RTL: A Robust Time Series Labeling Algorithm

Frederique van Leeuwen[1,2](✉), Bas Bosma[3], Arjan van den Born[1,2], and Eric Postma[1,2]

[1] Jheronimus Academy of Data Science, 's-Hertogenbosch, The Netherlands
f.c.a.v.leeuwen@JADS.nl
[2] Tilburg University, Tilburg, The Netherlands
[3] VU Amsterdam, Amsterdam, The Netherlands

Abstract. Time series classification is one of the most important problems in data mining. With the growth in availability of time series data, many novel classification algorithms have been proposed. Despite the promising progress in accuracy, the performance of many algorithms still strongly depends on an initial training session containing labeled examples of all classes to be learned. In most realistic applications, however, labels are lacking or only partially available; limiting the practical applicability of time series classification algorithms with this requirement. To remedy this, we introduce the Robust Time series Labeling (RTL) algorithm and show its ability to increase labeling accuracy and robustness across a wide variety of time series datasets. Given its flexibility, the RTL algorithm can successfully be applied in many real-life situations.

Keywords: Time Series · Labeling · Classification

1 Introduction

Time Series Classification (TSC) is arguably one of the most interesting, common, and challenging problems in data mining. With the growth in availability of time series data, many novel TSC algorithms have been proposed, increasing the accuracy of classification significantly. Although this progress is promising, the performance of many algorithms still strongly depends on an initial training session containing labeled examples of all classes (or concepts) to be learned. In most real-life situations, however, this is unrealistic; severely limiting the practical applicability of these TSC algorithms.

To deal with situations without prior understanding of the concepts involved, TSC algorithms need to interactively invoke human expertise. While humans have an innate ability to extract meaningful knowledge from the shape of time series, this remains a complex problem for computers [3]. Unlike humans, TSC algorithms are not able to understand the context of the time series. However, in ever more real-life situations data is generated at such high rates that unassisted labeling by experts is no longer feasible. To be applicable in practice, TSC

P. H. Abreu et al. (Eds.): IDA 2021, LNCS 12695, pp. 414–425, 2021.
https://doi.org/10.1007/978-3-030-74251-5_33

algorithms thus need to support the expert in labeling efficiently and effectively. In other words, the *relative labeling effort* – which is defined as the fraction of machine- and human-based labeling relative to complete human labeling of a time series—should be minimized.

In this paper we introduce the Robust Time series Labeling (RTL) algorithm, that aims to minimize the labeling effort without affecting the overall quality of TSC tasks. The contribution of the paper is fourfold: (1) through an extensive review of an existing time series labeling algorithm we identify the key features needed to have a robust method for reducing the labeling effort while maintaining high quality labels, (2) we present the RTL algorithm that is based on these key features, (3) as part of the RTL algorithm we introduce a novel *zoom-in step* that secures the quality of the assigned labels, and (4) we show RTL's ability to increase the efficiency and robustness of the labeling procedure across a wide variety of time series datasets. RTL clears the path for TSC tasks in many application domains where large amounts of complex data are generated and need to be labeled continuously.

The outline of the remainder of the paper is as follows. In Sect. 2 we review related work. Then, in Sect. 3 we review the Like-Behaviors Labeling Routine (LBLR) introduced by Madrid et al. [9] and discuss possible improvements, which we use as starting point for the introduction of the RTL algorithm in Sect. 4. In Sects. 5 and 6 we evaluate and discuss the RTL algorithm and finally, we conclude and mention future research directions in Sect. 7.

2 Related Work

While the detection and classification of (anomalous) sub-sequences has been researched extensively (see e.g. [3]), there is almost no research dedicated to time series labeling. As far as we are aware, there are four papers on time series labeling [1,9,12,13].

Peng et al. [12] proposed the Active Learning for Time Series (ACTS) algorithm and showed that it achieved a higher classification accuracy than traditional active learning methods. Unlike our focus, theirs was on supervised active learning. Moreover, their method is built on the assumption that all classes to be learned are known before the learning procedure starts and that the time series data is perfectly arranged into segments having equal length and approximate alignment. In practice, this is seldom the case [1]. While Souza et al. [13] focus on unsupervised active learning, their experiments, like those of Peng et al. [12], were based on the perfectly segmented data from the UCR TSC Archive [2]. As stressed before, such perfectly segmented data is quite unlike the data encountered in real-life situations. In fact, segmentation is part of the challenge to find proper labels [6].

Chen et al. [1] and Madrid et al. [9] proposed the use of *motif discovery* for time series labeling. While computers are not able to understand the context of the time series, they are very proficient in motif discovery. A motif refers to pairs of sub-sequences of one or more time series that are highly similar[11] and may therefore reflect a common underlying cause. By combining the ability of algorithms to detect motifs with the ability of experts to label these motifs, the efficiency and effectiveness of labeling may increase significantly. Chen et al. [1] proposed an algorithm in which an agent examines an unbounded stream of data and occasionally asks a teacher – human or algorithm – for a label. Compared to Peng et al. [12], they do consider the case where no prior knowledge about the concepts to be learned is available. Likewise, Madrid et al. [9] proposed the LBLR algorithm to label an entire time series dataset with minimum human effort. Both algorithms, however, require the lengths of the motifs as input. Whilst the LBLR is closest to the aim of our RTL algorithm to deal with realistic unsegmented and unlabeled data, the empirical realities of different length motifs and a lack of a priori knowledge on what is to be discovered [4,5,10], suggest room for further improvement. Before introducing our RTL algorithm, we review the LBLR [9] and suggest improvements to it.

3 Requirements for a Robust Time Series Labeling Method

In this section we identify the strengths and potential shortcomings of the LBLR. To quantify the effects of these potential shortcomings, we introduce a variant called LBLR'. Subsequently, we define the two measures used for evaluating LBLR's performance. Based on the results obtained through a diverse set of experiments, we conclude with the key features needed to have a robust method for reducing labeling effort while maintaining high quality labels.

The LBLR increases the labeling efficiency of TSC by using the Matrix Profile (MP) and Minimum Description Length (MDL). The former is a method to perform motif discovery efficiently (see [14] for a detailed explanation), the latter is an information-theoretic measure to determine which motifs carry the same information content. When a set of sub-sequences is similar in terms of MDL (referred to as "semantically similar"), the whole set is assigned a single label by the user. However, during this procedure, the LBLR allows for an *extra user interaction* in which the user can add or remove sub-sequences from the elements in this set. This enables the user to remove misclassifications during the labeling procedure, which assumes that the user is able to do so. While such user-based removal could improve the quality of the assigned labels, it undeniably increases the labeling effort.

The use of motif discovery techniques helps to decrease the labeling effort. However, it also may negatively affect the quality of the labels. In the LBLR, labeled motifs are assumed to be all of a certain fixed length l. Not surprisingly, though, motifs can vary considerably in length. Assuming otherwise thus may affect the operation of LBLR during two phases: the detection of motifs and the so-called *cleanup phase*. The latter refers to the action in which all unlabeled sub-sequences of length less than l get assigned their neighboring labels automatically. Put differently, the cleanup phase labels heuristically, without any use of machine or human expertise.

To measure the effect of the extra user interaction, we compare the performance of the original LBLR with LBLR'. In the latter, all the selected motifs, misclassifications or not, receive the same label given by the user. This provides insight into the actual labeling effort of the expert. To assess how the use of fixed-length motifs affects the performance of LBLR, we vary the predefined motif lengths. To evaluate the performance of LBLR and LBLR', two measures are used: the relative labeling effort and labeling accuracy.

– **Relative Labeling Effort (RLE)** is defined as the fraction of machine- and human-based labeling to complete human labeling of a time series. Given a time series of length n and fixed motif length l, the maximum number of Labeling Rounds is defined as $LR_{max} = \frac{n}{l}$. Automatically detecting and labeling semantically similar motifs gives rise to a reduced number of labeling rounds LR. The Relative Labeling Effort is defined as $RLE = \frac{LR}{LR_{max}}$.
– **Labeling Accuracy (LA)** is defined as the percentage of correctly labeled instances and is therefore directly related to the label quality. It is important to remark that we can only determine accuracy, because we select labeled datasets for evaluation. As stated in the introduction, in most realistic applications, labels are lacking or only partially available.

In the following we will determine how fixed length motifs and the extra user interaction affect the labeling performance of LBLR. To do this, we use the following six different pre-labeled time series datasets.

1. **ACP1:** Entomology dataset. Data from an Asian Citrus Psyllid (snippet 1).
2. **ACP5:** Entomology dataset. Data from an Asian Citrus Psyllid (snippet 5).
3. **EER:** Epilepsy dataset representing distinct epilepsy episodes.
4. **ECG:** Electrocardiogram dataset. Each series traces the electrical activity recorded during one heartbeat.
5. **SLC:** Part of the StarLight Curves dataset.
6. **HCS:** A Hydraulic Control System dataset.

The datasets *ACP1*, *ACP5*, and *EER* have been used by Madrid et al. to evaluate their algorithm.[1] Datasets *ECG* and *SLC* are from the UCR Archive [2], and dataset *HCS* is a real-world industrial dataset collected by the authors.[2] These datasets were selected to cover as broad a range of characteristics as possible. Labeling each dataset with its numerical characteristics as a sequence of numbers, i.e., {number of classes, number of class transitions, length of the time series}, yields: $ACP1_{\{2,1,5203\}}$, $ACP5_{\{2,1,13126\}}$, $EER_{\{3,2,2734\}}$, $ECG_{\{2,41,9600\}}$, $SLC_{\{3,11,20480\}}$ and $HCS_{\{2,15,25000\}}$ (cf. Fig. 1). This diversity enables us to assess the performance of LBLR across different domains and situations.

Table 1 contains six sub-tables representing the performances of LBLR and LBLR' for each of the six datasets and four motif lengths. We make four observations from these results. First, rejecting the **extra user interaction**

[1] The datasets are downloaded from the web-page of the authors (www.cs.ucr.edu/~fmadr002/LBLR.html), where the names of some datasets differ from the original naming convention.

[2] Due to the nature of the *ECG* and *SLC* datasets, we needed to concatenate the separate sequences into a single time series.

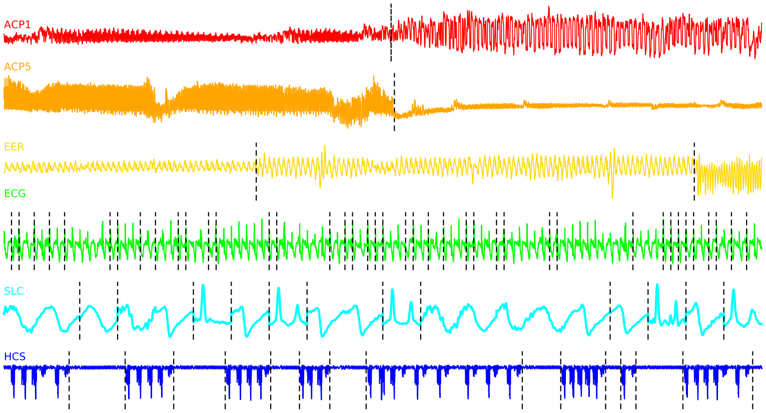


Fig. 1. The time series of the six datasets. Different time scales are depicted, due to the different time series lengths. The black vertical lines indicate the class transitions; clearly visible for the first three datasets, less so for the last three.

(LBLR vs. LBLR') results in a drop in LA that varies from zero ($APC1(l = 200)$) to huge ($EER(l = 10)$). Second, using **fixed-length motif discovery** yields a decrease in LA of LBLR for some datasets. This (negative) effect is even more substantial for the LBLR', revealing that the combined effect of abandoning both the extra user interaction and fixed-length motif discovery is large. Third, LBLR and LBLR' perform well on the datasets selected by Madrid et al., but the accuracy drops significantly on the additional datasets we tested. This drop in performance is due to the cleanup phase. As the three additional datasets include many class transitions, the chance of wrongly labeling motifs of length less than l increases. Fourth, the RLE decreases for almost all dataset-motif length combinations, and especially for LBLR'. Whilst in itself a positive effect, it only materializes in combination with a (possibly severely) reduced LA.

There are three surprising results worth mentioning. First, the LA of 34.8% for $EER(l = 20)$ and 57.4% for $HCS(l = 500)$ is lower than the *baseline accuracy*, that is obtained by assigning the majority label to the entire time series. Second, for the two pre-processed datasets, i.e. ECG and SLC, the fixed motif length ($l = 96$ and $l = 1024$, respectively) does *not* provide the best results. This even led to a substantial drop from 74.0% ($l = 100$) to 69.5% ($l = 96$) for the ECG dataset. So, even when the data is perfectly segmented and arranged into patterns of fixed length, this fixed motif length does not return the best results in terms of accuracy. Third, for $ACP1(l = 200)$ the RLE is close to 100%. This means that using the LBLR for this specific motif length is as efficient as just labeling all the sub-sequences separately.

As these results indicate, there is room for improvement with regards to: a) not being dependent on extra user interactions; and b) being able to robustly handle various situations, including varying motif lengths.

Table 1. The Labeling Accuracy LA (higher is better) and Relative Labeling Effort RLE (lower is better) per motif length l of both LBLR and LBLR'.

ACP1

	LA		**RLE**	
l	*LBLR*	*LBLR'*	*LBLR*	*LBLR'*
50	99.6	90.3	34.6	28.8
100	99.5	83.9	44.2	36.5
150	98.5	77.0	75.0	54.8
200	98.9	98.9	99.9	99.9

ACP5

	LA		**RLE**	
l	*LBLR*	*LBLR'*	*LBLR*	*LBLR'*
50	99.8	90.5	50.3	36.9
100	99.3	87.3	48.8	40.4
150	99.8	88.8	57.1	48.0
200	98.8	90.0	70.1	50.3

EER

	LA		**RLE**	
l	*LBLR*	*LBLR'*	*LBLR*	*LBLR'*
10	99.5	34.8	19.4	1.8
20	99.6	88.2	11.7	2.2
40	99.4	72.7	21.9	8.7
50	98.1	80.7	21.9	11.0

ECG

	LA		**RLE**	
l	*LBLR*	*LBLR'*	*LBLR*	*LBLR'*
50	85.9	70.0	34.4	1.6
96	85.3	69.5	25.0	3.0
100	79.6	74.0	20.8	3.1
200	74.7	69.1	20.8	6.2

SLC

	LA		**RLE**	
l	*LBLR*	*LBLR'*	*LBLR*	*LBLR'*
500	79.1	75.6	34.2	7.3
1000	78.8	68.3	29.3	9.8
1024	78.9	68.4	30.0	10.0
1500	63.6	65.0	43.9	14.6

HCS

	LA		**RLE**	
l	*LBLR*	*LBLR'*	*LBLR*	*LBLR'*
100	95.9	80.1	27.2	18.4
200	93.9	87.1	33.6	26.4
250	91.0	82.7	34.0	35.0
500	81.9	57.4	24.0	28.0

4 Robust Time Series Labeling

In this section we introduce a novel algorithm to efficiently and effectively label time series. It is based on a motif discovery method that groups motifs, and allows experts to label them efficiently. Motifs of variable length are considered and high labeling accuracy is maintained by the introduction of a *zooming in* function for the detected motifs.

Introducing some notation, we define a time series T as an ordered sequence of n real-valued numbers, often measured at fixed intervals. Given $T = \{t_1, \ldots, t_n\}$, we want to detect motifs of different lengths l for labeling. The steps included in the RTL algorithm are summarized in Algorithm 1.[3]

As an initial pre-processing step to raise efficiency, RTL discretizes $\boldsymbol{T}$ into a symbolic representation $\boldsymbol{D}$ (line 1). The details of this step are provided in Sect. 4.1 below. The procedure of finding motifs for labeling starts on line 2 and continues until an automatic stopping criterion is met (cf. Sect. 4.3). On line 3, variable length motifs are detected within an automatically updated range of lengths $\boldsymbol{L = \{MIN, MAX\}}$ and are saved in $\boldsymbol{M_L}$ (cf. Sect. 4.1). Note that

[3] The code and datasets have been made publicly available at GitHub.

Algorithm 1. The RTL algorithm

```
Require: Time Series $T$
1: D = SymbolicRepresentation(T)
2: while $T$ has unlabeled data $\wedge \neg$ algorithm(quits) do
3:     $M_L$ = FindMotifs(D*, $L$)
4:     $M'_L$ = Zoom-in($M_L$)
5:     $S$ = SelectandLabelMotif($M'_L$)
6:     Update(D*)
7: end while
8: return $S$: Labels corresponding to T
```

only those parts of D not yet labeled are considered to be candidates for motif discovery ($\boldsymbol{D*}$, line 6). To secure the quality of the assigned labels, the *zoom-in* step mentioned above is introduced (line 4). In fact, based on the (dis)similarity between all motifs which belong to a motif group, only the so-called *representative* motifs are selected and saved in $\boldsymbol{M'_L}$ (cf. Sect. 4.2). Within this set of motifs, the longest motif is selected to be labeled (line 5, cf. Sect. 4.3). These labels are saved in $\boldsymbol{S}$, which includes the label (or not-yet-labeled) information per data point n of T. Unlike LBLR, no *cleanup phase* is included. All data is labeled under supervision of the user and no labels are automatically assigned. As a consequence, LBLR returns an entire labeled dataset, while RTL returns S containing not-yet-labeled data points.

In our experiments in which we evaluate RTL and LBLR (cf. Sect. 5), we use an extra step between lines 7 and 8 in Algorithm 1. This step is purely for comparison and ensures RTL, like LBLR, returns a fully labeled dataset. In the extra step all not-yet-labeled data in S is labeled automatically based on their neighbors.

4.1 Variable-Length Motifs

To transform our time series into a symbolic representation, we use SAX (Symbolic Aggregate approXimation) [8]. The SAX algorithm has two parameters: w and a, which control the number of segments (PAA-size) and alphabet-size, respectively. Based on the extensive experiments carried out in [7], we set $a = 3$ for all experiments. The best value of w depends on the data (relatively smooth and slowly changing datasets favor a small value of w). We determine the value of w as follows: all data points n of the time series are transformed into a sequence of symbols of equal length n, for which we calculate the average number of consecutive identical symbols (q). This average is rounded to the closest integer for which it holds that $n\%q = 0$. w is then set to $w = \frac{n}{q}$. This procedure ensures that the running time of creating SAX strings remains low, and complex time series showing a lower average of consecutive identical symbols, i.e. more variability, receive a larger w.

To detect variable length motifs, we split the *string* representation of T, into so-called *words* of two symbols. For example, the string *ccaaaabb* becomes *cc aa aa bb*. Subsequently, we count the number of occurrences of (successive) word(s), e.g. $\{cc:1\},\{aa:2\}$, $\{ccaa:1\}$, $\{aaaabb:1\}$ and so on.[4] In this way, any word that occurs more than once in the string is considered to be a motif. While short words are likely to occur frequently in the string, they may be less representative of the underlying shape of the time series than longer motifs. Therefore, we start by labeling the longer – more unique – motifs and only subsequently label shorter motifs. To achieve this, we introduce $L = \{MIN, MAX\}$, with MIN being the minimum considered motif length and MAX the maximum per labeling round. Every labeling round the procedure starts with searching the longest word that appears twice (set the length of this word to MAX) and keeps searching until a word of smaller length appearing more than twice is found (set this value to MIN). In this way, a set of motifs M_L with different lengths is created in which longer motifs are prioritized over shorter ones.

4.2 Representative Motifs

Although the use of (symbolic) motif discovery techniques helps to increase the efficiency of the labeling procedure, important details within the motifs may be overlooked. More specifically, the selected motifs (Algorithm 1, line 3) may reveal significant variance when we consider more details by means of the *zoom-in* step (line 4). The zoom-in step is needed as grouped motifs may be very similar when compared to the rest of the time series, but may differ a lot when compared directly to each other. Hence, to safeguard the labeling accuracy, we need a method which is able to perceive detailed differences within the motif groups so that potential missclasifications can be removed. To achieve this, all motifs within the motif group are discretized into a small SAX string (e.g. $w = 3$: *bbb*). If these newly created strings are all the same, these so called *representative* motifs are considered to be candidates for labeling.

4.3 Motif Selection and Stopping Criteria

Every iteration round, a set M'_L of representative motifs is created (Algorithm 1, line 4). From this set, the motif group which includes most data points (motif frequency × motif length) is chosen to be labeled (line 5). This motif group is considered to be the *optimal* motif in terms of efficient labeling, as it contributes most to the goal of minimizing the labeling effort. After the motif is labeled by the human annotator, the initial discretized time series is updated (D*), so that in the next round the labeled data is not considered for labeling anymore. This procedure repeats until no more *representative* motifs can be found.

5 Results

We compare the RTL algorithm to the adjusted LBLR' algorithm, as they are both independent of (unquantifiable) extra user interactions. In this way, the

[4] We use the commonly used text analytics method *CountVectorizer* for this, but other methods may also work. CountVectorizer requires input words of at least size two.

actual LA and RLE of using motif discovery for efficient time series labeling is unveiled. We compare the performances of the RTL algorithm and LBLR' by plotting the LA against percentage labeled (Fig. 2) and listing the RLE per dataset (Table 2). In order to simplify the comparison and to unveil the robustness of the algorithms, we report the best and worst LA results obtained with the augmented LBLR' in Table 1.

Figure 2 shows the results for all datasets (Sect. 3) and can be interpreted as follows. The dots, interconnected by the lines, represent the percentage labeled (x-axis) and LA (y-axis) for each labeling round. The procedure starts in the upper left corner with 100% LA, i.e. correctly labeled data, and each following dot represents the LA of the (already) labeled part. The printed percentages at the end of the orange and red each curves express the LA of LBLR' after the labeling procedure was terminated. For RTL, the values at the end of the solid line represent the LA after the algorithm is terminated and thus before all left-overs are labeled automatically based on their neighboring behaviors (represented by dotted lines). As a reminder, automatic labeling is not part of the RTL algorithm and is only done so that RTL can be more easily compared to LBLR'.

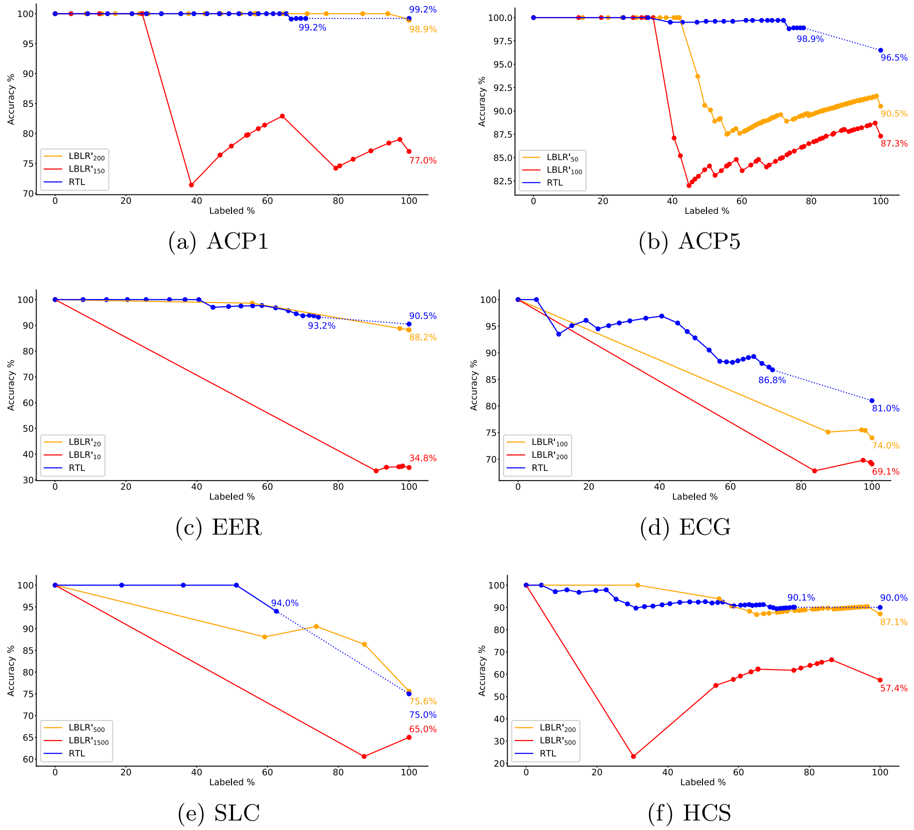
The challenge in achieving a minimum RLE with a maximum LA, lies in managing their trade-off. In Fig. 2, we can see that RTL (blue line) outperforms LBLR' (orange and red lines) with respect to the LA on all datasets, except SLC. In some cases ($ACP5$ and ECG) the improvement is substantial, in others it is relatively small ($ACP1$). However, the performance of LBLR' depends strongly on the chosen fixed motif length – as is evident from the considerable gaps between the orange and red curves – and is therefore not as robust as RTL. So, concerning LA, RTL is recommended over LBLR'.

Compared to LBLR', the RLE (Table 2) for RTL is significantly lower on datasets $ACP1$ and $ACP5$.[5] This observation, together with a higher LA, implies that on similar datasets RTL outperforms LBLR'. For the other datasets, RTL requires a higher RLE, which generally leads to a higher LA. Since the RLE is still not excessive, ranging from 8.0% to 29.6%, the higher LA may be worth the extra effort.

Table 2. Relative labeling effort %

	ACP1	ACP5	EER	ECG	SLC	HCS
RTL	20.2	8.0	13.9	25.0	9.8	29.6
LBLR'$_{best}$	99.9	36.9	2.2	3.1	7.3	26.4
LBLR'$_{worst}$	54.8	40.4	1.8	6.2	14.6	28.0

[5] As no fixed motif length l is used as input for RTL, we are actually not able to calculate the RLE as defined in Sect. 3. To still be able to compare, we used the length for which LBLR' performed best, e.g. $ACP1(l = 200)$.



(a) ACP1

(b) ACP5

(c) EER

(d) ECG

(e) SLC

(f) HCS

Fig. 2. Comparison between RTL and LBLR'. The parent curve indicates the best performing labeling algorithm concerning the obtained LA and thus label quality. The preferred performance is one with a high labeling quality and low labeling effort, corresponding to a flat curve with a minimal number of dots.

Unlike LBLR, RTL does not return a fully labeled dataset. As stated before, for the purpose of experimental evaluation, we introduced an extra step in the RTL algorithm that automatically labels unlabeled data after termination. The dotted lines in Fig. 2 show that automatically filling the gaps *after* the labeling procedure is terminated, affects the accuracy considerably for most datasets. Whereas ±75% can be labeled robustly by RTL, the so-called *rest-category* seems to require a different approach. These left-overs are unique and cannot be matched based on motif discovery and thus may include e.g. anomalies, novelties, state-changes or noise. Accordingly, we recommend to either not label this, automatically label it as rest-category or to label it manually. Although the latter increases the labeling effort, it may reveal interesting and previously unknown concepts to the user.

To summarize, being independent of fixed-length motifs and automatic labeling, benefits *LA*. Unfortunately, a higher *LA* may force a sacrifice with respect to *RLE*. But with a appreciable reduced *RLE*, RTL achieved a more robust compromise between the *RLE* and *LA* for all considered datasets. Due to this more balanced trade-off, RTL is useful for anyone who wants to efficiently and accurately label time series for TSC tasks in a wide variety of application domains.

6 Discussion

The use of symbolic motif discovery might result in overlooking important details. To remedy this, we introduced the extra *zoom-in* step (Algorithm 1, line 4). To determine the contribution of this extra step, we compared RTL to an adjusted version where line 4 in Algorithm 1 was removed. For all datasets the removal of this line led to a deterioration of *LA* by 1.4% points (*ACP5*) to 5.0% points (*ECG*). Thus, this extra step is important with respect to the labeling quality.

The performance of any algorithm depending on SAX, relies on alphabet-size a and PAA-size w. So does RTL.[6] The Matrix Profile used by LBLR is not dependent on such parameters and could be potentially used in future work. However, an extra step – such as using MDL – is needed to find all motifs which are *semantically* the same. Hence, one way or the other, some sort of discretization method is needed to find semantically similar motifs, so that they can be grouped and labeled efficiently.

7 Conclusion and Future Research

Despite the extensive research on TSC, research dedicated to time series labeling is scarce. We demonstrated that the implied user interaction and restriction of fixed motif length hamper LBLR's labeling performance. We presented RTL as an alternative and demonstrated its robustness by comparing the labeling accuracy and relative labeling effort to those of LBLR' on a variety of datasets. With an average accuracy of 93.7% and a significantly reduced labeling effort, the path for TSC tasks in practice is cleared.

The RTL algorithm can be further improved along at least three lines. First, more research should be done on the effect of the PAA-size and alphabet-size on the quality of the motifs and thus labels. Second, the use of e.g. the Euclidean Distance—instead of separate SAX strings—in the zoom-in step could be explored in future research. Finally, in some cases a high-frequency motif could be more relevant than a longer less-frequent motif. In this paper, motifs were selected based on the trade-off between three measures: the similarity, frequency and length of the motif. In other variable-length motif discovery algorithms, the

[6] After comparing different values for a and w no significant accuracy changes were obtained. However, more research is needed to fully understand the impact of both parameters on the *LA* and *RLE*.

motif is defined using only a similarity or frequency measure, often based on a threshold function. Depending on the application, the right trade-off between measures should be found.

References

1. Chen, Y., Hao, Y., Rakthanmanon, T., Zakaria, J., Hu, B., Keogh, E.: A general framework for never-ending learning from time series streams. Data Min. Knowl. Disc. **29**(6), 1622–1664 (2014). https://doi.org/10.1007/s10618-014-0388-4
2. Dau, H.A., et al.: The UCR time series classification archive, October 2018
3. Esling, P., Agon, C.: Time-series data mining. ACM Comput. Surv. (CSUR) **45**(1), 12 (2012)
4. Gao, Y., Lin, J.: Hime: discovering variable-length motifs in large-scale time series. Knowl. Inf. Syst. 1–30 (2018)
5. Gao, Y., Lin, J., Rangwala, H.: Iterative grammar-based framework for discovering variable-length time series motifs. In: 2017 IEEE International Conference on Data Mining (ICDM), pp. 111–116. IEEE (2017)
6. Hu, B., Chen, Y., Keogh, E.J.: Time series classification under more realistic assumptions. In: Proceedings of the 13th SIAM International Conference on Data Mining, Austin, Texas, USA, 2–4 May 2013, pp. 578–586 (2013). https://doi.org/10.1137/1.9781611972832.64
7. Keogh, E.J., Lin, J., Fu, A.W.: HOT SAX: efficiently finding the most unusual time series subsequence. In: Proceedings of the 5th IEEE International Conference on Data Mining (ICDM 2005), Houston, Texas, USA, 27–30 November 2005, pp. 226–233 (2005). https://doi.org/10.1109/ICDM.2005.79
8. Lin, J., Keogh, E.J., Wei, L., Lonardi, S.: Experiencing SAX: a novel symbolic representation of time series. Data Min. Knowl. Discov. **15**(2), 107–144 (2007). https://doi.org/10.1007/s10618-007-0064-z
9. Madrid, F., Singh, S., Chesnais, Q., Mauck, K., Keogh, E.: Matrix profile xvi: efficient and effective labeling of massive time series archives. In: 2019 IEEE International Conference on Data Science and Advanced Analytics (DSAA), pp. 463–472 (2019)
10. Mueen, A., Chavoshi, N.: Enumeration of time series motifs of all lengths. Knowl. Inf. Syst. **45**(1), 105–132 (2014). https://doi.org/10.1007/s10115-014-0793-4
11. Patel, P., Keogh, E., Lin, J., Lonardi, S.: Mining motifs in massive time series databases. In: 2002 IEEE International Conference on Data Mining, 2002. Proceedings, pp. 370–377. IEEE (2002)
12. Peng, F., Luo, Q., Ni, L.M.: ACTS: an active learning method for time series classification. In: 33rd IEEE International Conference on Data Engineering, ICDE 2017, San Diego, CA, USA, 19–22 April 2017, pp. 175–178 (2017)
13. Souza, V., Rossi, R.G., Batista, G.E., Rezende, S.O.: Unsupervised active learning techniques for labeling training sets: an experimental evaluation on sequential data. Intell. Data Anal. **21**(5), 1061–1095 (2017)
14. Yeh, C.C.M., et al.: Matrix profile i: all pairs similarity joins for time series: a unifying view that includes motifs, discords and shapelets. In: 2016 IEEE 16th International Conference on Data Mining (ICDM), pp. 1317–1322. IEEE (2016)