The Thirty-Eighth AAAI Conference on Artificial Intelligence (AAAI-24)

CGS-Mask: Making Time Series Predictions Intuitive for All

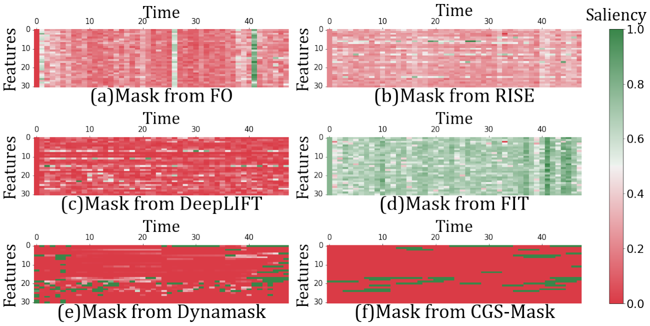
Feng Lu[*1], **Wei Li**[*†2], **Yifei Sun**[*1], **Cheng Song**[1], **Yufei Ren**[3], **Albert Y. Zomaya**[2]
[1]School of Computer Science and Technology, Huazhong University of Science and Technology, China
[2]The Australia-China Joint Research Centre for Energy Informatics and Demand Response Technologies, Centre for Distributed and High Performance Computing, School of Computer Science, The University of Sydney, Australia
[3]Tongji Hospital, Tongji Medical College, Huazhong University of Science and Technology, China
{lufeng,sunyifei,songch}@hust.edu.cn, {weiwilson.li,albert.zomaya}@sydney.edu.au, yfren@tjh.tjmu.edu.cn

Abstract

Artificial intelligence (AI) has immense potential in time series prediction, but most explainable tools have limited capabilities in providing a systematic understanding of important features over time. These tools typically rely on evaluating a single time point, overlook the time ordering of inputs, and neglect the time-sensitive nature of time series applications. These factors make it difficult for users, particularly those without domain knowledge, to comprehend AI model decisions and obtain meaningful explanations. We propose CGS-Mask, a post-hoc and model-agnostic cellular genetic strip mask-based saliency approach to address these challenges. CGS-Mask uses consecutive time steps as a cohesive entity to evaluate the impact of features on the final prediction, providing binary and sustained feature importance scores over time. Our algorithm optimizes the mask population iteratively to obtain the optimal mask in a reasonable time. We evaluated CGS-Mask on synthetic and real-world datasets, and it outperformed state-of-the-art methods in elucidating the importance of features over time. According to our pilot user study via a questionnaire survey, CGS-Mask is the most effective approach in presenting easily understandable time series prediction results, enabling users to comprehend the decision-making process of AI models with ease.

Introduction

Artificial intelligence (AI) is increasingly being used for time series prediction, particularly in fields like healthcare (Esteva et al. 2019), physics (Naul et al. 2018), energy (Bedi and Toshniwal 2019), and sensor data (Moin et al. 2021). Ensuring that users can quickly comprehend the reasons behind AI decisions is crucial, particularly in time-sensitive applications where AI is employed to aid in decision-making. Explanation methods are designed to investigate the factors behind a decision and expose any biases or unintended effects of AI models. The transparency and explainability provided by these methods increase human trust in AI models, allowing for more extensive and profound use of AI (Kanamori et al. 2020; Albini et al. 2020; Lu et al. 2023b).



(a)Mask from FO
(b)Mask from RISE
(c)Mask from DeepLIFT
(d)Mask from FIT
(e)Mask from Dynamask
(f)Mask from CGS-Mask

Figure 1: Six explanation masks are employed to analyze data from MIMIC-III, covering the 48 hours preceding a patient's death. In Fig. 1(f), the green strips reveal the key features indicative of patient outcomes. Specifically, a decline in blood pressure, tachycardia of heart rate, and rapid breathing signal an impending risk of death, thereby enabling timely intervention by doctors. However, the other masks do not distinctly identify the periods and features contributing to this outcome, particularly as observed in Fig. 1(a)−(d).

Saliency methods aim to explain the importance of input features for predictions, which is crucial for building trust among stakeholders (Adebayo et al. 2018; Sun et al. 2023). Perturbation-based methods (Ivanovs, Kadikis, and Ozols 2021) like Feature Occlusion (FO) (Suresh et al. 2017) and RISE (Petsiuk, Das, and Saenko 2018), perturb inputs and compare the resulting outcomes to provide explanations. Gradient-based methods like Integrated Gradient (Sundararajan, Taly, and Yan 2017), DeepLIFT (Shrikumar, Greenside, and Kundaje 2017), and GradSHAP (Lundberg et al. 2018) use feature representation gradients for explanation. Attention mechanisms are used with saliency methods (Vaswani et al. 2017; Kwon et al. 2018; Gomez, Fréour, and Mouchère 2022), but their efficacy for explaining black-box models remains a topic of debate (Bai et al. 2021; Wiegreffe and Pinter 2019). SHAP (Lundberg and Lee 2017) uses Shapley values, while LIME (Ribeiro, Singh, and Guestrin 2016) derives explanation from a random local combination of records and their neighborhoods weighted by proximity.

Some saliency methods have been implemented as explainable AI tools and worked well on images, text, and tab-

---

[*]These authors contributed equally.

[†]Corresponding author.

ular data. The direct use of these saliency methods in interpreting time series predictions generally produces results that are hard for humans to comprehend. Existing tools are effective at scoring feature importance at a fine time scale, producing saliency maps with precise numeric values for each feature's saliency score over a time horizon, as shown in Fig. 1(a)−(c). However, these fragmented landscapes may not provide meaningful and engaging explanations to users.

Researchers have started designing saliency methods considering the time-sensitive nature of time-series applications. FIT (Tonekaboni et al. 2020) quantifies the importance of observations over time by evaluating their contribution to the prediction output. However, the feature importance is still scored by numeric values and thus leading to the output as shown in Fig. 1(d), somewhat less intuitive for users. Dynamask (Crabbé and Van Der Schaar 2021) considers the time dependency of time series data in design and perturbs the input with a dynamic combination of the adjacent values of any given data in the input. It avoids rapid changes in feature saliency score over time, as shown in Fig. 1(e), and makes it much more user-friendly than the previous methods. These methods are aware of the continuity of feature importance over time and try to keep it in the explanation (Fong and Vedaldi 2017). Unfortunately, they all score feature importance by numeric values or require other internal network states from the model like gradients. For example, the gradient-based methods are limited to models that provide internal information or have a specific architecture to optimize the final out. Their practicality in real-world applications is constrained (Petsiuk, Das, and Saenko 2018).

To address the above challenges, we propose ***C**ellular **G**enetic **S**trip **Mask*** (CGS-Mask), a saliency method using the perturbation mechanism to explain multivariate time series predictions. CGS-Mask utilizes a strip mask approach, treating consecutive time steps as a cohesive entity to evaluate their impact on the final prediction. It scores feature importance as a binary value for better explanation as shown in Fig. 1(f). We also develop an enhanced cellular genetic algorithm to obtain the final explanation in a reasonable time and make CGS-Mask model agnostic. Our contributions are as below:

1. We consider consecutive time steps as a mask strip to incorporate temporal continuity of features in saliency scoring for AI models used in time series predictions. The resulting feature saliency score is binarized, facilitating clear and intuitive result interpretation.

2. CGS-Mask is a strictly model-agnostic, self-adaptive metaheuristic approach that accurately identifies salient features in time series applications, relying solely on the input and output without requiring knowledge of AI models' inner workings.

3. We compare our method to eight state-of-the-art methods on both synthetic and real-world data sets. Results indicate that our method can identify salient features consistently and provide the best intuitive explanations, as demonstrated by the user study.

Preliminaries

Our mission is to explain the prediction $\mathbf{Y} = f(\mathbf{X})$ of a pre-trained black box model f for the upstream prediction task. In our approach, the input $\mathbf{X}$ is a time series data $\mathbf{X} = (x_{d,t})_{(d,t)\in[1:D]\times[1:T]} \in \mathbb{R}^{D\times T}$, where D is the number of features with T observations over time, $x_{d,t} \in \mathbb{R}^{D\times T}$ is the data point of feature $d \in [1 : D]$ at time $t \in [1 : T]$. $\mathbf{Y} = (y_{d_\mathcal{Y},t_\mathcal{Y}})_{d_\mathcal{Y},t_\mathcal{Y}\in[1:D_\mathcal{Y}][1:T_\mathcal{Y}]} \in \mathbb{R}^{D_\mathcal{Y}\times T_\mathcal{Y}}$ is the prediction from the black box model $f : \mathbb{R}^{D\times T} \to \mathbb{R}^{D_\mathcal{Y}\times T_\mathcal{Y}}$, where $D_\mathcal{Y} \times T_\mathcal{Y}$ is the size of $\mathbf{Y}$. For instance, in a regression task, where the output is a single value, we have $D_\mathcal{Y} = T_\mathcal{Y} = 1$ and $\mathbf{Y} = y_{1,1}$ represents a real number.

Our goal is to identify the parts of the input $\mathbf{X}$ that have a significant impact on the predictions made by a pre-trained black box model f. Drawing upon prior studies (Fong and Vedaldi 2017; Fong, Patrick, and Vedaldi 2019) and established notations (Crabbé and Van Der Schaar 2021; Tonekaboni et al. 2020; Ismail et al. 2020), we use the mask $\mathbf{M}$ to identify the salient features in the input $\mathbf{X}$. The mask is defined as follows:

Definition 1 (Mask $\mathbf{M}$). *For a given input* $\mathbf{X} = (x_{d,t})_{(d,t)\in[1:D]\times[1:T]} \in \mathbb{R}^{D\times T}$ *and the black box model* $f : \mathbb{R}^{D\times T} \to \mathbb{R}^{D_\mathcal{Y}\times T_\mathcal{Y}}$, *the corresponding mask is* $\mathbf{M} = (m_{d,t})_{(d,t)\in[1:D]\times[1:T]} \in [0,1]^{D\times T}$. *The mask* $\mathbf{M}$ *has the same dimensions as the input* $\mathbf{X}$. *Each element* $m_{d,t}$ *in the matrix represents the importance of the feature* d *at time* t *to produce the prediction* $\mathbf{Y} = f(\mathbf{X})$. *In our work,* $m_{d,t}$ *close to 1 indicates that* $x_{d,t}$ *is salient, while* $m_{d,t}$ *close to 0 indicates the opposite.*

As a perturbation-based method, it is necessary to determine how the mask $\mathbf{M}$ perturbs the input $\mathbf{X}$. Different applications may require different methods to explore the input data and produce significant perturbations. Inspired by the perturbation operator used in the context of image classification (Fong, Patrick, and Vedaldi 2019), we propose our definition as below:

Definition 2 (Perturbation Operator $\Gamma_\mathbf{M}$). *The perturbation operator* $\Gamma_\mathbf{M} : \mathbb{R}^{D\times T} \to \mathbb{R}^{D\times T}$ *uses* $\mathbf{M} = (m_{d,t})_{(d,t)\in[1:D]\times[1:T]} \in [0,1]^{D\times T}$ *to perturb* $\mathbf{X} = (x_{d,t})_{(d,t)\in[1:D]\times[1:T]} \in \mathbb{R}^{D\times T}$ *and get the perturbed version* $\widehat{\mathbf{X}} = \Gamma_\mathbf{M}(\mathbf{X})$, *where* $\widehat{\mathbf{X}} = (\hat{x}_{d,t})_{(d,t)\in[1:D]\times[1:T]} \in \mathbb{R}^{D\times T}$. $\hat{x}_{d,t} \in \widehat{\mathbf{X}}$ *can be calculated as follows:*

$$[\Gamma_\mathbf{M}(\mathbf{X})]_{d,t} = \hat{x}_{d,t} = m_{d,t} \times p_{d,t} + (1 - m_{d,t}) \times x_{d,t} \quad (1)$$

Here, $p_{d,t}$ is the perturbation value used to perturb $x_{d,t}$, $x_{d,t}$ is the value of feature d at the time t, and $m_{d,t}$ is the saliency value for $x_{d,t}$. For time series data, $p_{d,t}$ can also be set to some specific values that consider the time dependency of input data (Crabbé and Van Der Schaar 2021). $p_{d,t}$ can thus be calculated in one of the following forms:

$$p_{d,t} = \frac{1}{T}\sum_{t'=1}^{T} x_{d,t'} \quad (2)$$

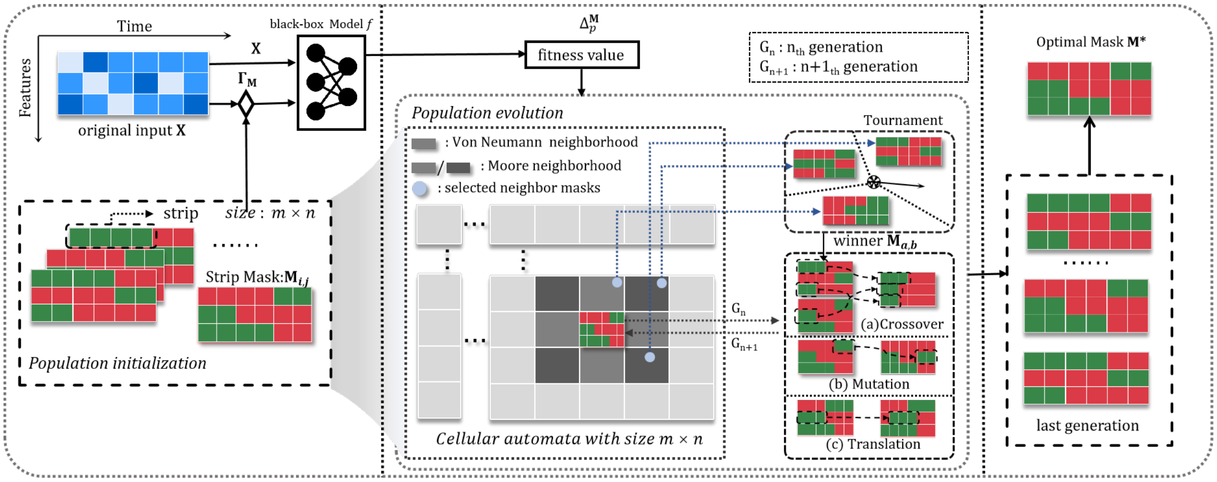


Figure 2: The overall framework of CGS-Mask.

$$p_{d,t} = \frac{1}{2\mathbf{K}}\Big(\sum_{t'=t-\mathbf{K}}^{t-1} x_{d,t'} + \sum_{t'=t+1}^{t+\mathbf{K}} x_{d,t'}\Big) \tag{3}$$

Equation (2) represents the perturbation value as the average value of feature d. Equation (3) indicates that the perturbation value is the average value of feature d within the time range of $[t-\mathbf{K}, t-1] \cup [t+1, t+\mathbf{K}]$. Other suitable forms for specific applications are also acceptable.

Ideally, the objective mask identifies a subset of salient features in the input $\mathbf{X}$ that explain the prediction. Our work will assign a high value to $m_{d,t}$ if $x_{d,t}$ impacts the prediction $\mathbf{Y}$. Specifically, we have the following proposition:

Proposition 1. If $x_{d,t} \in \mathbf{X}$ is salient for $f(\mathbf{X})$, $p_{d,t}$ is not salient for $f(\mathbf{X})$, $m_{d,t} \in \mathbf{M}$ and $m_{d,t} > 0$, then $||f(\mathbf{X}) - f(\mathbf{\Gamma_M}(\mathbf{X}))|| > 0$.

Proof. The proof of proposition 1 is quite straightforward. Say $x_{d,t}$ is salient for $f(\mathbf{X})$. After the perturbation by Equation (1), $x_{d,t}$ will change to $\hat{x}_{d,t}$ with the perturbed value $p_{d,t}$, which is not as the same salient as $x_{d,t}$ for $f(\mathbf{X})$. It means that $f(\hat{\mathbf{X}})$ differs from $f(\mathbf{X})$, so we can conclude that $||f(\mathbf{X}) - f(\mathbf{\Gamma_M}(\mathbf{X}))|| > 0$ □

According to the above proposition, we remark that the more salient area a mask $\mathbf{M}$ can represent, the bigger the difference between $f(\mathbf{X})$ and $f(\mathbf{\Gamma_M}(\mathbf{X}))$. Therefore, comparing the predictions from both unperturbed and perturbed inputs is helpful. We call the difference of the predictions with perturbed and unperturbed values as *perturbation error* $\Delta_p^\mathbf{M}$ and give a definition as below:

Definition 3 (Perturbation Error $\Delta_p^\mathbf{M}$)**.** *The perturbation error $\Delta_p^\mathbf{M}$ evaluates the performance of each potential mask. It can be calculated by $\mathbf{X}$ and its perturbed version $\widehat{\mathbf{X}} = \Gamma_\mathbf{M}(\mathbf{X})$. The specific calculation of $\Delta_p^\mathbf{M}$ varies depending on the prediction task. In the case of a regression task, where $\mathbf{Y} \in \mathbb{R}^{D_\mathcal{Y} \times T_\mathcal{Y}}$, we use the squared error between the original $\mathbf{X}$ and the perturbed prediction $\hat{\mathbf{X}}$ to derive $\Delta_p^\mathbf{M}$ as follows:*

$$\Delta_p^\mathbf{M} = \sum_{d_y=1}^{D_\mathcal{Y}} \sum_{t_y=1}^{T_\mathcal{Y}} \left([f(\mathbf{X})]_{d_y,t_y} - [f(\Gamma_\mathbf{M}(\mathbf{X}))]_{d_y,t_y}\right)^2 \tag{4}$$

To evaluate the impact of the perturbation of input $\mathbf{X}$ on the prediction in classification tasks, we use the cross entropy to derive $\Delta_p^\mathbf{M}$. Here, $f(\mathbf{X})$ represents the predicted probability by the classifier for $\mathbf{X}$. Then $\Delta_p^\mathbf{M}$ can be defined as follows:

$$\Delta_p^\mathbf{M} = -\sum_{t_\mathcal{Y}=1}^{T_\mathcal{Y}} \sum_{c=1}^{D_\mathcal{Y}} [f(\mathbf{X})]_{t_\mathcal{Y},c} \log[f(\Gamma_\mathbf{M}(\mathbf{X}))]_{t_\mathcal{Y},c} \tag{5}$$

To obtain the optimal mask $\mathbf{M}^*$, we aim to maximize $\Delta_p^\mathbf{M}$. The objective function is thus defined as:

$$\mathbf{M}^* = \underset{\mathbf{M}\in[0,1]^{D\times T}}{\arg\max}\ \Delta_p^\mathbf{M} \tag{6}$$

Methodology

To highlight the continuity of feature importance in the time domain, we first design a strip mask to incorporate the temporal dependency in the results. Then, CGS-Mask uses a modified cellular genetic algorithm to optimize the masks to get the optimal one with the highest $\Delta_p^\mathbf{M}$.

Strip Mask

As discussed, the mask built by Definition 1 captures the feature saliency in the finest time scale but may result in a less coherent narrative. To enhance interpretability, we need a mask that provides a more intuitive explanation of the prediction and is easily understandable at both fine and coarse time scales. Thus, we propose designing a mask that spans consecutive time steps and allows for scaling when necessary. Additionally, relying on numerical ranks to convey the saliency score might not efficiently convey the information to users, as the subtle differences between adjacent numbers can be difficult to interpret adequately. To tackle these, we introduce the concept of **strip** to build our mask. For the

Algorithm 1: CGS-Mask generation

```
Input: The black-box prediction model $f$, two-dimensional cellular automata (CA) in size $m \times n$, N rounds of generations, the probability of crossover operator $P_c$, the probability of mutation operator $P_m$, and the probability of translation operator $P_t$.
Output: The optimal mask $\mathbf{M}^*$
Initialize the population of strip masks with size $m \times n$ and put each strip mask into CA, n=0. Ensure $P_c, P_m, P_t \in [0,1], P_c + P_m + P_t \leq 1$
while $n \leq N$ do
    Calculate the fitness value of each mask using the fitness function of Equation (4) or (5) ;
    for each mask $\mathbf{M_{i,j}}$ do
        generate a random number $r \in [0,1]$;
        if $0 \leq r \leq P_c$ then
            $\mathbf{M_S} = Select(\mathbf{M_E})$;
            $\mathbf{M_{a,b}} = Tournament(\mathbf{M_S})$;
            $Crossover(\mathbf{M_{i,j}}, \mathbf{M_{a,b}})$;
        else if $P_c < r \leq P_c + P_m$ then
            $Mutation(\mathbf{M_{i,j}})$;
        else if $P_c + P_m < r \leq P_c + P_m + P_t$ then
            $Translation(\mathbf{M_{i,j}})$;
        end if
    end for
    n++;
end while
$\mathbf{M}^* = Optimal(\{\mathbf{M_{i,j}} | i \in [1,m], j \in [1,n]\})$;
return $\mathbf{M}^*$
```

strip mask $\mathbf{M}$, the strips represent a subset of input $\mathbf{X}$ that significantly impacts the prediction result of a pre-trained black-box model f. The strips are arranged in consecutive time steps, and the values in the mask are binary to enhance its interpretability. We define the strip mask as follows:

Definition 4 (Strip Mask)**.** *For a given input* $\mathbf{X} = (x_{d,t})_{(d,t)\in[1:D]\times[1:T]} \in \mathbb{R}^{D\times T}$*, the corresponding strip mask* $\mathbf{M} = (m_{d,t}) \in \{0,1\}^{D\times T}$ *contains binary values. Each strip mask* $\mathbf{M}$ *has a strip set* $\mathbf{S} = \{S_n \mid n \in [1,N]\}$*, where* $N \in \mathbb{N}$ *is the number of strips in the mask. To generate a strip* $S_n \in \mathbf{S}$*, we randomly initialize its starting position* $b \in [1,T]$*, the feature index* $d \in [1,D]$*, and the strip size* l*. For each point* $m_{d,t}$ *in the strip, we set* $m_{d,t} = 1\ for\ all\ t \in [b, b+l]$.

Please note that in our design, the strip size l can be set to one, effectively resulting in a traditional time point-based mask. Therefore, in the rest of our work, the terms mask and strip mask are used interchangeably, as traditional masks can be seen as a specific form of our strip mask.

Strip Mask Optimization

Our objective is to generate a mask that can identify a subset of the original input to explain the prediction. This task can be seen as a subset selection problem, which is generally an NP-hard problem (Qian, Yu, and Zhou 2015; Natarajan 1995). Metaheuristic algorithms are commonly employed to tackle such subset search problems (Bian et al. 2020; Rostami et al. 2021). In contrast to existing models that rely on the internal state of the model, such as gradient-based methods (Zhang et al. 2018; Simonyan, Vedaldi, and Zisserman 2013), metaheuristic algorithms do not necessitate any internal knowledge of the model, resulting in a model-agnostic interpretation algorithm.

We present a model-agnostic metaheuristic approach called the self-adaptive cellular genetic algorithm (SA-CGA) to solve the strip search problem. With its help we can optimize the strip as the basic unit of the mask without relying on the internal state of models. SA-CGA seamlessly combines the genetic algorithm and cellular automata (Li, Wu, and Li 2018). By treating a single cell as an individual in the population, we can implement the operators of the genetic algorithm alongside the neighbor rule utilized in the cellular automata model. This design enables individuals to gather information from local surroundings, learn from neighboring nodes in a complex system, adapt actively to the environment, improve their fitness value, and ultimately make better decisions.

Based on SA-CGA, we treat each strip mask as an individual and place them into a two-dimensional cellular automata. We design genetic operators to enable information exchange between neighboring masks in the cellular automata. Our algorithm, called CGS-Mask, is shown in Algorithm 1. Initially, we create a population of strip masks and map them into the cellular automata. We then optimize each mask to evolve into the next generation using genetic operators like crossover, mutation, and translation operators. The optimal mask will be found after N rounds of generations and we select the mask with highest fitness value as the optimal mask.

Population Initialization CGS-Mask randomly initializes the mask set $\{\mathbf{M_{i,j}} = (m_{d,t}^{(i,j)}) \in \{0,1\}^{D\times T} | i \in [1,m], j \in [1,n]\}$. As illustrated in Fig. 2, we map $\mathbf{M_{i,j}}$ to the (i, j) position of the cellular automata with the size of $m \times n$. For each mask, there is a set of neighbor masks $\mathbf{M_E}$. The neighbors of $\mathbf{M_{i,j}}$ can be set in various ways, e.g., the von Neumann neighbor or Moore neighbor shown in Fig. 2.

Population Evolution To evaluate the performance of the strip mask, we need to calculate the fitness value for each strip mask $\mathbf{M_{i,j}}$. Our approach leverages the perturbation error $\Delta_p^{\mathbf{M}}$, as defined in Definition 3, to determine the fitness value of each mask. The fitness function, denoted by Equation (4) for regression tasks and Equation (5) for classification tasks, is employed to calculate the fitness value. Next, we need to optimize the original population iteratively. The initial population is denoted as $\mathbf{G_0}$, and the n-th generation is $\mathbf{G_n}$. CGS-Mask aims to optimize $\mathbf{G_n}$ to obtain the next generation $\mathbf{G_{n+1}}$. For each mask $\mathbf{M} \in \mathbf{G_n}$, we use the crossover operator with probability P_c, the mutation operator with probability P_m, and the translation operator with probability P_t to generate the new masks that move towards optimality. We denote this process as $\mathbf{M} \longrightarrow \mathbf{M}'$, where $\mathbf{M}' \in \mathbf{G_{n+1}}$.

Crossover Operator: As mentioned, each mask $\mathbf{M_{i,j}}$ has the neighbor set $\mathbf{M_E}$. For example, the Moore neighbor set

	rare feature				rare time			
Methods	$AUP \uparrow$	$AUR \uparrow$	$\mathcal{D}_{\mathbf{M}} \downarrow$	$\mathcal{E}_{\mathbf{M}} \downarrow$	$AUP \uparrow$	$AUR \uparrow$	$\mathcal{D}_{\mathbf{M}} \downarrow$	$\mathcal{E}_{\mathbf{M}} \downarrow$
FO	$\mathbf{1.00 \pm 0.00}$	0.14 ± 0.02	128.70 ± 2.33	11.14 ± 0.69	$\mathbf{1.00 \pm 0.00}$	0.13 ± 0.02	119.45 ± 2.48	48.34 ± 1.12
FP	$\mathbf{1.00 \pm 0.00}$	0.16 ± 0.03	121.02 ± 2.72	14.34 ± 0.78	$\mathbf{1.00 \pm 0.00}$	0.23 ± 0.02	125.76 ± 2.76	54.57 ± 0.97
IG	0.99 ± 0.01	0.14 ± 0.03	137.98 ± 1.29	11.95 ± 0.56	0.99 ± 0.01	0.17 ± 0.03	128.39 ± 1.93	48.94 ± 0.59
SVS	0.99 ± 0.01	0.18 ± 0.04	141.93 ± 2.08	11.18 ± 0.63	$\mathbf{1.00 \pm 0.00}$	0.19 ± 0.03	122.31 ± 2.31	48.73 ± 0.14
FIT	0.98 ± 0.01	0.27 ± 0.02	138.23 ± 2.43	13.76 ± 0.52	0.99 ± 0.01	0.39 ± 0.03	127.45 ± 3.02	51.56 ± 1.93
RISE	0.97 ± 0.02	0.35 ± 0.02	145.24 ± 2.97	14.44 ± 0.67	0.96 ± 0.01	0.33 ± 0.02	129.49 ± 1.93	47.44 ± 1.45
DeepLIFT	0.99 ± 0.00	0.33 ± 0.03	129.75 ± 1.96	12.58 ± 0.33	0.99 ± 0.01	0.42 ± 0.03	131.24 ± 2.57	41.53 ± 1.89
Dynamask	0.99 ± 0.01	0.58 ± 0.03	82.73 ± 1.22	0.83 ± 0.02	0.99 ± 0.01	0.63 ± 0.04	95.45 ± 1.19	7.13 ± 0.37
w/o Strip	0.99 ± 0.01	0.75 ± 0.02	27.29 ± 0.92	$\mathbf{0.00 \pm 0.00}$	0.98 ± 0.01	0.76 ± 0.02	79.23 ± 1.37	$\mathbf{0.00 \pm 0.00}$
w/o Cell	0.99 ± 0.01	0.73 ± 0.02	14.71 ± 0.08	$\mathbf{0.00 \pm 0.00}$	0.99 ± 0.01	0.75 ± 0.02	37.83 ± 0.89	$\mathbf{0.00 \pm 0.00}$
w/o Trans	0.98 ± 0.01	0.78 ± 0.02	15.27 ± 0.97	$\mathbf{0.00 \pm 0.00}$	0.99 ± 0.01	0.78 ± 0.03	36.52 ± 0.97	$\mathbf{0.00 \pm 0.00}$
CGS-Mask	$\mathbf{1.00 \pm 0.00}$	$\mathbf{0.81 \pm 0.02}$	$\mathbf{14.34 \pm 0.93}$	$\mathbf{0.00 \pm 0.00}$	0.99 ± 0.01	$\mathbf{0.82 \pm 0.04}$	$\mathbf{34.78 \pm 1.15}$	$\mathbf{0.00 \pm 0.00}$
	mixture				random			
FO	$\mathbf{1.00 \pm 0.00}$	0.18 ± 0.03	226.78 ± 8.64	55.24 ± 2.93	0.99 ± 0.00	0.26 ± 0.02	222.95 ± 4.35	83.56 ± 2.32
FP	0.99 ± 0.00	0.28 ± 0.03	223.81 ± 5.79	69.33 ± 2.33	0.99 ± 0.00	0.23 ± 0.03	221.42 ± 7.32	84.44 ± 3.75
IG	0.99 ± 0.00	0.16 ± 0.02	234.52 ± 8.93	57.53 ± 3.75	$\mathbf{1.00 \pm 0.00}$	0.17 ± 0.02	237.57 ± 6.93	101.16 ± 3.72
SVS	0.99 ± 0.01	0.18 ± 0.03	229.40 ± 7.64	55.23 ± 2.34	$\mathbf{1.00 \pm 0.00}$	0.26 ± 0.02	236.83 ± 3.54	83.58 ± 3.38
FIT	0.99 ± 0.01	0.36 ± 0.03	225.73 ± 5.39	61.39 ± 1.32	0.99 ± 0.00	0.32 ± 0.03	231.73 ± 3.76	87.39 ± 2.24
RISE	0.96 ± 0.01	0.33 ± 0.02	289.43 ± 6.73	52.44 ± 3.44	0.93 ± 0.04	0.35 ± 0.03	269.37 ± 7.39	91.44 ± 3.92
DeepLIFT	0.98 ± 0.01	0.37 ± 0.03	231.47 ± 3.72	53.72 ± 1.35	0.99 ± 0.00	0.29 ± 0.02	246.52 ± 4.39	83.26 ± 2.34
Dynamask	$\mathbf{1.00 \pm 0.00}$	0.59 ± 0.02	122.96 ± 2.93	7.89 ± 0.68	0.99 ± 0.00	0.56 ± 0.03	145.83 ± 3.65	21.08 ± 0.83
w/o Strip	0.98 ± 0.01	0.76 ± 0.03	104.72 ± 2.34	$\mathbf{0.00 \pm 0.00}$	0.98 ± 0.01	0.75 ± 0.03	142.67 ± 3.74	$\mathbf{0.00 \pm 0.00}$
w/o Cell	0.99 ± 0.00	0.74 ± 0.02	59.34 ± 1.73	$\mathbf{0.00 \pm 0.00}$	0.96 ± 0.01	0.74 ± 0.02	71.89 ± 3.26	$\mathbf{0.00 \pm 0.00}$
w/o Trans	0.99 ± 0.00	0.77 ± 0.03	$\mathbf{58.32 \pm 2.71}$	$\mathbf{0.00 \pm 0.00}$	0.98 ± 0.01	0.77 ± 0.03	71.67 ± 2.34	$\mathbf{0.00 \pm 0.00}$
CGS-Mask	0.99 ± 0.01	$\mathbf{0.82 \pm 0.03}$	58.45 ± 2.33	$\mathbf{0.00 \pm 0.00}$	0.96 ± 0.01	$\mathbf{0.79 \pm 0.03}$	$\mathbf{69.73 \pm 1.03}$	$\mathbf{0.00 \pm 0.00}$

Table 1: Results on the synthetic data sets.

$\mathbf{M_E}$ for $\mathbf{M_{i,j}}$ includes eight neighbors as shown in Fig.2. Then, we select the mask set $\mathbf{M_S} \subseteq \mathbf{M_E}$ with the predefined probability P_c. In this case, the fitness value of the $\mathbf{M} \in \mathbf{M_S}$ should not be less than that of $\mathbf{M_{i,j}}$. We implement a tournament on $\mathbf{M_S}$, and based on the probability calculated from the fitness values of $\mathbf{M} \in \mathbf{M_S}$, we choose one of them, denoted as $\mathbf{M_{a,b}} \in \mathbf{M_S}$, to perform crossover with $\mathbf{M_{i,j}}$. The probability is defined as follows:

$$P_{a,b} = \frac{\Delta_p^{\mathbf{M_{a,b}}}}{\sum\limits_{\mathbf{M_{a',b'}} \in \mathbf{M_S}} \Delta_p^{\mathbf{M_{a',b'}}}} \tag{7}$$

where $P_{a,b}$ represents the probability of $\mathbf{M_{a,b}}$ winning the tournament against other masks in $\mathbf{M_S}$ to undergo crossover with $\mathbf{M_{i,j}}$.

After obtaining $\mathbf{M_{i,j}}$ and $\mathbf{M_{a,b}}$, the crossover operator will combine them to generate a new mask. In CGS-Mask, the strip is the atomic genetic unit for the crossover operation. The strip offspring can inherit from either parent in the crossover operator. Assuming mask $\mathbf{M_{i,j}}$ has strip set $\mathbf{S_{i,j}}$ and mask $\mathbf{M_{a,b}}$ has $\mathbf{S_{a,b}}$, let $S_{i,j}^{\alpha} \in \mathbf{S_{i,j}}$ and $S_{a,b}^{\beta} \in \mathbf{S_{a,b}}$, where $\alpha, \beta \in [1, U]$ and U is the number of strips for both $\mathbf{S_{i,j}}$ and $\mathbf{S_{a,b}}$. Mask $\mathbf{M_o}$ is the offspring of $\mathbf{M_{i,j}}$ and $\mathbf{M_{a,b}}$, and $\mathbf{M_o}$ has the strip set $\mathbf{S_o} = \bigcup_{\alpha,\beta=1}^{U} Choose(S_{i,j}^{\alpha}, S_{a,b}^{\beta})$, where

$$P(Choose(S_{i,j}^{\alpha}, S_{a,b}^{\beta}) = S_{i,j}^{\alpha}) = \frac{\Delta_p^{\mathbf{M_{i,j}}}}{\Delta_p^{\mathbf{M_{i,j}}} + \Delta_p^{\mathbf{M_{a,b}}}} \tag{8}$$

The acts of the crossover operator are shown in lines 7-10 in Algorithm 1 and (a) Crossover in Fig. 2.

Mutation Operator: The mutation operator in CGS-Mask fosters genetic diversity and prevents masks from converging to local minima. During mutation, a strip in the mask may be replaced by a new strip with a certain probability. Let $\mathbf{M}$ represent the mask with strip set $\mathbf{S}$. $S_i \in \mathbf{S}$ denotes the strip that will be deleted in the offspring of $\mathbf{M}$, while S_j represents the strip that will be added. Hence, the strip set of the offspring $\mathbf{M_o}$ can be calculated as follows:

$$\mathbf{S_o} = (\mathbf{S} - \{S_i\}) \cup \{S_j\} \tag{9}$$

The process of the mutation operator is given in line 11 in Algorithms 1 and (b) Mutation in Fig. 2.

Translation Operator: In a common scenario, we may encounter a strip S_i that shares the same features and length as the true salient strip S_t but is misaligned by a few time steps. Simply mutating S_i could alter its features and reduce the mask's fitness value. To address this, we propose the translation operator. It adjusts the position offset of strips on the timeline in CGS-Mask. Given a mask $\mathbf{M}$ with strip set $\mathbf{S}$, we select a strip $S_i \in \mathbf{S}$ to be translated in the offspring. By shifting the original beginning t_i of S_i by a translation distance t_m, the new beginning t_n in the offspring is determined as $t_n = t_i \pm t_m$. The translation operator is indicated in line 13 of Algorithm 1 and (c) Translation in Fig. 2.

After N generations, the last generation $\mathbf{G_N}$ is obtained. The mask with the highest fitness value is selected as the output mask $\mathbf{M}^*$.

Experiments

We conducted experiments to evaluate the performance of CGS-Mask and compared it with eight saliency methods on four *synthetic data sets* and four *real-world data sets*.

	MIMIC-III			LSST		
Methods	$\Delta_p^{\mathrm{M}} \uparrow$	$\mathcal{D}_{\mathrm{M}} \downarrow$	$\mathcal{E}_{\mathrm{M}} \downarrow$	$\Delta_p^{\mathrm{M}} \uparrow$	$\mathcal{D}_{\mathrm{M}} \downarrow$	$\mathcal{E}_{\mathrm{M}} \downarrow$
FO	0.035 ± 0.002	118.63 ± 2.34	62.35 ± 2.07	1.21 ± 0.13	29.75 ± 1.23	9.24 ± 1.25
FP	0.034 ± 0.002	115.73 ± 2.31	77.28 ± 2.02	1.33 ± 0.11	28.30 ± 1.23	10.93 ± 1.34
IG	0.051 ± 0.002	128.34 ± 2.27	83.94 ± 1.93	1.47 ± 0.12	29.36 ± 2.01	10.96 ± 0.97
SVS	0.031 ± 0.001	132.38 ± 2.32	61.53 ± 1.95	1.29 ± 0.11	30.14 ± 1.37	10.02 ± 0.83
FIT	0.034 ± 0.003	122.53 ± 2.97	27.83 ± 2.39	1.51 ± 0.12	32.35 ± 2.31	9.53 ± 0.72
RISE	0.033 ± 0.002	119.40 ± 3.23	43.44 ± 2.34	1.28 ± 0.08	29.48 ± 1.75	10.77 ± 0.93
DeepLIFT	0.035 ± 0.002	129.83 ± 2.56	37.65 ± 1.38	1.43 ± 0.05	27.39 ± 1.92	9.06 ± 0.99
Dynamask	0.072 ± 0.003	91.88 ± 2.01	0.77 ± 0.06	1.77 ± 0.09	17.45 ± 1.36	0.53 ± 0.08
w/o Strip	0.082 ± 0.004	83.45 ± 2.07	$\mathbf{0.00 \pm 0.00}$	1.91 ± 0.03	14.34 ± 1.03	$\mathbf{0.00 \pm 0.00}$
w/o Cell	0.080 ± 0.003	46.75 ± 1.49	$\mathbf{0.00 \pm 0.00}$	1.87 ± 0.05	10.64 ± 0.93	$\mathbf{0.00 \pm 0.00}$
w/o Trans	0.082 ± 0.003	47.32 ± 1.74	$\mathbf{0.00 \pm 0.00}$	1.89 ± 0.04	11.13 ± 0.89	$\mathbf{0.00 \pm 0.00}$
CGS-Mask	$\mathbf{0.084 \pm 0.004}$	$\mathbf{46.45 \pm 1.93}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{1.92 \pm 0.05}$	$\mathbf{10.34 \pm 0.85}$	$\mathbf{0.00 \pm 0.00}$
	NATOPS			AE		
FO	1.37 ± 0.05	110.77 ± 2.34	69.72 ± 2.01	106.01 ± 3.57	$327, 65 \pm 5.31$	137.62 ± 2.03
FP	1.63 ± 0.04	115.76 ± 2.34	63.84 ± 1.29	110.23 ± 4.87	329.04 ± 4.45	131.52 ± 5.31
IG	1.69 ± 0.05	114.39 ± 3.87	65.60 ± 2.36	107.97 ± 2.73	332.83 ± 7.86	119.74 ± 2.37
SVS	1.72 ± 0.04	119.34 ± 3.75	63.52 ± 2.37	109.09 ± 4.93	322.34 ± 9.38	121.03 ± 3.94
FIT	1.81 ± 0.03	129.38 ± 2.71	46.93 ± 2.39	112.77 ± 4.93	315.14 ± 7.34	93.63 ± 4.32
RISE	1.71 ± 0.04	113.50 ± 2.33	69.77 ± 3.71	106.93 ± 3.97	329.49 ± 8.34	116.65 ± 5.31
DeepLIFT	1.74 ± 0.04	102.30 ± 3.40	63.75 ± 2.39	116.39 ± 4.91	315.37 ± 7.84	129.61 ± 4.23
Dynamask	1.89 ± 0.06	89.95 ± 4.32	0.69 ± 0.07	121.15 ± 2.93	217.41 ± 6.42	2.84 ± 0.42
w/o Strip	2.02 ± 0.08	73.29 ± 4.03	$\mathbf{0.00 \pm 0.00}$	124.33 ± 3.71	164.59 ± 5.37	$\mathbf{0.00 \pm 0.00}$
w/o Cell	1.99 ± 0.05	39.42 ± 3.70	$\mathbf{0.00 \pm 0.00}$	122.72 ± 4.69	99.59 ± 4.31	$\mathbf{0.00 \pm 0.00}$
w/o Trans	2.09 ± 0.07	38.56 ± 2.87	$\mathbf{0.00 \pm 0.00}$	127.56 ± 3.54	101.59 ± 3.74	$\mathbf{0.00 \pm 0.00}$
CGS-Mask	$\mathbf{2.15 \pm 0.09}$	$\mathbf{38.29 \pm 2.95}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{131.24 \pm 3.72}$	$\mathbf{97.59 \pm 3.33}$	$\mathbf{0.00 \pm 0.00}$

Table 2: Results on the real-world data sets.

We also conducted a pilot user study to gain insights into how different saliency methods help users understand the decision-making process of AI models. More details about hardware resources, implementation, metrics, and results can be found in Technical Appendix (Lu et al. 2023a).

Experiments With Synthetic Data Sets

Data sets and settings We conducted experiments using synthetic data sets, including the *rare features* and *rare time* data sets from (Ismail et al. 2020). The *rare features* data set consists of a small subset of salient features, while the *rare time* data set contains a small subset of salient time points. We also created a *mixture* data set that combines the *rare features* and *rare time* data sets, and a *random* data set with randomly located salient input regions. In the experiments, the prediction model only relies on a known subset area $A = \{a_{p,q} | p \in [1, D], q \in [1, T]\}$ of the input $\mathbf{X}$ and f is a function of A that enables us to derive the ground truth. In our experiments, we set $D = T = 50$. For the former two data sets $|A| = 125$, and for the latter two data sets $|A| = 250$. For simplicity, the perturbation value was set to zero. Since all tasks were regression, we evaluated the mask by Equation (4). The input $\mathbf{X}$ is generated through ARMA process (Crabbé and Van Der Schaar 2021).

Metrics As the ground truth explanations were known for the four synthetic data sets, we used the four below metrics to evaluate the performance. The *area under the precision curve* (AUP) is the first metric to measure the proportion of identified features that are indeed salient. We used the *area under the recall curve* (AUR) to measure the portion of the succesful identified salient features. Both metrics are higher the better (Crabbé and Van Der Schaar 2021). The *Mask Discreteness* ($\mathcal{D}_{\mathrm{M}}$) assesses the temporal continuity of the mask (Mercier et al. 2022). It quantifies the difference between $m_{t-1,d}$ and $m_{t,d}$ for each feature and time step. Lastly, the *Mask Entropy* ($\mathcal{E}_{\mathrm{M}}$) (Crabbé and Van Der Schaar 2021) is to measure the masks' sharpness and legibility. They are lower the better.

Benchmarks We compared our approach CGS-Mask with Dynamask (Crabbé and Van Der Schaar 2021), DeepLIFT (Shrikumar, Greenside, and Kundaje 2017), RISE (Petsiuk, Das, and Saenko 2018), FIT (Tonekaboni et al. 2020), Shapley Value Sampling (SVS) (Molnar 2020), Feature Occlusion (FO), Feature Permutation (FP) (Fisher, Rudin, and Dominici 2019), and Integrated Gradient (IG) in the tests.

Results and Analysis The experiment results on four synthetic data sets are shown in Table 1. We can see that all methods performed well on the AUP metric in determining the salient features. CGS-Mask significantly outperformed all baselines on AUR. The result suggests that CGS-Mask can determine more salient features over time due to its perturbation operator designed to incorporate temporal ordering and the time-sensitive nature of the input. CGS-Mask's $\mathcal{D}_{\mathrm{M}}$ is significantly lower than the baselines, ranging from 52.18% to 82.67%. The most consistent explanation over time comes from the strip mask's temporal continuity, which makes the results easy to understand. $\mathcal{E}_{\mathrm{M}}$ of CGS-Mask achieved zero means that its polarization reaches the theoretical minimum. This result is a direct outcome of the strip mask's binary value. In contrast, other methods are multi-valued based and exhibit diversity in feature saliency representation. The runtime of CGS-Mask is comparable to that of other methods, such as Dynamask. Find the runtime details in Technical Appendix (Lu et al. 2023a).

Ablation study We performed an ablation study on CGS-Mask by a set of component variations to measure the effects

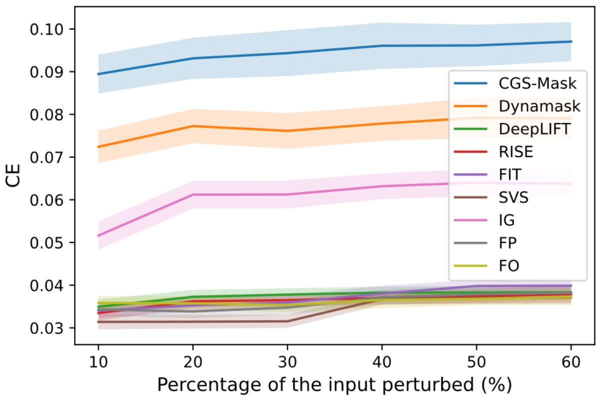


Figure 3: Cross entropy (CE) of saliency methods used in MIMIC-III.

of each component in CGS-Mask. Specifically, we considered the following variants of our method:

- **w/o Strip**: The length of strips in CGS-Mask was set to 1, which means that the algorithm considers input saliency on an individual point basis. Therefore, the strip mask is transformed into a regular mask that operates on individual points.
- **w/o Cell**: The standard genetic algorithm replaced the cellular genetic algorithm.
- **w/o Trans**: The translation operator was excluded from the population evolution module as it is not a standard component of the cellular genetic algorithm.

The ablation study results in Table 1 demonstrate the impact of each component on the performance of CGS-Mask. Notably, the absence of the strip component (**w/o Strip**) significantly increased the value of $\mathcal{D}_{\mathrm{M}}$, suggesting that the strip design contributes to capturing the continuity of features in the explanation results. Additionally, the use of the cellular genetic algorithm and translation operator proved beneficial in recovering salient areas, as observed by the lower AUR performance in the cases of **w/o Cell** and **w/o Trans** compared to CGS-Mask.

Results on Real-World Data

Data sets and settings To further evaluate the performance of CGS-Mask, we conducted experiments on four individual real-world data sets from the healthcare, astronomy, sensors, and energy domains. The *MIMIC-III* data set (Johnson et al. 2016) contains the health record of 40,000 ICU patients at the Beth Israel Deaconess Medical Center. We used it to predict patient survival rate 48 hours ahead with 31 patient features. It is a classification task with two classes, survival and death. For a fair comparison, the data selection, preprocessing, and model training were the same as (Tonekaboni et al. 2020). *LSST* simulated astronomical time-series data in preparation for observations from the Large Synoptic Survey Telescope (Emille et al. 2018). The prediction model needs to classify these data into 14 different classes. *NATOPS* data set is generated by sensors on the hands, elbows, wrists, and thumbs for gesture recognition (Ghouaiel, Marteau, and Dupont 2017). These data need to be classified into six different kinds of gestures. *AE* was obtained from the Appliances Energy Prediction data set from the UCI repository to predict the total energy usage of a house (Candanedo, Feldheim, and Deramaix 2017). It is a regression task, and the prediction model outputs a value representing the total energy usage in kWh. We used accuracy to evaluate the classification model's performance for *MIMIC-III*, *LSST*, and *NATOPS*. We also used Mean Square Error (MSE) to evaluate the regression model's performance for *AE*.

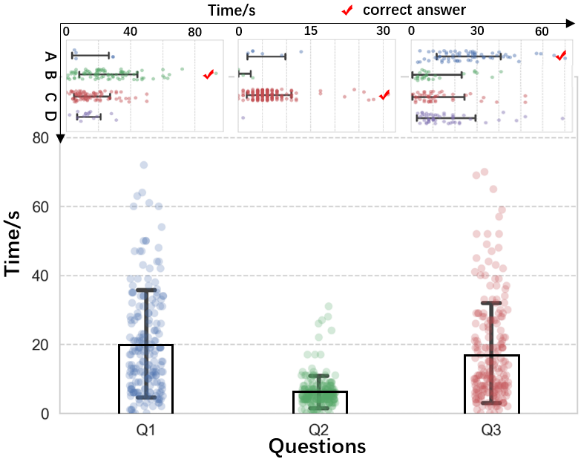


Figure 4: User response time and selection results.

Metrics Since we did not have the real-world data sets' ground truth explanation, we used the perturbation error Δ_p^{M} to evaluate the performance of each result mask delivered by benchmarks. For *MIMIC-III*, *LSST*, and *NATOPS*, Δ_p^{M} is the cross entropy (CE) between $f(\mathbf{X})$ and $f(\hat{\mathbf{X}})$. For *AE*, Δ_p^{M} is the square error between $f(\mathbf{X})$ and $f(\hat{\mathbf{X}})$. We also used the $\mathcal{D}_{\mathrm{M}}$ and $\mathcal{E}_{\mathrm{M}}$ metrics as before.

Benchmarks We used the same benchmarks as before.

Results and Analysis For a fair comparison, we evaluated Δ_p^{M} of each method by considering saliency regions of 10%, 20%, 30%, 40%, 50%, and 60% of the input on the *MIMIC-III* data set (Crabbé and Van Der Schaar 2021; Tonekaboni et al. 2020). The results are shown in Fig. 3. The largest value of CE (Δ_p^{M}) on *MIMIC-III* data set indicates our ability to identify the subset of features with the greatest impact on the models. We also evaluated all methods on diverse real-world datasets, using a saliency region comprising 10% of the input. The results are given in Table 2. Once again, CGS-Mask outperformed other methods across all metrics. The strip mask showed the best continuity as it had the lowest $\mathcal{D}_{\mathrm{M}}$ value. Thanks to the binary nature of CGS-Mask, it attained a perfect $\mathcal{E}_{\mathrm{M}}$ of 0. The runtime of CGS-Mask remains efficient, and additional details can be found in Technical Appendix (Lu et al. 2023a).

Ablation study We conducted an ablation study on real-world data, replicating the previous variations. The results in Table 2 confirm the influence of all components on the performance of CGS-Mask. The design of the strip mask directly impacted $\mathcal{D}_{\mathrm{M}}$, while the cellular genetic algorithm and translation operator improved the generation of effective masks for explanation purposes.

Pilot User Study

To assess the legibility of the generated masks, we conducted a survey involving 254 participants across different age groups (5 to 83 years old) and with varying levels of domain knowledge. The participants were asked to rank the saliency masks obtained from six different methods based on their effectiveness in helping the user understand the salient features and their temporal relevance. Our study showed that users preferred CGS-Mask for its intuitive comprehension of time series models. The results shows that over 65% of users ranked CGS-Mask as their top choice, while over 85% rated it among their top three. In addition, we conducted a user study to measure the reaction time and accuracy of determining feature importance using three saliency masks (Q1, Q2, and Q3) with four features (A, B, C, and D) over 10 time steps. As shown in Fig 4, CGS-Mask (Q2) exhibited a significantly shorter response time of 6.26 ± 4.62s and higher answer accuracy of 85.4% compared to the numerical masks (Q1 and Q3), which had a response time of 19.22 ± 15.04s and an accuracy of 40.6%. Further details of the pilot user study can be found in Technical Appendix (Lu et al. 2023a).

Conclusion

CGS-Mask is a model-agnostic saliency approach that explains time series prediction in an intuitive and user-friendly way. It performs well on synthesis and real-world data and outperforms state-of-the-art solutions. In the future, we aim to demonstrate its applicability in more time series applications, particularly in healthcare, for identifying salient features from medical records to reveal disease occurrence, development, and deterioration. While CGS-Mask's focus on local interpretability is intentional and aligns with our initial goals, we acknowledge that a global explanation is also valuable in some contexts. We are also actively working on enhancing CGS-Mask to incorporate global interpretability while maintaining its effectiveness at the local level.

Acknowledgments

This work was partially supported by China National Key R&D (Program No. 2021YFC2600203). We are also grateful to AMD product (China) Co., Ltd. and the Sugon Information Industry Co., Ltd. for their X785-g30 series GPU server. We would like to extend our gratitude to all the volunteers in this study. Their heartfelt support and active engagement is the cornerstone of ensuring the success of this study.

References

Adebayo, J.; Gilmer, J.; Muelly, M.; Goodfellow, I.; Hardt, M.; and Kim, B. 2018. Sanity checks for saliency maps. *Advances in neural information processing systems*, 31.

Albini, E.; Rago, A.; Baroni, P.; and Toni, F. 2020. Relation-Based Counterfactual Explanations for Bayesian Network Classifiers. In *IJCAI*, 451–457.

Bai, B.; Liang, J.; Zhang, G.; Li, H.; Bai, K.; and Wang, F. 2021. Why attentions may not be interpretable? In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 25–34.

Bedi, J.; and Toshniwal, D. 2019. Deep learning framework to forecast electricity demand. *Applied energy*, 238: 1312–1326.

Bian, C.; Feng, C.; Qian, C.; and Yu, Y. 2020. An efficient evolutionary algorithm for subset selection with general cost constraints. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 3267–3274.

Candanedo, L. M.; Feldheim, V.; and Deramaix, D. 2017. Data driven prediction models of energy use of appliances in a low-energy house. *Energy and buildings*, 140: 81–97.

Crabbé, J.; and Van Der Schaar, M. 2021. Explaining time series predictions with dynamic masks. In *International Conference on Machine Learning*, 2166–2177. PMLR.

Emille; Gautham, N.; Ranpal, G.; Renee, H.; Richard-Kessler; Sohier, D.; and Timo, B. 2018. PLAsTiCC Astronomical Classification.

Esteva, A.; Robicquet, A.; Ramsundar, B.; Kuleshov, V.; DePristo, M.; Chou, K.; Cui, C.; Corrado, G.; Thrun, S.; and Dean, J. 2019. A guide to deep learning in healthcare. *Nature medicine*, 25(1): 24–29.

Fisher, A.; Rudin, C.; and Dominici, F. 2019. All Models are Wrong, but Many are Useful: Learning a Variable's Importance by Studying an Entire Class of Prediction Models Simultaneously. *J. Mach. Learn. Res.*, 20(177): 1–81.

Fong, R.; Patrick, M.; and Vedaldi, A. 2019. Understanding deep networks via extremal perturbations and smooth masks. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2950–2958.

Fong, R. C.; and Vedaldi, A. 2017. Interpretable explanations of black boxes by meaningful perturbation. In *Proceedings of the IEEE international conference on computer vision*, 3429–3437.

Ghouaiel, N.; Marteau, P.-F.; and Dupont, M. 2017. Continuous pattern detection and recognition in stream-a benchmark for online gesture recognition. *International Journal of Applied Pattern Recognition*, 4(2).

Gomez, T.; Fréour, T.; and Mouchère, H. 2022. Metrics for saliency map evaluation of deep learning explanation methods. In *International Conference on Pattern Recognition and Artificial Intelligence*, 84–95. Springer.

Ismail, A. A.; Gunady, M.; Corrada Bravo, H.; and Feizi, S. 2020. Benchmarking deep learning interpretability in time series predictions. *Advances in neural information processing systems*, 33: 6441–6452.

Ivanovs, M.; Kadikis, R.; and Ozols, K. 2021. Perturbation-based methods for explaining deep neural networks: A survey. *Pattern Recognition Letters*, 150: 228–234.

Johnson, A. E.; Pollard, T. J.; Shen, L.; Lehman, L.-w. H.; Feng, M.; Ghassemi, M.; Moody, B.; Szolovits, P.; Anthony Celi, L.; and Mark, R. G. 2016. MIMIC-III, a freely accessible critical care database. *Scientific data*, 3(1): 1–9.

Kanamori, K.; Takagi, T.; Kobayashi, K.; and Arimura, H. 2020. DACE: Distribution-Aware Counterfactual Explanation by Mixed-Integer Linear Optimization. In *IJCAI*, 2855–2862.

Kwon, B. C.; Choi, M.-J.; Kim, J. T.; Choi, E.; Kim, Y. B.; Kwon, S.; Sun, J.; and Choo, J. 2018. Retainvis: Visual analytics with interpretable and interactive recurrent neural networks on electronic medical records. *IEEE transactions on visualization and computer graphics*, 25(1): 299–309.

Li, X.; Wu, J.; and Li, X. 2018. Cellular genetic algorithms. In *Theory of Practical Cellular Automaton*, 131–191. Springer.

Lu, F.; Li, W.; Sun, Y.; Song, C.; Ren, Y.; and Zomaya, A. Y. 2023a. CGS-Mask: Making Time Series Predictions Intuitive for All. arXiv:2312.09513.

Lu, F.; Li, W.; Zhou, Z.; Song, C.; Sun, Y.; Zhang, Y.; Ren, Y.; Liao, X.; Jin, H.; Luo, A.; et al. 2023b. A composite multi-attention framework for intraoperative hypotension early warning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 14374–14381.

Lundberg, S. M.; and Lee, S.-I. 2017. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30.

Lundberg, S. M.; Nair, B.; Vavilala, M. S.; Horibe, M.; Eisses, M. J.; Adams, T.; Liston, D. E.; Low, D. K.-W.; Newman, S.-F.; Kim, J.; et al. 2018. Explainable machine-learning predictions for the prevention of hypoxaemia during surgery. *Nature biomedical engineering*, 2(10): 749–760.

Mercier, D.; Bhatt, J.; Dengel, A.; and Ahmed, S. 2022. Time to Focus: A Comprehensive Benchmark Using Time Series Attribution Methods. *arXiv preprint arXiv:2202.03759*.

Moin, A.; Zhou, A.; Rahimi, A.; Menon, A.; Benatti, S.; Alexandrov, G.; Tamakloe, S.; Ting, J.; Yamamoto, N.; Khan, Y.; et al. 2021. A wearable biosensing system with in-sensor adaptive machine learning for hand gesture recognition. *Nature Electronics*, 4(1): 54–63.

Molnar, C. 2020. *Interpretable machine learning*. Lulu. com.

Natarajan, B. K. 1995. Sparse approximate solutions to linear systems. *SIAM journal on computing*, 24(2): 227–234.

Naul, B.; Bloom, J. S.; Pérez, F.; and van der Walt, S. 2018. A recurrent neural network for classification of unevenly sampled variable stars. *Nature Astronomy*, 2(2): 151–155.

Petsiuk, V.; Das, A.; and Saenko, K. 2018. RISE: Randomized Input Sampling for Explanation of Black-box Models. In *Proceedings of the British Machine Vision Conference (BMVC)*.

Qian, C.; Yu, Y.; and Zhou, Z.-H. 2015. Subset selection by Pareto optimization. *Advances in neural information processing systems*, 28.

Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2016. ” Why should i trust you?” Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 1135–1144.

Rostami, M.; Berahmand, K.; Nasiri, E.; and Forouzandeh, S. 2021. Review of swarm intelligence-based feature selection methods. *Engineering Applications of Artificial Intelligence*, 100: 104210.

Shrikumar, A.; Greenside, P.; and Kundaje, A. 2017. Learning important features through propagating activation differences. In *International conference on machine learning*, 3145–3153. PMLR.

Simonyan, K.; Vedaldi, A.; and Zisserman, A. 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*.

Sun, Y.; Song, C.; Lu, F.; Li, W.; Jin, H.; and Zomaya, A. Y. 2023. ES-Mask: evolutionary strip mask for explaining time series prediction (student abstract). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 16342–16343.

Sundararajan, M.; Taly, A.; and Yan, Q. 2017. Axiomatic attribution for deep networks. In *International conference on machine learning*, 3319–3328. PMLR.

Suresh, H.; Hunt, N.; Johnson, A.; Celi, L. A.; Szolovits, P.; and Ghassemi, M. 2017. Clinical intervention prediction and understanding using deep networks. *arXiv preprint arXiv:1705.08498*.

Tonekaboni, S.; Joshi, S.; Campbell, K.; Duvenaud, D. K.; and Goldenberg, A. 2020. What went wrong and when? Instance-wise feature importance for time-series black-box models. *Advances in Neural Information Processing Systems*, 33: 799–809.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wiegreffe, S.; and Pinter, Y. 2019. Attention is not not Explanation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 11–20.

Zhang, J.; Bargal, S. A.; Lin, Z.; Brandt, J.; Shen, X.; and Sclaroff, S. 2018. Top-down neural attention by excitation backprop. *International Journal of Computer Vision*, 126(10): 1084–1102.