Outlier classification using Autoencoders: application for fluctuation driven flows in fusion plasmas

R. Kube[1] F. M. Bianchi [1] D. Brunner[2] B. LaBombard[3]

[1]Department of Physics and Technology, UiT - The Arctic University of Norway
[2]Commonwealth Fusion systems
[3]MIT Plasma Science and Fusion Center

May 30, 2018

UNIVERSITY OF TROMSØ

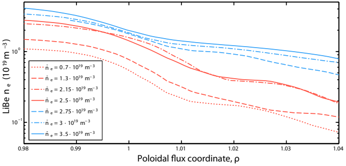
UiT / THE ARCTIC UNIVERSITY OF NORWAY

PSFC Plasma Science and Fusion Center

Alcator C-Mod

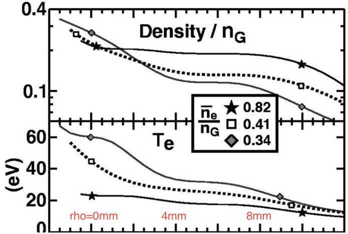
Universality: Density profiles in the SOL broaden with increasing plasma line-averaged density

ASDEX Upgrade



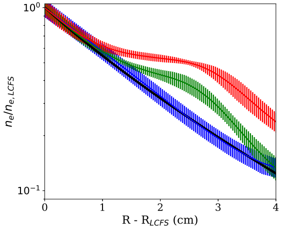
D. Carralero et al. NF **54** 123005 (2015)

Alcator C-Mod



B. LaBombard et al. PoP **8** 2107 (2001)

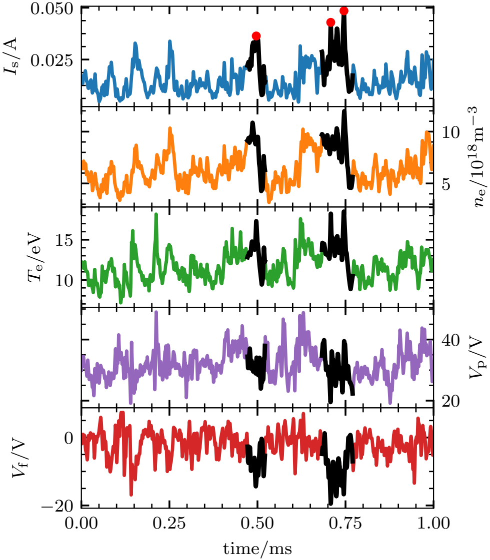
MAST



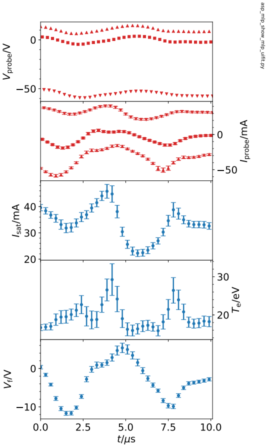
N.R. Walkden et al. PPCF **59** 085009 (2017)

How are changes in fluctuation driven flows connected to this broadening?

Working hypothesis: Fluctuation driven ExB flows govern SOL dynamics



MLP measures plasma state parameters on time scales shorter than the turbulent flows



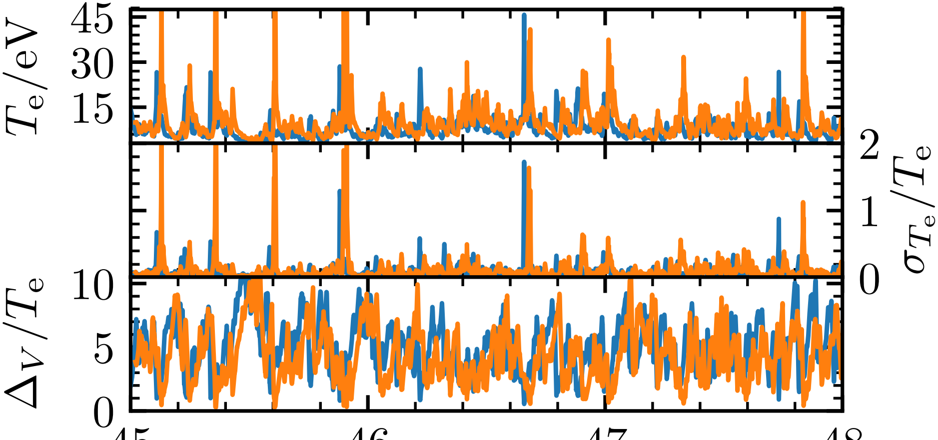
$$I_{\mathrm{probe}} = I_{\mathrm{sat}} \left[\exp\left(\frac{V_{\mathrm{probe}} - V_{\mathrm{f}}}{T_{\mathrm{e}}} \right) - 1 \right]$$

- Turbulence time scale $t_{\mathrm{turb}} \approx 10\mu\mathrm{s}$.
- Classical Langmuir probes: $t_{\mathrm{sweep}} \approx 1\mathrm{ms}$
- MLP electronics switches between V^+, V^0, V^- in $1\mu\mathrm{s}$
- Attempt Fit on U-I characteristic
- Map I_{sat}, V_{f}, T_{e} one-to-one on I_{probe}, V_{probe} samples

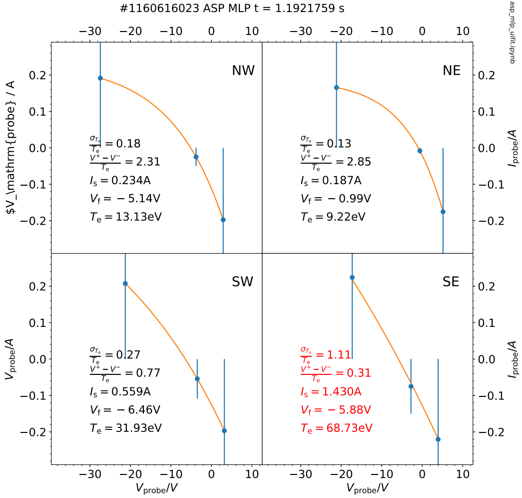
Problem: How do we set V^+, V^0, V^-?

- Use average T_{e} sample value from last $1\mathrm{ms}$
- Intermittent, large-amplitude bursts require large fit domain.

Some large amplitude T_e-peaks are inconsistently identified



Poor fits can be identified through T_e, σ_{T_e} and $\triangle V / T_\mathrm{e}$



Error threshold allow to identify good and bad samples. What about the rest?

Quantity	relaxed	mid	strict
$T_{\mathrm{e}}/\mathrm{eV}$	45/50	40/45	35/40
$\sigma_{T_{\mathrm{e}}}$	0.75/1	0.5/0.75	0.25/0.5
$\triangle V/T_{\mathrm{e}}$	2.5/1.5	3/2	3.5/2.5
uncertain/ *bad*	**20.3%** / 0.1%	**30.0%** / 0.1%	**40.2%** / 0.2%

outliers: ≥ 2 bad fits

inliers: > 2 good fits

uncertain: neither condition is fulfilled

Idea: Label uncertain data as valid or invalid, depending on how "close" they are to good/bad data.

Problem: The data is 12-dimensional.

Dimensionality reduction

- Reduces the number of random variables in the data by obtaining a set of principal variables.
- Two different approaches: feature selection and **features extraction**.
- Feature extraction transforms the data in the high-dimensional space to a space of fewer dimensions.

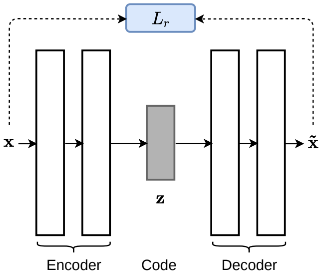
Anomaly detection

- Identification of items, events or observations which do not conform to an expected pattern or other items in a dataset.
- Methods based on dimensionality reduction procedures: anomalous samples do not belong to the subspace containing nominal data learned during training.
- Subspace is computed considering only samples of a nominal class (in our case, measurements with a good fit).
- The representations generated for samples of a new, unseen class will arguably fail to capture important characteristics of the data.
- Such representations will be very different from the ones of good measurements.
- Easy to discriminate between good and bad measurements in the low dimensional space.

Principal Component Analysis

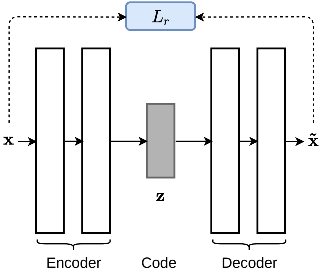
- Performs a linear mapping of the data to a lower-dimensional space in such a way that the variance of the data in the low-dimensional representation is maximized.
- The new space is spanned by the first eigenvectors of the empirical covariance matrix.
- PCA is a linear method and captures only 2nd order moments of variations among the data.
- Nonlinear models, such as kernel PCA and Autoencoder, learn nonlinear embeddings of the data.
- Those methods can model higher order dependencies in the data.

Autoencoders



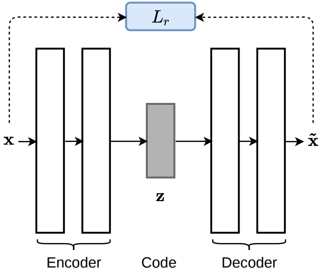
- AEs are a particular class of neural networks, which learn unsupervised compressed, or lossy, representations of data.
- AEs are trained to map the input into a lower dimensional space through a bottleneck layer and then reconstruct the original input.
- The output of the innermost layer of the network $\mathbf{z}$ is called *code* and is the low dimensional representation of the input $\mathbf{x}$.

Autoencoders



- AE learns two functions at the same time. The first one is called *encoder* and provides a mapping from an input domain, $\mathcal{X}$, to a code domain, $\mathcal{Z}$, i. e. the latent representation space.
- The second function, called *decoder*, implements a mapping from $\mathcal{Z}$ back to $\mathcal{X}$.

Autoencoders

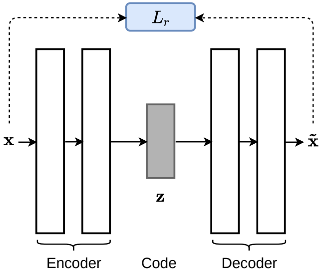


- The encoding function $E(\cdot) : \mathcal{X} \rightarrow \mathcal{Z}$ and the decoding function $D(\cdot) : \mathcal{Z} \rightarrow \mathcal{X}$ of the AE define the following deterministic posteriors

$$\mathbf{z} = E(\mathbf{x}) = p(\mathbf{z}|\mathbf{x}; \theta_E)$$
$$\tilde{\mathbf{x}} = D(\mathbf{z}) = q(\tilde{\mathbf{x}}|\mathbf{z}; \theta_D),$$

- θ_E and θ_D are the trainable parameters of the two functions.
- The encoding and decoding function are usually implemented as two feed-forward neural networks, which are constrained to be *symmetric*.

Autoencoders



- To minimize the discrepancy between $\mathbf{x}$ and $\tilde{\mathbf{x}}$, the parameters θ_E and θ_D are adjusted by minimizing through stochastic gradient descent the following reconstruction loss

$$L = L_r + \lambda L_2 = \mathbb{E}_{\mathbf{x}\sim\mathcal{X}}\left[\|\mathbf{x} - \tilde{\mathbf{x}}\|^2\right] + \lambda\left(\|\theta_E\|^2 + \|\theta_D\|^2\right).$$

- The term L_r minimizes the mean squared error between original inputs and their reconstructions.
- L_2 penalizes large model weights. The hyperparameter λ controls the latter contribution to the total loss.

Autoencoders hyperparameters

- Regularization parameter λ for the L_2 norm penalty in the loss function L.
- Network configuration (number of layers and neurons per layer).
- Probability p_{drop} to drop neural connections during the training (prevents overfitting).
- Learning rate η used in stochastic gradient descent:

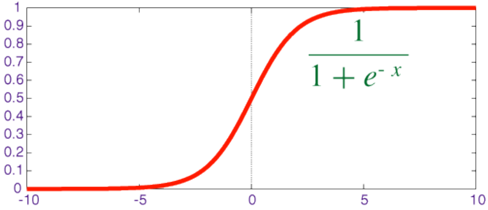
$$\Theta_{k+1} = \Theta_k + \eta \nabla L(\Theta_k).$$

  with $\Theta = \{\theta_E, \theta_D\}$.
- Type of activation function implementing the non-linearities within each AE layer.
  - In case of fully connected layers, each layer output is

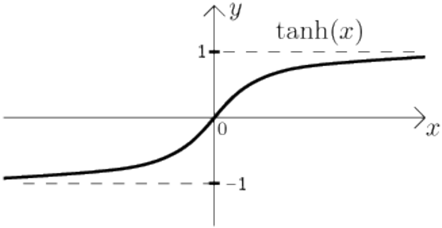
$$\mathbf{x}_t = f(\mathbf{W}_t \mathbf{x}_{t-1} + \mathbf{b}_t),$$

    with $f()$ the activation function.

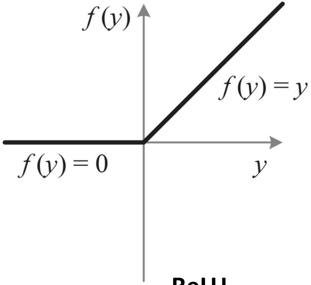
Activation functions



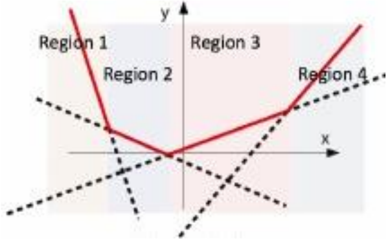
Logistic function



tanh



ReLU



Maxout (k=4)

Classification

- One the AE is trained on good data $\mathcal{X}^g$, both good and bad data are processed to obtain the low dimensional representations $\mathcal{Z}^g$ and $\mathcal{X}^b$.
- A classifier is trained to discriminate between $\mathcal{Z}^g$ and $\mathcal{X}^b$.
- Thanks to the AE pre-processing, the class should be easier to separate, compared to the original input space.
- In our work, we considered:
  - Support Vector Machine classifier;
  - Least square classifier;
  - Prototype classifier.

Prototype classifier

- For each class c, a prototype is computed as

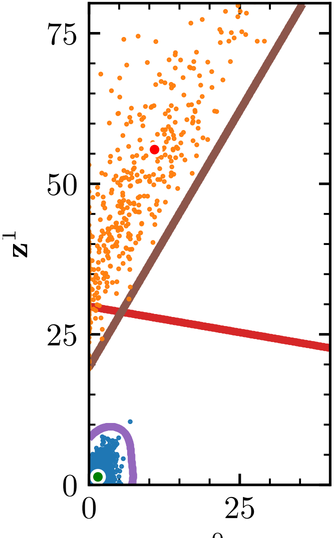
$$\mu_c = \frac{1}{|\mathcal{X}^c|} \sum_{i \in \mathcal{X}^c} x_i \tag{1}$$

- The class label ℓ of an uncategorized data sample $\bar{x}$ is assigned as

$$\ell = \mathrm{argmin}_{j \in \{g,b\}} \|\bar{x} - \mu_j\|^2 \tag{2}$$

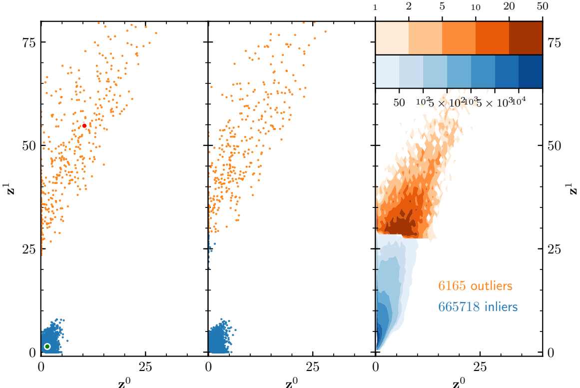
- This classifier does not depend on any hyperparameter and requires to maintain only the representative of each cluster to classify new data.
- Due to its simplicity, this classifier cannot identify complex decision boundaries to separate samples of different classes.
- Is a viable option for easily separable data.

Classifier algorithms learn a decision boundary in code space from labelled data

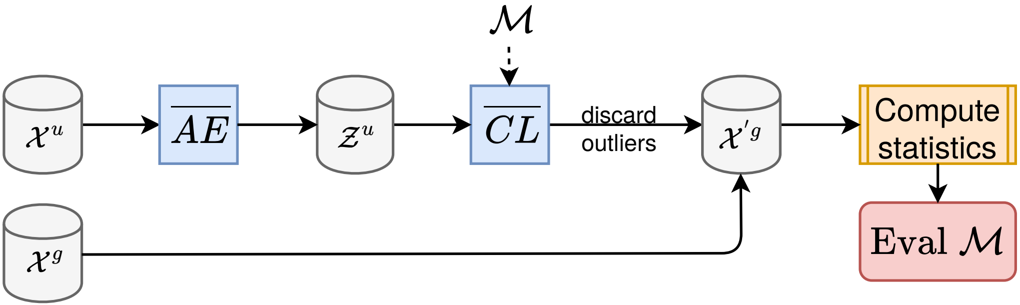


- Least-squares classifier (brown): Tight boundary around outliers
- Nearest prototype (Red): Boundary approximately equidistant between prototypes
- SVM with Radial basis function kernel (purple): Tight boundary around inliers

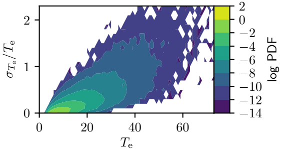
Assign label to uncategorized data in code space



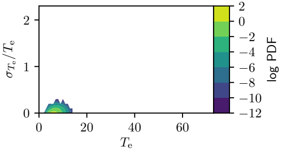
Proposed pipeline to find the optimal classifier $\mathcal{M}$



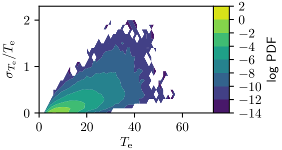
Classifiers remove qualitatively different samples



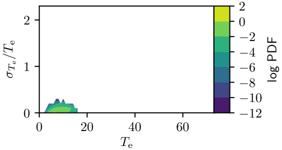
(a) All data, $\mathcal{X}$



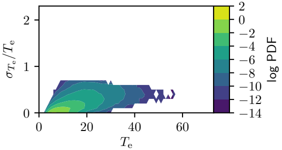
(b) Only good data, $\mathcal{X}^g$



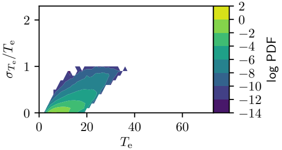
(c) No bad data, $\mathcal{X} \setminus \mathcal{X}^b$



(d) SVC

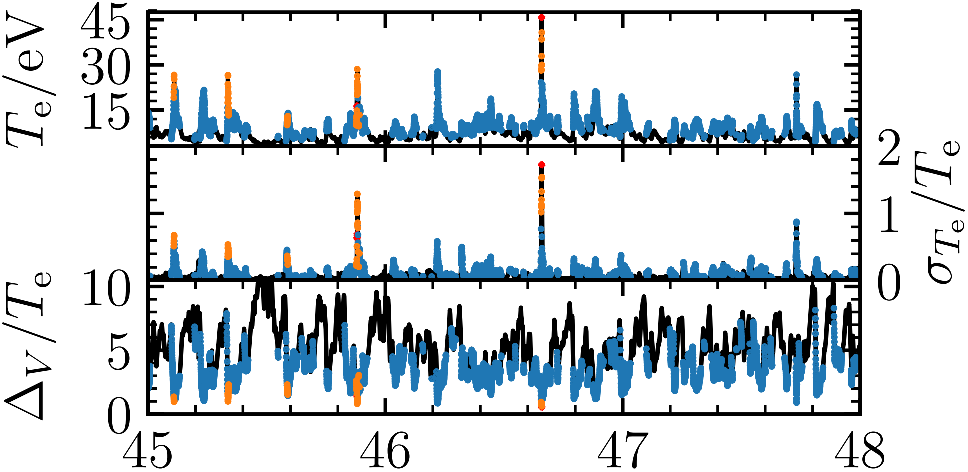


(e) Nearest prototype



(f) Least squares

Large amplitude fluctuations are often identified as outliers.



Removing outliers reduces mean contributions by 10 – 20%.

		$\mathcal{X}$	$\mathcal{X}^g$	$\mathcal{X} \setminus \mathcal{X}^b$	$\mathcal{X}'^g_{\text{pro}}$	$\mathcal{X}'^g_{\text{SVC}}$	$\mathcal{X}'^g_{\text{lsq}}$
$\Gamma_{T,\text{cond}}$	Mean	21.0	1.83	19.4	18.3	9.93	17.0
	Std	101	11.9	82.2	74.4	32.8	66.0
$\Gamma_{T,\text{conv}}$	Mean	11.8	2.18	11.3	11.0	7.18	10.4
	Std	38.8	8.83	35.0	33.1	19.3	39.0
$\Gamma_{T,\text{tcor}}$	Mean	8.72	-0.093	6.58	5.72	1.21	4.65
	Std	102	2.63	59.0	49.5	13.3	39.0
Γ_T	Mean	41.4	3.92	37.3	34.9	18.3	32.1
	Std	232	19.8	170	151	61.0	130

Heat flux in units of heat flux, in units of $10^{20}\,\mathrm{eV m^{-2} s^{-1}}$