# 9

# Future Work

The work developed in this thesis shows promising results that can be improved in several ways, and it also set a fertile ground for novel promising ideas. In this section, we will discuss the possibility for improvement in each of the methods presented, as well as which novel ideas follow the currently developed strategies.

## 9.1 Overall Improvements to compute the SSM and Segmentation Process

The SSM was computed by using the standard approach found at [**muller**, **fmp**]. A set of features were extracted for that purposes using ?? [**tsfel**]. The results show promise in using this strategy for several tasks. We believe there are improvements that can be made in the feature extraction process, and contribute to a better SSM. Currently, neither a dimensionality reduction method, such as Principal Component Analysis (PCA), nor a feature selection process, are performed prior to computing the SSM. Performing these might be a way of improving the general matrix representation and remove features that do not contribute to explain the similarity between subsequences. Additionally, feature stacking has been proved to help in increasing the accuracy of feature-based classification of time series [**feature_stacking**]. This means that it should better represent differences between subsequences and should be tested on the current approach.

Another relevant improvement would be to find a way to minimize the number of parameters used for the novelty segmentation task. Currently, the size of the sliding window that extracts features and the size of the sliding kernel that computes the novelty function are independent. It would be interesting to reduce the number of variables by finding a relationship between these sizes, make the sizes adaptive based on a specific metric or perform a training process where these parameters are computed for a specific domain and kept the same for future iterations (as we showed the values of these parameters are close together for the same dataset and type of task).

The general idea when this method was used, was that changes on the signal should be represented by a change in the overall set of features. This leads to the idea that different

changes can be associated with a specific group of features. In the future, we should study which are the features that better describe a specific type of feature.

## 9.2 Unsupervised Automatic Segmentation and Labeling of Time Series

We have already mentioned in a previous chapter that the proposed methods help moving towards unsupervised and automatic segmentation and labeling of time series. We have showed that the segmentation process is well performed, but lacked the validation for automatic labeling. We believe that using the similarity profiles after a first segmentation process would provide this ability of automatically returning a completely segmented and annotated signal. In the future, we will test this approach on public datasets.

## 9.3 Hierarchical Segmentation of Time Series

When analyzing a long time series visually, the user has to zoom-in and zoom-out to search for areas of interest. We believe that using the proposed segmentation method we could perform a hierarchical segmentation, that is, apply multiple segmentation stages, with different window lengths, iteratively shorter. This would provide a multi-layered set of information that can highlight the areas of interest in different *zoom* levels.

As a pilot for this method, the user could define the number of hierarchies and corresponding window sizes to perform this process, but ideally, the process would be performed with an adaptive sliding window, that alters the dimensions based on a specific metric. This process would be helpful for long time series, with highly variable information along time.

## 9.4 Periodic Segmentation

The current approach for periodic segmentation is not able to search for the off-diagonals (paths) that are used for this purpose. We believe a better approach could be performed if the paths were highlighted. For this purpose we can perform image processing filters and then extract the resulting paths. Identifying their beginning would be the best way to find the initial sample of a period.

## 9.5 Online Unsupervised Segmentation

This work has not discussed the application of the proposed method for online purposes. We believe the method can be adapted for both novelty segmentation and periodic segmentation. Of course that if the entire SSM is computed over time with continuous incoming data, the memory required for this process would not be enough and the algorithm would

fail very quickly. For this, the method should be adapted by only keeping samples from the segmentation point forward (with a fixed buffer size), and compare the next incoming samples with the kept ones until a relevant change is identified or a new off-diagonal starts. After this, the previous samples kept on a buffer up to the segmentation point can be erased and the method can search for the next relevant change point. In the future, an online version should be developed.

## 9.6 Tool for Time Series Profiling

Currently, we introduced *TSSummarize*, which provides a summarization of the time series based on segmentation points and similarity profiles. We believe the development of an interactive tool that has an internal report on the time series, such as, statistical patterns on the segments, number of segments, how similar they are, percentage of time each segment is represented on the signal, level of periodicity, how many periods are there in each segment, presence of anomalies/discords or motifs, among other measures. In the future, this should be considered.

## 9.7 SSTS Improvements and Further Applications

The current SSTS method has several improvements to be made. Some of them have been introduced when developing HeaRTS, but others still require some thought. The fact that we are performing a symbolic representation gives the opportunity to use compression techniques typically used for text, such as the run length encoding (RLE). Having a compressed representation can make the search process faster. In combination with this, it would be interesting to include the higher level translation performed in HeaRTS, which has standard queries for standard structures (peaks, up, plateau, etc...). Using a compression of the time series can be beneficial for HeaRTS to speed up the text representation process.

A regex query is a text pattern, which is convenient to express a general pattern. However, it is sometimes difficult to generalize. The fact that the regex is not flexible makes this very brittle to patterns that we are looking for but have a slight difference with our text pattern. For now, this flexibility can be introduced with a good pre-processing/simplification of the data. In the future, a flexible search process, based on a meta-regex mechanism should be developed to perform a less brittle search.

The fact that we are searching for patterns makes it convenient to perform interactive adaptations to the data. Several methods have been thought for the *edition* of time series subsequences found with the text pattern, for instance *ssts.annotate*, *ssts.split*, *ssts.modify*, *ssts.replace*, *ssts.reverse*, *ssts.repeat*, *ssts.recursive*. Some of these functions are inspired in text edition mechanisms, others are used for general edition processes. In this, we find that having a tool that could be used to edit, search or adapt a signal would be interesting.

~~Having ways of performing a robust search of patterns enables the usage of this kind of methods.~~

We have showed that SSTS can be used in combination with existing NLP methods for classification processes and pattern search. We believe that with the current rise in NLP knowledge, a symbolic representation of time series can be useful to design novel ways of extracting information from time series. For instance, several methods are available for text topic modeling, such as Latent Semantic Analysis (LSA), Latent Dirichlet Allocation (LDA) or Non-Negative Matrix Factorization (NMF). These methods could be directly tested with the symbolic/textual representation of the time series. Other strategies, that rely in neural networks, such as *bert* and *transformers* could be explored to get ideas regarding several time series problems, namely for time series classification and generation. We believe these approaches can even be more relevant regarding interpretability and explainability. These are complex problems with time series, and having text as a medium of communication between analysts and the time series could improve current approaches on this domain. This was explored with HeaRTS, but the process was only evaluating the differences between the data and not explaining ~~why the classifier was deciding to classify the signal in that way.~~

Regarding the topic of classification with HeaRTS, we believe it should be adapted to work ~~for~~ multidimensional data. This could be made by combining several classifiers for each dimension and then combine the classification results by a standard voting method.

## 9.8 Further Developments for QuoTS

We have introduced a novel method for pattern search on time series with word-feature vectors. This approach is more expressive and provides an easy way to search for patterns. In the future, this expressiveness should be measured with a study group. A group should develop a solution for a problem without using *quots*, and then use QuoTS to solve the same problem. The average time in solving the problem would be measured to associate with the expressiveness. In the long run, this method should be considered for non-experts in time series as well, to understand what could be improved to make the process more expressive for non-experienced users.

In terms of the method itself, more keywords and operators could be introduced. Some of the keywords could even be represented by several features that contribute for the used keyword. The fact that a static window is used, should be improved. For example, SSTS can find patterns with any size, while QuoTS searches for patterns on a fixed window size. Another improvement is related with the feedback given. QuoTS could have a set of methods that help get intuition over what the keywords mean in the time series. For instance, the user could highlight a segment of the time series and the keywords with higher value would be presented.