it still produces better performance than most of its peers. Critical Difference diagram for Nemenyi tests (Demšar, 2006) on all datasets is presented in Fig. 4, where classifiers that are not connected by a bold line are significantly different in average ranks.

4.2.2. Study of Gaussian prior

We investigate the impact of incorporating Gaussian Prior into the Transformer model, aiming to ascertain its effect on enhancing the model's attentiveness to local features within time series data. This inquiry stems from the acknowledgment that while the sinusoidal position coding in the vanilla Transformer architecture is adept at encoding relative positional information, there exists a potential for further improvement in model performance through the Gaussian hypothesis. This hypothesis posits that by modulating self-attention mechanisms to emphasize local tokens, we can achieve a more nuanced understanding of time series dynamics. To rigorously test this proposition, we experimented with various models, adjusting the prior probabilities to examine their influence on classification accuracy across a comprehensive range of datasets. The findings, as delineated in Fig. 5, reveal a significant pattern: incorporating Gaussian Prior generally results in a marked improvement in classification accuracy for the majority of datasets. This effect is especially pronounced in specific datasets such as AtrialFibrillation and StandWalkJump, underscoring the potential benefits of enhanced local feature modeling in certain contexts. However, it is crucial to note that this improvement is not uniformly observed across all datasets. In instances like UWaveGestureLibrary, the augmented model did not surpass the performance of the vanilla Transformer. This variance underscores the nuanced nature of time series data and suggests that the efficacy of the Gaussian Prior may be contingent upon the intrinsic characteristics of the datasets. Not all temporal sequences may benefit from an increased focus on local dependencies, highlighting the importance of dataset-specific considerations in model optimization. Further refinement of the Gaussian Prior approach, through the introduction of linear factors and bias adjustments, led to the enhanced performance of our CTNet framework across the board. This variant demonstrates CTNet's robustness

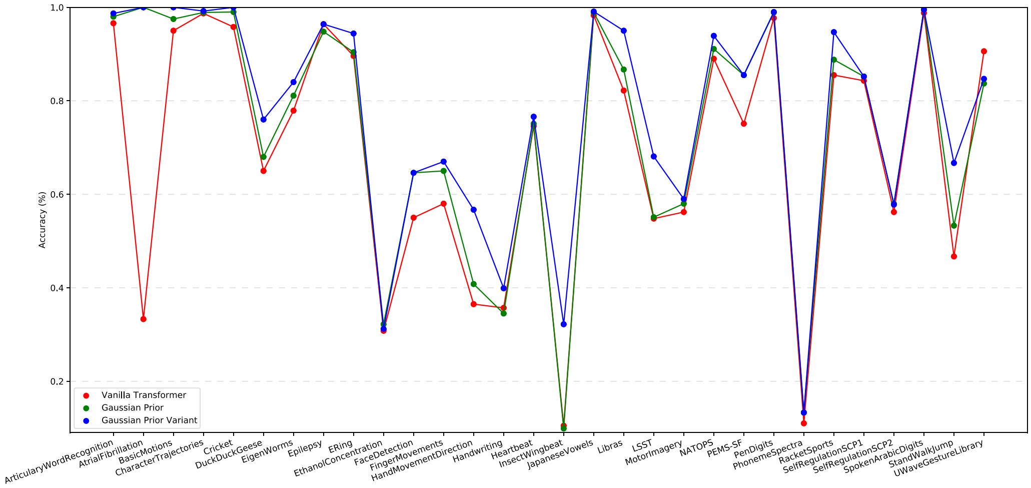

Fig. 5. Accuracy of different Gaussian prior variants on all datasets. Red line represents Vanilla Transformer. Green line represents model with Gaussian Prior (Eq. (3)). Blue lines represents model with Gaussian Prior Variant (Eq. (4)).

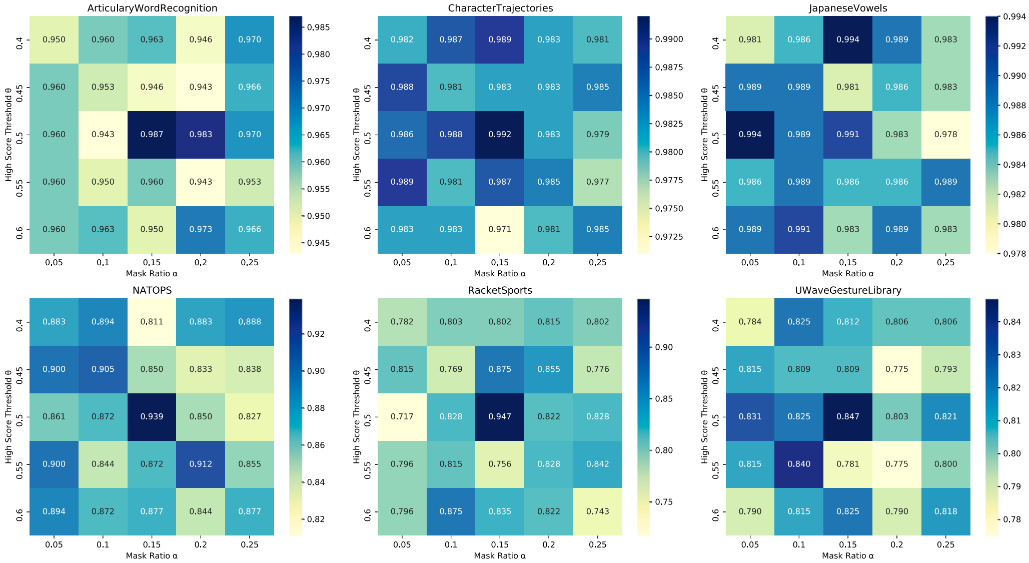

Fig. 6. Heatmaps of accuracy on six different datasets with varying mask ratio and high-score interval. The darker the color, the better the result.

and adaptability, showcasing its superior capability in modeling local features and its strong generalizability across diverse dataset tasks. We posit the outcomes not only affirm the validity of the Gaussian Prior hypothesis but also illuminate the critical balance between global context and local feature emphasis in time series classification.

4.2.3. Analysis of masking strategy

A fundamental component of CTNet involves treating sequences as fundamental semantic units and selectively masking a percentage of subsequences with high self-attention scores to form broken inputs. Consequently, we investigate the definition of high-score interval and the impact of mask ratio on classification accuracy. It is important to note that we maintain consistent settings for all other hyper-parameters to ensure equitable comparisons. In Fig. 6, we give the heatmaps of classification accuracy of each dataset in different configurations, where the darker the color is, the better the result is, representing model works best under this configuration. According to the results we obtained, in order to make the model have better generalization, we employ a uniform setting, that is, the mask ratio of 0.15 and the high-score interval is 0.5.

We present the corresponding experimental results across six carefully chosen datasets in Fig. 6. From these findings, we draw the following conclusions: Firstly, it is evident that both factors exert a notable influence on the model's generalization performance. The former factor highlights the relative importance of fundamental semantic elements, while the latter signifies the complexity of the recovery task. As a result, different hyper-parameter choices can significantly impact the classification outcomes. Notably, on the RacketSports dataset, we observe that the CTNet model exhibits greater sensitivity to the mask ratio than to the high-score interval setting, and a similar trend is discernible on the UWaveGestureLibrary dataset. What is more, we find that not all datasets are sensitive to varying changes in hyper-parameters. For example, when mask ratio is 0.05, the accuracy of ArticularyWordRecognition barely changes with high-score interval varying. These observations notably reflect the considerable diversity across datasets.

4.2.4. Study of positional encoding

We delve deeply into the nuances of positional encoding to assess the relative contributions of different encoding strategies to the overall performance of our CTNet framework. The average results of these experiments across all datasets are presented in Table 3. This investigation is pivotal, as it disentangles the effects of context-aware positional encoding from those achieved through static absolute position encoding, learnable relative position encoding, and scenarios devoid of any

Table 3
Multivariate time-series classification results with different positional encoding. “WPE”, “APE”, “RPE”, “CAPE” represent “Without PE”, “Absolute PE”, “Relative PE”, “Context-Aware PE”. The best and second-best results are highlighted in **bold** and underlined, respectively.

Datasets	WPE	APE	RPE	CAPE	Datasets	WPE	APE	RPE	CAPE
ArticularyWordRecognition	0.927	0.963	**0.987**	**0.987**	InsectWingbeat	0.101	0.100	0.101	**0.322**
AtrialFibrillation	**1.000**	**1.000**	**1.000**	**1.000**	JapaneseVowels	0.703	0.984	0.989	**0.991**
BasicMotions	0.975	**1.000**	**1.000**	**1.000**	Libras	0.794	0.850	0.867	**0.950**
CharacterTrajectories	0.961	0.988	0.989	**0.992**	LSST	0.426	0.334	0.509	**0.681**
Cricket	0.986	0.986	0.986	**1.000**	MotorImagery	**0.590**	**0.590**	**0.590**	**0.590**
DuckDuckGeese	0.680	0.720	0.720	**0.760**	NATOPS	0.844	0.767	0.928	**0.939**
EigenWorms	0.618	0.811	0.811	**0.840**	PEMS-SF	**0.867**	0.734	0.855	0.855
Epilepsy	0.906	0.928	0.948	**0.964**	PenDigits	0.960	0.985	0.989	**0.990**
ERing	0.837	0.793	0.874	**0.944**	PhonemeSpectra	0.078	0.126	**0.151**	0.133
EthanolConcentration	0.247	0.247	0.285	**0.312**	RacketSports	0.809	0.829	0.803	**0.947**
FaceDetection	0.545	0.609	0.556	**0.646**	SelfRegulationSCP1	0.775	0.758	**0.852**	**0.852**
FingerMovements	0.620	0.560	0.580	**0.670**	SelfRegulationSCP2	0.544	**0.578**	**0.578**	**0.578**
HandMovementDirection	0.315	0.392	0.419	**0.567**	SpokenArabicDigits	0.108	0.102	0.986	**0.995**
Handwriting	0.181	0.181	**0.451**	0.399	StandWalkJump	0.467	0.467	0.467	**0.667**
Heartbeat	0.727	0.746	0.751	**0.766**	UWaveGestureLibrary	0.666	0.803	**0.847**	**0.847**

positional information. The synthesized results of these comparative analyses are methodically presented in Table 3, offering a panoramic view of CTNet’s performance under varied encoding schemes across a spectrum of datasets. The empirical findings unequivocally demonstrate that the incorporation of context-aware positional encoding significantly elevates CTNet’s performance, propelling it to outshine nearly all alternative encoding variants. This enhancement is particularly notable, as it underscores the value of embedding rich, context-sensitive location information within the model’s architecture. The context-aware approach marries the benefits of capturing both the absolute temporal positions and the dynamic, relative interactions between data points, thereby providing a more holistic understanding of the time series data.

However, an intriguing variance in performance is observed across different datasets. In certain cases, models employing either relative or absolute position encoding strategies momentarily surpass CTNet. This variability highlights the unique characteristics and independence of specific datasets, which may sometimes align more closely with the assumptions underlying these alternative encoding strategies, making contextual information less critical for their classification. Moreover, the study reveals a paradox where the absence of any positional encoding leads to a notable performance decrement in CTNet, reaffirming the integral role of positional information in effective time series analysis. Yet, fascinatingly, in certain datasets such as PEMS-SF, the elimination of positional cues does not deter performance and, in some instances, even results in superior outcomes. This counterintuitive result suggests that the inherent structural robustness and distinguishable features of certain time series can render them less reliant on explicit positional guidance, thereby simplifying the classification task.

4.2.5. *Visualization analysis*

Finally we conclude our analysis with a visual examination of the proposed method. In particular, we employ T-SNE for the visualization of learned features and labels, facilitating a comparison between CTNet and TS2Vec across three datasets. In Fig. 7, we present a visualization of features obtained through random initialization encoding, followed by a comparative assessment of the two methods’ outcomes. The figure vividly demonstrates that, in contrast to TS2Vec, instances within the same class are notably more tightly clustered and distinctly separated from other classes after undergoing CTNet processing.

Moreover, we generate heat maps to visualize the instance features of the CharacterTrajectories dataset Fig. 8. The dataset comprises three-dimensional samples, and we select one to present the characteristics of each dimension. The figure illustrates that our model effectively detects local data features overall and exhibits distinct sensitivity to changes in each dimension. For instance, dimensions 1 and 2 exhibit similar changing trends, with the latter showing a slight lag. Furthermore, dimension 3 significantly deviates from other dimensional features, and this difference in feature representation is observable in the figure. All the aforementioned results collectively demonstrate the remarkable effectiveness of CTNet in feature modeling.

5. Conclusion and future work

5.1. *Conclusion*

In this study, we introduce CTNet, a network designed to enhance the classification performance of multivariate time series by reconstructing crucial timestamps. CTNet is designed to enhance the intrinsic local features of time series and utilize this information for acquiring comprehensive representations. We have devised an innovative Gaussian prior self-attention mechanism and employ context-aware positional coding to enhance the modeling of temporal dependencies in time series. We conducted comprehensive experiments on 30 publicly available datasets and attained exceptional performance. In addition, the reconstruction process of CTNet is completely unsupervised, so we believe that CTNet can also be extended to tasks such as regression (Ilic et al., 2021) and anomaly detection (Audibert et al., 2022).

5.2. *Future work*

The method proposed in this paper shows potential for application in energy consumption estimation (Połap et al., 2023; Savi & Olivadese, 2021; Yan et al., 2023). However, the centralized approach used in this paper to process data and train models has certain limitations regarding data privacy and computational resources. Future research can address these issues by introducing decentralized Federated Learning (FL) to handle the distributed data processing tasks of multiple households. Specifically, future work can combine the modeling methods proposed in this paper with distributed training under the FL framework. This approach not only enables more personalized and accurate energy consumption predictions but also promotes collaborative optimization among households.

CRediT authorship contribution statement

Da Zhang: Conceptualization, Methodology, Validation, Formal analysis, Investigation, Writing – original draft, Visualization. **Junyu Gao:** Writing – review & editing, Project administration. **Xuelong Li:** Supervision, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

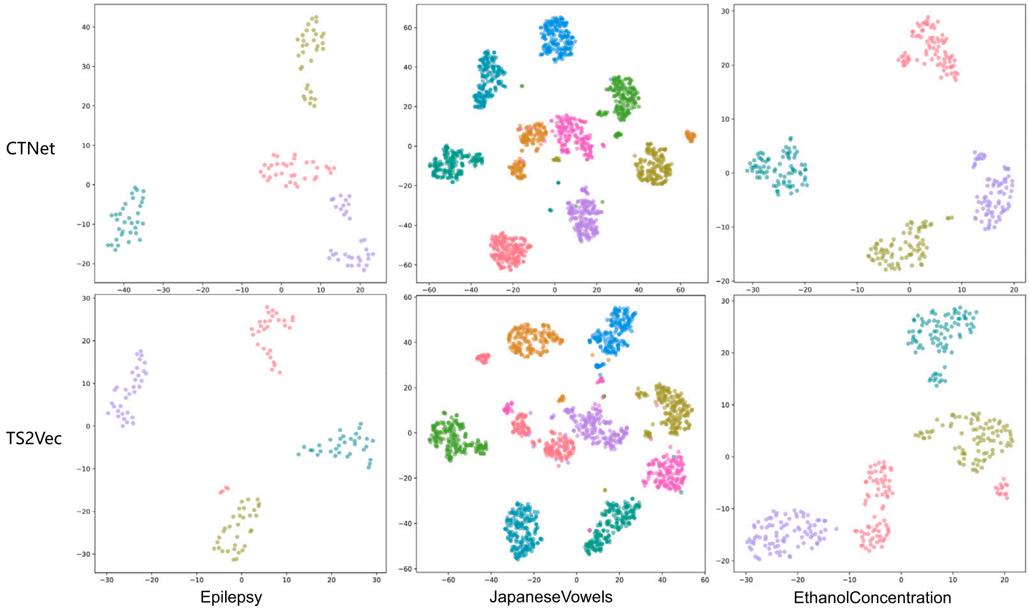

Fig. 7. T-SNE visualization of representations from CTNet (first line) and TS2Vec (second line).

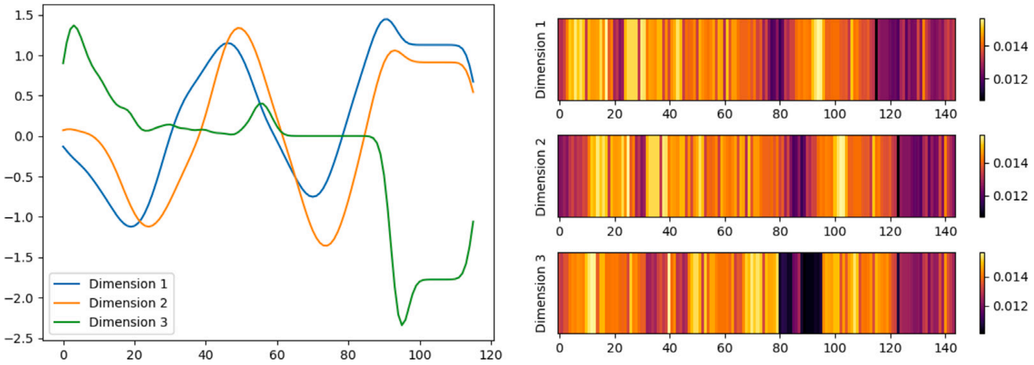

Fig. 8. Heatmap visualizations of feature-specific representation from CTNet. Instance is selected from CharacterTrajectories dataset. Sampled features of each dimension are plotted from the original time-series data.

Acknowledgments

This work was supported in part by the National Key Research and Development Program of China under Grant 2022ZD0160400 and in part by the Natural Science Foundation of China under Grant 62306241 and Grant 61871470.

References

Audibert, J., Michiardi, P., Guyard, F., Marti, S., & Zuluaga, M. A. (2022). Do deep neural networks contribute to multivariate time series anomaly detection? *Pattern Recognition*, *132*, Article 108945.

Bagnall, A., Dau, H. A., Lines, J., Flynn, M., Large, J., Bostrom, A., Southam, P., & Keogh, E. (2018). The UEA multivariate time series classification archive, 2018. arXiv preprint arXiv:1811.00075.

Bi, K., Xie, L., Zhang, H., Chen, X., Gu, X., & Tian, Q. (2023). Accurate medium-range global weather forecasting with 3D neural networks. *Nature*, 1–6.

Borlea, I.-D., Precup, R.-E., & Borlea, A.-B. (2022). Improvement of K-means cluster quality by post processing resulted clusters. *Procedia Computer Science*, *199*, 63–70.

Brookes, B. C. (1968). The derivation and application of the Bradford-Zipf distribution. *Journal of Documentation*, *24*(4), 247–265.

Chen, Y., Ding, F., & Zhai, L. (2022). Multi-scale temporal features extraction based graph convolutional network with attention for multivariate time series prediction. *Expert Systems with Applications*, *200*, Article 117011.

Cheng, M., Liu, Q., Liu, Z., Li, Z., Luo, Y., & Chen, E. (2023). FormerTime: Hierarchical multi-scale representations for multivariate time series classification. arXiv preprint arXiv:2302.09818.

Chowdhury, R. R., Zhang, X., Shang, J., Gupta, R. K., & Hong, D. (2022). Tarnet: Task-aware reconstruction for time-series transformer. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining* (pp. 212–220).

Chu, X., Tian, Z., Zhang, B., Wang, X., Wei, X., Xia, H., & Shen, C. (2021). Conditional positional encodings for vision transformers. arXiv preprint arXiv:2102.10882.

Dau, H. A., Bagnall, A., Kamgar, K., Yeh, C.-C. M., Zhu, Y., Gharghabi, S., Ratanamahatana, C. A., & Keogh, E. (2019). The UCR time series archive. *IEEE/CAA Journal of Automatica Sinica*, *6*(6), 1293–1305.

Dempster, A., Petitjean, F., & Webb, G. I. (2020). ROCKET: Exceptionally fast and accurate time series classification using random convolutional kernels. *Data Mining and Knowledge Discovery*, *34*(5), 1454–1495.

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, *7*, 1–30.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Du, M., Wei, Y., Zheng, X., & Ji, C. (2023). Multi-feature based network for multivariate time series classification. *Information Sciences*, *639*, Article 119009.

Dusmanu, M., Rocco, I., Pajdla, T., Pollefeys, M., Sivic, J., Torii, A., & Sattler, T. (2019). D2-Net: A trainable CNN for joint description and detection of local features. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 8092–8101).

El Amouri, H., Lampert, T., Gançarski, P., & Mallet, C. (2023). Constrained DTW preserving shapelets for explainable time-series clustering. *Pattern Recognition*, *143*, Article 109804.

Eldele, E., Ragab, M., Chen, Z., Wu, M., Kwoh, C. K., Li, X., & Guan, C. (2021). Time-series representation learning via temporal and contextual contrasting. arXiv preprint arXiv:2106.14112.

Fulcher, B. D. (2018). Feature-based time-series analysis. In *Feature engineering for machine learning and data analytics* (pp. 87–116). CRC Press.

Gong, Z., Chen, H., Yuan, B., & Yao, X. (2018). Multiobjective learning in the model space for time series classification. *IEEE Transactions on Cybernetics*, *49*(3), 918–932.

Guo, M., Zhang, Y., & Liu, T. (2019). Gaussian transformer: A lightweight approach for natural language inference. In *Proceedings of the AAAI conference on artificial intelligence*: *vol. 33*, (pp. 6489–6496).

Hao, S., Wang, Z., Alexander, A. D., Yuan, J., & Zhang, W. (2023). MICOS: Mixed supervised contrastive learning for multivariate time series classification. *Knowledge-Based Systems*, *260*, Article 110158.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., & Girshick, R. (2022). Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 16000–16009).

Hills, J., Lines, J., Baranauskas, E., Mapp, J., & Bagnall, A. (2014). Classification of time series by shapelet transformation. *Data Mining and Knowledge Discovery*, *28*, 851–881.

Hu, M., Feng, X., Ji, Z., Yan, K., & Zhou, S. (2019). A novel computational approach for discord search with local recurrence rates in multivariate time series. *Information Sciences*, *477*, 220–233.

Huang, F., & Deng, Y. (2023). TCGAN: Convolutional generative adversarial network for time series classification and clustering. *Neural Networks*, *165*, 868–883.

Ilic, I., Görgülü, B., Cevik, M., & Baydoğan, M. G. (2021). Explainable boosted linear regression for time series fforecasting. *Pattern Recognition*, *120*, Article 108144.

Islam, M. A., Jia, S., & Bruce, N. D. (2020). How much position information do convolutional neural networks encode? arXiv preprint arXiv:2001.08248.

Ismail Fawaz, H., Forestier, G., Weber, J., Idoumghar, L., & Muller, P.-A. (2019). Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery*, *33*(4), 917–963.

Ismail Fawaz, H., Lucas, B., Forestier, G., Pelletier, C., Schmidt, D. F., Weber, J., Webb, G. I., Idoumghar, L., Muller, P.-A., & Petitjean, F. (2020). Inceptiontime: Finding alexnet for time series classification. *Data Mining and Knowledge Discovery*, *34*(6), 1936–1962.

Jastrzebska, A., Nápoles, G., Homenda, W., & Vanhoof, K. (2021). Fuzzy cognitive map-driven comprehensive time-series classification. *IEEE Transactions on Cybernetics*.

Karim, F., Majumdar, S., Darabi, H., & Harford, S. (2019). Multivariate LSTM-FCNs for time series classification. *Neural Networks*, *116*, 237–245.

Kim, S., Chung, E., & Kang, P. (2023). FEAT: A general framework for feature-aware multivariate time-series representation learning. *Knowledge-Based Systems*, *277*, Article 110790.

Korban, M., Youngs, P., & Acton, S. T. (2023). A multi-modal transformer network for action detection. *Pattern Recognition*, *142*, Article 109713.

Korytkowski, M., Scherer, R., Szajerman, D., Połap, D., & Woźniak, M. (2020). Efficient visual classification by fuzzy rules. In *2020 IEEE international conference on fuzzy systems* (pp. 1–6). IEEE.

Langfu, C., Zhang, Q., Yan, S., Liman, Y., Yixuan, W., Junle, W., & Chenggang, B. (2023). A method for satellite time series anomaly detection based on fast-DTW and improved-KNN. *Chinese Journal of Aeronautics*, *36*(2), 149–159.

Li, X. (2022). Vicinagearth security. *Communications of the CCF*, *18*, 44–52.

Li, S., Jin, X., Xuan, Y., Zhou, X., Chen, W., Wang, Y.-X., & Yan, X. (2019). Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. In *Advances in neural information processing systems*: *vol. 32*.

Liu, M., Ren, S., Ma, S., Jiao, J., Chen, Y., Wang, Z., & Song, W. (2021). Gated transformer networks for multivariate time series classification. arXiv preprint arXiv:2103.14438.

Lohit, S., Wang, Q., & Turaga, P. (2019). Temporal transformer networks: Joint learning of invariant and discriminative time warping. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 12426–12435).

Ma, Q., Chen, Z., Tian, S., & Ng, W. W. (2020). Difference-guided representation learning network for multivariate time-series classification. *IEEE Transactions on Cybernetics*, *52*(6), 4717–4727.

Middlehurst, M., Large, J., Flynn, M., Lines, J., Bostrom, A., & Bagnall, A. (2021). HIVE-COTE 2.0: A new meta ensemble for time series classification. *Machine Learning*, *110*(11–12), 3211–3243.

Mohamed, A., Okhonko, D., & Zettlemoyer, L. (2019). Transformers with convolutional context for ASR. arXiv preprint arXiv:1904.11660.

Oh, J., Wang, J., & Wiens, J. (2018). Learning to exploit invariances in clinical time-series data using sequence transformer networks. In *Machine learning for healthcare conference* (pp. 332–347). PMLR.

O'Reilly, C., Moessner, K., & Nati, M. (2017). Univariate and multivariate time series manifold learning. *Knowledge-Based Systems*, *133*, 1–16.

Pal, A., Karkhanis, D., Roberts, M., Dooley, S., Sundararajan, A., & Naidu, S. (2023). Giraffe: Adventures in expanding context lengths in llms. arXiv preprint arXiv:2308.10882.

Pei, W., Dibeklioğlu, H., Tax, D. M., & van der Maaten, L. (2017). Multivariate time-series classification using the hidden-unit logistic model. *IEEE Transactions on Neural Networks and Learning Systems*, *29*(4), 920–931.

Phyo, J., Ko, W., Jeon, E., & Suk, H.-I. (2022). Transsleep: Transitioning-aware attention-based deep neural network for sleep staging. *IEEE Transactions on Cybernetics*.

Połap, D., Srivastava, G., & Jaszcz, A. (2023). Energy consumption prediction model for smart homes via decentralized federated learning with LSTM. *IEEE Transactions on Consumer Electronics*.

Protic, D., & Stankovic, M. (2023). XOR-based detector of different decisions on anomalies in the computer network traffic. *Science and Technology*, *26*(3–4), 323–338.

Ruß wurm, M., & Körner, M. (2020). Self-attention for raw optical satellite time series classification. *ISPRS Journal of Photogrammetry and Remote Sensing*, *169*, 421–435.

Savi, M., & Olivadese, F. (2021). Short-term energy consumption forecasting at the edge: A federated learning approach. *IEEE Access*, *9*, 95949–95969.

Tan, C. W., Dempster, A., Bergmeir, C., & Webb, G. I. (2022). MultiRocket: Multiple pooling operators and transformations for fast and effective time series classification. *Data Mining and Knowledge Discovery*, *36*(5), 1623–1646.

Tang, W., Long, G., Liu, L., Zhou, T., Blumenstein, M., & Jiang, J. (2020). Omni-scale CNNs: A simple and effective kernel size configuration for time series classification. arXiv preprint arXiv:2002.10061.

Tian, S., Zhang, Y., Feng, Y., Elsagan, N., Ko, Y., Mozaffari, M. H., Xi, D. D., & Lee, C.-G. (2023). Time series classification, augmentation and artificial-intelligence-enabled software for emergency response in freight transportation fires. *Expert Systems with Applications*, *233*, Article 120914.

Tonekaboni, S., Eytan, D., & Goldenberg, A. (2021). Unsupervised representation learning for time series with temporal neighborhood coding. arXiv preprint arXiv:2106.00750.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems*: *vol. 30*.

Vos, K., Peng, Z., Jenkins, C., Shahriar, M. R., Borghesani, P., & Wang, W. (2022). Vibration-based anomaly detection using LSTM/SVM approaches. *Mechanical Systems and Signal Processing*, *169*, Article 108752.

Wang, Z., Yan, W., & Oates, T. (2017). Time series classification from scratch with deep neural networks: A strong baseline. In *2017 international joint conference on neural networks* (pp. 1578–1585). IEEE.

Wang, X., Zhang, Y., Bai, N., Yu, Q., & Wang, Q. (2024). Class-imbalanced time series anomaly detection method based on cost-sensitive hybrid network. *Expert Systems with Applications*, *238*, Article 122192.

Wen, Q., Zhou, T., Zhang, C., Chen, W., Ma, Z., Yan, J., & Sun, L. (2022). Transformers in time series: A survey. arXiv preprint arXiv:2202.07125.

Wu, S., Liang, M., Wang, X., & Chen, Q. (2023). Vgbel: An exploration of ensemble learning incorporating non-euclidean structural representation for time series classification. *Expert Systems with Applications*, *224*, Article 119942.

Xiao, Z., Xing, H., Qu, R., Feng, L., Luo, S., Dai, P., Zhao, B., & Dai, Y. (2024). Densely knowledge-aware network for multivariate time series classification. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*.

Yan, S., Fang, H., Li, J., Ward, T., O'Connor, N., & Liu, M. (2023). Privacy-aware energy consumption modeling of connected battery electric vehicles using federated learning. *IEEE Transactions on Transportation Electrification*.

Yu, J., Gao, X., Zhai, F., Li, B., Xue, B., Fu, S., Chen, L., & Meng, Z. (2023). An adversarial contrastive autoencoder for robust multivariate time series anomaly detection. *Expert Systems with Applications*, Article 123010.

Yue, Z., Wang, Y., Duan, J., Yang, T., Huang, C., Tong, Y., & Xu, B. (2022). TS2vec: Towards universal representation of time series. In *Proceedings of the AAAI conference on artificial intelligence*: *vol. 36*, (pp. 8980–8987).

Zeng, Z., Zhao, W., Qian, P., Zhou, Y., Zhao, Z., Chen, C., & Guan, C. (2021). Robust traffic prediction from spatial–temporal data based on conditional distribution learning. *IEEE Transactions on Cybernetics*, *52*(12), 13458–13471.

Zerveas, G., Jayaraman, S., Patel, D., Bhamidipaty, A., & Eickhoff, C. (2021). A transformer-based framework for multivariate time series representation learning. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining* (pp. 2114–2124).

Zhang, D., Gao, J., & Li, X. (2024). Learning long-range relationships for temporal aircraft anomaly detection. *IEEE Transactions on Aerospace and Electronic Systems*.

Zhang, X., Gao, Y., Lin, J., & Lu, C.-T. (2020). Tapnet: Multivariate time series classification with attentional prototypical network. In *Proceedings of the AAAI conference on artificial intelligence*: *vol. 34*, (pp. 6845–6852).

Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., & Zhang, W. (2021). Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*: *vol. 35*, (pp. 11106–11115).