Mini-Project - MVA 2021/2022
Anomaly detection using LSTM networks

Mohammed Hssein mohammed.hssein@ens-paris-saclay.fr
Soufiane Fafe soufiane.faf@gmail.com

April 27, 2022

1 Introduction and contributions

In this mini-project, we work on an interesting topic in machine learning for time series that is the problem of anomaly detection. Anomaly detection, is about finding in a given time series, the points that represent an unusual behavior compared to the way the time series behave usually. Many phenomenon can cause anomalies in time series, as for example **distribution shift**, **noise**, ... etc. There are plenty of methods that have been designed to address such task. Many of these methods can be categorized in what we can call **model-free** methods, meaning they do not need to use any black-box model to detect anomalies. Other approaches suggest to make use of machine learning models that are able to well model the behavior of the time series and use them to detect anomalous behavior. Many machine learning models can be applied, as for example SVMs and random forests. However, the findings in the field of deep learning in recent years, have lead to the LSTM (Long Short Term Memory) networks that are improvements in term of ability to model Long time dependencies over the classic RNNs (Recurrent neural networks). This models have shown tremendous success particularly in modeling time series. The paper we worked on for this project, makes use of **Stacked LSTMs**, ie sequence of LSTMs, and developed a **Model-based** approach for anomaly detection. In the sequel, we will explain the method we used from the main paper [1], we discuss the experiments we did, and discuss the results we have found.

We note that the code we used for these experiments is a pure product of our work. We used the deep learning library **TensforFlow**, to build and train our model. We reproduced two experiments from the main paper using real datasets as **ECG** and **Respiration** [2]. We even did further experiments to understand what could be the drawbacks of such method. We found that if the time series we use $\{\mathbf{y}_t\}_{t\in\mathbb{N}}$ is noisy for example with the classic formulation $\mathbf{y}_t = \mathbf{x}_t + \epsilon_t$ where $\epsilon_t \sim \mathcal{N}(0, \sigma)$ are iid noise values, then we cannot be confident enough about the anomalies we detect. The code we provided, does not include of course the hyper-parameters search we did separately. In the notebook, the hyper-parameters we used are the final best ones. Our contribution for making this mini-project is equal, each student has read, understood the method from the article and developed a part of the code.

[1] https://www.researchgate.net/publication/304782562_Long_Short_Term_Memory_Networks_for_Anomaly_Detection_in_Time_Series

[2] https://www.cs.ucr.edu/~eamonn/discords/

2 Method

Let us consider a time series $X := \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, ..., \mathbf{x}^{(n)}\}$ in which each point $\mathbf{x}^{(t)} \in \mathbb{R}^m$, ie each term is a multi-dimensional vector $\mathbf{x}^{(t)} = \{x_1^{(t)}, ..., x_m^{(t)}\}$. We assume, that there are samples within X that are anomalous, ie mark a change in the normal behavior of the data in X. The Goal is then to detect these points with a good accuracy. The method we aim to use here is a **model-based** anomaly detection method that utilizes a deep learning model known as **Stacked-LSTMs** model.

The intuition behind the method is quite simple. Indeed, let us assume that the deep learning model is well trained to fit some **normal** data that does not contain any kind of anomalies. If new data that has never seen before by the model during the training, comes from the same distribution as for the training data, then the model should be capable of well predicting its values. On the other hand, if the new unseen data, contains some perturbations, then the predictions must be far from the data values. We only then have to find a threshold τ so that the problem of detecting anomalies becomes a simple binary classification problem.

More concretely, suppose our time series X is divided into two parts, one with contains anomalies, we denote by X_A, and another part which contains only normal behavior we denote by X_N. We divide X_A into two subsets called validation anomalous data denoted by v_A and a test anomalous data denoted by t_A. We divide the normal data X_N into three sets, S_N a training data used to train the model to fit the time series, v_{N_1} a validation set used during the training process for monitoring, v_{N_2} a validation data used along with v_A to estimate a certain **threshold** τ, and finally t_N to test the fit of the model to the time series. The method then is based on two parts:

- **Stacked-LSTM based prediction model**: We have a model of two LSTMs stacked with 16 units each, trained using S_N, and v_{N_1} sets to predict l steps in the future based on history of h time stamps in the past. We train the model for 100 epochs, using a batch-size of 64. In the original paper, the authors use **Early stopping** criterion which is a form of algorithmic regularization. We instead used $L2$ penalty in the weights of the first LSTM, and we also used **Dropout** which consists on shutting down some neurons randomly during the training. It has been proven experimentally that this technique acts also as a regularization. For the optimizer, we used **Adam** optimizer, with 10^{-3} learning rate.
- **Errors modeling**: Once the model is trained, we compute the prediction errors on the test set t_N. We denote the set of these vectors $\mathbf{e} := \{\mathbf{e}_1, ..., \mathbf{e}_n\}$. We model these error vectors, with a multivariate Gaussian distribution $\mathcal{N}(\mu, \Sigma)$. To estimate its parameters, we use the Maximum Likelihood Estimation. We can easily show that $\hat{\mu} := \frac{1}{n}\sum_{i=1}^{n} \mathbf{e}_i$ and $\hat{\Sigma} := \frac{1}{n}\sum_{i=1}^{n}(\hat{\mu} - \mathbf{e}_i)(\hat{\mu} - \mathbf{e}_i)^T$. For future error vectors g_t, we have $p^t := \mathbb{P}(g_t|\text{model, data}) = \mathcal{N}(g_t|\hat{\mu}, \hat{\sigma}) = \frac{1}{(2\pi)^{\frac{l}{2}}|\hat{\Sigma}|^{\frac{1}{2}}} \exp\big(-a(g_t)\big)$ is the probability that models the errors where $a(x) := (\hat{\mu} - x)^T\hat{\Sigma}^{-1}(\hat{\mu} - x)^T$ is called the **Mahalanobis distance**. An observation is more likely to be an anomalous observation if and only if is located in one of the ques of the distribution. We then have to find a threshold τ such that if $\mathbb{P}(g_t|\text{model, data}) \leq \tau$ then g_t is classified as anomaly (positive class), and vice versa. At this level, we use v_{N_2} and v_A to estimate τ. The paper suggests to choose as criterion a value of τ that maximizes the F_β score for a small value of β ($\beta = 0.1$ [3]). We note at this level that the condition $p^t := \mathbb{P}(g_t|\text{model, data}) \leq \tau$ is

[3] In fact Recall is proportional to the True Positives. However is our case, anomaly regions in data are very small,

equivalent to $\frac{1}{2}a(g_t) + C > \log \frac{1}{\tau}$

- **Anomaly detection**: Once the threshold τ is estimated, we can apply the model to detect anomalies. Here we use the anomalous data t_A we did not used yet. We feed the data to the model and get our predictions. Then we compute predictions errors as in the previous step. We then are able to evaluate the probabilities of the errors conditioned on the model and the data, and thus we can easily classify times series points as normal or anomalous. Instead of computing probabilities, we can directly work on $-\log$ of the probabilities, since $-\log \mathbb{P}(\text{error vector}|model, data) = \frac{1}{2}a(\text{error vector}) + C$, and compare it directly to $\log \frac{1}{\tau}$.

3 Data

As we stressed before, we have reproduced two experiments from the main paper using **ECG** dataset, and **Respiration** datasets. To study the impact of adding noise to the data on the method, we used a synthetic dataset we designed by hand. The synthetic data we designed is a periodic data we generated using the sum of three sin function with three different frequencies $f = 50$, $f = 100$, and $f = 300$. We repeated the records for more than 300 periods. Each dataset among the three, has been standardized before the use. For Respiration dataset, only standerdizing has been performed. The ECG dataset, we used, comes with one anomaly interval. At the first attempt we used it after a standardization step directly. At a second attempt, we designed many similar anomalies intervals and retried the experiment. The performance of the method gets significantly better. See 1.

4 Results

In the first experiment, we used only a synthetic data to test the method, and also to try to understand its weaknesses. We studied the effect of corrupting the clean data with Gaussian noise with various noise levels. We can clearly see that when noise becomes significant, many normal values are classified as anomalies. See 3 for qualitative results and 2 for quantitative results. In the second experiment, we used real data to see how the method can perform in more realistic scenarios. The following figure 1 2, and table 1 illustrate the results.

Dataset	R2 score	MSE	Recall	Precision	$F_{0.1}$ score
ECG (one anomaly block)	0.99	0.003	0.03	0.34	0.32
ECG (Many anomaly blocks)	0.99	0.045	0.08	0.55	0.53
Respiration	0.98	0.008	0.02	1.0	0.68

Table 1: Metrics for performance evaluation

which will lead to small values of Recall and large values of precision. We beleive that authors have chosen small values of β to counter this effect by reducing the precision contribution while keeping a good compromise between precision and recall.

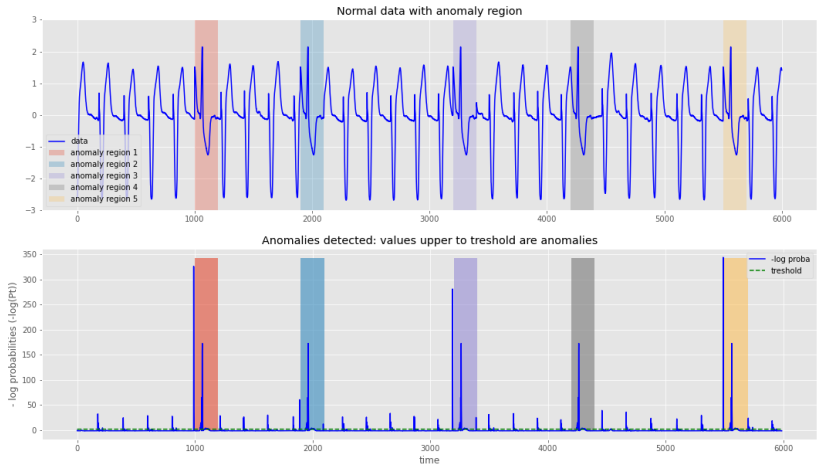


Figure 1: Anomaly detection using ECG dataset

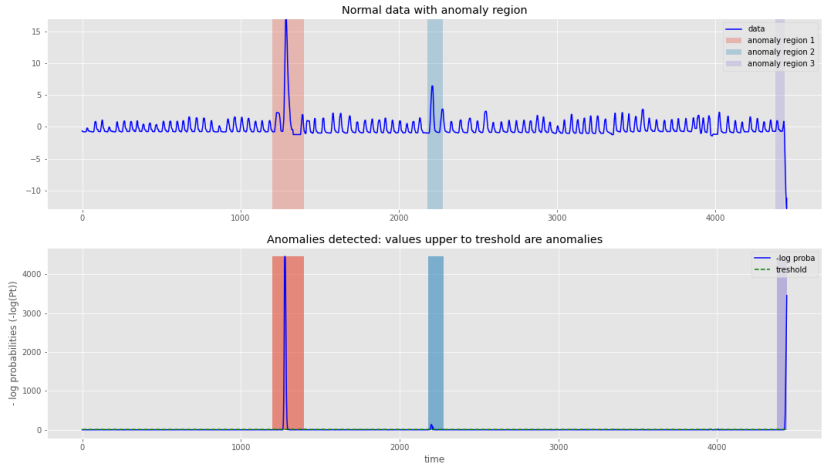
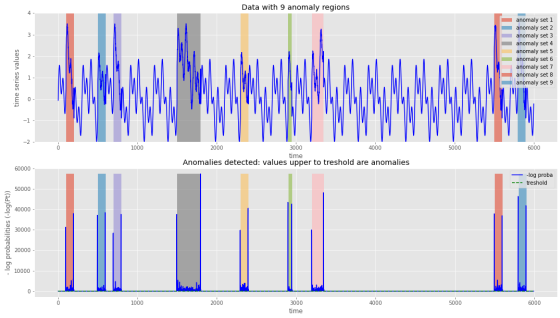
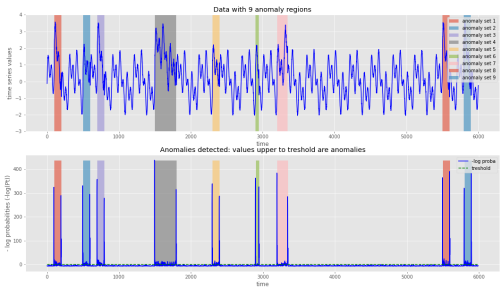


Figure 2: Anomaly detection using Respiration dataset

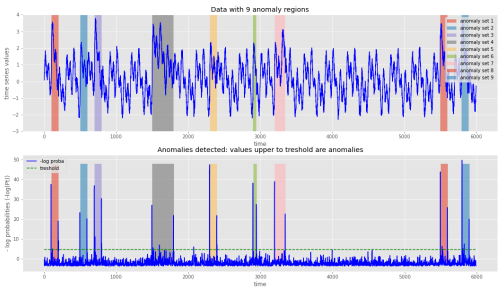
We note that, in each figure the threshold τ is displayed, and thus values higher than the threshold level are more likely to be anomalies, and other values are considered to be normal. In 3, we can clearly see that for $\sigma = 0.8$, we are not confident at all, since many many of the normal points are classified to be anomalous. The percentage of anomalous regions in the time series is also an important factor in our analysis as it impacts the evaluations metrics (recall and precision). The more the data contains anomalous intervals, the more the estimation of the threshold τ is confident, and thus we can trust the detection.



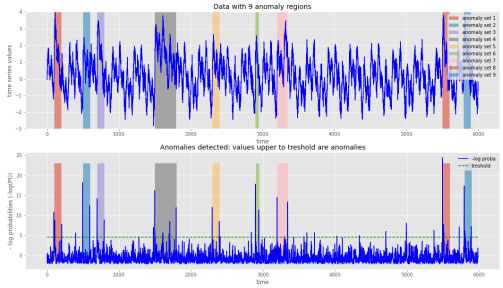
(a) Data without any noise ($\sigma = 0$)



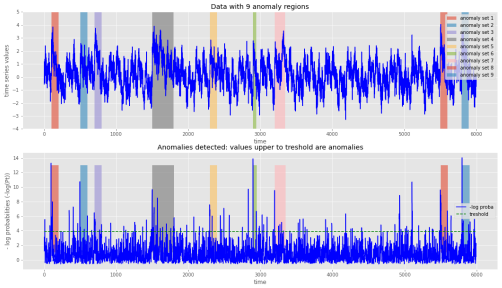
(b) noise level $\sigma = 0.05$



(c) noise level $\sigma = 0.2$



(d) noise level $\sigma = 0.4$



(e) noise level $\sigma = 0.8$

Figure 3: Effect of the noise level on the performance

Noise level	R2 score	MSE	Recall	Precision	$F_{0.1}$ score
$\sigma = 0.0$ (no noise)	0.997	0.002	0.19	1.0	0.95
$\sigma = 0.05$	0.98	0.006	0.08	0.99	0.88
$\sigma = 0.2$	0.93	0.058	0.01	0.96	0.57
$\sigma = 0.4$	0.83	0.15	0.01	0.74	0.51
$\sigma = 0.8$	0.58	0.38	0.03	0.59	0.51

Table 2: Metrics for performance evaluation