

LEAN-DMKDE: Quantum Latent Density Estimation for Anomaly Detection (Student Abstract)

Joseph A. Gallego-Mejia, Oscar A. Bustos-Brinez, Fabio A. González

Universidad Nacional de Colombia
Bogotá, Cundinamarca 111311, Colombia
{jagallegom, oabustosb, fagonzalezo}@unal.edu.co

Abstract

This paper presents an anomaly detection model that combines the strong statistical foundation of density-estimation-based anomaly detection methods with the representation-learning ability of deep-learning models. The method combines an autoencoder, that learns a low-dimensional representation of the data, with a density-estimation model based on density matrices in an end-to-end architecture that can be trained using gradient-based optimization techniques. A systematic experimental evaluation was performed on different benchmark datasets. The experimental results show that the method is able to outperform other state-of-the-art methods.

Introduction

Anomaly detection is a critical task in many machine learning frameworks, and its main objective is to detect whether a data point is anomalous or not (that is, if it deviates from some notion of “normality”) and leverage it to make actionable decisions (Ruff et al. 2021). Anomalies can also be known as outliers, novelties, exceptions or peculiarities, among other terms. In this paper, we propose a new method, named LEAN-DMKDE, that uses the deep representation obtained by an autoencoder and the representation capabilities of density matrices, a formalism originally developed for quantum mechanics. This new method can be trained end-to-end, thus solving the shortcomings of other methods based on kernel density estimation.

The contributions of the present work include the description of the method architecture, including how it combines autoencoders with kernel methods in order to calculate a likelihood estimation of density. Also, we present a systematic comparison of LEAN-DMKDE over multiple datasets against some state-of-the-art anomaly detection methods (Gallego-Mejia, Bustos-Brinez, and González 2022).

Quantum Latent Density Estimation for Anomaly Detection (LEAN-DMKDE)

The proposed method, named Quantum Latent Density Estimation for Anomaly Detection (LEAN-DMKDE), is shown in Figure 1. This model is composed of an autoencoder, an adaptive Fourier feature layer, a quantum measurement layer

and a threshold anomaly detector. LEAN-DMKDE can be seen as a coupling of the anomaly detection framework AD-DMKDE, presented in (Bustos-Brinez, Gallego-Mejia, and González 2022), with the deep latent representation given by the usage of an autoencoder in the initial stage.

The autoencoder is responsible for performing a dimensionality reduction and calculating the reconstruction error. The adaptive Fourier feature layer maps the autoencoder latent space to a Hilbert space whose dot product approximates a Gaussian kernel. The quantum measurement layer uses this kernel approximation to produce density estimates of the given data points, as developed in (González et al. 2021). The last step of the algorithm is an anomaly detector that takes the density estimates and classifies each sample as anomalous or normal by using a threshold value.

Autoencoder. Given an input space \mathbb{R}^d , and a latent space \mathbb{R}^p , where typically $p \ll d$, a point in the input space is sent through an ψ -encoder and also a θ -decoder step, where $\psi : \mathbb{R}^d \rightarrow \mathbb{R}^p$ and $\theta : \mathbb{R}^p \rightarrow \mathbb{R}^d$, such that:

$$\mathbf{z}_i = \psi(\mathbf{x}_i, \mathbf{w}_\psi) \quad \hat{\mathbf{x}}_i = \theta(\mathbf{z}_i, \mathbf{w}_\theta)$$

where \mathbf{w}_ψ and \mathbf{w}_θ are the network parameters of the encoder and the decoder respectively, and $\hat{\mathbf{x}}$ is the reconstruction of the original data.

Quantum Measurement for Kernel Density Estimation. The latent space obtained in the autoencoder space is sent to an adaptive Fourier feature layer and its outputs sent to the density matrix. A density matrix is defined as:

$$\rho = \frac{1}{N} \sum_{i=1}^N \phi_{\text{aff}}(\mathbf{x}_i) \phi_{\text{aff}}^T(\mathbf{x}_i)$$

where ϕ_{aff} represents an encoding scheme called Adaptive Fourier features (AFF), first introduced in (Gallego-Mejia and González 2022). The AFF mapping is given by the function $\phi_{\text{aff}}(\mathbf{x}_i) = \sqrt{2} \cos(\mathbf{w}^T \mathbf{x}_i + \mathbf{b})$, where \mathbf{w} and \mathbf{b} are vector parameters, which are built using an optimization process. This encoding creates an explicit map that approximates a Gaussian kernel.

Nonetheless, the memory requirements to utilize the density matrix ρ can be reduced by using the matrix eigen-decomposition $\rho \approx V^T \Lambda V$, where $\Lambda \in \mathbb{R}^{r \times r}$ is a diagonal matrix whose values correspond to the r largest eigenvalues

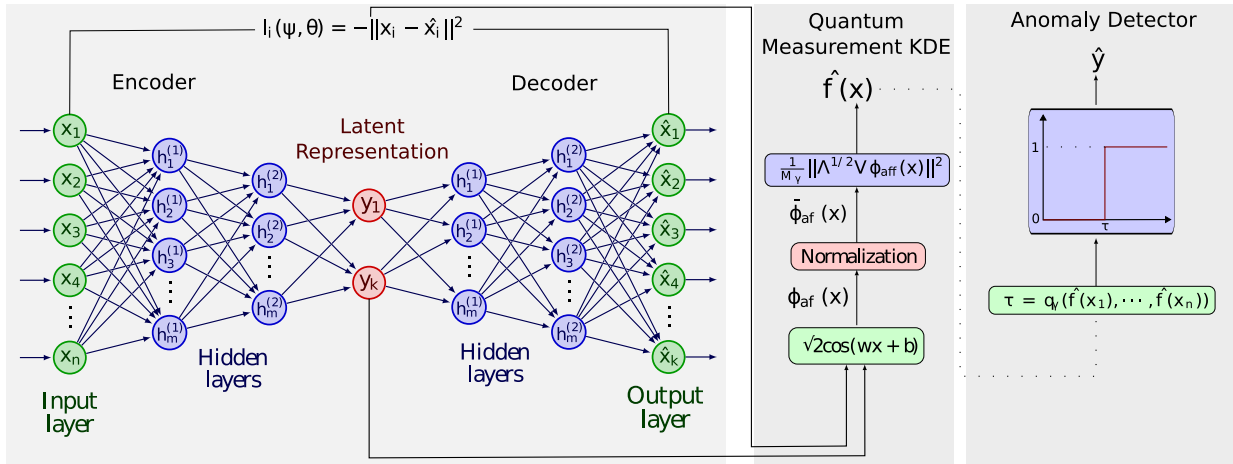


Figure 1: Quantum Anomaly Detection through Density Matrices, Adaptive Fourier Features and Autoencoders (LEAN-DMKDE) method. Step 1, autoencoder representation learning. Step 2, training of the model using both the reconstruction error and the maximum likelihood estimation of the density matrix ρ . Step 3, estimation of the density of a new sample using the density matrix $\rho = V^T \Lambda V$. Step 4, anomaly detector using the proportion of the anomalies.

of ρ , and $V \in \mathbb{R}^{r \times D}$ is a matrix whose rows contain the r eigenvectors associated with these eigenvalues.

In order to calculate the density of a new test data point \mathbf{x} , the density matrix is used as:

$$\hat{f}(\mathbf{x}) = \frac{1}{M_\gamma} \phi_{\text{aff}}^T(\mathbf{x}) \rho \phi_{\text{aff}}(\mathbf{x})$$

where M_γ is a normalization constant.

Anomaly Detection. The last step of the proposed algorithm is to detect whether a given data point is normal or anomalous. The algorithm uses the proportion of anomalies within the data set γ (it can be a known value or an expected ratio) to obtain a threshold value τ given by

$$\tau = q_\gamma(\hat{f}(\mathbf{x}_1), \dots, \hat{f}(\mathbf{x}_n))$$

where q is the percentile function. Finally, the algorithm uses the τ value to classify anomalous points as:

$$\hat{y}(\mathbf{x}_i) = \begin{cases} \text{'normal'} & \text{if } \hat{f}(\mathbf{x}_i) \geq \tau \\ \text{'anomaly'} & \text{otherwise} \end{cases}$$

Results

The proposed method LEAN-DMKDE was tested against eleven different anomaly detection algorithms, including classical, shallow and deep learning methods. The implementation of these baseline methods came from the well-known Scikit-Learn and PyOD Python libraries. The experimental setup for all these frameworks included eighteen anomaly detection datasets, taken from ODDS Library (Rayana 2016), a parameter grid-search for each method, and the selection of F1-Score as the main metric to compare their performances. As shown in Fig. 2, LEAN-DMKDE clearly outperforms the majority of the other considered methods, showing the best average performance and having a consistently high scoring over the datasets.

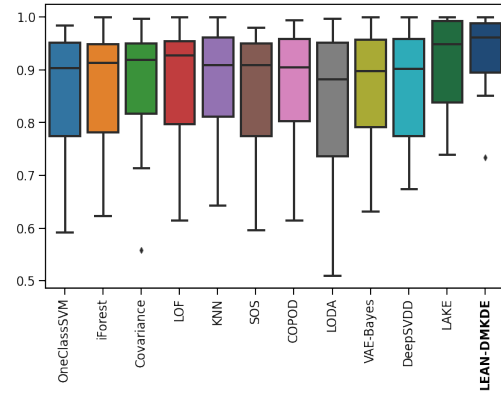


Figure 2: Boxplot of F1 scores obtained for all algorithms (including LEAN-DMKDE) over all datasets.

References

- Bustos-Brinez, O.; Gallego-Mejia, J.; and González, F. A. 2022. AD-DMKDE: Anomaly Detection through Density Matrices and Fourier Features. *arXiv preprint arXiv: 2210.14796*.
- Gallego-Mejia, J.; Bustos-Brinez, O.; and González, F. A. 2022. LEAN-DMKDE: Quantum Latent Density Estimation for Anomaly Detection. *arXiv:2211.08525*.
- Gallego-Mejia, J. A.; and González, F. A. 2022. Quantum Adaptive Fourier Features for Neural Density Estimation. *arXiv e-prints, arXiv-2208*.
- González, F. A.; Gallego, A.; Toledo-Cortés, S.; and Vargas-Calderón, V. 2021. Learning with density matrices and random features. *arXiv preprint arXiv: 2102.04394*.
- Rayana, S. 2016. ODDS Library. <http://odds.cs.stonybrook.edu>. First accessed: 2022-08-08.
- Ruff, L.; Kauffmann, J. R.; Vandermeulen, R. A.; Montavon, G.; Samek, W.; Kloft, M.; Dietterich, T. G.; and Müller, K.-R. 2021. A Unifying Review of Deep and Shallow Anomaly Detection. *Proceedings of the IEEE*, 109: 756–795.