# LSTM autoencoder for anomaly detection on ECG data

R. Lorusso

2023 - University of Bari Aldo Moro

## Abstract

*The aim of this work is to provide a model for effectively identify anomalies in vital parameters time series selected from VitalDB dataset [1]. The employed model is an LSTM autoencoder as a tool to identify anomalies in such temporal data by the evaluation of the mean reconstruction error. Furthermore an adaptive threshold selection based on the training data reconstruction error it's proposed, with the aim of improving the explainability of the model without relying on an arbitrary threshold.*

## 1 Introduction

The detection of anomalies, also known as outliers, is an integral practice across a diverse range of disciplines. Anomalies can indicate a breach of a system or network, signal abnormal physiological levels, or even flag the occurrence of fraud. Regardless of the particular application, analytics and machine learning models have the potential to provide both predictive and descriptive value. Anomaly detection can be used to inform about anomalous data that can be indicative of health complications. This kind of information can be very useful to professionals as a first insight to a possible patient problem. Since providing care and track patient vitals signals can be hard to maintain trough the time, machine learning can be very helpful by automatically detect rare deviations from normal data trends, thus providing health care professionals with useful information with which make more accurate and quicker clinical decisions. However what is to be considered as an anomaly isn't ever clearly defined. The main assumptions are that anomalies are infrequent an somehow they differ from normal data. Furthermore anomalies can differ substantially one from each other, thus introducing more uncertainty about their nature. Sometimes oscillations in the test data can be detected as anomalies, introducing false alarms. Especially in the e-health domain, we are interested in reducing or minimizing the number of false alerts, since an high false alarm rate can decrease the trust in the system and in the alarm itself. In order to cope with these problems, an adaptive threshold selection based on the mean reconstruction error on training data is proposed. In section 2 we'll start with a formal definition of a general autoencoder and how it can be employed to identify anomalies. Section 3 describes the dataset and its selected features through a preliminary exploratory data analysis. Then we can move to section 4, focusing our attention on the features and preprocessing strategy. Finally in section 5 it's introduced the model implementation, which is further evaluated in section 7 by means of the obtained results.

## 2 Employed model

Autoencoders are widely used for anomaly detection tasks in a variety of different fields and for many purposes. The choice of an LSTM autoencoder is motivated by the use of time series in health domain [1]. An autoencoder is a feed forward neural networks used to learn a latent representation of unlabeled input data and it is trained to reconstruct the input at the output layer. Autoencoders are made by an encoder and a decoder, which can be described by two maps $\Phi$ and $\Psi$ on $X \in \Re^n$, $Y \in \Re^d$, $d < n$ such as:

$$\Phi : X \to Y$$
$$\Psi : Y \to X$$

for which the goal is to minimize the mean squared error between the input and the output layer

$$argmin_{\Phi,\Psi}||X - (\Psi o \Phi)X||^2.$$

The main assumption is that the data fed to the model in the training phase is to be considered normal. Since the neural network encodes an input data point in a latent space learned from normal data, and decodes it from the very same space to reconstruct its original form, we can evaluate the degree of fitness to the model by measuring the reconstruction error through a distance metric. If the reconstruction error is large enough, then we can assume that the tested sample can be considered an anomaly since it doesn't fit the latent space, being different from the training data.
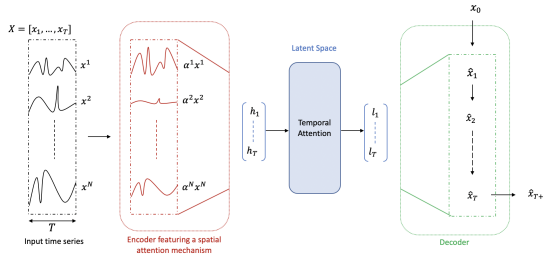
**Figure 1:** General autoencoder architecture



**(a)** ASA $< 3$      **(b)** ASA $== 3$

**Figure 2:** Data distribution

# 3 Dataset Description

The dataset employed is called VitalDB [1], it contains high-resolution multi-parameter data from 6,388 cases of surgical patients. The mean age of the patient is 57 years. Among the plethora of different features, the following have been selected:

- Diastolic blood pressure,

- Systolic blood pressure,

- Body temperature,

- Heart rate,

- Respiratory rate,

which are the basic vital parameters, useful for a first assessment of a patient health state. Since the dataset contains records of vital parameters of surgical patient, the ones with the most healthiest values have been selected by means of the ASA physical status system [2]. Only those records with an ASA grade less than three are retained for training the model, resulting in the distribution depicted in the diagram 2a. A percentage of this data is held out for testing and evaluating the model, which will be used further for detecting anomalies on the portion of dataset with ASA equal to three i.e. "patients with severe systemic disease". From diagram 2b it's possible to observe how the distribution of such data it's more irregular and slightly shifted to the right side of the chart for the first three parameter, while for the distribution of respiratory rate and body temperature values is very similar.
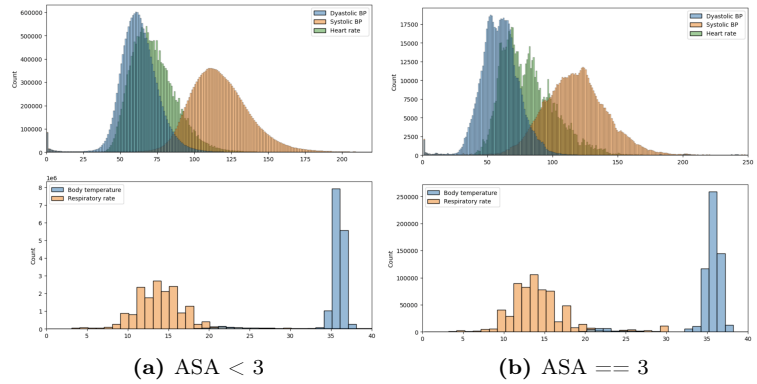
Since every feature presents anomalous values, the following section describes the preprocessing strategy applied to retain only the features values which can be considered normal. Before going into details of the preprocessing strategy it's worth to notice that the data has been collected from surgical patients during a perioperative period [1]. Considering this context it is possible to deduce that the results obtained by a model trained on such data could be useful in the same kind of context but may be wrong in another one, such as during physical activity. We will talk about this in the section 9 related to the ethical issues.

# 4 Feature Engineering

As anticipated, it has been needed to retain only the features values which can be considered healthy. This is done by following both common knowledge, as in the case of body temperature, and international approved guidelines [3] as in the case of the blood pressure.

## 4.1 Data cleaning

Every feature records have been cleaned from NaN and negative values, which where consistently present into the data. Then time series are cleaned from values outside the following ranges:

- Diastolic blood pressure: 40-100;

- Systolic blood pressure: 85-140;

- Body temperature: 35-37.2;

- Heart rate: 40-120;

- Respiratory rate: 8-22.

These range values are a bit wider than the optimal ones to preserve some variability and noise in the data without making the model too rigid in detecting anomalies. However restricting the data in a range can introduce sudden changes in time series values, thus

negatively affecting its quality. To partially overcome this problem, the values outside a range are replaced with the mean value of the sample; a better solution would be that of interpolation for smoothing the gap between two values.

After restricting the values in ranges by replacing the outsiders with the sample mean, we obtain a new data distribution depicted in the diagram 3a. Wider data intervals will result in a loosen anomaly detection.
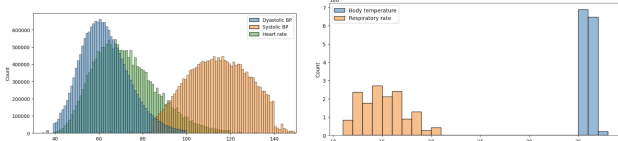


**Figure 3:** Data distribution restricted to ranges

As ultimate step, it is removed every record which is either empty or has negative mean since vital parameters can't have negative values nor be null.

## 4.2 Data bundling and chunking

After the data has been cleansed from anomalous values, the time series along the different features are concatenated to create a unique one. This big sequence is then chunked into ones of equal length to be fed to the model. Since feature values follow a normal distribution, a Std normalization has been applied.

## 5 Model development

The model employed for this task is a multivariate LSTM autoencoder, the choice is motivated by the time series data. It is implemented using Keras library. By the moment that we are in a semi-supervised context since we are selecting the data by means of the ASA parameter, has been possible to hold out a part of the data in order to evaluate the model performances on such testing data. In diagram 4a is depicted the model architecture. As we can see from the input shape, the time series length is equal to 4678. This value can be diminished on enlarged to carry out a more local or global analysis respectively. In this case the input shape is selected as the minimum length of the records across the different features.

```
Model: "sequential"

 Layer (type)                Output Shape          Param #
=================================================================
 lstm (LSTM)                 (None, 4768, 128)     68608

 dropout (Dropout)           (None, 4768, 128)     0

 lstm_1 (LSTM)               (None, 4768, 64)      49408

 lstm_2 (LSTM)               (None, 4768, 32)      12416

 leaky_re_lu (LeakyReLU)     (None, 4768, 32)      0

 lstm_3 (LSTM)               (None, 4768, 64)      24832

 lstm_4 (LSTM)               (None, 4768, 128)     98816

 dropout_1 (Dropout)         (None, 4768, 128)     0

 time_distributed (TimeDistr (None, 4768, 5)       645
 ibuted)

=================================================================
Total params: 254,725
Trainable params: 254,725
Non-trainable params: 0
```

**Figure 4:** LSTM autoencoder architecture

### 5.1 Configuration

The final configuration of the parameters has been obtained after several training sessions. The best results are yielded with the following parameter settings:

- epochs = 50;
- loss function = Mean Squared Error;
- bias regularizer = 0.3;
- recurrent regularizer = 0.1;
- batch_size = 64;
- validation split = 0.1.

The 80% of data with ASA less than three is retained for training and the remaining 20% is held out for testing. In section 7 it is showed a comparison between the mean reconstruction errors made on the training and testing data in order to see if the model behaves as expected by producing a similar mean reconstruction error for the two sets.

## 6 Adaptive threshold selection

As already discussed in the introduction, identification of anomalies relies on the assumption that these are rare and differ from normal data. In the context of autoencoders an anomaly is detected evaluating the reconstruction error by means of a threshold, manually left to the human. Most of the times threshold value is decided by looking to the distribution of reconstruction errors, although recent studies propose standard deviation based methods to adapt such value [4]. In this section is described a simple approach to adapt the threshold by means of the mean reconstruction error made by the autoencoder on training data. Given the reconstruction error for all the training sample it is possible to select the threshold value which encloses a given percentage of data below such value. Since we are interested to identify anomalies, we can start by the mean reconstruction error made on the training set and increment it until the above condition si met.

---
Adaptive threshold selection
---

$X$ {Autoencoder input, array of time series}
$Y$ {Autoencoder output, array of time series}
$n\_samples \leftarrow len(X)$
$mean\_rec\_err$ {Train set mean reconstruction err}
$target \leftarrow 0$
$percentage \leftarrow .98$
$step \leftarrow 1 - percentage$
$factor \leftarrow 1 - step$
**while** $target < percentage$ **do**
   $j \leftarrow 0$
   $factor \leftarrow factor + step$
   **for** i in range(0,n\_samples) **do**
     $err \leftarrow norm2(X[i] - Y[i])$
     **if** $err \leq mean\_rec\_err \times factor)$ **then**
       $j \leftarrow j + 1$
     **end if**
   **end for**
   $target \leftarrow j \div n\_samples$
**end while**
$threshold \leftarrow factor \times mean\_rec\_err$

---

# 7 Experimental results

# 8 Related Works

# 9 Ethical issues and opportunities

# References

[1] Yoon S.B. et al. Lee HC. Park Y. "VitalDB, a high-fidelity multi-parameter vital signs database in surgical patients." In: *Sci Data* 9.279 (2022).

[2] D Mayhew, V Mendonca, and BVS Murthy. "A review of ASA physical status–historical perspectives and modern developments". In: *Anaesthesia* 74.3 (2019), pp. 373–379.

[3] Bryan Williams et al. "2018 ESC/ESH Guidelines for the management of arterial hypertension: The Task Force for the management of arterial hypertension of the European Society of Cardiology (ESC) and the European Society of Hypertension (ESH)". In: *European heart journal* 39.33 (2018), pp. 3021–3104.

[4] James Clark, Zhen Liu, and Nathalie Japkowicz. "Adaptive threshold for outlier detection on data streams". In: *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE. 2018, pp. 41–49.