Adaptive threshold for anomaly detection in vital parameters

R. Lorusso

2023 - University of Bari Aldo Moro

Abstract

The aim of this work is to provide an algorithm for the adaptive threshold selection starting by the training data reconstruction error, to effectively identify anomalies in vital parameters time series selected from VitalDB dataset [1]. This could result in the improvement of explainability in evaluation methods based on the reconstruction error. The employed model is an LSTM autoencoder.

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 into a possible patient problem. Since providing care and tracking patient vitals signals can be hard to maintain over time, machine learning can be very helpful by automatically detecting rare deviations from normal data trends, thus providing health care professionals with useful information with which to make more accurate and quicker clinical decisions. However, what is to be considered an anomaly is never clearly defined. The main assumptions are that anomalies are frequent and somehow differ from normal data. Frustrating anomalies can differ substantially 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 a high false alarm rate can decrease 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 on to Section 4, focusing our attention on the features and preprocessing strategy. Finally, in Section 5 , it introduces 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 the health domain [1]. An autoencoder is a feed-forward neural network 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 Φ and Ψ 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 of 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 and is different from the training data.

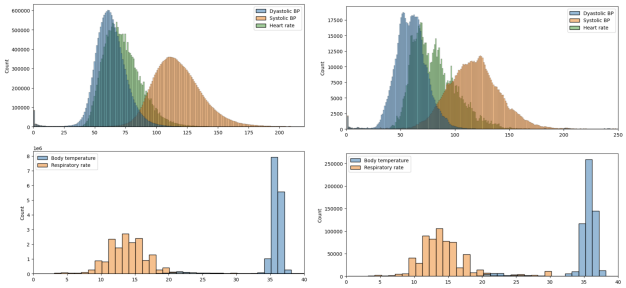
3 Dataset Description

The dataset employed is called VitalDB [1], it contains high-resolution multi-parameter data from 6,388 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 (DBP). Unit: mmHg;
- Systolic blood pressure (SBP). Unit: mmHg;
- Body temperature (BT). Unit: C°;
- Heart rate (HR). Unit: beats/minute;
- Respiratory rate (RR). Unit: breaths/minute;

which are the basic vital parameters useful for a first assessment of a patient's health state. Since the dataset contains records of surgical patients vital parameters, the ones with the healthiest values have been selected by means of the ASA physical status system [2]. Only those records with an ASA grade lower than three are retained for training the model, resulting in the distribution depicted in the figure. 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 the dataset with an ASA equal to three, i.e. "patients with severe systemic disease". From Fig. 1a, it's possible to observe how the distribution of such data is more irregular and slightly shifted to the right side of the chart for the first three parameters, while the distribution of respiratory rate and body temperature values is very similar.



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

Figure 1: Data distribution

Since every feature presents anomalous values, the following section describes the preprocessing strategy applied to retain only the feature values that can be considered normal. Before going into details of the preprocessing strategy, it's worth noting that the data has been collected from surgical patients during the 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 8 related to the ethical implications.

4 Feature Engineering

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

4.1 Data cleaning

Every feature record has been cleaned of NaN and negative values, which were consistently present in the data. Then time series are cleaned from values outside the following ranges:

- Diastolic blood pressure: 30-100;
- Systolic blood pressure: 85-130;
- 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 to 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 could be replaced with the mean value of the sample or interpolated for smoothing the gap between two values. After restricting the values in ranges, we obtain a new data distribution depicted in the figure 2a. Wider data intervals will result in looser anomaly detection.

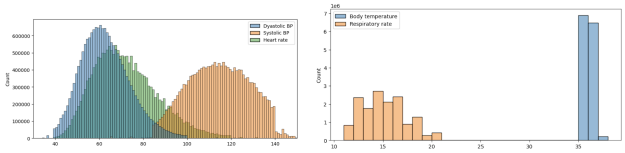
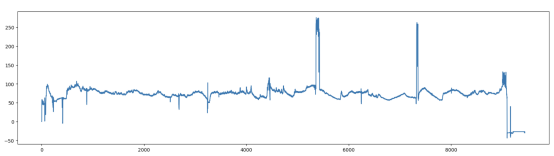
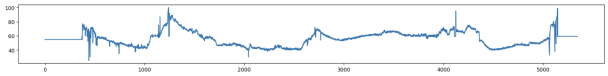


Figure 2: Data distribution restricted to ranges

As the ultimate step, every record that is either empty or has a negative mean is removed since vital parameters can't have negative values or be null. Figure 3 shows an example of a diastolic blood pressure sample, before and after it's cleansed.



(a) Before cleansing



(b) After cleansing

Figure 3: Diastolic blood pressure sample cleansed from noise and anomalous values

4.2 Data bundling and chunking

After the data has been cleansed of anomalous values, the time series along the different features are concatenated to create five unique sequences. These are then chunked into ones of equal length to be fed into the model. This step was necessary because the model expects a uniform sequence length among all the features. Other techniques have been tried, such as padding or truncation, but poor results were obtained. Since feature values follow a normal distribution, mean-variance normalization has been applied.

5 Model development

The model employed for this task is a multivariate LSTM autoencoder, and the choice is motivated by the time series data. It is implemented using the Keras library. In figure 4, the model architecture is depicted. As we can see from the input shape, the time series length is equal to 4678. This value can be diminished or 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 ibuted)	(None, 4768, 5)	645

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.

By the moment that we are in a semi-supervised context since we are selecting the data by means of the ASA parameter, it has been possible to hold out a part of the data in order to evaluate the model performances on such testing data. The 80% of data with an ASA less than three is retained for training, and the remaining 20% is held out for testing. In section 7, a comparison between the mean reconstruction errors made on the training and testing data is shown in order to see if the model behaves as expected by producing a similar mean reconstruction error for the two sets. Finally, the model is employed to reconstruct the population data with an ASA of greater than 3, for which a greater reconstruction error is expected.

6 Adaptive threshold selection

As already discussed in the introduction, the identification of anomalies relies on the assumption that they are rare and differ from normal data. In the context of autoencoders, an anomaly is detected by evaluating the reconstruction error by means of a threshold whose value is set manually. Most of the time, this value is determined by looking at the distribution of reconstruction errors, although recent studies propose standard deviation-based methods to adapt such a value [4]. In this section, a simple approach to adapting the threshold is described on the basis of the mean reconstruction error made by the autoencoder on the training data. Given the reconstruction error for the entire training set, it is possible to select the threshold value that encloses a given percentage of data below such a value. Since we are interested in identifying anomalies, we can start with the mean reconstruction error made on the training set and increment it until the above condition is met. As a result, an input sequence will be considered an anomaly if its reconstruction error differs from a given percentage of train data. A formal definition is given below.

Algorithm 1 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
$threshold \leftarrow factor \times mean\_rec\_err$
return $threshold$

Convergence is guaranteed by additional constraints, such as the maximum number of iterations, not presented in the formal definition. As reported in the algorithm comments, X and Y are arrays of time series; since every feature can be represented as an array of time series, the algorithm has to be executed N times, where N is the number of features. By doing so, it is possible to identify anomalies by analyzing the single features, thus enhancing the model's explainability.

7 Experimental results

In this section are illustrated the results obtained by applying the adaptive threshold selection algorithm on the training data, which will be further used to detect anomalies in testing data by the evaluation of the mean reconstruction error obtained on the features. Figure 5 reports the loss obtained after training the model for 50 epochs. Then the autoencoder is used to perform predictions both on the portion of data held out from the training set and on the data with ASA equal to three. As expected, the reconstruction error of the former is lower than the latter; the results are shown in the table 1 for every feature.

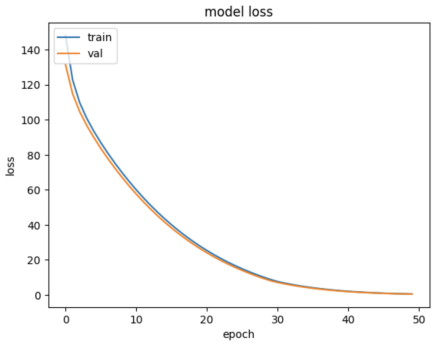


Figure 5: Loss function

7.1 Threshold selection

Features	Train set	Test set	ASA == 3
DBP	7.5	7.6	21.8
SBP	6.3	6.4	16.0
BT	3.8	3.8	15.3
HR	5.8	5.5	10.0
RR	4.8	4.8	15.4
Average	5.6	5.6	15.7

Table 1: Features mean reconstruction error

It is worth noting that the mean reconstruction error of the features is very similar for the training and test sets since they belong to the same population of data, i.e., the data cleansed from anomalous values, whereas the portion of data with ASA == 3 has a significantly greater reconstruction error for all the features. Starting with the values shown in the column Train Set, execution of Algorithm 1 on the training data returned the following threshold values for features:

Features	Threshold
Diastolic Blood Pressure	10.9
Systolic blood pressure	9.3
Body Temperature	6.9
Heart Rate	13.6
Respiratory Rate	11.9

Table 2: Features threshold

This values are depicted in the figure 6. As we can see, the threshold encloses 98% of mean reconstruction errors, and indeed, the outliers are more sparse and distant from the value distribution.

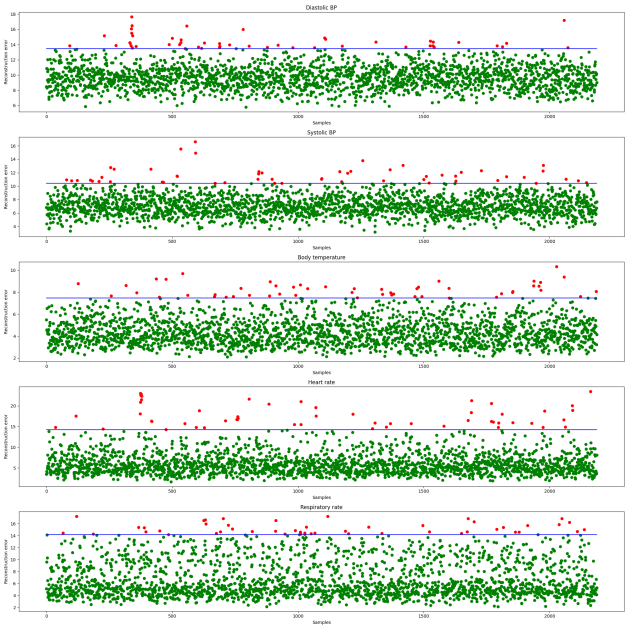


Figure 6: Thresholds obtained from training set

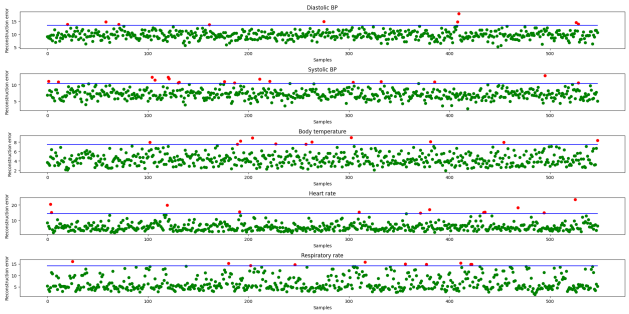


Figure 7: Anomalies in testing data

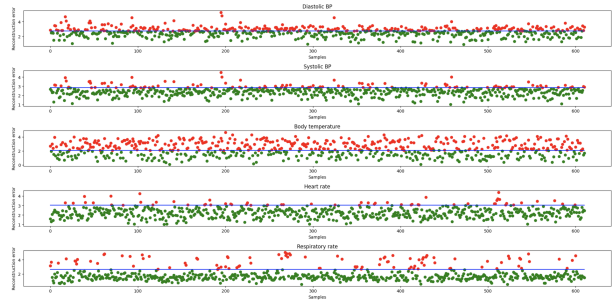


Figure 8: Anomalies in data with ASA == 3

7.2 Anomaly detection on test data

Now it is possible to use the threshold values presented previously to identify anomalies in testing data. Looking at tables 1 and 1, we expect a high anomaly rate on data with an ASA equal to three and a lower one on the held-out data. Table **??** shows the percentage of anomalies identified in the two groups.

Features	ASA < 3 (test set)	ASA $== 3$
DBP	2%	63%
SBP	3%	57%
BT	2%	56%
HR	2%	20%
RR	2%	19%
Average	2.2%	43%

Table 3: Percentage of anomalies in tested data

As expected, the group with an ASA equal to three has a high anomaly rate since it doesn't fit the autoencoder latent space, while the testing group has exactly the same amount of anomalies as the training set from which it was picked. Figures 7 and 8 depict the results obtained by applying the thresholds shown in table 2 to evaluate reconstruction error on such data.

An example of anomaly is shown in figure 9.

Figure 9: Anomaly in test data

8 Ethical implications

As the research in the e-health domain develops, a focus on the ethical implications is needed. As highlighted by Floridi, [5] the explainability is a core ethical principle on which AI-enabled systems must be designed and developed in order to provide useful information about their processes and results. A sufficient, explicable process makes it possible to track the

responsibilities and identify fallacies in an AI-enabled system. This requirement becomes crucial when such systems are employed in the health domain, since it is important to provide reliable information and verifiable processes. In the case of autoencoders where such properties are partially or merely satisfying, it's necessary to verify carefully their results before taking decisions, thus avoiding what Floridi calls "recycling of actions" i.e., acting as a computer suggests, only because we trust the results and how they are obtained. Lastly, the nature and quality of the data used to train AI systems play a central role in their behavior and decision-making. This is why autoencoders like the one developed in this work, which is trained on surgical patient data, should be used to identify anomalies in the very same context where training data is collected since it could be misleading in other circumstances.

9 Further developments

The normal data can be bundled with the training set to expand it. By doing so, it is possible to detect concept drifts and adapt the threshold by running the algorithm 1 on such new data. Other techniques for anomaly detection could be applied to the very same dataset configuration in order to compare the results obtained from the autoencoder. It is also possible to compare normalization schemes and see how they behave.

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.

[5] Luciano Floridi. *Etica dell'intelligenza artificiale: Sviluppi, opportunità, sfide*. Raffaello Cortina Editore, 2022.