GLAD: **GL**ocalized **A**nomaly **D**etection via Active Feature Space Suppression

September 30, 2018

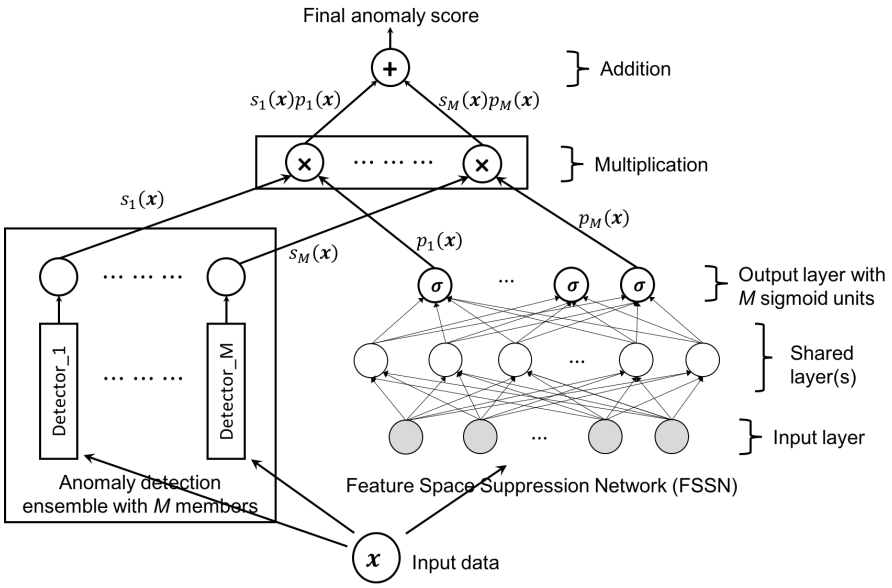
1 Introduction

Approach In order to customize an anomaly detector $\mathcal{D}_m$ for a particular task, we start by decomposing the anomaly score it assigns to an instance $\mathbf{x}$ into two parts: **(a)** the original score $s_m(\mathbf{x})$, and **(b)** the *relevance* $p_m(\mathbf{x}) \in [0, 1]$. The score assigned by $\mathcal{D}_m$ is then $s_m(\mathbf{x})p_m(\mathbf{x})$. The overall anomaly score is computed as: $score(\mathbf{x}) = \sum_{m=1}^{M} s_m(\mathbf{x})p_m(\mathbf{x})$.

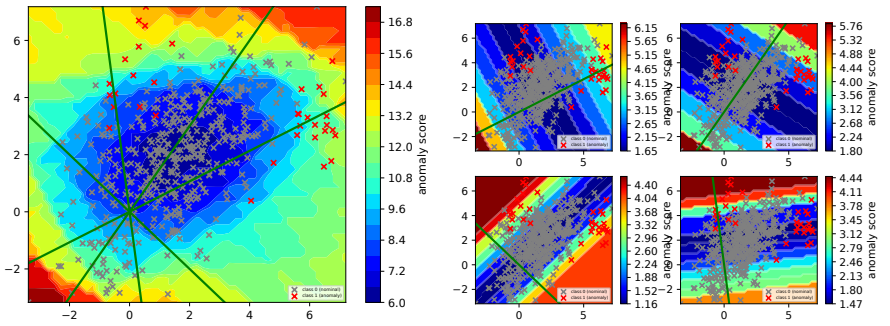
Initially, each detector is assumed to be uniformly relevant in every part of the input space. This assumption is implemented by priming a neural network referred to as *FSSN* to predict the same probability value $b \in [0, 1]$ for every input using the ℓ_{prior} loss in Equation 1. *In effect, this imposes* ***an uniform prior over the input space*** $\mathcal{X}$ (rather than the parameter space) for the relevance of each detector. When all detectors have the same relevance, the final anomaly score simply corresponds to the average score across all detectors (up to a multiplicative constant), and is **a good starting point for active learning.** Next, the algorithm receives label feedback from an analyst and determines whether the ensemble made an error (i.e., assigned a high score to a nominal). If so, the FSSN tries to suppress all errant detectors using a combination of ℓ_{prior} and the AAD loss ℓ_{AAD} for similar inputs: $\ell_{FSSN}(\mathbf{x}) = \ell_{AAD}(\mathbf{x}) + \lambda\ell_{prior}(\mathbf{x})$.

$$\ell_{prior}(\mathbf{x}) = \sum_{m=1}^{M} -b \log(p_m(\mathbf{x})) - (1 - b) \log(1 - p_m(\mathbf{x})) \tag{1}$$

Toy Data The *Toy* dataset and the corresponding LODA ensembles have been shown below as an illustration of how GLAD works. We find that GLAD learns useful relevance information that can be of help to the analyst.



GLAD Architecture. The anomaly detection ensemble contains M detectors. We assume that these are all pre-trained and cannot be modified. The final layer of the Feature Space Suppression Network (FSSN) contains M sigmoid outputs, with each one paired with a corresponding ensemble member. Each output node in the FSSN is initially primed to predict the same probability (0.5 in our experiments) across the entire feature space. It then receives feedback from the analyst and learns which parts of the feature space are ***relevant*** for each detector. For an instance $\mathbf{x}$, $s(\mathbf{x})_m$ denotes the score assigned to it by the m-th detector, while $p(\mathbf{x})_m$ denotes the probability computed by the FSSN that the m-th detector is relevant. The final anomaly score for an instance $\mathbf{x}$ is the sum of all its scores from each detector weighted by the corresponding relevances.



(a) Baseline LODA score contours (b) Score contours for each LODA projection

(c) Initial uniform relevance ($b = 0.5$) (d) Relevance after 30 feedback iterations

Toy data. More red on the **top row** indicates more *anomalous*. More red on the **bottom row** indicates more *relevant*. The red '×' are true anomalies and grey '×' are true nominals. **(a)** LODA with four projections (green lines) applied to the *Toy* dataset. **(b)** The contours of only the *bottom left* LODA projection are somewhat aligned with the true anomalies, i.e., most anomalies lie in the higher anomaly score regions. Other projections are highly inaccurate. **(c)** The output nodes of the FSSN are initially primed to return a relevance of 0.5 everywhere in the feature space. **(d)** The points circled in green were shown to the analyst for labeling, one per feedback iteration. After 30 iterations, the bottom left projection was found to be most relevant in the top-right half-space, whereas it is completely irrelevant in the bottom-left half-space. Other projections were less relevant in most parts of the feature space.