Making Time Series Embeddings More Interpretable in Deep Learning: Extracting Higher-Level Features via Symbolic Approximation Representations

Leonid Schwenke
Semantic Information Systems Group
Osnabrück University
Osnabrück, Germany
leonid.schwenke@uni-osnabrueck.de

Martin Atzmueller
Semantic Information Systems Group
Osnabrück University & DFKI
Osnabrück, Germany
martin.atzmueller@uni-osnabrueck.de

Abstract

With the success of language models in deep learning, multiple new time series embeddings have been proposed. However, the interpretability of those representations is often still lacking compared to word embeddings. This paper tackles this issue, aiming to present some criteria for making time series embeddings applied in deep learning models more interpretable using higher-level features in symbolic form. For that, we investigate two different approaches for extracting symbolic approximation representations regarding the frequency and the trend information, i. e., the Symbolic Fourier Approximation (SFA) and the Symbolic Aggregate approXimation (SAX). In particular, we analyze and discuss the impact of applying the different representation approaches. Furthermore, in our experimentation, we apply a state-of-the-art Transformer model to demonstrate the efficacy of the proposed approach regarding explainability in a comprehensive evaluation using a large set of time series datasets.

Introduction

Deep Learning (DL) has demonstrated its powerful modeling options in many domains, specifically for text-based (Devlin et al. 2018; Dale 2021) as well as image-based (Islam 2022) data. However, regarding time series data non-DL models still dominate the field in many cases (Ismail Fawaz et al. 2019). For example, Hierarchical Vote Collective of Transformation-based Ensembles (HIVE-COTE) (Lines, Taylor, and Bagnall 2018) is still considered to be the state-of-the-art model for most datasets in the UCR/UEA time series repositories (Bagnall et al. 2017).

One major difference in DL modeling concerns the representation of the data, where often an embedding is introduced to better cope with internal information, e. g., word2vec (Mikolov et al. 2013). Recently, a lot of new time series embedding methods emerged, e. g., Kim, Hong, and Cha (2020); Chengyang and Qiang (2022); Cheng et al. (2020); Yue et al. (2022); Ye and Ma (2022); Tabassum, Menon, and Jastrzebska (2022); Boniol and Palpanas (2022), demonstrating that they can decrease model run time, structure information more informative, as well as improve the model performance. However, compared to words, for example, time series data is often regarded as rather complex for human interpretation. Therefore, most of the respective embeddings suffer from an interpretation problem, as indicated in Figure 1. In this paper, we focus on the comprehensibility of time series embeddings, to enhance the overall model explainability. In particular, we apply SFA (Schäfer and Högqvist 2012) and SAX (Lin et al. 2003) to extract high-level features for obtaining an interpretable symbolic representation of time series, similar to non-DL methods like e. g., Schäfer (2015). This representation is then exploited by a Transformer model (Vaswani et al. 2017). We demonstrate the efficacy of our approach in our experimental evaluation.

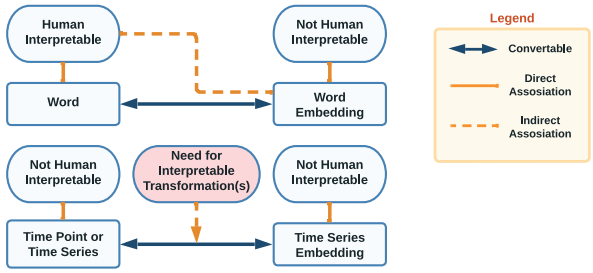

Figure 1: Visualization of the interpretability challenge for time series embeddings in DL, compared to words.

Our contributions are summarized as follows:

1. We emphasize shortcomings of current time series embeddings regarding their interpretability, and outline promising alternative options for feature extraction, e. g., SFA and SAX as symbolic representations.
2. We present SFA as a DL embedding on multiple time series classification tasks, using a Transformer model.
3. We analyze and compare the advantages and limitations of SAX and SFA in our given context, demonstrating that a DL model can benefit from multiple time series representations, similar to COTE (Bagnall et al. 2015).

The rest of the paper is organized as follows: We first provide background notions on embeddings and time series, before we discuss related work. After that, we describe our SFA-based method for enhancing interpretability on time series data. Next, we present our experimental evaluation and discuss our results. Finally, we conclude with a summary and outlook on interesting directions for future research.

Background

Modeling time series data is already quite hard to grasp due to the continuous, numeric and unintuitive nature of the data (Rojat et al. 2021). Therefore, to make the models more comprehensible, methods from the field of eXplainable Artificial Intelligence (XAI) can be applied to enable more human understandable representations. For non-DL methods, e. g., Shapelets (Ye and Keogh 2009) can be applied for feature extraction to provide shape and trend information. However, in DL, model specific XAI methods are often rather limited to saliency-like methods (Ismail Fawaz et al. 2019; Rojat et al. 2021; Xu et al. 2019). While those are not perfect, for images and text, such explanations are often more human relatable. In contrast, for time series data where the target information is often hidden or encoded in a more complex manner, such types of explanations are typically suboptimal. This is depicted in Figure 2 (A) and (B), indicating that for time series extracting information is quite limited. We further clarify this with the following examples: For word embeddings we can encode a word including to a certain extent its meaning; here, subtracting two embedding vectors can lead to results that do make sense (Allen and Hospedales 2019) – e. g., *King - Man + Woman* $\approx$ *Queen*. Additionally, semantic relations can also be extracted e. g., *Queen* and *Her*, i. e., the explanation often approximates the human understanding of semantics. Zhang and Lim (2022) analyses more human relatable features on images, matching a human process. In contrast, for time series, most of the current embeddings fail to accomplish this due to the lack of human comprehensible references in the input and embedding as shown in Figures 1 and 2 (D). In the following, we further discuss challenges when embedding time series data.

Problems of Embedding a Time Series

Time series often exhibit multiple properties which are hard to handle for humans (Rojat et al. 2021) and DL models (Shen, Wei, and Wang 2022), such as complex time relations, non-normal distributions, non-stationarity, noise/anomalies as well as having lot of redundant but highly interrelated information (Shen, Wei, and Wang 2022; Kim, Hong, and Cha 2020). Hence, visual clues can be misleading due to encoded information, e. g., frequencies. Further, due to continuity, often no perfect window size exists.

As indicated in Figure 2 (B), those properties make embedding each sample similar to word embeddings suboptimal, as a single data point on its own rarely has any meaning. However, using abstraction techniques like e. g., SAX the trend information can be extracted. Embeddings like e. g., Kim, Hong, and Cha (2020); Tonekaboni, Eytan, and Goldenberg (2021); Yue et al. (2022) encode the whole time series into one vector – sometimes based on specific features (mostly time, period and trend information). While they might include high-level information, they fail to provide insights, e. g., which time series components are relevant due to the lack of comprehensible information; as exemplified in Figure 2 (D). This is further extended when trying to relate two embeddings to each other; due to the described time series properties finding meaning is hard, compared to word embeddings where each word is one vector.

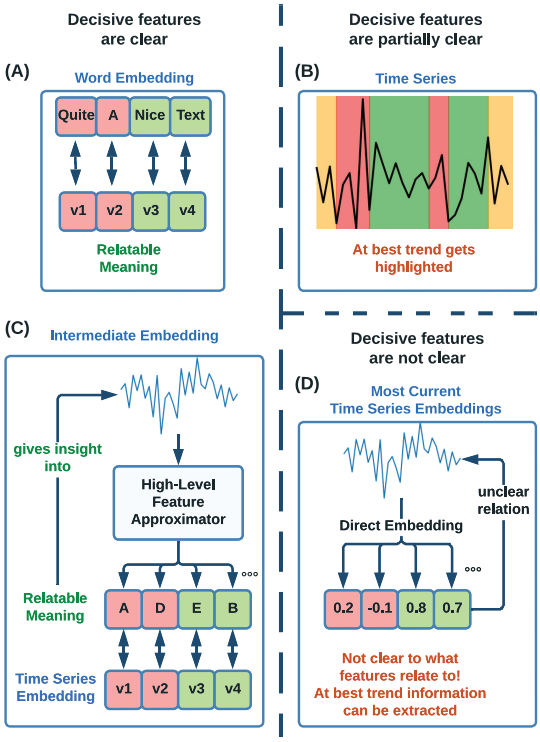

Figure 2: Explanation of four different cases regarding interpretability. (A) points out the relatability of word embeddings, where v is a vector. (B) shows how salient maps can at best provide trend information. (C) shows how our solution provides insights using meaningful features. (D) emphasizes the missing link in most current embeddings between single vector values and the time series as a whole.

Proposed Solution

Shen, Wei, and Wang (2022) and Ismail et al. (2020) point out that directly applying methods from other domains often fail to succeed on time series regarding interpretability. Thus, adaptations of those methods onto the time series domain are required. Here, leading non-DL methods can play a crucial role, where patterns can include relevant information and be put into relation to each other in order to boost performance and interpretability. They use a set of human accessible high-level features, representing e. g., the trend in order to reduce redundant information, while introducing an intermediate interpretable representation. Our proposed solution is similar: it provides an intermediate layer between embedding and data, applying discrete high-level features which together with e. g., a salient approach can enable enhanced insights into the data. Further, the association of feature and input is always clear as shown in Figure 2 (C). This simplifies adding new additional variations of representations, e. g., approximating COTE, outlining a component-based time series modeling toolkit based on different data transformations. Therefore, our focus in this work lies on outlining this toolkit to improve interpretability and clarify data extraction, rather than outperforming other embeddings.

Related Work

Using preprocessing to transform the representation and extract features is especially crucial and important for time series data (Lavangnananda and Sawasdimongkol 2012; Bagnall et al. 2015; Nalmpantis and Vrakas 2019; Chengyang and Qiang 2022); this process can be viewed as some form of embedding. Methods like e. g., SAX and SFA are very typical processing steps which are used in several time series classifiers like Shapelets (Rakthanmanon and Keogh 2013), BOSS (Schäfer 2015), WEASEL (Schäfer and Leser 2017a), MUSE (Schäfer and Leser 2017b) and SEQL (Le Nguyen et al. 2019). COTE (Bagnall et al. 2015) and its newest version HIVE-COTE 2.0 (Middlehurst et al. 2021) stress how different approaches with varying data representations are beneficial regarding the performance of general time series modeling. For neural networks on time series data, changing the representation is still done quite rarely. In contrast, in e. g., NLP, embeddings like word2vec (Mikolov et al. 2013) are core elements in most models.

Lavangnananda and Sawasdimongkol (2012) were one of the first to apply SAX in combination with a neural network, using Control Chart Patterns (CCP) data in order to highlight the general trend of the data using symbols. While such symbolic approaches on time series data are relatively seldomly applied in general, e. g., Giles, Lawrence, and Tsoi (2001); Yang et al. (2019); Elsworth and Güttel (2020); Wang et al. (2020); Criado-Ramón, Ruiz, and Pegalajar (2022), SAX as symbolic representation has recently been successfully applied again, e. g., Schwenke and Atzmueller (2021c); (2021b); Criado-Ramón, Ruiz, and Pegalajar (2022); Tabassum, Menon, and Jastrzebska (2022). To the best of the authors' knowledge, in published works SFA has never been used as representation for DL, even though it is a core element in non-DL methods and solves some typical time series challenges e. g., with fixed window sizes. In the following, we categorize recent time series embeddings into their representation type and show the lack of interpretable frequency-based approaches.

Vector Embeddings

Many recently emerged time series embeddings transform one time series into one vector, cf. Kazemi et al. (2019); Franceschi, Dieuleveut, and Jaggi (2019); Nalmpantis and Vrakas (2019); Kim, Hong, and Cha (2020); Tonekaboni, Eytan, and Goldenberg (2021); Yue et al. (2022). They mostly focus on extracting time and/or trend information in order to boost the model's performance, some use a similar approach to word2vec. However, all those approaches fail to clearly solve the problem shown in Figure 2 (D), i. e., the respective vector provides limited or no useful insights. Here, we present an overview of interesting approaches considering their interpretability: TS2Vec (Yue et al. 2022) learns a scale-invariant representation by using hierarchical contrasting and contextual consistency. With a visual heatmap, the learned representations can somewhat be related to the input, cf. Figure 2 (B). Similar in name, TS2V (Ye and Ma 2022) is a trend based embedding, which builds features using 1D-LBP. However, by mixing them all into one vector, the relation between trend and vector is identified less clearly. Similar to our proposed approach, they also emphasize the need for more time series features inside their embedding. Last but not least, Tabassum, Menon, and Jastrzebska (2022) proposed SAFE, a dictionary-based approach similar to WEASEL and BOSS. They symbolify the input via SAX and find sequences of symbols to form words and encode them into a word embedding. Due to the comprehensibility of the symbolic word encoding, trends and patterns can be interpreted via the embedding. This makes it one of the few approaches which addresses our points of criticism in the context of trend information.

Graph Representations

A different way to represent complex data and find hidden features are graphs, e. g., Gao, Small, and Kurths (2017); Interdonato et al. (2019); Bloemheuvel, van den Hoogen, and Atzmueller (2021). By transforming time series data into a graph, the interpretability can also be increased (Zheng et al. 2021; Silva et al. 2021). Also, different types of graph representations can be supplied as an input embedding. E. g., Boniol and Palpanas (2022) proposed a method which maps shape-related information into a graph, enabling the detection of single and recurrent anomalies. Another encoding emerged from Chengyang and Qiang (2022), where their proposed approach separates the input into frequency-based segments and builds a semantic encoded graph between those using a GCN. One important method for highlighting trend information is currently Time2graph (Cheng et al. 2020). By using a time-aware Shapelet-based transition graph, they model transitions between Shapelets as the general trend flow over time. Using UCR datasets, they show how Time2graph can act as a relatable embedding.

In contrast to the approaches discussed above, in this paper, we do not use graphs for modeling embeddings. Instead, we focus on a symbolic attention-based approach with the goal of presenting criteria for making time series embeddings more interpretable using higher-level features which are directly extracted from the time series.

Methods

Below, we present our applied symbolic approaches as well as the Transformer architecture in more detail.

Symbolic Aggregate ApproXimation

The Symbolic Aggregate Approximation (SAX) is one prominent example of a symbolic embedding, which abstracts/maps one data point or a sequence of fixed length inside the time series onto one of n symbolic bins (Lin et al. 2003; 2007; Rojat et al. 2021). This also enhances the interpretability and computational sensemaking by using a more human related representation, cf. Atzmueller et al. (2017); Ramirez, Wimmer, and Atzmueller (2019). SAX reduces the complexity of the continuous in $\mathbb{R}$ valued time series domain, into a discrete symbolic string representation, in order to facilitate interpretation. The resulting high-level representation of time series data highlights the general trend of the data (cf. Figure 3). The mapping can be either

done using ”uniform” or ”quantile” bins; for more information, we refer to the implementation[1]. The algorithm was previously used to improve *motif detection* in data due to the simpler data shape, as well to enhance multiple time series classifiers, e. g., Shapelets (Ye and Keogh 2009). One drawback of SAX is the loss of detailed value changes, hence selecting a valid vocabulary size is important. Therefore, SAX is mostly applicable on data which is rather trend based, e. g., contains shapes which could be extracted. Rojat et al. (2021) summarized some SAX based XAI methods for DL.

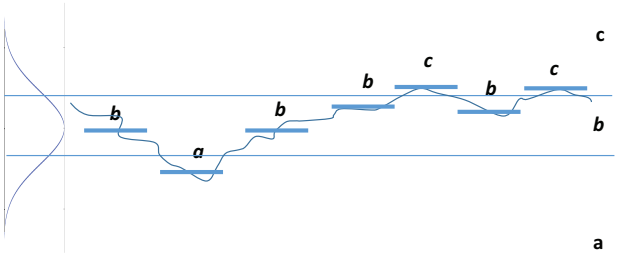

Figure 3: Example visualization for a SAX discretization, cf. (Atzmueller et al. 2017): Each data point from the original time series is mapped to a discrete symbol (a, b, c) based on the quantiles of the standard normal distribution.

Symbolic Fourier Approximation

Similar to SAX, the Symbolic Fourier Approximation (SFA) (Schäfer and Högqvist 2012) converts a numeric continuous time series into a symbolic sequence. However the SFA is frequency based, rather than trend based. This is important because a lot of information can be encoded inside the frequency of the samples rather than in a shape. For this type of data, methods like Shapelets (Ye and Keogh 2009) can struggle to find significant shapes and likewise it is harder for a human to visually grasp the important classification information. Figure 4 shows the SFA pipeline with an example. First, the time series is decomposed into base functions with different frequencies and amplitudes using the Discrete Fourier Transformation (DFT) (Winograd 1978). Using the DFT is a typical preprocessing and approximation step to apply frequency based information (Nalmpantis and Vrakas 2020). The first m Fourier-coefficients that describe the differences in the base functions are then mapped into n discrete intervals in the second step. For this step, multiple techniques can be selected to determine the intervals, e. g.,”uniform” or quantile” cf. SAX. Finally, based on the fit intervals we receive a symbolic output string, i. e., the symbols represent different Fourier coefficients for the different frequencies. By selecting only the first m Fourier coefficients, we are reducing noise from higher frequencies (Schäfer and Högqvist 2012; Nalmpantis and Vrakas 2020), as also avoiding the usage of windowing while building an informative and more comprehensible data structure; i. e., models are smaller and train faster.

[1]https://pyts.readthedocs.io/en/stable/generated/pyts.approximation.SymbolicAggregateApproximation.html

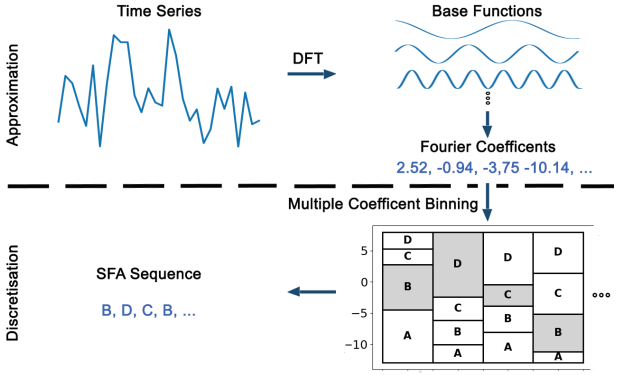

Figure 4: Example pipeline for the SFA discretization: Each Fourier coefficient is mapped onto a discrete symbol; based on and adapted from Schäfer and Högqvist (2012).

Transformer Architecture

The Transformer Architecture (Vaswani et al. 2017) is currently one of the most successful models in several application domains, e. g., mostly focusing on Natural Language Processing (NLP) and Computer Vision (CV). Recently, the Transformer has also been applied on time series data (Lim et al. 2019; Li et al. 2019; Wen et al. 2022) due to its ability to catch long-term dependencies. However, its success is limited by the memory scaling of the method with longer input sequences; this is another currently strongly ongoing research area (Tay et al. 2020). Multi-Head Attention (MHA) and Scaled Dot-Product Attention are crucial components inside the Transformer, introducing so-called Attention which can help to find relations between inputs and thus can enhance the interpretability of the model. By symbolifying the data, a more NLP-like structure emerges which can better indicate the symbol-to-symbol relations at each position; like in typical NLP XAI methods (Vig 2019; Škrlj et al. 2020) or recently emerged symbolic time series explanation methods on Transformers (Schwenke and Atzmueller 2021c; 2021b). Besides enabling referenced XAI methods, by applying SFA, we also tackle the memory problem with a fixed Fourier coefficient input size. The architecture for the Transformer Encoder is depicted in Figure 5.

Experiment

Below, we describe our applied datasets and models.

Model

For the baseline model of our experiments[2], we used a slightly adapted version from Schwenke and Atzmueller (2021c); (2021a), because they already explored the target datasets using Transformers. We however explore a slightly different range of hyperparameters, due to the complexity changes introduces by using the SFA; which we worked out via sample testing. Our experiment’s model architecture can be seen in Figure 5, including the Transformer Encoder and

[2]Code at: https://github.com/lschwenke/SFA4DL-Embedding

an example symbolic encoding. We first apply the preprocessing (SFA, SAX or no embedding) and map the symbols to [-1,1] as some sort of simple embedding, as suggested by Schwenke and Atzmueller (2021c). The symbolized data could also be used to train a word embedding, however we want to keep this experiment simple. Additionally, as argued by Schwenke and Atzmueller (2021c), with a simple mapping to [-1,1] with constant distances, we maintain the already known numeric relation between symbols and only need to learn relations in time. Afterwards we add the original position encoding from Vaswani et al. (2017) to the data. As main computational part, we apply l encoder layers afterwards. We finish with one flatten layer and an output dense layer to predict the class. To assess the effect of the SFA we train our model with the SFA embedding and compare it with the SAX embedding and with no embedding at all. As an optimizer, we used Adam with additional warm-up steps, mean-squared-error as error function and additionally an early stop with patience 70.

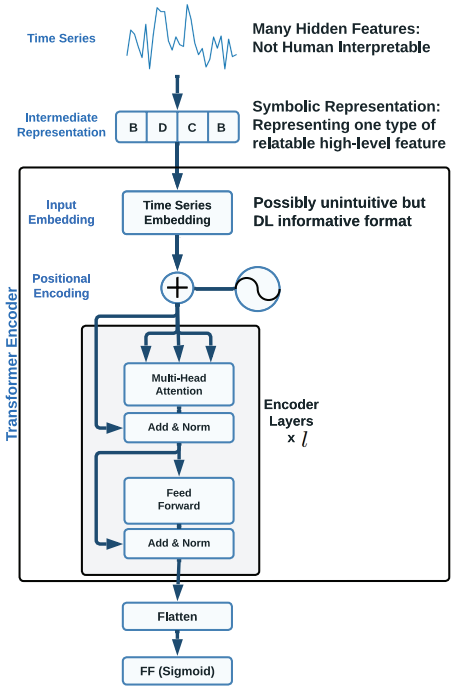

Figure 5: Our experiment's model including the original Transformer Encoder adapted from Vaswani et al. (2017). Furthermore, we show how intermediate representations can enhance interpretability, depicted at the start of the process.

Datasets

To explore the SFA methods as embedding for time series data in combination with the Transformer architecture, we used all univariate datasets from the UCR UEA time series repository (Bagnall et al. 2017) via the tslearn-toolkit (Tavenard et al. 2020). We decided only to use the univariate datasets, to better explore the information extraction abilities of the methods. Univariate data has less information to work with and thus extraction is more challenging, cf. Yue et al. (2022). Hence, we tested our method on 85 datasets from various domains.

Due to the fixed window size, after applying the SFA, we have a smaller variety in the parameter and in general simpler model, allowing us in theory to train a model for each dataset without running into memory issues. On the other side, the SAX and no preprocessing approaches could only handle 51 datasets, where however, we precariously excluded every dataset with a sample sequence size of > 500; to exclude datasets that will likely run into memory issues.

Hyperparmeters

Table 1 shows all hyperparameter assignments we explored in our experiments, resulting in 8.160 different possible configurations, with 2 different models times 5 folds, i. e., 10 trained models per experiment. All parameters below "grid search:", are analyzed using a grid search. The list options for *ncoef* means: e. g., for [128,2] that we take 128 as ncoef parameter (number of Fourier coefficients for SFA), unless the time series sequence s is smaller than 128, then we take $\frac{s}{2}$; i. e., the second parameter in the list ensures the input size does not increase. [0,0] means we use SAX and not SFA.

Table 1: Hyperparameters for the experiments

Parameter	Values
fixed:	
number of epochs	500
limit input length	500
dropout	0.3
batch size	50
patience	70
number of folds	5
grid search:	
dataset	[0, ... ,85]
symbol count	[5,6,7]
ncoefs	[[256,1],[128,2],[64,4],[0,0]]
encoder layers l	[2,4]
MHA headers h	[8,16]
[dmodel, dff]	[[16,8],[8,4]]
bin method	['quantile,'uniform']

Results

Due to memory limitations some model configurations did not finish. To make a fair comparison, we only consider 51 univariate datasets, where each type of input resulted in at least one finished model configuration. It is important to note, that due to the window reduction via SFA, the SFA input has fewer memory problems and basically finished on all datasets – with a few exceptions, where the quantile SFA ran into some transformation issues. However, this happened only on datasets where the model failed for the other input types anyway, so we did not look further into this issue.

In the following, we present and discuss our results between all three types of inputs.

Table 2: Accuracy overall and w.r.t. baseline model, for which we only consider the best performing model configurations.

Best Configs	Accuracy	Avg. improvement w.r.t. baseline	Avg. decline w.r.t. baseline	Performed best	Performed better than baseline
Baseline	0.6664 ± 0.1670	-	-	22	-
SAX all	0.6804 ± 0.1677	0.0992 ± 0.0969	0.0410 ± 0.0595	9	24
SFA all	0.6823 ± 0.1754	0.1113 ± 0.1265	0.0758 ± 0.0799	19	27
SAX or SFA	$\mathbf{0.7191 \pm 0.1554}$	0.1134 ± 0.1227	0.0274 ± 0.0279	29	29
Univariate					
SAX	0.6556 ± 0.1794	0.1365 ± 0.1145	0.0554 ± 0.0748	10	16
SFA	0.6745 ± 0.1768	0.1431 ± 0.1341	0.0772 ± 0.0799	17	23
SAX or SFA	$\mathbf{0.7111 \pm 0.1598}$	0.1432 ± 0.1330	0.0331 ± 0.0418	27	27
Quantile					
SAX	0.5952 ± 0.1925	0.0459 ± 0.0709	0.0855 ± 0.0953	4	11
SFA	0.6629 ± 0.1822	0.1317 ± 0.1400	0.0767 ± 0.0784	20	21
SAX or SFA	$\mathbf{0.6796 \pm 0.1705}$	0.1145 ± 0.1344	0.0566 ± 0.0629	25	25

Performance

Table 2 compares the performance of the models. We always only consider the best performing model configuration per dataset, due to not all configurations finishing and because data augmentation can easily lead to different optimal model parameter (Li et al. 2021), e. g., the model can fail to converge. As one can see in the upper part of the table, the average performance on the datasets improved with SAX and SFA. On datasets where SAX or SFA resulted in an improvement (acc. $>$ baseline), the average increase was 0.0992 for SAX or respectively 0.1113 for SFA. This indicates that the selected representation can make a huge impact for a given dataset. SAX improved the baseline (i. e., no preprocessing) performance on 24 and SFA 27 times out of 51 datasets, i. e., roughly 50%. On 9 times, SAX or respectively 19 times, SFA performed best, showing that SFA holds multiple advantages over SAX. However, when taking the advantages of SAX and SFA into combination (by selecting the best model per dataset) the performance increases even further, i. e., trend and frequency information can complement each other. Further, in this combination the average performance decline (acc. $<$ baseline) was reduced to 0.0331, compared to the individual declines. Meaning, even in bad cases, often either SFA or SAX performed quite similar to the baseline.

Model Parameter Analysis

Looking at the bottom part of Table 2, it is quite clear that the "quantile" option for SFA and SAX performs only rarely (2 times) better than the "uniform" option. Therefore, "uniform" seems to be the better initial parameter to test out. We further analyzed the parameter which lead to the better uniform SFA and SAX models. We found a strong tendency ($> 70\%$) towards using 2 layers for the SFA and SAX models; which can not be seen in the baseline models and thus could be explained due to the simplification over symbols. Additionally, a dmodel of 16 was the most successful in over 70% of all models, indicating a possible potential to improve the models. Further, for SAX, more symbols and more headers lead more often to better models, e. g., 7 symbols were in over 50% and 16 headers in over 63% of the models the most successful. For SFA this effect could not be observed.

Discussion

In our experiments, we focused on the impact of abstractions and different representational forms in comparison to our baseline model. While SAX and SFA do not add value on all datasets, our experiments certainly stress out the importance of representations and that extracting discrete information for different properties of the time series can have a positive effect on the performance and interpretability of DL models. Due to the structure of the matrix multiplication inside the MHA it is important to select comprehensible elements as part of the sequence, to enable NLP XAI methods like e. g., Vig (2019); Škrlj et al. (2020), in contrast to what most currently embeddings do, cf. Figure 2 (D). While we did not use learned word embeddings to map our symbols – for simplicity and due to the often sparse training data – by extracting general high-level properties of time series, we hypothesize that it could be possible to build a component-based time series embedding toolkit. Here, reusable embedding components could model and describe different categories/classes of time series, similar to different languages applied in NLP.

Conclusions

In this paper, we discussed several limitations of prominent time series embeddings in terms of interpretability, which can occur when blindly applying approaches from other domains. We introduced some criteria for embeddings to make them more interpretable. Our experiments showed, that SFA demonstrates considerable potential w.r.t. time series tasks: It fixates the input window, can extract possible frequency-based information in order to improve the results and also introduces an interpretable symbolic aspect. Further, we showed that SAX can complement SFA, i. e., trend and frequency features can provide additional information for the task at hand. This motivates the application of frequency-based approaches, compared to current rather trend focused ones. Because methods like COTE show the benefits of multiple different representations, for future work, we aim to investigate further time series representations and methods, e. g., similar to WEASLE (Schäfer and Leser 2017a) – towards, ultimately, an interpretable embedding toolkit with a variety of reusable feature components.

References

Allen, C., and Hospedales, T. 2019. Analogies explained: Towards understanding word embeddings. In *Proc. International Conference on Machine Learning (ICML)*, 223–231. Long Beach, California, USA: PMLR.

Atzmueller, M.; Hayat, N.; Schmidt, A.; and Klöpper, B. 2017. 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)*, 799–804. IEEE.

Bagnall, A.; Lines, J.; Hills, J.; and Bostrom, A. 2015. Time-series classification with COTE: the collective of transformation-based ensembles. *IEEE Transactions on Knowledge and Data Engineering* 27(9):2522–2535.

Bagnall, A.; Lines, J.; Bostrom, A.; Large, J.; and Keogh, E. 2017. The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data mining and knowledge discovery* 31:606–660.

Bloemheuvel, S.; van den Hoogen, J.; and Atzmueller, M. 2021. A computational framework for modeling complex sensor network data using graph signal processing and graph neural networks in structural health monitoring. *Applied Network Science* 6(1):1–24.

Boniol, P., and Palpanas, T. 2022. Series2Graph: Graph-based subsequence anomaly detection for time series. *arXiv preprint arXiv:2207.12208*.

Cheng, Z.; Yang, Y.; Wang, W.; Hu, W.; Zhuang, Y.; and Song, G. 2020. Time2Graph: Revisiting time series modeling with dynamic shapelets. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 3617–3624.

Chengyang, Y., and Qiang, M. 2022. Representation learning of time series data with high-level semantic. *DEIM22*.

Criado-Ramón, D.; Ruiz, L.; and Pegalajar, M. 2022. Electric demand forecasting with neural networks and symbolic time series representations. *Applied Soft Computing* 122:108871.

Dale, R. 2021. Gpt-3: What's it good for? *Natural Language Engineering* 27(1):113–118.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Elsworth, S., and Güttel, S. 2020. Time series forecasting using LSTM networks: A symbolic approach. *arXiv preprint arXiv:2003.05672*.

Franceschi, J.-Y.; Dieuleveut, A.; and Jaggi, M. 2019. Unsupervised scalable representation learning for multivariate time series. *Advances in neural information processing systems* 32.

Gao, Z.-K.; Small, M.; and Kurths, J. 2017. Complex network analysis of time series. *Europhysics Letters* 116(5):50001.

Giles, C. L.; Lawrence, S.; and Tsoi, A. C. 2001. Noisy time series prediction using recurrent neural networks and grammatical inference. *Machine learning* 44(1):161–183.

Interdonato, R.; Atzmueller, M.; Gaito, S.; Kanawati, R.; Largeron, C.; and Sala, A. 2019. Feature-Rich Networks: Going Beyond Complex Network Topologies. *Applied Network Science* 4(4).

Islam, K. 2022. Recent advances in vision transformer: A survey and outlook of recent work. *arXiv preprint arXiv:2203.01536*.

Ismail, A. A.; Gunady, M.; Corrada Bravo, H.; and Feizi, S. 2020. Benchmarking deep learning interpretability in time series predictions. *Advances in neural information processing systems* 33:6441–6452.

Ismail Fawaz, H.; Forestier, G.; Weber, J.; Idoumghar, L.; and Muller, P.-A. 2019. Deep learning for time series classification: a review. *Data mining and knowledge discovery* 33(4):917–963.

Kazemi, S. M.; Goel, R.; Eghbali, S.; Ramanan, J.; Sahota, J.; Thakur, S.; Wu, S.; Smyth, C.; Poupart, P.; and Brubaker, M. 2019. Time2Vec: Learning a vector representation of time. *arXiv preprint arXiv:1907.05321*.

Kim, H. J.; Hong, S. E.; and Cha, K. J. 2020. seq2vec: analyzing sequential data using multi-rank embedding vectors. *Electronic Commerce Research and Applications* 43:101003.

Lavangnananda, K., and Sawasdimongkol, P. 2012. Neural network classifier of time series: A case study of symbolic representation preprocessing for control chart patterns. In *Proc. International Conference on Natural Computation*, 344–349. IEEE.

Le Nguyen, T.; Gsponer, S.; Ilie, I.; O'Reilly, M.; and Ifrim, G. 2019. Interpretable time series classification using linear models and multi-resolution multi-domain symbolic representations. *Data mining and knowledge discovery* 33(4):1183–1222.

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

Li, X.; Xiong, H.; Li, X.; Wu, X.; Zhang, X.; Liu, J.; Bian, J.; and Dou, D. 2021. Interpretable deep learning: Interpretation, interpretability, trustworthiness, and beyond. *arXiv preprint arXiv:2103.10689*.

Lim, B.; Arik, S. O.; Loeff, N.; and Pfister, T. 2019. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *arXiv preprint arXiv:1912.09363*.

Lin, J.; Keogh, E.; Lonardi, S.; and Chiu, B. 2003. A symbolic representation of time series, with implications for streaming algorithms. In *Proc. 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, 2–11. New York, NY, USA: ACM.

Lin, J.; Keogh, E.; Wei, L.; and Lonardi, S. 2007. Experiencing SAX: A novel symbolic representation of time series. *Data Mining and Knowledge Discovery* 15(2):107–144.

Lines, J.; Taylor, S.; and Bagnall, A. 2018. Time series classification with HIVE-COTE: The hierarchical vote collective of transformation-based ensembles. *ACM Transactions on Knowledge Discovery from Data* 12(5).

Middlehurst, M.; Large, J.; Flynn, M.; Lines, J.; Bostrom, A.; and Bagnall, A. 2021. HIVE-COTE 2.0: a new meta ensemble for time series classification. *Machine Learning* 110(11):3211–3243.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Nalmpantis, C., and Vrakas, D. 2019. Signal2Vec: Time series embedding representation. In *Proc. International conference on engineering applications of neural networks*, 80–90. Springer.

Nalmpantis, C., and Vrakas, D. 2020. On time series representations for multi-label NILM. *Neural Computing and Applications* 32(23):17275–17290.

Rakthanmanon, T., and Keogh, E. 2013. Fast shapelets: A scalable algorithm for discovering time series shapelets. In *proceedings of the 2013 SIAM International Conference on Data Mining*, 668–676. SIAM.

Ramirez, E.; Wimmer, M.; and Atzmueller, M. 2019. 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*, 132–147. Springer.

Rojat, T.; Puget, R.; Filliat, D.; Del Ser, J.; Gelin, R.; and Díaz-Rodríguez, N. 2021. Explainable artificial intelligence (XAI) on timeseries data: A survey. *arXiv preprint arXiv:2104.00950*.

Schäfer, P., and Högqvist, M. 2012. SFA: a symbolic fourier approximation and index for similarity search in high dimensional datasets. In *Proceedings of the 15th international conference on extending database technology*, 516–527.

Schäfer, P., and Leser, U. 2017a. Fast and accurate time series classification with WEASEL. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 637–646.

Schäfer, P., and Leser, U. 2017b. Multivariate time series classification with WEASEL+ MUSE. *arXiv preprint arXiv:1711.11343*.

Schäfer, P. 2015. The BOSS is concerned with time series classification in the presence of noise. *Data Mining and Knowledge Discovery* 29(6):1505–1530.

Schwenke, L., and Atzmueller, M. 2021a. Abstracting local transformer attention for enhancing interpretability on time series data. In *Proceedings of the LWDA 2021 Workshops: FGWM, KDML, FGWI-BIA, and FGIR, Online, September 1-3, 2021*, volume 2993 of *CEUR Workshop Proceedings*, 205–218. CEUR-WS.org.

Schwenke, L., and Atzmueller, M. 2021b. 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.

Schwenke, L., and Atzmueller, M. 2021c. Show me what you're looking for: Visualizing abstracted transformer attention for enhancing their local interpretability on time series data. In *Proc. FLAIRS*, 402–407.

Shen, L.; Wei, Y.; and Wang, Y. 2022. Respecting time series properties makes deep time series forecasting perfect. *arXiv preprint arXiv:2207.10941*.

Silva, V. F.; Silva, M. E.; Ribeiro, P.; and Silva, F. 2021. Time series analysis via network science: Concepts and algorithms. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 11(3):e1404.

Škrlj, B.; Eržen, N.; Sheehan, S.; Luz, S.; Robnik-Šikonja, M.; and Pollak, S. 2020. AttViz: Online exploration of self-attention for transparent neural language modeling. *arXiv preprint arXiv:2005.05716*.

Tabassum, N.; Menon, S.; and Jastrzebska, A. 2022. Time-series classification with SAFE: Simple and fast segmented word embedding-based neural time series classifier. *Information Processing & Management* 59(5):103044.

Tavenard, R.; Faouzi, J.; Vandewiele, G.; Divo, F.; Androz, G.; Holtz, C.; Payne, M.; Yurchak, R.; Rußwurm, M.; Kolar, K.; and Woods, E. 2020. Tslearn, a machine learning toolkit for time series data. *Journal of Machine Learning Research* 21(118):1–6.

Tay, Y.; Dehghani, M.; Bahri, D.; and Metzler, D. 2020. Efficient transformers: A survey. *arXiv preprint arXiv:2009.06732*.

Tonekaboni, S.; Eytan, D.; and Goldenberg, A. 2021. Unsupervised representation learning for time series with temporal neighborhood coding. *arXiv preprint arXiv:2106.00750*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.

Vig, J. 2019. Visualizing attention in transformer-based language representation models. *arXiv preprint arXiv:1904.02679*.

Wang, Y.; Zhang, Y.; Wu, Z.; Li, H.; and Christofides, P. D. 2020. Operational trend prediction and classification for chemical processes: A novel convolutional neural network method based on symbolic hierarchical clustering. *Chemical Engineering Science* 225:115796.

Wen, Q.; Zhou, T.; Zhang, C.; Chen, W.; Ma, Z.; Yan, J.; and Sun, L. 2022. Transformers in time series: A survey. *arXiv preprint arXiv:2202.07125*.

Winograd, S. 1978. On computing the discrete fourier transform. *Mathematics of computation* 32(141):175–199.

Xu, F.; Uszkoreit, H.; Du, Y.; Fan, W.; Zhao, D.; and Zhu, J. 2019. Explainable AI: A brief survey on history, research areas, approaches and challenges. In *CCF international conference on natural language processing and Chinese computing*, 563–574. Springer.

Yang, Y.; Zheng, H.; Li, Y.; Xu, M.; and Chen, Y. 2019. A fault diagnosis scheme for rotating machinery using hierarchical symbolic analysis and convolutional neural network. *ISA transactions* 91:235–252.

Ye, L., and 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*, 947–956.

Ye, C., and Ma, Q. 2022. TS2V: A transformer-based siamese network for representation learning of univariate time-series data. In *2022 IEEE 25th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, 1245–1250. IEEE.

Yue, Z.; Wang, Y.; Duan, J.; Yang, T.; Huang, C.; Tong, Y.; and Xu, B. 2022. TS2Vec: Towards universal representation of time series. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 8980–8987.

Zhang, W., and Lim, B. Y. 2022. Towards relatable explainable AI with the perceptual process. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, 1–24.

Zheng, M.; Domanskyi, S.; Piermarocchi, C.; and Mias, G. I. 2021. Visibility graph based temporal community detection with applications in biological time series. *Scientific Reports* 11(1):1–12.