

Variants of MANTIS

September 25, 2025

1 Introduction

Here we briefly outline our method for anomaly and fraud detection using tensor networks for Project MANTIS: Multiple Anomaly-detection Networks for Tensor-Inspired Solutions.

2 Model description

Let the dataset $\{(\mathbf{x}_n, y_n)\}_{n=1}^N$, where $\mathbf{x}_n \in \mathbb{R}^L$ is an n th input having L features and $y_n \in \{0, 1\}$ is a label of the n th data, be the dataset for the binary classification. At the beginning of the training, the dataset will be normalized by the *rank normalization* to prevent any outliers from the raw data. In the fundamental knowledge of deep learning, any model could impose nonlinearity to improve learning efficiency. Thus, we have to encode any input \mathbf{x}_n into an input MPS $|\Psi_n\rangle$ by defining a feature map $\Psi : [0, 1]^L \mapsto (\mathbb{R}^{2P})^{\otimes L}$, i.e.,

$$|\Psi_n^{(p)}\rangle = \bigotimes_{l=1}^L \begin{pmatrix} \cos\left(\frac{\pi}{2^p}x_{nl}\right) \\ \sin\left(\frac{\pi}{2^p}x_{nl}\right) \end{pmatrix}. \quad (1)$$

Next, we learn our MPO($\boldsymbol{\theta}$) = $\text{Tr}\left(C \prod_{l=1}^L A^{[l]}\right)$ where an m, p element of a tensor $A^{[l]} \in \mathbb{R}^{M \times P \times 2 \times 2}$ defined as a rotation matrix

$$A_{m,p}^{[l]} = \begin{pmatrix} \cos\theta_l^{m,p} & -\sin\theta_l^{m,p} \\ \sin\theta_l^{m,p} & \cos\theta_l^{m,p} \end{pmatrix} \quad (2)$$

and $C = (c_{m,p}) \in \mathbb{R}^{M \times P}$ is a coefficient tensor. An output for the n th input will end up as

$$|\Phi_n\rangle = \sum_{m=1}^M \sum_{p=1}^P c_{mp} \bigotimes_{l=1}^L \begin{pmatrix} \cos\left(\theta_l^{m,p} + \frac{\pi}{2^p}x_{nl}\right) \\ \sin\left(\theta_l^{m,p} + \frac{\pi}{2^p}x_{nl}\right) \end{pmatrix}, \quad (3)$$

noting that the n th output MPS will be normalized by a normalization constant

$$\langle \Phi_n | \Phi_n \rangle = \sum_{m,m'=1}^M \sum_{p,p'=1}^P c_{m,p} c_{m',p'} \prod_{l=1}^L \cos\left(\theta_l^{m,p} - \theta_l^{m',p'} + \left(\frac{\pi}{2^p} - \frac{\pi}{2^{p'}}\right)x_{nl}\right). \quad (4)$$

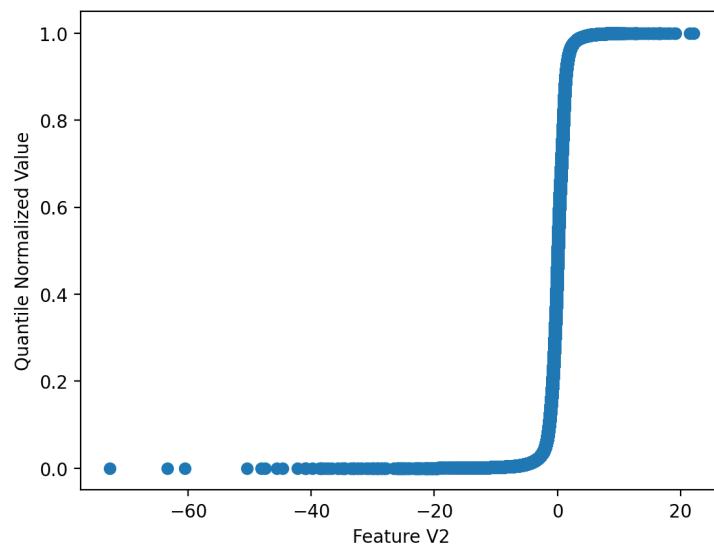


Figure 1: Mapping of Feature V2 onto its quantile normalized values.

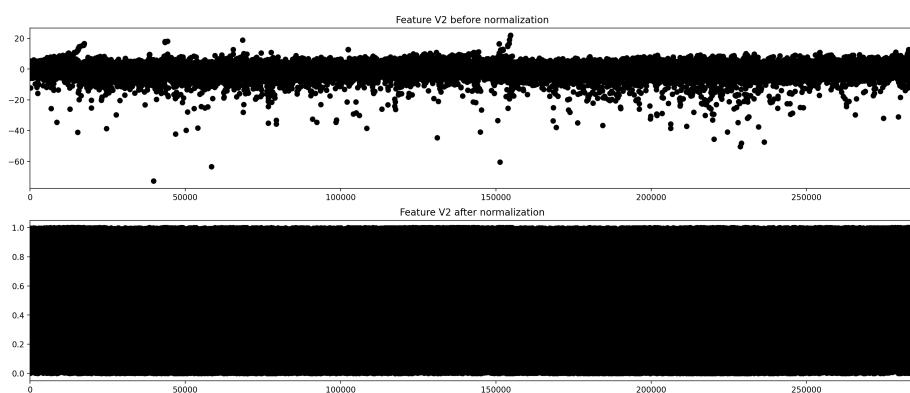


Figure 2: Feature V2, before and after normalization across the 284, 807 transactions.

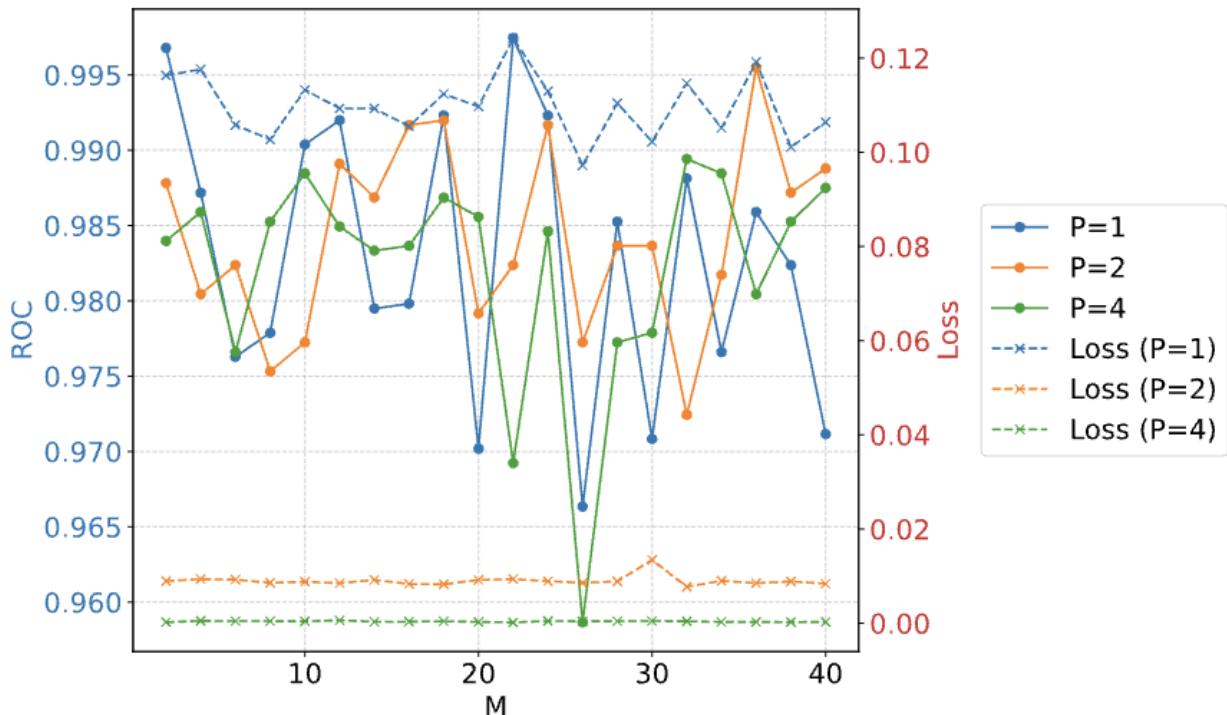
(09.17.25) Regularization

The loss function

$$\mathcal{L} = \frac{1}{N} \sum_{n=1}^N (1 - m_n^z)^2 + \lambda_c \text{Var}(\vec{c}) + \lambda_m \mathbb{E}_{p,l} \text{Var}_m(\{\theta_l^{m,p}\}) + \lambda_p \mathbb{E}_{m,l} \text{Var}_p(\{\theta_l^{m,p}\})$$

We use Bayesian optimization to find the optimal $(\lambda_c, \lambda_m, \lambda_p)$.

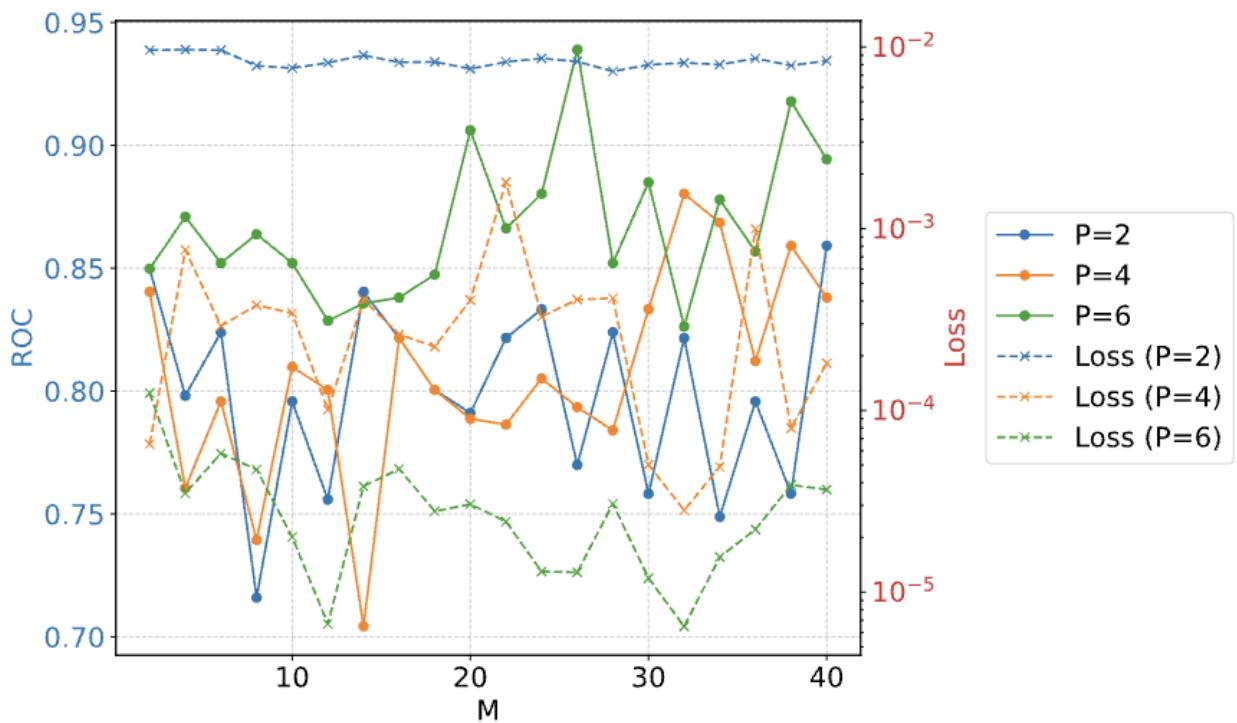
Wine dataset



Best Performance (by ROC)

```
P = 1
M = 22
reg = 1e-06
lr = 0.01
num_batch = 32.0
loss = 0.1242166683077812
roc = 0.9974358974358974
acc = 0.9380530714988708
```

Lympho



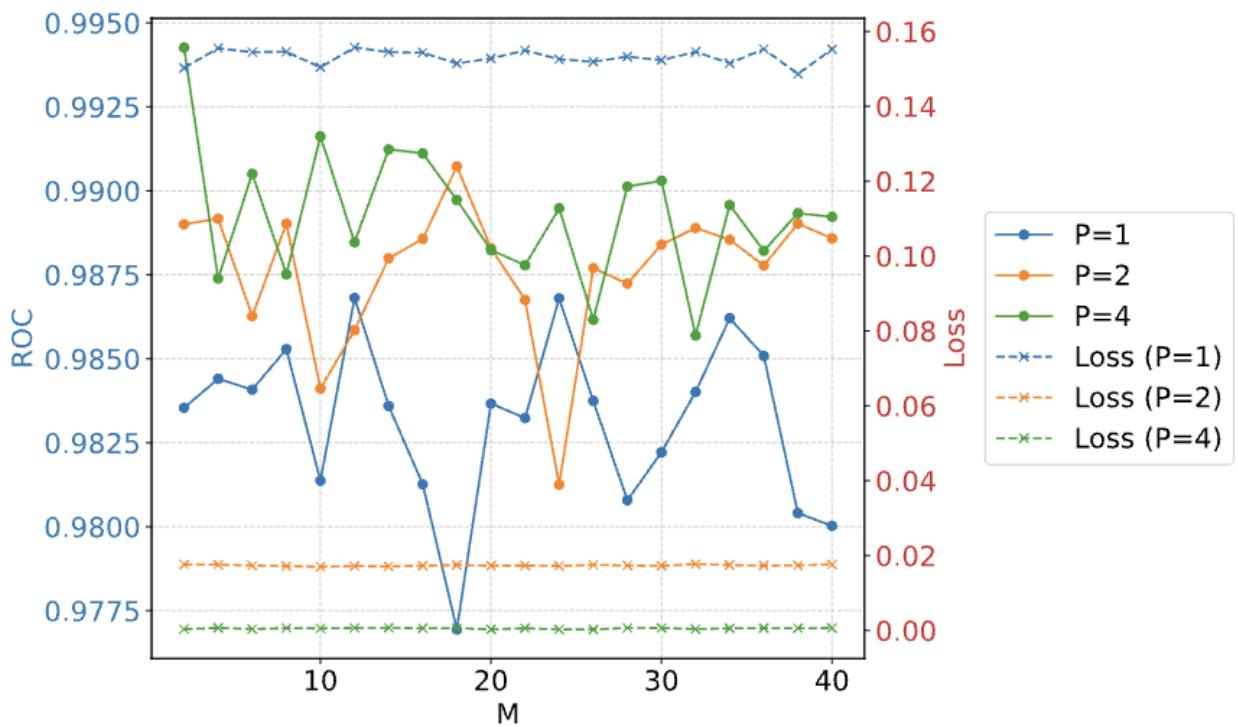
Best Performance (by ROC)

```

P = 6
M = 26
reg = {'c': 0.002516285793393072, 'theta_m': 0.044831935530461, 'theta_p': 0.012633289779453124}
lr = 0.01
num batch = 32
loss = 1.2863081792602316e-05
roc = 0.9389671361502347

```

Thyroid



Best Performance (by ROC)

```

P = 4
M = 2
reg = 1e-06
lr = 0.003
num_batch = 32.0
loss = 0.0003138229658361
roc = 0.9942671809256662
acc = 0.9518882632255554

```