Tensor Data Imputation by PARAFAC with updated Chaotic Biases by Adam Optimizer

Pooja Choudhary, Kanwal Garg

***Abstract:** The big data pattern analysis suffers from incorrect responses due to missing data entries in the real world. Data collected for digital movie platforms like Netflix and intelligent transportation systems is Spatio-temporal data. Extracting the latent and explicit features from this data is a challenge. We present the high dimensional data imputation problem as a higher-order tensor decomposition. The regularized and biased PARAFAC decomposition is proposed to generate the missing data entries. The biases are created and updated by a chaotic exponential factor in Adam's optimization, which reduces the imputation error. This chaotic perturbed exponentially update in the learning rate replaces the fixed learning rate in the bias update by Adam optimization. The idea has experimented with Netflix and traffic datasets from Guangzhou, China.*

***Keywords:** Tensor decomposition, PARAFAC, Adam optimization, Data imputation, etc.*

I. INTRODUCTION

The low quality and homogenous data require the verification and validation process, which is more time-consuming. Data mining approaches are used to manage a large amount of data in a particular context. Data mining schemes increases the quality of data with a large amount of data set. The meaningful pattern and rules are discovering in a large amount of data by exploration and analyzing them. In data mining approaches, various numbers of models and methods are considered to filter large data. The data mining methods are divided into different categories based on objective evaluation. The main categories are classification, estimation, prediction, clustering, association, and profiling. As we studied in the literature [1-11], various data mining models are proposed for Big Data analytics. The missing data problem is a setback for big data analytics. Regeneration of missing entries with similar latent features with the rest of the data poses a challenge. We here pick two domains to impute the big data. Netflix movie recommendation data and data gathered for intelligent transportation. Both datasets are collected continuously by corresponding companies and analyzing them for imputation is quite complicated for such a vast dataset. We convert the data into tensors and perform tensor operations to impute it. In the modern era, an intelligent transportation system (ITS) is used to collect large traffic data. Various mobile and stationary sensors are placed in the ITS configuration. The correct information collection in the transportation system represents the state of the system for the development of correct analysis. The collected data is not directly connected to the transportation system due to the presence of traditional data (traffic volume) and a new kind of data (Bluetooth and GPS data). The missing data problem has commonly arrived in a large traffic dataset. The missing data can degrade the performance of the intelligent transportation system. To improve the reliability of the traffic data collection and record the data mining approaches are used in state of the art. Previously various data imputation methods are proposed to solve the missing traffic data problem. The traffic data imputation problems are categorized into two sections: traffic volume imputation and traffic speed data estimation. Both traffic volume and traffic speed data have similar characteristics, like temporal and spatial correlation. Some imputation method of traffic volume data can be directly implemented to the traffic speed data. The traffic speed data can fluctuate more than traffic volume data. The traffic states, along with the closest roads, are correlated, which is implemented in the low dimension model. The matrix and tensor decomposition methods are used to obtain the missing data values [12-26]. The e-commerce market and digital platforms for entertainment offer a wide range of products and movies/songs etc. Selecting and browsing a movie of the tastes by the consumer is a tedious and cumbersome task. Also, the matching services to customer's tastes may increase the revenue of the company too. Due to this, many companies are nowadays interested in the recommendation system. After the Netflix competition for the movie recommendation system, the analysis of the user's pattern for a particular product/interest is more accepted by retailers. Several user events are analyzed to offer recommended movies from Netflix. The recommendation engine can't bear the loss of information as it may result in a negative recommendation too. Continuous pattern analysis also requires clean data without missing entries, but that is not possible in real-life data collection. Data imputation plays a significant role here. The high dimensional data is converted into the tensor as it inherits the high algebraic information than the matrix. The purpose of the selection of traffic and Netflix data is, these contain the temporal as well as spatial resolutions. The pattern is learned by tensor matrix factorization for a part of the data, and that learning is used to impute the missing entries. This paper aims to propose the advanced tensor factorization for data imputation.

Manuscript received on 23 December 2020 | Revised Manuscript received on 27 December 2020 | Manuscript Accepted on 15 January 2021 | Manuscript published on 30 January 2021.

***** Correspondence Author

Pooja Choudhary, Department of Electronics and Communication, Swami Keshvanand Institute of Technology, Management & Gramothan (SKIT), Jaipur (Rajasthan), India

Kanwal Garg, Department of Electronics and Communication, Swami Keshvanand Institute of Technology, Management & Gramothan (SKIT), Jaipur (Rajasthan), India

Contribution

Our contribution to the work is two-fold: regularized tensor factorization and Adam optimization with dynamic learning rate. The data used in our work is a tensor matrix that is a multidimensional matrix.

Several tensor factorization schemes are available like CANDECOMP/PARFAC, tucker, SVD, etc. [20, 22-28], but the PARAFAC method provides more stable tensor decomposition with faster convergence. The tensor is decomposed to rank one matrix. Few researchers used tensor decomposition for the high dimensional data imputations too [29] [30]. The PARAFAC decomposition still doesn't deal with the unexpected differences in the data values, which are counted as noise in the data. We propose here modified regularized PARAFAC, which decomposes the tensor into three factors [20] as well as into bias matrices for those tensor factors which avoid the fall into local optima.

We used the dynamic learning rate to update the bias values during Adam's optimization. The fixed learning rate of 0.001 is changed to a logistic mapped chaotic exponential decaying learning rate with every iteration.

Paper Organization

Further, in this article, we have discussed previous related work in section 2. Section 3 discusses the problem in data imputation and PARAFC tensor factorization. The proposed extended PARAFC tensor factorization for imputation has been discussed in section 4, which is followed by simulation results in the next section. We conclude the results in section 5.

II. LITERATURE REVIEW

State of the art schemes is utilized for the data mining process to achieve processed data. A high-quality data mining approach was presented in [1] by using the K-nearest neighbor algorithm. The K-nearest neighbor algorithm had a wide range of applications in terms of smart data obtained for Big data context. Multiple smart data packages were developed with the K-nearest neighbor approach. A predictive data mining framework was presented with the help of various up to date algorithms schemes like machine learning [2]. The machine learning algorithms improves the computational efficiency in removal of unwanted noise from the massive dataset. The missing data and noise are the key characteristics of real-world data. The Ghost algorithm was used to reconstruct the missing data from a large dataset that recover off period segments of missing data [3]. The ghost algorithm searched in a sequential manner dataset to achieve the data segment. A caching approach was also used to minimize the search space and improves the computational complexity to linear. In [4], various deep generative models were compared in terms of data mining. The key comparison among Variational Autoencoder (VAE) and Generative Adversarial Networks (GAN) was presented for imputing missing values problem. The imputation power of GAN and VAE approaches was improved by splitting methods to separate variables. A Graph Neural Network (GNN) model was enhanced for the missing data imputation (MDI) application. In the GNN approach, each edge of the graph shows the similarity between the different patterns. With the help of GNN, a Graph convolutional autoencoder was designed to reconstruct the complete dataset [5]. The adversarial loss and global information were included in the dataset during the reconstruction phase. A Distributed Neural Network (DNN) was designed for the imputation of missing value in terms of Big data context. The DNN was implemented in the spark and provided easy imputation [6]. The outcomes of the DNN data imputation method were compared with the K-nearest neighbor approach. They provided a better response. A self-developed competitive neural network Adaptive Response Theory 2 (ART2) was proposed for the data imputation [7]. The ensemble approach provided accurate results on intracluster non-missing value elements. The ART2 approach also improved imputation accuracy. The Big Data K-mean and Big Data Fuzzy imputation cluster-based methods were presented in [8]. Both the algorithms were provided better results than the simple eliminating faulty examples and easily implemented in the Spark Mlib configuration. Similarly, in [9], two algorithms were used to remove the noisy data from Big Data set. The two methods were Heterogeneous and Homogenous ensemble filters. These algorithms were formed with a combination of Deep learning classification models. The ensemble algorithms achieved smart datasets from Big Data set accurately and efficiently. The deep learning models required cleaned data to achieve better classification results. An MLClean approach was presented for the Big Data preprocessing to trained the Deep Learning model [10]. It provided data cleaning, unfair mitigation, and data sanitization. The MLClean approach was commonly used in Big Data analytics cases. The ROSEFW-RF algorithm was developed based on the Map-reduce function for imbalanced data mining [11]. The medical field data is imbalanced due to its number of parameters. To achieve balanced data, the relevant features, classes, category, and values are classified with the proposed ROSEFW-RF approach. The Random Forest classifier was used for the classification purpose and features evaluated with the evolutionary features weighting process. Various methods were proposed previously, which related to the tensor factorization with different types of datasets. In a study [12], the author provided the Bayesian CP factorization to handle the noisy tensor data. A non-deterministic model was developed using Bayesian CP factorization, which estimated the rank of CP automatically. A robust Bayesian generative model developed in [13] for tensor factorization. It provided the robustness to the boundary conditions and reduced the Gaussian noise, which minimizes the overfitting problem. The low-rank decomposition using a Bayesian model for multiway tensors data with missing observations was presented in [14]. This method was implemented for a large scale problem with linear configurations. Some real and benchmark datasets were tested with the above Bayesian model approaches. The Variation Bayesian (VB) techniques are commonly applied in the large scale models in the past. The single and coupled tensor factorization models were tested by the full Bayesian Inference using VB [15]. The full Bayesian Inference method was easily implemented to any large scale model. A fully Bayesian Probabilistic Matrix Factorization (PMF) model was presented in [16].

The overall model parameters and hyperparameters controlled the model capacity. The Bayesian PMF model was trained by the Markov Chain Monte Carlo Method by using the Netflix dataset and provided higher predicted accuracy. A factor-based algorithm was presented for the classification application known as tensor factorization.

The fully Bayesian Probabilistic Tensor Factorization (PTF) was used to analyze the large scale dataset [17].

Some real-world problems of classification were analyzed by the Bayesian PTF model with accurate outcomes. A Bayesian Gaussian CANDECOMP/PARAFAC (BGCP) decomposition model was used to impute the multidimensional imbalance traffic data [18]. In tensor representation with BGCP, the accurate outcomes were found in the third-order case for both random missing and fiber missing traffic data scenarios. The multiple metrics and tensors combined joints were presented by the Bayesian Multi-tensor Factorization model [19]. The Bayesian MTF model analyzed the total number of data collection factors. A multidimensional EEG dataset was analyzed by the Bayesian Tensor Factorization model [20]. The noninformative EEG signals were removed from the BTF method, which improved the classification response in the medical research field. The BTF also eliminated the white noise present in the EEG dataset. A fully Bayesian system was presented for the automatically learning parameters of missing values traffic data model with Variational Bayes (VB) [21]. The data was collected from the urban traffic speed data set collected in Guangzhou, China. The two methods were tested on the traffic data imputation. The Bayesian Augmented Tensor Factorization (BATF) provided better accuracy in terms of missing values imputations.

A three procedure framework was proposed for the missing traffic data recovery with traffic patterns recognition. The key latent features were learned by the truncated singular value decomposition (SVD) and removed the noise. These latent features were applied to the tensor decomposition, and missing data were evaluated with the combination of SVD –tensor decomposition (STD). The proposed STD approach provided higher accuracy in terms of missing traffic data [22]. A matrix factorization technique based on a K-nearest neighbor method was proposed for the Netflix missing data recovery [23]. The matrix factorization based KNN model provided higher accuracy, and it can be easily implemented on real-world datasets. The collaborative filtering approaches were applied for the data filter in [24, 25]. The unique properties of implicit feedback datasets were recognized by a factor model [25]. A scalar optimization algorithm (latent factor) was also suggested by the authors to scale the large size data linearly. Kingma et al. presented a method for stochastic optimization (Adam) based on the adaptive calculations of lower-order moments [26].

Table 1: List of studied approaches of data mining and tensor factorization

Research paper	Techniques	Advantages	Improved parameters
Data Mining and imputation approaches			
1	KNN	Achieved smart data in Big data context	70% accuracy
2	Machine learning models	Effective performance among all datasets	NA
3	Ghost algorithm	Evaluated off period segments data	18% higher F-Score
4	GAN and VAE	Improved missing data imputation	90% accuracy
5	Graph Convolutional Network	Provided large missing data values in a short period of time	NA
6	Distributed Neural Network (DNN)	Execution time is fast	70% speedup
7	ART2	It can deal with outlier and improved imputation accuracy	25% missing imputation accuracy
8	Big Data K-mean and Big Data Fuzzy	Easily implemented in Spark ML lib	98% accuracy
9	Heterogeneous and Homogenous ensemble filters	Efficiently obtained the smart dataset from Big Data	65% and 80% noise removal accuracy
10	MLClean method	Train accurate and fair models	NA
11	ROSEFW-RF	Good balance and classification of classes	NA

Bayesian model approaches			
12	Bayesian CP Factorization	Automatically determine the rank of the incomplete tensor	Execution time improved
13	Bayesian Robust Tensor Factorization (BRTF)	It provided robustness to the outliers	NA
14	Scalable Bayesian low-rank decomposition	It solved the large scale problem and provided linearity to the observed tensors	Minimizes reconstructing errors
15	Bayesian Tensor Factorization (BTF)	Implemented on large models	90% accuracy with lesser execution time
16	BPMF	Easily trained with the Markov chain model and tested on the 100 million video dataset Netflix	NA
17	BPTF	Analyzed large scale dataset	Higher latent features learning rate
18	Extended BPMF model (BGCP)	Impute multidimensional imbalance traffic data	NA
19	BMTF	Joint multiple metrics and tensor	NA
20	BTF	Tested multidimensional EEG dataset and accurately impute missing data in the large domain	Minimize the standard error
21	BATF	Efficient impute high traffic missing data	NA
Tensor factorization approaches			
22	SVD with tensor decomposition (STD)	Extract efficient latent features from missing traffic data	Easily trained the large scale model
23	KNN	Higher accuracy in a real-world dataset	Impute Netflix data
24	Filter model	Removal of noise in a large dataset	NA
25	Factor model and scalar optimization algorithm	Implicit feedback dataset analyzed	NA
26	ADAM	A stochastic approach for data classification	NA

III. PROBLEM STATEMENT

3.1 Problem Statement

Due to vast applications of tensor, its decomposition for multidimensional data has been the researcher's interest. Their work can be broadly classified into two categories: decomposition by CANDECOMP/PARAFAC (Canonical Decomposition/ Parallel Factor), also referred to as CP and tucker factorization. Both are higher-order principal component analysis (PCA). The convex hull analysis in [36] proves the CP decomposition is better than tucker decomposition on the criteria of good fit analysis. The PARAFAC itself takes a long time in reaching the local minima solution and also has the probability to stuck in the local minima. To avoid it with local optima, we regularized the PARAFAC Tikhonov regularization factor. Still, this form of PARAFAC only considers the latent features, but the data is Spatio-temporal. So, we need to add the biases for all the three dimensions of the data, which can extract the temporal features too.

Besides the decomposition part, the rank selection is also uncertain in the tensor decomposition. The Bayesian approach has also been suggested by a few researchers for rank selection [12-21]. But these focused either on prediction problems or factorization with missing data. Whereas we couldn't find it suitable for data imputation problem as Bayesian optimization works preferably for automatic Rank detection for higher-order low-rank factorization. In the imputation application of CP, the normalized imputation error increases with the increase in rank. We have tested the scheme of regularized augmented PARAFACfor various ranks from 1-10. A plot in figure 1 demonstrates that rank 1 is suitable for data imputation scheme and imputed data for this rank by PARAFAC, and regularized augmented PARAFAC is shown in figure 1(b). It supports our convention that PARAFAC lacks in providing a global solution for missing data generation.

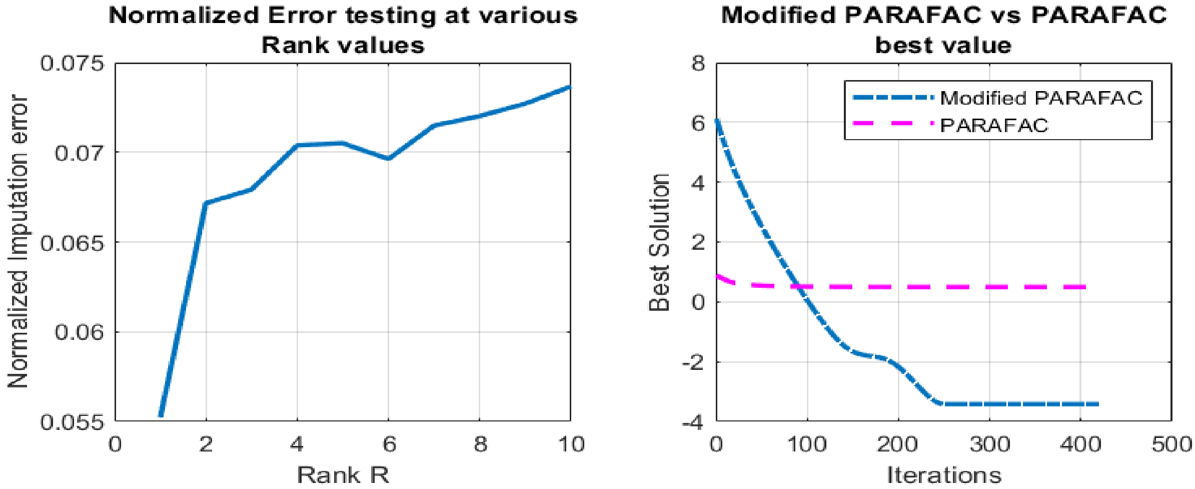


Fig. 1. (a) Rank Selection for proposed regularized augmented PARAFAC (b) comparison of the iterated best value in each iteration of PARAFAC and suggested PARAFAC decomposition with Adam optimization

Missing Data Generation

A tensor $\mathcal{X}$ and matrix Y are computed as $\mathcal{X} = [\![A, B, C]\!]$ and $Y = AV^T$. The matrices $A \in \mathbb{R}^{I\times R}$, $B \in \mathbb{R}^{J\times R}$, $C \in \mathbb{R}^{K\times R}$, and $V \in \mathbb{R}^{M\times R}$ are generated by the random entries drawn from the standard normal distribution. A new matrix $\bar{\mathcal{X}} \in \mathbb{R}^{I\times J\times K}$ is generated by a random setting of $M\%$ of the entries of $\mathcal{X}$ to be missing. So $\bar{\mathcal{X}} = W * \mathcal{X}$, here binary tensor is represented by the W which has an equal dimension of tensor $\mathcal{X}$ and $w_{i,j,k} = 0$ for the set $\mathcal{X}_{i,j,k}$ to missing. Figure 2 shows the tensor with random missing entries.

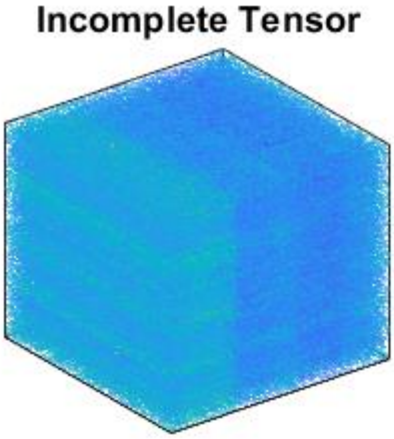


Fig. 2. Arrangement of incomplete tensor $\bar{\mathcal{X}}$by random indexing.

IV. PROPOSED TENSOR FACTORIZATION

4.1 Regularized-PARAFAC

As discussed above, we factorize the tensor matrix by the PARAFAC method. For the data imputation in our work, the data is of third-order tensor $\mathcal{X} \epsilon \mathbb{R}^{I\times J\times K}$ and as for large rank ($R > 10$), the PARFAC decomposition can be shown as in equation 1 [29][33].

$$\mathcal{X}_{itj} \approx [\![D, J, F]\!] := \textstyle\sum_{r=1}^{R} D_{ir} J_{tr} F_{jr} \qquad (1)$$

Equation 1 holds for the sufficiently large rank R, but since in our case, we are interested in the decomposition up to $R = 1$ [13]. The tensor matrix can be decomposed into N factors of rank 1 as:

$$\mathcal{X} \epsilon\, a^{(1)} \circ a^{(2)} \circ a^{(3)} \circ \ldots .\, a^{(N)} \qquad (2)$$

Where $\circ$ represents the cross product. For the third-order tensor, it is $\mathcal{X} \epsilon\, a \circ b \circ c$. The conviction of decomposition of equation 1 to the first-order tensor is:

$$\mathcal{L}(\mathcal{X}; D, J, F) = \frac{1}{2} \|\mathcal{X} - [D, J, F]\|_F^2 \qquad (3)$$

The $\|\cdot\|_F^2$ Represents the Frobenius norm. This one is the non-convex optimization problem and can be solved iteratively as

$$\min_{D,J,F} \mathcal{L}(\mathcal{X}; D, J, F) \qquad (4)$$

The equation 4 has an issue of local optimum solution due to non-uniform scaling of factors [33], so Paatero [28] suggested the addition of a Tikhonov regularization factor in equation 4 which resolved the issue of many local solutions in equation 3 and updated it as

$$\mathcal{L}_P(\mathcal{X}; D, J, F) = \frac{1}{2} \|\mathcal{X} - [D, J, F]\|_F^2 + \frac{\lambda}{2}(\|D\|_F^2 + \|J\|_F^2 + \|F\|_F^2)$$

(5)

Where λ is a regulation parameter with $\lambda > 1$. In our work, we are using the data set from China Traffic Data[13]. This kind of tensor data has three factors for road segments, the number of time slots, and recorded time intervals. Equation 5 maps the combined rating for users and content in the latent space, but it is not always the case. The temporal dependency on the data is not examined in that, due to which the imputation might be less correct. This biasing for user and content must be included in equation 5 [29]. The augmented equation 5 is written in equation 6. This way, the PARAFAC factorization is called augmented PARAFAC.

$$\mathcal{L}_A(\mathcal{X}; D, J, F) = \frac{1}{2} \|\mathcal{X} - \mu - \emptyset_D - \theta_J - \eta_F - [D, J, F]\|_F^2 + \frac{\lambda}{2}(\|D\|_F^2 + \|J\|_F^2 + \|F\|_F^2 + \|\emptyset_D\|_F^2 + \left\|\theta_J\right\|_F^2 + \|\eta_F\|_F^2)$$

(6)

This can be further simplified as

$$\mathcal{L}_A(\mathcal{X}; D, J, F) = \frac{1}{2}\Big(\|\chi\|_F^2 + \|\mu\|_F^2 + \|\phi_D\|_F^2 + \left\|\theta_J\right\|_F^2 + \|\eta_F\|_F^2 + \|[D, J, F]\|_F^2\Big) - \langle(\chi, \mu)\rangle - \langle(\chi, \phi_D)\rangle - \langle(\chi, \theta_J)\rangle - \langle(\chi, \eta_F)\rangle - \langle(\chi, [D, J, F])\rangle + \langle(\mu, \phi_D)\rangle + \langle(\mu, \theta_J)\rangle + \langle(\mu, \eta_F)\rangle + \langle(\mu, [D, J, F])\rangle + \langle(\phi_D, \theta_J)\rangle + \langle(\phi_D, \eta_F)\rangle + \langle(\phi_D, [D, J, F])\rangle + \langle(\theta_J, \eta_F)\rangle + \langle(\theta_J, [D, J, F])\rangle + \langle(\eta_F, [D, J, F])\rangle + \frac{\lambda}{2}(\|D\|_F^2 + \|J\|_F^2 + \|F\|_F^2 + \|\emptyset_D\|_F^2 + \left\|\theta_J\right\|_F^2 + \|\eta_F\|_F^2)$$

(7)

Where $\emptyset_D$, θ_J, η_F are the bias factors for the three tensor dimensions, μ is the average of all three factors [31]. The biases are the deviation of each factor from the average μ. The proposed tensor $\mathcal{X}$ is shown in figure 3. The factor matrix of the tensor can also be represented as $\hat{\mathcal{X}}_{i,j,t} = \mu + \emptyset_D + \theta_J + \eta_F + [D, J, F]$.

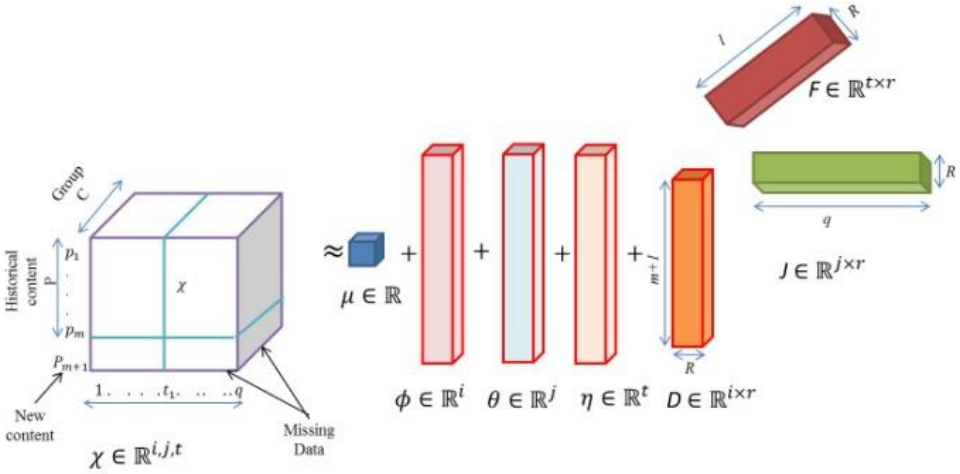

Group
C
Historical content
P
p_1
p_m
P_{m+1}
χ
New content
$1. \ . \ . \ . t_1. \ .. \ ..q$
Missing Data
$\chi \in \mathbb{R}^{i,j,t}$
$\approx$
$\mu \in \mathbb{R}$
+
$\phi \in \mathbb{R}^i$
$\theta \in \mathbb{R}^j$
$\eta \in \mathbb{R}^t$
$m+l$
R
$D \in \mathbb{R}^{i \times r}$
l
R
$F \in \mathbb{R}^{t \times r}$
R
q
$J \in \mathbb{R}^{j \times r}$

Fig. 3. Proposed augmented PARAFAC factorization for 3rd order tensor

The proposed modified-PARAFACis an iterative process, and equation 7 is minimized by gradient descent optimization [13], Alternating least square method [31] recently. But these methods are easily stuck in local minima and less rate of convergence. The advancement in deep learning is dependent upon the fast and global optimized Adam optimization [35]. We factorize the tensor $\mathcal{X}$ by Adam optimization. The bias is also updated iteratively in each stochastic step. These are initialized at some random value between 0 and 0.1. Though the initial feeding point for these biases can be random, however for our application, we set $b\epsilon[0,0.1]$, where b represents the biases. Biases are updated in each iteration of Adam's optimization. Equation 6 is the objective function of adam optimization, and the error thus calculated is used to update the bias values. We multiply the error from equation 6 by an exponentially decreasing regularization parameter, which reduces with iteration to minimize the error. To add the chaotic perturbation in error, we multiply it with a less random factor, which is calculated by a logistic map. The biases are updated as:

$$\emptyset_{D,t} = \emptyset_{D,t-1} + (\mathcal{X}_{itj} - \hat{\mathcal{X}}_{i,j,t}) * x_n * e^{-\alpha\frac{t}{T}} \qquad (8)$$

Here $(\mathcal{X}_{itj} - \hat{\mathcal{X}}_{i,j,t})$ is the deviation in the factors, α is a constant whose values if fixed to 20 in our case. t and T represent the current iteration and maximum iteration in the adam optimization. x_n is the perturbed parameter by logistic mapping which is calculated as $x_n = rx_n(1-x_n), n = 0,1,2,3 \ldots.$. The r is a system parameter $(0,4] and\, x_n \epsilon[0,1]$. This way θ_J and η_F are also updated.

The pseudo-code for the update of biases is shown in algorithm1.

Algorithm 1: Pseudo Code for the tensor biases update iteratively

```
Input: $\mathcal{X}$, $[D, J, F]$
Output: $\emptyset_D$, $\theta_J$, $\eta_F$, $\mu$
1. Initialize the $\emptyset_D$, $\theta_J$, $\eta_F$ with uniform random numbers in between 0 and 0.1
2. Calculate the average $\mu = \frac{1}{\Omega}\sum_{(i,j,t)\epsilon\Omega} x_{i,j,t}$
3. While not converge do
4.     For $i\epsilon(1, n1)$
5.         Update
           $\emptyset_{D,t} = \emptyset_{D,t-1} + (\mathcal{X}_{itj} - \hat{\mathcal{X}}_{i,j,t}) * x_n * e^{-\alpha\frac{t}{T}}$
6.     end for
7.     for j$\epsilon(1, n2)$
8.         Update
           $\theta_{J,t} = \theta_{D,t-1} + (\mathcal{X}_{itj} - \hat{\mathcal{X}}_{i,j,t}) * x_n * e^{-\alpha\frac{t}{T}}$
9.     end for
10.    for t$\epsilon(1, n3)$
11.        Update
           $\eta_{F,t} = \eta_{D,t-1} + (\mathcal{X}_{itj} - \hat{\mathcal{X}}_{i,j,t}) * x_n * e^{-\alpha\frac{t}{T}}$
12.    end for
13. end while loop
```

V. EXPERIMENTS

5.1 Datasets

Traffic Dataset

The Traffic dataset used in this work is released by the communication commission of Guangzhou municipality of China. The dataset provides speed data of the 214 road segment. It contains 61 days traffic activity of urban areas from 1 August 2016 to 30 September 2016. During each day144-speed value, observation is provided for each road segment with 10 minutes time window aggregation.

Netflix Dataset

This dataset is collected by Netflix, which shows the distribution of all the ratings Netflix achieved between October 1998 to December 2005. It consists of a total of 100,480,507 ratings from 480,189 randomly chosen users on the 17,770 movie titles. Netflix also provides training data for validation purposes which consists of 1,408,395 ratings. The Netflix dataset provides a testset containing 2,817,131 movie pairs with ratings. The latest movie pairs ratings are selected from a recent subset of users in the training dataset.

5.2Evaluation

We have followed the random missing entries scheme for missing values generation. After creating the missing values for 10%, 20%, and 50%, we use rank 1 from figure 1 for the tensor data completion. The proposed decomposition solution with bias update is compared with the state of the art schemes. We compared the imputation with Tucker decomposition [30], Bayesian inference [29] for the random missing data. The Netflix dataset is also checked for only random missing entries. The proposed scheme is also evaluated for gradient descent (GD) and Adam optimization for the optimal solution of equation 6. The evaluation of results is statistically done based on mean absolute error (MAE), mean absolute percentage error (MAPE), and root means square error (RMSE). If the y_i and z_i are estimated and true missing values, then these performance measures are defined as:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - z_i|$$

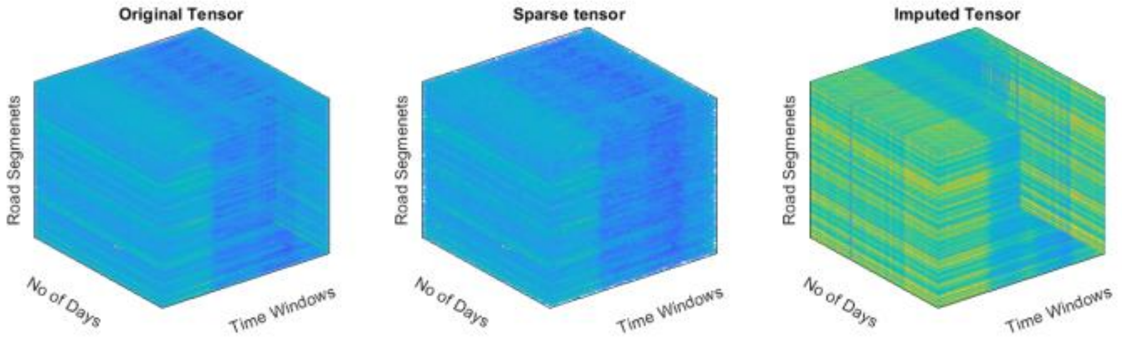
$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\frac{|y_i - z_i|}{y_i}$$

$$RMSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - z_i)^2$$

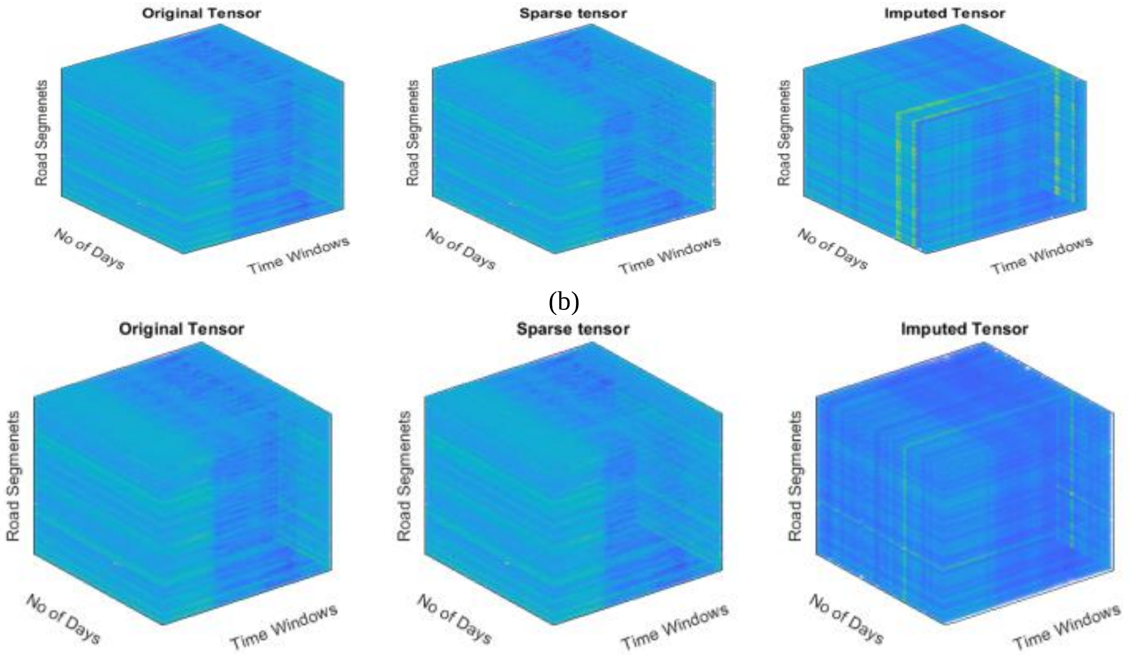
Netflix dataset and traffic datasets are evaluated by the proposed scheme and compared with states of the art SVD combined tucker decomposition [30], BATF [29], BGCP [20], BCPF [21], and modified PARAFAC variants. The schemes in [20], [21],[29] have common Bayesian Gaussian tensor factorization. The CANDECOMP/PARAFAC factors are trained by Bayesian optimization. The [21] presented the hyperparameter of automatic rank determination. In contrast, the solution doesn't differ significantly from other variants in [20] and [29] as the researchers only used the higher number of hyperparameters for optimization. However, the Bayesian optimization of factorization parameters performed well over Tucker-SVD decomposition [30]. Table 2 shows the performance comparison of the proposed data imputation scheme for random missing entries for traffic datasets. It can be observed from table 2 that performance is somewhat good at lower missing ratio; however, the change is not significant with increasing missing values. The dynamic, chaotic learning rate and modified regularized tensor decomposition lead to superior performance than state of the art methods. Figure 4 represents the imputed tensor for different random missing ratios for the traffic dataset by the proposed scheme. The proposed scheme doesn't get affected by the change in the missing rate like other schemes.

Table 2: Comparative Performance Evaluation for randomly Missing Entries for Traffic Dataset

		10% Missing			30% Missing			50% Missing		
		MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE
	Proposed Imputation	3.449	0.0826	3.6101	3.4537	0.0831	3.60	3.4707	0.0834	3.6116
Variants of Proposed Scheme	Modified PARAFAC-E-Biased	3.459	0.0833	3.695	3.462	0.0848	3.784	3.5103	0.08904	3.819
	Modified PARAFAC-Adam optimization	3.5361	0.0872	3.70	3.5380	0.0896	3.79	3.780	0.0907	3.829
	Modified PARAFAC-GD Optimization	3.8101	0.0920	3.7027	3.8183	0.0928	3.7260	3.8190	0.0929	3.7190
States of the art schemes	SVD-Tucker [30]	-	-	-	4.3821	-	6.1060	4.3794	-	6.1007
	BATF [29]	-	0.0825	3.5745	-	0.0834	3.5969	-	0.0841	3.6290
	BGCP [20]	-	0.0823	3.5614	-	0.0827	3.6340	-	0.0833	3.6009
	BCPF [21]	-	0.0832	3.5988	-	0.0843	3.5775	-	0.0852	3.6784



(a)



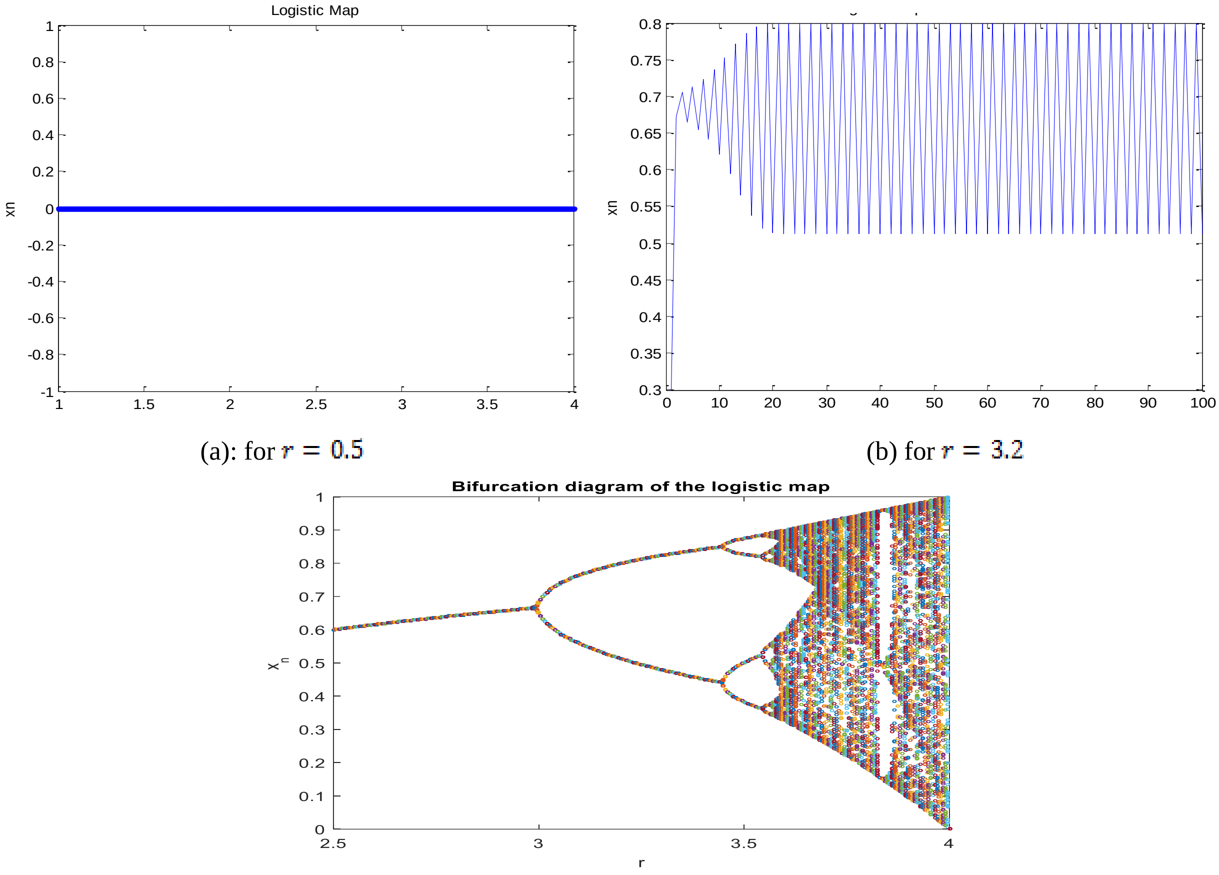
(b)

(c)

Fig. 4. The imputed tensor with random missing entries percentage (a) 50% (b) 30 % and (c) 10%

The bias to generate the missing values is updated by gradient descent (GD) optimization in [30]; however, as discussed in the previous section, the Adam optimizer is better than GD. We have followed the update as in algorithm 1. The SVD-Tucker decomposition proposed the constant learning rate to update the bias. Still, it has been proven recently that for the sparse dataset, the adaptive learning rate is better than a constant learning rate. The learning rate can be adaptive dependent upon time, step size, or exponentially changing with iteration. Other Deep learning toolboxes like Keras also provide the feature to use the adaptive learning rate during training. We have perturbed the exponentially decaying learning rate by logistic mapping. It is a linear mapping with variable $x_n = rx_n(1 - x_n), n = 0,1,2,3 \ldots.$. The r is a system parameter $(0,4] and\, x_n \epsilon [0,1]$. The logistic mapping shows different behavior for different values of r. Figure 5 shows the stationary, periodic, and complete bifurcation diagram for the logistic map. Figure 5(a) and 5(b) shows the behavior of logistic mapping if the system parameter r is less than 1 and $3 \leq r < 1 + \sqrt{6}$ respectively. For $0 \leq r < 1$, updated biases with this logistic map value can lead to premature convergence of the training. Conclusively, data imputation won't be similar to extracted latent features of the rest of the data. The non-diminishing oscillatory behavior (figure 5(b)) also doesn't converge at all. The chaotic behavior starts beyond 3.56994, and figure 5(c) shows the logistic mapping for a complete range of $r \in (0,4]$. The area between $r \in (3.54409,4]$ is the stable oscillations area, and convergence in optimization can be achieved in this area. The solid lines in figure 5(c) point to the stable solution. The more uniform is the bias, the less deviation is observed in variance for a dimension. Just to recall, the three biases are for the three dimensions of the Spatio-temporal data. The size of a bias matrix is equal to the size of the decomposed factor matrices. The first row in figure 6 shows the uniformity comparison for the bias matrices for the final selected solution for a 50% missing ratio. We have used the ternary plot since we have 3-dimensional tensor. The final updated bias for the minimum gradient in equation 7 is recorded for plotting. The vertices of the ternary plot represent the updated bias by Adam optimized modified regulated PARAFAC with a dynamic learning rate, with a fixed learning rate and proposed PARAFAC segmentation by GD optimization. These three plots are for each dimension of the data. The dots inside the plots are the bias values that arrange themselves in an elliptical shape with alignment towards the proposed scheme. The elliptical shape is formed due to uniformity in the bias values; the least is the standard deviation; more is the uniformity.

The alignment towards proposed scheme vertex indicates the clustering of biases is greatly affected by larger bias magnitudes generated by the proposed scheme. However, the uniformity of bias$\theta_j < \emptyset_D \& \eta_F$, as indicated in figure 6(b) for bias for the number of days. The second row in figure 7 has the plots for imputed values and original entries. We have plotted these curves for a few samples. The sparse tensor's entries are also plotted in 6(d) whose entries are all zeros for these samples.



(a): for $r = 0.5$

(b) for $r = 3.2$

(c) for $r \in (0,4]$

Fig. 5.Stationary, periodic and bifurcation behavior of logistic mapping

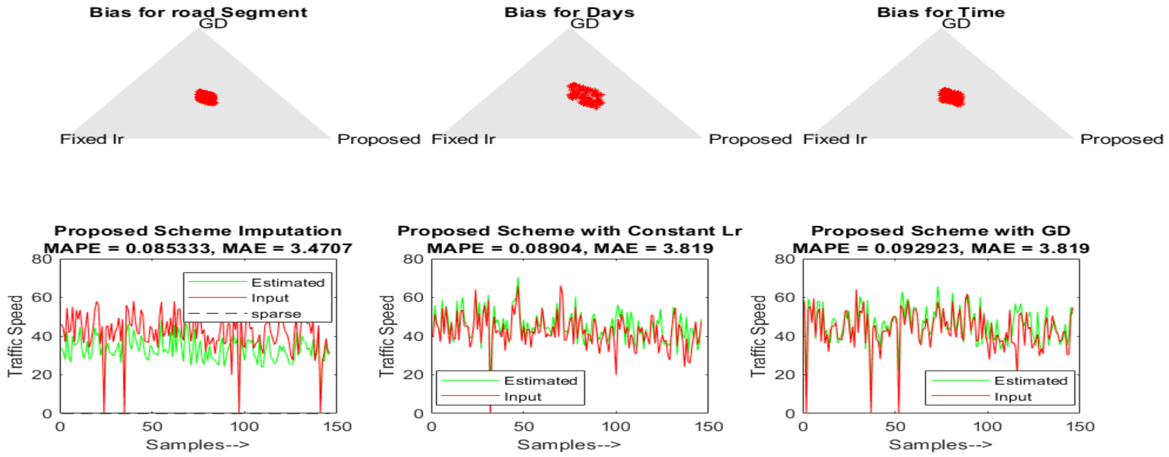


Fig. 6. Ternary Plots between final bias values after training by proposed chaotic exponential updated bias, updated bias with constant learning rate and Gradient Descent updated (a) bias for road segment (b) bias for traffic speed at various Days (c) Bias for the 10 minutes time windows. The second row of the comparative plot for original and estimated traffic data with (d) proposed tensor decomposed imputation scheme (e) proposed decomposition scheme with a fixed learning rate and (f) proposed scheme with Gradient descent optimization

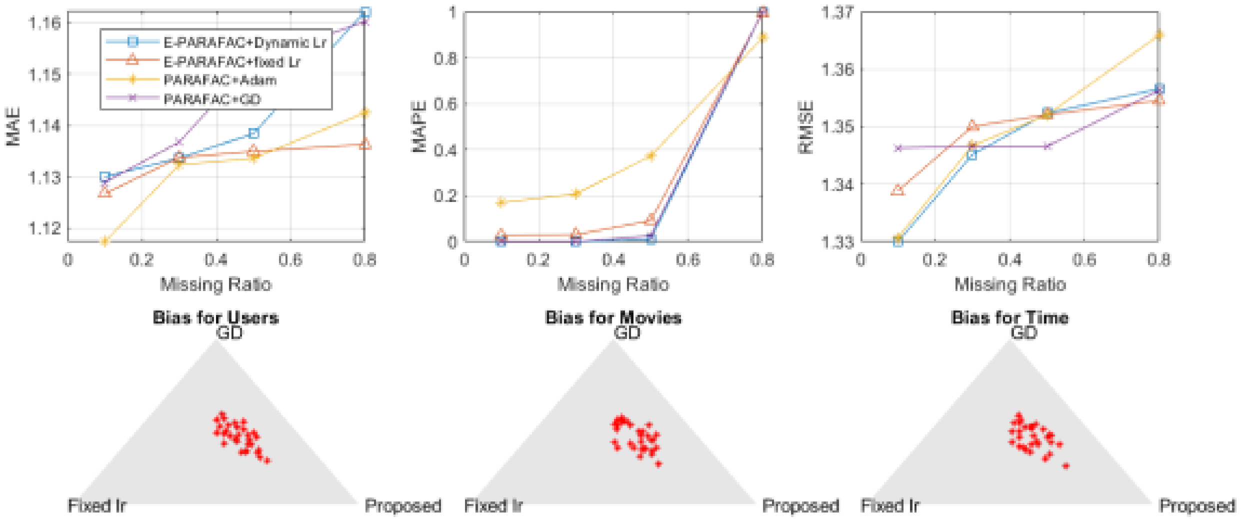


Fig. 7. Performance comparison for the Netflix data imputation

For the Netflix dataset, we have compared the proposed scheme with its three other variants only due to unavailability of state of the art schemes results on this data. The figure shows the performance plot comparison for the Netflix dataset. This dataset has primarily tested for the Netflix prize competition for recommending the most suitable movies to the users. The modified PARAFAC with the proposed dynamic learning rate has minimum MAPE and RMSE. The MAPE has shown no significant change with the increase in the missing entries, but with 80% missing data, all variants have degraded. However, no such behavior is recorded for MAE and RMSE curves. The proposed scheme has the least MAPE and RMSE values at various missing ratios. This dataset also strengthens the theory that Adam optimization outperforms the PARAFAC with gradient descent optimization. The bias for each dimension of the Netflix data is shown in the second row of figure 7.

VI. CONCLUSION

The data imputation is the first step in the data denoising and to process the data with deep learning. In this research article, we have profoundly compared the performance of imputed tensor data with state of the art schemes. The Spatio-temporal real-world tensor data can be decomposed into the low-rank matrix for imputation using two kinds of decomposition primarily: CD/PARAFAC and Tucker decomposition. We compared the proposed modified PARAFAC with a dynamic learning rate with the latest work of Tucker-SVD decomposition on the Traffic dataset. It has been observed that our method is able to reduce the mean absolute error by 20% for the 50% missing entries in the data. Few other researchers have focused on the Bayesian learning training for the tensor decomposition. It has also been suggested previously that automatic rank determination leads to a better generation of missing entries. Our dynamic Adam optimization scheme has powered the modified PARAFAC with a 0.48% reduced root mean square error for 50 % missing data. We have tested the algorithm on Netflix movies dataset too. The size of the dataset has backed the performance due to our hardware constraint, but it has been noticed that dynamically learned modified PARAFAC had won the competition. In the next part of our research, we will test the performance on Hadoop and scala as the large size of the data created the havoc in the testing. We will also cluster the imputed tensor and semantically compared the imputation.

REFERENCES

1. Wu, B., Cheng, W. H., Zhang, Y., Huang, Q., Li, J., & Mei, T. (2017). Sequential prediction of social media popularity with deep temporal context networks. *arXiv preprint arXiv:1712.04443*.
2. De, S., Maity, A., Goel, V., Shitole, S., & Bhattacharya, A. (2017, April). Predicting the popularity of instagram posts for a lifestyle magazine using deep learning. In *2017 2nd International Conference on Communication Systems, Computing and IT Applications (CSCITA)* (pp. 174-177). IEEE.
3. Das, S., Syiem, B. V., & Kalita, H. K. (2014). Popularity Analysis on Social Network: A Big Data Analysis. *International Journal of Computer Applications*, *975*, 27-31.
4. Stieglitz, S., Mirbabaie, M., Ross, B., & Neuberger, C. (2018). Social media analytics–Challenges in topic discovery, data collection, and data preparation. *International journal of information management*, *39*, 156-168.
5. Birjali, M., Beni-Hssane, A., & Erritali, M. (2017). Analyzing social media through big data using infosphere biginsights and apache flume. *Procedia computer science*, *113*, 280-285.
6. Hu, W., Singh, K. K., Xiao, F., Han, J., Chuah, C. N., & Lee, Y. J. (2018, February). Who will share my image? Predicting the content diffusion path in online social networks. In *Proceedings of the eleventh ACM international conference on web search and data mining* (pp. 252-260).
7. Yamaguchi, K., Berg, T. L., & Ortiz, L. E. (2014, November). Chic or social: Visual popularity analysis in online fashion networks. In *Proceedings of the 22nd ACM international conference on Multimedia* (pp. 773-776).
8. Shulman, B., Sharma, A., & Cosley, D. (2016). Predictability of popularity: Gaps between prediction and understanding. *arXiv preprint arXiv:1603.09436*.
9. Van Canneyt, S., Leroux, P., Dhoedt, B., & Demeester, T. (2018). Modeling and predicting the popularity of online news based on temporal and content-related features. *Multimedia Tools and Applications*, *77*(1), 1409-1436.
10. Uddin, M. T., Patwary, M. J. A., Ahsan, T., & Alam, M. S. (2016, October). Predicting the popularity of online news from content metadata. In *2016 International Conference on Innovations in Science, Engineering and Technology (ICISET)* (pp. 1-5). IEEE.
11. Shestakov Andrey, EngelbertMephuNguifo (2014), “Predicting web-page popularity with Machine Learning and Heuristic Time-Series Prediction approaches”, ECML/PKDD Discovery Challenge on Predictive Web Analytics, Nancy, France, September ,, pp 1-5.
12. Vanwinckelen, G., & Meert, W. (2014). Predicting the popularity of online articles with random forests. In *ECML/PKDD Workshop on Predictive Web Analytics, Date: 2014/09/19-2014/09/19, Location: Nancy, France* (pp. 1-6).

13. Hoang, M. X., Dang, X. H., Wu, X., Yan, Z., & Singh, A. K. (2017, April). GPOP: Scalable group-level popularity prediction for online content in social networks. In *Proceedings of the 26th International Conference on World Wide Web* (pp. 725-733).
14. Kieu, B. T., Ichise, R., & Pham, S. B. (2015). Predicting the popularity of social curation. In *Knowledge and systems engineering* (pp. 413-424). Springer, Cham.
15. Hu, Y., Hu, C., Fu, S., Shi, P., & Ning, B. (2016). Predicting the popularity of viral topics based on time series forecasting. *Neurocomputing*, *210*, 55-65.
16. Aghababaei, S., & Makrehchi, M. (2016, October). Mining social media content for crime prediction. In *2016 IEEE/WIC/ACM International Conference on Web Intelligence (WI)* (pp. 526-531). IEEE.
17. Moro, S., Rita, P., & Vala, B. (2016). Predicting social media performance metrics and evaluation of the impact on brand building: A data mining approach. *Journal of Business Research*, *69*(9), 3341-3351.
18. Barnard, S. T. (1995, December). PMRSB: Parallel multilevel recursive spectral bisection. In *Proceedings of the 1995 ACM/IEEE conference on Supercomputing* (pp. 27-es).
19. Karypis, G., & Kumar, V. (1998). Multilevelk-way partitioning scheme for irregular graphs. *Journal of Parallel and Distributed computing*, *48*(1), 96-129.
20. Chen, X., He, Z., & Sun, L. (2019). A Bayesian tensor decomposition approach for spatiotemporal traffic data imputation. *Transportation research part C: emerging technologies*, *98*, 73-84.
21. Zhao, Q., Zhang, L., & Cichocki, A. (2015). Bayesian CP factorization of incomplete tensors with automatic rank determination. *IEEE transactions on pattern analysis and machine intelligence*, *37*(9), 1751-1763.
22. Chen, Y. L., Hsu, C. T., & Liao, H. Y. M. (2013). Simultaneous tensor decomposition and completion using factor priors. *IEEE transactions on pattern analysis and machine intelligence*, *36*(3), 577-591.
23. Sorber, L., Van Barel, M., & De Lathauwer, L. (2013). Optimization-based algorithms for tensor decompositions: Canonical polyadic decomposition, decomposition in rank-(L_r,L_r,1) terms, and a new generalization. *SIAM Journal on Optimization*, *23*(2), 695-720.
24. Allman, E. S., Jarvis, P. D., Rhodes, J. A., & Sumner, J. G. (2013). Tensor rank, invariants, inequalities, and applications. *SIAM Journal on Matrix Analysis and Applications*, *34*(3), 1014-1045.
25. Hillar, C. J., & Lim, L. H. (2013). Most tensor problems are NP-hard. *Journal of the ACM (JACM)*, *60*(6), 1-39.
26. Goldfarb, D., & Qin, Z. (2014). Robust low-rank tensor recovery: Models and algorithms. *SIAM Journal on Matrix Analysis and Applications*, *35*(1), 225-253.
27. Maehara, T., Hayashi, K., & Kawarabayashi, K. I. (2016, February). Expected tensor decomposition with stochastic gradient descent. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence* (pp. 1919-1925).
28. Paatero, P. (2000). Construction and analysis of degenerate PARAFAC models. *Journal of Chemometrics: A Journal of the Chemometrics Society*, *14*(3), 285-299.
29. Chen, X., He, Z., Chen, Y., Lu, Y., & Wang, J. (2019). Missing traffic data imputation and pattern discovery with a Bayesian augmented tensor factorization model. *Transportation Research Part C: Emerging Technologies*, *104*, 66-77.
30. Chen, X., He, Z., & Wang, J. (2018). Spatial-temporal traffic speed patterns discovery and incomplete data recovery via SVD-combined tensor decomposition. *Transportation research part C: emerging technologies*, *86*, 59-77.
31. Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, *42*(8), 30-37.
32. Charlier, J., & Makarenkov, V. (2019). VecHGrad for Solving Accurately Complex Tensor Decomposition. *arXiv preprint arXiv:1905.12413*.
33. Koren, Y. (2009, June). Collaborative filtering with temporal dynamics. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 447-456).
34. Hu, Y., Koren, Y., & Volinsky, C. (2008, December). Collaborative filtering for implicit feedback datasets. In *2008 Eighth IEEE International Conference on Data Mining* (pp. 263-272). Ieee.
35. Kingma DP, Adam BJ (2015).,"A method for stochastic optimization:,arXiv preprint arXiv:1412.6980. pp 1-15.
36. E. Ceulemans and H. A. L. Kiers, Selecting among three-mode principal component models of different types and complexities: A numerical convex hull based method, British J. Math. Statist. Psych., 59 (2006), pp. 133–150.