Show Me What You're Looking For: Visualizing Abstracted Transformer Attention for Enhancing Their Local Interpretability on Time Series Data

Leonid Schwenke and **Martin Atzmueller**
Osnabrück University, Institute of Computer Science
Semantic Information Systems Group
Osnabrück, Germany

Abstract

While Transformers have shown their advantages considering their learning performance, their lack of explainability and interpretability is still a major problem. This specifically relates to the processing of time series, as a specific form of complex data. In this paper, we propose an approach for visualizing abstracted information in order to enable computational sensemaking and local interpretability on the respective Transformer model. Our results demonstrate the efficacy of the proposed abstraction method and visualization, utilizing both synthetic and real world data for evaluation.

Introduction

Deep Learning approaches are becoming pervasive with their recent advances on handling complex data. One recently emerging architecture is given by Transformers, cf. (Vaswani et al. 2017), as a prominent approach for processing sequential data. For example, BERT (Devlin et al. 2018) is one of many successful state-of-the-art uses of Transformers in the context of natural language processing (NLP). While Transformers are still mostly used in NLP, more recent research applies them successfully to image processing (Dosovitskiy et al. 2020) as well as time series prediction (Li et al. 2019). The latter is also the context which we will consider in this paper, specifically relating to the important concept of *attention* which is central to Transformers, i. e., focussing on specific subsets of the data that are relevant to the task at hand (Vaswani et al. 2017).

Problem. In this paper, we focus on the general problem of computational sensemaking, e. g., (Atzmueller 2018) of Transformers on complex time series data. In particular, we focus on *local interpretability* considering abstracted time series sequences using *attention* of the Transformer which can indicate interesting data points/patterns leading to a local classification (decision). In general, this is a difficult problem; so far, for Transformers, the interpretation of the underlying attention mechanism, in particular when using *Multi-Headed Attention (MHA)*, is still not fully understood nor explainable and only seen as partially transparent (Jain and Wallace 2019; Baan et al. 2019; Clark et al. 2019).

Objectives. We aim to enhance the understanding and visualization of attention on time series data. For tackling the problem described above, we make use of two main ideas:

1. In data preprocessing we transform the time series data into symbols using Symbolic Aggregate Approximation (SAX) (Lin et al. 2003; 2007), which is often better suited for human interpretation, e. g., (Atzmueller et al. 2017).
2. We exploit Transformer attention for data abstraction, leading to data indication and according visualization.

Thus, our general objective is to increase the understandability of Transformers' MHA, in particular in the context of time series data for which to our knowledge very few works have been published. In order to increase the interpretability and for helping humans to understand the learned problem better, we abstract the input data via attention and the SAX algorithm. The results are fewer indicative data points in the time series. This abstract view can afterwards help humans to see and comprehend more easily on which indicative data points the neural network's main decisions are based on. In addition, with fewer data points and a good performing model this can be an important medium for a human in order to comprehend the general problem setting more easily.

Contributions. Our contributions are given as follows:

1. We propose a semi-automatic approach for abstracting time series data using attention and symbolic abstraction (using SAX) in order to enhance local interpretability.
2. We present according visualizations, also for supporting the human-in-the-loop in the semi-automatic process.
3. We demonstrate the efficacy of the proposed approach in an evaluation using synthetic as well as real-world data. Our experimentation indicates both the effectiveness of the abstracted representation using indicative data points, as well as visualization examples indicating the interpretability of the abstraction.

Outline. The rest of the paper is structured as follows: We first discuss related work, before we present our proposed method, including preprocessing, architecture, symbolic abstraction and result interpretation. After that, we present and discuss our results in detail using one synthetic and one real-world dataset. Finally, we conclude the paper with a summary and outline interesting directions for future work.

Preprint of: Leonid Schwenke and Martin Atzmueller (2021) Show Me What You're Looking For: Visualizing Abstracted Transformer Attention for Enhancing Their Local Interpretability on Time Series Data. Proc. 34th International FLAIRS Conference, North Miami Beach, FL, USA

Related Work

In the following, we outline related work regarding Transformers, and visualization options for making them more understandable, as well as on symbolic abstraction.

Transformers have emerged as a prominent Deep Learning architecture for handling sequential data (Vaswani et al. 2017), e. g., for NLP. Transformers have also recently started to be successfully applied to time series problems (Lim et al. 2019), also addressing efficient architectures (Tay et al. 2020) and approaches (Li et al. 2019) for increasing the performance of transformers on time series prediction. Ramsauer et al. analyzed the pattern filter ability of Transformers. They show that the first Transformer layers perform a more generic preprocessing. In contrast, the later layers are the ones still trained at the end while containing more class specific filtering. They reasoned that the MHA is encoding and storing multiple patterns for the classification. In this work, we build on this principle in order to provide such summarized patterns of interest. It is important to note, that the given limitations of transformers are currently a rather strong research topic in general; recently many slightly modified Transformer architecture arose which take on different limitations of the original Transformer (Tay et al. 2020). In comparison to those approaches, we focus on the specific problem of time series classification, also considering the MHA in its original form, to create a first baseline. Therefore we also do not focus on scalability/runtime.

Regarding **Analysis and Sensemaking of Attention**, most of the methods for MHA analysis and visualisation – in order to increase their understandability – are found in the context of Image Processing (Dosovitskiy et al. 2020) and NLP (Vig 2019), where the input is already rather accessible for humans. Attention itself was already previously used on time series data with RNNs. Serrano and Smith found though, that the interpretability properties only works suboptimal compared to other techniques. However, they did not look into the attention of Transformers itself; we tackle this – with a refined approach – in this paper.

Also, Baan et al. showed that MHA is at least partly interpretable even though multiple heads can be pruned without reducing the accuracy (Pruthi et al. 2019). Therefore, MHA can be used for transparent interpretations under certain conditions and with specific methods. Hence, we extend on this to gain a better understanding of MHA with time series data – this is also why we decided to analyze the MHA in its original form. In addition, Wang, Liu, and Song 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. We adapt this important finding for our approach in time series abstraction.

Symbolic Abstraction – SAX is one prominent example of an aggregation and abstraction technique in the area of time series analysis, (Lin et al. 2003; 2007); it results in a high-level representation of time series. A key feature of the technique is its symbolic representation by discretizing time series into symbolic strings (cf. Figure 2). To the best of the authors' knowledge, this is the first time that applying SAX in combination with Transformers has been proposed.

Methods

Process Model. Figure 1 depicts our proposed process. First, the data is preprocessed (scaling, SAX). After that, a Transformer model is trained. Next, the attention is applied for data abstraction and indication as described below. The data can now be visualized in order to enhance human understandability. Furthermore, in order to validate the data and to show that the abstracted information is relevant, we train a new transformer model for validation. In a human-in-the-loop approach, this model is ultimately applied to refine the thresholds of the abstraction method for fine-tuning.

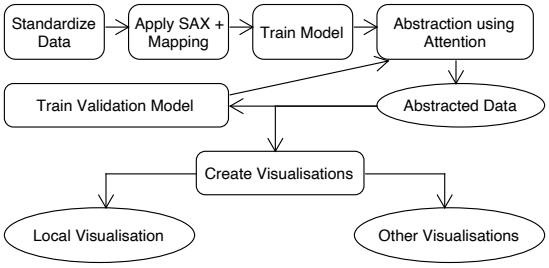

Figure 1: Pipeline of the data processing, from preprocessing the data to abstraction, validation and visualization.

It is important to note that we actually apply two data abstraction steps: (1) We apply SAX to transform continuous time series values to symbols ("symbolification" – cf. Figure 2). This already reduces the complexity for better human accessibility of the data. (2) We apply data abstraction using attention (indicating interesting data points) – see below.

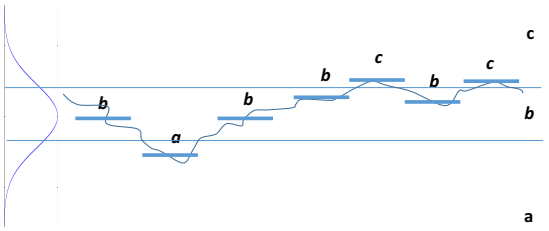

Figure 2: SAX discretization, cf. (Atzmueller et al. 2017): Each data point is mapped to a discrete symbol (a, b, c), e. g., using the quantiles from the standard normal distribution.

Transformer Architecture: In its original form (Vaswani et al. 2017), it consists of an encoder and decoder, but for classification problems only the encoder is used. At its core is the MHA which uses so called self attention to learn important points to focus on. This self attention comprises an attention matrix, which highlights relations between elements of the inputs. This matrix is calculated and applied inside the scaled dot product with itself for all inputs (i. e., V, K, and Q in Figure 3). Figure 3 shows an example of a Transformer encoder corresponding to our applied model.

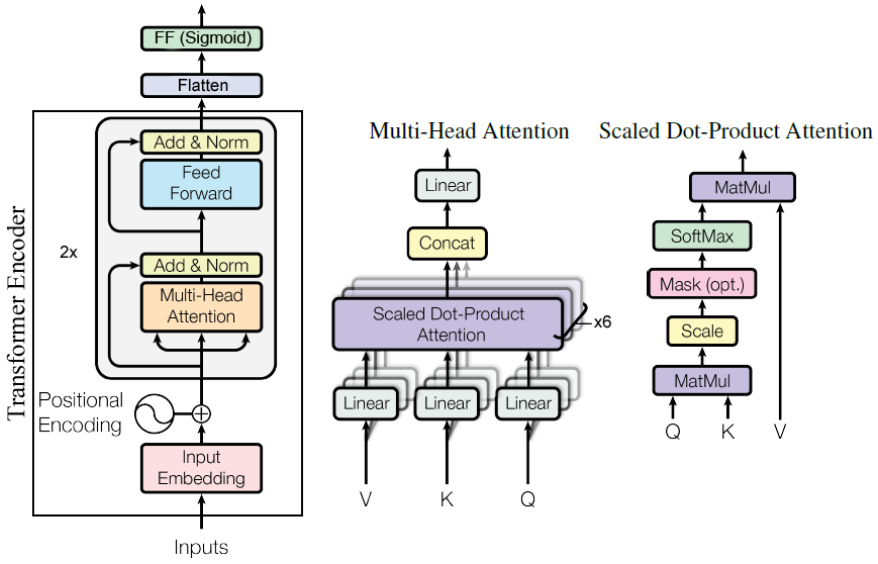

Figure 3: Our applied architecture including the Transformer encoder, adapted from the architecture in (Vaswani et al. 2017).

Abstraction using Attention This main data abstraction step utilizes attention for simplifying the data points within the time series. To merge the attention information from MHA and to reduce the problem of unimportant heads – as discussed by Baan et al. – we simplify all attention matrices into one. Wang, Liu, and Song preferred the average of the attention in the last heads, because the meaning of words (for NLP) was mostly found in the last layers. We on the other hand want to rather visualize what MHA is seeing in our time series in general. Therefore, we aim to show the combination of the strongest highlighted features.

For this reason and further to be able to apply our reduced matrix to a one dimension sequence, we calculate the abstract matrix A_m, by averaging over all layers and summing the maximum of each attention column of all average heads: $A_m = \sum_{j=1}^{m} max_c(\frac{1}{n}\sum_{i=1}^{n} h_i^j)$ where n is the number of layers, m the number of heads and h_i^j is the attention matrix of the j-th head from the i-th layer. The function max_c takes the maximum of each column of the given averaged matrix. In comparison to other possible combination options from the *maximum*, *average* or *sum* of the attention matrices, this strategy turned out to be the best in our experimentation.

At the end of our *abstraction via attention* step we obtain a subset sequence in order to show what the MHA extracts (see Figure 4 (c), (d)) given the attention-abstracted information from the symbolic time series (see Figure 4 (b)). According to Pruthi et al.; Baan et al., this subset is probably not the minimal subset covering the important information; however, it contains particular important information for interpreting the problem regarding the MHA. We propose a subdivision into three subsets with different projections, for including data points into our resulting abstraction, using two thresholds t_1 and t_2:

1. The first threshold t_1 includes every data point with relatively high attention score s. Every data point with attention $s > t_1$ is projected directly into our abstraction.
2. The second group of data contains data points with a medium-strong attention score s, i. e., with $t_2 < s < t_1$. With this, we include partial information into our abstraction. For every directly subsequent sequence of according medium-strongly attended data points, we take the median of the sequence for our abstraction.
3. The remaining group of data points with low attention is removed, because they are considered as unimportant.

Because those thresholds always depend on the problem, we propose a human-in-the-loop procedure to optimize t_1 and t_2 in order to fine-tune the abstraction and interpretability.

Abstraction Validation For interpreting a model, often a subset of data extracted from the original model is used (Mujkanovic et al. 2020). In contrast, we aim at validating our abstraction method. Thus, with the validation step, we aim to assess whether our *obtained subset (in the abstraction)* still contains all necessary information to train a standard model with similar accuracy. Thus, we do not focus the validation on the *original* model, but rather at assessing whether a *newly trained model* performs with similar accuracy using our abstracted data. Here, we consider two data imputation methods: 1. We apply a linear interpolation between all known values to impute the removed ones, and remove all points which cannot be interpolated using a masking layer; 2. We use a *masking layer* to ignore all removed values.

To assess the reduction in complexity in our abstraction, we introduce a simple complexity measure, which counts how many direction changes occur in the time series. We argue that a time series with more shifts in the trend direction is more complex for the human, while a line would be the most simple form with no direction changes.

Results

For demonstrating the efficacy of our presented approach, we applied two datasets – one synthetic and one real-world dataset – which have a quite small univariant sequence length and can somewhat easily be classified by humans, to better judge the results according to one's expectations:

1. The first one is the Synthetic Control Chart time series (Alcock, Manolopoulos, and others 1999), which contains synthetic data for six different data trends. The train and test data is both 300 samples long, where each sequence has the length 60 and each class occurrence is balanced.
2. The second one is an ECG5000 dataset (Goldberger et al. 2000), 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 rare classes.

We preprocessed all data as described above and used a 5-fold cross-validation procedure to generate training and validation sets and to reduce according training biases. The final validation and test statistics are given by the averages obtained across all 5 folds. For a clear assessment and visualization, we analyzed our abstraction on the attention of every fold-model to visualize what each model processes.

Data Preprocessing. All data was standardized to unit variance with the Sklearn (Buitinck et al. 2013) StandardScaler, which was fit on the training data. Afterwards in the first abstraction step the values of each time series is transformed into symbols using SAX, also fit on the training data. To support easy human interpretability while accommodating discriminating power at the same time, we experimented with several scales and abstracted to five symbols (5 bins), i. e., to a value range of very low, low, medium, high and very high, where we used a uniform distribution to calculate the bins. In our experiments, this balanced human interpretation with a relatively low number of individual symbols. We defined a mapping of the ordered set of symbols to the interval $[-1, 1]$ and then mapped the values of the original sequence accordingly. Hereby, we keep the ordering information on the time series, thus preserving the known trend information, rather than approximating it with a word embedding. This is one advantage compared to NLP problems, where this is not possible.

Threshold. With respect to the abstraction thresholds, we experimented with different options and value choices in our semi-automatic process, and determined two approaches with according values on our experimentation. For the *Average* threshold option we selected $t_1 = \tilde{A_m}$ and $t_2 = \frac{t_1}{1.2}$ and for the *Max* threshold: $t_1 = \frac{max(A_m)}{2}$ and $t_2 = \frac{max(A_m)}{3}$. The *Average*-abstraction should provide a good relative abstraction, while the *Max*-abstraction can be used to highlight large attention spikes and therefore provide a spike-dependent data reduction. We tried out different options for the parameters, to roughly minimize the abstraction while aiming at accurate results for both datasets.

Model. As for the model[1], we decided to use a quite simple attention model with acceptable accuracies. Therefore we did not optimize the model, but tried out different parameters, which performed quite similar in regard to the accuracy and abstraction. We used a two layered Transformer encoder, based on the original paper (Vaswani et al. 2017), with 6 heads and a dropout of 0.3, followed by a dense layer which takes in the flattened encoder output. As final output layer we used a sigmoid-based dense layer with one neuron for each output-class. For the training we used an Adam optimizer with included warm-up steps. As for the loss-function we took the mean squared error. The architecture of our model can further be seen in Figure 3.

Model Performance

In Table 1 the performance of the model on the original data and the symbolic-abstracted data can be seen. For the ECG dataset the SAX algorithm did not change much regarding the performance, while for the synthetic dataset the test accuracy dropped by 1,2%. The latter could be explained by the similarity of some classes, which can make it quite hard to distinguish them from each other. Nevertheless the results are acceptable to analyse the models.

Tables 2, 3, 4 and 5 show the results for the different combinations of thresholds and validation inputs while considering the accuracy, the amount of data which was removed (Data Red. By), how many instances changed its prediction compared to the SAX-model (Pred. Changes) and how many trend changes occurred in the data (Trend Shifts). It can be seen that for most instances the models *performed quite similar* or sometimes even slightly better than the SAX model. The worst instances also show, that selecting a *good threshold is quite important* to include the most relevant data points. With respect to performance the *Average* threshold with interpolation performed overall the best. The reduction of the data was always *bigger than 47% and up to 91%* on the ECG dataset without loosing more than 4% accuracy. When considering the trend shifts it can be seen that the data-change-complexity decreased to less than 5% for the ECG and to less than 32% for the Synthetic dataset. This indicates an even bigger data size decrease. Further, it can be seen, that some reduction is already taking place in the SAX applied data, which also displays its abstraction ability. In the reduction it can as well be noticed that the *distribution of the attention is quite important for the threshold selection*. For example, the *Max* threshold worked quite well for the ECG data, where the highest attention points are rather focused, and a lot worse for the Synthetic dataset reduction which has a more equally distributed attention. When looking at the percentile of changed classification it can be seen, that about less than 10% of predictions change. Through our experimental design with cross validation most model dependent influences should be minimized. In some cases, examples in the data suggest that some abstractions can also make some classification decisions harder for humans; therefore, our proposed approach includes visualization and validation as central human-centered steps.

[1] https://github.com/cslab-hub/LocalTSMHAInterpretability

Dataset	Base Acc.	Base Shifts	SAX Acc.	SAX Shifts
ECG Val.	0.9520	137.44	0.9480	22.37
ECG Test	0.9332	90.17	0.9302	22.11
Synth. Val.	0.9500	57.99	0.9533	45.88
Synth. Test	0.9547	56.29	0.9427	45.71

Table 1: Baseline accuracies for the original time series and the over SAX abstracted symbolified time series.

Dataset	Accuracy	Data Red. By	Pred. Changes	Trend Shifts
ECG Val.	0.9520	0.6950	0.0200	16.72
ECG Test	0.9263	0.6950	0.0286	13.22
Synth. Val.	0.9567	0.5166	0.0667	19.17
Synth. Test	0.9540	0.5133	0.0680	17.96

Table 2: Statistics for the *Average* threshold with an interpolated validation input.

Dataset	Accuracy	Data Red. By	Pred. Changes	Trend Shifts
ECG Val.	0.9480	0.6965	0.0220	14.35
ECG Test	0.9258	0.6963	0.0373	19.46
Synth. Val.	0.9500	0.5176	0.0633	17.85
Synth. Test	0.9340	0.5142	0.0667	22.53

Table 3: Statistics for the *Average* threshold with a masked validation input.

Dataset	Accuracy	Data Red. By	Pred. Changes	Trend Shifts
ECG Val.	0.9240	0.9152	0.0440	6.85
ECG Test	0.9090	0.9125	0.049	3.86
Synth. Val.	0.9467	0.4791	0.0733	25.05
Synth. Test	0.9140	0.4825	0.0933	23.46

Table 4: Statistics for the *Max* threshold with an interpolated validation input.

Dataset	Accuracy	Data Red. By	Pred. Changes	Trend Shifts
ECG Val.	0.9300	0.9156	0.0440	5.14
ECG Test	0.9208	0.9129	0.0467	5.83
Synth. Val.	0.9300	0.4808	0.0800	23.85
Synth. Test	0.8973	0.4848	0.0993	24.72

Table 5: Statistics for the *Max* threshold with a masked validation input.

Local Abstract Attention Visualisation

Figure 4 visualizes the local data at each step of the process, for the Synthetic (left) and ECG (right) dataset. In *(a)* the scaled original time series can be seen. In *(b)* the symbolified and to $[-1, 1]$ projected time series can be seen, while right below the attention values are represented. The last visualizations *(c)* and *(d)* show the abstracted interpolated time series for the *Average (c)* and *Max (d)* thresholds. Like also seen in the accuracies, we observe that the *Max* threshold does a better abstraction job for the ECG data in contrary to the Synthetic data. In general, we find that the abstractions are closer to how we would describe the generic pattern of the class while the validation results show us, that the important key elements are still included. Accordingly this verifies that the extraction somewhat describes a pattern which approximates the class definition. Moreover, the found differences in the abstractions between folds were quite similar, but included some variations of small artefacts. This further enhances the idea that our method works in general and additionally that it can help to find artifacts in trained models.

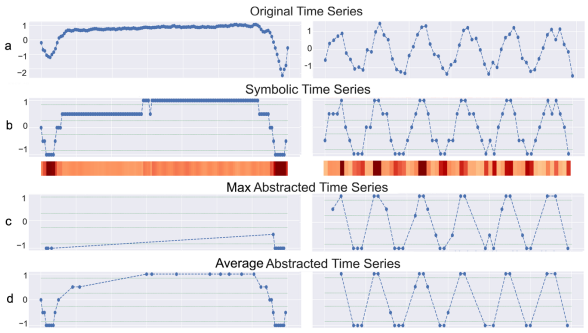

Figure 4: Example abstraction from the Synthetic dataset class 2 (right) and ECG dataset class 2 (left).

Discussion

We already argued that the abstraction tends to remove less important data points and therefore the data gets more accessible for human interpretation. Mainly, our method removes about 47% and up to 91% of less relevant data, while increasing interpretability and preserving performance. When looking at the amount of trend shifts it is being reduced even more. Therefore, we argue that this abstracted shape can help the human to interpret the problem more easily while it also opens up the possibility to analyse what the MHA focuses on. But it is important to keep in mind, that overdoing the abstraction can also make the interpretation somewhat harder to grasp for a human (e. g., Figure 4 (c), left).

From our experience in *fine tuning the thresholds* on different datasets in our proposed human-centered approach, we observed a specific phenomenon: here, each class abstraction is also influenced by the complexity of all other classes, e. g., when optimizing for a simple class only compared to more complex classes. In our opinion, we think that the MHA works as a noise reduction/preprocessing, but also highlights possibly important data for every other class. This could be the case, because the real class decision knowledge is contained in the followup dense layer, while the MHA only shows interesting class coherences. This also could explain how attention could help the learning process, like Pruthi et al. suspected, while not being a straightforward relevancy measurement as argued by Serrano and Smith.

One *general limitation of human-in-the-loop* approaches is the time spent during the iterations in the process. Therefore, with multiple models to train, this can use up a lot of time and hence is suboptimal for very complex models, in general. However, scalability was not a focus of this paper, which we aim to investigate in future work in more detail.

In general, with respect to *more complex models*, we tried to check for the influences of other parameters on our abstraction; we suspect that more heads can increase the performance of the abstraction. Moreover, we noticed that more layers and an input embedding — like typically used for NLP — make the attention more vague. This could be because an embedding layer and every additional MHA layer mixes the relations and important information of every position and therefore similar information-rich data points get grouped together to attention areas in the attention matrix.

Conclusions

In this paper, we focused on the problem of making sense of both the MHA mechanism as well as the complex time series, in order to find out about relevant data points upon which the transformer MHA bases the respective decisions on. For this, we presented an approach making use of interpretable symbolic data representations, supported by a novel visualization technique. In our evaluation, we showed the efficacy of the presented approach – both referring to the performance on the data abstracted via our approach, and on the interpretability of the visualization. In summary, we can see that with our approach the relevant information is identified and preserved, while the information can also be conveniently visualized.

For future work, we aim to apply the proposed approach on further datasets and compare it with other transformer architectures as well as further abstractions. Another interesting direction is given by a more detailed exploration of the symbolic-position coherence matrices in order to analyze the respective dependencies in more detail. Finally, investigating the scalability of the process, also in the context of larger datasets is an important direction which we plan to consider in the future.

Acknowledgements

This work has been supported by Interreg NWE, project DiPlast - Digital Circular Economy for the Plastics Industry.

References

Alcock, R. J.; Manolopoulos, Y.; et al. 1999. Time-series similarity queries employing a feature-based approach. In *7th Hellenic conference on informatics*, 27–29.

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 *2017 IEEE 15th International Conference on Industrial Informatics (INDIN)*, 799–804. IEEE.

Atzmueller, M. 2018. Declarative Aspects in Explicative Data Mining for Computational Sensemaking. In *Proc. International Conference on Declarative Programming (DECLARE)*, 97–114. Heidelberg, Germany: Springer.

Baan, J.; ter Hoeve, M.; van der Wees, M.; Schuth, A.; and de Rijke, M. 2019. Understanding multi-head attention in abstractive summarization. *arXiv preprint arXiv:1911.03898*.

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

Clark, K.; Khandelwal, U.; Levy, O.; and Manning, C. D. 2019. What does bert look at? an analysis of bert's attention. *arXiv preprint arXiv:1906.04341*.

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*.

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

Goldberger, A. L.; Amaral, L. A.; Glass, L.; Hausdorff, J. M.; Ivanov, P. C.; Mark, R. G.; Mietus, J. E.; Moody, G. B.; Peng, C.-K.; and Stanley, H. E. 2000. Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. *circulation* 101(23):e215–e220.

Jain, S., and Wallace, B. C. 2019. Attention is not explanation. *arXiv preprint arXiv:1902.10186*.

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. *arXiv preprint arXiv:1907.00235*.

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.

Mujkanovic, F.; Doskoč, V.; Schirneck, M.; Schäfer, P.; and Friedrich, T. 2020. timexplain–a framework for explaining the predictions of time series classifiers. *arXiv preprint arXiv:2007.07606*.

Pruthi, D.; Gupta, M.; Dhingra, B.; Neubig, G.; and Lipton, Z. C. 2019. Learning to deceive with attention-based explanations. *arXiv preprint arXiv:1909.07913*.

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

Serrano, S., and Smith, N. A. 2019. Is attention interpretable? *arXiv preprint arXiv:1906.03731*.

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

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762*.

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

Wang, C.; Liu, X.; and Song, D. 2020. Language models are open knowledge graphs. *arXiv preprint arXiv:2010.11967*.