POLITECNICO DI TORINO
Repository ISTITUZIONALE

Tensor Neural Networks Based Optical Fault Localization

Original
Tensor Neural Networks Based Optical Fault Localization / Gao, Y., Rottondi, C., Sgambelluri, A., Valcarenghi, L., Paolini, E., Guo, B., Huang, S., Bianco, A.. - ELETTRONICO. - (2026), pp. 1-4. (2026 26th International Conference on Transparent Optical Networks (ICTON) Prague (Cze) 12-16 July 2026) [10.1109/icton71309.2026.11705195].

Availability:
This version is available at: 11583/3016914 since: 2026-10-23T12:06:37Z

Publisher:
Institute of Electrical and Electronics Engineers Inc.

Published
DOI:10.1109/icton71309.2026.11705195

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright
IEEE postprint/Author's Accepted Manuscript

©2026 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collecting works, for resale or lists, or reuse of any copyrighted component of this work in other works.

(Article begins on next page)

29 November 2026

Tensor Neural Network Based Optical Fault Localization

Yuxuan Gao,[1,2] Cristina Rottondi,[2] Andrea Sgambelluri,[3] Luca Valcarenghi,[3] Emilio Paolini,[3] Bingli Guo,[1] Shanguo Huang,[1] Andrea Bianco[2]

[1]Beijing University of Posts and Telecommunications, Beijing, China

[2]Department of Electronics and Telecommunications, Politecnico di Torino, Turin, Italy

[3]Scuola Superiore Sant'Anna, Pisa, Italy

ABSTRACT

We propose a fault localization method based on a tensor neural network to address the challenges of: *i)* extracting high order interaction features from multidimensional optical monitoring data for fault localization and *ii)* poor interpretability of neural networks-based models. The method significantly reduces the number of hyperparameters and computational complexity compared to long short-term memory recurrent neural networks and multilayer perceptrons, achieves higher fault localization accuracy, and generates interactive heatmaps that explain the results, providing an efficient and interpretable solution for intelligent operation and maintenance.

Keywords: Optical Networks, Fault Localization, Tensor Neural Networks

1. INTRODUCTION

Optical networks (ONs) serve as the backbone of modern telecommunications, offering unparalleled reliability, security, and high data throughput. However, ONs are susceptible to various types of failures that can severely degrade service quality, making Fault Localization (FL) essential for their operation and maintenance [1].

FL relies on monitoring and analyzing parameter variations across multiple network devices. As ONs continue to scale in size and complexity, the volume of monitoring data has become increasingly massive and high dimensional. In response, a growing body of research has introduced Artificial Intelligence (AI) techniques [2-3] to tackle fault classification and localization tasks. Nevertheless, conventional Neural Networks (NNs)-based approaches face two fundamental limitations when processing high dimensional monitoring data. First, high dimensional inputs lead to parameter proliferation, which not only increases computational and memory requirements but also exacerbates the risk of gradient explosion during training, hindering model convergence. Second, NNs operate as black boxes, providing only fault classification results without direct reasoning evidence, making it difficult to verify prediction correctness and formulate informed maintenance decisions.

To address these challenges, this paper introduces a Tensor Neural Network (TNN) [4] to achieve lightweight and accurate FL, generating parameter importance heatmaps for each device. This facilitates physical analysis of parameter interactions and their correspondence to fault locations, thereby assisting operators in understanding decision rationales. To validate the effectiveness of the proposed approach, we simulate attenuation faults at various locations on an optical testbed and collect monitoring data from multiple devices as model inputs. Experimental results demonstrate that the TNN achieves a FL accuracy of 0.991 while significantly reducing the number of parameters and computational complexity compared to a Long Short-Term Memory recurrent NN (LSTM - typically used for processing time series data), and Multi-Layer Perceptrons (MLP - commonly adopted in multi-classification scenarios).

2. METHODOLOGY

The core of the proposed FL scheme (Fig. 1) is the TNN. Unlike conventional NNs that flatten high-dimensional data into vectors, TNNs directly process multidimensional inputs, preserving high-order dependencies and enabling more efficient extraction of interaction features.

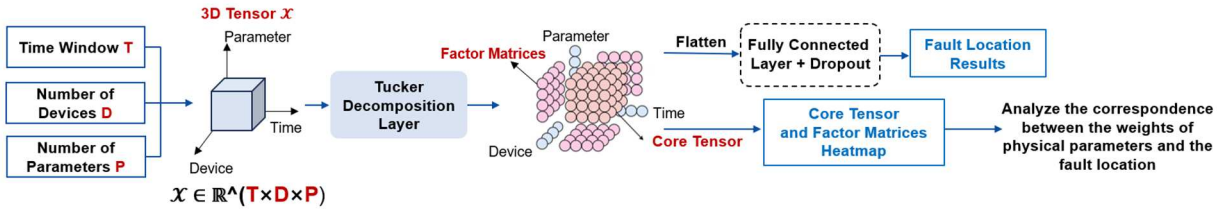


Figure 1. Flowchart of the FL scheme based on TNNs

Tucker decomposition [5] is employed to reduce latent space parameters while maintaining essential information. Additionally, TNN outputs are interpretable through heatmap visualization of the core tensor and factor matrices. The algorithm steps are as follows:

1) Data Preprocessing: The raw data inputs are transformed into three-dimensional tensors of size (T, D, P) using a sliding time window approach with window size T and stride s, where D denotes the number of monitored

devices and P denotes the number of monitored parameters per device. Parameter values that are not available are set to zero.

2) Tucker Decomposition: The input three-dimensional tensor $\mathcal{X} \in \mathbb{R}^{T\times D\times P}$ is decomposed using Tucker decomposition as $\boldsymbol{\mathcal{X}} \approx \boldsymbol{\mathcal{G}} \times_1 \boldsymbol{U^{(t)}} \times_2 \boldsymbol{U^{(d)}} \times_3 \boldsymbol{U^{(p)}}$. where, $\boldsymbol{\mathcal{G}} \in \mathbb{R}^{\boldsymbol{R_t}\times\boldsymbol{R_d}\times\boldsymbol{R_p}}$ is the *core tensor*, which contains the interaction information among the latent components. $\boldsymbol{U^{(t)}} \in \mathbb{R}^{\boldsymbol{T}\times\boldsymbol{R_t}}$、$\boldsymbol{U^{(d)}} \in \mathbb{R}^{\boldsymbol{D}\times\boldsymbol{R_d}}$、$\boldsymbol{U^{(p)}} \in \mathbb{R}^{\boldsymbol{P}\times\boldsymbol{R_p}}$ are the *factor matrices* for the time, device, and parameter dimensions respectively. Each column of these factor matrices corresponds to a latent component, and each row represents the weight of the original dimension on that component. The symbol $\times_n$ denotes the n-mode product, i.e., the multiplication of a tensor and a matrix along a specified mode. Through Tucker decomposition, the three-dimensional tensor (T, D, P) is decomposed into a core tensor of rank $(\boldsymbol{R_t}, \boldsymbol{R_d}, \boldsymbol{R_p})$ and three factor matrices $U1$, $U2$, and $U3$ of size $\boldsymbol{T}\times\boldsymbol{R_t}$, $\boldsymbol{D}\times\boldsymbol{R_d}$, and $\boldsymbol{P}\times\boldsymbol{R_p}$ respectively. The elements of the three factor matrices and the core tensor constitute the trainable parameters of the model, which are jointly updated via backpropagation along with the classification loss (cross entropy). More specifically, each element $\mathcal{G}_{i,j,k}$ of the *Core Tensor* represents the interaction strength among time i, device j, and parameter k. Each column of the *Time Factor Matrix* $U1$ represents a specific temporal pattern, such as an upward trend, a downward trend, or a periodic fluctuation. The value at the t-th row and i-th column of this matrix denotes the contribution weight of time step t to the i-th temporal pattern. By observing which time steps have larger weights, the critical moments when a fault occurs can be identified. Similarly, each column of the *Device Factor Matrix* $U2$ represents a specific device combination pattern. The magnitude of the weights reflects the degree of participation of each device in that pattern, thereby helping to identify the main devices affected by the fault. Finally, each column of the *Parameter Factor Matrix* $U3$ represents a specific parameter variation pattern, such as a decrease in power accompanied by an increase in Bit Error Rate (BER). By analyzing the weights, it is possible to reveal which physical quantities are most sensitive to the fault.

3) Output Classification: The core tensor is flattened and fed into a fully connected layer for nonlinear transformation. The ReLU activation function is used to enhance the representational capacity, and a Dropout layer is applied to prevent overfitting. Finally, the features are mapped to the class space, a probability distribution is obtained via the Softmax function, and the class corresponding to the highest probability identifies the fault location.

4) Interpretability/Results Analysis: The three factor matrices and the core tensor are extracted from the Tucker decomposition and visualized as heatmaps.

3. EXPERIMENTAL RESULTS

3.1 Testbed and Emulated Faults

The testbed setup, shown in Fig. 2, consists of three Reconfigurable Optical Add-Drop Multiplexers (ROADMs) interconnected by two 80-km-long links (L1 and L2). Two commercial DP-QPSK 100 Gb/s transponders are used to generate two Wave-Division-Multiplexed (WDM) signals (Ch1 and Ch2). At the end nodes, signals are collected by two coherent receivers. A Wavelength Selective Switch (WSS) with variable attenuation on L1 and a Variable Optical Attenuator (VOA) on L2 are used to introduce 5 dB attenuation to simulate faults on different links and channels: attenuation on Ch1 only, attenuation on Ch2 of L1 only, attenuation on Ch2 of L2 only, attenuation on Ch2 of both L1 and L2, attenuation on both Ch1 and Ch2 of L1 with L2 not attenuated, and attenuation on both Ch1 and Ch2 of L1 with L2 attenuated. Performance data (samples) are collected for the normal state and the above-described faulty states, measuring on D=5 devices, P=4 parameters on D=5 devices, namely *i*) the input and *ii*) output optical power of three amplifiers (A1, A3, and A5), the *iii*) Optical Signal to Noise Ratio (OSNR) and *iv*) BER from the receivers of Ch1 and Ch2. Samples are collected every 4 seconds, and each normal/faulty state is monitored for one hour, resulting in 900 samples. Afterwards, 19 sequences of samples are created, where the proportion of normal and faulty samples within each sequence is varied to simulate different transition (or no transition) scenarios from normal state to faulty state. Each resulting sequence is labelled according to the fault location (labels 0–6), if any. The obtained sequences are then subdivided into training, validation, and test sets according to a 60%, 20%, and 20% split.

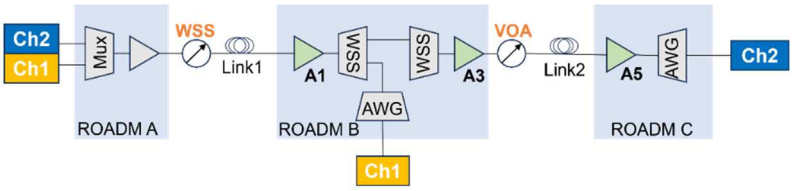


Figure 2. Testbed for data collection

3.2 Model Comparison

We evaluate the performance of TNN, LSTM, and MLP models on the above-described dataset. To emulate transitions from normal to faulty conditions, three scenarios are simulated by varying the concatenation ratio of normal data to faulty data: (1: 1), (4: 1) and (9:1). Each sequence of samples is partitioned using a window size $T = 10$ and stride $s = 2$ (i.e., 446 partially-overlapping windows per sequence are generated). Consequently, the three-dimensional tensor (T, D, P) input to the TNN has shape (10,5,4).

Focusing on the (1:1) scenario, the Tucker rank and the number of neurons per layer for the three models are optimized via grid search. The number of layers (2), learning rate (0.001), dropout rate (0.5), and batch size (32) were instead chosen based on empirical values [6-8]. Based on the results reported in Tab.1, the Tucker rank of the TNN is set to (4, 3, 2) and the two fully connected layers have 128 and 64 neurons respectively. The LSTM model uses 64 hidden units and a fully connected layer with 32 neurons. The MLP model consists of two hidden layers with 128 and 64 neurons respectively.

The three models are trained, validated, and tested using the datasets from the three scenarios. Results shown in Fig. 3 highlight that in the (1:1) scenario, all models achieve accuracy above 0.96. TNN achieves the highest accuracy at 0.991. As the proportion of faulty data decreases, the performance of all models declines significantly. In the extremely imbalanced scenario where fault data accounts for only 10% of the observed time window, the accuracy of MLP drops sharply to 0.429, while TNN reaches 0.824 accuracy.

3.3 Parameter Count

Since the TNN directly performs low rank decomposition and interaction modelling on high dimensional data, it does not require the large amount of parameters introduced by the fully connected layers of MLP and the gating structures of LSTM. Consequently, as shown in Tab.2, the comparison of parameter count and computational complexity among TNN, LSTM, and MLP reveals that the parameter count of TNN is about 35% of MLP's and 21% of LSTM's. Furthermore, the computational complexity of TNN, computed in terms of Multiply-ACcumulate operations (MACs), is merely 3% of LSTM's.

Table 1. Comparison of Macro F1 scores for TNN, LSTM and MLP with different architectures

Model	Hidden Units	FC Layer Neurons	Macro F1
LSTM	32	16	0.955
		32	0.958
		64	0.956
	64	**32**	**0.978**
		16	0.973
		64	0.975
	128	16	0.962
		32	0.966
		64	0.964

Model	Hidden Layers Neurons (1st, 2nd)	Macro F1
MLP	(64,32)	0.951
	(128,64)	**0.967**
	(256,128)	0.959

Model	Tucker rank	FC Layer Neurons (1st, 2nd)	Macro F1
TNN	(3,2,2)	(64,32)	0.972
		(128,64)	0.976
		(256,128)	0.974
	(4,3,2)	(64,32)	0.984
		(128,64)	**0.989**
		(256,128)	0.986
	(5,4,3)	(64,32)	0.980
		(128,64)	0.983
		(256,128)	0.981
	(4,2,2)	(64,32)	0.965
		(128,64)	0.969
		(256,128)	0.967

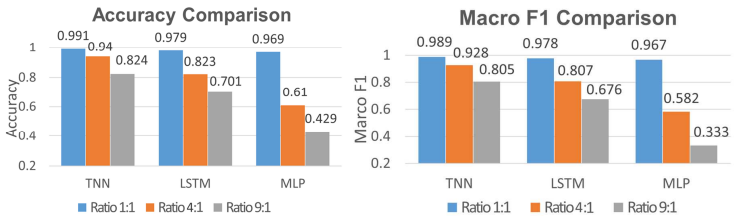


Figure 3. Accuracy and Macro F1 of TNN, LSTM, and MLP under scenarios with different proportions of normal data to fault data

Table 2. Comparison of parameter count and computational complexity for TNN, LSTM, And MLP

Model	Parameters Count	MACs (per sample)
TNN	12.0k	16.5k
LSTM	57.1k	545k
MLP	34.4k	34.2k

3.4 Explainability Analysis

Fig. 4 shows the results for a sample of the (1:1) scenario when the fault localization result is identified as attenuation occurring on Ch1 of L1: the core tensor extracted by the TNN is presented, along with the factor matrices for the time, device, and parameter dimensions. If the output fault localization result indicates a different fault, the corresponding core tensor and factor matrix will also be output. The left side of Fig.4. shows the three-dimensional core tensor. For ease of observation, it is sliced along the time axis into four two-dimensional heatmaps, reported on the right, revealing that the highest value of -0.554 corresponds to time component 3, device component 2, and parameter component 0. Fig.5. shows the Time Factor Matrix, Device Factor Matrix, and Parameter Factor Matrix respectively. From the column corresponding to component 0 in the Time Factor Matrix, it can be observed that after T equals 5, the data values increase significantly, indicating that a fault occurs halfway

through the time period. From the column corresponding to component 2 in the Device Factor Matrix, higher weights are observed on A1 and Ch1. From the column corresponding to component 0 in the Parameter Factor Matrix, higher weights are observed on Input power and OSNR. From these observations, it can be deduced that in the fault scenario with Ch1 attenuation only on L1, the most prominent features are the input optical power of A1 and the OSNR of Ch1, and the fault occurs halfway through the time period.

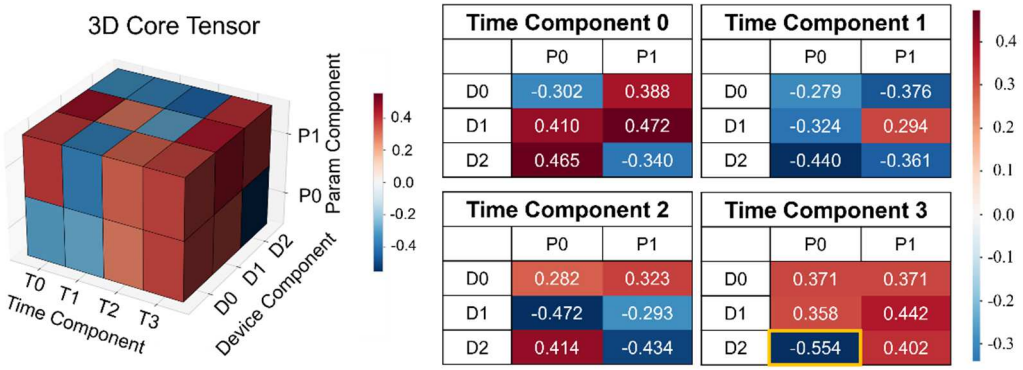
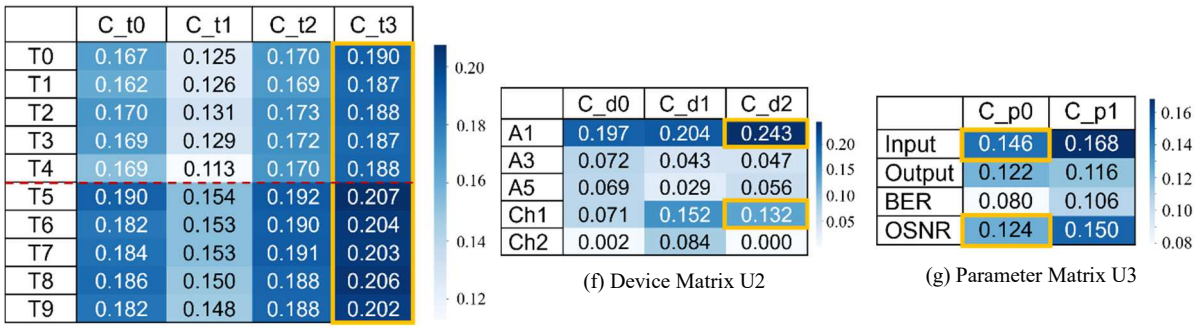


Figure 4. Heatmap of the 3D core tensor and its 2D slices along the time axis



(e) Time Matrix U1

(f) Device Matrix U2

(g) Parameter Matrix U3

Figure 5. Heatmaps of the time, device, and parameter factor matrices

4. CONCLUSIONS

This paper introduces the usage of TNNs for fault localization in optical networks. Through experimental validation, we demonstrate that the proposed scheme achieves higher accuracy and F1 score compared to LSTM and MLP under both balanced and imbalanced scenarios of normal and fault data distribution, with the highest accuracy reaching 0.991. While attaining high performance, the parameter count and computational complexity are both significantly lower than those of the baseline models. Moreover, the scheme outputs interactive heatmaps that explain the FL results alongside the results themselves, greatly enhancing the interpretability of the approach.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the financial support from the China Scholarship Council (CSC) (202506470012), the Ministry of Science and Technology of the People's Republic of China (2022YFB2903300), and the National Natural Science Foundation of China (62125103, 62171059, U24B20134).

REFERENCES

[1] Danshi Wang, Chunyu Zhang, Wenbin Chen, Hui Yang, Min Zhang, and Alan Pak Tao Lau, "A review of machine learning-based failure management in optical networks", in *Science China Information Sciences*, 65(11), 2022

[2] T. Liu, H. Mei, Q. Sun and H. Zhou, "Application of neural network in fault location of optical transport network," in *China Communications*, vol. 16, no. 10, pp. 214-225, Oct. 2019

[3] X. Zhao, H. Yang, H. Guo, T. Peng and J. Zhang, "Accurate Fault Location based on Deep Neural Evolution Network in Optical Networks for 5G and Beyond," *2019 Optical Fiber Communications Conference and Exhibition (OFC)*, San Diego, CA, USA, 2019

[4] Wang M, Pan Y, Xu Z, et al. Tensor networks meet neural networks: A survey and future perspectives. arXiv:2302.09019, 2023

[5] X. Liu and K. K. Parhi, "Tensor Decomposition for Model Reduction in Neural Networks: A Review [Feature]," in *Circuits and Systems Magazine*, vol. 23,8-28, Second quarter 2023

[6] N. Srivastava, G. Hinton, A. Krizhevsky, et al., "Dropout: a simple way to prevent neural networks from overfitting," in *Journal of Machine Learning Research* 15, 1929–1958 (2014)

[7] D. P. Kingma, J. L. Ba, "Adam: a method for stochastic optimization," in *Int. Conference on Learning Representations (ICLR)* (2015)

[8] R. Lin, "Analysis on the Selection of the Appropriate Batch Size in CNN Neural Network," in *2022 International Conference on Mhine Learning and Knowledge Engineering (MLKE)*, Guilin, China, 2022, pp. 106-109