LIFE: Learning Individual Features for Multivariate Time Series Prediction with Missing Values

Zhao-Yu Zhang, Shao-Qun Zhang, Yuan Jiang, and Zhi-Hua Zhou
National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China
Email: {zhangzhaoyu, zhangsq, jiangy, zhouzh}@lamda.nju.edu.cn

***Abstract*—Multivariate time series (MTS) prediction is ubiquitous in real-world fields, but MTS data often contains missing values. In recent years, there has been an increasing interest in using end-to-end models to handle MTS with missing values. To generate features for prediction, existing methods either merge all input dimensions of MTS or tackle each input dimension independently. However, both approaches are hard to perform well because the former usually produce many unreliable features and the latter lacks correlated information. In this paper, we propose a Learning Individual Features (LIFE) framework, which provides a new paradigm for MTS prediction with missing values. LIFE generates reliable features for prediction by using the correlated dimensions as auxiliary information and suppressing the interference from uncorrelated dimensions with missing values. Experiments on two real-world data sets verify the superiority of LIFE to existing state-of-the-art models. The full version of this work can refer to arXiv (2109.14844).**

***Index Terms*—Multivariate Time Series Prediction, Missing Values, Correlated Dimensions, Individual Features**

I. Introduction

Multivariate time series (MTS) data is prevalent in many fields, such as health care [12, 13] and weather forecasting [2]. Due to sensor damage, irregular sampling, and other reasons, real-world MTS data usually contains lots of missing values.

For MTS prediction with missing values, a natural idea is the two-step approach: first fill the missing values with replacement values, that is, *data imputation*, and then apply the complete data to prediction. However, previous studies suggest that the separation of imputation and prediction processes has a proclivity for suboptimal results [2, 3]. Besides, it is usually unnecessary to perform imputation since many real-world tasks care more about prediction results than restoring original data. Recent years have witnessed an increasing interest in exploring end-to-end models, which perform better than the two-step methods [3, 11]. These models usually estimate the missing values or treat them as zeros, and then merge all input dimensions [2, 13] or feed each dimension independently [7, 15] into Recurrent Neural Networks (RNNs) or attention-based models to generate features for prediction.

However, merging all input dimensions may result in many unreliable features, which is caused by the interference of missing values. Fig. 1 gives a vivid illustration of the feature generation process above, where $\mathbf{X}_{:,d}$ denotes the d-th dimension. At each timestamp, three input points are converted to a feature. Once there is a missing value of any input dimension, the generated feature is unreliable. Adhering to this line of thought, we can calculate that 30% of input values are missing, but 80% of generated features are unreliable. On the other hand, some approaches tackle each input dimension independently. Thus, the generated feature sequence has the same unreliable rate as the missing rate of inputs. Nevertheless, this manner ignores the information provided by the correlated dimensions. Take the inputs in Fig. 1 as an example, it's observed $\mathbf{X}_{:,1}$ and $\mathbf{X}_{:,2}$ are highly correlated. If one dimension is missing and the other dimension is observed, we should employ the correlated dimensions as auxiliary information to make the generated feature sequence more credible.

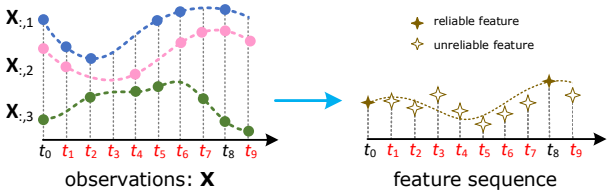


Fig. 1. Feature generation of most existing end-to-end models. Merging all input dimensions leads to numerous unreliable features.

In this paper, we propose a Learning Individual Features (LIFE) framework for MTS prediction with missing values. The roadmap is shown in Fig. 2. The key idea of LIFE is Step 1 and Step 2, which collect *credible and correlated* dimensions to build features, that is, *individual features*. Individual features not only fuse with auxiliary information provided by correlated dimensions but also discard most of the dimensions uncorrelated to the concerned dimension. Therefore, LIFE can generate more reliable features than existing approaches. Our main contributions are summarized as follows:

- We propose a novel framework LIFE, which provides a new paradigm for MTS prediction with missing values. LIFE builds individual features by credible and correlated dimensions, enabling it to generate many reliable features for the downstream prediction task.
- We present a general approach for extracting stable and credible dimensional correlations for MTS with missing values. We also provide a concrete algorithm for implementing this approach.
- We empirically verify that LIFE outperforms the state-of-the-art (SOTA) models on two real-world data sets.

II. Related Work

MTS prediction with missing values: Recently, some end-to-end methods based on deep learning have achieved good

2374-8486/21/$31.00 ©2021 IEEE
DOI 10.1109/ICDM51629.2021.00197

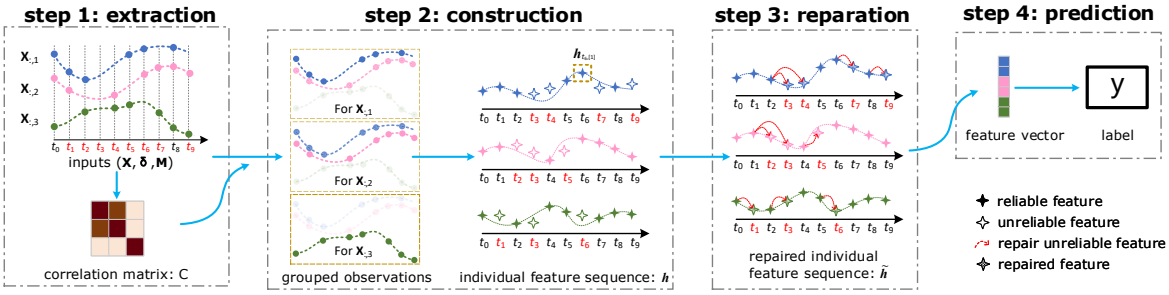


Fig. 2. Roadmap of LIFE framework. Steps 1 – 4 corresponds to subsections IV-A – IV-D, respectively.

performance in MTS prediction with missing values. These methods can be roughly divided into two categories: 1) Models based on RNNs generate features recurrently. Furthermore, the feature of the current timestamp is repaired concerning that of the adjacent timestamps. GRU-D [3] develops a decay mechanism to capture the temporal correlations to repair missing information. BRITS [2] applies the decay mechanism to the bidirectional Long Short Term Memory (Bi-LSTM) [6] model. FG-LSTM [15] focuses on modeling the temporal dependency for a single input dimension and can be regarded as running an LSTM for each dimension independently; 2) Models based on Self Attention employ attention mechanism instead of recurrence to generate features. SAnD [13] applies the attention-based method to healthcare applications without repairing the missing information. However, when generating features from inputs, the above methods calculate each input dimension independently or fuse all dimensions. Both manners would damage the performance because of the lack of dimensional correlations or the problem of unreliable features.

Dimensional correlation measurement: There have been many studies exploring dimensional correlations for complete MTS data. The Pearson Correlation Coefficient and Mutual Information can express the correlation between two time series. Besides, time series distance/dissimilarity can be used to imply correlation after a simple conversion (e.g., calculating the reciprocal or negative exponent). Euler distance and Dynamic Time Warping (DTW) are two commonly used distances. However, these methods may lead to fictitious correlations when the data contains missing values. Some kernel-based methods can handle MTS with missing values and provide similarity measurement [4, 10]. Still, the missing values could also damage the results, especially with a large missing rate.

III. Preliminaries

Let $\mathbf{X} \in \mathbb{R}^{T \times D}$ denote a D-dimensional time series with T timestamps, and $\boldsymbol{s} = [s_t]_{t=1}^{T} \in \mathbb{R}^{T}$ is the corresponding timestamp sequence. The masking matrix $\mathbf{M} \in \{0,1\}^{T \times D}$ indicates whether the values in $\mathbf{X}$ are missing: if X_{td} is observed, $\mathrm{M}_{td} = 1$, otherwise $\mathrm{M}_{td} = 0$. Let $\boldsymbol{\delta} \in \mathbb{R}_{+}^{T \times D}$ denote the time interval matrix, which consists of the time gap δ_{td} from the timestamp of last observation to current timestamp. δ_{td} is defined as:

$$\delta_{td} = \begin{cases} s_t - s_{t-1} + \delta_{t-1,d}, & t > 1, \mathrm{M}_{t-1,d} = 0; \\ s_t - s_{t-1}, & t > 1, \mathrm{M}_{t-1,d} = 1; \\ 0, & t = 1. \end{cases}$$

Note that subscripts $[\cdot]_{t,:}$ and $[\cdot]_{:,d}$ indicate the raw and column vectors of a matrix, respectively. For example, $\mathbf{X}_{t,:}$ and $\mathbf{X}_{:,d}$ denote the vectors of the t-th timestamp and d-th dimension, respectively. Suppose a vector $\boldsymbol{u} \in \mathbb{R}^{kD}$ consists of D sub-vectors, each of which has a size of positive integer k, then let $\boldsymbol{u}_{[d]}$ denote the d-th sub-vector:

$$\boldsymbol{u}_{[d]} = \left(u_{k(d-1)+1},\ u_{k(d-1)+2},\ \ldots,\ u_{kd}\right). \tag{1}$$

Let the *correlation matrix* $\mathbf{C} \in [0,1]^{D \times D}$ denote the pairwise correlations of D dimensions in MTS, where C_{ij} represents the correlation value between $\mathbf{X}_{:,i}$ and $\mathbf{X}_{:,j}$. The larger C_{ij} is, the more relevant the two corresponding dimensions are. $\mathbf{C}$ is symmetric, and the diagonal elements are 1.

This work aims to predict the supervised signals $\left\{y^{(n)}\right\}_{n=1}^{N}$ using the incomplete MTS values $\left\{\mathbf{X}^{(n)}\right\}_{n=1}^{N}$, time interval sequences $\left\{\boldsymbol{\delta}^{(n)}\right\}_{n=1}^{N}$, and masking matrices $\left\{\mathbf{M}^{(n)}\right\}_{n=1}^{N}$, where N denotes the number of MTS samples.

IV. LIFE Framework

In this section, we present the LIFE framework for MTS prediction with missing values. Fig. 2 shows the roadmap of LIFE. Step 1 extracts a credible and stable correlation matrix by penalizing missing values in qualifying dimensional correlation. In Step 2, we group the observations by the correlation matrix, and thus, build individual features using Self Attention. Step 3 repairs the individual features according to the temporal information. In Step 4, we obtain the prediction results by the MTS classifiers or regressors. Finally, we can optimize the parameters in LIFE by jointly minimizing the imputation and prediction loss.

A. Correlation Matrix Extraction

This subsection aims to extract the credible and stable correlation matrix of the MTS data with missing values, which requires robustness to missing values and hyper-parameter settings. However, it is quite difficult for the conventional *correlation matrix extraction* (CME) methods mentioned in

Algorithm 1 CME-PDTW algorithm

Input: A data set with N samples. The n-th sample consists of D-dimensional time series $\mathbf{X}^{(n)}$, time interval matrix $\boldsymbol{\delta}^{(n)}$ and masking matrix $\mathbf{M}^{(n)}$.
Output: Correlation Matrix $\mathbf{C}$.

1: **for** $n = 1 \to N$ **do** ▷ traverse each sample
2: $\quad \mathbf{X}^{(n)} = \text{interpolate}\left(\mathbf{X}^{(n)}\right)$ ▷ impute missing values
3: $\quad \mathbf{S}^{(n)} = \text{zeros}(D, D)$ ▷ PDTW distance
4: $\quad \mathbf{Q}^{(n)} = \text{zeros}(D, D)$ ▷ weight of this sample
5: $\quad$ **for** $i = 1 \to D$ **do**
6: $\quad\quad$ **for** $j = i + 1 \to D$ **do**
7: $\quad\quad\quad \mathrm{S}_{ij}^{(n)} = \mathrm{S}_{ji}^{(n)} = \text{PDTW}\left(\mathbf{X}_{:,i}^{(n)}, \mathbf{X}_{:,j}^{(n)}\right)$
8: $\quad\quad\quad \mathrm{Q}_{ij}^{(n)} = \mathrm{Q}_{ji}^{(n)} = \text{sum}\left(\mathbf{M}_{:,i}^{(n)} + \mathbf{M}_{:,j}^{(n)}\right)$
9: $\quad\quad$ **end for**
10: $\quad$ **end for**
11: **end for**
12: $\bar{\mathbf{S}} = \text{weighted\_mean}(\mathbf{S}, \mathbf{Q})$
13: $\mathbf{C}^{(\text{non-diag})} = \text{normalize}\left(1/\bar{\mathbf{S}}^{(\text{non-diag})}\right)$
14: $\mathbf{C}^{(\text{diag})} = 1$

Section II. The reasons lie in two folds. First, some of these methods exclude the missing values and calculate similarity only when both points are observed. Obviously, this approach is not advisable because it discards much useful information. Second, the other methods first impute the missing values and then conduct CME. However, the imputation of missing values could be unreliable. Thus some dimensions with large missing rates may lead to unstable or even fictitious correlations.

To tackle the drawbacks above, we attempt to punish the missing values for CME. Since the imputation of missing values could be untrustworthy, we rely more on observed values and trust the correlations extracted by dimensions with low missing rates. Note that we are not trying to give each pairwise dimensional correlation a precise estimation but only look for those reliable and stable correlations. We believe penalizing missing values is a general method for CME and can be applied to various distances. In the following, we give a DTW-based algorithm as a concrete implementation.

The key idea of Correlation Matrix Extraction - Penalty Dynamic Time Warping (CME-PDTW) algorithm is to convert the pairwise PDTW distances/dissimilarities to a dimensional correlation matrix. The PDTW distance between $\mathbf{X}_{:,d_1}$ and $\mathbf{X}_{:,d_2}$ works by adding penalties of missing values to the original DTW algorithm:

$$\text{PDTW}(\mathbf{X}_{:,d_1}, \mathbf{X}_{:,d_2}) = \min_{\pi} \sum_{(i,j)\in\pi} \left[(\mathrm{X}_{id_1} - \mathrm{X}_{jd_2})^2 + \phi(\mathrm{X}_{id_1}, \mathrm{X}_{jd_2}) \right], \quad (2)$$

where π is the search path of DTW and $\phi(\cdot)$ is the penalty term. The farther the last observation is, the more unreliable the estimation of missing value is. So $\phi(\cdot)$ should impose more punishment on the consecutive missing values than sporadic missing values. For i, j, we formulate $\phi(\cdot)$ as follows:

$$\phi\left(\mathrm{X}_{id_1}, \mathrm{X}_{jd_2}\right) = p \left[\delta_{id_1}(1 - \mathrm{M}_{id_1}) + \delta_{jd_2}(1 - \mathrm{M}_{jd_2}) \right]. \quad (3)$$

The produce of CME-PDTW is shown in Algorithm 1. We use linear interpolation to estimate the missing values (Line 2) and calculate the pairwise PDTW distances for each sample (Lines 5-10). Next, we take the weighted mean over all samples to obtain the averaged distance matrix $\bar{\mathbf{S}}$, and the weight is proportional to the number of observations (Line 12). Thus, $\bar{\mathbf{S}}$ is mainly based on those samples with few missing values. Finally, we transform the distance matrix to the correlation matrix (Line 13-14). The interference of missing values is inhibited from two aspects to extract credible and stable correlations. On the one hand, samples with more missing values are assigned with smaller weights, which weakens their influence on the results. On the other hand, missing values will cause large PDTW distances, leading to small values in the correlation matrix. Thus, only those credible and stable correlations can be extracted, mainly relying on the samples and dimensions with low missing rates.

B. Individual Features Construction

The correlation matrix is used to collect the correlated dimensions for each concerned input dimension. Firstly, provided a positive integer k, which controls the model complexity, we expand $\mathbf{C}$ to a larger matrix $\mathbf{C}' \in \mathbb{R}^{kD \times 3D}$:

$$\mathrm{C}'_{ij} = \mathrm{C}_{i'j'}, \quad i' = \lceil i/k \rceil, \quad \text{and} \quad j' = j \bmod D,$$

where $\lceil\rceil$ denotes the rounding up operation. Next, let $\mathbf{I}_t = (\mathbf{X}_{t,:}, \boldsymbol{\delta}_{t,:}, \mathbf{M}_{t,:})$ denote the input vector at time t and any missing values are treated as zero. We can transform $\mathbf{I}_t$ into an embedding vector $\boldsymbol{e}_t \in \mathbb{R}^{kD}$ by a one-layer neural network and the *positional encoding* $\text{pe}(\cdot)$ as following:

$$\boldsymbol{e}_t = \sigma\left[\mathbf{W} \odot \mathbf{C}' \times \mathbf{I}_t + \boldsymbol{b}\right] + \text{pe}(t), \quad (4)$$

where $\mathbf{W} \in \mathbb{R}^{kD \times 3D}$ is the connection weights, $\boldsymbol{b} \in \mathbb{R}^{kD}$ is the bias, $\odot$ and $\times$ are element-wise and matrix product, respectively, σ is the sigmoid function, and $\text{pe}(\cdot)$ is an embedding layer (a.k.a. lookup table) which maps the one-hot encoded timestamp t to an kD-dimensional vector. Notice that the multiplier $\mathbf{W} \odot \mathbf{C}'$ discards the uncorrelated information for each concerned dimension, and thus, group the observations by correlated dimensions. As a result, the k-dimensional sub-vector $\boldsymbol{e}_{t,[d]} = (e_{t,k(d-1)+1}, \ \ldots \ , e_{t,kd})$, as defined in Eq. 1, only relies on the correlated observations corresponding to X_{td}. We call this sub-vector $\boldsymbol{e}_{t,[d]}$ the *individual embedding vector* for $d \in \{1, 2, \ldots, D\}$. Thus, the embedding vector $\boldsymbol{e}_t$ consists of D individual embedding vectors.

Provided the embedding vector $\boldsymbol{e}_t$, we can obtain the *individual features* $\boldsymbol{h}_t$ using Self Attention. Formally, for each timestamp t, we use $\boldsymbol{e}_t$ as the query vector and generate the feature vector $\boldsymbol{h}_t$ as follows:

$$\boldsymbol{h}_t = \textstyle\sum_{s=1}^{T} a_s \boldsymbol{e}_s \quad \text{and} \quad a_s = \text{softmax}\left(f\left(\boldsymbol{e}_s, \boldsymbol{e}_t\right)\right),$$

where f is a one-layer network that calculates dependency score between $\boldsymbol{e}_s$ and $\boldsymbol{e}_t$. Notice that the individual vector $\boldsymbol{h}_t$ still has the same size as $\boldsymbol{e}_t$. As shown in Fig. 2, we can obtain the individual feature sequence $[\boldsymbol{h}_t]_{t=1}^{T}$ in Step 2, and an example of individual feature is $\boldsymbol{h}_{t_6,[1]}$, which is a k-dimensional vector corresponding to $\mathrm{X}_{t_6,1}$.

C. Feature Reparation

Note that some individual features $[\boldsymbol{h}_t]_{t=1}^T$ are still unreliable. If X_{td} is missing, then the individual feature $\boldsymbol{h}_{t,[d]}$ is unreliable (e.g., the hollow star in Fig. 2) according to Eq. (4). A natural method is to repair the unreliable by temporal dependency. Here, we employ the decay mechanism [3] and obtain the repaired individual feature $\tilde{\boldsymbol{h}}_{t,[d]}$ as follows:

$$\tilde{\boldsymbol{h}}_{t,[d]} = \begin{cases} \boldsymbol{h}_{t,[d]}, & \text{if } \mathrm{M}_{td} = 1; \\ \gamma_{td}\ \boldsymbol{h}_{t',[d]} + (1 - \gamma_{td})\, \boldsymbol{h}_{t,[d]}, & \text{if } \mathrm{M}_{td} = 0, \end{cases}$$

where $t' = t - \boldsymbol{\delta}_{td}$ is the timestamp of the last observed value for X_{td}, $\boldsymbol{\gamma}_{td}$ is the decay rate, which indicates the how much information of last observed value remains. Intuitively, the decay rate can be calculated according to:

$$\boldsymbol{\gamma}_{td} = \exp\left[-\max\left(0,\ w_d \boldsymbol{\delta}_{td} + a_d\right)\right],$$

where both $\boldsymbol{w} = [w_d]_{d=1}^D \in \mathbb{R}^D$ and $\boldsymbol{a} = [a_d]_{d=1}^D \in \mathbb{R}^D$ are learnable parameters. The repair process is illustrated in Fig. 2, for example, both the hollow stars $\tilde{\boldsymbol{h}}_{t_3,[1]}$ and $\tilde{\boldsymbol{h}}_{t_4,[1]}$ are repaired by $\tilde{\boldsymbol{h}}_{t_2,[1]} = \boldsymbol{h}_{t_2,[1]}$.

D. Prediction

Provided the repaired individual features $(\tilde{\boldsymbol{h}}_1, \tilde{\boldsymbol{h}}_2, \ldots, \tilde{\boldsymbol{h}}_T)$, we can obtain the output using a MTS classifier or regressor:

$$\hat{y} = f_{out}\left(f_{agg}\left(\tilde{\boldsymbol{h}}_1, \tilde{\boldsymbol{h}}_2, \ldots, \tilde{\boldsymbol{h}}_T\right)\right),$$

where $\hat{y}$ denotes the output, f_{agg} aggregates the feature sequence $[\tilde{\boldsymbol{h}}_t]_{t=1}^T$ to a fixed-size vector, and f_{out} is a one-hidden-layer network with softmax for classification or only linear layer for regression. We implement f_{agg} by Dense Interpolation [14], which shows better performance than the mean pooling and the attention pooling. The key idea of Dense Interpolation is to calculate the weighted mean of the whole sequence at specific timestamps. For more, we refer to [14].

LIFE jointly optimize both prediction and imputation loss:

$$L = L_{pred} + \alpha L_{imp},$$

where L_{pred} is the prediction loss, L_{imp} denotes the imputation loss, and $\alpha \in \mathbb{R}_+$ is a balancing weight. The prediction loss is usually the cross-entropy loss for a classification task or the mean square error (MSE) for a regression/forecasting task. The intuition of adding imputation loss is to provide more supervised information for our model, leading to a better representation of the concerned MTS data. Here, we formulate L_{imp} as the MSE between the observed and imputation values:

$$L_{imp} = \sum_{t=1}^{T}\sum_{d=1}^{D} \mathrm{M}_{td}\left(\mathrm{X}_{td} - \hat{\mathrm{X}}_{td}\right)^2 \Big/ \sum_{t=1}^{T}\sum_{d=1}^{D} \mathrm{M}_{td}\ ,$$

where the imputation values $\hat{\mathrm{X}}_{td}$ are generated by

$$\hat{\mathrm{X}}_{td} = g\left(\tilde{\boldsymbol{h}}_{t,[d]}\right),$$

in which $g(\cdot)$ is a one-layer perceptron.

V. Experiments

In this section, we will evaluate the performance of the proposed model and the CME-PDTW algorithm.

A. Data Sets

In order to verify our idea and evaluate the performance of LIFE, we use two widely used benchmark data sets [11] in the community to conduct experiments.

PhysioNet [12] data set contains 4000 records from intensive care unit (ICU). Following [2], we preprocess each record into 48 timestamps containing 35 measurements (such as pH, heart rate, etc.), and the total average missing rate is 82.36%. On this data set, we do mortality classification.

Human Activity [8] data set contains 3D positions of the waist, chest, and ankles (12 dimensions in total) of 11 activities (e.g., walking). We fix the sampling interval to 100 milliseconds and take the records of 1000 milliseconds as one sample. There are 4,817 samples in total, and each sample has 10 timestamps and 12 dimensions. The average missing rate is 24.45%. In real-world applications, sensor damage can cause high missing rates of corresponding dimensions. We first randomly choose n sensors (dimensions) as "damaged sensors", and then randomly eliminate 90% observations of the damaged sensors. We alter n from 0 – 11 to generate 12 data sets and do multi-class classification task on them.

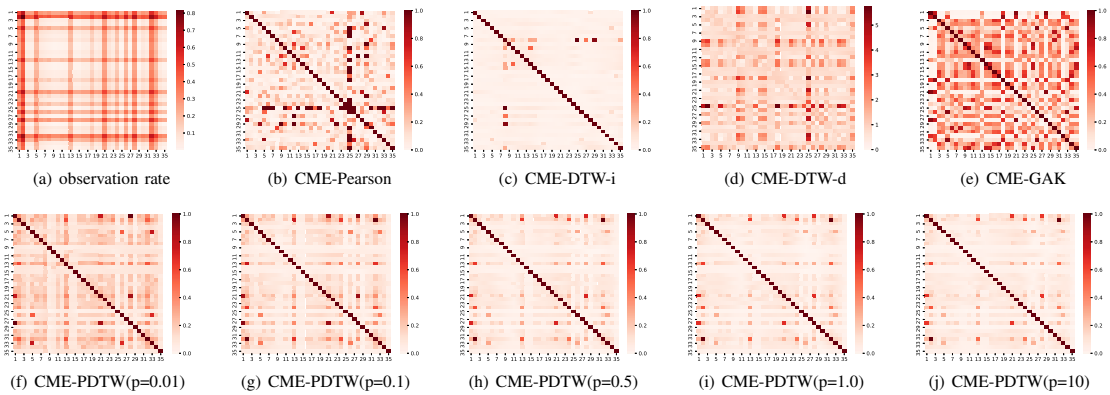
B. Benchmark Methods

We use the following two kinds of methods as baselines:

- *Two-Step Models* comprise imputation and prediction independently. We first do imputation: "-a" means filling in missing values with the average observations over time and "-m" denotes MICE [1], which is a widely used imputation method. Then for prediction, we employ ROCKET [5], one of the SOTA classifiers for MTS without missing values, or LSTM. Thus, the baselines are **ROCKET-a**, **ROCKET-m**, **LSTM-a** and **LSTM-m**.
- *End-to-End Models* jointly optimize both imputation and prediction processes. **GRU-D** [3], **BRITS** [2], and **SAnD** [13] merge all input dimensions, while **FG-LSTM** [15] handles each dimension independently.

C. Settings

We normalize the data to ensure that each input dimension has zero mean and unit variance. Following [3, 13], the missing values for all end-to-end models are treated as zero. We make sure all models have comparable numbers of parameters, that is, about 130 K parameters for PhysioNet and 20 K parameters for human activity data, which are sufficient to make each model perform well at an acceptable computational cost. As the model complexity of LIFE is mainly dominated by the vector size k of each individual feature, so the number of parameters requires $k \leq 6$ and we set $k = 6$. For some parameters that have little impact on the complexity and performance of the model, we set them to fixed values: the number of dense interpolated timestamps is 3 and the weight balancing factor $\alpha = 1$. All deep



(a) observation rate (b) CME-Pearson (c) CME-DTW-i (d) CME-DTW-d (e) CME-GAK

(f) CME-PDTW(p=0.01) (g) CME-PDTW(p=0.1) (h) CME-PDTW(p=0.5) (i) CME-PDTW(p=1.0) (j) CME-PDTW(p=10)

Fig. 3. Pairwise observation rate and correlation matrices extracted by different CME methods.

models are trained by Adam optimizer [9] with learning rate 0.001 and batch size 64. Training epochs for PhysioNet and human activity data are 200 and 100, respectively. The hyper-parameters are optimized by cross validation. We run the codes on Ubuntu 18.04 system with a single Nvidia TITAN XP graphic card and 32G memory. Our codes are available at http://www.lamda.nju.edu.cn/code_LIFE.ashx.

The baseline CME methods are based on Algorithm 1 with differences of missing-value processing in Step 2 and of distance calculation in Step 7. **CME-Pearson** indicates the weighted absolute value of the pairwise Pearson Correlation Coefficient, where all missing values are excluded. **CME-DTW-i** and **CME-DTW-d** impute (linear interpolation) and drop the missing values, respectively, and then, conduct CME based on DTW. **CME-GAK** employs Global Alignment Kernel (GAK) [4], which is a widely used kernel for time series. It drops missing values and calculate similarity via alignment and warping technics. **CME-PDTW** applys the CME-PDTW algorithm and alter p from $\{0.01, 0.1, 0.5, 1.0, 10\}$.

D. CME Results

The first experiment is to evaluate the robustness and performance of the CME-PDTW algorithm and the baselines. We conduct CME on the PhysioNet data set. Fig. 3 displays the average pairwise observation rate (i.e., 1 minus missing rate) and the extracted correlation matrices.

From Fig. 3, we can see that the missing values will interfere with CME to a large extent. The correlation matrix of CME-DTW-i is inundated with fictitious correlations and far away from the ground truth. The missing values bring serious fictitious correlations to it. The correlation matrices obtained by CME-Pearson, CME-DTW-d, and CME-GAK are not stable. Most large values of the correlation matrices are corresponding to low observation rates. However, CME-PDTW performs well and is not sensitive to the only hyper-parameter penalty coefficient p. As we can see, the correlation matrices tend to be stable with an increase of p. CME-PDTW can suppress the interference caused by missing values and obtain a credible and stable correlation matrix. We employ $p = 0.5$ through the experiments.

E. Performance Evaluation

Following [2, 3], we report the results of 5-fold cross validation and use the following evaluation criteria: area under ROC curve (AUC score) for PhysioNet since it is a class imbalance data and accuracy for human activity data.

TABLE I
CLASSIFICATION PERFORMANCES ON PHYSIONET.

	Models	AUC ($\pm$ std)	Paras ($\sim$ K)
Two-Steps	ROCKET-a	0.8084 $\pm$ 0.018	/
	ROCKET-m	0.8103 $\pm$ 0.015	/
	LSTM-a	0.8091 $\pm$ 0.017	130.8
	LSTM-m	0.8046 $\pm$ 0.018	130.8
End-to-End	GRU-D	0.8379 $\pm$ 0.012	127.7
	BRITS	0.8329 $\pm$ 0.008	129.0
	SAnD	0.8011 $\pm$ 0.026	140.9
	FG-LSTM	0.8193 $\pm$ 0.018	130.3
	LIFE	**0.8451 $\pm$ 0.012**	**129.3**

The results on the PhysioNet data set, as shown in Table I, indicate that the LIFE model outperforms the others. Note that the two steps methods perform significantly worse than the SOTA end-to-end methods, which is consistent with previous research [2, 3]. Next, we only focus on end-to-end models.

We further investigate the performances of models with different missing rates. The mean and standard deviation of accuracy on the human activity data sets are shown in Fig. 4. The more damaged sensors are, the higher the missing rate is. We can see that LIFE achieves the best results and significantly outperforms the others on most of the missing rates. When the missing rate is high, it can suppress the interference of missing values and generate as reliable features as possible. When there are few missing values, it also explores correlations effectively and maintains good performance. An interesting aspect of this result is FG-LSTM. It shows that handling each input dimension separately can bring benefits when the missing rate is high, but it can also damage the performance when the missing rate is low. Therefore, it is important to not

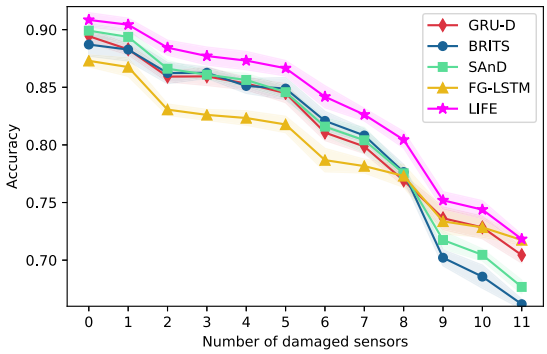


Fig. 4. Performances on human activity data with different numbers of damaged sensors. Shaded regions is the standard deviation over five trials.

only suppress the interaction of missing values but also utilize dimensional correlations, as LIFE does.

F. Individual Features Construction Comparison

In order to explore whether the individual feature construction does bring benefits, we keep the hyper-parameters of LIFE consistent and change the correlation matrices:

- **Ones** denotes the Full-1 matrix, which leads LIFE to fuse all input dimensions, just as most existing models do.
- **Rand** is a random matrix sampling from $[0, 1]$.
- **Diag** is a diagonal-1 matrix, which makes LIFE build feature for each input dimension independently.
- **CME-Pearson**, **CME-DTW-i**, **CME-DTW-d**, **CME-GAK**, and **CME-PDTW** are correlation matrices extracted via Algorithm 1 with different distances.

TABLE II
CLASSIFICATION PERFORMANCES OF LIFE WITH DIFFERENT CORRELATION MATRICES ON PHYSIONET.

Correlation Matrix	AUC ($\pm$ std)	Paras ($\sim$ K)
Ones	0.8204 $\pm$ 0.010	129.3
Rand	0.8251 $\pm$ 0.009	129.3
Diag	0.8355 $\pm$ 0.012	129.3
CME-Pearson	0.8336 $\pm$ 0.007	129.3
CME-DTW-i	0.8232 $\pm$ 0.010	129.3
CME-DTW-d	0.8346 $\pm$ 0.009	129.3
CME-GAK	0.8374 $\pm$ 0.012	129.3
CME-PDTW	**0.8451 $\pm$ 0.012**	**129.3**

The results of LIFE on the PhysioNet data set are shown in Table II. LIFE with the correlation matrix extracted by CME-PDTW achieves the highest AUC, which shows the effectiveness of building individual features and the CME-PDTW algorithm. On the contrary, Ones performs significantly worse than CME-PDTW, verifying that merging all input dimensions is not acceptable since it will damage the classification performance. Diag, CME-Pearson, CME-DTW-d, and CME-GAK also achieve competitive AUCs, verifying that building individual features can bring benefits. The poor performances of Rand and CME-DTW-i show the wrong correlations can damage the performance of downstream prediction task.

VI. CONCLUSION

In this paper, we proposed a novel framework LIFE, which provides a new paradigm for MTS prediction with missing values. For each input dimension, LIFE utilizes credible and correlated dimensions to build individual features. So LIFE is able to not only suppress the interference of missing values but also generate reliable and effective features for prediction. Besides, we also present a CME-PDTW algorithm for extracting credible and stable dimensional correlations. Experiments conducted on two real-world data sets show that LIFE outperforms the existing SOTA models and the CME-PDTW algorithm can extract credible and stable correlations.

ACKNOWLEDGMENT

This research was supported by the National Key Research and Development Program of China (2020AAA0109400). The corresponding author for this work is Yuan Jiang.

REFERENCES

[1] M. J. Azur, E. A. Stuart, C. Frangakis, and P. J. Leaf, "Multiple imputation by chained equations: What is it and how does it work?" *International Journal of Methods in Psychiatric Research*, vol. 20, no. 1, pp. 40–49, 2011.

[2] W. Cao, D. Wang, J. Li, H. Zhou, L. Li, and Y. Li, "Brits: Bidirectional recurrent imputation for time series," in *Advances in Neural Information Processing Systems 31*, 2018, pp. 6775–6785.

[3] Z. Che, S. Purushotham, K. Cho, D. Sontag, and Y. Liu, "Recurrent neural networks for multivariate time series with missing values," *Scientific Reports*, vol. 8, no. 1, pp. 1–12, 2018.

[4] M. Cuturi, "Fast global alignment kernels," in *Proceedings of the 28th International Conference on Machine Learning*, 2011, pp. 929–936.

[5] A. Dempster, F. Petitjean, and G. I. Webb, "Rocket: Exceptionally fast and accurate time series classification using random convolutional kernels," *Data Mining and Knowledge Discovery*, vol. 34, no. 5, pp. 1454–1495, 2020.

[6] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional lstm and other neural network architectures," *Neural Networks*, vol. 18, no. 5-6, pp. 602–610, 2005.

[7] H. Harutyunyan, H. Khachatrian, D. C. Kale, G. Ver Steeg, and A. Galstyan, "Multitask learning and benchmarking with clinical time series data," *Scientific Data*, vol. 6, no. 1, pp. 1–18, 2019.

[8] B. Kaluža, V. Mirchevska, E. Dovgan, M. Luštrek, and M. Gams, "An agent-based approach to care in independent living," in *Proceedings of the 1st International Joint Conference on Ambient Intelligence*, 2010, pp. 177–186.

[9] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv:1412.6980*, 2014.

[10] Z. Lu, T. K. Leen, Y. Huang, and D. Erdogmus, "A reproducing kernel hilbert space framework for pairwise time series distances," in *Proceedings of the 25th International Conference on Machine Learning*, 2008, pp. 624–631.

[11] S. S. Narayan and B. M. Marlin, "A survey on principles, models and methods for learning from irregularly sampled time series: From discretization to attention and invariance," *arXiv:2012.00168*, 2020.

[12] I. Silva, G. Moody, D. J. Scott, L. A. Celi, and R. G. Mark, "Predicting in-hospital mortality of ICU patients: The physionet/computing in cardiology challenge 2012," in *2012 Computing in Cardiology*, 2012, pp. 245–248.

[13] H. Song, D. Rajan, J. J. Thiagarajan, and A. Spanias, "Attend and diagnose: Clinical time series analysis using attention models," in *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, 2018, pp. 4091–4098.

[14] A. Trask, D. Gilmore, and M. Russell, "Modeling order in neural word embeddings at scale," in *Proceedings of the 32nd International Conference on Machine Learning*, 2015, pp. 2266–2275.

[15] K. Zhang, Y. Xue, G. Flores, A. Rajkomar, C. Cui, and A. M. Dai, "Modelling EHR timeseries by restricting feature interaction," *arXiv:1911.06410*, 2019.