Abstracting Local Transformer Attention for Enhancing Interpretability on Time Series Data

Leonid Schwenke[1], Martin Atzmueller[1]

[1]Osnabrück University, Semantic Information Systems (SIS) Group, Osnabrück, Germany

Abstract

Transformers have demonstrated considerable performance on sequential data, recently also towards time series data. However, enhancing their interpretability and explainability is still a major research problem, similar to other prominent deep learning approaches. In this paper, we tackle this issue specifically for time series data, where we build on our previous research regarding attention abstraction, aggregation and visualization. In particular, we combine two of our initial attention aggregation techniques and perform a detailed evaluation of this extended scope with our previously used local attention abstraction technique, demonstrating its efficacy on one synthetic as well as three real-world datasets.

Keywords

Transformer, Attention, Deep Learning, Interpretability, Time Series Analysis

1. Introduction

While Deep Learning approaches demonstrate considerable performance, e. g., on classification tasks, their interpretability and explainability in general is still limited. In addition, the modeling on complex data such as sequential data – in particular, time series data, is still a challenging and prominent area of research. Regarding *interpretability*, [1] state that "systems are interpretable if their operations can be understood by a human, either through introspection or through a produced explanation." This then also extends to *explanation*: Interpretable models are explainable per se, while explainability denotes "any action or procedure taken by a model with the intent of clarifying or detailing its internal functions" [2], as a more active characteristic for generating an explanation. We focus on making models *more* interpretable, i. e., enhancing their interpretability via specific methods. Ultimately, this then also improves on their explainability.

In previous work [3, 4] we have presented two approaches for (1) enhancing *local* interpretability of Transformers' attention, and (2) enhancing *global* interpretability by generating global coherence representations – as a form of class representation – of Transformer attention via respective abstraction methods. Both methods initially aggregate the attention inside one Transformer model state into a summarized attention matrix, using different aggregation methods. In this paper, we combine those initial attention aggregations, thus significantly extending our local approach, cf. [3]. As our main contribution, we perform a detailed experimentation and analysis of the respective integrated aggregations on univariate classification tasks for one synthetic and three real-world datasets demonstrating the efficacy of our proposed approach.

LWDA'21: Lernen, Wissen, Daten, Analysen September 01–03, 2021, Munich, Germany

EMAIL: leonid.Schwenke@uos.de (L. Schwenke); martin.atzmueller@uos.de (M. Atzmueller)

2. Related Work

This section discusses related work, and provides an overview on the Transformer architecture and the respective interpretation methods on Transformer attention. Furthermore, we summarize a symbolic abstraction technique for time series.

Transformers and Attention *Transformers* [5, 6] have emerged as a prominent Deep Learning architecture for handling sequential data [5], e. g., for natural language processing (NLP). Transformers have also recently started to be successfully applied to time series problems [7], e. g., addressing efficient architectures [6] and approaches [8] for increasing the performance of Transformers on time series prediction problems. One particularly interesting aspect of the Transformer architecture is the use of so called *attention* inside of the Multi-Head Attention (MHA), which shows how strongly an element at a specific position attends another element at any other position in the sequence. Attention is seen as the main reason for the success of Transformers, but the MHA is nevertheless not fully understood and thus is currently further actively researched.

We showed in our previous work [3] that MHA can act as some form of abstraction to locally – i. e., applied per one input [9] – reduce the input complexity. Thus and additionally based on [10, 11, 12] we argued in our more recent work [4] that MHA acts as some form of preprocessing that tries to simplify the underlying problem by highlighting the most interesting pattern of all classes of the given problem.

Analysis and Interpretation of Attention on Time Series Regarding the analysis of *attention*, most methods for MHA analysis and visualization specifically with respect to their understandability, are found in the context of Image Processing [13] and NLP, e. g., [14], where the input is already rather accessible for humans. As shown in [10], the MHA is at least partly interpretable even though multiple heads can be pruned without reducing the accuracy [15]. In addition, [12] demonstrated that it is possible to reduce words from sentences via MHA, also showing that attention can abstract important key coherences, while inputs with lower attention can be neglected for the purpose of interpretability.

In this work, we mainly focus on the local attention interpretation enhancement method on time series data by our previously proposed local abstraction approach [3]. This approach reduces the input complexity by reducing the input data to a smaller and simpler input shape, via a human-in-the-loop process, including visualisations for increased understandability of the underlying problem. We build upon this work by complementing additional information in the form of testing further parameters and providing an additional complexity quantification.

Furthermore, in [4] we presented a global interpretation technique to abstract the input into class based global coherence representations which could also be used for classification. The local and the global approach both contain an initial attention aggregation step to summarize one transformer state – i. e., the attention matrices after processing one input – but they handle it differently via different aggregation methods. Thus, we extend this step of our local approach by making use of the ones applied in the global approach, yielding a combined approach.

Time Series Patterns as Interpretation Approach One current approach to better understand and analyse time series data is based on repeating subsequences, pattern and shapes, also called *motifs* [16]. Those *motifs* can help to better understand and interpret the data because they reducing the sequence information to repetitive patterns [17, 18]. A more classification-based approach is given by *shapelets* [19], which are a special form of *motifs* that focus more on maximizing the representativity for a class [20], thus helping to interpret the classification task based on separative patterns. Those approaches are similar to both our approaches [4], in the sense of highlighting interesting class distinguishable patterns. However, our approaches focus more on Transformer attention in order to reduce the time series data to more informative and simpler data – i. e., aiming at those which the transformer uses to solve the given task.

Symbolic Abstraction – SAX The Symbolic Aggregate Approximation (SAX) is one prominent example of an aggregation and abstraction technique in the area of time series analysis, e. g., [21], also enhancing interpretability and computational sensemaking via its specific representation, cf. [22, 23]. Basically, it transforms the continuous time series, discretizing it into a symbolic string representation in order to both facilitate interpretation as well as abstraction of the time series elements, thus resulting in a high-level representation of time series data. This was previously used to improve *motif detection* in data due to the simpler data shape [17]. We also already successfully applied SAX on time series data with Transformers and showed its abstraction potential in previous work [3, 4], to which we refer for a detailed discussion.

3. Method

Below, we provide an overview of our applied model and abstraction techniques, summarizing and building on the presentation which we provided in [3]. Furthermore, we present an extended description of the analysed attention aggregations and complexity measures, including [4].

3.1. Transformer Model

We build on the Transformer model which we presented in [3], which is itself adapted from the original basic Transformer architecture [5]. Essentially, in its original form [5], a Transformer consists of an encoder and a decoder, but for classification problems only the encoder is used – as we do in this work. At its core is the MHA which uses an attention matrix to learn important points to focus on. An attention matrix essentially shows how strongly one input at a specific position attends another input at another position. These matrices are calculated and applied inside the Scaled Dot-Product, of which multiple exists inside the MHA, e. g., see Figure 1. In most cases the so-called self attention is applied, where all inputs of the MHA are the same input (i. e., V, K, and Q in Figure 1). We already made use of those properties to highlight the more important features in the time series over two human interactive approaches [3, 4]. Figure 1 shows an example of a Transformer encoder corresponding to our applied model, cf. [3]. It is important to note that we do not use a classic word embedding; we simply map each symbol to a number in the interval of $[-1, 1]$ (see [3]), because it preserves positional information (y-axis); otherwise, this would need to be approximated with an embedding, which was found to be a sub-optimal solution in our context – in contrast to e. g., NLP problems – as we discussed in [3].

Data Preprocessing. For preprocessing, we apply the same methods and parameters as we did before to enable comparability, cf. [3]. First, we scaled all data to unit variance with the Sklearn [24] standard-scaler, before all values of each time series are transformed into symbols using SAX – both fitted on the training data. We abstract to five symbols (covering five bins), i. e., to a value range of very low, low, medium, high and very high, with a uniform distribution bin calculation. The translated symbols are afterwards mapped to the interval $[-1, 1]$ while keeping the ordering information of the values of the time series accordingly as suggested in [3]. As discussed there, this method is preserving the known trend information, rather than approximating it with a word embedding.

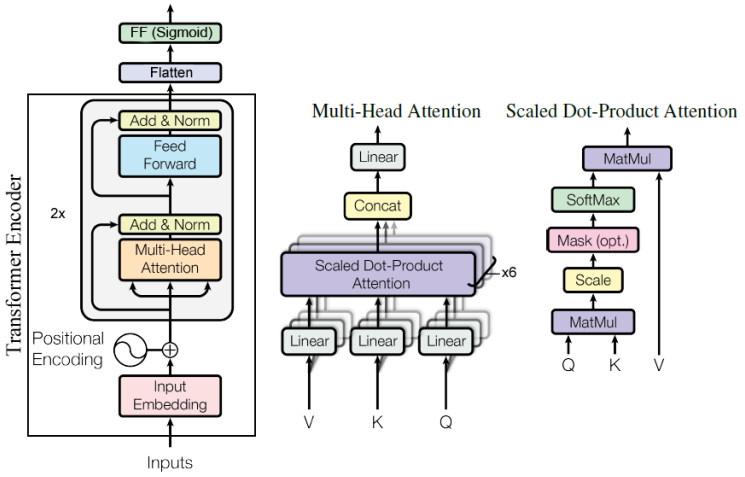

Figure 1: Our applied Transformer (encoder) architecture, adapted from the basic architecture in [5].

Model and Process As discussed above, we apply the same model as in [3] for comparability, cf. Figure 1, while extending the used datasets. We further do not optimize each model for each dataset to keep further comparability between all attention aggregation methods and datasets. A two-layered Transformer encoder is used, based on the original paper [5], with 6 heads, a head size of 6 and a dropout of 0.3, followed by a dense layer which takes in the flattened encoder output. As final output layer a sigmoid-based dense layer follows, with one neuron for each output-class. For training an Adam optimizer with 10000 warm-up steps, and for the loss function the mean squared error is used. To limit the effect of variance, the final results were determined over the average of a 5-fold cross-validation. For more detailed hyperparameters and implementation specifics we refer to our implementations provided in [3, 4]. To reduce the complexity of the time series based on abstracted attention, we build on our dynamic human-in-the-loop process to abstract the input data as presented in [3]. Figure 2 summarizes the basic process; for details we refer to [3].

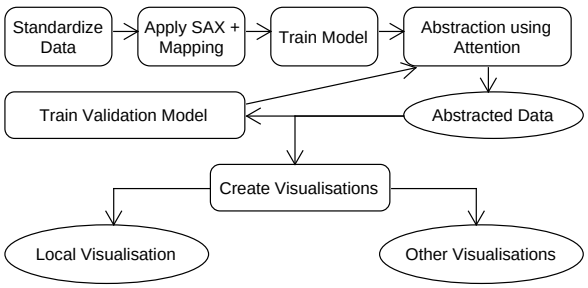

Figure 2: Process, cf. [3]: preprocessing to abstraction, validation and visualization.

Our local approach [3] should essentially enhance human interpretability of the current input, by reducing the input data into a simpler shape (i. e., reducing the data complexity), without loosing too much in terms of information and accuracy. First the data is standardized and afterwards abstracted via SAX, which is used for model training. Then, we perform data abstraction using the aggregated attention vector (the different aggregation methods are described in Section 3.2 in more detail). We map the data into three categories according to two thresholds: High-, medium- and low-attention. High attended data points are kept as is, while a sequence of medium strong attended points get reduced to the median as one central point. Low attended data points are discarded. To validate the abstraction a new model with the same parameters is trained using the abstracted data as input, where removed data is interpolated. This validates if the most important informations are still included. Essentially, this closes the cycle for the human-in-the-loop approach: the validation model is ultimately applied to refine the thresholds of the abstraction method for fine-tuning. The best threshold is approximated by manually reviewing the validation process and improving the two thresholds accordingly to maximize the accuracy to data reduction ratio. The data reduction is defined as $\frac{l_r}{l_s}$, where l_s is the input sequence length and l_r is the number of data points before the interpolation, i. e., showing in percent how much central data points are still included. In [3] we suggested to take the average attention as a first initial option, and a maximum based threshold for data with rather high attention spikes. Figure 3 shows an example of these abstraction strategies. For these two strategies, cf. [3], we consider the thresholds t_1 and t_2, where (a) for the *Average* approach we take $t_1 = \tilde{A_m}$ and $t_2 = \frac{t_1}{1.2}$; (b) for the *Max* we take the threshold $t_1 = \frac{max(A_m)}{2}$ and $t_2 = \frac{max(A_m)}{3}$. In both cases A_m is the abstracted attention vector representing the aggregated attention of the Transformer model for a specific input and $\tilde{A_m}$ is the average of A_m. We investigate a larger set of aggregations than before, which we present in more detail in Section 3.2. For making them comparable, we keep the same threshold for each individual attention aggregation run.

3.2. Aggregations

In our original approach in [3] we only tried out attention aggregations (with the maximum and average, per local input) where we first collapse the layers and afterwards the heads. In our more recent work [4], we consider the effect of multiple other aggregations (over the sum and the maximum) where we first collapse the heads and afterwards the layers. Essentially, both approaches provide different advantageous properties. To the authors' best knowledge no guideline for transformer attention aggregation exists nor is it clear how to best abstract attention matrices to extract the most information. This question is especially important because we showed in [4] that for different given classification tasks, different aggregation approaches seem to work the best. This effect was often smaller for each initial aggregation method, when aggregating the attention matrices of one input rather than the aggregation selection in the later steps needed for our global approach [4]. Thus, in this work we investigate the set of aggregations for the local process proposed in [3] in more detail, and analyse the emerging effects per aggregation. This can then further help to understand how to aggregate attention.

We investigate each combination of the sum and the maximum for each of all three of the following aggregation steps. The initial data state is an $n \times h$ matrix A consisting of $s \times s$

attention matrices $A_{(i,j)}$, where h is the number of heads per layer, n is the number of layers and s is the input length. $A_{(i,j)}$ is the attention matrix at the i-th head in the j-th layer. We define a function $f_{sn}(x)$ as any of the previously stated aggregation functions (max, sum) for the n-th step, to reduce one input dimension and to enable interchangeability.

1. Step 1 considers the collapse of the head matrices inside of A. Hence, with this we obtain $A_j = f_{s1}(\{A_{(1,j)} ..., A_{(h,j)}\})$ as the reduction of the j-th layer. Thus the output of step 1 is $A_{s1} = \{A_1, ...A_n\}$ in the form of one vector consisting of $s \times s$ attention matrices.
2. Step 2 accordingly results in $A_{s2} = f_{s2}(A_{s1})$ in the form of a $s \times s$ matrix.
3. Step 3 involves the transformation of the matrix – by reducing the columns – into one attention vector $A_m = f_{s3}(A_{s2})$ – of length s.

Additionally, we look also into all combinations when swapping the order of step 1 and step 2 above (and thus the aggregation concept) – accordingly with this the dimension we reduce first changes. The average was not considered, due to the fact that the relative scale of the data is the same as for the sum, because for each matrix entry the number of data points is the same. To differentiate the results we introduce the following abbreviations: "l" stands for *layer*, while "h" stands for the *heads* and hence "lh" would denote the collapse of first the layers, and second the heads. Attached and separated by a "-" are the collapsing method(s) i. e., *max* or *sum*. For example, "hl-max-sum-max" (abbreviated as "hl-msm") would be the maximum for step 1, the sum for step 2 and finally again the maximum for step 3, while step 1 is the collapse of the heads and step 2 the collapse on of the layers. At the end, we compare 2 options (sum, max)$^{3 \text{ steps}} \times$ 2 paths (lh or hl) $\times 2$ threshold options $\times 4$ datasets, resulting in 128 individual results.

3.3. Complexity Measures

As an extension of our work presented in [3], we include in our analysis three well known complexity metrics from information theory. This allows us to investigate in more detail, to which extent data complexity is reduced via abstraction. Our first measure is the Singular Value Decomposition Entropy (SvdEn) which measures the Shannon Entropy for the vector components which can construct the dataset [25, 26]. Thus a higher value indicates a higher data complexity. The further two measurements are the Approximate Entropy (ApEn) [27] and the Sample Entropy (SampEn) [26], which both estimate the randomness and

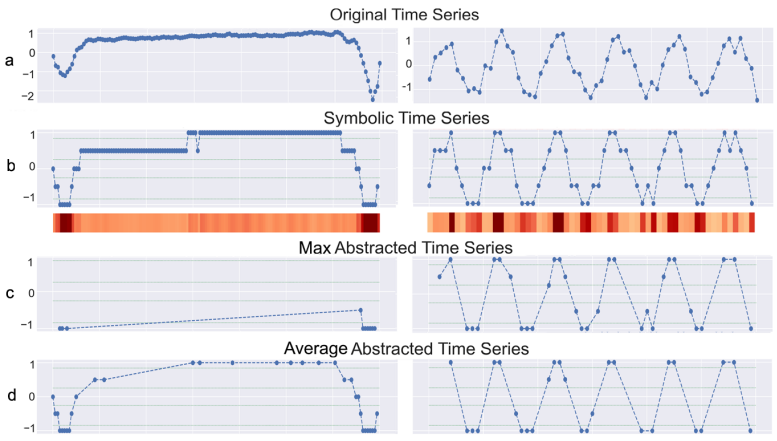

Figure 3: Example abstraction from the Synthetic dataset class 2 (right) and ECG dataset class 2 (left), cf. [3] for details.

similarity based on existing patterns. [28] describe ApEn as follows: "ApEn is a parameter that measures correlation, persistence or regularity in the sense that low ApEn values reflect that the system is very persistent, repetitive and predictive, with apparent patterns that repeat themselves throughout of the series, while high values mean independence between the data, a low number of repeated patterns and randomness". On the other hand, SampEn measures the same, but is more consistent and mostly independent from the input length while not self-counting each component vector [28]. This however has the disadvantage that SampEn can be undefined and is less reliable for small numbers [26]. Therefore we consider both of those measures for our analysis. The smaller all three measures are, the simpler and fewer are the elements/patterns which construct the time series, i. e., the data in general and the core shapes per class are easier accessible to a human; in this sense, they promote interpretability of the underlying problem in a similar way as *motifs* and *shapelets* do. For the calculation of all three metrics we used the implementation provided by the Antropy package[1].

4. Results

Below, we present the results of our analysis on the different attention aggregation methods discussed in Section 3.2; further, we discuss data complexity in proportion to the data reduction.

4.1. Datasets

To further enhance the view on our local approach [3], we investigate more attention aggregation approaches (as summarized in Section 3.2); additionally we consider the extra datasets we used in [4], i. e., applying four classification task datasets for evaluation in total. The dataset selection was limited to univariate dataset with quite small sequence lengths – due to the memory complexity of the Transformer – where the classification information is intuitively somehow contained in the "rough trend" – due to the value abstraction properties of SAX – with not to few train and test samples – due to the neural network training limitations. Further, please note that for better comparability all datasets are trained on the same model.

1. The first dataset is the Synthetic Control Chart time series dataset (**Synth**) [29, 30], with synthetic data for 6 different data trends. The train and test data both contain 300 samples; each sequence has a length of 60 and each class occurrence is balanced.
2. The second dataset is an ECG5000 dataset (**ECG**) [31, 30], which contains preprocessed ECG samples for 5 classes of length 140. The class distribution is unbalanced and the training size is 500, while the test data amounts to 4500 samples. This makes this dataset quite challenging, especially for the more infrequent classes.
3. For the third dataset, we chose a plane outline dataset (**Plane**) [30] with 7 classes and a sequence length of 144. The test and train size is 105; class occurrences are balanced. Here the classification is based on forms, which can be compared to the abstracted forms.
4. Finally, for the fourth dataset, we opted for a balanced 2 class Power Consumption dataset (**Power**) [30] which contains 180 train and test samples of length 144. It differentiates between the power consumption of a household in warm and cold months.

[1] https://github.com/raphaelvallat/antropy

Table 1
Accuracies for the **average** threshold for each aggregation and dataset

	Synth	ECG	Plane	Power
lh-sss	0.9420	0.9272	0.9429	0.9100
lh-ssm	0.9380	0.9256	0.9276	0.9111
lh-sms	0.9540	0.9264	0.7848	0.9467
lh-smm	0.9407	0.9188	0.5867	0.9144
lh-mss	0.9507	0.9257	0.7981	0.9356
lh-msm	0.9527	0.9255	0.9505	0.9333
lh-mms	0.9460	0.9220	0.8438	0.9256
lh-mmm	0.9293	0.9137	0.8724	0.9178
hl-sss	0.9447	0.9265	0.9448	0.9467
hl-ssm	0.9393	0.9282	0.9067	0.9300
hl-sms	0.9460	0.9219	0.9467	0.9278
hl-smm	0.9333	0.9221	0.8381	0.9356
hl-mss	0.9513	0.9269	0.9619	0.9411
hl-msm	0.9520	0.9243	0.9295	0.9311
hl-mms	0.9480	0.9065	0.9676	0.8844
hl-mmm	0.9367	0.9234	0.8762	0.9156

Table 2
Data reduction for the **average** threshold for each aggregation and dataset

	Synth	ECG	Plane	Power
lh-sss	0.4384	0.6208	0.5912	0.5793
lh-ssm	0.4904	0.687	0.6369	0.6281
lh-sms	0.5367	0.7018	0.6281	0.6751
lh-smm	0.5884	0.7537	0.6498	0.7133
lh-mss	0.4751	0.6450	0.5704	0.5974
lh-msm	0.5284	0.7295	0.5622	0.6608
lh-mms	0.5810	0.7405	0.6597	0.7002
lh-mmm	0.6141	0.7776	0.6886	0.7338
hl-sss	0.4384	0.6208	0.5912	0.5793
hl-ssm	0.5317	0.7014	0.6532	0.6787
hl-sms	0.4713	0.6916	0.5871	0.6237
hl-smm	0.5934	0.7464	0.6902	0.7074
hl-mss	0.4963	0.6783	0.5442	0.6259
hl-msm	0.5686	0.7497	0.6520	0.7031
hl-mms	0.5369	0.7308	0.5549	0.6704
hl-mmm	0.6141	0.7776	0.6886	0.7338

Table 3
Accuracies for the **maximum** threshold for each aggregation and dataset

	Synth	ECG	Plane	Power
lh-sss	0.9347	0.9167	0.8000	0.8367
lh-ssm	0.8887	0.9061	0.7943	0.7844
lh-sms	0.9113	0.9037	0.6990	0.7122
lh-smm	0.8633	0.9064	0.5810	0.7144
lh-mss	0.9160	0.9084	0.8019	0.8344
lh-msm	0.8787	0.9004	0.9238	0.7189
lh-mms	0.8947	0.9073	0.5486	0.7300
lh-mmm	0.8300	0.9064	0.5429	0.6400
hl-sss	0.9407	0.9137	0.9714	0.8978
hl-ssm	0.8800	0.9039	0.7048	0.7267
hl-sms	0.8913	0.9106	0.9638	0.8600
hl-smm	0.8640	0.9099	0.5276	0.6844
hl-mss	0.9273	0.9069	0.9333	0.7922
hl-msm	0.8860	0.9047	0.5752	0.7322
hl-mms	0.9053	0.9078	0.6057	0.6256
hl-mmm	0.8247	0.9083	0.5352	0.7022

Table 4
Data reduction for the **maximum** threshold for each aggregation and dataset

	Synth	ECG	Plane	Power
lh-sss	0.2133	0.6417	0.0164	0.5163
lh-ssm	0.5203	0.9201	0.4212	0.8497
lh-sms	0.5276	0.9395	0.8267	0.9338
lh-smm	0.7291	0.9508	0.8829	0.9623
lh-mss	0.3758	0.8545	0.3818	0.7899
lh-msm	0.6203	0.9383	0.6687	0.9151
lh-mms	0.6608	0.9510	0.8725	0.9503
lh-mmm	0.7866	0.9546	0.9139	0.9661
hl-sss	0.2133	0.6417	0.0164	0.5163
hl-ssm	0.5397	0.9177	0.8507	0.9281
hl-sms	0.3707	0.8585	0.3429	0.7826
hl-smm	0.6876	0.9454	0.8952	0.9518
hl-mss	0.4691	0.9156	0.4431	0.8744
hl-msm	0.7056	0.9501	0.8855	0.9598
hl-mms	0.6147	0.9381	0.6790	0.9281
hl-mmm	0.7866	0.9546	0.9139	0.9661

4.2. Aggregation Results

Tables 1 to 4 show the accuracies and data reduction per aggregation, applying our local approach [3] on the test data. For comparison, Table 8 provides the baseline accuracies, where *Original* is based on models trained with the not reduced input data from the datasets and *SAX* based on models trained with the symbolized input data – the models use exactly the same parameters as the ones with the reduced inputs. Comparing all 5 tables it can be seen that the best settings

Table 5
Ordered ranking position (per column) for the highest **accuracy** (left) and highest **data reduction** (right) of the **average** threshold per attention aggregation per dataset.

	Synth Acc.	ECG Acc.	Plane Acc.	Power Acc.	Avg Acc.	Synth Red.	ECG Red.	Plane Red.	Power Red.	Avg Red.
lh-sss	10	2	6	15	8,25	15	15	10	15	13,75
lh-ssm	13	7	8	14	10,50	12	12	8	11	10,75
lh-sms	1	5	15	1	5,50	8	9	9	8	8,50
lh-smm	11	14	16	13	13,50	4	3	7	3	4,25
lh-mss	5	6	14	4	7,25	13	14	13	14	13,50
lh-msm	2	8	3	6	4,75	10	8	14	10	10,50
lh-mms	8	12	12	10	10,50	5	6	4	6	5,25
lh-mmm	16	15	11	11	13,25	1	1	2	1	1,25
hl-sss	9	4	5	2	5,00	16	16	11	16	14,75
hl-ssm	12	1	9	8	7,50	9	10	5	7	7,75
hl-sms	7	13	4	9	8,25	14	11	12	13	12,50
hl-smm	15	11	13	5	11,00	3	5	1	4	3,25
hl-mss	4	3	2	3	3,00	11	13	16	12	13,00
hl-msm	3	9	7	7	6,50	6	4	6	5	5,25
hl-mms	6	16	1	16	9,75	7	7	15	9	9,50
hl-mmm	14	10	10	12	11,50	2	2	3	2	2,25

Table 6
Ordered ranking position (per column) for the highest **accuracy** (left) and highest **data reduction** (right) of the **maximum** threshold per attention aggregation per dataset.

	Synth Acc.	ECG Acc.	Plane Acc.	Power Acc.	Avg Acc.	Synth Red.	ECG Red.	Plane Red.	Power Red.	Avg Red.
lh-sss	2	1	6	3	3,00	15	16	15	15	15,25
lh-ssm	9	12	7	6	8,50	11	10	12	12	11,25
lh-sms	5	15	9	12	10,25	10	7	8	7	8,00
lh-smm	14	10	11	11	11,50	3	4	5	3	3,75
lh-mss	4	5	5	4	4,50	13	14	13	13	13,25
lh-msm	12	16	4	10	10,50	7	8	10	10	8,75
lh-mms	7	8	13	8	9,00	6	3	6	6	5,25
lh-mmm	15	11	14	15	13,75	1	1	1	1	1,00
hl-sss	1	2	1	1	1,25	16	15	16	16	15,75
hl-ssm	11	14	8	9	10,50	9	11	7	9	9,00
hl-sms	8	3	2	2	3,75	14	13	14	14	13,75
hl-smm	13	4	16	14	11,75	5	6	3	5	4,75
hl-mss	3	9	3	5	5,00	12	12	11	11	11,50
hl-msm	10	13	12	7	10,50	4	5	4	4	4,25
hl-mms	6	7	10	16	9,75	8	9	9	8	8,50
hl-mmm	16	6	15	13	12,50	2	2	2	2	2,00

are quite close to the baseline. When comparing the accuracies and reductions between the threshold per dataset it can be seen that the average threshold tends to perform better in terms of accuracy and the maximum threshold in terms of reduction as already stated in [3]. When selecting a setting, the accuracy to data reduction trade off needs to be considered; e. g., for the Synth dataset the worst average threshold accuracy has a larger reduction than the best

Table 7
Average ranking position for the **average-, maximum and overall** per dataset and aggregation

	Avg Rank Avg Threshold	Avg Rank Max Threshold	Avg Rank Accuracies	Avg Rank Reductions	Avg Rank Overall
lh-sss	11,00	9,13	5,63	14,5	10,06
lh-ssm	10,63	9,88	9,50	11,00	10,25
lh-sms	7,00	9,13	7,88	8,25	8,06
lh-smm	8,88	7,63	12,5	4,00	8,25
lh-mss	10,38	8,88	5,875	13,38	9,63
lh-msm	7,63	9,63	7,625	9,63	8,63
lh-mms	7,88	7,13	9,75	5,25	7,50
lh-mmm	7,30	7,38	13,5	1,13	7,31
hl-sss	9,88	8,5	3,125	15,25	9,19
hl-ssm	7,63	9,75	9,00	8,38	8,69
hl-sms	10,38	8,75	6,00	13,13	9,56
hl-smm	7,13	8,25	11,38	4,00	7,69
hl-mss	8,00	8,25	4,00	12,25	8,13
hl-msm	5,88	7,38	8,50	4,75	6,63
hl-mms	9,63	9,13	9,75	9,00	9,38
hl-mmm	6,89	7,25	12,00	2,13	7,06

performing accuracy for the maximum threshold, even though both accuracies are quite close. Thus selecting a good threshold is important, but in the following we focus more on the general performance of the aggregation strategies. In order to enable a simpler comparison of accuracies and data reduction, we ranked them in the Tables 5-6, per dataset and property in descending order. Hence, we can observe which aggregation tends to perform well on each parameter (accuracy, data reduction). It is important to note, that the ranks do not include information on the size of gaps in the ranking. For this comparison the data in the Tables 1 to 4 can provide more detailed insight. When looking at Tables 5-6 (see right) it is quite noticeable that each aggregation has a preferred relative position in the ranking independently of which threshold we look at, e. g., sum-sum-sum always has a small reduction while max-max-max has a rather large reduction. This is also – but less strongly – the case for the accuracies in Tables 5 and 6 (see left). When comparing the best average ranked aggregations for accuracy and data reduction, respectively, we observe that hl-max-sum-sum, lh-max-sum-max and hl-sum-sum-sum perform best for accuracy, while for the reductions max-max-max performs best and sum-sum-sum worst. Looking at the Pearson correlation between the ranks for accuracy and data reduction we obtain $r = -0.6199$ and for the real accuracy and reduction percentages $r = -0.4545$. This indicates that a high accuracy is not always correlated with a similar low data reduction.

Table 7 shows the average of all ranks grouped for each aggregation for each threshold, property and in total based on the Tables 5 or 6. Here, the average rank thresholds (left two columns) are the average ranks computed per aggregation (e. g., lh-sss), considering both accuracy and reduction ranks, differentiating only between the average and max threshold. The two central columns *Avg. Rank Acc.* and *Avg Rank Red.* show the average ranks

Table 8: Baseline accuracies for each dataset

	Synth	ECG	Plane	Power
Original	0.9513	0.9330	0.9619	0.7278
SAX	0.9407	0.9348	0.9810	0.9556

Table 9
Complexity measures for different data inputs on the Synth and ECG datasets

Dataset	Synth				ECG			
Entropy Meas.	**SvdEn**	**ApEn**	**SampEn**	**Reduction**	**SvdEn**	**ApEn**	**SampEn**	**Reduction**
Original	4.4015	2.3112	Inf	0	17.7833	7.7438	6.7871	0
SAX	4.4222	3.6817	5.6987	0	17.4514	7.3799	3.1481	0
max lh-msm	3.2314	1.1414	1.2173	0.7056	14.1539	1.3270	Na	0.9501
avg lh-msm	3.4219	1.5120	1.8043	0.5686	16.010	4.0247	2.4856	0.7497
max lh-mss	3.8290	2.1238	3.3277	0.4691	14.4326	1.9119	Na	0.9156
avg lh-mss	3.5402	1.7055	2.0192	0.4963	15.7437	3.9116	2.3597	0.6783
max lh-mmm	3.0328	0.8663	0.7950	0.7866	13.7738	1.2692	Na	0.9546
avg lh-mmm	3.3320	1.3603	1.5534	0.6141	16.013	3.4958	2.1704	0.7776

Table 10
Complexity measures for different data inputs on the Planes and Power datasets

Dataset	Planes				Power			
Entropy Meas.	**SvdEn**	**ApEn**	**SampEn**	**Reduction**	**SvdEn**	**ApEn**	**SampEn**	**Reduction**
Original	0.2995	0.2582	0.3294	0	0.7713	0.4533	0.2449	0
SAX	0.4512	0.3316	0.2497	0	0.7469	0.4496	0.1906	0
max lh-msm	0.2355	0.1156	Na	0.8855	0.5912	0.0893	0.0340	0.9598
avg lh-msm	0.3000	0.2344	0.1607	0.6520	0.6254	0.3236	0.1598	0.7031
max lh-mss	0.3448	0.2932	0.2210	0.4431	0.6183	0.2146	0.1041	0.8744
avg lh-mss	0.3417	0.2928	0.2309	0.5442	0.6539	0.3569	0.1721	0.6259
max lh-mmm	0.2580	0.0858	Na	0.9139	0.5897	0.0798	0.0290	0.9661
avg lh-mmm	0.2861	0.2228	0.1488	0.6886	0.6142	0.2996	0.1494	0.7338

achieved per aggregation for the average or reduction property regardless of the threshold. Furthermore, the most right column *Avg. Rank Overall* shows the average rank of one aggregation in any setting. Therefore, in Table 7 we can further observe similar rank tendencies per aggregation even on different thresholds, indicating that the aggregations provide different characteristics. Additionally multiple other model parameters could also influence the results, but this was not the focus of this work. When comparing the Power dataset in the Tables 1 and 3 we observe that choosing a good threshold is still very important to achieve a high accuracy with maximal reduction. Looking at the best average ranking (accuracy to reduction ratio) in Table 7, hl-max-sum-max performs best. In our first analysis [3] lh-sum-max-sum demonstrated the best trade-off regarding data reduction and accuracy. Here, however, it contains two outliers, one for the plane dataset in Table 5 and one for the ECG dataset in Table 6. Still we think lh-sum-max-sum is quite a good choice overall, as is hl-max-sum-max – which also relates to the strongest aggregation we observed in [4]. Nevertheless, different aggregations can fit different needs and thus no overall best aggregation of max and sum exists. Maybe an even more complex aggregation profile is needed to optimize the trade off. Also, it is interesting to note, that max results in a larger data reduction than the sum operation, as one might expect.

Considering the performance of the aggregations from our local approach [3], selecting the "best" aggregation should depend on the analysis goals. For a general strategy, we would still suggest to apply *hl-max-sum-max* due to its rather good trade-off regarding its properties. Considering the accuracy, a more defensive selection would be *hl-max-sum-sum*, because it

performed on average the second best while having not the worst data reduction. On the other hand, a rather stronger reduction while providing the second best trade off – if accuracy and data reduction would be equally important, which is normally not the case – is provided with the *hl-max-max-max* aggregation. Hence we decided – due to computational and space limitations – to only specifically consider those three aggregations for the further experimentation.

4.3. Complexity Measurement Results

Below, we present and compare the three complexity metrics discussed above, considering the three aggregations selected in the last section. Tables 9-10 show the results. Regarding SAX, it can be seen that based on the SvdEn and ApEn, the complexity increases or stays nearly the same, while only SampEn complexity is reduced by 20% at minimum. After abstracting and interpolating the data with the max or avg threshold, it can be seen that the data is always less complex than the original and the SAX data except for lh-max-sum-sum when looking at the ApEn or SvdEn. As well once the avg lh-max-sum-max is larger than the original complexity.

A Pearson correlation between data reduction to each complexity measure for each datasets shows an average $r = -0.9528 \pm 0.0282$, thus strongly suggesting that an increase in data reduction with our local approach [3] relatively reduces the data complexity. A less complex data sequence (less and simpler patterns) is easier to access for a human and thus core similarities between data from the same class can easier be concluded – similar on how *motifs* and *shapelets* improve the interpretability. Considering the interpretability from a human-centered perspective, it would be interesting to consider further metrics, also keeping in mind that the assessment can be somewhat subjective in human evaluation. Thus, a study with a questionnaire for investigating this issue is an interesting option for future work.

5. Conclusion

In this work, we investigated attention abstraction, aggregation and visualization in an adapted combined approach, building on previous work presented in [3, 4]. We investigated, how different attention aggregations perform regarding accuracy and data reduction, where we identified rather consistent results. Further, we investigated data complexity to show that data reduction via our approach enables better accessibility in terms of simplicity. For future work, we aim to further deepen the understanding of attention – e. g., on multivariate and more complex data – by further combining and refining our interpretation techniques [3, 4]. Additionally, this could be complemented by *shapelet* detection techniques to further improve interpretability and understandability of the classification task as well as the underlying model.

Acknowledgments

This work has been funded by the Interreg North-West Europe program (Interreg NWE), project Di-Plast - Digital Circular Economy for the Plastics Industry (NWE729).

References

[1] O. Biran, C. V. Cotton, Explanation and justification in machine learning : A survey, in: IJCAI 2017 Workshop on Explainable Artificial Intelligence (XAI), 2017.

[2] A. Barredo Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, F. Herrera, Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI, Information Fusion 58 (2020) 82–115.

[3] L. Schwenke, M. Atzmueller, Show me what you're looking for: Visualizing abstracted transformer attention for enhancing their local interpretability on time series data, in: Proc. FLAIRS, North Miami Beach, FL, USA, 2021, pp. 402–407.

[4] L. Schwenke, M. Atzmueller, Constructing global coherence representations: Identifying interpretability and coherences of transformer attention in time series data, in: Proc. IEEE International Conference on Data Science and Advanced Analytics (DSAA), IEEE, 2021.

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Proc. NIPS, 2017, pp. 5998–6008.

[6] Y. Tay, M. Dehghani, D. Bahri, D. Metzler, Efficient transformers: A survey, arXiv preprint arXiv:2009.06732 (2020).

[7] B. Lim, S. Ö. Arık, N. Loeff, T. Pfister, Temporal fusion transformers for interpretable multi-horizon time series forecasting, International Journal of Forecasting (2021).

[8] S. Li, X. Jin, Y. Xuan, X. Zhou, W. Chen, Y.-X. Wang, X. Yan, Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting, Advances in Neural Information Processing Systems 32 (2019) 5243–5253.

[9] D. V. Carvalho, E. M. Pereira, J. S. Cardoso, Machine learning interpretability: A survey on methods and metrics, Electronics 8 (2019) 832.

[10] J. Baan, M. ter Hoeve, M. van der Wees, A. Schuth, M. de Rijke, Understanding multi-head attention in abstractive summarization, arXiv preprint arXiv:1911.03898 (2019).

[11] H. Ramsauer, B. Schäfl, J. Lehner, P. Seidl, M. Widrich, L. Gruber, M. Holzleitner, M. Pavlović, G. K. Sandve, V. Greiff, et al., Hopfield networks is all you need, arXiv preprint arXiv:2008.02217 (2020).

[12] C. Wang, X. Liu, D. Song, Language models are open knowledge graphs, arXiv preprint arXiv:2010.11967 (2020).

[13] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al., An image is worth 16x16 words: Transformers for image recognition at scale, arXiv preprint arXiv:2010.11929 (2020).

[14] A. M. Braşoveanu, R. Andonie, Visualizing transformers for nlp: a brief survey, in: 2020 24th International Conference Information Visualisation (IV), IEEE, 2020, pp. 270–279.

[15] D. Pruthi, M. Gupta, B. Dhingra, G. Neubig, Z. C. Lipton, Learning to deceive with attention-based explanations, arXiv preprint arXiv:1909.07913 (2019).

[16] J. Lonardi, P. Patel, Finding motifs in time series, in: Proc. of the 2nd Workshop on Temporal Data Mining, 2002, pp. 53–68.

[17] P. Senin, S. Malinchik, Sax-vsm: Interpretable time series classification using sax and vector space model, in: Proc. IEEE International Conference on Data Mining (ICDM), IEEE, 2013, pp. 1175–1180.

[18] A. Mueen, E. Keogh, Q. Zhu, S. Cash, B. Westover, Exact discovery of time series motifs, in: Proceedings of the 2009 SIAM international conference on data mining, SIAM, 2009, pp. 473–484.

[19] L. Ye, E. Keogh, Time series shapelets: a novel technique that allows accurate, interpretable and fast classification, Data mining and knowledge discovery 22 (2011) 149–182.

[20] Z. Fang, P. Wang, W. Wang, Efficient learning interpretable shapelets for accurate time series classification, in: 2018 IEEE 34th International Conference on Data Engineering (ICDE), IEEE, 2018, pp. 497–508.

[21] J. Lin, E. Keogh, S. Lonardi, B. Chiu, A Symbolic Representation of Time Series, with Implications for Streaming Algorithms, in: Proc. SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, ACM, New York, NY, USA, 2003, pp. 2–11.

[22] M. Atzmueller, N. Hayat, A. Schmidt, B. Klöpper, Explanation-aware feature selection using symbolic time series abstraction: approaches and experiences in a petro-chemical production context, in: Proc. IEEE International Conference on Industrial Informatics (INDIN), IEEE, 2017, pp. 799–804.

[23] M. W. Erica Ramirez, M. Atzmueller, A Computational Framework for Interpretable Anomaly Detection and Classification of Multivariate Time Series with Application to Human Gait Data Analysis, in: Artificial Intelligence in Medicine: Knowledge Representation and Transparent and Explainable Systems, volume 11979 of *LNCS*, Springer, 2019, pp. 132–147.

[24] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, et al., Api design for machine learning software: experiences from the scikit-learn project, arXiv preprint arXiv:1309.0238 (2013).

[25] S.-y. Li, M. Yang, C.-c. Li, P. Cai, Analysis of heart rate variability based on singular value decomposition entropy, Journal of Shanghai University (English Edition) 12 (2008) 433.

[26] J. S. Richman, J. R. Moorman, Physiological time-series analysis using approximate entropy and sample entropy, American Journal of Physiology-Heart and Circulatory Physiology 278 (2000) H2039–H2049.

[27] S. M. Pincus, Approximate entropy as a measure of system complexity., Proceedings of the National Academy of Sciences 88 (1991) 2297–2301.

[28] A. Delgado-Bonal, A. Marshak, Approximate entropy and sample entropy: A comprehensive tutorial, Entropy 21 (2019) 541.

[29] R. J. Alcock, Y. Manolopoulos, et al., Time-series similarity queries employing a feature-based approach, in: 7th Hellenic conference on informatics, 1999, pp. 27–29.

[30] H. A. Dau, E. Keogh, K. Kamgar, C.-C. M. Yeh, Y. Zhu, S. Gharghabi, C. A. Ratanamahatana, Yanping, B. Hu, N. Begum, A. Bagnall, A. Mueen, G. Batista, Hexagon-ML, The ucr time series classification archive, 2018.

[31] A. L. Goldberger, L. A. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, H. E. Stanley, Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals, circulation 101 (2000) e215–e220.