Patterns
Continuous diagnosis and prognosis by controlling the update process of deep neural networks

Graphical abstract

Authors
Chenxi Sun, Hongyan Li, Moxian Song, Derun Cai, Baofeng Zhang, Shenda Hong

Correspondence
hongshenda@pku.edu.cn (S.H.), leehy@pku.edu.cn (H.L.)

In brief
Continuous diagnosis and prognosis are crucial for patients in intensive care. They can provide more opportunities for timely treatment and rational resource allocation. Based on the task of continuous classification of time series, a restricted update strategy for deep neural networks is proposed. It can not only achieve continuous diagnosis and prognosis but can also detect stages and biomarkers of the disease, making the deep-learning model interpretable.

Highlights
- Restricted update strategy for training deep-learning models is presented
- Method can be used for continuous classification of time series
- Method is effective in continuous diagnosis and prognosis in the medical field
- Method can find disease stages and biomarkers
Continuous diagnosis and prognosis by controlling the update process of deep neural networks

Chenxi Sun,1,2 Hongyan Li,1,2* Moxian Song,1,2 Derun Cai,1,2 Baofeng Zhang,1,2 and Shenda Hong3,4,5,*
1Key Laboratory of Machine Perception (Ministry of Education), Peking University, Beijing 100871, China
2School of Intelligence Science and Technology, Peking University, Beijing 100871, China
3National Institute of Health Data Science, Peking University, Beijing 100191, China
4Institute of Medical Technology, Health Science Center of Peking University, Beijing 100191, China
5Lead contact
*Correspondence: hongshenda@pku.edu.cn (S.H.), leehy@pku.edu.cn (H.L.)
https://doi.org/10.1016/j.patter.2023.100687

SUMMARY

Continuous diagnosis and prognosis are essential for critical patients. They can provide more opportunities for timely treatment and rational allocation. Although deep-learning techniques have demonstrated superiority in many medical tasks, they frequently forget, overfit, and produce results too late when performing continuous diagnosis and prognosis. In this work, we summarize the four requirements; propose a concept, continuous classification of time series (CCTS); and design a training method for deep learning, restricted update strategy (RU). The RU outperforms all baselines and achieves average accuracies of 90%, 97%, and 85% on continuous sepsis prognosis, COVID-19 mortality prediction, and eight disease classifications, respectively. The RU can also endow deep learning with interpretability, exploring disease mechanisms through staging and biomarker discovery. We find four sepsis stages, three COVID-19 stages, and their respective biomarkers. Further, our approach is data and model agnostic. It can be applied to other diseases and even in other fields.

INTRODUCTION

Continuous diagnosis and prognosis are of great significance for timely, personalized treatment and rational allocation of medical resources. Especially in the intensive care unit (ICU), status perception and disease diagnosis are needed at any time as real-time diagnosis provides more opportunities for doctors to rescue lives. For example, sepsis is a life-threatening condition, causing more than half of ICU deaths.1 Early detection and antibiotic treatment are critical for improving sepsis outcomes;2,3 COVID-19 outbreaks have caused health concerns worldwide. In the case of a sudden outbreak of a new epidemic, continuous...
prognosis can help with personalized treatment and rational allocation of scarce resources.\textsuperscript{4,5}

Different from the single-shot diagnosis, which is often made for the outpatient, the task of continuous diagnosis and prognosis emphasizes the multiple early diagnoses or prognoses for the inpatient at different stages over time. For example, in Figure 1, a patient in the ICU is monitored for vital signs in real time. Assuming that he will be in sepsis shock at 17:00, the common diagnostic system will give a warning when he is suffering or about to suffer from sepsis at about 17:00 (the single-shot diagnosis, blue dot). This is likely to miss the emergency treatment time for the acute disease, where each hour of delay has been associated with roughly a 4%–8% increase in sepsis mortality.\textsuperscript{2} Thus, we require a continuous prognosis mode for sepsis (red stars), where we can predict the patient outcome 4 h early, 1 h early, etc., at 13:00, 16:00, etc. To meet this practical need, we summarize four requirements for the task of continuous diagnosis and prognosis.

\textbf{Requirement 1: The ability to identify symptoms in different time stages before the disease onset}

Single-shot diagnosis only needs learning the clinical manifestation, which is easy under the guidance of the gold standard. But continuous prognosis needs to learn the underlying symptoms of the disease in its early stages, which are usually not obvious in the clinic and cannot be judged by clinicians, and the symptoms are not only from a certain stage but from multiple stages before the onset, leading to diversity and hybridity.

\textbf{Requirement 2: Potential for earlier diagnosis with guaranteed accuracy}

Earlier diagnosis is necessary for many severe illnesses. Each hour of delayed treatment could cause a 4%–8% increase in sepsis mortality,\textsuperscript{2} but basic questions about the limits of early detection remain unanswered. If one wants to pursue higher diagnostic accuracy, it will tend to predict late for clearer features. For example, a rapid drop in blood pressure (a major symptom of sepsis shock; the red dashed box in Figure 1) always occurs just before the shock.\textsuperscript{1} But we expect the continuous mode to diagnose earlier and more accurately than the single-shot mode.

\textbf{Requirement 3: Merits of explainability and disease staging}

The 22nd article of the European Union’s General Data Protection Regulation stipulates that a subject of algorithmic decisions has a right to meaningful explanation regarding said decisions.\textsuperscript{6} As clinicians always justify a result using medical-domain knowledge familiar to them, the explainable methods will be more popular in practice.\textsuperscript{7} Meanwhile, continuous prognosis is accompanied by disease progression. Disease staging is important to understand disease mechanisms and implement targeted treatment. A clinically useful staging system stratifies patients by their baseline risk of an adverse outcome and their potential to respond to therapy. The best developed and most explicit approach has evolved in oncology,\textsuperscript{9} but it is not clear for critical illnesses. Recently, the stratification for sepsis (sepsis, severe sepsis, and septic shock) has been questioned in the latest sepsis definition,\textsuperscript{1} and there are no criteria for temporal septic stages.

\textbf{Requirement 4: Function of offline and sustainable use}

In many scenarios, especially in the ICU, we need to directly use the mature system without constant adjustment. A well-informed system can reduce the risk of misjudgment.\textsuperscript{9} Further, in subsequent applications, when obtaining a batch of new data, such as new patients and new clinical observations, we hope to continue to use the current system instead of designing a new one because the data that have the old knowledge may still occur, while the new system cannot handle the old knowledge well.\textsuperscript{10}

Nowadays, many studies have shown that deep learning (DL) methods are superior to medical gold standards and experienced doctors in some medical tasks such as outcome prediction and disease diagnosis.\textsuperscript{11,12} Surprisingly, in these studies, sequential medical records, such as vital signs, multiple blood
samples, and serial medical imaging, provided more possibilities for DL models to implement diagnosis and prognosis. We uniformly name such sequential medical records as medical time series data. However, most DL-based models often give the single-shot diagnosis after learning the full-length medical time series but cannot prognose continuously. Although some sub-disciplines study the mode of continuous learning, they cannot satisfy the above requirements at the same time (see supplemental related work and concepts).

The labels (mortality, morbidity, etc.) of real-world medical time series are usually determined at the final time. If the model simply learns the full-length time series, it can only give the single-shot result at the onset time. For continuous diagnosis and prognosis, the model needs to learn time series from different advanced stages: when the data change, the model performance needs to maintain. But most medical time series have evolved distribution. As shown in Figure S4, learning the new knowledge may lead to forgetting old ones, and learning one distribution frequently may fall into local solutions with overfitting.

Meanwhile, interpretability is an elusive concept, and the artificial intelligence (AI) field holds no consensus regarding its definition. Although some studies have proposed methods to explain the DL black-box model, they are not fully interpretable, mostly explaining static models and depending on the actual scenario and task. Thus, interpretation problems have not yet been fully overcome when using DL for medical applications. When developing the method, we need to consider the possibility of it being explained and match it to the dynamic process of continuous diagnosis and prognosis.

To this end, we establish a training method for DL models, the restricted update strategy (RU) of neural network parameters. The RU can satisfy the above requirements: for requirement 1, it has the limitation mechanism (LM) to avoid catastrophic forgetting and overfitting; for requirement 2, it has the promotion mechanism (PM) to consolidate the knowledge of early distribution; for requirement 3, we define the importance coefficient of parameters to reveal the model development and achieve disease staging with typical biomarkers; and for requirement 4, we train the model by real-world datasets with separate training and test sets and test continual use. Experimental results show that the RU is more accurate than all baselines, achieving accuracies of 90%, 97%, and 85% on sepsis prognosis, COVID-19 mortality prediction, and eight disease diagnoses, respectively.

RESULTS

We test the RU and eight baselines on six datasets, using 5-fold cross-validation, expressed as the mean and SD (mean ± SD). The classification accuracy is evaluated by the area under the curve of the receiver operating characteristic (AUC-ROC; the higher the better) and the AUC confidence interval. The continuous classification performance is evaluated by backward transfer (BTW) and forward transfer (FWT; the higher the better). The statistical significance is evaluated by the Bonferroni-Dunn test. The learning stability is evaluated by the gradient fluctuation R (the lower the better). Eight baselines are long short-term memory (LSTM), stopping rule (SR), effective confidence-based early classification (ECEC), online stochastic recursive gradient-based Frank-Wolfe (ORGFW), gradient episodic memory (GEM), elastic weight consolidation (EWC), continual learning with experience and replay (CLEAR), and continual learning of physiological signals (CLOS).

Three medical datasets are SEPSIS, COVID-19, and MIMIC-III. Three additional datasets are United States Historical Climatology Network (USHCN), University of California Riverside time series classification archive (UCR), and human ACTIVity dataset (ACTIV) (more experimental details are in the supplemental experimental procedures).

An RU to train DL models for continuous diagnosis and prognosis

For continuous medical diagnosis and prognosis, we focus on continuous sepsis prognosis, continuous COVID-19 mortality prediction, and continuous eight disease classification based on medical time series, including vital signs from various monitors, and continuous blood sample records during hospitalization. All used data are available: the SEPSIS dataset has 30,336 ICU patient records with 2,359 diagnosed with sepsis from three separate hospital systems; the COVID-19 dataset has 6,877 blood sample records of 485 COVID-19 patients from Tongji Hospital, Wuhan, China; and the MIMIC-III dataset has 19,993 admission records from 7,537 patients, and we focus on 8 diseases.

A time series dataset \( T = \{X^d_n\}_{n=1}^N \) has \( N \) samples. Each sample \( X = \{x^m_{nm}\}_{m=1}^M \) has \( M \) observations with value \( x^m_n \) and time \( t_m \). Multivariate time series can be described by \( X = \{X^d_n\}_{d=1}^D \), \( d \) is the d-th dimension. DL models have achieved great success in modeling medical time series data, especially recurrent neural networks (RNNs). However, the real-world time series is usually long and irregularly sampled. For example, critically ill patients are often hospitalized for several months; thus, records often have hundreds of observations. And due to the change in the patient's health status, the relevant measurement requirements are also changing, which may be several hours or days apart. Thus, in order to model the long-term dependency and eliminate the impact of uneven time intervals, we implement time-aware
LSTM (T-LSTM), a variant of the RNN. As shown in Figure 2A, our DL architecture has two blocks: block 1 uses T-LSTM to model the input data and represent their hidden features with the consideration of time decay $\Delta m = t_m - t_{m-1}$, and block 2 uses multilayer perceptron (MLP) to map features to the class.

After training T-LSTM by the full-length time series dataset, it can achieve average accuracies of 92%, 97%, and 88% on single-shot diagnosis for sepsis, COVID-19, and eight diseases, respectively. However, when applying it to continuous diagnosis and prognosis, the accuracy drops by more than 15%. Thus, we find the continuous mode has more potential in medical diagnosis and prognosis than the single-shot mode.

Finding 1: The continuous mode has more potential in medical diagnosis and prognosis than the single-shot mode

For the CCTS task, as shown in Figures 3G–3L, our method, RU, can classify more accurately at every time. It is significantly better than all 8 baselines in the Bonferroni-Dunn test ($7 = 3.5 > CD = 2.724$). The average accuracy is about 2% higher, especially in the early time, being 5% higher for 10%-length data. CCTS is important for time-sensitive applications, especially for acute and critical illnesses. Take sepsis diagnosis as an example: compared with the best baseline, the RU improves the accuracy by 1.4% on average and 2.2% in the early 50% time stage where the key features are unobvious. Each hour of delayed treatment increases mortality by 4%–8%. With the same accuracy, we can predict 0.972 h in advance.

The RU can alleviate catastrophic forgetting and overfitting when classifying time series continuously. Figure S3 shows that all three medical datasets have multiple distributions in the time stage where the key features are unobvious. Each hour of delayed treatment increases mortality by 4%–8%. With the same accuracy, we can predict 0.972 h in advance. The importance of parameter $\theta$ is measured by the importance coefficients $\alpha(\theta)$. LM is implemented by using a loss in Equation 5. Its key feature is the use of the diagonal of the Fisher information matrix $F$ to represent the importance coefficient $\alpha$, allowing for the quantification of parameter importance.

PM helps the DL model to classify time series earlier in time-sensitive applications. It regards the process of a DL model learning early distributions and new distributions as the same continuous optimization problem with regret minimization. PM is projection free and estimates a stochastic recursive estimator to alleviate the complexity and training instability. As shown in Figure 2B, when learning distribution $D_m$, PM changes the current gradient from an obtuse angle to an acute angle with the gradient on previous distribution $D_{m-1}$ because when the new gradient and the old gradient are at an acute angle, the model performance on the old distribution will improve, or at least not decrease. Most importantly, the promotion of learning new distributions has the potential for early classification. PM is implemented by a recursive estimator in Equation 12. It can fill the gap between the optimal regret bound and the low per-round computational cost, holding a nearly optimal regret bound $O(\sqrt{M})$, where $M$ is the number of distributions.

Equations 5 and 12 serve as the main conduits for implementing the RU. It is clear that they have no bearing on the model’s structure or tasks. Thus, the RU is a model-agnostic, task-agnostic, and easy-to-use plug in.
Finding 2: The change of importance coefficients interprets the learning process of the DL model

When a DL model learns time series in different stages, its parameters are updated constantly. After using the RU, if the model encounters a new data distribution, the importance distribution is likely to change significantly. Thus, we can explain the learning process of the DL model from the perspective of the change of importance coefficient.

We divide the DL model into three blocks as shown in Figure 4. Block 1 is the input block. We focus on the parameter update process related to input features. For an input feature \( x^d \) (d-th dimension sequence of input multivariate time series), we use the overall importance coefficient of its related parameters to measure its importance: \( \alpha'(x^d) = \sum \alpha(\theta_{\text{input},d}) \), where \( \theta_{\text{input},d} \) is the weight between input feature \( x^d \) and the n-th neuron in layer \( l_t \). Block 2 is the T-LSTM block. We focus on the parameter update process related to different gates. For a gate \( G \), we use the overall importance coefficient of its parameters to measure its importance: \( \alpha'(G) = \sum \alpha(\theta_{\text{gate}}) \). Block 3 is the output block. We focus on the parameter update process related to network neurons. For the j-th neuron in layer \( l_t \), we use the overall importance coefficient of its output weights to measure its importance: \( \alpha'(G_j) = \sum \alpha(\theta_{\text{output},j}) \). The test on three blocks can enhance the interpretability based on the input data and network structures.

When the model learns time series with different lengths (in different time stages), its perceptual sensitivity to input features is different. As shown in Figures 4A1 and 4B1, for sepsis diagnosis, the blood pressure’s \( \alpha' \) increases, which means that the model’s perception of blood pressure improved in the later stage. For COVID-19 mortality prediction, lymphocytes’ \( \alpha' \) is always high, which means that the model pays attention to this feature at all stages. Thus, \( \alpha' \) can be used to evaluate biomarkers.

We regard the importance coefficient of model parameters in the continuous learning process as a sequence and use the Bayesian online changepoint detection (BOCD) model \(^{34} \) to find change points in this sequence. N change points divide the sequence into \( n + 1 \) stages. As shown in Figures 4A2 and 4B2, in the output block, the training process of the model can be roughly divided into four stages for sepsis diagnosis and three stages for COVID-19 mortality prediction. In each stage, the important parameters are different. In the T-LSTM block, this change is obvious for the output gate but not obvious for the input and output gates. These observations reveal the intrinsic mechanism of model learning under the RU: for different stages of time series (different distributions), the DL model activates different neurons to perceive data. This also shows the potential of wide neural networks for \( \text{CCTS} \). Networks with more neurons in one layer are more likely to learn multidistributed data.

Finding 3: Continuous prognosis reveals the disease biomarkers and stages

Semantically, the important feature is the input that has a great impact on the classification results. To quantify them, we define that the important feature is the input with a large overall importance coefficient \( \alpha' \). Thus, we can find biomarkers of specific diseases: as shown in Figure 4A1 and 4B1, for sepsis, the biomarkers are heart rate (HR), respiration (Resp), mean arterial pressure (MAP), PaCO2, platelets count, total bilirubin, and creatinine. For COVID-19, the biomarkers are lymphocytes (lymph), lactic dehydrogenase (LDH), high-sensitivity C-reactive protein (hs-CRP), indirect bilirubin, creatinine, etc.

The response change of the model when learning disease records continuously can reflect the disease development. After finding change points by BOCD, the training process of the model can be divided into four stages for sepsis diagnosis and three stages for COVID-19 mortality prediction. Then, as shown in Figure 5, we visualize the hidden layer of block 3 and show the statistics of the corresponding characteristics.

Sepsis has four disease stages. Each stage has different reference levels of biomarkers. In some cases, the closer to the onset time, the greater the difference in biomarker reference levels among different prognoses. For example, in stage one (the interval from early 48 h to early 40 h before the onset time), the Resp difference between sepsis and non-sepsis is 1, while in stage four (the interval from early 6 h to the onset time), the Resp difference is 7. In other cases, reference levels of biomarkers with different prognoses are different at all stages, such as creatinine. These two conditions may explain the two mechanisms of sepsis: (1) the acute sepsis onset is reflected in the changes of some specific vital signs, e.g., a drop in blood pressure, increased lactate, and tachycardia. (2) Patients who have some congenital characteristics are more likely to get sepsis, e.g., nephropathy with abnormal creatinine and hepatopathy with increased total bilirubin.

COVID-19 has three disease stages. Compared with sepsis, two classes can be distinguished more clearly in the representation space. This also explains the higher accuracy of continuous COVID-19 mortality prediction than that of continuous sepsis diagnosis. Besides, in the presentation space, the hidden features of the two classes in the later stage are further apart. This shows the difficulty in early classification: the conflict between earliness and accuracy.
A Continuous Sepsis Prognosis

1. The Importance of Input Feature and Network Parameters

2. The Importance of Input Feature and Network Parameters

B Continuous COVID-19 Mortality Prediction

1. The Importance of Input Feature and Network Parameters

2. The Importance of Input Feature and Network Parameters

Figure 4. The importance of input features and network parameters
A Sepsis Disease Staging

1. Sepsis sample representation without RU

- Representation change of a sepsis sample
- Sepsis in Stage 1
- Sepsis in Stage 2
- Sepsis in Stage 3
- Sepsis in Stage 4
- Non-sepsis in Stage 1
- Non-sepsis in Stage 2
- Non-sepsis in Stage 3
- Non-sepsis in Stage 4

Optimal number of clusters (stages) without RU

Optimal number of clusters (stages) with RU

2. Sepsis sample representation with RU

Segnss Stages (Sepsis/Non-sepsis)

- Mean value
- Stage 1: 8
- Stage 2: 22
- Stage 3: 12
- Stage 4: 6
- Within time before onset (hours)
- Stage 1: 48
- Stage 2: 40
- Stage 3: 18
- Stage 4: 6

B COVID-19 Disease Staging

1. Death sample representation without RU

- Representation change of a sepsis sample
- Death in Stage 1
- Death in Stage 2
- Death in Stage 3
- Death in Stage 4
- Survival in Stage 1
- Survival in Stage 2
- Survival in Stage 3
- Survival in Stage 4

Rationality of clustering (staging) without RU

Rationality of clustering (staging) with RU

2. Death sample representation with RU

COVID-19 Stages (Death/Survival)

- Mean value
- Stage 1: 1.6
- Stage 2: 8.9
- Stage 3: 2.1
- Stage 4: 8.5
- Mortality rate (%)
- Stage 1: 6.3
- Stage 2: 7.7
- Stage 3: 9.0
- Stage 4: 2.9
- In-CRP levels
- Stage 1: 15516
- Stage 2: 10519
- Stage 3: 431
- LDH (U/L)
- Stage 1: 338328
- Stage 2: 370273
- Stage 3: 499198
- Platelets (count x 10^3/μL)
- Stage 1: 245.5
- Stage 2: 224.5
- Stage 3: 205.1
- Stage 4: 204.3
- Creatinine (mg/dL)
- Stage 1: 2.4
- Stage 2: 3.0
- Stage 3: 3.3
- Stage 4: 3.5

Figure 5. Disease stages and biomarkers of sepsis and COVID-19
The classification of eight diseases has four stages as shown in Figure S8. Since the task is not specific to a disease, we call task stages and important features instead of disease stages and biomarkers. The important features found in the four stages illustrate the necessity of continuous vital sign monitoring, blood routine examination during hospitalization, and detailed laboratory examination in later stages. Meanwhile, there is a big difference between the last three stages and the first stage. It implies that patients’ states have changed significantly since the second stage.

**Finding 4: RU enhances the model for atypical scenarios and sustainable use**

The RU can avoid model overfitting and guarantee certain model generalization. As shown in Tables 1, S10, and S11, we divide datasets according to gender and age; for most baselines, the accuracy on the validation set is much lower than that on the training set, but the RU helps the DL model to maintain robustness.

Meanwhile, the RU can prevent the result difference caused by the different orders of training sets. The method we have introduced is to use time series of different stages to train the model, and the order is based on time. Another order is the data similarity as shown in Figure S6. For example, as many vital signs are periodic, the cycle of blood pressure is 1 day. Therefore, after using vital signs within 24 h, we will use the data within 25 h according to the time order but use the data within 48 h according to the similarity order. No matter what order is adopted, the RU has a stable accuracy as shown in Tables 2, S8, and S9. It shows the potential of PM’s global optimization and the potential of the RU’s sustainable use.

Furthermore, the RU is a data-agnostic, model-agnostic, and easy-to-use plug in. It can not only improve the accuracy of continuous classification of medical time series but also plays a role in other fields. For example, as shown in Table 3, the RU outperforms baselines on meteorological data for tasks of continuous earthquake early warning and rainfall prediction. Figures S2–S6 show the ability of the RU in more scenarios. The RU can also be used to train other DL models such as the convolutional neural network (CNN) and transformer. It is easy to use and does not need to change the network structure. As shown in Table 4, if we use the RU to train base models, the accuracy can be improved by more than 5%, and the RU is not limited by hyper-parameters. The hyper-parameters are ρ and λ, ρ determines the correlation between current and previous gradients in Equation 12. We find that PM performs well when ρ is the same as the learning rate ρm = ηm = 676100687.2, = 1, λ decides the constraint degree on parameter update in Equation 5. We can optimize it using the search method supplied by mature tools.

**DISCUSSION**

DL has the potential to explore disease mechanisms

The importance coefficient not only explained the working mechanism of the DL model but also dug out the disease biomarkers and stages. Different from the statistics and case analysis of the medical gold standard, these biomarkers are based on the judgment basis of the DL model. It can provide a new horizon for medical research. For example, based on the learning process of the DL model, for sepsis, a drop in...
The accuracy is significantly improved after using the RU (p < 0.05), specifically \( p = 0.0040, 0.0000, \) and 0.0015. 

\( ^a \)The accuracy is increased by more than 5%, and the BWT is increased by more than 5%. 

### Table 3: Performance (AUC-ROC \( \downarrow \), BWT \( \downarrow \)) for two meteorological datasets

| Dataset | SR \( ^a \) | ECEC \( ^a \) | ORGFW \( ^a \) | GEM \( ^a \) | CLOPS \( ^a \) | RU |
|---------|---------------|---------------|---------------|---------------|----------------|----------|
| UCR-EQ  | AUC-ROC       | 0.902 ± 0.002 | 0.909 ± 0.010 | 0.920 ± 0.001 | 0.921 ± 0.001 | 0.919 ± 0.004 | 0.931 ± 0.004 |
|         | BWT           | 0.003         | 0.033         | 0.112         | 0.123         | 0.149       | 0.162    |
| USHCN   | AUC-ROC       | 0.911 ± 0.012 | 0.902 ± 0.012 | 0.916 ± 0.004 | 0.920 ± 0.003 | 0.921 ± 0.005 | 0.930 ± 0.005 |
|         | BWT           | 0.034         | 0.047         | 0.072         | 0.098         | 0.082       | 0.124    |

\( ^a \)The UCR-EQ dataset \( ^{28} \) has 471 earthquake records from the UCR time series classification archive. It is the univariate time series of seismic feature value. Natural disaster early warning, like earthquake warning, helps to reduce casualties and property losses. \( ^{37} \) The USHCN dataset \( ^{27} \) has the daily meteorological data of 48 states in the US from 1887 to 2014. It is the multivariate time series of five weather features. Rainfall warning is not only a demand of daily life but can also help prevent natural disasters. \( ^{38} \) In the Bonferroni-Dunn test, \( k = 6, n = 2, m = 5, q_{0.05} = 2.676, N = n \times m = 10, CD = 2.16, r = 4.00 > CD. \) RU is significantly better, specifically, better than baselines with \( p = 0.00004, 0.00241, 0.0099, \) 0.0137, and 0.0329 (p < 0.05). 

### Table 4: Performance (AUC-ROC \( \downarrow \), BWT \( \downarrow \)) improvement of different neural networks after using RU

| Model     | SEPSIS       | COVID-19     | MIMIC-III    | UCR-EQ       | USHCN       |
|-----------|--------------|--------------|--------------|--------------|-------------|
| LSTM \( ^a \) | 0.837 ± 0.008/0.002 | 0.909 ± 0.003/0.047 | 0.786 ± 0.002/0.054 | 0.881 ± 0.004/0.032 | 0.891 ± 0.003/0.054 |
| +RU       | 0.907 ± 0.009/0.065 \( ^{29} \) | 0.969 ± 0.003/0.115 \( ^{29} \) | 0.856 ± 0.002/0.102 | 0.931 ± 0.004/0.162 \( ^{30} \) | 0.930 ± 0.005/0.124 |
| CNN \( ^a \) | 0.848 ± 0.002/0.004 | 0.903 ± 0.002/0.037 | 0.784 ± 0.004/0.032 | 0.878 ± 0.005/0.030 | 0.881 ± 0.004/0.057 |
| +RU       | 0.904 ± 0.003/0.067 \( ^{29} \) | 0.960 ± 0.006/0.095 \( ^{29} \) | 0.832 ± 0.002/0.099 | 0.929 ± 0.006/0.150 \( ^{30} \) | 0.922 ± 0.005/0.118 |
| Transformer \( ^a \) | 0.843 ± 0.011/0.005 | 0.906 ± 0.005/0.040 | 0.784 ± 0.006/0.059 | 0.889 ± 0.010/0.029 | 0.880 ± 0.015/0.059 |
| +RU       | 0.903 ± 0.008/0.067 \( ^{29} \) | 0.960 ± 0.007/0.109 \( ^{29} \) | 0.852 ± 0.008/0.124 \( ^{29} \) | 0.920 ± 0.008/0.132 | 0.921 ± 0.008/0.120 |

\( ^{29} \) The accuracy is significantly improved after using the RU (p < 0.05), specifically \( p = 0.0040, 0.0000, \) and 0.0015. 

\( ^{29} \) The accuracy is increased by more than 5%, and the BWT is increased by more than 5%.

### RU helps with disease staging

At present, except for cancers, it is difficult to define clear stages for most diseases. For sepsis, disease stratification is implemented by recommended clinical criteria (e.g., systemic inflammatory response syndrome [SIRS], sequential organ failure assessment [SOFA], quick SOFA [qSOFA], etc.), but they focus on severity and not the progression. We emphasize that disease staging is the disease change over time. The RU can achieve this according to the model change when learning the medical time series from different time stages. As shown in Figure 5A, the RU can identify stages directly according to the importance coefficient change of model parameters, instead of using unsupervised clustering methods. Without the RU, the clustering method has trouble finding the stages. The number of clusters with the best silhouette coefficient is 2, and the number of stages is 1.

For COVID-19, most work categorizes it roughly into early stage and late stage. \( ^{42} \) Some existing DL-based methods can perform disease staging by using representation learning. For example, our previous work \( ^{32} \) clustered features in hidden layers of T-LSTM and got four COVID-19 stages. As shown in Figure 5B1, this clustering-based method can get a good silhouette coefficient but cannot guarantee the time constraint. When identifying these four stages, only about 40% of the samples will be divided into stages corresponding to chronological order. For example, a death sample (green dot) is initially judged to be stage two, then stage one, and finally stage four. But stages 1–4 are in time order. Using the RU, this inconsistency is largely alleviated: the percentage of samples with a time-increasing stage is raised. The death sample (green dot) is judged as stage one, then stage two, and finally stage three over time.

### Learning multidistributed data is the general trend

Currently, many sophisticated DL models have shown outstanding achievements in time series modeling in many fields. For offline learning, after the model has learned the dataset, the model is only sensitive to the learned distribution. For example, when the model has learned the full-length vital signs of sepsis, it usually classifies accurately at the onset time, but it is too late for critical illness. To gain treatment time, the model needs to learn early data. However, there are also problems in learning early data at only one stage. For example, a time series may have missed the learned stage for most diseases. For sepsis, disease stratification is implemented by recommended clinical criteria (e.g., systemic inflammatory response syndrome [SIRS], sequential organ failure assessment [SOFA], quick SOFA [qSOFA], etc.), but they focus on severity and not the progression. We emphasize that disease staging is the disease change over time. The RU can achieve this according to the model change when learning the medical time series from different time stages. As shown in Figure 5A, the RU can identify stages directly according to the importance coefficient change of model parameters, instead of using unsupervised clustering methods. Without the RU, the clustering method has trouble finding the stages. The number of clusters with the best silhouette coefficient is 2, and the number of stages is 1.

For COVID-19, most work categorizes it roughly into early stage and late stage. \( ^{42} \) Some existing DL-based methods can perform disease staging by using representation learning. For example, our previous work \( ^{32} \) clustered features in hidden layers of T-LSTM and got four COVID-19 stages. As shown in Figure 5B1, this clustering-based method can get a good silhouette coefficient but cannot guarantee the time constraint. When identifying these four stages, only about 40% of the samples will be divided into stages corresponding to chronological order. For example, a death sample (green dot) is initially judged to be stage two, then stage one, and finally stage four. But stages 1–4 are in time order. Using the RU, this inconsistency is largely alleviated: the percentage of samples with a time-increasing stage is raised. The death sample (green dot) is judged as stage one, then stage two, and finally stage three over time.
stages, i.e., multidistributed data. In this way, the model can realize CCTS.

**Reasonable training strategy is the icing on the cake**

We help the DL model to learn multidistributed data from the perspective of model updating strategy. Our empirical study shows that a meaningful model training strategy plays a key role in improving the model performance and generalization power. Compared with the study of model structure design, the application of strategy-based design is more extensive. It pays more attention to the overall goal and has few requirements for specific data and used models, meaning it is data agnostic, model agnostic, and easy to use.

**Quantifying the updating process makes it possible to interpret the DL model**

Interpretability remains one of the key issues to be solved to achieve the trust of clinicians and insert the DL algorithm into clinical workflow. DL models are often considered to be black box because they typically have high-dimensional non-linear operations, many model parameters, and complex model architectures, which makes them difficult for a human to understand. In this work, the RU implements CCTS by updating model parameters with constraints. The constraint is achieved by quantifying the importance of the parameters. Surprisingly, the importance of parameters can be used to explain the DL model. As shown in experimental results, it can identify both the input features and the structural parts that are important for classification. Figure S7 shows the important samples in continuous binary classification. In fact, measuring parameter changes is not only suitable for the dynamic process in CCTS but also provides the possibility to interpret the learning processes of DL models.

**Opportunities of CCTS**

Currently, some sub-disciplines (online learning, continual learning, anomaly detection) also study the mode of continuous classification, but their setting methods cannot satisfy the summarized requirements simultaneously (see related work in the supplemental information). CCTS is a new concept and a potential task that we propose when facing practical problems.

Meanwhile, we find that the neural network structure with a reasonable width is more conducive to continuous classification and continual learning because with the learning of the new data distribution, the change of important parameters on the scale of network width is more obvious and regular but that on the scale of depth is confused. Therefore, future work can study the impact of model structure on CCTS from the perspective of network depth and network width.

Besides, the imbalanced data are very common in many real-world datasets. For example, in electronic health records, the records of common diseases are much more than those of rare diseases. And for classification task, the lack of data in a minority class may lead to uneven accuracy. When encountering such data, although not involved in this study, we can change the cross-entropy loss \( \mathcal{L} \) in Equation 5 to other specific losses such as weighted cross-entropy, mean false error loss, and focal loss. Further, we find that different learning orders have little effect on our method. This demonstrates the potential of our approach for offline continual learning: We use the existing data to train the model and put it into use. After a period of time, some new data may be generated. We can continue to train the current model with the new data instead of designing a new model. In addition, the method for CCTS can be context independent in the future. The model can perform not only different medical tasks but also tasks in other fields like meteorology.

**EXPERIMENTAL PROCEDURES**

**Resource availability**

**Lead contact**

Request for information and resources used in this article should be addressed to Prof. Shendra Hong (nongshta@pku.edu.cn).

**Materials availability**

No new materials were generated by this study.

**Data and code availability**

All datasets are publicly available. SEPSIS: https://doi.org/10.13026/v64v-d857, COVID-19: https://doi.org/10.13026/C2XW26, USHCN: NCDC (DSI-3206, DSI-3208, DSI-3210), UCR https://www.cs.ucr.edu/~eamonn/time_series_data_2018/, and ACTIV: https://archive.ics.uci.edu/ml/datasets/Localization+Data+for+Person+Activity. All original code has been deposited at Zenodo under https://doi.org/10.5281/zenodo.7496021 (https://github.com/SCXsunchenii/CCTS) and is publicly available as of the date of publication. Any additional information required to reanalyze the data reported in this paper is available from the lead contact upon request.

**Methods**

**CCTS**

**Definition 1 (CCTS).** A dataset \( T = \{X^n\}^N_{n=1} \) contains \( N \) time series. Each time series \( X = \{x_k\}^M_{k=1} \) has \( M \) observations with value \( x_k \) at time \( t_k \). At the final time \( t_M \), \( X \) is labeled with a class \( c \in C \). As time series vary among time, dataset \( T' = \{X'^m\}^M_{m=1} \), consisting of subsequence \( X_{1_m} \) of each \( X \), has a distribution \( Pm \). Thus, \( T'' = \{X''_{1_m} \}^M_{m=1} \) has a series distribution \( D = \{pm\}^M_{m=1} \). CCTS learns every \( Pm \) and introduces a task series \( M = \{M^m\}^M_{m=1} \) to minimize the additive risk \( \sum_m = \sum(X'[Pm(\theta), C]) \) with model \( \theta \), \( Pm \) is the model \( \theta \) after being trained for \( M^m \), and its performance on all observed data cannot degrade: \( \sum_m = \sum(X'[Pm(\theta), C]) \leq \sum_m = \sum(X'[pm(\theta), C]) \).

Notations are summarized in Table S1. Without the loss of generality, we use the univariate time series to present the problem. Multivariate time series can be described by changing \( x_k \) to \( x_k \), \( d \) is the \( d \)-th dimension. The classification task uses cross-entropy loss:

\[
\mathcal{L}(f_m, M^m) = \mathcal{L}(f_m, \{T^m, C^m\}) = -\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{C} C_i \log \hat{C}_i \quad \text{(Equation 1)}
\]

**T-LSTM**

The real-world time series, especially vital signs, have long sequences and are irregularly sampled. The classical RNN only processes uniformly distributed longitudinal data by assuming that the sequences have an equal distribution of time differences. Thus, we implement T-LSTM.\[\text{Equation 3}]\]

\[
\begin{align*}
C_{m-1}^{\text{in}} &= \tanh(W_{c} c_{m-1} + d_{b}) \quad \text{short} & \text{ term memory} \\
C_{m-1}^{\text{out}} &= \text{tanh} \left( C_{m-1}^{\text{in}} + (\Delta^m) \right) \quad \text{discounted short} & \text{ term memory} \\
C_{m-1}^{\text{long}} &= C_{m-1}^{\text{in}} - C_{m-1}^{\text{out}} \quad \text{long} & \text{ term memory} \\
C_{m-1}^{\text{adj}} &= C_{m}^{\text{in}} - C_{m-1}^{\text{adj}} \quad \text{adjusted previous memory} \\
f_{m} &= \sigma(W_{f} x_{m} + U_{f} h_{m-1} + b_{f}) \quad \text{forget gate} \\
f_{m} &= \sigma(W_{f} x_{m} + U_{f} h_{m-1} + b_{f}) \quad \text{input gate} \\
C_{m}^{\text{in}} &= \tanh(W_{x} x_{m} + U_{h} h_{m-1} + b_{x}) \quad \text{candidate memory} \\
C_{m} &= f_{m} c_{m-1} + \imath_{m} C_{m} + \text{out}_{m} \quad \text{current memory} \\
o_{m} &= \sigma(W_{o} x_{m} + U_{o} h_{m-1} + b_{o}) \quad \text{output gate} \\
h_{m} &= o_{m} \tanh(C_{m}) \quad \text{current hidden state}
\end{align*}
\]

Patterns 4, 100687, February 10, 2023 11
Algorithm 1. RU

| Step | Description |
|------|-------------|
| 1    | Initialize a model \( P \) with parameter \( \theta^0 \) |
| 2    | Initialize a gradient memory \( \theta^* \) to \( \emptyset \) |
| 3    | Initialize parameters \( \eta^m, \theta^m = (1/m) \) |
| 4    | for \( m = 1 \) to \( |\mathcal{M}| \) do |
| 5    | Extract current task \( \mathcal{M}^m \rightarrow \mathcal{M} \) |
| 6    | Get loss \( \mathcal{O}^m \rightarrow \text{Equation 5} \) Limitation Mechanism |
| 7    | \( g_{m-1} \rightarrow \theta \) |
| 8    | \( d^m \rightarrow \text{Equation 12} \) |
| 9    | \( v^m \rightarrow \text{Equation 8} \) |
| 10   | \( \theta^m, \theta^m \rightarrow \text{Equation 9} \) Promotion Mechanism |
| 11   | \( \theta^* \rightarrow \theta^m \) |
| 12   | Get model \( m \rightarrow \theta^m \) |
| 13   | end for |
| 14   | Output model \( f \rightarrow f(\mathcal{M}) \) |

T-LSTM has some new designs. The \( C^0 \), component learns the short-term memory of a sequence by learnable network parameters. \( C^0 \) is the long-term memory calculated from the former memory cell \( C_0 \) by getting rid of \( C^0 \). The new data helps the model learn the old data, which can reduce the unstable solution caused by the different learning orders. The continuous optimization problem is defined as regret minimization. For task \( \mathcal{M} \), the regret \( R \) is the difference between the total loss and that of the best parameter \( \theta^* \) of the fixed decision in hindsight.

\[
\mathcal{O}(\theta^m) = \mathcal{L}(\theta^m(\theta^m), \mathcal{M}^m) + 1_{\mathcal{M}} \sum_{i=1}^{n} \mathcal{L}((\theta^m - \theta^m_{i-1})\mathcal{M})^2 \quad \text{(Equation 5)}
\]

\[
F_i = \frac{1}{m} \sum_{x=1}^{m} \left( \log p(D_i(x)) \right)^2 \quad \text{(Equation 8)}
\]

For regret minimization, we design PM by projection-free mechanisms and stochastic recursive gradient. It focuses the quality of the final performance instead of iterates produced from the course of optimization. For task \( \mathcal{M} \), the regret \( R \) is the difference between the total loss and that of the best parameter \( \theta^* \) of the fixed decision in hindsight.

\[
R_{\mathcal{M}} = \sum_{m=1}^{M} \left( \mathcal{L}(\theta^m(\theta^m), \mathcal{M}_m) - \mathcal{L}(\theta(\theta^m), \mathcal{M}_m) \right) \quad \text{(Equation 6)}
\]

\[
\mathcal{O}(\theta^m) = \mathcal{L}(\theta^m(\theta^m), \mathcal{M}^m) + 1_{\mathcal{M}} \sum_{i=1}^{n} \mathcal{L}((\theta^m - \theta^m_{i-1})\mathcal{M})^2 \quad \text{(Equation 4)}
\]

\[
F_i = \frac{1}{m} \sum_{x=1}^{m} \left( \log p(D_i(x)) \right)^2 \quad \text{(Equation 8)}
\]

\[
R_{\mathcal{M}} = \sum_{m=1}^{M} \left( \mathcal{L}(\theta^m(\theta^m), \mathcal{M}_m) - \mathcal{L}(\theta(\theta^m), \mathcal{M}_m) \right) \quad \text{(Equation 6)}
\]

\[
\mathcal{L}(\theta^m) = \mathcal{L}(\theta^m(\theta^m), \mathcal{M}^m) + 1_{\mathcal{M}} \sum_{i=1}^{n} \mathcal{L}((\theta^m - \theta^m_{i-1})\mathcal{M})^2 \quad \text{(Equation 5)}
\]

\[
F_i = \frac{1}{m} \sum_{x=1}^{m} \left( \log p(D_i(x)) \right)^2 \quad \text{(Equation 8)}
\]

\[
R_{\mathcal{M}} = \sum_{m=1}^{M} \left( \mathcal{L}(\theta^m(\theta^m), \mathcal{M}_m) - \mathcal{L}(\theta(\theta^m), \mathcal{M}_m) \right) \quad \text{(Equation 6)}
\]

\[
\mathcal{O}(\theta^m) = \mathcal{L}(\theta^m(\theta^m), \mathcal{M}^m) + 1_{\mathcal{M}} \sum_{i=1}^{n} \mathcal{L}((\theta^m - \theta^m_{i-1})\mathcal{M})^2 \quad \text{(Equation 5)}
\]

\[
F_i = \frac{1}{m} \sum_{x=1}^{m} \left( \log p(D_i(x)) \right)^2 \quad \text{(Equation 8)}
\]
to update $\theta^n$ in the direction of gradient $g^n$:

$$g^n = \nabla \theta^n - \gamma^n \cdot \theta^n, \text{ where } \gamma^n = \beta^n \cdot \gamma^{n-1}, \text{ and } g^n$$  \hspace{1cm} (Equation 9)

Such a method randomly selects samples to guide the change of gradient and leads to faster converges.

**Overall training process.** When the new gradient $g_m$ and the old gradient $g_k$ are at an acute angle, the model performance dose not decrease and even improves20:

$$\langle g^n, g^k \rangle = \frac{\partial \mathcal{L}(F^n, M^n)}{\partial \theta^n} \cdot \frac{\partial \mathcal{L}(F^k, M^k)}{\partial \theta^k} \geq 0$$  \hspace{1cm} (Equation 10)

$k = 1, \ldots, m - 1$

The regularization projects $g_m$ to the closest gradient $g_k$ by satisfying all the constraint of acute angle:

$$\min \|g^n - g^k\|_2 \quad \text{subject to } \langle g^n, g^k \rangle \geq 0, k = 1, \ldots, m - 1$$  \hspace{1cm} (Equation 11)

In LM, F is positive semi-definite. This property not only guarantees that seeing each task as a factor of the posterior (LM) but also guarantees the acute angle change of a vector after the product (PM). Thus, the PL updates network parameters by using the regularized loss $\mathcal{O}$ in Equation 5, and we re-arrange Equations 7, 8, 9, 10, 11, and 12:

$$f^n \leftarrow f^n + \frac{1}{1 - \rho^n}(d^n - f^n)$$  \hspace{1cm} (Equation 12)

Algorithm 1 gives the algorithm description of PL. In practice, we optimize hyper-parameters using the search method supplied by mature tools.

| Figure S5 | Shows that the method performs better when $\alpha = 0.9393 + 0.907$. Regret and complexity. PM holds a nearly optimal regret bound $O(1)$ with probability at least $1 - \delta$ for any $\delta \in (0, 1)$. $R_M \leq (\log M + 1)(f(\theta') - f(\theta')) + (16LD^2 + 16\delta + 4B) \sqrt{2\log \frac{\alpha}{\delta}} + \frac{1}{1 - \rho^n} \mathcal{O}(\log M + 1)^2$. $D$ is diameter of convex set, and $L$ is $L$-Lipschitz continuous. PM and LM achieve a $O(1)$ per-round computational cost. If the complexity of training a base model to convergence is $O$ and data length is $M$, the overall complexity will be $OM$ (see supplemental mathematics).

**Evaluation metrics**

The classification accuracy is evaluated by assessing the AUC-ROC (the higher the better). The ROC is a curve of the true positive rate (TPR) and the false positive rate (FPR). TN, TP, FP, and FN represent true positive, true negative, false positive, and false negative, respectively.

$$\text{TPR} = \frac{TP}{TP + FN}, \text{ FPR} = \frac{FP}{TN + FP}$$  \hspace{1cm} (Equation 13)

The AUC confidence interval is equal to $\text{AUC} \pm ae \cdot Z_{0.05}$. $Z_{0.05}$ is the two-tailed critical value of the standard normal distribution $NORM.S.INV(1 - \alpha / 2), \alpha = 0.05$. $ae$ is Equation 14, where $s_1$ and $s_2$ are the sizes of the two samples in different labels and $q_0 = \text{AUC}(1 - \text{AUC}), q_1 = \text{AUC}(2 - \text{AUC}) - \text{AUC}^2, q_2 = 2\text{AUC}^2(1 + \text{AUC}) - \text{AUC}^2$. For example, in the SEPSIS dataset, $s_1 = 27,977$ (non-sepsis) and $s_2 = 2,559$ (sepsis), and in COVID-19 dataset, $s_1 = 201$ (survival) and $s_2 = 174$ (death).

$$se = \sqrt{\frac{q_0 + (s_1 - 1)q_1 + (s_2 - 1)q_2}{n_1n_2}}$$  \hspace{1cm} (Equation 14)

The statistical significance is evaluated by the Bonferroni-Dunn test, and $k, n, m$, and $q$ are the number of methods, the number of datasets, the number of cross-validation folds, and the critical value, respectively. If the average rank of baselines $\bar{r} > CD$, the tested method is significantly better.

$$CD = \sqrt{\frac{k(k + 1)}{6N} \cdot N = n \times m}$$  \hspace{1cm} (Equation 15)

Learning stability is evaluated by the gradient fluctuation R (the lower the better). It quantifies the frequency of gradient direction changes during training.

$$R = \frac{1}{n - 1} \sqrt{\sum_{i=1}^{n} (g_i - d_i)^2}, d = \begin{cases} -1, & \text{if } g < 0 \\ 1, & \text{if } g > 0 \end{cases}$$  \hspace{1cm} (Equation 16)

The continuous classification performance is evaluated by the BWT and FWT (the higher the better). They are the influences that learning a new distribution on old and future distributions. $R_D$ is the accuracy on distribution $D_i$ after completing task $D_i$. $R$ is the accuracy with random initialization.

$$\text{BWT} = \frac{1}{|D_i| - 1} \sum_{i=1}^{n} R_{D_ij} - R_{D_i}$$  \hspace{1cm} (Equation 17)

SUPPLEMENTAL INFORMATION

Supplemental information can be found online at https://doi.org/10.1016/j.cellpatter.2023.100687.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China (nos. 62102008 and 62172018) and the National Key Research and Development Program of China under grant 2021YFE0205300.

AUTHOR CONTRIBUTIONS

Conceptualization, C.S., S.H., and H.L.; methodology, C.S.; validation, C.S., M.S., D.C., and B.Z.; investigation, C.S.; writing – original draft, C.S.; writing – review & editing, C.S., H.L., M.S., D.C., B.Z., and S.H.; funding acquisition, H.L. and S.H.; supervision, H.L. and S.H.

DECLARATION OF INTERESTS

The authors declare no competing interests.

INCLUSION AND DIVERSITY

We support inclusive, diverse, and equitable conduct of research.

REFERENCES

1. Singer, M., Deutschman, C.S., Seymour, C.W., Shankar-Hari, M., Annane, D., Bauer, M., Bellomo, R., Bernard, G.R., Chiche, J.D., Coopersmith, C.M., et al. (2016). The third international consensus definitions for sepsis and septic shock (sepsis-3). JAMA, J. Am. Med. Assoc. 315, 801–810. https://doi.org/10.1001/jama.2016.0287.

2. Seymour, C.W., Gesten, F., Prescott, H.C., Friedrich, M.E., Iwashyna, T.J., Phillips, G.S., Lemeshow, S., Osborn, T., Terry, K.M., and Levy, M.M. (2017). Time to treatment and mortality during mandated emergency care for sepsis. N. Engl. J. Med. 376, 2235–2244. https://doi.org/10.1016/j.jemermed.2017.08.088.

3. Zong, N., Li, N., Wen, A., Ngo, V., Yu, Y., Huang, M., Chowdhury, S., Jiang, C., Fu, S., Weinshilboum, R., et al. (2022). BETa: a comprehensive benchmark for computational drug-target prediction. Brief Bioinform. 23, bbach199. https://doi.org/10.1093/bib/bbach199.

4. Wanyan, T., Honarvar, H., Jaladanki, S.K., Zang, C., Naik, N., Somani, S., De Freitas, J.K., Paranjpe, I., Vaid, L., Yao, A., Zhang, J., et al. (2021). Contrastive learning improves critical event prediction in COVID-19 patients. Patterns 2, 100389. https://doi.org/10.1016/j.cellpatter.2021.100389.

5. Rambratna, S., Zeigham, S., Shahabi, K., Shahabi, C., and Liu, Y. (2022). Toward accurate spatiotemporal COVID-19 risk scores using high-resolution
real-world mobility data. ACM Trans. Spat. Algorithms Syst. 8, 1–30. https://doi.org/10.1145/3481044.
6. Goodman, B., and Flaxman, S. (2017). European Union regulations on algorithmic decision-making and a “right to explanation”. AI Mag. 38, 50–57. https://doi.org/10.1609/aimag.v38i3.2741.
7. Tsang, M., Rambhatta, S., and Liu, Y. (2020). How does this interaction affect me? interpretable attribute for feature interactions. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), pp. 6147–6159.
8. Cao, S., Wang, J.R., Ji, S., Yang, P., Dai, Y., Guo, S., Montierth, M.D., Shen, J.P., Zhao, X., Chen, J., et al. (2022). Estimation of tumor cell total mRNA expression in 15 cancer types predicts disease progression. Nat. Biotechnol. 40, 1624–1633. https://doi.org/10.1038/s41587-022-01342-x.
9. Si, N., Zhang, F., Zhou, Z., and Blanchet, J. (2020). Distributionally robust policy evaluation and learning in offline contextual bandits. In Proceedings of the International Conference on Machine Learning (ICML), pp. 8884–8894.
10. Chen, L., Harshaw, C., Hassan, H., and Karbasi, A. (2018). Projection-free online optimization with stochastic gradient: from convexity to submodularity. In Proceedings of International Conference on Machine Learning (ICML), pp. 814–823.
11. Fu, T., Huang, K., Xiao, C., Glass, L.M., and Sun, J. (2022). HINT: hierarchical interaction network for clinical-trial-outcome predictions. Patterns 3, 100445. https://doi.org/10.1016/j.jatter.2022.100445.
12. Hannun, A.Y., Rajpurkar, P., Haghpanahi, M., Tison, G.H., Bourn, C., Du, M., Liu, N., and Hu, X. (2019). Techniques for interpretable machine learning. Commun. ACM 63, 68–77. https://doi.org/10.1145/3350786.
13. Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nat. Mach. Intell. 1, 206–215. https://doi.org/10.1038/s42256-019-0048-x.
14. Tang, X., Jing, L., Zhang, M., et al. (2019). An interpretable mortality prediction model for covid-19 patients. Nat. Mach. Intell. 2, 283–288. https://doi.org/10.1038/s42256-020-0180-7.
15. Johnson, A.E.W., Pollard, T.J., Shen, L., Lehman, L.W.H., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Celi, L.A., and Mark, R.G. (2016). Mimic-iii, a freely accessible critical care database. Sci. Data 3, 160035. https://doi.org/10.1038/sdata.2016.35.
16. Menne, M., Williams, C., Jr., and Vose, R. (2016). Long-term daily and monthly climate records from stations across the contiguous United States (as historical climatology network)(ndp-019). In Tech. Rep., Environmental System Science Data Infrastructure for a Virtual Ecosystem. https://doi.org/10.3334/CDIAC/CLIN/DP019.
17. Dau, H.A., Bagnall, A., Kamgar, K., Yeh, C.C.M., Zhou, Y., Gharghabi, S., Ratanamahatana, C.A., and Keogh, E. (2019). The ucr time series archive. IEEE/CAA J. Autom. Sinica 6, 1293–1305. https://doi.org/10.1109/JAS.2019.1911747.
18. Sun, C., Hong, S., Zhou, Y., Jiang, X., Xiao, C., and Sun, J. (2020). Opportunities and challenges of deep learning methods for electrocardiogram data: a systematic review. Comput. Biol. Med. 122, 103801. https://doi.org/10.1016/j.compbiomed.2020.103801.
19. Sun, C., Hong, S., Song, M., and Li, H. (2021). Te-esn: time encoding echo state network for prediction based on irregularly sampled time series data. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), pp. 5321–5331.
20. Hong, S., Zhou, Y., Jiang, X., Xiao, C., and Sun, J. (2020). Continual lifelong learning with neural networks: a review. Neural Netw. 113, 54–71. https://doi.org/10.1016/j.neunet.2019.01.012.
21. Gee, A.H., Chang, J., Ghosh, J., and Paydarfar, D. (2018). Bayesian online changepoint detection of physiological transitions. In International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 45–48. https://doi.org/10.1109/EMBC.2018.8512204.
22. Rieke, M.A., Seyedi, S., and Jeter, R. (2019). Early prediction of sepsis from clinical data: the physionet/computing in cardiology challenge 2019. In Computing in Cardiology, pp. 1–4. https://doi.org/10.23919/CinC49843.2019.9005736.
23. Reyna, M.A., Josef, C., Seyedi, S., and Jeter, R. (2019). Early prediction of sepsis from clinical data: the physionet/computing in cardiology challenge 2019. In Computing in Cardiology, pp. 1–4. https://doi.org/10.23919/CinC49843.2019.9005736.
39. Wang, Y., Wang, L., Rastegar-Mojarad, M., Moon, S., Shen, F., Afzal, N., Liu, S., Zeng, Y., Mehrabi, S., Sohn, S., and Liu, H. (2018). Clinical information extraction applications: a literature review. J. Biomed. Inform. 77, 34–49. https://doi.org/10.1016/j.jbi.2017.11.011.

40. Yu, G., Cheng, K., Liu, Q., Wu, W., Hong, H., and Lin, X. (2022). Clinical outcomes of severe sepsis and septic shock patients with left ventricular dysfunction undergoing continuous renal replacement therapy. Sci. Rep. 12, 9360. https://doi.org/10.1038/s41598-022-13243-9.

41. Maslove, D.M., Tang, B., Shankar-Hari, M., Lawler, P.R., Angus, D.C., Baillie, J.K., Baron, R.M., Bauer, M., Buchman, T.G., Calfee, C.S., et al. (2022). Redefining critical illness. Nat. Med. 28, 1141–1148. https://doi.org/10.1038/s41591-022-01843-x.

42. Danlos, F.X., Grajeda-Iglesias, C., Durand, S., Sauvat, A., Roumier, M., Cantin, D., Colomba, E., Rohmer, J., Pommeret, F., Baciarello, G., et al. (2021). Metabolomic analyses of covid-19 patients unravel stage-dependent and prognostic biomarkers. Cell Death Dis. 12, 258. https://doi.org/10.1038/s41419-021-03540-y.

43. Mallya, A., and Lazebnik, S. (2018). Packnet: adding multiple tasks to a single network by iterative pruning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7765–7773. https://doi.org/10.1109/CVPR.2018.00810.

44. Bai, B., Liang, J., Zhang, G., Li, H., Bai, K., and Wang, F. (2021). Why attentions may not be interpretable? In ACM SIGKDD Conference on Knowledge Discovery and Data Mining (SIGKDD), pp. 25–34. https://doi.org/10.1145/3447548.3467307.

45. Johnson, J.M., and Khoshgoftaar, T.M. (2019). Survey on deep learning with class imbalance. J. Big Data 6, 27. https://doi.org/10.1186/s40537-019-0192-5.