Anomaly detection in time series for Tunneling Boring Machines

A literary review of useful method

Mathieu Giamberini, 2024

Introduction

By default, Tunnel Boring Machines (TBMs) are complicated technical objects and involved a lot of technologies working together. Any defect in the chain could stop the entire machine. Therefore, detecting those before they become a real issue can save a lot of time and resources. Fortunately, the number of sensors on the machines keeps raising and thanks to platforms like Herrenknecht.Connected [9], their data can be accessed anywhere by any authorized user. These data must carry a lot of information regarding the state of the machine, which begs the question of how to extract it to detect anomalies. At the time of writing, the task is mostly done by experienced users of the TBM, e.g. a weird noise in the hydraulic pack, an odd torque increase on one of the main drive motors, and so on. This method is clearly limited and uses only a small portion of the available data stream. Despite this, the problem of anomaly detection is quite old, and with the rise of machine learning technic, it keeps getting better.

Therefore, in this paper, we will, have a first look of at the literary landscape of Anomaly detection (AD) methods. First, let's define exactly what we mean by that, given a multivariate time series:

$$\mathcal{X} = (\mathbf{x}_1, \dots \mathbf{x}_t, \dots \mathbf{x}_T) \in \mathbb{R}^{N \times T} \tag{1}$$

with N the number of sensors and T the number of temporal indices. The method $\mathcal{M}$ will return a binary category for each data point telling where and when there is anomalies, ie:

$$\mathcal{M} : \mathbb{R}^{N \times T} \to \{0, 1\}^{N \times T} \ . \tag{2}$$

We need to address some loose ends with this definition. First, in most cases, there is no already labeled data to train the method on, and labeling it by hand is not feasible. So the method must be unsupervised, which raises the need for a systemic definition of anomalies. Most papers consider the training data to be normal and define some anomaly score, which tells how far away the new data is. Here, the methods will, be categorized based on the way this score is defined.

In our context, the training data is stems from historic tunnel projects, which contains anomalies. Therefore, we will also discuss some papers which tried to address this issue by modifying the methods.

1 Overview

AD methods can be categorized in many ways [6]. In this paper is divided in two parts (cf Figure 1), general methods and useful tricks to improve theses. The former represents general ways to solve the AD problem. It is composed of three categorizes: forecasting (Sec 1.1), clustering (Sec 1.2) and index monitoring (Sec. 1.3). The latter are so-called add-ons (Sec 1.4). The papers in this category present an AD method with some tricks that, in our minds can be applied to any method to improve it.

1.1 Forecasting Methods

When working with some equipment, you know it so well that, if asked, you could almost predict the sound it will make in the near future. But if suddenly the sound doesn't match your expectations you know there are some things wrong. That is the core idea for the forecasting method, predict the near future using some generative method and if the error goes over some threshold, the data point is labeled as an anomaly. Example of those forecasting technics may be LSTM [10] or using TCN [1].

1.2 Clustering method

While storing metric screws in the warehouse, someone picks up an imperial one, he wants to find any box to put it in. Therefore he will conclude there was an error in the shipment, even if he didn't know this type of screw. This is the main idea for the clustering method, define clusters, if a new data point can't fit in one of them, it is labeled as an anomaly. There are multiple ways to define the cluster. A good part uses some defined distance [7] function and uses a variant of K-nearest-neighbor (KNN) [13]. Similarly, use the density of the data point in relation to the training data, which will give less, strict categorizes [3]. Another method is to estimate the probability density function (PDF) and put a threshold on low values [2].

1.3 Index monitoring

After assembly, the disks cutters there are put under pressure for some time. When the time is up, the pressure is measured, if the pressure is too low, there is a problem somewhere. This is the intuition behind index monitoring method, define some function with an acceptable range of normal values, if the values are outside the range it's marked as an anomaly. This technic is one of the oldest, simplest and most used in the manufacturing industries, in this context is referred as Control chart [15]. Another example of application is [11] which use a modified wavelet transform to define a Health Index to monitor bearings.

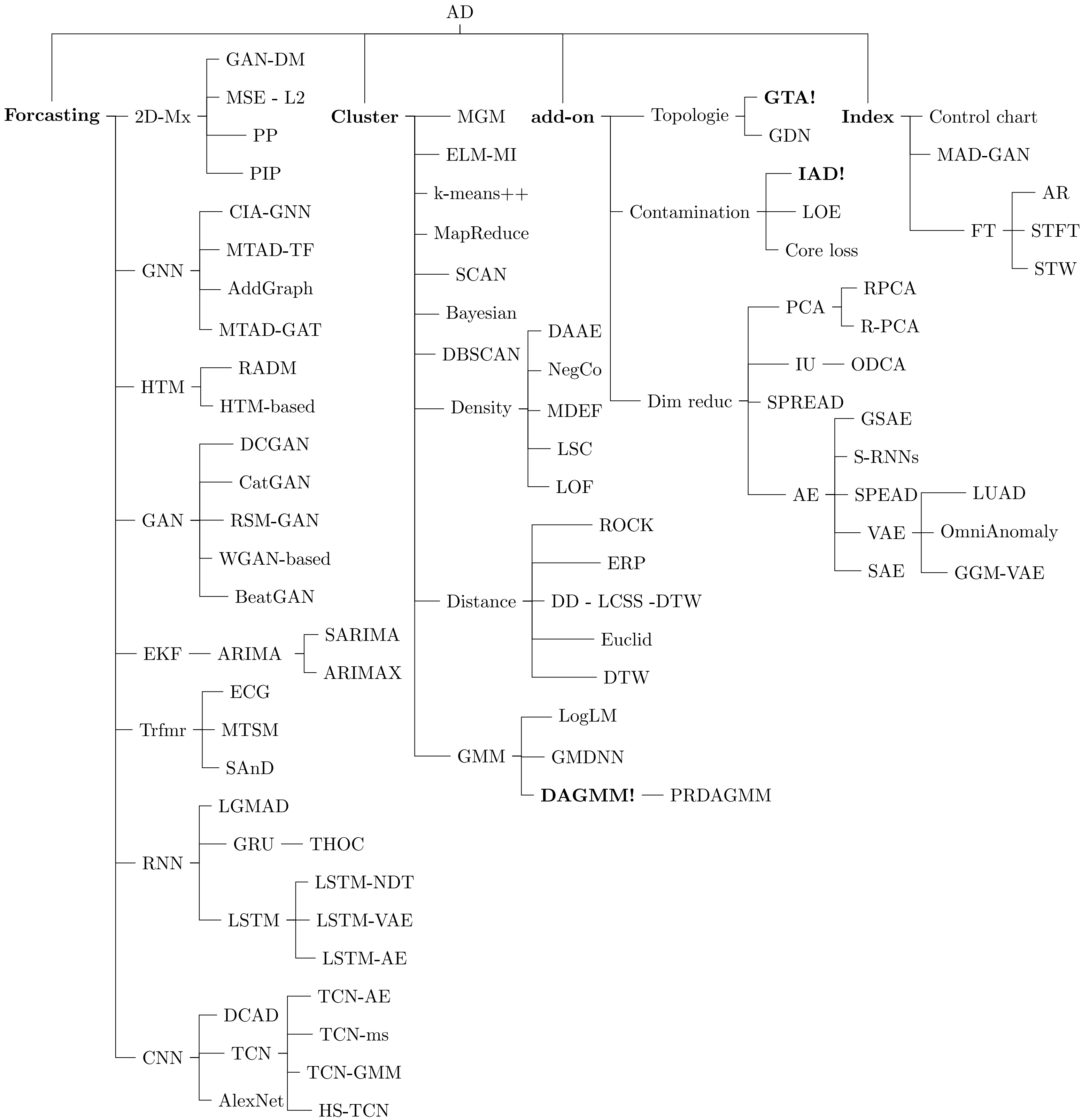

Figure 1: Anomaly detection method categorization

1.4 Add-ones

In order to improve those techniques and solve issues intrinsic to the TBM case, here are some interesting solutions.

Topology, TBM AD is intrinsically a multivariate problem (ie there is more than one sensor on the machine). And it is safe to assume that the anomaly information for some problems is carried by multiple sensors. For example, if there is a defective sensor, the others want to raise any anomaly, which will greatly help to pin the root cause down. This what [8] and [4] did so by introducing a directed graph to model the relationship between sensors. This information was then used, in these papers, to do forecasting using attention techniques.

Contamination, the training data for those models will surely be some already-bore tunnel data. During those bores, by Murphy's law, at some point, some things went wrong. The issue is that most methods assume an anomaly free dataset. This is why [12] [5] [16] proposed some modifications to the training steps to deal with this contamination. To do so, they generally modify the loss function to account for the uncertainty of the data and improve this uncertainty iteratively.

Dimension reduction, a large quantity of sensors, with a quite small data rate, even with a small time window can make the input dimension of the model certainly significant. This can render the training hard and sensitive to noise. to solve this issue, most papers use some kind of dimension reduction. For example, [2] uses an AE and a modified reconstruction error concatenated with the latent vector as an input to a GMM. Or [14] used a LSTM as an AE to do forecasting.

2 Description

2.1 Graph Deviation Network (GDN)

In a multivariate settings, it's sensible to assume that related sensors have redundant information (e.g. a pressure and a extension sensors for a hydraulic cylinder). Correlating those should therefore increase the efficiency of prediction. This is a base idea of the GDN architecture in [8]. Mark the related sensor in a directed Graph and apply an attention layer on the past measurement of the neighborhoods to predict the next ones. Here is some short assertion and nice propriety of this approach :

- Even if every sensor of the TBM are in the graph, the AD is still localize to each sensor.
- The training data is assume to be free of anomaly.
- Prior knowledge of relation between sensor can be embeded in the graph by restricted some relationship. This can lower the complexity and the time of training.

To test their method the authors used dataset of simulated attack on water treatment physical test-bed systems, the Secure Water Treatment (SWaT) and Water Distribution (WADI) dataset. As a base line they use other method which we discus in other Section.

	SWaT			WADI		
Method	Prec	Rec	F1	Prec	Rec	F1
PCA	24.92	21.63	0.23	39.53	5.63	0.10
KNN	7.83	7.83	0.08	7.76	7.75	0.08
FB	10.17	10.17	0.10	8.60	8.60	0.09
AE	72.63	52.63	0.61	34.35	34.35	0.34
DAGMM	27.46	69.52	0.39	54.44	26.99	0.36
LSTM-VAE	96.24	59.91	0.74	87.79	14.45	0.25
MAD-GAN	98.97	63.74	0.77	41.44	33.92	0.37
GDN	**99.35**	**68.12**	**0.81**	**97.50**	**40.19**	**0.57**

Table 1: Anomaly detection accuracy in terms of precision(%), recall(%), and F1-score, on two datasets with ground-truth labelled anomalies. Table from [8].

2.2 Latent Outlier Exposure (LOE)

If the contamination of the training data is if not dealt with, the method will learn the anomalies as normal data. In a supervised setting, the problem would not arise, the known labels would just be directly used to train the model. Nonetheless, in our context, the "normal" data should be in majority and so the most influential on the gradient at the start. In other words, the assumption is, after the first few iterations the method already has some anomaly detection capability. Using those would make it possible to act as if we were in a supervised settings.

Inspired by this, [5] proposed to learn the unknown labels of the training data. In each optimization step they guessed the most probable anomalies using the current method parameter θ and use this info to update θ for the next iteration.

More precisely they first define α the possible contamination rate, $\mathcal{L}_n^\theta(x)$ a loss function aimed to be minimized for "normal" data point x, likewise $\mathcal{L}_a^\theta(x)$ for anomalies and $S^\theta(x)$ being the anomaly score used for the detection. To then combine it in a global loss function :

$$\mathcal{L}^\theta = \sum_{i=1}^{N}(1-y_i)\mathcal{L}_n^\theta(x) + y_i\mathcal{L}_a^\theta(x) \tag{3}$$

Where y_i is the anomaly label, which for each optimization iteration, is assigned 1 for the $(1-\alpha)$-quartile of the rank anomalies score $S^\theta(x_i)$ and 0 to the remaining ones.
With the same core idea [12] and [16] dropped the $\mathcal{L}_a^\theta(x)$ and used more soft classes (See the original paper for more information).

For the Tunneling applications look, in the author's opinion, quite promising. It's very general, easy to implement, the only downside is this initial assumption, which is possible but not proven.

2.3 DAGMM! (DAGMM!)

- principle
  - AE -¿ MLP -¿ GMM
  - train all at once
- need
  - figure
- pro con
  -

2.4 Autoregression - Health index (AR)

-

Conclusion

In this paper we have seen that there is three main ways to do AD, forecasting, clustering, index monitoring. It can be observed that the dataset usually used to compare the forecasting methods (cf Table 1) may not be generalized to TBM. The reasoning is that those data set is homogenous in time, when the TBM's ones highly depend on a varying geology. We also found, some ways to improve those methods using topologies information, contamination handling and dimension reduction. The aim of future works will be to test those methods on actual TBM data to see how well they work, and under which condition.

References

[1] Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. *An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling*. URL: http://arxiv.org/pdf/1803.01271.

[2] Bo Zong et al. "Deep Autoencoding Gaussian Mixture Model for Unsupervised Anomaly Detection". In: *International Conference on Learning Representations* (2018). URL: https://openreview.net/forum?id=BJJLHbb0-.

[3] Markus M. Breunig et al. "LOF". In: *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, May 16-18, 2000, Dallas, Texas, USA* 29.2 (2000), pp. 93–104. ISSN: 0163-5808. DOI: 10.1145/342009.335388. URL: https://www.researchgate.net/publication/221214719_LOF_Identifying_Density-Based_Local_Outliers.

[4] Zekai Chen et al. *Learning Graph Structures With Transformer for Multivariate Time-Series Anomaly Detection in IoT*. 2022. DOI: 10.1109/JIOT.2021.3100509. URL: http://arxiv.org/pdf/2104.03466.

[5] Chen Qiu et al. "Latent Outlier Exposure for Anomaly Detection with Contaminated Data". In: *International Conference on Machine Learning* (2022), pp. 18153–18167. ISSN: 2640-3498. URL: https://proceedings.mlr.press/v162/qiu22b.html.

[6] Kukjin Choi et al. "Deep Learning for Anomaly Detection in Time-Series Data: Review, Analysis, and Guidelines". In: *IEEE Access* 9.99 (2021), pp. 120043–120065. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2021.3107975. URL: https://www.researchgate.net/publication/354155314_Deep_Learning_for_Anomaly_Detection_in_Time-Series_Data_Review_Analysis_and_Guidelines.

[7] D. Berndt and J. Clifford. "Using Dynamic Time Warping to Find Patterns in Time Series". In: *KDD Workshop* (1994). URL: https://www.semanticscholar.org/paper/Using-Dynamic-Time-Warping-to-Find-Patterns-in-Time-Berndt-Clifford/1ac57524ba2d2a69c1bb6defed7352a06fd7050d.

[8] Ailin Deng and Bryan Hooi. "Graph Neural Network-Based Anomaly Detection in Multivariate Time Series". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 35.5 (2021), pp. 4027–4035. ISSN: 2374-3468. DOI: 10.1609/aaai.v35i5.16523. URL: https://ojs.aaai.org/index.php/AAAI/article/view/16523.

[9] *Herrenknecht.Connected*. Herrenknecht AG. May 22, 2024. URL: https://www.herrenknecht.com/en/services/herrenknechtconnected/.

[10] S. Hochreiter and J. Schmidhuber. "Long short-term memory". In: *Neural computation* 9.8 (1997), pp. 1735–1780. ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735.

[11] Xiaohang Jin et al. "Anomaly Detection and Fault Prognosis for Bearings". In: *IEEE Transactions on Instrumentation and Measurement* 65.9 (2016), pp. 2046–2054. ISSN: 0018-9456. DOI: 10.1109/TIM.2016.2570398.

[12] Minkyung Kim et al. "An Iterative Method for Unsupervised Robust Anomaly Detection Under Data Contamination". In: *IEEE transactions on neural networks and learning systems* PP (2023). DOI: 10.1109/TNNLS.2023.3267028.

[13] M. Ester et al. "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise". In: *Knowledge Discovery and Data Mining* (1996). URL: https://www.semanticscholar.org/paper/A-Density-Based-Algorithm-for-Discovering-Clusters-Ester-Kriegel/5c8fe9a0412a078e30eb7e5eeb0068655b673e86.

[14] Ruei-Jie Hsieh, Jerry Chou, and Chih-Hsiang Ho. "Unsupervised Online Anomaly Detection on Multivariate Sensing Time Series Data for Smart Manufacturing". In: *IEEE International Conference on Service-Oriented Computing and Applications* (2019). URL: https://www.semanticscholar.org/paper/Unsupervised-Online-Anomaly-Detection-on-Sensing-Hsieh-Chou/f2dc4fb527a4165efbed142338d00cd56d61d8b7.

[15] Nesma A. Saleh et al. *A Review and Critique of Auxiliary Information-Based Process Monitoring Methods.* URL: http://arxiv.org/pdf/2110.00198.

[16] Zuogang Shang et al. "Core loss: Mining core samples efficiently for robust machine anomaly detection against data pollution". In: *Mechanical Systems and Signal Processing* 189 (2023), p. 110046. ISSN: 0888-3270. DOI: 10.1016/j.ymssp.2022.110046. URL: https://www.sciencedirect.com/science/article/pii/S0888327022011141.

Appendices

A Acronyms

TBM Tunnel Boring Machine

2D-Mx 2D matrix

AD Anomaly detection

AE Autoencoder

AR Autoregression - Health index

ARIMA Auto-regressive Integrated Moving-average

ARIMAX ARIMA exogenous

ATF-UAD Adversarial Time-Frequency Reconstruction Network for Unsupervised Anomaly Detection

AlexNet AlexNet

Bayesian Bayesian network

CIA-GNN Correlation- and Interaction-Aware Anomaly Detection

CNN Convolutional neural networks

Classical Classical methods

Cluster Clustering-based methods

Contamination Deal with contamiated training data for UAD

Control chart Control chart

Core loss Core loss

Cube data cube technique

DAAE Distribution alignment autoencoder

DAGMM Deep Autoencoding Gaussian Mixture Model

DBSCAN Density-based spatial clustering of applications with noise

DCAD Densely Contrastive Anomaly Detection

DCGAN multi-time scale deep convolutional generative adversarial network

DD - LCSS -DTW Derivative distance with DTW and LCSS

DTW Dynamic time warping

Deep Deep leaning

Density Density-based methods

Deviation Data deviation AD

Dim reduc Dimensional reduction

Distance Distance-based methods

EKF Extended Kalman Filter

ELM-MI Extreme learning machine and mutual information

ERP Edit Distance on Real Sequence

Euclid Euclidean distance

FT Fourier transform

Forcasting Forcasting

GAN Generative adversarial network

GAN-DM Distance image GAN

GDN Graph Deviation Network

GMDNN Gaussian mixeddensity neural network

GMM Gaussian mixture model

GNN Graph Neural Network

GRU Gated recurrent units

GSAE Attention graph stacked autoencoder

GTA Graph learning with Transformer for Anomaly detection

HTM Hierarchical temporal memory

IAD Iterative Anomaly Detection

ILSFS Knowledge-based target variables into inner feature selectors

IU Independent univariate

Index Index monitoring

LOE Latent Outlier Exposure

LOF local outlier factor

LSC local sparsity coefficient

LSTM Long short term memory neural networks

LUAD Lightweight unsupervised anomaly detection

Linear Linear model-based methods

LogLM Log-linear mixture

MDEF multi-granularity deviation factor

MGM multivariate Gaussian Models

MSE - L2 Least squard method with L2 regularisation

MVDD Multiscale support vector data description

NegCo Negative correlation

OCSVM One-class support vector machine

PCA Principal component analysis

PIP Pairwise inner-product

PP Pairwise phase

PRDAGMM Pyramid Reconstruction Assisted Deep Autoencoding Gaussian Mixture Model

R-PCA recursive principal component analysis

RNN Recurrent neural networks

RP-LREC Recurrence plot with local recurrence rates

RPCA Robust PCA

SAE Stacked autoencoder

SARIMA Seasonal ARIMA

SHAP Shapley Additive explanation

SPREAD Sparse Recurrent Neural Network based Anomaly Detection

STW Stationary wavelet transform

TCN Temporal convolutional networks

THOC Temporal hierarchical one class network

Topologie Multivariate topologies

Tree Decision tree

Trfmr Transformer

VAE Variational Autoencoder

VAR Vector autoregression

Vector Support vector machines

add-on Methode to addapt existimg techniaque to solve specific problem

KNN K-nearest-neighbor