

Active Learning for LSTM-autoencoder-based Anomaly Detection in Electrocardiogram Readings

Tomáš Šabata¹ and Martin Holeňa²

¹ Faculty of Information Technology, Czech Technical University in Prague,
Prague, Czech Republic

`tomas.sabata@fit.cvut.cz`

² Institute of Computer Science of the Czech Academy of Sciences,
Prague, Czech Republic

`martin@cs.cas.cz`

Keywords: Active Learning, Anomaly detection, LSTM-Autoencoder, Time series

1 Introduction

Recently, the amount of generated time series data has been increasing rapidly in many areas such as healthcare, security, meteorology and others. However, it is very rare that those time series are annotated. For this reason, unsupervised machine learning techniques such as anomaly detection are often used with such data. There exist many unsupervised algorithms for anomaly detection ranging from simple statistical techniques such as moving average or ARIMA till complex deep learning algorithms such as LSTM-autoencoder. For a nice overview of the recent algorithms we refer to read [2,1].

Difficulties with the unsupervised approach are: defining an anomaly score to correctly represent how anomalous is the time series, and setting a threshold for that score to distinguish between normal and anomaly data. Supervised anomaly detection, on the other hand, needs an expensive involvement of a human expert. An additional problem with supervised anomaly detection is usually the occurrence of very low ratio of anomalies, yielding highly imbalanced data.

In this extended abstract, we propose an active learning extension for an anomaly detector based on a LSTM-autoencoder. It performs active learning using various classification algorithms and addresses data imbalance with over-sampling and under-sampling techniques. We are currently testing it on the ECG5000 dataset from the UCR time series classification archive [3].

2 Active learning for LSTM-autonecoder-based anomaly detection

LSTM-autoencoder [9] is nowadays increasingly used to detect anomalies in time series data [11,5,4]. The algorithm aims to learn the identity function. It consists of two parts – an encoder and a decoder. The encoder compresses the

input representation of the data into a low-dimensional latent representation (usually called *code*) from which the decoder reconstructs the original input. The model parameters are found by minimizing the reconstruction error. Samples are then considered anomalous if their reconstruction error is higher than a selected threshold.

Although LSTM-autoencoder works well for time-series with complicated patterns, setting the anomaly score threshold can be very complicated without labelled data. Furthermore, with a higher ratio of anomalies present in the training data, a simple setting of the threshold might produce a lot of false positives and false negatives. Therefore, we incorporated active learning into building the anomaly detector that uses the code of a previously trained LSTM autoencoder. The most related research has been done by Pigmentel [8], who proposed to use a classifier (logistic regression) with the latent layer of autoencoder together with anomaly score. In contrast, we experiment with the latent layer of a recurrent autoencoder without anomaly score and propose to use resampling techniques.

First, an LSTM autoencoder is trained on unlabelled data. An initial anomaly detection threshold on reconstruction error distribution is selected. At the initial value of the threshold, we decided to use the mean plus three times the standard deviation of the reconstruction error as an initial threshold. Every sample with the reconstruction error above the threshold is labelled as an anomaly and the same number of samples below the threshold are labelled as normal. Furthermore, instead of the original time-series data, we use their representation in the code layer. This provides us an artificially created, balanced dataset and converts the anomaly detection to a binary classification task. A classifier is trained on the created dataset and an active learning loop starts. In each iteration, a resampler is used to balance the new updated labelled dataset. The resampler can be either an undersampling or oversampling algorithm. The classifier is fitted using the resampled data. Uncertainty sampling (US) active learning framework [7] is used to select instances which should be labelled by an oracle. We use margin US, i.e. we select instances leading to the smallest difference between the likelihood of anomalous and normal data classes. Anomaly detection is then based on predictions of that classifier instead of on the anomaly threshold. The pseudo-code for the algorithm is shown in Algorithm 1.

3 Experiment and Results

The proposed algorithm was evaluated on a benchmark time-series dataset with electrocardiogram readings [3]. Each heart-beat record in the dataset is labelled with one of 5 classes where the last three classes are very rare and we consider them as an anomaly. The dataset was split into training, validation and testing in the ratio 70:15:15. The validation dataset was used to find the hyperparameters for the LSTM-autoencoder. The autoencoder achieving the lowest f1 score in the anomaly detection on the validation data set was chosen. The final architecture of the encoder consisted of two LSTM cells. The cells have one hidden layer with 48 neurons in the first cell and 24 neurons in the second cell. The hidden state

Algorithm 1: Active learning for LSTM-autoencoder anomaly detection

```

Input:
 $\mathcal{U}$ : unlabelled data set of sequences
 $\theta$ : anomaly score threshold
 $\phi(\cdot)$ : query strategy utility function
begin
  train LSTM autoencoder on data set  $\mathcal{U}$ 
  // calculate anomaly scores
   $a_i = |x_i - \text{decoder}(\text{encoder}(x_i))|, x \in \mathcal{U}$ 
  sort  $a$  in the descending order
  sort  $x$  respectively to  $a$ 
   $i = 0, \mathcal{L} = \emptyset$ 
  // Add anomalous data samples into the labelled dataset
  while  $a_i > \theta$  do
     $\mathcal{L} = \mathcal{L} \cup \langle \text{encoder}(x_i), 1 \rangle$ 
     $\mathcal{U} = \mathcal{U} \setminus x_i$ 
     $i = i + 1$ 
  end
  // Add normal data samples into the labelled dataset
  for  $j = i$  to  $2i$  do
     $\mathcal{L} = \mathcal{L} \cup \langle \text{encoder}(x_j), 0 \rangle$ 
     $\mathcal{U} = \mathcal{U} \setminus x_j$ 
  end
  while stopping criterion is not met do
     $\mathcal{R} = \text{resampler}(\mathcal{L})$ 
    train binary classifier  $m$  on  $\mathcal{R}$ 
    // Find the most informative sequence from  $\mathcal{U}$  and ask for label
     $x^* = \text{argmax}_{x \in \mathcal{U}} \phi(x)$ 
     $y^* = \text{query}(x^*)$ 
     $\mathcal{L} = \mathcal{L} \cup \langle x^*, y^* \rangle$ 
     $\mathcal{U} = \mathcal{U} \setminus x^*$ 
  end
end

```

of the last cell is copied and used as the input of the first cell of the decoder. The decoder has architecture mirrored.

We experimented with 5 classifiers: logistic regression, decision tree, Gaussian naive Bayes classifier, k-nearest neighbours classifier and support vector machines. We experimented with 11 under-sampling and 4 over-sampling techniques taken from the imbalanced-learn python toolbox [6]. In the presented results, we report 5 under-sampling techniques that were best on average. In the experiment, we compared a fully unsupervised approach, in which anomalies are detected by using a chosen threshold, with our extension with respect to F1 score.

Figure 1 shows how actively asking for annotations can improve the unsupervised anomaly detection with an LSTM-autoencoder (red dashed line). The

best results were achieved with repeated edited nearest neighbours [10] and a k-nearest neighbour classifier. However, using an SVM as the base classifier yielded more stable performance.

Moreover, a classifier model fed by labels outperformed the LSTM-autoencoder with the initially set anomaly score threshold (red dashed line) and the best possible anomaly score threshold (green dashed line). The source code is available in GitHub repository ³.

4 Conclusion

We presented an active Learning for LSTM-autoencoder-based anomaly detection for time-series data. An experiment on the ECG5000 data set has shown that the proposed method is able to boost the performance of the model significantly with only approximately 200 labelled samples. We plan next to experiment with variational LSTM-autoencoders and to pay attention to the interpretability of the detected anomalies .

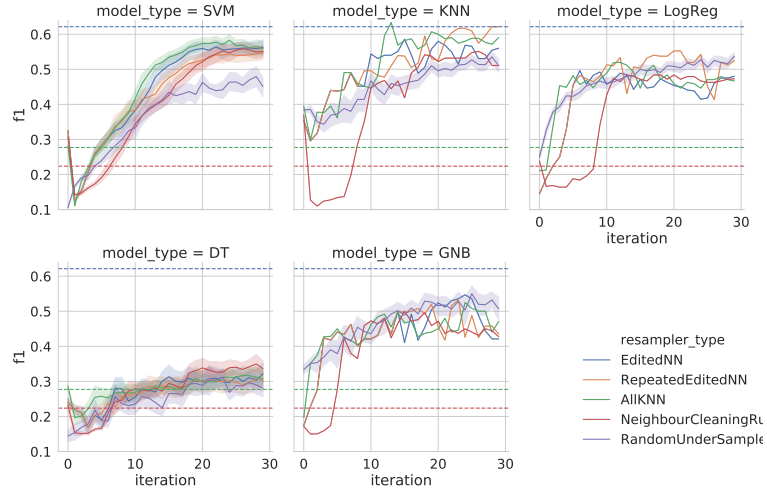
Acknowledgements

The work has been supported by the grant 18-18080S of the Czech Science Foundation (GAČR).

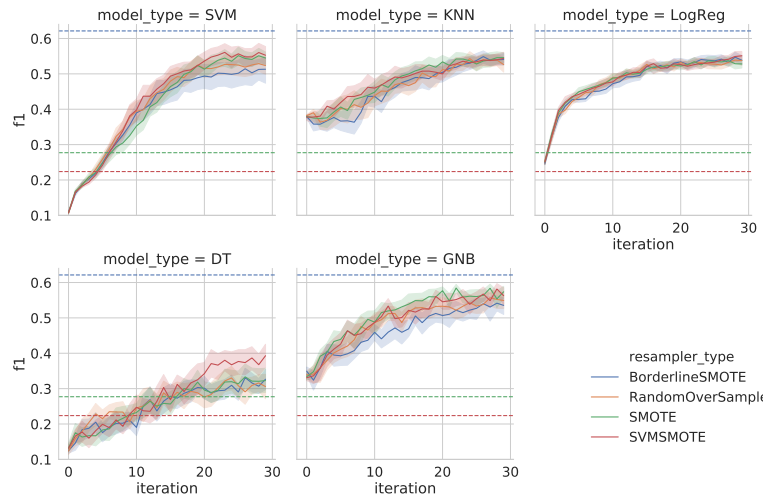
References

1. Ahmad, S., Lavin, A., Purdy, S., Agha, Z.: Unsupervised real-time anomaly detection for streaming data. *Neurocomputing* **262**, 134–147 (2017)
2. Cook, A., Misirlı, G., Fan, Z.: Anomaly detection for iot time-series data: A survey. *IEEE Internet of Things Journal* (2019)
3. Dau, H.A., Keogh, E., Kamgar, K., Yeh, C.C.M., Zhu, Y., Gharghabi, S., Ratanamahatana, C.A., Yanping, Hu, B., Begum, N., Bagnall, A., Mueen, A., Batista, G., Hexagon-ML: The ucr time series classification archive (October 2018), https://www.cs.ucr.edu/~eamonn/time_series_data_2018/
4. Ergen, T., Mirza, A.H., Kozat, S.S.: Unsupervised and semi-supervised anomaly detection with lstm neural networks. arXiv preprint arXiv:1710.09207 (2017)
5. Guo, Y., Liao, W., Wang, Q., Yu, L., Ji, T., Li, P.: Multidimensional time series anomaly detection: A gru-based gaussian mixture variational autoencoder approach. In: *Asian Conference on Machine Learning*. pp. 97–112 (2018)
6. Lemaître, G., Nogueira, F., Aridas, C.K.: Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *The Journal of Machine Learning Research* **18**(1), 559–563 (2017)
7. Lewis, D.D., Gale, W.A.: A sequential algorithm for training text classifiers. In: *SIGIR'94*. pp. 3–12. Springer (1994)
8. Pimentel, T., Monteiro, M., Viana, J., Veloso, A., Ziviani, N.: A generalized active learning approach for unsupervised anomaly detection. *stat* **1050**, 23 (2018)

³ https://github.com/tsabata/active_anomaly_detection



(a) Under-sampling techniques.



(b) Over-sampling techniques.

Fig. 1: F1 score performance metric in active learning loop. The figure contains 5 models (support vector machines (SVM), k-nearest neighbours (KNN), logistic regression (LogReg), decision tree (DT) and Gaussian naive Bayes (GNB)) and 9 resampling techniques. The greyed area represents standard deviation. Red dashed line represents the performance of the LSTM-autoencoder anomaly detector without active learning and initial setting of the anomaly score threshold. Green dashed line represents the performance of the LSTM-autoencoder anomaly detector without active learning and the best setting of the anomaly score threshold, and blue dashed line represents the best-achieved performance in the last iteration of active learning loop.

9. Srivastava, N., Mansimov, E., Salakhudinov, R.: Unsupervised learning of video representations using lstms. In: International conference on machine learning. pp. 843–852 (2015)
10. Tomek, I., et al.: An experiment with the edited nearest-neighbor rule. *IEEE Transactions on Systems, Man, and Cybernetics* **6**, 448–452 (1976)
11. Zhang, C., Chen, Y.: Time series anomaly detection with variational autoencoders. arXiv preprint arXiv:1907.01702 (2019)