Machine learning and deep learning approaches for multivariate time series prediction and anomaly detection

Thill, M.

Chapter 6

Learning Quasi-periodic ECG Time Series with LSTM Networks

6.1 Introduction

The previous chapter introduced the DWT-MLEAD algorithm and demonstrated its effectiveness when applied to time series where anomalies appear on different frequency scales.

However, even with DWT-MLEAD, it is challenging to process time series with complex periodic or quasi-periodic behavior where an anomaly might be a time-shifted peak, a peak with a different shape, or other patterns. For such cases, an algorithm has to learn long-term correlations in the data accurately to detect anomalies reliably. A prominent real-world example where quasi-periodic signals have to be analyzed is in electrocardiography. In an electrocardiogram (ECG), the heart's activity is monitored over time by measuring voltages with several electrodes placed on the chest and limbs. A common task in health care and medical diagnosis is to monitor a patient's heart activity and detect abnormal heartbeat patterns, indicating cardiac arrhythmias or other heart-related problems. There are well-maintained databases, e.g., the MIT-BIH database [52], where medical experts annotate a large body of data with numerous types of anomalies. Automated anomaly detection in such ECG data is still challenging because the deviations from nominal behavior are often subtle and require long-range analysis. Furthermore, there are considerable signal variations from patient to patient or even within an ECG time series (noise, baseline wandering, etc.). A common issue is the heart rate variability (even for healthy hearts, there is a variation in the time interval between consecutive heartbeats of several milliseconds), leading to quasi-periodic signals and increasing the complexity of the analysis.

Long short-term memory (LSTM) networks [63], which are a particular form of recurrent neural networks (RNN) and thus belong to the class of deep learning methods, have proven to be particularly useful in learning sequences with long-range dependencies. They avoid the vanishing gradient problem [62] and are more stable and better scalable [54] than other RNN architectures. LSTMs have been successfully advanced the state-of-the-art in many application areas like language modeling and translation, acoustic modeling of speech, analysis of audio data, handwriting recognition, and others [54]. We will use stacked LSTMs as the building block for our ECG time series prediction.

This chapter presents an unsupervised time series anomaly detection based on LSTM networks that learn to predict the normal time series behavior. The prediction error on several prediction horizons is used to build a statistical model of normal behavior. We propose new methods (window-based error correction and outlier removal) essential for a successful model-building process and a high signal-to-noise-ratio. We apply our method to the well-known MIT-BIH ECG data set and present initial results. We obtain a good recall of anomalies while having a very low false alarm rate (FPR) in a fully unsupervised procedure. We also compare with other anomaly detectors (NuPIC, ADVec) from the state-of-the-art.

In Section 6.2, we will describe an LSTM prediction model that is trained to predict over multiple horizons and is applied to time series containing nominal and rare anomalous data. We observe multidimensional error vectors (one vector for each point in time) and estimate a mean and covariance matrix. Based on the Mahalanobis distance, we can assign a probability of being anomalous at each point in time. Section 6.3 describes our experimental setup and the MIT-BIH Arrhythmia Database used in our experiments. Section 6.4 presents and discusses our results, while Section 6.5 concludes.

6.1.1 Related Work

Publicly, there are only relatively few benchmarks for ECG arrhythmia detection available: The MIT-BIH Arrhythmia database [52, 112, 113], the CU Ventricular Tachyarrhythmia Database [121], and the St. Petersburg INCART 12-lead Arrhythmia Database [52]. We will use the MIT-BIH benchmark in this and the following chapter since it contains the most patients (47 patients) and sufficiently long ECG recordings (30 minutes), and it is the most commonly used benchmark in the literature.

Much work is devoted to anomaly detection in ECG readings: Several authors use multi-resolution wavelet-based techniques [139, 162]. A novelty-search approach on ECG data is taken in [90] in order to perform unsupervised anomaly classification. Sivaraks et al. [151] use motif discovery for robust anomaly detection.

Many articles are concerned with the detection of arrhythmias in ECG signals. However, work on unsupervised approaches appears to be less common in this field. The presented approaches are mostly supervised algorithms that are trained to classify different arrhythmia types. Luz et al. give a comprehensive overview of various classification approaches for ECGs in [101]. In [56], Hannun et al. designed a 34-layer convolutional deep neural network to classify 12 different heart arrhythmia types. However, due to its nature, the architecture is supervised and requires annotated data for training. The authors use a massive labeled dataset containing 91,232 single-lead ECGs from 53,549 patients (not publicly available). The trained model achieves very high accuracy on a cardiologist level. Several researchers base their approaches on the discrete wavelet transform (DWT) [171, 143, 168, 7]. Thomas et al. [168] extract, next to other features (some of which might partially require expert knowledge, such as the RR-intervals), dual-tree complex wavelet-based features from the ECG signal and train a neural network for four arrhythmia classes. We found several

works that introduce anomaly detection methods in ECG readings [107, 161, 28, 2, 22, 151].Sivaraks et al. [151] use motif discovery and propose an approach for robust anomaly detection in ECG data.

The works of Malhotra et al. [107] and Chauhan & Vig [28] are the closest to our current approach. They describe the general idea of LSTM-based prediction and their application to ECG, motor sensor, or power-consumption time series. However, the big drawback of [107, 28] is that they need a manual and supervised separation into up to six data sets: training, validation, test sets which are further subdivided into nominal & anomalous subsets. This means that for a real-world application, the ECG data for a new person would need to undergo an expert anomaly classification prior to any training, which would be highly impractical in most application scenarios. Instead, our method aims to use the whole body of data for a person and train the LSTMs without the necessity to have supervised anomaly information.

While Chauhan & Vig [28] only apply their method to one-minute recordings of the ECG signals in the MIT-BIH [52, 112, 113] corpus, we test our approach on the full-length signals (30 minutes each). Additionally, we only consider those 13 time-series signals with less than 50 abnormal events (ECG-13 data, as described in section 2.3) for our initial tests. In the following chapter 7, we extend the benchmark to 25 time series (ECG-25 benchmark, < 250 events per time series). In [28], also ECG signals are used where the vast majority of heartbeats are considered anomalous (e.g., signals with paced beats). These cases lead to a trivial anomaly detection task since an algorithm can simply flag each heartbeat as anomalous.

6.2 Methods

6.2.1 LSTM for Time Series Prediction

The learning task is formulated as a time series forecasting problem. Hence, we attempt to train a model which can predict future values in the time series. This approach's intuition is that the usual quasi-periodic patterns in the ECG time series should be predictable with only minor errors. At the same time, abnormal behavior should lead to significant deviations in the predictions. Although the presented methodology is only applied to ECG data in this chapter, it is sufficiently general to be applied to other (predictable) time series as well.

6.2.1.1 Data Preparation

Consider a d-dimensional time series of length T. In a first step, it is often recommendable to scale or normalize the time series's dimensions. In our setup, each ECG signal dimension is scaled to the range $[-1, 1]$. The training and test samples are generated by extracting subsequences of suitable length from the original time series. This is done by sliding a window of length W with a lag of 1 over the time series and collecting the windowed data in a tensor

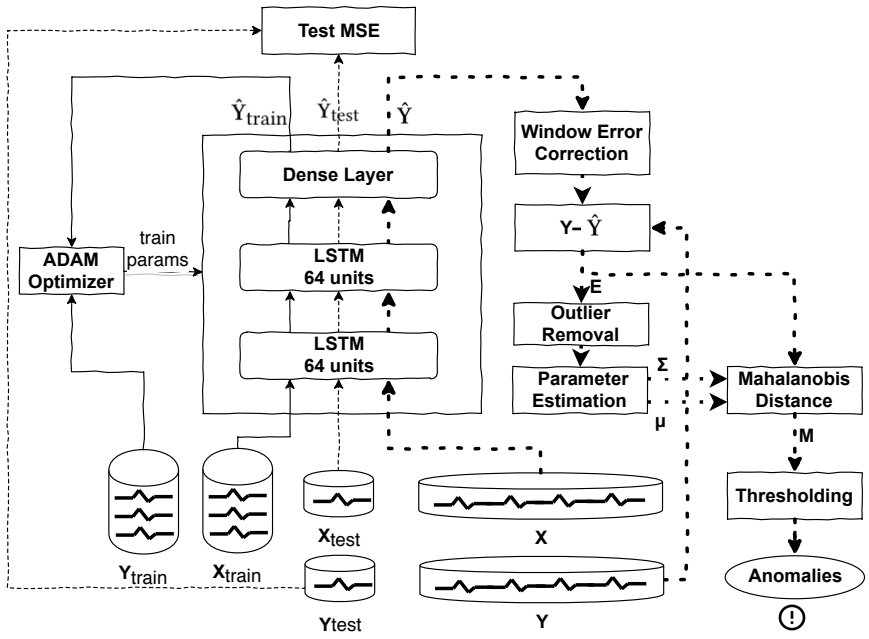

Figure 6.1: LSTM Architecture. The LSTM model is trained with $\mathbf{X}_{\text{train}}$ and $\mathbf{Y}_{\text{train}}$. $\mathbf{X}_{\text{test}}$ and $\mathbf{Y}_{\text{test}}$ are used to verify if the model is overfitting. In order to detect anomalies, the algorithm passes the whole time series $\mathbf{X}$ through the LSTM model and computes residuals for the predicted tensor $\hat{\mathbf{Y}}$.

$\mathcal{D} \in \mathbb{R}^{T' \times W \times d_{in}}$, where T' is the number of sub-sequences. We select all d dimensions of the time series so that $d_{in} = d$. Then, the first 80% of the samples of $\mathcal{D}$ are selected to form the training data $\mathbf{X}_{\text{train}}$. The remaining 20% are used as a test set $\mathbf{X}_{\text{test}}$ to compute an unbiased error estimate later. While the inputs are d_{in}-dimensional, the output-targets for each time step have the dimension m, since one can select for one time series multiple (m) prediction horizons. Technically, it is also possible to predict several time series dimensions with $d_{out} \leq d$ simultaneously in one model; however, in our experiments, the results do not improve in this case (for the investigated ECG-13 time series data). The targets $\mathbf{y}_t \in \mathbb{R}^m$ are future values of the selected signal at times $t + h_i$ for $i \in \{1, ..., m\}$, where the horizons are specified in $H = (h_1, h_2, \ldots, h_m)$. Since we follow a many-to-many time series prediction approach, where the algorithm performs a prediction at each instance of time t, the tensor containing the target signals has the shape $\mathbb{R}^{T' \times W \times m}$ with $T' = T - W - \max(H) + 1$. As before, the first 80% of the targets are used for training ($\mathbf{Y}_{\text{train}}$) and the remaining targets for the test set ($\mathbf{Y}_{\text{test}}$).

6.2.1.2 Model Architecture and Training

A stacked LSTM architecture [63] shown in Figure 6.1 with $L = 2$ layers is used to learn the prediction task. Each layer consists of $u = 64$ units. A dense output layer with m units and a linear activation generates the predictions for the specified prediction horizons in H.

The net is trained with the sub-sequences of length W taken in mini-batches of size B from the training inputs $\mathbf{X}_{\text{train}}$ and targets $\mathbf{Y}_{\text{train}}$. 10% of the training data is held out for the validation set. The LSTM model is trained by using the Adam optimizer [82] to minimize the mean-squared-error (MSE) loss. Other loss functions, such as log-cosh (logarithm of the hyperbolic cosine) and MAE (mean absolute error), were tested as well and produced similar results for our data. We could obtain similar results with the loss functions log-cosh (logarithm of the hyperbolic cosine) and MAE (mean absolute error). Early stopping is applied to prevent overfitting of the model and to reduce the overall time required for the training process. For this purpose, the MSE on the validation set is tracked. For most of the investigated time series, 10-20 epochs are sufficient to reach a minimum of the validation error.

6.2.2 Modeling the Residuals

After the LSTM prediction model is trained, the whole time series $\mathbf{X} \in \mathbb{R}^{T \times d_{in}}$ of length T is passed through the model, and a tensor $\hat{\mathbf{Y}}$ of shape $\mathbb{R}^{T \times m}$ is predicted. Then, the prediction errors $\mathbf{E} = \mathbf{Y} - \hat{\mathbf{Y}}$ are calculated, where $\mathbf{Y}$ contains the true target values for the horizons in H. Now, each row i in the matrix $\mathbf{E}$ represents an error vector $\mathbf{e}_i \in \mathbb{R}^m$.

We noticed in our initial experiments that it was not possible to find good Gaussian fits for the individual dimensions of the prediction errors in $\mathbf{E}$. An example for this is shown in Figure 6.2, upper part. This was due to the fact that the tails of the error distributions contained many outliers, which significantly distorted the estimated fit. Hence, we decided to remove the outliers in each dimension's tails (only during the Gaussian modeling phase). We could find good solutions by discarding the upper and lower 3% quantile in each dimension. We observed that this approach usually removes slightly more than 20% of the data records in our experiments. Note that the Mahalanobis distance itself does not require any particular assumptions about the data distribution (such as normality). However, we found that better results are obtained when removing the outliers in the distribution tails. Experimentally, we also tried using the minimum covariance determinant estimator (MCD) [142] (a robust estimator of mean and covariance matrix) but found that the results are similar to the simple heuristic described above. Since MCD is computationally less efficient, we decide to use our heuristic instead.

After removing the outliers from the prediction errors $\mathbf{E}$, the covariance matrix $\bar{\mathbf{\Sigma}}$ and mean vector $\bar{\mathbf{e}}$ are computed for the cleaned matrix $\mathbf{E}$. Then, the squared Mahalanobis distance

$$M(\mathbf{e}_i) = (\mathbf{e}_i - \bar{\mathbf{e}})^\intercal \bar{\mathbf{\Sigma}}^{-1} (\mathbf{e}_i - \bar{\mathbf{e}}) \tag{6.1}$$

to the mean vector $\bar{\mathbf{e}}$ is determined for each error vector $\mathbf{e}_i$ in $\mathbf{E}$.

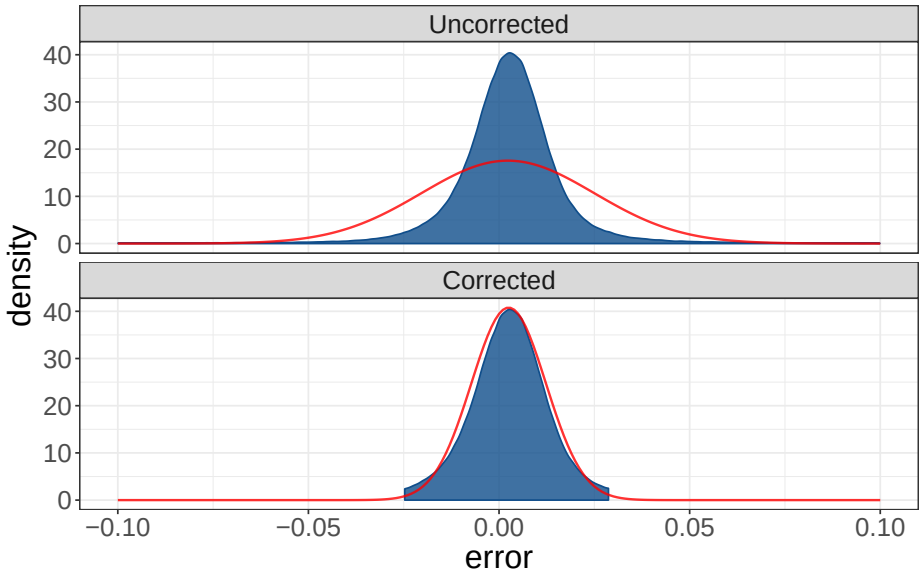

Figure 6.2: Gaussian fit without and with the removal of outliers in the tails of the error distribution. Exemplary, this is shown for one dimension of the overall error distribution. The blue curve shows the empiric error distribution. The red curve depicts the estimated Gaussian. Although our anomaly score metric, the Mahalanobis distance, does not require the assumption of Gaussian-distributed data, this plot nevertheless demonstrates that there are likely many outliers in the error distribution tails which might distort the estimations of mean and covariance matrix.

6.2.3 Anomaly Detection

For most data points in the time series, the corresponding Mahalanobis distance will be comparably small since they are located close to the distribution's mean. On the other side, unusual patterns in the error vectors $\mathbf{e}_i$ – such as large errors in one or more dimensions – will result in large values in the Mahalanobis distance. Therefore, the Mahalanobis distance can be used as an indicator for anomalous behavior in the time-series signal. In our LSTM-AD algorithm, points with a Mahalanobis distance larger than a specified anomaly threshold will be flagged as anomalous. Figure 6.3 shows exemplarily the Mahalanobis distance over time for a selected ECG signal. Depending on the choice of threshold, more or fewer points will be classified as anomalous. If the threshold is set too small, the algorithm will likely produce many false detections. If the threshold is chosen too large, many anomalies might be missed. Ideally, a threshold can be found which allows identifying all anomalies without any false detections. However, in practice, one usually has to trade-off true and false detections and select the threshold according to the own requirements.

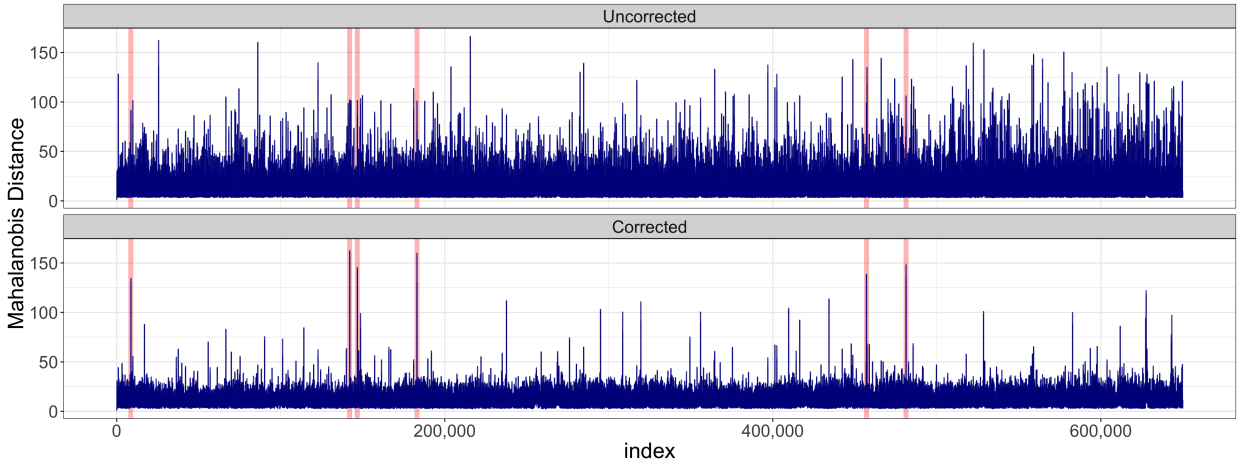

Figure 6.3: Mahalanobis distance over an ECG time series, before and after the window-based error correction method is applied.

6.2.4 Window-Based Error Correction

Initially, we could not obtain very good results when running our algorithm on several example ECG time series. The Mahalanobis distance signal was rather noisy and could not be used to distinguish between nominal and abnormal patterns in the data. In Figure 6.3 (top) this problem is visualized for one example time series. Further investigation showed that the LSTM network predictions were good in general but not sufficiently accurate near the heartbeat peaks. The prediction had been slightly shifted (up to ten time steps) forwards or backward compared to the actual signal. We could identify that the quasi-periodic character of most ECG signals (small but ubiquitous frequency changes in the heartbeat, often referred to as heart rate variability) is the primary source of this problem. The following solution for this problem is proposed: In order to address the variability in the frequency of the signal, small corrections in the predictions of the individual horizons will be permitted. For each output dimension $k \in \{1, \ldots, m\}$ the target values $y_{t,k}$ are compared to the neighbored predictions $\hat{y} \in \hat{Y}_{t,k}^{(win)}$ and the prediction with the smallest absolute error is effectively taken:

$$\hat{y}_{t,k} \leftarrow \arg\min_{\hat{y} \in \hat{Y}_{t,k}^{(win)}} |y_{t,k} - \hat{y}| \tag{6.2}$$

$$\hat{Y}_{t,k}^{(win)} = [\hat{y}_{t-c_k,k}, \ldots, \hat{y}_{t,k}, \ldots, \hat{y}_{t+c_k,k}] \tag{6.3}$$

We found that reasonable results can be achieved with window parameters c_k up to a length of 10, depending on the prediction horizon h_k:

$$c_k = \min(h_k, 10). \tag{6.4}$$

The window-based error correction is applied right after the LSTM prediction of $\hat{\mathbf{Y}}$ and before the prediction errors $\mathbf{E}$ are computed in subsection 6.2.2.

Although this approach corrects the predictions of the LSTM network explicitly with the actual target values $\mathbf{Y}$ of the time series, it is not supervised in the sense that no anomaly labels are presented to the algorithm at any time of the training and correction process. As will be shown in Sec. 6.4, we could significantly improve the performance of LSTM-AD utilizing this correction step.

6.3 Experimental Setup

6.3.1 The ECG-13 Benchmark

In this chapter, we use the ECG-13 dataset, as described in more detail in Section 2.3for our initial experiments. This dataset contains 13 time series taken from the MIT-BIH database [52], each with 650 000 2-dimensional points (which corresponds to roughly 30 minutes). The selection criterion for the 13 time series (patients) and the abnormal classes was such that each time series has 50 or fewer abnormal heartbeats of that class. In total, the ECG-13 dataset contains 130 anomalies from six anomaly classes. Apart from the normalization of the data, no further preprocessing is performed. In the following chapter 7, we extend the benchmark and use all MIT-BIH ECG time series with less than 250 events, resulting in a set of 25 time series (called the ECG-25 dataset).

6.3.2 Parameterization of the Algorithms

All algorithms compared in this work require a set of parameters, which are – if not mentioned otherwise – fixed for all experiments. For each algorithm, an anomaly threshold can be set, which specifies the algorithm's sensitivity towards anomalies and trades off false detections (false positives) and missed anomalies (false negatives). This threshold is usually set according to the anomaly detection task's requirements – allowing either a higher precision or a higher recall.

LSTM-AD We implemented our proposed algorithm using the Keras framework [30] with a TensorFlow [1] backend. For our LSTM-AD algorithm, both the MLII and V5 signal will be used as inputs of the model, and only the MLII signal is predicted for the specified horizons. The parameters of the algorithm are summarized in Table 6.1. Most of the parameters are related to the stacked LSTM network. The parameters were selected after a few informal pre-experiments, but we did not systematically tune the parameters.

ADVec There are mainly three parameters (described in Section 3.1.1) which have to be provided for Twitter's ADVec algorithm: The first parameter α represents the level of statistical significance with which to accept or reject anomalies. This parameter is used as anomaly threshold. The second parameter $\max_{anoms}$ specifies the maximum number of

Table 6.1: Summary of the the parameters used for the LSTM anomaly detector.

Parameter	value	Parameter	value
H	$(1, 3, \ldots, 47, 49)$	$\mathcal{L}$	MSE
L	3	u	$(64, 64)$
B	2048	W	80
optimizer	ADAM	α_{init}	0.001
d_{in}	2	d_{out}	1 (MLII)

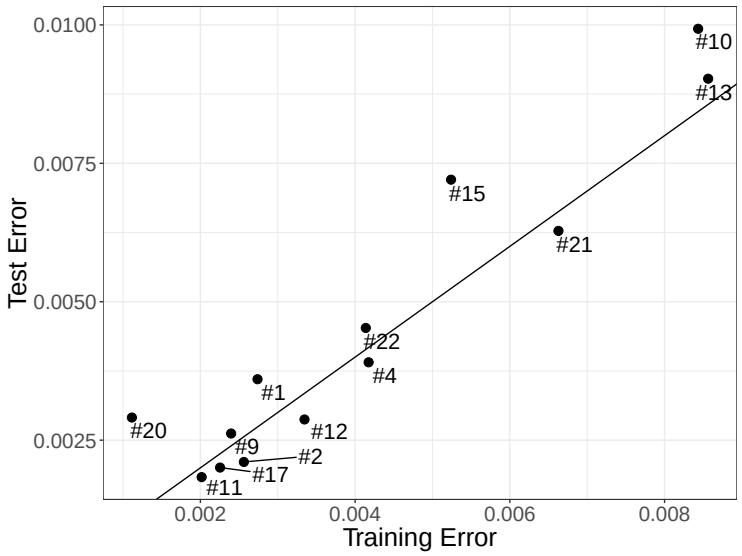

Figure 6.4: Training vs. test errors (MSE) for the individual time series. The median of all such training and test errors is $2.91 \cdot 10^{-3}$ and $3.43 \cdot 10^{-3}$, respectively. The black line depicts the identity line.

anomalies that the algorithm will detect as a percentage of the data. Although we did not tune the parameter extensively, we found the $\max_{anoms} = 0.05$ to deliver the best results.

NuPIC Numenta's anomaly detection algorithm [49] has a large set of parameters which have to be set. As in other experiments, we decided to use the standard parameter settings recommended in [89]. NuPIC outputs an anomaly likelihood for each time series point in the interval [0,1], which is suitably thresholded to control the sensitivity of the algorithm.

6.4 Results & Analysis

Firstly, we confirm with Fig. 6.4 that our LSTM models do not overfit since the training and test set errors are for all time series nearly the same. We note in passing that the time series with a somewhat larger test set error (No. 10, 15, and 20) have – as visual inspection shows

– a test set that varies from the bulk of the training data due to larger non-stationarities in the data set.

As seen in Table 6.2, the Mahalanobis distance is generally a good indicator for separating nominal from anomalous behavior in the heartbeat signals if a suitable threshold is known. For all time series, a recall value of 0.5 or larger can be observed, and with one exception, the F_1-score exceeds the value 0.5. On average, a F_1-score of approximately 0.81 can be achieved for all time series. Note that all FPRs are smaller than $3 \cdot 10^{-5}$.

Table 6.2: Results for the ECG-13 dataset (all ECG time series with less than 50 anomalies; in total 13 time series). For these results, the anomaly threshold is chosen for each time series individually so that the F_1-score is maximized. The row Σ represents the metrics for the sum of TP, FN and FP over all 13 time series.

No.	threshold	TP	FN	FP	Prec	Rec	F_1	FPR $*10^5$	PLR $/10^5$
1	49.31	17	17	14	0.55	0.50	0.52	2.15	0.23
2	71.31	6	0	4	0.60	1.00	0.75	0.62	1.62
4	11.25	1	1	0	1.00	0.50	0.67	0.00	Inf
9	27.12	15	13	3	0.83	0.54	0.65	0.46	1.16
10	14.96	33	7	6	0.85	0.82	0.84	0.92	0.89
11	60.75	1	0	0	1.00	1.00	1.00	0.00	Inf
12	59.18	2	0	0	1.00	1.00	1.00	0.00	Inf
13	116.93	6	0	0	1.00	1.00	1.00	0.00	Inf
15	99.74	5	0	5	0.50	1.00	0.67	0.77	1.30
17	40.02	1	0	0	1.00	1.00	1.00	0.00	Inf
20	75.40	1	0	0	1.00	1.00	1.00	0.00	Inf
21	30.30	1	0	0	1.00	1.00	1.00	0.00	Inf
22	121.42	3	0	7	0.30	1.00	0.46	1.08	0.93
mean	–	7	2	3	0.82	0.87	0.81	0.46	Inf
Σ	–	92	38	39	0.70	0.71	0.70	0.46	1.53

Since the anomaly threshold is used for trading off false-positive and false-negatives (precision and recall), one can vary the threshold in a specific range and collect the results for different values. This is done in Figure 6.5, which also shows the results for different thresholds for ADVec and NuPIC. It has to be noted that ADVec accounts only for fixed-length seasonalities, it is not built for quasi-periodic signals as they occur in ECG readings, so it is quite understandable that it has only low performance here.

In Figure 6.6, two excerpts of time series No. 13 (left) and No. 10 (right) with the detections of our LSTM-AD algorithm, NuPIC and ADVec are exemplarily shown. In both examples, it can be seen that LSTM-AD detects all indicated anomalies, while NuPIC and ADvec only detect two and one anomalies, respectively. Additionally, the other two algorithms produce several false positives.

The importance of our proposed window-based error correction method (Sec. 6.2.4) is illustrated in Figure 6.3 for ECG signal No. 13: If no window-based error correction is applied, the obtained Mahalanobis distance cannot be suitably used to distinguish between

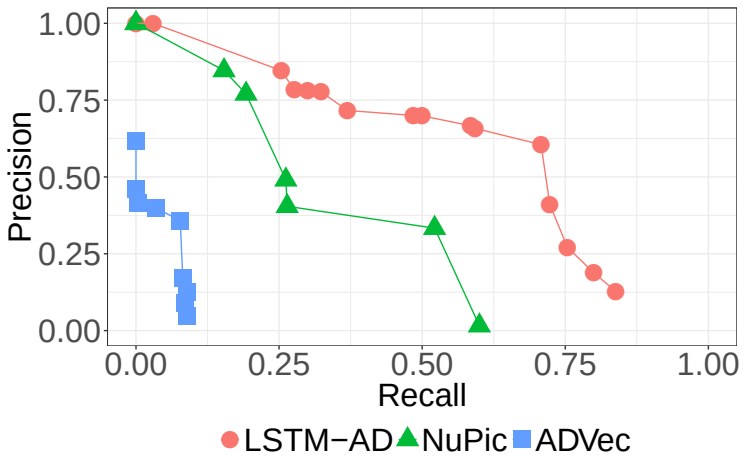

Figure 6.5: Precision-Recall plot for LSTM-AD, NuPIC and ADVec on the ECG-13 data. Precision and recall are computed over the sum of TP, FP, FN of the 13 ECG time series. For all three algorithms, each point is generated by scaling the individual best thresholds up and down by a common factor. For LSTM-AD, the best thresholds are reported in Table 6.2.

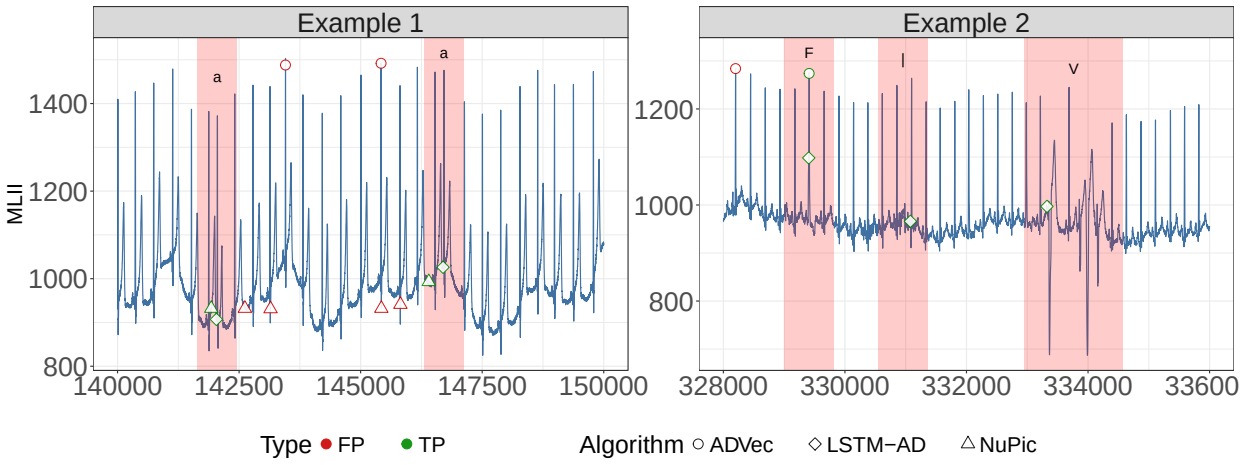

Figure 6.6: Subsets of two example time series taken from the MIT-ECG data with the anomalies detected by the algorithms LSTM-AD, NuPIC, and ADVec. The red rectangles in the plot indicate the true anomaly windows. Green colors indicate True-positives while False-positives are colored red.

Table 6.3: Comparison of the results on the ECG-13 data, with and without the window-based error correction, as described in subsection 6.2.4. The last column F_1(Corr) is copied from Table 6.2. The remaining columns depict the quantities obtained if no window-based error correction is applied (thresholds chosen such that F_1 is maximized).

ECG No.	threshold	TP	FN	FP	Prec	Rec	F_1	F_1(Corr)
1	20.60	12	22	23	0.34	0.35	0.35	0.52
2	5.83	2	4	2	0.50	0.33	0.40	0.75
4	7.03	1	1	0	1.00	0.50	0.67	0.67
9	16.83	13	15	10	0.57	0.46	0.51	0.65
10	16.21	28	12	2	0.93	0.70	0.80	0.84
11	40.60	1	0	0	1.00	1.00	1.00	1.00
12	28.48	1	1	1	0.50	0.50	0.50	1.00
13	93.83	5	1	130	0.04	0.83	0.07	1.00
15	87.97	2	3	5	0.29	0.40	0.33	0.67
17	35.90	1	0	38	0.03	1.00	0.05	1.00
20	25.10	1	0	1	0.50	1.00	0.67	1.00
21	32.31	1	0	0	1.00	1.00	1.00	1.00
22	77.33	2	1	37	0.05	0.67	0.10	0.46
mean	–	5	4	19	0.52	0.67	0.50	0.81
Σ	–	70	60	249	0.22	0.54	0.31	0.70

nominal and anomalous patterns in the displayed ECG data. Only after applying our approach, a better signal-to-noise ratio is established, which perfectly separates the anomalies from the nominal points. For most of the investigated ECG readings, we found that the window-based error correction significantly improves the signal-to-noise ratio in the Mahalanobis distance. Table 6.3 shows: The average F_1-score increases from F_1=0.50 when no window-based error correction is applied to F_1=0.81.

In Table 6.4, various measures are listed for the individual anomaly classes. The anomaly types a, F, and x, can all be detected by LSTM-AD. Also, for the anomaly class V, a high recall can be achieved. However, the two remaining types appear to be hard to detect for our algorithm.

Table 6.4: Various metrics for 5 different anomaly classes. The threshold was tuned individually for each time series by maximizing the F_1-score.

	TP	FN	Prec	Rec	F_1	FPR $*10^5$	PLR $/10^5$
A	23	21	0.65	0.52	0.58	0.14	3.64
V	44	9	0.73	0.83	0.78	0.19	4.36
|	9	8	0.63	0.53	0.58	0.06	8.49
a	6	0	0.79	1.00	0.88	0.02	53.44
F	2	0	0.65	1.00	0.79	0.01	80.16
x	8	0	0.73	1.00	0.85	0.03	29.15

Table 6.5: Same as Table 6.2. However, now the stacked LSTM is trained only with nominal sequences. Hence, training sequences containing anomalies are removed. The results slightly better in the F_1-score in comparison to Table 6.2.

ECG No.	threshold	TP	FN	FP	Prec	Rec	F_1	FPR	TPR/FPR
1	60.60	18	16	2	0.90	0.53	0.67	0.31	1.72
2	87.84	5	1	2	0.71	0.83	0.77	0.31	2.71
4	8.78	1	1	0	1.00	0.50	0.67	0.00	Inf
9	30.52	14	14	0	1.00	0.50	0.67	0.00	Inf
10	18.80	32	8	5	0.86	0.80	0.83	0.77	1.04
11	68.91	1	0	0	1.00	1.00	1.00	0.00	Inf
12	60.03	2	0	0	1.00	1.00	1.00	0.00	Inf
13	92.32	6	0	1	0.86	1.00	0.92	0.15	6.50
17	84.21	1	0	0	1.00	1.00	1.00	0.00	Inf
21	28.57	1	0	0	1.00	1.00	1.00	0.00	Inf
22	123.27	3	0	5	0.38	1.00	0.55	0.77	1.30
15	130.13	4	1	2	0.67	0.80	0.73	0.31	2.60
20	50.11	1	0	0	1.00	1.00	1.00	0.00	Inf
mean	–	6	3	1	0.88	0.84	0.83	0.20	Inf
Σ	–	89	41	17	0.84	0.68	0.75	0.20	3.40

Our method is unsupervised in the sense that no anomaly class labels are needed for training the algorithm. In fact, it is even not necessary that anomalous events are present at all in the training data, i.e., our algorithm can operate as a one-class classifier. We checked this by repeating the experiment leading to Table 6.2, but this time removing all data around anomalies during LSTM training. When using the trained model as an anomaly detector on all data, it worked as accurately as in Table 6.2, the mean F_1-score being now $F_1 = 0.83$. The results for this experiment are listed in Table 6.5.

6.5 Conclusion & Possible Future Work

We have presented a fully unsupervised method to detect anomalies in ECG readings. This method relies on an accurate LSTM predictor to learn the nominal behavior of the ECG for several prediction horizons. For the prediction errors, a mean vector and covariance matrix is estimated. Anomalous events have a high probability of being detected through an unusually high Mahalanobis distance.

We achieve for the ECG readings these high precision, recall, and F_1-values (on average higher than 80%, see Table 6.2), if we tune the final threshold for the Mahalanobis distance such that F_1 is maximized. Admittedly, this last step is not unsupervised since we calculate the confusion matrix based on the true anomaly labels.

We have shown that the window-based error correction is essential to achieve a Mahalanobis distance graph where the anomaly cases clearly stand out (Fig. 6.3 and Table 6.3).

Our LSTM-AD algorithm outperformed two state-of-the-art anomaly detection algorithms (NuPIC and ADVec) on the investigated ECG readings, achieving higher precision and recall over a large range of anomaly thresholds.

We have presented initial results of an unsupervised anomaly detector suitable for ECG readings or other quasi-periodic signals in this work. The results are encouraging, but there is still room for improvement. Possible future works include:

- Addressing the problem of concept-drifts in the ECG readings, e. g. by applying suitable preprocessing steps to reduce the effect of signal quality changes.
- Enrich the model by multi-resolution approaches to span larger prediction horizons on a coarser scale.
- Finding better hyper-parameters for the LSTM model in an automized fashion.

In the next chapter, we will introduce a temporal convolutional network autoencoder (TCN-AE), an architecture based on dilated convolutional layers, and compare TCN-AE to the LSTM-AD algorithm of this chapter on the Mackey-Glass Anomaly Benchmark (MGAB) and the extended ECG-25 benchmark.