GRU-CP: Non-stationary Time Series Forecasting with Change Point Detection

Zhihua Ding, Keqin Shi and Weiqiang Sun
Department of Electronic Engineering
Shanghai Jiao Tong University
Shanghai, China
{zhihuading, keqinshi, sunwq}@sjtu.edu.cn

***Abstract*—Time series forecasting finds numerous applications in real-world scenarios such as finance, healthcare, and weather prediction. However, a significant portion of real-world time series data exhibits non-stationarity. While some methods like Prophet and STL can decompose non-stationary time series due to seasonality and periodicity for accurate predictions, they may falter when faced with unforeseeable factors, such as structural breaks induced by government interventions in exchange rates. This study asserts that non-stationary time series can be segmented into multiple stationary segments via change points, and these segments may possess latent temporal characteristics. By employing change point detection algorithms to identify non-stationary features, these features can be incorporated into prediction models. The proposed GRU-CP model is introduced in this context. Compared to traditional models, GRU-CP yields lower errors when forecasting time series data.**

***Index Terms*—Time Series Forecasting; Change Point Detection; Non-stationarity; Gated Recurrent Unit**

I. Introduction

Time series forecasting (TSF) finds increasing applications across industries, including finance [1], weather [2], energy consumption [3], healthcare [4] and so on. In finance, TSF is used to predict stock prices and exchange rates, and in sales to forecast product demand and optimize supply chains [1]. Weather uses TSF to predict weather patterns [2], while in energy consumption, it assists in efficient resource planning and load balancing [3]. Additionally, TSF is applied to predict healthcare, it aids in predicting patient admissions and disease outbreaks for resource planning [4]. As data availability and machine learning advancements grow, the accuracy and versatility of TSF continue to improve, benefiting diverse sectors with informed insights for planning and resource allocation.

TSF methods can be broadly classified into two categories: statistical methods and deep learning methods. Statistical methods include Autoregressive Integrated Moving Average (ARIMA) [5] for stationary and seasonally-driven data, Exponential Smoothing (ES) [6] for capturing trend and seasonality patterns, and Seasonal Decomposition of Time Series (STL) [7] for component-wise analysis. Additionally, Prophet [8] is a robust additive model suited for business applications.

What's more, deep learning methods such as Long Short-Term Memory (LSTM) [9] and Gated Recurrent Unit (GRU) [10] are effective in handling complex temporal dependencies and long-range interactions in sequential data. The success of Transformer [11] in the domain of natural language processing has led to its widespread adoption in TSF, owing to its versatile applications and promising performance in handling sequential data.

In the natural world, time series data often tends to be non-stationary rather than stationary. Many real-world time series, including weather patterns, economic indicators, and biological processes, exhibit trends, seasonality, and other non-stationary patterns [7]. Analyzing and forecasting such time series data often requires special methods that can handle non-stationarity and capture the underlying temporal dependencies effectively.

In the domain of TSF, the prevailing methods have distinct requirements regarding the stationarity of the data. ARIMA and ES methods necessitate the time series to be stationary [5], [6]. Furthermore, STL and Prophet rely on decomposition-based techniques and are suitable for predicting non-stationary sequences arising from diurnal, seasonal, or other periodic patterns [7], [8].

While some deep learning models like LSTM and GRU have shown promising results in TSF [12], they still have room for optimization in accurately capturing the underlying patterns, in cases where the non-stationarity is severe or when dealing with complex non-linear trends. In situations where non-stationarity is pronounced, preprocessing techniques such as differencing or detrending may be imperative to enhance the model's performance and stability [13]. Additionally, augmenting the model with exogenous variables or features can further reinforce its capacity to handle non-stationary data and capture relevant patterns for accurate forecasting [14].

In scenarios where data non-stationarity arises from unpredictable influences, such as interest rate fluctuations resulting from government monetary policies [15], augmenting the predictive model with identifiable features capturing the dynamics around change points becomes crucial for enhancing the forecasting performance. However, the majority of sequences in the natural world are non-stationary. One approach to enhance prediction accuracy is to incorporate non-stationary features into the prediction model. For instance, Kyong Jo Oh et. [15] and Ayla Jungbluth et al. [16] first segment the sequence based on its stationarity and subsequently train and predict using the segmented data, resulting in improved predictive performance. However, these methods that leverage

2833-8898/23/$31.00 ©2023 IEEE
DOI 10.1109/CyberC58899.2023.00012

sequence stationarity features primarily operate during the data preprocessing stage, which may limit the model's ability to capture global sequence characteristics, thereby potentially affecting predictive performance.

This paper focuses on enhancing the accuracy of TSF by introducing time change point into neural networks, using GRU as the representative model. The key contributions of this research are as follows:

- We have manually crafted a dataset with distinct non-stationary features to explore the predicting behavior and performance of various forecasting techniques for different types of non-stationary time series.
- We performed change point detection on time series data from real-world datasets and subsequently integrated the change point with GRU to propose a novel model named Gated Recurrent Unit with Change Point Detection (GRU-CP).
- We conducted experiments on multiple datasets using GRU-CP and compared its performance against various high-performing TSF models. The prediction results demonstrate that the proposed model outperforms other models that do not consider time series change point, achieving state-of-the-art forecasting accuracy.

The rest of the paper is organized as follows. Section 2 provides an overview of the relevant background and reviews related work.The architecture of GRU-CP is comprehensively described in Section 3. In Section 4, the experimental datasets, methodologies, and resulting data are presented. Section 5 concludes the study, offering a summary of the findings and a discussion on future research directions.

II. Related Work

In this section, a comprehensive review of pertinent literature will be undertaken, encompassing non-stationary sequences, change point detection algorithms, TSF models, and predictive models incorporating change point detection. This exposition serves to underscore the novelty and necessity of the research endeavors.

A. Non-stationary Data

Non-stationary time series are prevalent in the natural world. For instance, in the foreign exchange market, exchange rates are influenced by economic and political events, leading to unstable fluctuations in exchange rate time series [15]. Similarly, traffic flow is often affected by factors such as weekdays, weekends, and holidays, displaying evident seasonality and periodicity within different time periods. Additionally, temperature time series exhibit non-stationary patterns due to variations in seasons, geographic locations, and climate conditions.

Common variations in time series include changes in mean, variance, and periodicity. Fig.1 illustrates the time series variations caused by changes in mean, variance, and periodicity, all based on a simple sinusoidal curve as the underlying pattern. These changes in the time series occur after time $t = 1$, affecting the mean, variance, and periodicity, respectively. More complex cases may involve considering changes in the correlations of multivariate variables [17]. The non-stationarity in time series can be attributed to rhythmic or periodic fluctuations, which exhibit regular patterns within specific time intervals, as well as the presence of noise leading to stochastic behavior. Additionally, external factors or significant events can cause abrupt structural breaks, further contributing to the non-stationary nature of the series. These variations in time series characteristics over time pose challenges to conventional forecasting methods in effectively addressing the inherent complexities.

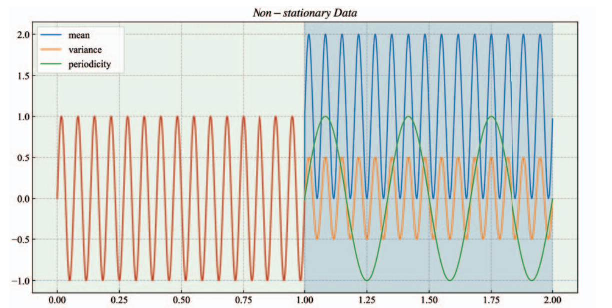

Fig. 1. Time series variations in "mean", "variance" and "periodicity".

B. Change Point Detection

Non-stationary time series can be viewed as a concatenation of multiple stationary segments, and change point detection algorithms can be employed to partition the non-stationary sequence into several stationary time series. Taking household electricity consumption series [18] as an example, its time series segmentation is illustrated in Fig.2, dividing the non-stationary data sequence of 150 days into five stationary segments.

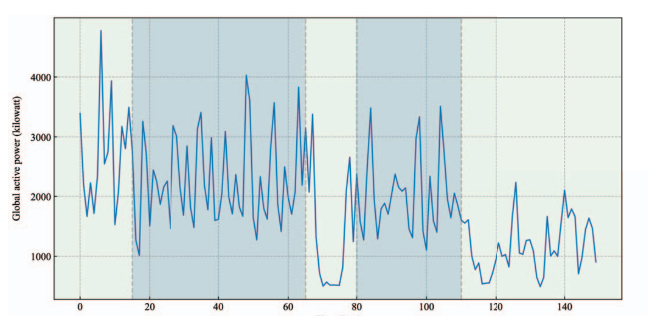

Fig. 2. Segmenting non-stationary data using Change Point Detection algorithms.

Time series decomposition algorithms typically consist of three elements: cost function, search method and constraint [19]. The cost function is used to measure the homogeneity of the sub-sequences obtained after segmentation. C_{L_2} is commonly employed for detecting time change points related to mean shifts and has found application in numerous studies. It also has been successfully utilized in various domains, such as DNA array data and geological data analysis. In financial literature, segmented linear regression is frequently employed for time series detection [20], where the cost function C_{Linear} is used to model the segmented linear regression.

The search method is employed to minimize the cost function. There are two categories of search methods: Optimal detection and Approximate detection. Optimal detection algorithms utilize optimization techniques to explore all points within the time series, seeking the segmentation point that minimizes the cost function, such as *Opt* and *Pelt* algorithms. While these algorithms offer high accuracy, they come with a high computational complexity, with *Opt* having a worst-case complexity of $O(KT^2)$. Optimal detection methods have been applied to DNA sequences, physiological signals, and oceanographic data. In addition, Approximate detection algorithms are based on sequential search and include *window-based* methods, *binary segmentation* [21], and *bottom-up segmentation*. These algorithms have lower computational complexity and find applications in analyzing biological signals, network data, and financial time series, among others.

In cases where the number of change points is unknown, constraints are added to constrain the optimization process. By applying various constraints, such as window constraints, minimum and maximum segment length constraints, and order constraints, these algorithms can efficiently explore the time series data, narrowing down the search to relevant regions and minimizing unnecessary computations. The incorporation of constraints aids in identifying potential change points that adhere to specific criteria, resulting in precise and meaningful change point detection across diverse applications.

C. TSF

Traditional TSF methods, including ARIMA [5], ES [6], and STL [7], rely on statistical approaches to compute the statistical characteristics of time series and subsequently make predictions. These methods are computationally efficient but may exhibit limitations in terms of prediction accuracy.

The Prophet model, introduced by Taylor et al., integrates essential features such as seasonality, trend, and holidays in time series data and possesses the capability to automatically detect and handle outliers [8] . It employs an additive model to fit time series, with trend and seasonal components modeled using segmented linear regression and seasonal decomposition techniques. Prophet also accounts for the impact of holidays on time series, showcasing adaptability to non-stationary data. However, in the case of complex time series data, Prophet may demonstrate inferior performance compared to more intricate time series models.

Recurrent Neural Networks (RNNs) demonstrate significant superiority in capturing temporal dependencies, with specialized variants such as LSTM and GRU effectively addressing the challenges related to vanishing and exploding gradients that are characteristic of conventional RNNs. Currently, RNNs have garnered substantial achievements in the field of sequence forecasting. For instance, Zha et al. achieved accurate gas field production predictions by leveraging a combination of LSTM and Convolutional Neural Network (CNN) [22]. Similarly, Yang et al. employed GRU to predict motion in beating heart surgery [23].

The Transformer [11] architecture has witnessed significant advancements in the domain of TSF, particularly in the context of long-term time series prediction. Wu et al. [24] introduced a novel decomposition framework called Autoformer, featuring a self-correlation mechanism that enhances its progressive decomposition capabilities for complex time series. Zhou et al. [25] proposed the Informer model, which reduces the temporal and memory complexity of Transformers while concurrently enhancing performance of predictions. Moreover, Liu et al. [26] leveraged multiple time resolutions in a low-complexity variant model known as Pyraformer. These developments exemplify the innovative applications of the Transformer structure in addressing challenges in time series prediction.

As TSF models become increasingly sophisticated, Zeng et al. [27] contend that within the Transformer architecture, the inherent permutation-invariant self-attention mechanism may result in the loss of positional information within sequences. They propose the DLinear and NLinear models to address this concern and conduct comprehensive testing across multiple datasets. The implications of their findings suggest that the intricate structure of the Transformer model might not be inherently suitable for TSF, as evidenced by the consistent superior performance exhibited by the proposed DLinear and NLinear models.

D. Non-stationary TSF

In the TSF literature, there exists a limited presence of techniques that explicitly incorporate time series change point into predictive models. Only a handful of studies have taken time series change point into consideration, primarily during data pre-processing phase, particularly in the context of handling non-stationary data. For instance, Kyong Jo Oh et al. [15] addressed financial domain data by first conducting change point detection on time series exhibiting “structural break”, followed by the aggregation of temporally similar segments with shared characteristics for subsequent individual predictions. Ayla Jungbluth et al. [16] introduced the DeepCAR model, which strategically eliminates time batches containing change points during the data collection phase, thereby circumventing the influence of change points.

However, the aforementioned approaches exhibit two inherent limitations. Firstly, the failure to incorporate time series change point directly into the modeling process results in the loss of pertinent sequence information. Secondly, the practice of discarding batches containing time series change points can render ineffective model training for datasets with limited available data volume, thereby impeding the robustness of the model. The proposed GRU-CP model, which integrates the GRU architecture with change point detection, addresses the limitations encountered in prior studies. This model effectively encompasses the influence of temporal change point characteristics on prediction outcomes while simultaneously retaining essential training data. Consequently, it has demonstrated remarkable efficacy in non-stationary TSF.

III. GRU-CP: Non-stationary TSF with Change Point Detection

This section commences by establishing the prescribed format for the research questions to be addressed. Subsequently, it delves into the exposition of algorithms employed for conducting change point detection on time series data, followed by a comprehensive elucidation of the intricate structure underpinning GRU-CP.

A. Mathematical Description

For a given time series, we denote the historical data as $X = \{x_1, x_2 \dots x_T\}$ with a length of T, and the objective is to predict data for the upcoming τ time points $\widetilde{X} = \{x_{T+1}, x_{T+2} \dots x_{T+\tau}\}$. Within sequence X, there exist K change points, denoted as $k_1, k_2 \dots k_K$.

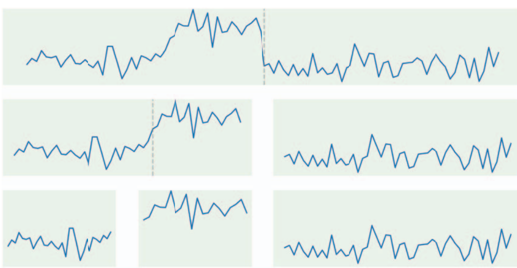

Fig. 3. Schematic example of Binary Segmentation.

B. Change Point Detection

The training of TSF models typically involves a substantial volume of data. To mitigate computational complexity, we opt for the *Binary Segmentation (BinSeg)* [21] detection method, employing C_{L_2} as the cost function for conducting change point detection on the time series. It is noteworthy that there exists a multitude of algorithms for performing change point detection on time series data. The selection of an appropriate detection algorithm should align with the specific characteristics of the temporal sequence variations. Importantly, this choice may not necessarily yield the optimal outcomes in terms of performance.

Binseg iteratively bifurcates the sequence, thereby yielding change points subject to constraints. Fig.3 depicts the iterative process of change point detection on a time series using the Binseg algorithm, and the comprehensive algorithm is presented in the form of pseudocode in Algorithm 1. The first detected change point $\hat{k}_1$ satisfies:

$$\hat{k}_1 = \arg \min_{1 \le t < T} c(x_{0 \dots t}) + c(x_{t \dots T}), \tag{1}$$

where $c(\cdot)$ corresponds to Eq.(2).

$$C_{L_2}(y_{a \dots b}) = \sum_{t=a+1}^{b} ||y_t - \overline{y}_{a \dots b}||_2^2, \tag{2}$$

where $y_{a \dots b}$ represents the sub-sequence, and $\overline{y}_{a \dots b}$ is the mean of $y_{a \dots b}$.

C. Model Structure

GRU-CP is an innovative extension of the GRU architecture, incorporating time series change points. This model retains the strengths of GRU in capturing temporal dependencies while integrating temporal non-stationary features, rendering it more adaptive to non-stationary data. GRU-CP introduces supplementary components, namely m, γ_x, and γ_h, building upon the foundation of GRU.

The m is employed to identify the occurrence of temporal sequence changes. For a time series $X_{n \times T}$ of length T with n variables, the corresponding mask m possesses the same dimensions as $X_{n \times T}$ which is obtained through the time series change point detection algorithm. In cases where change points are present in the sequence, the elements of the mask matrix after the change point locations are assigned a value of 1, while those before the change points remain 0. Conversely, if no change points are present, all elements of the mask matrix are set to 1. The γ_x and γ_h matrices are contingent on the distance, d, between the elements x_i^j and the last element x_i^T in $X_{n \times T}$. Following the detection of change points in n-dimensional sequence $X_{n \times T}$, the manifestation of the resulting m and d can be observed in Fig.4.

Algorithm 1 Algorithm *BinSeg*

Input:
α Threshold for the next segmentation
$x_{0 \dots T}$ Sequence for change point detection
$c(\cdot)$ Cost function
Output:
Change points set: $\{k_0, k_1, \dots, k_K\}$

Initialize:
  $list\_for\_BinSeg \leftarrow [x_{0,\dots,T}]$,
  $list\_of\_Change\_Point \leftarrow [0]$,
  $flag \leftarrow$ True
while $flag$ is True **do**
  $\_list\_for\_BinSeg \leftarrow [\,]$
  $list\_of\_Cost \leftarrow [\,]$
  for $x_{i \dots j}$ in $list\_for\_BinSeg$ **do**
    $list\_of\_Cost$.append($[c(x_{i \dots j})]$)
    if $c(x_{i \dots j}) > \alpha$ **then**
      $k \leftarrow \arg\min_{i \le t < j} c(x_{i \dots t}) + c(x_{t \dots j})$
      $list\_of\_Change\_Point$.append($[k]$)
      $\_list\_for\_BinSeg$.append($[x_{i \dots k}, x_{k \dots j}]$)
    end if
  end for
  $list\_for\_BinSeg \leftarrow \_list\_for\_BinSeg$
  if $\forall z \in list\_of\_Cost$,
    subject to $z \le \alpha$
   then $flag \leftarrow$ False
  end if
end while

Output: $list\_of\_Change\_Point$

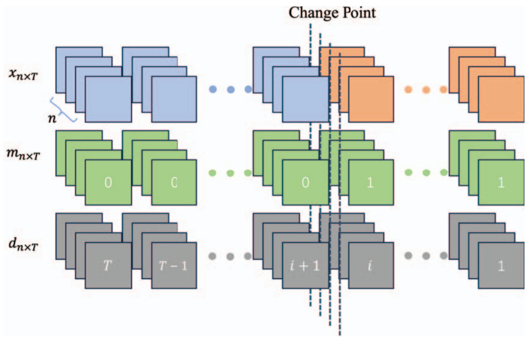

Fig. 4. Matrices m and d for a time series $X_{n\times T}$. $X_{n\times T}$ in different colors indicates that it is a non-stationary sequence consisting of two segments of stationary sequences.

The matrices γ_x and γ_h are derived from d, and their relationship is as follows:

$$\gamma_x = exp\{-max(0, W_x d + b_x)\}, \tag{3}$$
$$\gamma_h = exp\{-max(0, W_h d + b_h)\}. \tag{4}$$

GRU-CP introduces modifications in the input x, and the output h, through the incorporation of m, γ_x, and γ_h.

$$x \leftarrow m * x + (1 - m) * \gamma_x * x, \tag{5}$$
$$h_t \leftarrow m * h_t + (1 - m) * \gamma_{ht} * h_t. \tag{6}$$

The updating of the internal units within GRU-CP follows the equation provided below.

$$z_t = \sigma(W_z[x_t, h_{t-1}]), \tag{7}$$
$$r_t = \sigma(W_r[x_t, h_{t-1}]), \tag{8}$$
$$\widetilde{h}_t = tanh(W[r_t * h_{t-1}, x_t]), \tag{9}$$
$$h_t = (1 - z_t) * h_{t-1} + z_t * \widetilde{h}_t. \tag{10}$$

When there are no time series change points present in the sequence, GRU-CP is equivalent to GRU. Fig.5 provides a comparative illustration of the distinctions between GRU and GRU-CP.

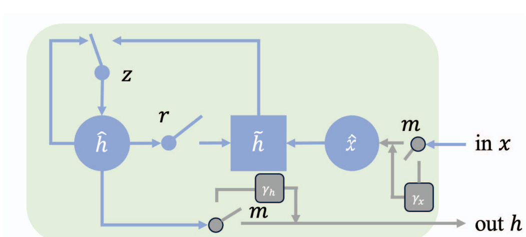

Fig. 5. Structure of GRU-CP, which introduces additional components compared to the standard GRU model, denoted by the shaded regions.

Zeng et al. argue that iterated multi-step (IMS) forecasting could lead to error accumulation, which is disadvantageous for long-term time series prediction [27]. Consequently, we opt for direct multistep (DMS) forecasting for data prediction, represented as:

$$\widetilde{X} = W_p h_T + b_p, \tag{11}$$

where $\widetilde{X}$ represents the predicted sequence. The complete architecture of GRU-CP network is depicted in Fig.6.

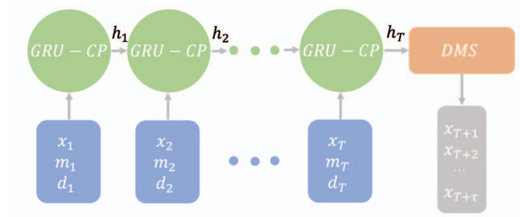

Fig. 6. Direct Multi-step Prediction with GRU-CP.

IV. Experiments

In this section, we introduce the utilized datasets. We then introduce the optimizer employed for model training, the loss function and the performance metric used in our evaluation. The experimental results of GRU-CP are then presented and discussed. As established by Zeng et al. [27], it has been demonstrated that the Transformer model does not perform well in the context of TSF. Therefore, we opt to conduct comparative experiments involving GRU [10], DLinear [27], and proposed GRU-CP.

A. Experimental Settings

1) DataSet: In this experiment, we employed a manually crafted dataset and three publicly available real-world datasets. Their characteristics are as follows:

- Manually Crafted Dataset: The manually crafted dataset encompasses a total sequence length of 5×10^5 data points, showcasing three distinct types of temporal sequence variations: altering mean, varying variance, and changing periodicity. The fundamental underlying pattern of this dataset is sinusoidal, with the three distinct change patterns depicted in Fig.1.
- Weather Dataset: The weather dataset documents data at 10-minute intervals throughout the year 2020 in Germany, comprising 21 weather indicators, including air temperature and humidity.
- Solar Energy Dataset: Contained within are solar power production records from the year 2006, sampled at 5-minute intervals across 137 photovoltaic (PV) plants situated in the state of Alabama.
- Exchange-Rate Dataset: Exchange-Rate [28] collects the daily exchange rates of 8 countries from 1990 to 2016.

2) Training Optimizer: For model training, the Adam [29] optimizer is chosen, and the loss function selected is Mean Squared Error (MSE).

3) Evaluation metric: Given that TSF is a type of regression problem, opting for Mean Squared Error (MSE) and Mean Absolute Error (MAE) as the evaluation metrics is most appropriate. Both MSE and MAE exhibit an inverse relationship with model performance, where smaller values indicate superior performance.

B. Experimental Results

1) Manually Crafted Dataset: In order to assess the adaptability and predictability of different methods concerning time series with various types of change points, experiments were conducted on a range of manually crafted datasets featuring diverse change point patterns. To ensure that the models could capture comprehensive sequential patterns, a historical window length of $\{100\}$ was set, and prediction window lengths were set at $\{3, 6, 12, 24\}$, respectively. The obtained results are summarized in Table I.

2) Short-term Forecast: In order to compare the performance of the models on real-world datasets, we conducted experiments on the *weather*, *solar*, and *exchange* datasets for TSF. Considering the varying time granularities of the three datasets, we set the historical lookback window sizes as 144, 288, and 96 for the *weather*, *solar*, and *exchange* datasets respectively. We predicted future time lengths of $\{3, 6, 12, 24\}$ units. The results are presented in Table II.

3) Long-term Forecast: Zhou et al. have highlighted the substantial demand for long-term TSF in real-world applications, underscoring the significance of a model's capacity for extended prediction. Therefore, we also investigate the long-term prediction capability of GRU-CP. In this context, we maintain the same historical lookback window size as used in short-term forecasting experiments, while extending the prediction horizon to $\{96, 192, 336, 720\}$ time units. The corresponding results are presented in Table III.

C. Result Analysis

The three datasets listed in Table I exhibits variations due to shifts in mean, changes in variance, and periodic fluctuations. The data presented in the table demonstrates that, for time series containing change points, the performance of GRU-CP, which incorporates change point features, significantly outperforms models that do not consider these features. Fig.7 illustrates the predictive adaptability of the three models to sequences containing different types of change points. It is evident that the recurrent neural network based on GRU surpasses linear models in TSF, particularly in scenarios involving mean shifts and periodic fluctuations. Both Table I and Fig.7 collectively indicate that linear models struggle to effectively capture changes in underlying periodicity when present in time series. This advantage can be attributed to the increased complexity of GRU and GRU-CP, characterized by a higher number of parameters and the inclusion of nonlinear units, rendering them more intricate than linear models.

Fig.8 illustrates the predictions of the three models on real-world datasets. From the graph, it is evident that the performance of GRU-CP surpasses that of the other two models across all three datasets, particularly notable in the case of the "solar" dataset. The "solar" dataset exhibits distinct variations in its sequence, which renders GRU-CP capable of achieving enhanced predictive accuracy. Conversely, the other two models lacking change point features are unable to accurately capture the structural shifts, or "structure break" in the sequence.

To further validate the superior predictive performance of GRU-CP on sequences with variations, we selected three batches of data from the "solar" dataset for observation. These three batches of data were excluded from the model training process and correspond to the time periods of *2006-10-16 00:00* to *2006-10-17 07:55*, *2006-10-16 08:00* to *2006-10-17 15:55*, and *2006-10-16 16:00* to *2006-10-17 23:55*, with data collected every five minutes to capture solar power data (MW). The predictive results of these three data segments are shown in Fig.9. The observed pattern indicates that GRU-CP is adept at swiftly capturing the underlying variations in the data, thereby achieving superior predictive performance. This further substantiates the adaptive capabilities of GRU-CP on non-stationary sequences.

Tables II and III present an intriguing observation: despite DLinear outperforming Transformer in TSF, GRU exhibits better performance than DLinear when utilizing direct multistep (DMS) forecasting. This outcome further reinforces the research of Zeng et al. [27], suggesting that the Transformer architecture is not well-suited for TSF tasks. Specifically, in the context of DMS forecasting, the effectiveness of a basic recurrent neural network surpasses that of a simple linear model. This prompts us to redirect the focus of TSF models towards more concise architectural designs.

The results in Tables II and III illustrate the enhanced performance of models that incorporate time series change point features when applied to real-world publicly available datasets. As demonstrated in Fig.5, it becomes evident that if change point detection is not applied to the data, GRU-CP is equivalent to GRU, thereby endowing GRU-CP with broader applicability. The comparative analysis of the performance between GRU-CP and GRU in this context underscores the benefit of incorporating change point features on TSF.

V. Conclusions

In this study, we initiated by constructing an artificial time series with distinct types of change points and applied various TSF models for prediction. This endeavor revealed that different models exhibited varying degrees of adaptability to different types of change points, with linear models demonstrating poor performance when faced with time series characterized by changing periodic patterns. This discovery can offer guidance for selecting suitable models for different types of TSF scenarios.

Furthermore, we devised GRU-CP by integrating change point features into GRU. Through validation, we show that GRU-CP possessed superior predictive capabilities for non-stationary time series, while it yielded equivalent performance to GRU when dealing with stationary time series. Additionally,

TABLE I
MANUALLY CRAFTED DATASET

Model	Metric	*mean*				*periodicity*				*variance*			
		3	*6*	*12*	*24*	*3*	*6*	*12*	*24*	*3*	*6*	*12*	*24*
GRU-CP	MSE	0.0195	**0.0336**	**0.0586**	0.1078	**0.0232**	**0.0448**	**0.0753**	**0.1300**	**0.0327**	**0.0479**	**0.0818**	0.1485
	MAE	0.0247	**0.0408**	**0.0729**	**0.1338**	**0.0415**	**0.0615**	**0.1048**	0.1677	0.0404	**0.0520**	**0.0892**	**0.1436**
GRU	MSE	**0.0194**	0.0354	0.0595	**0.1061**	0.0237	0.0531	0.0766	0.1419	0.0328	0.0483	0.0832	**0.1461**
	MAE	**0.0265**	0.0560	0.0766	0.1354	0.0436	0.0834	0.1087	**0.1664**	**0.0381**	0.0524	0.0897	0.1539
DLinear	MSE	0.0268	0.0416	0.0654	0.1165	0.0893	0.1642	0.3517	0.5590	0.0399	0.0548	0.0865	0.1494
	MAE	0.0537	0.0674	0.0928	0.1623	0.1722	0.2614	0.4163	0.5793	0.0626	0.0761	0.1014	0.1638

TABLE II
SHORT-TERM FORECAST

Model	Metric	*weather*				*solar*				*exchange*			
		3	*6*	*12*	*24*	*3*	*6*	*12*	*24*	*3*	*6*	*12*	*24*
GRU-CP	MSE	0.0002	**0.0004**	**0.0011**	0.0046	**0.0200**	0.0322	**0.0470**	0.1000	0.0118	0.0168	**0.0276**	**0.0452**
	MAE	0.0091	**0.0134**	**0.0220**	0.0490	**0.0558**	**0.0825**	0.1026	0.1670	0.0704	0.0834	**0.1163**	**0.1546**
GRU	MSE	0.0002	0.0006	0.0017	0.0085	0.0205	**0.0318**	0.0471	**0.0803**	0.0121	**0.0166**	0.1182	0.0464
	MAE	0.0090	0.0167	0.0296	0.0687	0.0573	0.0843	**0.0997**	**0.1366**	0.0685	**0.0832**	0.2809	0.1541
DLinear	MSE	0.0002	0.0010	0.0038	**0.0032**	0.0228	0.0364	0.0595	0.1009	**0.0113**	0.0167	0.0278	0.0461
	MAE	**0.0089**	0.0020	0.0469	**0.0385**	0.0733	0.1066	0.1400	0.1964	**0.0677**	0.0866	0.1187	0.1573

TABLE III
LONG-TERM FORECAST

Model	Metric	*weather*				*solar*				*exchange*			
		96	*192*	*336*	*720*	*96*	*192*	*336*	*720*	*96*	*192*	*336*	*720*
GRU-CP	MSE	**0.0323**	**0.0770**	0.1228	**0.1719**	**0.1980**	**0.2044**	**0.2191**	0.2579	0.1736	**0.2714**	**0.5076**	**0.3475**
	MAE	**0.1280**	**0.2069**	0.2677	**0.3282**	**0.2420**	**0.2599**	0.2706	0.2978	0.3291	**0.4144**	**0.5668**	**0.4709**
GRU	MSE	0.0454	0.0808	**0.1172**	0.2591	0.2117	0.2147	0.2197	**0.2328**	**0.1512**	0.2840	0.5179	0.4671
	MAE	0.1540	0.2142	**0.2572**	0.4020	0.2600	0.2722	**0.2628**	**0.2755**	**0.2976**	0.4253	0.5735	0.5543
DLinear	MSE	0.0639	0.0831	0.1338	0.1958	0.2110	0.2398	0.2574	0.3103	0.1789	0.3102	0.5354	0.4671
	MAE	0.1921	0.2104	0.2711	0.3524	0.2910	0.3142	0.3158	0.3548	0.3230	0.4432	0.5832	0.5450

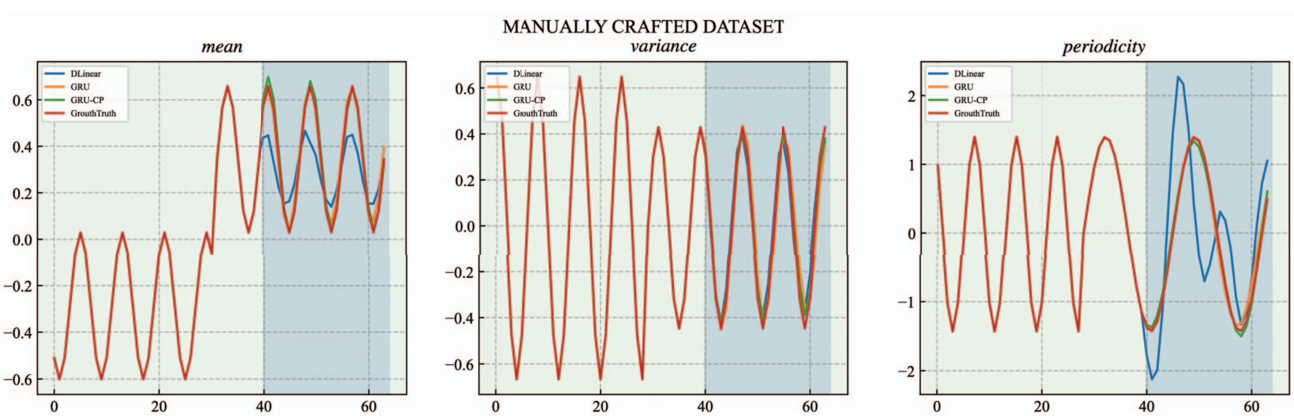

Fig. 7. Predictions of three models with an output length $\tau = 24$ on non-stationary sequences caused by “mean”, “variance”, and “periodicity”.

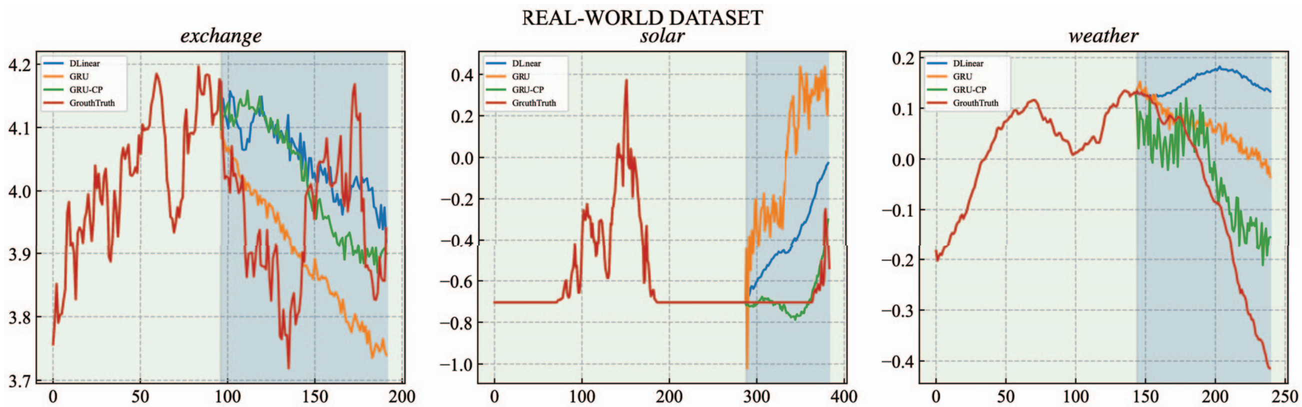

Fig. 8. Illustrations depicting predictions of three models with an output length $\tau = 96$ on real-world dataset, labeled as "exchange", "solar", and "weather".

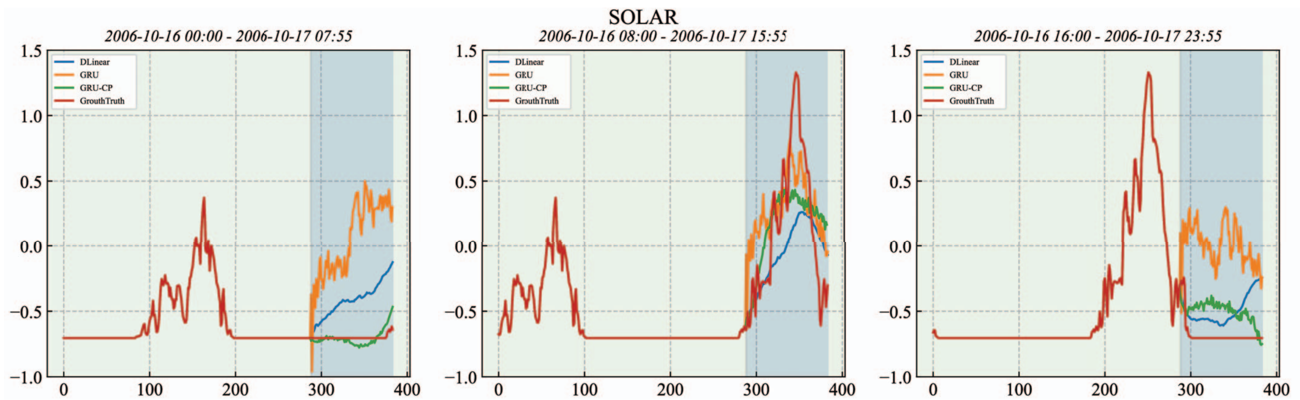

Fig. 9. Illustrations depicting predictions of three models on distinct three temporal segments.

the study reaffirmed the robust performance of simple recurrent neural networks in the field of TSF, prompting researchers to further prioritize streamlined model architectures for time series prediction.

There are certain aspects in this study that require further refinements. For instance, the performance of GRU-CP for non-stationary multivariate time series remains unexplored. In terms of performance comparison, the study lacks a comparison between GRU-CP and statistical methods such as Prophet. Furthermore, future research should delve into investigating the influence of different change point detection algorithms on time series prediction. These avenues for improvement suggest potential directions for extending the scope of this study.

Acknowledgment

This research is supported by the National Natural Science Foundation of China (NSFC) (62331017, 61901118, 62271306).

References

[1] Z. Zeng, R. Kaur, S. Siddagangappa, S. Rahimi, T. Balch, and M. Veloso, "Financial time series forecasting using cnn and transformer," 2023.

[2] S. Rasp, P. D. Dueben, S. Scher, J. A. Weyn, S. Mouatadid, and N. Thuerey, "WeatherBench: A benchmark data set for data-driven weather forecasting," *Journal of Advances in Modeling Earth Systems*, vol. 12, nov 2020.

[3] L. Saad Saoud, H. Al-Marzouqi, and R. Hussein, "Household energy consumption prediction using the stationary wavelet transform and transformers," *IEEE Access*, vol. 10, pp. 5171–5183, 2022.

[4] M. A. Morid, O. R. L. Sheng, and S. Abdelrahman, "Leveraging patient similarity and time series data in healthcare predictive models," 2017.

[5] S. Ho and M. Xie, "The use of arima models for reliability forecasting and analysis," *Computers Industrial Engineering*, vol. 35, no. 1, pp. 213–216, 1998.

[6] B. Billah, M. King, R. Snyder, and A. Koehler, "Exponential smoothing model selection for forecasting," *International Journal of Forecasting*, vol. 22, pp. 239–247, 02 2006.

[7] R. B. Cleveland, W. S. Cleveland, J. E. McRae, and I. Terpenning, "Stl: A seasonal-trend decomposition procedure based on loess (with discussion)," *Journal of Official Statistics*, vol. 6, pp. 3–73, 1990.

[8] S. J. Taylor and B. Letham, "Forecasting at scale," *PeerJ Preprints*, vol. 5, p. e3190v2, 2017.

[9] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, pp. 1735–1780, 11 1997.

[10] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," 2014.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2023.

[12] H. Hewamalage, C. Bergmeir, and K. Bandara, "Recurrent neural networks for time series forecasting: Current status and future directions," *International Journal of Forecasting*, vol. 37, pp. 388–427, jan 2021.

[13] A. Raudys and E. Goldstein, "Forecasting detrended volatility risk and financial price series using lstm neural networks and xgboost regressor," *Journal of Risk and Financial Management*, vol. 15, no. 12, 2022.

[14] T. Guo and T. Lin, "Multi-variable lstm neural network for autoregressive exogenous model," 2018.

[15] K. J. Oh and I. Han, "Using change-point detection to support artificial neural networks for interest rates forecasting," *Expert Systems with Applications*, vol. 19, no. 2, pp. 105–115, 2000.

[16] A. Jungbluth and J. Lederer, "The deepcar method: Forecasting time-series data that have change points," 2023.

[17] A. Jungbluth and J. Lederer, "The deepcar method: Forecasting time-series data that have change points," 2023.

[18] G. Hebrail and A. Berard, "Individual household electric power consumption." UCI Machine Learning Repository, 2012. DOI: https://doi.org/10.24432/C58K54.

[19] C. Truong, L. Oudre, and N. Vayatis, "Selective review of offline change point detection methods," *Signal Processing*, vol. 167, p. 107299, 2020.

[20] J. Bai, "Testing for parameter constancy in linear regressions: An empirical distribution function approach," *Econometrica*, vol. 64, no. 3, pp. 597–622, 1996.

[21] A. Sen and M. S. Srivastava, "On Tests for Detecting Change in Mean," *The Annals of Statistics*, vol. 3, no. 1, pp. 98 – 108, 1975.

[22] W. Zha, Y. Liu, Y. Wan, R. Luo, D. Li, S. Yang, and Y. Xu, "Forecasting monthly gas field production based on the cnn-lstm model," *Energy*, vol. 260, p. 124889, 2022.

[23] B. Yang, Y. Li, W. Zheng, Z. Yin, M. Liu, L. Yin, and C. Liu, "Motion prediction for beating heart surgery with gru," *Biomedical Signal Processing and Control*, vol. 83, p. 104641, 2023.

[24] H. Wu, J. Xu, J. Wang, and M. Long, "Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting," in *Advances in Neural Information Processing Systems* (M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, eds.), vol. 34, pp. 22419–22430, Curran Associates, Inc., 2021.

[25] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, "Informer: Beyond efficient transformer for long sequence time-series forecasting," 2021.

[26] S. Liu, H. Yu, C. Liao, J. Li, W. Lin, A. X. Liu, and S. Dustdar, "Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting," in *International Conference on Learning Representations*, 2022.

[27] A. Zeng, M. Chen, L. Zhang, and Q. Xu, "Are transformers effective for time series forecasting?," 2022.

[28] G. Lai, W.-C. Chang, Y. Yang, and H. Liu, "Modeling long- and short-term temporal patterns with deep neural networks," 2018.

[29] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017.