Computational Statistics (2019) 34:945–976
https://doi.org/10.1007/s00180-018-0824-9

ORIGINAL PAPER

Time series anomaly detection based on shapelet learning

Laura Beggel[1,2] · Bernhard X. Kausler[1] · Martin Schiegg[1] · Michael Pfeiffer[1] · Bernd Bischl[2]

Received: 7 July 2017 / Accepted: 23 June 2018 / Published online: 16 July 2018

Abstract
We consider the problem of learning to detect anomalous time series from an unlabeled data set, possibly contaminated with anomalies in the training data. This scenario is important for applications in medicine, economics, or industrial quality control, in which labeling is difficult and requires expensive expert knowledge, and anomalous data is difficult to obtain. This article presents a novel method for unsupervised anomaly detection based on the shapelet transformation for time series. Our approach learns representative features that describe the shape of time series stemming from the normal class, and simultaneously learns to accurately detect anomalous time series. An objective function is proposed that encourages learning of a feature representation in which the normal time series lie within a compact hypersphere of the feature space, whereas anomalous observations will lie outside of a decision boundary. This objective is optimized by a block-coordinate descent procedure. Our method can efficiently detect anomalous time series in unseen test data without retraining the model by reusing the learned feature representation. We demonstrate on multiple benchmark data sets that our approach reliably detects anomalous time series, and is more robust than competing methods when the training instances contain anomalous time series.

Keywords Unsupervised learning · Feature learning · Support vector data description · Block-coordinate descent

1 Introduction

Detecting anomalous instances in temporal sequence data is an important problem in domains such as economics (Hyndman et al. 2015b), medicine (Chuah and Fu 2007), astronomy (Rebbapragada et al. 2009), and computer safety and intrusion detection

✉ Laura Beggel
laura.beggel@de.bosch.com

1 Bosch Center for Artificial Intelligence, Robert Bosch GmbH, Robert-Bosch-Campus 1, 71272 Renningen, Germany

2 Department of Statistics, Ludwig-Maximilians-University, Munich, Germany

(Zhong et al. 2007). Consider for example the application in medicine for monitoring electrocardiogram data by a wearable sensor. A person's healthy heartbeat shows more or less the same electrocardiographic measurement pattern. So for each heartbeat, we have a temporal sequence of measurements, resulting in a data set of heartbeat measurements of a single person. Now the patient develops some first signs of arrhythmia which cause deviations from the healthy heartbeats, the anomalies. The anomaly detection model in the sensor detects these anomalous measurements and can raise an alarm or inform the patient's doctor.

Here we address the problem of learning to detect anomalous time series without having access to labels that indicate which instances in the training set are normal and which are not. This is a highly relevant scenario, since anomalies by definition occur very rarely, and are of diverse, unpredictable nature. It is therefore difficult to collect a large enough set of anomalous time series, and it would require domain experts to spot these few anomalies within a large database containing mostly normal instances. Whereas several anomaly detection methods learn models of normal time series under the assumption that all training data is normal (Mahoney and Chan 2005; Salvador and Chan 2005; Rebbapragada et al. 2009), we present a novel method based on the Support Vector Data Description (SVDD) (Tax and Duin 2004) that learns to detect anomalous time series even if the training set is contaminated with occasional anomalies. Simultaneously, our method characterizes what constitutes normal behavior of a time series. Despite the practical relevance of detecting entire time series as anomalies, this research field has attracted moderate attention (Hyndman et al. 2015b) and learning representative features from unlabeled data containing anomalies for testing new data without retraining the model is—to the best of our knowledge—even less examined.

Mining of entire time series may lead to intractable memory and time requirements. It is thus desirable to train models that work on a smaller number of extractable features. A popular approach is to let experts design characteristic features that reveal anomalous instances according to their experience (Hyndman et al. 2015b), but such domain expert knowledge is often not available (Zhong et al. 2007). Since important characteristics of time series are likely contained in short sub-sequences (Forrest et al. 1996), Ye and Keogh (2009) proposed so called *shapelets*, i.e., representative subsequences of time series, which yield state-of-the-art results in classification benchmarks (Bagnall et al. 2017a). Whereas the original approach considered only subsequences that were observed in the training set, Grabocka et al. (2014) extended the framework towards learning optimal shapelets for supervised classification.

In this article, we demonstrate the first application of shapelet learning in the context of unsupervised time series anomaly detection. Shapelets have shown good performance when indeed relevant information is contained in subsequences, but are of limited use in scenarios where statistics or spectral features over the whole time series determine the class. The focus of the present article is thus on time series problems where recurring short temporal patterns are indicative of normality or class membership.

The main contribution of this article is a novel unsupervised method to learn shapelet-based features that are representative of the normal class from a training set of time series contaminated with anomalies and—at the same time—detect anomalous instances in this set of time series. Those learned features can then be efficiently

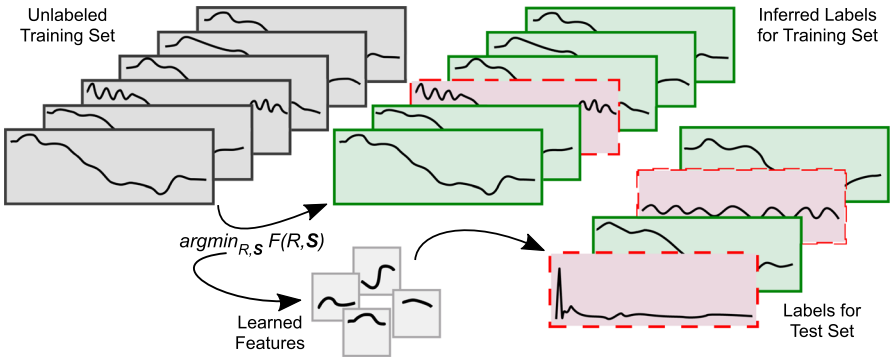

Fig. 1 Unsupervised anomaly detection in time series with shapelet learning. Starting from an unlabeled data set, our method optimizes an objective function $F(\cdot)$ and simultaneously detects anomalies (red) in the training set. The anomalies are detected based on a set of learned shapelet features $\mathbf{S}$, short subsequences that characterize normal (green) time series, and a decision boundary R. The learned features and decision boundary can be used to detect time series on novel test data without retraining the model (color figure online)

used for further time series analysis without retraining the model, and without storing entire training time series. Furthermore, our method only requires a small number of time series for training and still generalizes well. The basic concept of our method is displayed in Fig. 1.

This is the first approach that combines shapelet features for time series, shapelet learning, and unsupervised anomaly detection. There are not many competing methods that can deal with this scenario, hence we additionally propose a baseline approach, which uses extracted shapelets for anomaly detection. In a series of experiments we demonstrate the superior performance of our combined method, in particular when training sets contain anomalies.

The remainder of this article is organized as follows: Sect. 2 starts with an overview of related work on time series anomaly detection and shapelet-based methods for time series classification. The latter is described in more detail in Sect. 3 together with the theory of detecting anomalous observations with SVDD. We motivate how the concept of shapelets is suitable for anomaly detection in Sect. 4, and propose a baseline approach that extracts shapelets. Additionally, we introduce a modification of SVDD which is adapted to the problem of shapelet-based anomaly detection. We propose our novel unsupervised method that in contrast to the proposed baseline method from Sect. 4 learns shapelet-based features and simultaneously detects anomalous instances in Sect. 5. We explain our experimental setup in Sect. 6 before we evaluate our proposed method in Sect. 7, and discuss our results and the applicability of shapelets for anomaly detection in Sect. 8. We conclude our article with general remarks and recommendations on when to use our proposed method in Sect. 9.

2 Related work

Anomalies are in general defined as deviations from normal behavior (Chandola 2009; Chandola et al. 2009), assuming that normal instances originate from some common

data generating process. The main assumption of anomaly detection is therefore that the data set is mostly made up of normal instances, which share common characteristics, and only a small percentage of instances will be anomalous. Since anomalies are rare and there may be many diverse types, it cannot be assumed that the space of all possible anomalies can be exhaustively sampled. In many real-world applications, it cannot be assumed either that it is a priori known which examples are anomalous. The setup of anomaly detection as an unsupervised and highly unbalanced problem is different from typical supervised classification tasks. A detailed overview of anomaly detection can be found in Chandola (2009).

Detecting anomalies in time series is a particularly challenging topic, because abnormal behavior may manifest itself in any short sub-sequence, as well as in longer trends (Forrest et al. 1996). Two major types of anomaly detection problems for time series can be distinguished: On the one hand, the task can be to identify entire time series as anomalous in relation to a set of other time series (Mahoney and Chan 2005; Salvador and Chan 2005; Hyndman et al. 2015b). On the other hand, the task can be to identify short parts or even single points within a time series as anomalous (Ma and Perkins 2003; Keogh et al. 2005; Fu et al. 2006; Chatfield 2016). We focus here on the first scenario of detecting entire time series as anomalous, which has numerous applications and has been extensively studied, e.g., for light curves in astronomy (Rebbapragada et al. 2009), server monitoring (Hyndman et al. 2015b), or identification of failures in space shuttle valves (Salvador and Chan 2005).

The typical approach for detecting entire time series as anomalies is to first learn a representative model for normality, and then identify anomalies by their deviation from this normal model. Whereas our approach explicitly allows occasional anomalies in the training data, previous methods rely on a training set consisting entirely of normal instances. Salvador and Chan (2005) segment the normal training data into characteristic states and learn logical rules for state transitions in the form of a deterministic finite automaton. At test time, time series are classified as normal if they lead to a valid state sequence in the automaton. The automaton learns to produce valid state sequences for every time series in the training set, hence anomalies in the training data can lead to misclassifications at test time. Mahoney and Chan (2005) map time series into a 3-dimensional feature space, and learn sequences of minimal enclosing bounding boxes that contain the trajectories of every instance of the training set. Anomalous instances in the training data would result in too large bounding boxes, and leads to misclassifications at test time. Additionally, clustering approaches based on similarity measures between the time series or subsequences can be used for anomaly detection (Protopapas et al. 2006; Wei et al. 2006; Rebbapragada et al. 2009). The basic principle is that anomalies will have large distance to any cluster formed from instances in the training set. Anomalies in the training set may result in a cluster containing those observations, which again will lead to misclassifications.

The approach closest to ours is the recent work of Hyndman et al. (2015b), which can learn from an unlabeled set of time series containing anomalies. Their method extracts basic representative features such as mean and trend, and additional domain specific features such as the number of zero-crossings. The features are then projected onto the first two principal components, and anomalies are detected either by estimating the density in the PCA space (Hyndman 1996), or by learning an enclosing region of

the normal data points with the α-hull method (Pateiro-López and Rodríguez-Casal 2010).

Our method is based on kernel methods for anomaly detection, such as One-class Support Vector Machines (Schölkopf et al. 2000) and SVDD (Tax and Duin 2004), which have been used successfully in a broad range of applications. Since those methods are not directly applicable to time series data without first extracting features, Ma and Perkins (2003) apply the method to time series data by using a time-delay embedding process to generate a feature vector representation for each point in a time series. In the resulting feature space a one-class SVM is used to detect anomalous observations within the series.

To detect anomalous time series, our method utilizes shapelets to create a representative set for normal characteristics. The concept of shapelets to characterize time series was first proposed by Ye and Keogh (2009). They considered a supervised scenario and defined shapelets as time series primitives that are maximally representative of a class. Shapelets are found by first extracting all potential subsequences and selecting a representative subset based on the information gain of each candidate shapelet for classification in a decision tree. Lines et al. (2012) defined the shapelet transform of time series by computing the minimal distance between the time series and each of the k shapelets. The resulting k-dimensional feature vectors can then be fed to standard classifiers. Grabocka et al. (2014) proposed the method of shapelet learning for linear classification of time series instances. Their objective is to learn shapelets that optimize the linear separability of the time series data after the shapelet transform. Shapelets learned in this way need not be present in the original data. Using shapelet learning for unsupervised anomaly detection is a novel concept that we introduce in this article.

3 Preliminaries

In the following, $T_i, i = 1, \ldots, N$, denote N univariate time series with feature representation $\Phi(T_i; \psi) = \mathbf{x}_i \in \mathbb{R}^K$, where K is the number of features, and ψ the transformation parameter vector. Assuming that all time series are of length Q, they can be combined into one matrix $\mathbf{T} = (T_1^\top, \ldots, T_N^\top)$, $\mathbf{T} \in \mathbb{R}^{N \times Q}$. For evaluation we also define a class label $y_i \in \{0, 1\}$ for an entire time series where $y_i = 0$ denotes the normal class and $y_i = 1$ indicates an anomaly. However, during training we assume that the true class label is unknown. The result of the anomaly detection learning procedure is a linear prediction model $\hat{y}_i = \mathbf{w}^\top \mathbf{x}_i + w_0$, $\mathbf{w} \in \mathbb{R}^K$, $w_0 \in \mathbb{R}$.

3.1 One-class support vector data description

Our anomaly detection framework is based on SVDD (Tax and Duin 2004). The data points $\mathbf{x}_i, i = 1, \ldots, N$, are assumed to lie within a single high density region that can be circumscribed—in a first approximation—by a minimal-volume hypersphere with radius R and center $\mathbf{a}$, see Fig. 2a. Finding this sphere is equivalent to solving the optimization problem

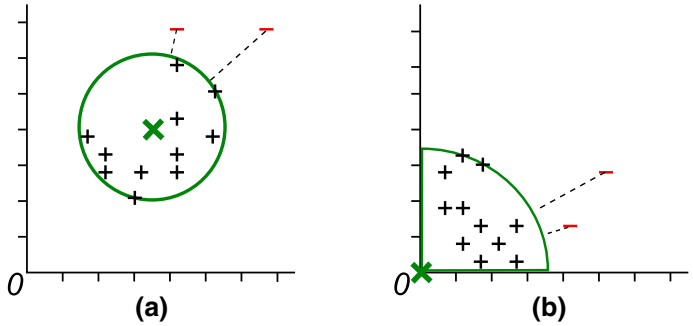

Fig. 2 Anomaly detection with **a** the standard SVDD (Tax and Duin 2004) and **b** our modified SVDD*. A black '+' indicates instances from the normal class, a red '−' indicates anomalies. The green cross is the center of the enclosing hypersphere that represents the anomaly boundary. In **b**, we fix the center of the hypersphere to the origin of the feature space to account for the shapelet-based feature values (color figure online)

$$\min_{R,\mathbf{a},\boldsymbol{\xi}} R^2 + C \cdot \sum_{i=1}^{N} \xi_i,$$
$$s.t.\ \|\mathbf{x}_i - \mathbf{a}\|^2 \leq R^2 + \xi_i,\ \xi_i \geq 0,\ \forall i = 1, \ldots, N \tag{1}$$

with slack variables $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_N)$. By introducing $\boldsymbol{\xi}$, we allow data points to lie outside the hypersphere. These points are however penalized linearly in the objective to keep the amount of outlying points low. This encourages a trade-off between the volume of the hypersphere and the number of data points that lie outside. A kernel function is typically applied to transform the feature space. After solving (1), a new data point $\mathbf{x}_{test}$ can be easily tested, and is detected as anomalous if $\|\mathbf{x}_{test} - \mathbf{a}\|^2 > R^2$.

3.2 Shapelet transformation

Shapelets $S_1, \ldots, S_K$ are defined as subsequences of fixed length L of a time series T_i. In Ye and Keogh (2009) the set of shapelets is chosen so as to optimize prediction of the discriminative class y_i.

The shapelet transformation $\Phi(T_i; \mathbf{S})$ as defined by Lines et al. (2012) depends on the chosen set of shapelets $\mathbf{S} = \{S_1, \ldots, S_K\}$. For a given shapelet S_k of length L, its distance $D_{i,k,j}$ to a time series subsequence $(T_{i,j}, \ldots, T_{i,(j+L-1)})$ with initial time point j (and length L) is defined as

$$D_{i,k,j} = \frac{1}{L} \sum_{l=1}^{L} (T_{i,(j+l-1)} - S_{k,l})^2.$$

The distance between shapelet S_k and an entire time series T_i is then defined as

$$M_{i,k} \coloneqq \min_{j=1,\ldots,J} D_{i,k,j}, \tag{2}$$

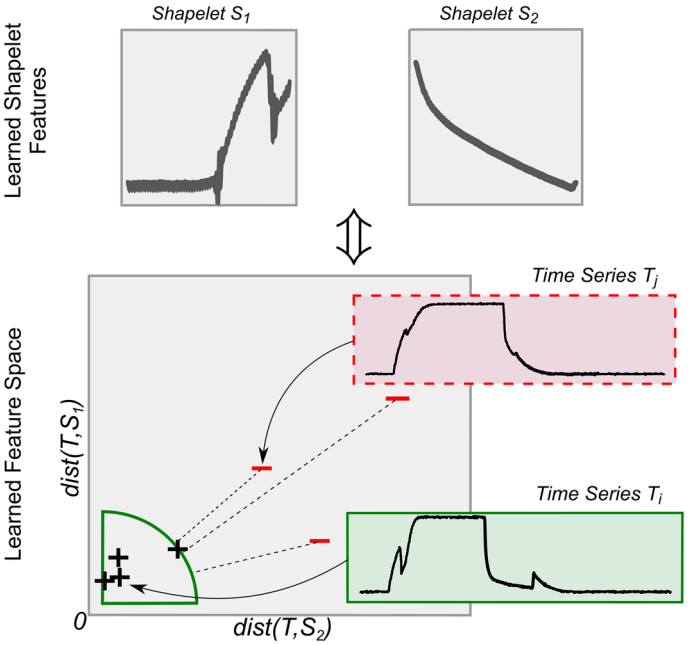

Fig. 3 Shapelets S_1 and S_2 learned from the time series data set provided by Ferrell and Santuro (2015). Each time series is transformed into a vector in a 2-dimensional feature space via $\Phi(T; S_1, S_2)$. The two time series T_i and T_j show the normal and the anomalous class, respectively. Note that we set $K = 2$, i.e., the number of shapelets, only for illustrative reasons

where J is the number of all subsequences of length L within T_i. The K-dimensional feature representation of time series T_i is then given as

$$\Phi(T_i; \mathbf{S}) \coloneqq \Phi(T_i; S_1, \ldots, S_K) = (M_{i,1}, \ldots, M_{i,K})^\top. \tag{3}$$

An example of the shapelet transformation can be seen in Fig. 3.

In their original formulation, shapelets are always subsequences of time series contained in the training set. Grabocka et al. (2014) extend this concept to learn maximally representative shapelets for classification. Learned shapelets need not necessarily be present as exact duplicates in the training data. For training, a regularized empirical risk formulation with logarithmic loss is used, i.e.,

$$\min_{\mathbf{S},\mathbf{w}} F(\mathbf{S}, \mathbf{w}) = \min \sum_{i=1}^{N} \mathscr{L}(y_i, \hat{y}_i) + \lambda_{\mathbf{w}} \|\mathbf{w}\|^2, \tag{4}$$

where $\mathscr{L}(y_i, \hat{y}_i) = -y_i \cdot \ln \sigma(\hat{y}_i) - (1 - y_i) \cdot \ln(1 - \sigma(\hat{y}_i))$ with $\sigma(y_i) = (1 + \exp(-y_i))^{-1}$, and linear prediction model $\hat{y}_i = \mathbf{w}^\top \mathbf{x}_i + w_0$ with shapelet transformation

$$\mathbf{x}_i = \Phi(T_i; S_1, \ldots, S_K) = (M_{i,1}, \ldots, M_{i,K})^\top.$$

For minimizing this empirical loss, the gradients w.r.t. the shapelets $\mathbf{S}$ need to be calculated which implies (by the chain rule) to differentiate $M_{i,k}$ w.r.t. $D_{i,j,k}$. Since the minimum operator within $M_{i,k}$ is not differentiable, Grabocka et al. (2014) propose to apply the smooth-minimum approximation for gradient calculation. Formally,

$$\hat{M}_{i,k} = \frac{\sum_{j=1}^{J} D_{i,k,j} \mathrm{e}^{\mu D_{i,k,j}}}{\sum_{j'=1}^{J} \mathrm{e}^{\mu D_{i,k,j'}}}, \quad \mu < 0 \tag{5}$$

which is the smoothed minimum function of the distance between one shapelet S_k and one time series T_i. The distance function is created by sliding the shapelet over the time series and measuring the L2-distance for each window, yielding the windows starting at different time points j with smaller distances higher values. The parameter μ defines how smooth the function is, i.e., how many peaks the function has. The shapelets are initialized for the learning procedure with the k-means centroids of all potential shapelets. For a more detailed explanation, see (Grabocka et al. 2014).

4 Anomaly detection with shapelets

This section motivates why and how the concept of shapelets is suitable for anomaly detection. The main components are the use of shapelet-based features for anomaly detection, a modified version of SVDD, and a baseline algorithm for detecting anomalies, which uses extracted shapelets.

4.1 Shapelet-based features for anomaly detection

The underlying assumption for anomaly detection is that the normal time series are generated by the same unknown data generating process. Hence, they show similar temporal characteristics, i.e., the information is available in the shape of the time series (Esling and Agon 2012). Here we have chosen shapelets as a natural fit to represent features of time series suitable for anomaly detection. The basic idea is to find a set of shapelets that can reliably be detected in normal time series, but at least some of them do not match in the anomalies. Given a set of shapelets $\mathbf{S} = \{S_1, \ldots, S_K\}$, we use the shapelet transformation from Eq. (3) to obtain the feature representation of a time series T_i as $\mathbf{x}_i = \Phi(T_i; \mathbf{S})$. Since $\Phi(T_i; \mathbf{S})$ computes distances between the best match of the shapelet and the time series, it is clear that all feature vectors $\mathbf{x}_i$ contain only non-negative values. Furthermore, feature values will be small if there is a good match to the corresponding shapelet in the time series.

This implies that if the shapelets $\mathbf{S}$ are representative of the normal class, then all feature vectors for normal time series should lie close to the origin, and form a compact hypersphere with a small radius. Thus, finding shapelets that yield small values for normal time series is equivalent to finding shapelets that are representative for normal data.

4.2 Modified one-class SVDD

We present a modification of the original SVDD from Sect. 3.1, which we denote as SVDD*, and which is adapted to the problem of shapelet-based anomaly detection. In this modification, the center $\mathbf{a}$ of the hypersphere is fixed to the origin of the feature space, i.e., $\mathbf{a} = \mathbf{0}$. Consequently, only the radius R and the slack variables $\boldsymbol{\xi}$ remain as optimization parameters.

Definition 1 The SVDD* constrained optimization problem is

$$\begin{aligned} &\min_{R,\boldsymbol{\xi}} R^2 + C \cdot \sum_{i=1}^{N} \xi_i, \\ &s.t.\ \|\mathbf{x}_i\|^2 \leq R^2 + \xi_i, \quad \xi_i \geq 0, \quad \forall i = 1, \ldots, N \end{aligned} \tag{6}$$

with data points $\mathbf{x}_i \in \mathbb{R}^K$, $i = 1, \ldots, N$, the radius of the hypersphere R, the regularization parameter C, and slack variables $\boldsymbol{\xi} = \xi_1, \ldots, \xi_N$, respectively.

Similar to SVDD, we require the data points to lie within a single high-density region, in our case close to the origin. We use the linear kernel, i.e., the dot product of two vectors in $\mathbb{R}^K$. Without loss of generality, we assume non-negative data points. This simplifies the optimization problem, and is guaranteed when using the shapelet transform. The idea of this modification is depicted in Fig. 2b.

4.3 Shapelet extraction for anomaly detection as a baseline method

Based on the previously introduced formalism, we propose a simple baseline approach for detecting anomalous instances in a set of unlabeled time series contaminated with anomalies. This method directly extracts shapelets from the original training data set by searching through all subsequences present in the data set, and extracting those that are most representative of the normal class. The anomaly detector is then learned afterwards with SVDD* using the extracted shapelets. The proposed baseline shapelet extraction algorithm is outlined in Algorithm 1.

First, all P subsequences of length L from the time series data base $\mathbf{T} = (T_1^\top, \ldots, T_N^\top)$ are extracted as potential shapelets S_p, $p = 1, \ldots, P$. Next the overall distances $M_p = \sum_i M_{i,p}$ of each potential shapelet S_p to the time series T_i, $i = 1, \ldots, N$, are calculated and sorted, resulting in $M_{(1)} \leq M_{(2)} \leq \cdots \leq M_{(P)}$. The subsequences S_p can be ordered according to their overall distance value M_p, i.e., $S_{(1)}, \ldots, S_{(P)}$. The subsequence $S_{(1)}$ corresponding to the minimal overall distance $M_{(1)}$ is extracted as the first shapelet.

In the next step, a distance boundary is computed based on the inter-shapelet distance $\mathrm{ISD}(S_{(1)}, p) = \sum_{l=1}^{L} (S_{(1),l} - S_{p,l})^2$ of all remaining subsequences S_p, $p = 1, \ldots, P-1$, to the shapelet $S_{(1)}$ to avoid redundancy. All subsequences with distances within this boundary are deleted. The procedure continues on the remaining candidates, and is repeated until K shapelets $S_1, \ldots, S_K$ are extracted.

Algorithm 1 Shapelet Extraction for Anomaly Detection

procedure EXTRACTSHAPELETS(Time series set $\mathbf{T} = (T_1^\top, \ldots, T_N^\top)$, number of shapelets K)
  extract $S_1, \ldots, S_P$ from $\mathbf{T}$
  calculate $M_p = \sum_i^N M_{i,p} \forall p \in 1, \ldots, P$
  order $M_1, \ldots, M_P$ resulting in $M_{(1)} \leq M_{(2)} \leq \ldots \leq M_{(P)}$
  sort $S_1, \ldots, S_P$ according to corresponding M_p ordering, resulting in $S_{(1)}, \ldots, S_{(P)}$
  repeat
    extract $S_{(1)}$ as shapelet
    calculate $\text{ISD}(S_{(1)}, p) = \sum_{l=1}^{L} (S_{(1),l} - S_{p,l})^2$
    set similarity boundary:
      $0.1 \cdot \text{median}(\text{ISD}(\text{S}(1), \text{m}))$
    delete S_p with
      $\text{ISD}(S_{(1)}, p) < \text{boundary}$
  until K shapelets extracted
  return K shapelets
end procedure

At test time, a new time series Z with feature representation $\mathbf{z} = \Phi(Z; S_1, \ldots, S_K)$ will be classified as normal if $\|\mathbf{z}\|^2 \leq R^2$, otherwise it is detected as an anomaly.

5 Joint shapelet learning and anomaly detection

We now turn to the problem of jointly learning a representative set of shapelets $\mathbf{S}$, while using $\mathbf{S}$ to solve the unsupervised anomaly detection problem. Learning the shapelets no longer restricts the candidate shapelets to those that are actually present in the data set, but finds shapes that are optimally suited to characterize normal time series.

5.1 Anomaly detection with shapelet-based feature learning

This section introduces our new method called *Anomaly Detection algorithm with Shapelet-based Feature Learning* (ADSL), c.f. Fig. 3. ADSL detects anomalous instances in a set of unlabeled time series contaminated with anomalies and—at the same time—learns features that are highly representative of the normal class. The learned features are shapelet-based and can be efficiently used for analyzing further test data. In contrast to the previously introduced shapelet extraction procedure for anomaly detection outlined in Algorithm 1, our joint learning approach learns representative subsequences that may not necessarily be present in any training instance. For a given set of shapelets, we can solve the anomaly detection problem by solving the optimization problem defined in (6). However, the shapelets in $\mathbf{S}$ need to be optimized to become representative of the normal class leading to a shapelet transformation as visualized in Fig. 4.

In Sect. 4.1, we noted that the shapelet features of normal time series are expected to be small if the shapelets are representative of the normal class. By regularizing the feature learning with $\ell(\mathbf{x}_i) = \|\mathbf{x}_i\|^2 = \sum_{k=1}^{K} M_{i,k}^2$, we can encourage that the (normal) data points will be pulled towards the origin during learning. The regularization stabilizes the convergence of the learned model, and guarantees that shapelets are learned

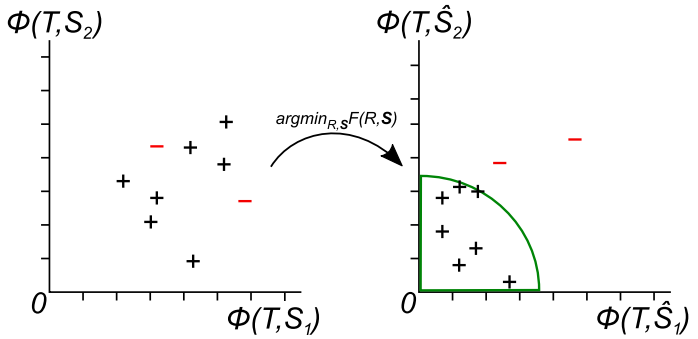

Fig. 4 Anomaly detection with learned shapelets: On the left side, the initial features based on the initial shapelets $S_k, k = 1, 2$, are displayed. Both, the normal (+) and anomalous (−) class are scattered in the feature space. On the right, the transformed setting after learning the shapelet transformation by optimizing the objective function $F(R, \mathbf{S})$ are visualized. The learned normality boundary is shown in green (color figure online)

that are representative of the normal class. This gives rise to our key contribution, the ADSL algorithm for learning representative shapelets for anomaly detection:

Definition 2 The ADSL model for shapelet-based anomaly detection is given as the solution of the constrained optimization problem

$$\begin{aligned} \arg\min_{R,\mathbf{S},\boldsymbol{\xi}} F(R,\mathbf{S}) = \; & R^2 + C \cdot \sum_{i=1}^{N} \xi_i + \sum_{i=1}^{N} \ell(\mathbf{x}_i), \\ s.t. \; & \|\mathbf{x}_i\|^2 \leq R^2 + \xi_i, \quad \xi_i \geq 0, \quad \forall i = 1, \ldots, N \end{aligned} \tag{7}$$

where $\mathbf{x}_i = \Phi(T_i; S_1, \ldots, S_K) = (M_{i,1}, \ldots, M_{i,K})^\top$ and $\ell(\mathbf{x}_i) = \|\mathbf{x}_i\|^2 = \sum_{k=1}^{K} M_{i,k}^2$, the radius of the decision boundary R, regularization parameter C, and slack variables ξ.

Similar to the heuristic in Schölkopf et al. (2000), the regularization parameter C is set in relation to the assumed anomaly rate α as $C = \frac{1}{\alpha \cdot N}$. We thus force the algorithm to assume $\alpha \cdot N$ observations being anomalous. That is, we choose the hyperparameter C using prior information about the amount of anomalies we expect in the training data. In the experiments in Sect. 6.2, we will show that a conservative setting of C that assumes an anomaly rate of $\alpha = 5\%$ can reliably solve anomaly detection problems where the true anomaly rate lies between 0 and 5%.

The procedure for detecting an anomalous time series instance remains the same as for the extraction approach: at test time, a new time series Z with feature representation $\mathbf{z} = \Phi(Z; \mathbf{S})$ will be classified as an anomaly if $\|\mathbf{z}\|^2 > R^2$.

5.2 Optimization

To solve the optimization problem defined in Eq. (7), we employ a block coordinate optimization algorithm (Boyd and Vandenberghe 2004). The algorithm utilizes (sub-)

Algorithm 2 Learning Shapelet-Features for Anomaly Detection

procedure LEARNINGSHAPELETS(Shapelets S, step rate η)
  Init: $f^{best} \leftarrow \infty$, $cnt \leftarrow 1$
  repeat
    for $i = 1, \ldots, N$ **do**
      for $k = 1, \ldots, K; l = 1, \ldots, L$ **do**
        $u_{i,k,l} \leftarrow \frac{\partial F_i}{\partial S_{k,l}}$ ▷ Sec 5.2
      end for
      $\bar{u}_{k,l} \leftarrow \frac{1}{N} \sum_i u_{i,k,l}$
    end for
    $S^{new} \leftarrow S - \eta \cdot \bar{u}$
    $f^{new} \leftarrow F(S^{new})$ ▷ Equation (8)
    $update(f^{best}, S^{best})$
    $S \leftarrow S^{new}$
    $cnt = cnt + 1$
  until $cnt = 10$
  $S \leftarrow S^{best}$
  return S
end procedure

gradients and updates at any time just one or a few blocks of variables while leaving the other blocks unchanged. In our case, there are two blocks with alternating updates: either we update the radius R of the decision boundary, or the set of representative shapelets $\mathbf{S}$.

Anomaly Detection Keeping the shapelets $S_1, \ldots, S_K$ and thus the feature representation $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)^\top$ fixed, we learn the radius of the decision boundary R for anomaly detection. This reduces the optimization problem in Eq. (7) to the SVDD* anomaly detection formulation in Eq. (6). Applying Lagrangian multipliers for constrained optimization as in Tax and Duin (2004), this yields the linear problem

$$\arg\max_{\beta} L(\mathbf{x}; \beta) = \sum_i \beta_i \left(\mathbf{x}_i^\top \cdot \mathbf{x}_i \right),$$
$$s.t. \sum_i \beta_i = 1, \quad 0 \leq \beta_i \leq C, \ \forall i = 1, \ldots, N$$

which can be optimized with standard methods for linear programming.

Learning Shapelet Features Now, keeping the radius R of the decision boundary fixed, we update the set of shapelets $\mathbf{S}$ with the goal of making it representative of the normal class. Importantly, altering $\mathbf{S}$ will change all feature representations $\mathbf{x}_i$ of all time series T_i in Eq. (3). Since Eq. (7) now only depends on the shapelets $\mathbf{S}$ by fixing the radius R, we can rewrite it as a regularized empirical risk minimization problem yielding the shapelet learning problem

$$\arg\min_{\mathbf{S}} F(R, \mathbf{S}) = R^2 + C \cdot \sum_{i=1}^{N} \max\{0, \|\mathbf{x}_i\|^2 - R^2\} + \sum_{i=1}^{N} \ell(\mathbf{x}_i), \tag{8}$$

where $\ell(\mathbf{x}_i) = \|\mathbf{x}_i\|^2 = \sum_{k=1}^{K} M_{i,k}^2$ as motivated in Sect. 5.1.

The shapelet learning problem is non-convex and non-differentiable (Grabocka et al. 2014). However, we can calculate the subgradients by using a suitable approximation of the non-differentiable parts (Grabocka et al. 2014; Boyd et al. 2003):

A subgradient g of a function f (that is defined for $\mathbf{dom}(f)$) at point γ fulfills $f(\gamma) \geq f(\eta) + g^\top(\gamma - \eta), \forall \gamma \in \mathbf{dom}(f)$, i.e., the subgradient is a lower bound to the function f at γ (Boyd et al. 2003). Here, we apply subgradients on the hinge-loss $\max\{0, \|\mathbf{x}_i\|^2 - R^2\}$, which is not differentiable for $\|\mathbf{x}_i\|^2 = R^2$. Additionally, we need to differentiate $M_{i,k}$ containing the minimum operator to calculate the subgradients. We follow Grabocka et al. (2014) and approximate $M_{i,k}$ with the smooth-minimum $\hat{M}_{i,k}$ in Eq. (2) that is differentiable.

To learn the optimal shapelets for detecting anomalies, we need to minimize the learning function in Eq. (8) w.r.t. the shapelets $\mathbf{S} = \{S_1, \ldots, S_K\}$, which define the shapelet transformation $\mathbf{x}_i = \Phi(T_i; S_1, \ldots, S_K) = (M_{i,1}, \ldots, M_{i,K})^\top$. We calculate the subgradients for the set of shapelets $\mathbf{S}$ independently for each time series T_i and use the average shapelet gradient over all time series T_i for updating in the learning iteration.

The individual learning function of Eq. (8) for each time series T_i is

$$F_i(R, \mathbf{S}) = \frac{R^2}{N} + C \cdot \max\{0, \|\mathbf{x}_i\|^2 - R^2\} + \ell(\mathbf{x}_i). \tag{9}$$

The derivative for each shapelet $S_k, k = 1, \ldots, K$, at point $l = 1, \ldots, L$ is

$$\frac{\partial F_i}{\partial S_{k,l}} = C \cdot \frac{\partial}{\partial S_{k,l}} \max\{0, \|\mathbf{x}_i\|^2 - R^2\} + \frac{\partial}{\partial S_{k,l}} \|\mathbf{x}_i\|^2.$$

With the smooth-minimum approximation $\hat{M}_{i,k}$ from Eq. (2), we derive the approximated subgradients for the first summand as

$$C \cdot \frac{\partial}{\partial S_{k,l}} \max\{0, \|\mathbf{x}_i\|^2 - R^2\} \approx \delta\{\|\mathbf{x}_i\|^2 \geq R^2\} \cdot 2C\hat{M}_{i,k} \cdot \cdot \left(1 - \delta(\|\mathbf{x}_i\|^2 = R^2) \cdot \mathscr{U}[0, 1]\right) \sum_j \frac{\partial \hat{M}_{i,k}}{\partial D_{i,k,j}} \frac{\partial D_{i,k,j}}{\partial S_{k,l}},$$

and the second part of the derivative can be calculated as

$$\frac{\partial}{\partial S_{k,l}} \|\mathbf{x}_i\|^2 \approx 2 \frac{\partial}{\partial S_{k,l}} \hat{M}_{i,k} = 2\hat{M}_{i,k} \sum_j \frac{\partial \hat{M}_{i,k}}{\partial D_{i,k,j}} \frac{\partial D_{i,k,j}}{\partial S_{k,l}}.$$

The local optimum found by the algorithm depends on the initialization of the shapelets. To overcome this multimodality problem, a standard approach is to repeat the analysis with varying initialization which in our evaluation is carried out through statistical replication. To make the shapelets more robust, we repeat the shapelet update step 10 times in each shapelet learning iteration and keep track of the best solution due to the subgradients. The pseudocode for our method is described in Algorithm 2.

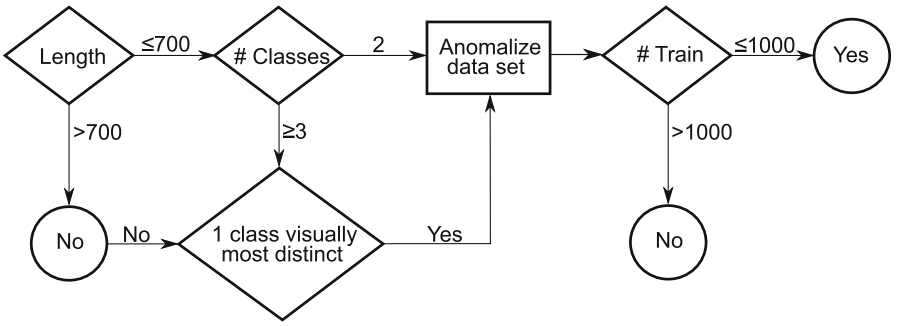

Fig. 5 Decision criteria to select a classification benchmark being suitable for transformation into an anomaly detection problem. This process is based on data set characteristics such as length of time series, number of classes, and number of normal training observations, and does not require domain knowledge

The iterative learning procedure is carried out until the relative change in the objective function over two iterations is smaller than 0.0001.

6 Experiments

Our proposed model for anomaly detection is evaluated on a representative set of publicly available benchmark data sets.[1]

6.1 Data set generation, performance measures and experimental setup

There is a lack of publicly available benchmark data sets for anomaly detection (Keogh and Kasetty 2003; Emmott et al. 2013). We thus use data sets from the UCR Time Series Classification Archive (Bagnall et al. 2017b), that were originally intended for evaluating time series classification, and transform them into anomaly detection problems. However, this transformation is not straightforward: Transforming a classification problem into an anomaly detection problem requires exhaustive data and domain knowledge, e.g., which class is defined as normal and how is this class characterized, that we do not have. We thus decide if a data problem is suitable for evaluating our anomaly detection method ADSL based on several decision criteria, which is also displayed in Fig. 5: first, time series longer than 700 measurements per observation are excluded because of processing capacities. To select a multiclass classification problem as suitable for anomaly detection, one class needs to be (without domain knowledge) visually distinguishable as normal class. The selected multiclass problems and all binary classification problems are then transformed into anomaly detection data sets following the procedure described in the next paragraph. To overcome further limitations in memory capacity, we exclude data sets with more than 1000 training time

[1] The complete code for our evaluation is available on request from the authors. Additionally, we provide our method as an `R` package which will be publicly available. Currently, access to the package can also be requested from the authors.

series. Following this approach, 28 time series data sets from several problem domains are left for evaluating our proposed methods.

To create data sets for anomaly detection from classification benchmarks, we loosely follow the approach suggested by Emmott et al. (2013): first, the original split between training and test data is lifted. For binary classification data, the most frequent class is defined as the normal case (Emmott et al. 2013); if the number of observations is identical, the class occurring first is chosen as the normal case. For multiclass data, the class that is visually most distinct from the remaining classes is defined as normal. This yields a variety of anomaly types. In order to generate independent training and test sets for the transformed data, 80% of the normal time series are randomly selected into the training set, and the remaining 20% become the test set. The anomalous training instances are randomly selected from the available examples. Their number is proportional to the number of normal instances, and the fraction α of anomalous instances in the training set is fixed to either 0.1, 1, or 5%. This simulates scenarios in which the training set contains no anomalies, a realistic proportion, or an extreme proportion of anomalies, respectively. In this way studying of how much the performance of anomaly detection algorithms is affected by contamination of the training set with anomalies is allowed for. Since we want to use all observations for our evaluation, the remaining anomalous time series are then added to the test set. Because of this, the proportion of anomalies in the test data is rather high. However, this does not influence the analysis of the performance itself, if a suitable performance measure is chosen that is insensitive for imbalanced classes (Brodersen et al. 2010), as described below. The large variety of anomalies in the test set additionally requires our anomaly detector to identify various types of anomalies.

Each algorithm is evaluated on ten random train–test splits created from each data set, and the median performance over the ten runs is reported for comparison. A detailed description of the data sets is given in Table 1.

Performance Measure In order to quantify the performance of different anomaly detection methods we measure the sensitivity ($\frac{TP}{TP+FN}$) and specificity ($\frac{TN}{TN+FP}$), where a *positive event* is an anomaly, and a *negative event* is a normal time series. High sensitivity indicates that all anomalies are reliably detected, whereas high specificity indicates that there are not many false alarms. Furthermore, we compute the balanced accuracy (BA), defined as $\frac{\text{Sensitivity}+\text{Specificity}}{2} \in [0, 1]$. The BA compensates for imbalanced data, because in order to reach a high value, the method has to perform well on both classes (Brodersen et al. 2010).

Comparison The performance of each method (ADSL and 2 baselines) is evaluated separately for each of the three anomaly rates $\alpha \in \{0.1\%, 1\%, 5\%\}$ in order to analyze their ability to cope with anomalies in the training set. The Wilcoxon signed-rank test (Gravetter and Wallnau 2017) is used to test for significant improvement in each anomaly rate setting. For this, a pair-wise test (ADSL vs. Hyndman et al. (2015b) and ADSL vs. Extract) on the vector of observations of balanced accuracy values over the 28 data sets is performed, using a significance level of 0.05.

Table 1 Description of the 28 data sets used for evaluation

Data sets	# TS	# CL (normal class)	# N train/test	# A: $\alpha = 5\%$ train/test	# A: $\alpha = 1\%$ train/test	# A: $\alpha = 0.1\%$ train/test
Adiac	781	37 (9)	16/4	1/760	0/761	0/761
ArrowHead	211	3 (2)	52/13	3/143	1/145	0/146
Beef	60	5 (1)	10/2	0/48	0/48	0/48
BeetleFly	40	2 (1)	16/4	1/19	0/20	0/20
BirdChicken	40	2 (1)	16/4	1/19	0/20	0/20
CBF	930	3 (2)	248/62	12/608	2/618	0/620
Chlorine Concentration	4307	3 (1)	800/200	40/3267	8/3299	1/3306
Coffee	56	2 (0)	23/6	1/26	0/27	0/27
ECG200	200	2 (1)	54/13	3/130	1/132	0/133
ECGFiveDays	884	2 (1)	354/88	18/424	4/438	0/442
FaceFour	112	4 (3)	23/6	1/82	0/83	0/83
GunPoint	200	2 (1)	80/20	4/96	1/99	0/100
Ham	214	2 (1)	82/21	4/107	1/110	0/111
Herring	128	2 (1)	62/15	3/48	1/50	0/51
Lightning2	121	2 (1)	38/10	2/71	0/73	0/73
Lightning7	143	7 (3)	15/4	1/123	0/124	0/124
Meat	120	3 (2)	32/8	2/78	0/80	0/80
MedicalImages	1141	10 (5)	36/9	2/1094	0/1096	0/1096
MoteStrain	1272	2 (1)	548/137	27/560	5/582	1/586
Plane	210	7 (5)	24/6	1/179	0/180	0/180
Strawberry	983	2 (1)	281/70	14/618	3/629	0/632
Symbols	1020	6 (6)	134/33	7/846	1/852	0/853
ToeSegmentation1	268	2 (0)	112/28	6/122	1/127	0/128
ToeSegmentation2	166	2 (0)	99/25	5/37	1/41	0/42
Trace	200	4 (1)	40/10	2/148	0/150	0/150
TwoLeadECG	1162	2 (1)	465/116	23/558	5/576	0/581
Wafer	7164	2 (1)	610/152	30/6372	6/6396	1/6401
Wine	111	2 (1)	46/11	2/52	0/54	0/54

TS is the total number of time series instances in the data set. # CL describes the number of classes in the original classification setup; the label of the class chosen as the normal class is given in parentheses. The number of normal time series is denoted with # N, the number of anomalous instances with # A. The number of normal time series for a data set remains unchanged for each train-test split. The last three columns give the number of anomalous time series (# A) within the created data sets with 5%, 1%, and 0.1% of anomalies in the training data

6.2 Sensitivity analysis for hyperparameters

A general problem in unsupervised anomaly detection is hyperparameter optimization since standard optimization algorithms that rely on supervised metrics are not appli-

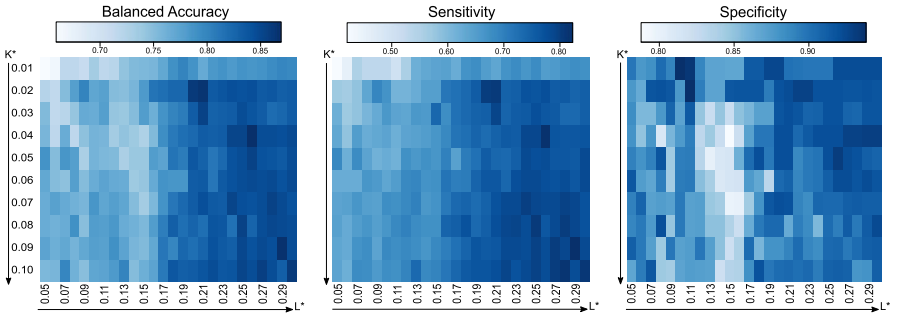

Fig. 6 Sensitivity analysis for the hyperparameters K and L of ADSL, defined relative to the time series lengths Q as $K = K^* \cdot Q$ and $L = L^* \cdot Q$. For each combination (K^*, L^*), the median test performance over all considered data sets is shown

cable. However the correct hyperparameters are critical for the anomaly detection methods to perform (Thomas et al. 2016).

The ADSL optimization problem in Eq. (7) depends on the hyperparameters C, K and L. The heuristic from Schölkopf et al. (2000) is used to set the penalty parameter C in relation to the assumed anomaly rate α as $C = \frac{1}{\alpha \cdot N}$. In order to choose the number K and the length L of the shapelets, a sensitivity analysis is performed. To account for the fact that different data sets contain time series of different lengths Q, we express K and L relative to Q, i.e., $K = K^* \cdot Q$ and $L = L^* \cdot Q$. In the sensitivity analysis, K^* and L^* are varied in the intervals $K^* \in [0.01, 0.1]$ and $L^* \in [0.05, 0.3]$. Larger values are not investigated to preserve the meaning of shapelets as short and characteristic subsequences, and to achieve reasonable run time. During the sensitivity analysis the anomaly rate is fixed to $\alpha = 5\%$. The sensitivity experiments are run on 11 data sets (`Adiac`, `Beef`, `BeetleFly`, `BirdChicken`, `ECGFiveDays`, `FaceFour`, `Lightning7`, `MedicalImages`, `Symbols`, `Trace`, `TwoLeadECG`), and the median test performance for each (K^*, L^*) combination over these data sets is reported. There is no individual tuning for each data set. Figure 6 shows the results of the hyperparameter sensitivity analysis.

A general trend can be observed in the sensitivity analysis, that ADSL performs better with longer and a larger number of shapelets. The algorithm learns more characteristics of the normal instances in the training data, and consequently detects anomalies better. However, in this parameter regime there is an increased chance of creating false alarms (i.e., detecting normal time series as anomalies) due to overfitting the training data, which leads to a decreased specificity. In contrast, reducing the number of shapelets increases the specificity, because the small set of shapelets is most discriminative of the normal class.

This analysis provides a guideline on how to choose K and L according to the requirements of the application: if a higher true positive rate is required, we advise the reader to set K^* close to 0.1 and L^* close to 0.3. This is relevant, e.g., in healthcare, where we would rather accept a false positive alarm than miss a medical issue. If avoiding false alarms is of greater importance, L^* should be close to 0.3 and K^* around 0.04. The latter applies, e.g., for alarm systems. For the evaluation in the

remainder of the article, we choose $K^* = 0.02$ and $L^* = 0.2$. This combination does not bias the results in either of the two mentioned preferences. Furthermore, it is computationally efficient for long time series.

6.3 Comparison to competing methods

The performance of the ADSL algorithm (Algorithm 2) is compared to the state-of-the-art algorithm for unsupervised anomaly detection for entire time series from Hyndman et al. (2015b). The implementation of the method is available as an `R` package (Hyndman et al. 2015a). As described in Sect. 2, the algorithm is based on the estimated density of the first two principle component scores extracted from 13 predefined features. Anomalous series are detected by their deviation from the highest density region. In the investigated data sets, some of these predefined features are practically constant. We add random noise with standard deviation 10^{-5} by applying the function `jitter` to the data since the method is not applicable in those cases. Additionally, we implemented a method to extract the estimated density and the principal components of the transformed training data and a test function, since the open source implementation does not yield predictions for test data without retraining the model. The test function extracts the 13 predefined features and projects them onto the learned principal components. These transformed feature values are then evaluated for their corresponding density value. The test time series instance is detected as an anomaly if this value is below the density threshold for the corresponding anomaly rate $\alpha \in \{0.1\%, 1\%, 5\%\}$.

In addition, we compare ADSL with shapelet learning to the shapelet extraction method described in Sect. 4.3. This baseline method only searches over subsequences that are actually present in the training data, but does not optimize shapelets. We use the abbreviation "Extract" in the following to denote this baseline approach.

7 Results

The first performance evaluation investigates the case where the true anomaly rate α is known, and the hyperparameter C is set accordingly. The results for this case are illustrated in Fig. 7. Table 2 contains the detailed numbers for each experiment, as well as the aggregated performance and rank comparisons between the three investigated methods. Averaged over all data sets, ADSL outperforms both baseline methods in terms of median balanced accuracy for all levels of α. The difference is statistically significant. Furthermore, ADSL's performance on average is very stable across all anomaly rate settings α, showing that the method yields reliable results even if the expected number of anomalies is relatively large.

All three methods struggle with correctly detecting anomalies for several data sets (`ChlorineConcentration`, `ECG200`, `Ham`, `Herring`, `Wafer`, `Wine`). On closer examination, we observe that in each of those data sets the time series are not clearly distinguishable even though belonging to different classes in the original classi-

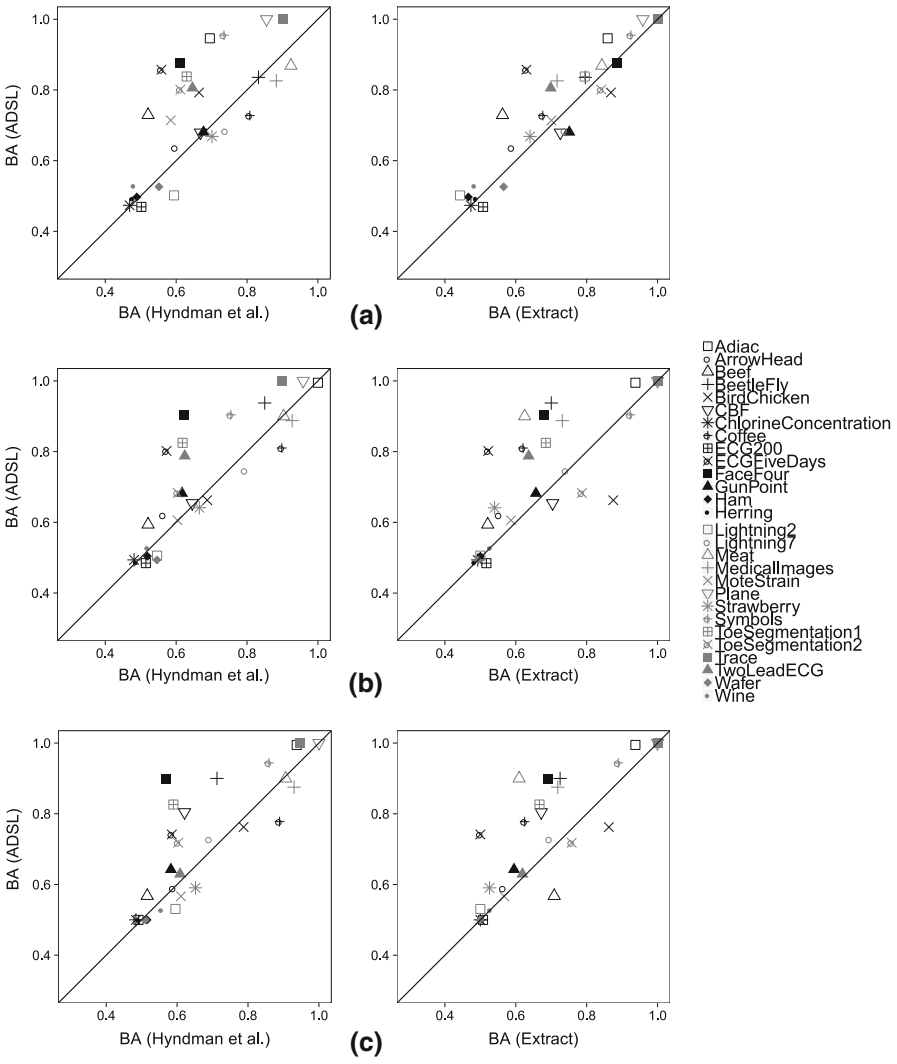

Fig. 7 Comparison of balanced accuracy (BA) on the 28 investigated data sets between ADSL and two competing methods. The left column compares our method with Hyndman et al. (2015b), the right column compares with the shapelet extraction method from Sect. 4.3. The x- and y-axes show the balanced accuracy of the competing method and ADSL, respectively. Points above the diagonal show an advantage for ADSL. The three rows correspond to three different settings of the true anomaly rate: **a** $\alpha = 5\%$, **b** $\alpha = 1\%$, **c** $\alpha = 0.1\%$ and setting the anomaly rate parameter C accordingly. ADSL outperforms the competing methods on most data sets, with the most visible advantage over Hyndman et al. (2015b) for the high-anomaly case $\alpha = 5\%$

Table 2 Performance comparison on 28 benchmark data sets

Data sets	Anomaly rate $\alpha = 5\%$			Anomaly rate $\alpha = 1\%$			Anomaly rate $\alpha = 0.1\%$		
	ADSL	Hynd. et al.	Extract	ADSL	Hynd. et al.	Extract	ADSL	Hynd. et al.	Extract
Adiac	**0.95** (0.05)	0.70 (0.13)	0.86 (0.16)	0.99 (0.06)	**1.00** (0.13)	0.94 (0.09)	**0.99** (0.10)	0.94 (0.09)	0.94 (0.09)
ArrowHead	**0.64** (0.03)	0.60 (0.06)	0.59 (0.03)	**0.62** (0.03)	0.56 (0.05)	0.55 (0.04)	**0.59** (0.00)	**0.59** (0.05)	0.57 (0.02)
Beef	**0.73** (0.12)	0.52 (0.19)	0.56 (0.12)	**0.59** (0.15)	0.52 (0.18)	0.52 (0.13)	0.57 (0.15)	0.52 (0.20)	**0.71** (0.18)
BeetleFly	**0.84** (0.08)	0.83 (0.21)	0.80 (0.08)	**0.9**4 (0.08)	0.85 (0.12)	0.70 (0.06)	**0.90** (0.08)	0.71 (0.15)	0.72 (0.07)
BirdChicken	0.79 (0.09)	0.66 (0.10)	**0.87** (0.05)	0.66 (0.11)	0.69 (0.12)	**0.8**8 (0.12)	0.76 (0.13)	0.79 (0.07)	**0.86** (0.08)
CBF	0.68 (0.03)	0.67 (0.07)	**0.73** (0.04)	0.65 (0.01)	0.65 (0.05)	**0.70** (0.02)	**0.80** (0.04)	0.62 (0.05)	0.67 (0.02)
ChlorineConcentration	**0.47** (0.01)	**0.47** (0.01)	**0.47** (0.01)	**0.49** (0.00)	0.48 (0.01)	**0.49** (0.01)	**0.50** (0.00)	0.48 (0.01)	**0.50** (.000)
Coffee	0.73 (0.05)	**0.81** (0.11)	0.68 (0.09)	0.81 (0.03)	**0.90** (0.08)	0.62 (0.06)	0.78 (0.05)	**0.89** (0.06)	0.62 (0.06)
ECG200	0.47 (0.04)	0.50 (0.07)	**0.51** (0.05)	0.48 (0.04)	0.51 (0.07)	**0.52** (0.02)	0.50 (0.03)	0.49 (0.05)	**0.51** (0.02)
ECGFiveDays	**0.86** (0.01)	0.56 (0.06)	0.63 (0.03)	**0.80** (0.16)	0.57 (0.07)	0.52 (0.05)	**0.74** (0.09)	0.59 (0.04)	0.50 (0.02)
FaceFour	0.88 (0.11)	0.61 (0.09)	**0.89** (0.04)	**0.90** (0.05)	0.62 (0.09)	0.68 (0.07)	**0.90** (0.10)	0.57 (0.09)	0.69 (0.11)
GunPoint	0.68 (0.04)	0.68 (0.05)	**0.75** (0.03)	**0.68** (0.06)	0.62 (0.05)	0.66 (0.05)	**0.64** (0.04)	0.58 (0.06)	0.60 (0.01)
Ham	**0.50** (0.03)	0.49 (0.04)	0.47 (0.03)	0.50 (0.03)	**0.52** (0.04)	0.50 (0.01)	0.50 (0.01)	**0.52** (0.02)	0.50 (0.02)
Herring	**0.49** (0.04)	0.48 (0.05)	**0.49** (0.04)	**0.49** (0.02)	**0.49** (0.02)	0.48 (0.03)	**0.50** (0.00)	0.49 (0.03)	**0.50** (0.01)
Lightning2	0.50 (0.07)	**0.59** (0.06)	0.44 (0.04)	0.51 (0.05)	**0.55** (0.06)	0.50 (0.03)	0.53 (0.05)	**0.60** (0.02)	0.50 (0.02)
Lightning7	0.68 (0.07)	**0.74** (0.08)	**0.74** (0.15)	0.75 (0.10)	**0.79** (0.12)	0.74 (0.08)	**0.73** (0.11)	0.69 (0.16)	0.70 (0.05)
Meat	0.87 (0.05)	**0.92** (0.11)	0.84 (0.09)	**0.90** (0.01)	**0.90** (0.12)	0.62 (0.11)	0.90 (0.01)	**0.91** (0.13)	0.61 (0.14)
MedicalImages	0.83 (0.05)	**0.88** (0.05)	0.72 (0.05)	0.89 (0.03)	**0.93** (0.07)	0.73 (0.03)	0.88 (0.03)	**0.93** (0.04)	0.72 (0.05)
MoteStrain	**0.71** (0.03)	0.58 (0.03)	0.70 (0.01)	**0.61** (0.01)	0.60 (0.03)	0.59 (0.01)	0.57 (0.01)	**0.61** (0.02)	0.57 (0.02)

Table 2 continued

Data sets	Anomaly rate $\alpha = 5\%$			Anomaly rate $\alpha = 1\%$			Anomaly rate $\alpha = 0.1\%$		
	ADSL	Hynd. et al.	Extract	ADSL	Hynd. et al.	Extract	ADSL	Hynd. et al.	Extract
Plane	**1.00** (0.04)	0.85 (0.11)	0.96 (0.07)	**1.00** (0.04)	0.96 (0.08)	**1.00** (0.06)	**1.00** (0.04)	**1.00** (0.04)	**1.00** (0.06)
Strawberry	0.67 (0.02)	**0.70** (0.02)	0.64 (0.01)	0.64 (0.01)	**0.67** (0.01)	0.54 (0.01)	0.59 (0.01)	**0.65** (0.03)	0.53 (0.01)
Symbols	**0.95** (0.03)	0.73 (0.04)	0.92 (0.04)	0.90 (0.03)	0.75 (0.08)	**0.92** (0.05)	**0.94** (0.01)	0.86 (0.12)	0.89 (0.04)
ToeSegmentation1	**0.84** (0.03)	0.63 (0.04)	0.79 (0.03)	**0.82** (0.02)	0.62 (0.03)	0.68 (0.03)	**0.83** (0.01)	0.59 (0.03)	0.67 (0.04)
ToeSegmentation2	0.80 (0.10)	0.61 (0.04)	**0.84** (0.04)	0.68 (0.06)	0.60 (0.04)	**0.79** (0.03)	0.72 (0.03)	0.60 (0.04)	**0.76** (0.02)
Trace	**1.00** (0.02)	0.90 (0.08)	**1.00** (0.00)	**1.00** (0.03)	0.90 (0.10)	**1.00** (0.02)	**1.00** (0.04)	0.95 (0.09)	**1.00** (0.02)
TwoLeadECG	**0.81** (0.02)	0.65 (0.01)	0.70 (0.01)	**0.79** (0.02)	0.62 (0.01)	0.64 (0.01)	**0.63** (0.05)	0.61 (0.01)	0.62 (0.00)
Wafer	0.53 (0.01)	0.55 (0.06)	**0.57** (0.01)	0.49 (0.00)	**0.55** (0.04)	0.49 (0.00)	0.50 (0.00)	**0.51** (0.02)	0.50 (0.00)
Wine	**0.53** (0.02)	0.48 (0.05)	0.48 (0.06)	**0.53** (0.03)	0.52 (0.09)	**0.53** (0.00)	0.53 (0.02)	**0.56** (0.04)	0.53 (0.01)
Median BA	**0.73**	0.64	0.71	**0.68**	0.62	0.63	**0.72**	0.61	0.62
Average rank (sd)	**1.57** (0.69)	2.39 (0.83)	2.00 (0.77)	**1.61** (0.69)	2.04 (0.88)	2.21 (0.88)	**1.57** (0.63)	2.11 (0.96)	2.00 (0.77)
p-value	–	0.006	0.035	–	0.033	0.005	–	0.045	0.003

The ADSL method is compared to the method by Hyndman et al. (2015b) (*Hynd. et al.*), and the shapelet extraction method from Sect. 4.3 (*Extract*). The median balanced accuracy (BA) is reported over 10 different train–test splits, standard deviations are in parentheses. The results are presented separately for each true anomaly rate α in the training data. Bold numbers indicate the best performance for each scenario. We further show the median performance over all data sets, the average rank of each of the three methods, and the corresponding standard deviation (sd). The p values of the pair-wise Wilcoxon test are shown in the last row. For both shapelet-based methods we use the hyperparameter setting $K^* = 0.02$ and $L^* = 0.2$. The advantage of ADSL is statistically significant ($p < 0.05$) for all three α scenarios

fication setting, leading to the conclusion that these classification benchmarks can not be transformed into anomaly detection problems without further domain knowledge.

7.1 Analysis of limitations of shapelet-based methods

Shapelet-based methods are not necessarily optimal for all types of time series problems. When analyzing the cases in our experiments where ADSL does not outperform the method of Hyndman et al. (2015b), we observe that this is for data sets with high variability, which means there are no characteristic shapes for time series of one class. The data set `Lightning7` and `Lightning2` fall into this category, where standard global time series features such as slope or trend, together with domain specific features capture the characteristics of the data set better than shapelet features.

7.2 Comparing learned and extracted shapelets

The comparison to the Extract method from Sect. 4.3 shows that learning and optimizing shapelets indeed helps in almost all cases. ADSL on average performs better for all three anomaly settings and the difference is statistically significant. In contrast to Extract, ADSL not only learns shapelets that are representative of the majority of the data, but additionally forces the least similar training examples under this shapelet transformation to lie outside the decision boundary. This advantage is slightly diminished when the training set consists of many anomalous series since the overall distance in Sect. 4.3 is sensitive to large values. This is the case for $\alpha = 5\%$, leading to a less significant improvement of ADSL over the baseline method from Sect. 4.3. However, in the cases where Extract outperforms ADSL, the improvement typically is small.

7.3 Influence of anomaly rate

In three data sets (`TwoLeadECG`, `ECGFiveDays`, `MoteStrain`) a noticeable drop in balanced accuracy for ADSL can be observed if α is reduced. In the same data sets, for the high anomaly setting, ADSL performs well, but the competitive methods struggle. We now investigate the reasons for this behavior.

The data sets `TwoLeadECG` and `ECGFiveDays` are electrocardiogram measurements, and contain time series showing only small differences between the two classes. `MoteStrain` contains sensor measurements, however, the time series show similar characteristics as electrocardiogram data. Fig. 8 illustrates the time series from the two classes in `ECGFiveDays`, as well as the learned and extracted shapelets. In Table 3 the corresponding radius of SVDD* are displayed.

When the training set contains anomalies, ADSL with an appropriate setting of parameter C will detect the least similar time series in the training set, and learns a tighter description of normality, even though no labels are used. Comparing the normal and anomalous time series, a very homogeneous peak around time point $t = 50$ can be observed, whereas peaks of different amplitudes are present in the anomalies. The anomalous time series have a prolonged and flattened plateau between $t = 60$ and

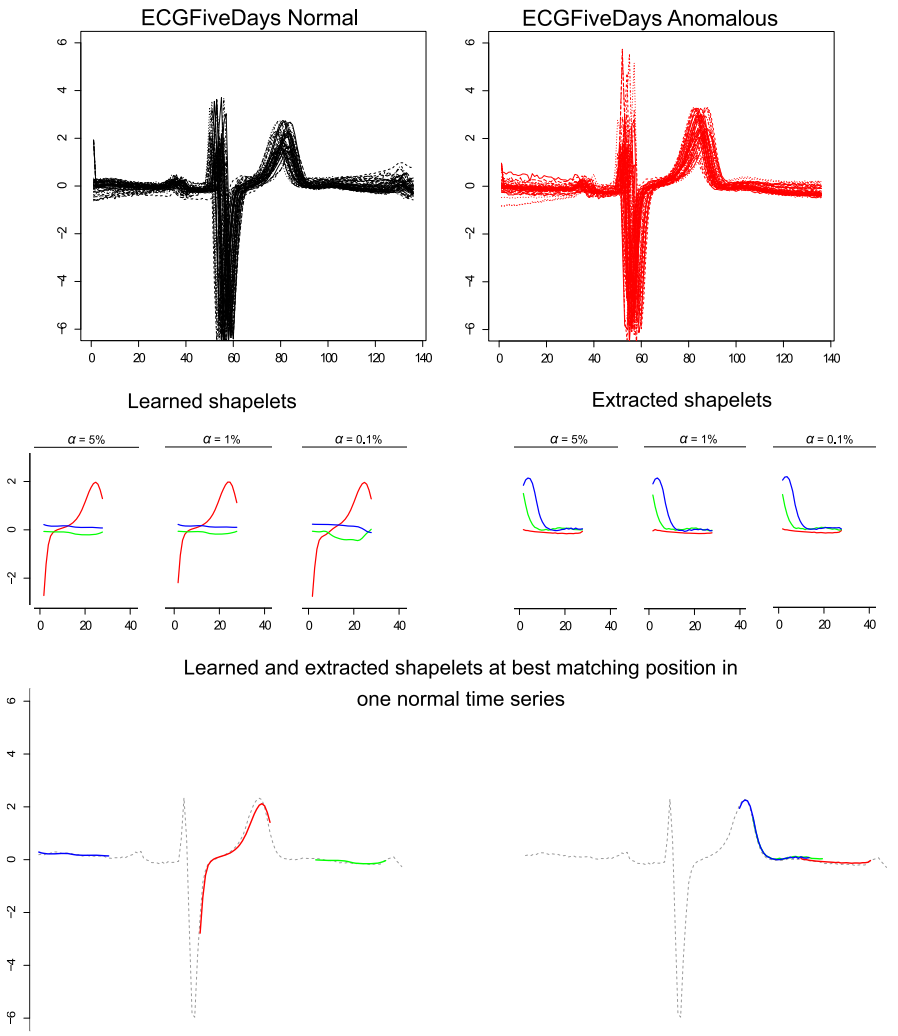

Fig. 8 Analysis of anomaly detection in the `ECGFiveDays` data set. (top) Visualization of 50 instances of the normal (left) and anomalous (right) class. (middle) The learned (left) and extracted (right) shapelets for each anomaly rate $\alpha \in 5\%, 1\%, 0.1\%$. (bottom) The learned and extracted shapelets for the setting $\alpha = 5\%$ plotted at the minimal-distance location for a randomly chosen normal time series

$t = 70$. Moreover, normal time series show more variability towards the end of the observation, whereas in anomalies the variability is bigger at the beginning. The $K = 3$ learned shapelets account for the smooth measurements at the beginning of the normal time series and at the end, and for the slope of the curve between time points $t = 60$ and $t = 80$.

In contrast, the shapelets that are extracted (without shapelet learning) only represent characteristic patterns towards the end of the time series. Large parts are overlapping in time, and are very similar in all anomaly rate settings. This hinders

Table 3 Median radius of SVDD*. We display the median radius of SVDD* over the 10 resampling iterations on ECG5Days for ADSL with learned shapelets and for Extract from Sect. 4.3 with extracted shapelets, respectively. The median radius is shown for each anomaly rate $\alpha \in \{5\%, 1\%, 0.1\%\}$ and one standard deviation is added in parentheses

Median radius (sd)	5%	1%	0.1%
ADSL	0.18 (0.003)	0.20 (0.018)	0.21 (0.015)
Extract	0.14 (0.010)	0.18 (0.016)	0.31 (0.026)

the baseline method from Sect. 4.3 to correctly detect anomalous time series in test data even though the radius of the SVDD* is smaller for the extracted shapelets for $\alpha \in \{1\%, 5\%\}$.

For $\alpha = 0.1\%$, the model assumes that all time series are normal and thus all training time series must be within the enclosing hypersphere. This results in a higher radius compared to the other anomaly rate settings for both learned and extracted shapelets. Nevertheless, the radius for ADSL is still small with 0.21, leading to a better performance compared to Extract. Extract uses a radius of 0.31, which is large in comparison to the radii for the settings $\alpha = \{5\%, 1\%\}$.

7.4 ADSL with unknown anomaly rate

We now evaluate the behavior of the three methods in case the expected percentage of anomalies in the training set is a priori unknown. In order to demonstrate that ADSL can cope with an unknown number of expected anomalies during training, we repeat the experiments and create training sets with true anomaly rates of $\alpha = 0.1\%$ and $\alpha = 1\%$. This time, however, the anomaly-rate related parameter C is set to a non-ideal value that corresponds to $\alpha = 5\%$ during training. The performance values are visualized in Fig. 9, and the exact values for individual data sets can be found in Table 4.

By assuming a higher anomaly rate than really present, we force the SVDD* to learn a tighter anomaly boundary and consequently a more compact shapelet transformation of our data, leading to an improved anomaly detection. Consequently, ADSL shows an improved median performance under both true anomaly rates if we overestimate the true anomaly rate. Compared to the method of Hyndman et al. (2015b), ADSL's performance is significantly better.

8 Discussion

In this article, we have shown that learned shapelet features, which have so far only been used for time series classification (Ye and Keogh 2009; Lines et al. 2012; Grabocka et al. 2014), hold great promise for the unsupervised detection of anomalous time series. Together with the modified SVDD framework (Tax and Duin 2004), and a novel objective function for optimization (Sect. 5.1), a powerful mechanism has been

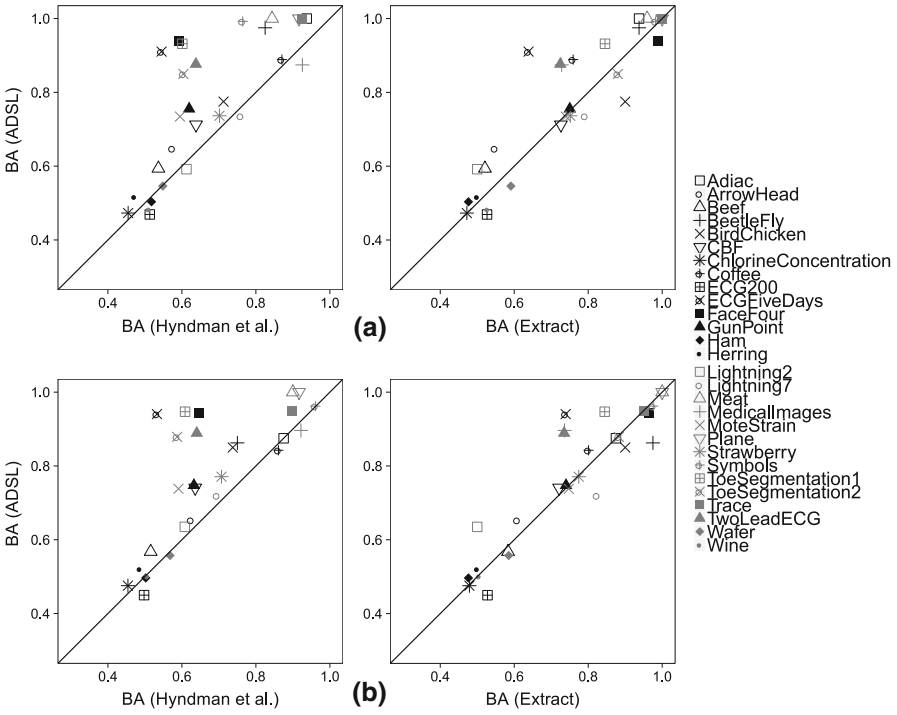

Fig. 9 Same as Fig. 7, but under the assumption that the anomaly rate is unknown. The true anomaly rate in the training set [$\alpha = 1\%$ in (**a**), and $\alpha = 0.1\%$ in (**b**)] does not match the setting of the C parameter, which is set as if $\alpha = 5\%$. ADSL outperforms Hyndman et al. (2015b) significantly on most data sets. The Extract method also benefits from the assumed higher anomaly rate, thus the difference to ADSL is not as big

proposed that can not only learn to distinguish normal from anomalous time series in a training set, but also learns characteristic features of normal time series.

To discuss our results, we additionally summarize the most important data set characteristics in Table 5: the number of training observations, the length of the series and the average normalized euclidean distance (ED) between the training series. The latter is equivalent to the shapelet distance from Eq. (2) where a shapelet is now an entire time series. Additionally, we evaluate the standard deviation of the differences between two successive measurements within a time series as measure Ψ where a smaller value indicates smoother time series.

Interpreting Table 5 together with the results from Sect. 7 and Table 2, we observe that for smooth series ($\Psi < 0.4$) the shapelet-based methods perform better than the method by Hyndman et al. (2015b) which uses fixed time series features. The smooth shape makes it easier for the shapelet-based methods to find meaningful patterns. Moreover, the length of the time series influences the results only in combination with other data set characteristics: in general, longer time series are better for the shapelet-based method to find representative shapes of the normal class. If however the time

Table 4 Performance comparison on 28 data sets when the true anomaly rate is unknown

Data sets	True anomaly rate $\alpha = 1\%$			True anomaly rate $\alpha = 0.1\%$		
	ADSL	Hynd. et al.	Extract	ADSL	Hynd. et al.	Extract
Adiac	**1.00** (0.12)	0.94 (0.12)	0.94 (0.09)	**0.88** (0.09)	**0.88** (0.11)	**0.88** (0.08)
ArrowHead	**0.65** (0.04)	0.58 (0.05)	0.55 (0.03)	**0.65** (0.03)	0.63 (0.07)	0.61 (0.05)
Beef	**0.59** (0.15)	0.54 (0.13)	0.52 (0.09)	0.57 (0.15)	0.52 (0.18)	**0.58** (0.21)
BeetleFly	**0.98** (0.11)	0.82 (0.13)	0.94 (0.10)	0.86 (0.09)	0.75 (0.11)	**0.98** (0.10)
BirdChicken	0.78 (0.14)	0.71 (0.12)	**0.90** (0.13)	0.85 (0.15)	0.74 (0.10)	**0.90** (0.12)
CBF	0.71 (0.02)	0.64 (0.08)	**0.73** (0.05)	**0.74** (0.03)	0.64 (0.04)	0.72 (0.04)
ChlorineConcentration	**0.47** (0.01)	0.45 (0.01)	**0.47** (0.01)	**0.48** (0.01)	0.45 (0.02)	**0.48** (0.01)
Coffee	**0.89** (0.04)	0.87 (0.04)	0.76 (0.08)	0.84 (0.04)	**0.86** (0.12)	0.80 (0.09)
ECG200	0.47 (0.03)	0.51 (0.04)	**0.53** (0.03)	0.45 (0.04)	0.50 (0.08)	**0.53** (0.02)
ECGFiveDays	**0.91** (0.12)	0.55 (0.02)	0.64 (0.06)	**0.94** (0.11)	0.53 (0.08)	0.74 (0.06)
FaceFour	0.94 (0.04)	0.59 (0.08)	**0.99** (0.03)	0.94 (0.10)	0.65 (0.10)	**0.96** (0.08)
GunPoint	**0.76** (0.01)	0.62 (0.08)	0.75 (0.02)	**0.75** (0.03)	0.63 (0.05)	0.74 (0.03)
Ham	0.50 (0.04)	**0.52** (0.03)	0.48 (0.03)	**0.50** (0.02)	**0.50** (0.03)	0.48 (0.03)
Herring	**0.52** (0.03)	0.47 (0.03)	0.50 (0.02)	**0.52** (0.02)	0.49 (0.07)	0.50 (0.03)
Lightning2	0.59 (0.04)	**0.61** (0.08)	0.50 (0.04)	**0.63** (0.07)	0.61 (0.06)	0.50 (0.06)
Lightning7	0.74 (0.08)	0.76 (0.13)	**0.79** (0.12)	0.72 (0.10)	0.70 (0.13)	**0.82** (0.07)
Meat	**1.00** (0.02)	0.84 (0.12)	0.96 (0.03)	**1.00** (0.04)	0.90 (0.16)	**1.00** (0.06)
MedicalImages	0.87 (0.04)	**0.93** (0.06)	0.73 (0.04)	0.90 (0.03)	**0.92** (0.05)	0.74 (0.06)
MoteStrain	0.73 (0.01)	0.59 (0.01)	**0.74** (0.01)	0.74 (0.01)	0.59 (0.02)	**0.75** (0.01)

Table 4 continued

Data sets	True anomaly rate $\alpha = 1\%$			True anomaly rate $\alpha = 0.1\%$		
	ADSL	Hynd. et al.	Extract	ADSL	Hynd. et al.	Extract
Plane	**1.00** (0.04)	0.92 (0.07)	**1.00** (0.06)	**1.00** (0.04)	0.92 (0.11)	**1.00** (0.06)
Strawberry	0.74 (0.02)	0.70 (0.03)	**0.75** (0.04)	**0.77** (0.03)	0.71 (0.03)	**0.77** (0.03)
Symbols	**0.99** (0.01)	0.76 (0.07)	0.98 (0.01)	0.96 (0.02)	0.96 (0.04)	**0.98** (0.01)
ToeSegmentation1	**0.93** (0.03)	0.60 (0.04)	0.85 (0.03)	**0.95** (0.01)	0.61 (0.05)	0.84 (0.01)
ToeSegmentation2	0.85 (0.02)	0.60 (0.03)	**0.88** (0.03)	**0.88** (0.02)	0.59 (0.04)	**0.88** (0.02)
Trace	**1.00** (0.03)	0.92 (0.10)	**1.00** (0.03)	**0.95** (0.06)	0.90 (0.08)	**0.95** (0.06)
TwoLeadECG	**0.88** (0.01)	0.64 (0.01)	0.72 (0.00)	**0.89** (0.01)	0.64 (0.01)	0.73 (0.01)
Wafer	0.55 (0.02)	0.55 (0.10)	**0.59** (0.01)	0.56 (0.02)	0.57 (0.10)	**0.59** (0.00)
Wine	0.48 (0.03)	0.51 (0.10)	**0.53** (0.02)	0.50 (0.04)	**0.51** (0.06)	**0.51** (0.04)
Median BA	**0.77**	0.62	0.74	**0.81**	0.63	0.74
Average rank (sd)	**1.61** (0.74)	2.50 (0.69)	1.79 (0.79)	1.68 (0.67)	2.50 (0.79)	**1.61** (0.79)
p value	–	0.001	0.061	–	0.000	0.260

The median balanced accuracy is reported for two settings where $\alpha = 1\%$ or $\alpha = 0.1\%$ in the training set, but the hyperparameter C, which is related to the assumed anomaly rate is set as if $\alpha = 5\%$. The test performance is measured over 10 random train–test splits for each of the 28 data sets. The best performance value is highlighted in bold for each comparison. We further show the median performance over all data sets, the average rank of the three methods, and the corresponding standard deviation (sd). The p-values of the pair-wise Wilcoxon test are presented in the last row. For the shapelet-based methods we use the hyperparameter setting $K^* = 0.02$ and $L^* = 0.2$. ADSL performs significantly better than the method of Hyndman et al. (2015b), and slightly better than the shapelet extraction algorithm for the case of anomalies present in training data

Table 5 Data set characteristics. We display additional data set characteristics, including the number of normal time series in the training set, the length of the time series and the average normalized euclidean distance (ED) between the training time series. Additionally, the standard deviation of the differences between two successive measurements within a time series is provided as measure Ψ which is an inverse estimate for smoothness. Smaller values indicate smoother behavior

	# N in training	Length	Average ED	Ψ (inv. smoothness)
Adiac	16	176	0.05	0.08
ArrowHead	52	251	0.09	0.06
Beef	10	470	1.16	0.08
BeetleFly	16	512	1.61	0.09
BirdChicken	16	512	1.13	0.04
CBF	248	128	1.31	0.70
ChlorineConcentration	801	166	0.51	0.64
Coffee	23	286	0.02	0.08
ECG200	54	96	1.11	0.27
ECGFiveDays	354	136	0.89	0.52
FaceFour	23	350	1.13	0.37
GunPoint	80	150	0.24	0.07
Ham	82	431	0.41	0.21
Herring	62	512	0.10	0.04
Lightning2	38	637	1.34	0.42
Lightning7	11	319	1.39	1.04
Meat	32	448	0.02	0.04
MedicalImages	36	99	0.25	0.54
MoteStrain	549	84	0.76	0.49
Plane	24	144	0.39	0.16
Strawberry	281	235	0.01	0.08
Symbols	134	398	0.14	0.02
ToeSegmentation1	112	277	1.97	0.20
ToeSegmentation2	99	343	1.95	0.14
Trace	40	275	1.07	0.23
TwoLeadECG	465	82	0.24	0.21
Wafer	611	152	1.72	0.38
Wine	46	234	0.00	0.12

series are short (Length < 100), more series are needed for training the shapelet-based methods. In contrast, the method proposed by Hyndman et al. (2015b) is able to extract meaningful features from few short series. In case the training set consists of many normal series (in our case # N > 100), only ADSL shows a good performance. Here, ADSL generalizes the characteristic shapelets over an already broad range of normal observations and consequently can learn a very distinct description of the generalized normal behavior. However, if the number of training observations is too large (in our

case # N $>$ 600), ADSL learns on a too broad spectrum of samples from normal observations and thus struggles to learn a distinct description of normal behavior.

Our method shows good performance in an unsupervised scenario, i.e., no labels are available a priori to indicate which training examples are anomalies. We believe that this is practically the most relevant scenario, as it only requires a typical data set, which contains mostly normal data, and an estimate of the anomaly rate to set the anomaly rate related hyperparameter C. If the normal class is homogeneous (i.e., the average ED between the training time series is small, in our case < 0.4), it is not even necessary to have large training sets to obtain good performance, which can also be seen in our results and in Table 5. However, if there is almost no variability in the normal class (i.e., average ED < 0.02), ADSL requires more series for training.

The main assumption of shapelet-based methods is that short subsequences exist that are characteristic of the different classes. This is of course not always the case, but shapelets have shown state-of-the-art performance on many relevant time series classification benchmarks (Bagnall et al. 2017a). By evaluating ADSL on a variety of data sets, we find that ADSL equally gives excellent performance for anomaly detection in most cases. We also identified characteristics of the data sets that are challenging for ADSL. The main requirement for ADSL to work well is to have representative patterns and little variability in the normal class, which are common assumptions for anomaly detection. This circumstance can be observed for the data set `Lighting7`: this data set has a high inverse-smoothness value ($\Psi > 1$ in Table 5) indicating a rather non-smooth time series. ADSL misses to learn a representative shapelet from this data as discussed in Sect. 7.1. The same holds for `MedicalImages`, where the training data shows an inverse-smoothness value of 0.54. This would not lead automatically to ADSL performing worse (as can be seen in `CBF` and `MoteStrain`), but in combination with only a small number of short series, ADSL is not able to learn characteristic shapelet features. In those cases, the method of Hyndman et al. (2015b) performs best.

The alternative approach of assuming a training set without anomalies implicitly requires reviewing every time series by a domain expert, which in the end results in a supervised scenario. Even less attractive is the case where a representative labeled set of anomalies is needed, because this will require the collection and labeling of an even larger data set. Even then, in practice no guarantee can be given that every possible anomaly has been observed, or can be identified as such. Furthermore, in many real-world applications such as medicine or process control, the situations in which anomalies occur are typically highly undesirable or unlikely to ever be observed. The performance of ADSL does not depend on ever seeing any anomalies during training, and neither is it required to learn from a completely clean training set as is the case in most of the current literature (Mahoney and Chan 2005; Salvador and Chan 2005; Rebbapragada et al. 2009).

Besides the accuracy and minimal requirements on the training data, ADSL is also an efficient method for classifying new time series at test time: the method only searches for the best match of each shapelet within the time series, and then performs a simple comparison with the decision boundary. For a time series of length Q, and a set of K shapelets with length L this results in a run time of $O(Q \cdot L \cdot K)$, which is efficient since K and L are typically small. In contrast, in the current implementation

of Hyndman et al. (2015b), testing an unseen series requires retraining the complete model: extracting the 13 predefined features for all time series (training series and the new unseen series), calculating the principal component transformation and the density, and finally evaluating if the transformed new observation is below a density value threshold. This nevertheless is only an issue in the current implementation available in Hyndman et al. (2015a) and its extension to a separate testing functionality by extracting the learned principal component basis and density threshold is straightforward.

Furthermore, shapelet features have an intuitive explanation as the distance of the best match between the shapelet and any subsequence of a time series. This implies that we expect feature vectors for normal time series to have small norm, and that we can choose a radius of the decision boundary that reflects our assumptions on the expected anomaly rate. Optimizing shapelets as in our proposed ADSL method via shapelet learning (Grabocka et al. 2014) had a positive impact on the anomaly detection performance compared to simple shapelet extraction from Sect. 4.3. The main reason is the better generalization capability of learned shapelets, which adapt to the average over the whole training set, rather than representing a short piece of a single time series. Since the decision is strongly influenced by the maximum shapelet distance, tighter bounds for anomaly detection can be learned if the features generalize well. This still holds when anomalies are present in the training data by setting the hyperparameter C appropriately, and their presence is expected.

In general, we observe that overestimating the true anomaly rate α is a good recipe to achieve robust anomaly detection. The performance remains at a high level, even if the true anomaly rate is lower, and is consistently better than for the method of Hyndman et al. (2015b). The SVDD*, which is largely responsible for this robustness, also improves the performance in this scenario when shapelet features are extracted rather than learned. By setting the anomaly-rate related parameter C to a higher anomaly rate, ADSL has an incentive to detect more anomalies in the training data than are actually present. This means that also the least similar normal time series instances will be categorized as anomalies. On the other hand, a setting of C corresponding to an underestimated anomaly rate will force the method to include anomalous characteristics in the model for normality. This will clearly have a negative impact on the performance. In the case where the training set is entirely made up of normal data, which is a typical assumption for other methods (Rebbapragada et al. 2009; Mahoney and Chan 2005; Salvador and Chan 2005), ADSL still performs well, even though this is not the setup for which ADSL was originally intended. In summary, this means that ADSL with an overestimated anomaly rate is well suited for anomaly detection in real-world applications where the true anomaly rate is unknown.

9 Conclusion

In this paper, we proposed a novel method for detecting anomalous time series in the scenario where the training set contains no labels and is contaminated with an unknown amount of anomalies. The ADSL method is able to learn representative features of normality by combining a shapelet learning approach with SVDD-based anomaly detection method, making our method highly efficient for processing large

sets of time series at test time. The shapelet approach, however, requires recurring short temporal patterns to indicate normal behavior or class membership, and a smooth behavior of the time series (cf. Sect. 8). If these conditions are met, ADSL can learn a representative set of shapelets from very small training sets. If the training set consists of many normal time series, the smoothness requirement does not need to hold. Moreover, we demonstrated that the use of a SVDD-based anomaly detection makes our method robust against the occasional presence of anomalies in the training set. Good performance with up to 5% of anomalies in training sets is shown on multiple benchmarks, and ADSL still performs well even if the true anomaly rate is unknown.

In conclusion, the experimental results show that 1) using shapelet features instead of statistic time series features as used in Hyndman et al. (2015b), and 2) an SVDD-based anomaly detection approach in contrast to an high-density region approach improves the performance in learning to detect anomalous time series instances. A generalization of our method to a non-linear transformation of the anomaly boundary in order to capture non-linear characteristics in the data thus looks promising. Future work will also address the extension to multivariate time series.

References

Bagnall A, Lines J, Bostrom A, Large J, Keogh E (2017a) The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. Data Min Knowl Disc 31(3):606–660

Bagnall A, Lines J, Vickers W, Keogh E (2017b) The UEA & UCR time series classification repository. www.timeseriesclassification.com. Accessed Dec 2017

Boyd S, Vandenberghe L (2004) Convex optimization. Cambridge University Press, Cambridge

Boyd S, Xiao L, Mutapcic A (2003) Subgradient methods. Lecture Notes of EE392o, Stanford University, Autumn Quarter 2004. http://web.mit.edu/6.976/www/notes/subgrad_method.pdf

Brodersen KH, Ong CS, Stephan KE, Buhmann JM (2010) The balanced accuracy and its posterior distribution. In: Proceedings of the 20th international conference on pattern recognition, IEEE, Istanbul, Turkey, pp 3121–3124

Chandola V (2009) Anomaly detection for symbolic sequences and time series data. Ph.D. thesis, University of Minnesota

Chandola V, Banerjee A, Kumar V (2009) Anomaly detection: a survey. ACM Comput Surv (CSUR) 41(3):1–58

Chatfield C (2016) The analysis of time series: an introduction, 6th edn. Chapman and Hall, London, UK

Chuah MC, Fu F (2007) ECG anomaly detection via time series analysis. In: Thulasiraman P, He X, Xu TL, Denko MK, Thulasiram RK, Yang LT (eds.), Frontiers of High Performance Computing and Networking ISPA 2007 Workshops. Lecture Notes in Computer Science, vol 4743, Springer, Berlin, Heidelberg, pp 123–135

Emmott AF, Das S, Dietterich T, Fern A, Wong WK (2013) Systematic construction of anomaly detection benchmarks from real data. In: Proceedings of the ACM SIGKDD workshop on outlier detection and description, ACM, Chicago, Illinois, pp 16–21

Esling P, Agon C (2012) Time-series data mining. ACM Comput Surv (CSUR) 45(1):1–34

Ferrell B, Santuro S (2015) Nasa shuttle valve data. http://www.cs.fit.edu/~pkc/nasa/data/

Forrest S, Hofmeyr SA, Somayaji A, Longstaff TA (1996) A sense of self for unix processes. In: Proceedings of the 1996 IEEE symposium on research on security and privacy, IEEE, pp 120–128

Fu AWC, Leung OTW, Keogh E, Lin J (2006) Finding time series discords based on haar transform. In: Li X, Zaïane OR, Li Z (eds) Proceedings of international conference on advanced data mining and applications, Springer, Berlin, Heidelberg, pp 31–41

Grabocka J, Schilling N, Wistuba M, Schmidt-Thieme L (2014) Learning time-series shapelets. In: Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining, ACM, New York, USA, pp 392–401

Gravetter FJ, Wallnau LB (2017) Statistics for the behavioral sciences, 10th edn. Cengage Learning, Boston

Hyndman RJ (1996) Computing and graphing highest density regions. Am Stat 50(2):120–126

Hyndman RJ, Wang E, Laptev N (2015a) Anomalous—a R package for unusual time series detection. https://github.com/robjhyndman/anomalous

Hyndman RJ, Wang E, Laptev N (2015b) Large-scale unusual time series detection. In: Proceedings of the 2015 IEEE international conference on data mining workshop, IEEE, Atlantic City, NJ, USA, pp 1616–1619

Keogh E, Kasetty S (2003) On the need for time series data mining benchmarks: a survey and empirical demonstration. Data Min Knowl Disc 7(4):349–371

Keogh E, Lin J, Fu A (2005) Hot sax: efficiently finding the most unusual time series subsequence. In: Proceedings of the 5th IEEE international conference on data mining, IEEE, Houston, TX, USA, pp 226–233

Lines J, Davis LM, Hills J, Bagnall A (2012) A shapelet transform for time series classification. In: Proceedings of the 18th ACM SIGKDD international conference on knowledge discovery and data mining, ACM, Beijing, China, pp 289–297

Ma J, Perkins S (2003) Time-series novelty detection using one-class support vector machines. In: Proceedings of the 9th international joint conference on neural networks, IEEE, Portland, OR, USA, vol 3, pp 1741–1745

Mahoney MV, Chan PK (2005) Trajectory boundary modeling of time series for anomaly detection. In: Proceedings of the KDD workshop on data mining methods for anomaly detection, KDD, Las Vegas, USA

Pateiro-López B, Rodríguez-Casal A (2010) Generalizing the convex hull of a sample: the R package alphahull. J Stat Softw 34(5):1–28

Protopapas P, Giammarco JM, Faccioli L, Struble MF, Dave R, Alcock C (2006) Finding outlier light curves in catalogues of periodic variable stars. Mon Not R Astron Soc 369(2):677–696

Rebbapragada U, Protopapas P, Brodley CE, Alcock C (2009) Finding anomalous periodic time series. Mach Learn 74(3):281–313

Salvador S, Chan P (2005) Learning states and rules for detecting anomalies in time series. Appl Intel 23(3):241–255

Schölkopf B, Smola AJ, Williamson RC, Bartlett PL (2000) New support vector algorithms. Neural Comput 12(5):1207–1245

Tax DM, Duin RP (2004) Support vector data description. Mach Learn 54(1):45–66

Thomas A, Clemencon S, Feuillard V, Gramfort A (2016) Learning hyperparameters for unsupervised anomaly detection. In: Proceedings of the 2016 international conference on machine learning workshop on anomaly detection, New York, USA

Wei L, Keogh E, Xi X (2006) SAXually explicit images: finding unusual shapes. In: Proceedings of the 2006 sixth IEEE international conference on data mining, IEEE, Hong Kong, China, pp 711–720

Ye L, Keogh E (2009) Time series shapelets: a new primitive for data mining. In: Proceedings of the 15th ACM SIGKDD international conference on knowledge discovery and data mining, ACM, Paris, France, pp 947–956

Zhong S, Khoshgoftaar TM, Seliya N (2007) Clustering-based network intrusion detection. Int J Reliab Qual Saf Eng 14(02):169–187