Identifying Informative Nodes in Attributed Spatial Sensor Networks Using Attention for Symbolic Abstraction in a GNN-based Modeling Approach

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

Stefan Bloemheuvel
Tilburg University
Jheronimus Academy of Data Science
’s-Hertogenbosch, The Netherlands
s.d.bloemheuvel@jads.nl

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

Abstract

Modeling complex data, e. g., time series as well as network-based data, is a prominent area of research. In this paper, we focus on a combination of both, analyzing network-based spatial sensor data which is attributed with high frequency time series information. We apply a symbolic representation and an attention-based local abstraction approach, to enhance interpretability on the respective complex high frequency time series data. For this, we aim at identifying informative measurements captured by the respective nodes of the sensor network. To do so, we demonstrate the efficacy of the Symbolic Fourier Approximation (SFA) and the attention-based symbolic abstraction method to localize relevant node sensor-information, by using a transformer architecture as an encoder for a graph neural network. In our experiments, we compare two seismological datasets to their previous state-of-the-art model, demonstrating the advantages and benefits of our presented approach.

Introduction

The analysis of complex sensor network data is typically rather challenging, e. g., due to high dimensionality, large volumes of data, high frequency information attributed to the network nodes, etc. (Tubaishat and Madria 2003). Here, standard modeling techniques are often facing severe limitations, considerably restricting our ability to draw meaningful insights from the data. To address these challenges, specialized modeling techniques that can effectively handle complex data in a structured and interpretable way are required. We tackle this outline issue in two different steps:

1. To model a spatial sensor network, where the nodes are attributed with high frequency time series data, we exploit and adapt a graph neural network (GNN) based approach introduced by Bloemheuvel et al. (2022).
2. By introducing an attention-based Transformer encoder (Vaswani et al. 2017) into the model, we enable the use of a symbolic abstraction technique to draw insight from the data. In particular, it enables us to find informative nodes, thus essentially reducing complexity and potential cognitive load for human interpretation.

In our presented approach, we therefore combine the modeling power of graphs with an interpretable abstraction method to identify informative nodes embedded into an extrinsic regression task (Tan et al. 2021). While there are existing methods for identifying important nodes in networks and graphs, respectively, e. g., by Ying et al. (2019) and Baldassarre and Azizpour (2019), such techniques are not optimized for regression tasks, even less so in our domain which combines graphs with complex high-frequency time series data. In general, time series encode information in a redundant, continuous, numeric and non-intuitive way, making them hard to grasp for humans (Rojat et al. 2021), emphasizing the need for suitable explanation methods. Furthermore, they introduce rather challenging properties for neural networks (Shen, Wei, and Wang 2022). Thus, we introduce SFA as a symbolic representation into our approach, with the goal of enhancing its interpretability – while maintaining the overall performance.

We demonstrate the efficacy of these methods on two seismic datasets (Michelini et al. 2016), i. e., on spatial sensor networks attributed with high frequency time series data. Our contributions are summarized as follows:

1. We present an attention-based approach for identifying informative nodes in an attributed spatial sensor networks, enhanced by symbolic representations.
2. We evaluate the proposed approach in the context of a GNN-based extrinsic regression task. In our experimentation, we compare the proposed method in various instantiations to the state-of-the-art baseline, demonstrating the efficacy of the presented approach. Furthermore, we discuss insights and implications w.r.t. its interpretability.

The rest of the paper is structured as follows: We first introduce some background notions as well as related work on symbolic time series embeddings, the Transformer architecture – corresponding to our attention-based abstraction method – as well as GNNs and discuss the issue of explainability. After that, we briefly summarize properties and characteristics of the applied datasets in the evaluation. Next, we present our proposed approach, combining the SFA, local attention-based symbolic abstraction as well as the applied GNN models. Then, we present and discuss the results of our experimental evaluation, before we conclude with a summary and discuss interesting options for future work.

Related Work and Background

Below, we summarize foundational notions and related work on symbolic time series embeddings, Transformers as well as explainability in the context of GNNs and time series.

Symbolic Time Series Embeddings

Recently, multiple new word2vec-like (Mikolov et al. 2013) embeddings emerged, e. g., cf. Nalmpantis and Vrakas (2019); Kim, Hong, and Cha (2020); Yue et al. (2022); Ye and Ma (2022); Tabassum, Menon, and Jastrzebska (2022). However, most of those representations are suboptimal in terms of interpretability (Schwenke and Atzmueller 2023), due to the less accessible nature of time series data and their mapped representation, compared to words and word embeddings. Therefore, we aim to use more symbolic embedding approaches, like Symbolic Aggregate approXimation (SAX) (Lin et al. 2003) or Symbolic Fourier Approximation (SFA) (Schäfer and Högqvist 2012). SAX, in particular, has been successfully applied as a symbolic embedding for deep learning (Lavangnananda and Sawasdimongkol 2012; Schwenke and Atzmueller 2021c; 2021b; Criado-Ramón, Ruiz, and Pegalajar 2022; Tabassum, Menon, and Jastrzebska 2022), via a more human related representation, cf. Atzmueller et al. (2017); Ramirez, Wimmer, and Atzmueller (2019). However, it is rather limited to the context of trend-based data. As shown by Schwenke and Atzmueller (2023), SFA can be quite helpful, in principle, to improve the interpretability and even performance on frequency-based data.

Transformer Architecture

The Transformer architecture (Vaswani et al. 2017) with its Multi-Head Attention (MHA) mechanism is a recently emerged neural network architecture which provides the basis for multiple state-of-the-art applications; focusing mostly on Natural Language Processing (NLP) (Vaswani et al. 2017) and Computer Vision (CV) (Dosovitskiy et al. 2020). Lately, Transformers were successfully adapted to time series tasks (Lim et al. 2019; Li et al. 2019; Wen et al. 2022), for example, also addressing some challenges of time series data, like reducing the memory bottleneck easily reached by time series data (Tay et al. 2020). Transformers are especially interesting for time series data due to their ability to handle long-term dependencies (Li et al. 2019), while also being able to act as data encoder, e. g., the original encoder-decoder application. Additionally, the introduction of attention with the MHA enables further explainable methods (Vig 2019; Škrlj et al. 2020; Schwenke and Atzmueller 2021c; 2021b). Here, we utilize the Transformer architecture as a foundation of our attention-based abstraction approach.

Explainability: GNNs and Time Series

In general, graphs enable the modeling of complex relationships between variables, observations, and objects, in the context of complex systems (Strogatz 2001; Albert and Barabási 2002; Kipf and Welling 2017; Li et al. 2017; Jozinović et al. 2020; Bloemheuvel et al. 2022). Various DL methods exist to operate on such graph-structured data, generally called Graph Neural Networks (GNNs). GNNs can preserve the graph structure and capture the complex relational information between nodes, including attributes such as edge and node features. One of the most popular techniques is given by Graph Convolutional Neural Networks (GCNs) (Kipf and Welling 2017), while recently Graph Attention Networks (Veličković et al. 2018) have also emerged.

Interpretability and explainability of GNNs are important facets, cf. Yuan et al. (2022); Li et al. (2022). One of the central questions involves the identification of a set of nodes or sub-graphs which are more important regarding a specific class: In this paper, we tackle this using attention on symbolic time series representations on a GNN model. However, most of these techniques are designed for node or graph-classification tasks. Whereas our task is a multivariate extrinsic regression tasks where the main focus lies on the node attributes, i. e., processing the whole graph structure using fewer nodal information. Additionally, attention-based methods are still quite sparsely researched on graphs (Holzinger et al. 2021; Yuan et al. 2022), which further motivates the application of our proposed abstraction approach.

Due to the complex nature of time series data, explainable artificial intelligence (XAI) on time series data is challenging. Nonetheless, many methods and models already exist to cope for interpretability, where a few were summarized by Rojat et al. (2021) and Theissler et al. (2022). With the recent rise of Transformer models on time series data (Wen et al. 2022), the need for corresponding attention-based XAI methods arises. This is even more important because attention-based methods from other domains seem not to work well on time series data (Ismail et al. 2020). Recently, e. g., Schwenke and Atzmueller (2021c) introduced an attention-based local abstraction method for time series data. In this paper, we adapted this method to our sensor graph context, including high-frequency time series assigned to the respective nodes of the network as attributes. Thus, in contrast to the methods described above, we tackle both a graph structure and complex attributed information given by high-frequency time series data.

Datasets

Our analysis is performed in the context of a multivariate extrinsic regression task (Tan et al. 2021) using two datasets of the Italian national seismic network (Michelini et al. 2016). Graphs can be regarded as a rather suitable modeling method for the analysis of seismic data, due to seismic measurements containing sensors that are geographically grounded and contain a large amount of data. Each sensor in a seismic network measures seismic waves in three dimensions: up-down, north-south and east-west. Regarding our applied datasets, the first dataset is the so-called CI dataset, consisting of 915 earthquakes recorded by 39 stations. This network is densely populated by seismic stations, and all earthquakes are originating from a small region (Michelini et al. 2016). The second dataset, called the CW dataset, consists of 266 earthquakes (also recorded by 39 stations), but covers a larger land area (Michelini et al. 2016). In addition, the earthquakes are also more scattered. Therefore, both datasets in combination provide distinct scenarios, providing a nice general overview for evaluation.

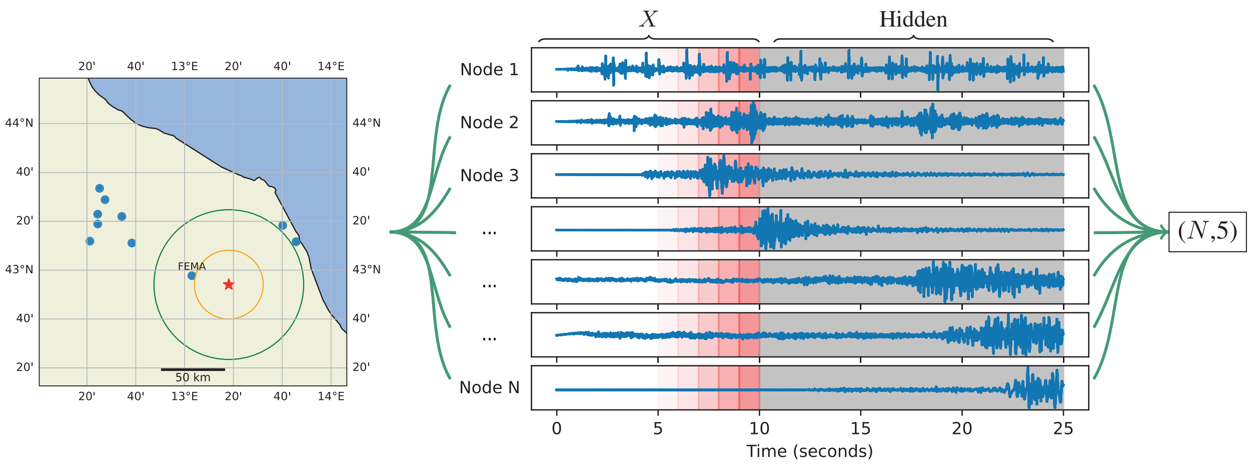


Figure 1: Dataset task overview from Bloemheuvel et al. (2022). The FEMA station already measured the earthquake in the initial 10 seconds, but further away stations have not. By taking this initial n seconds, the goal is to predict 5 different intensity measurements of the earthquake at each station.

The task in both datasets is to only take an initial 10 seconds of the input data of an incoming earthquake from multiple stations, in order to predict the future maximum intensity measurements of all stations; including stations faraway from the earthquakes' epicenter, where the acceleration waveforms did not reach yet. The 10-second window makes sure that only nearby stations have had the opportunity to actually measure the earthquake. The procedure is visually explained in Figure 1. The regression target values are external parameters of the input, which do not necessarily depend on recent values, but rather on the whole length of the time series (Tan et al. 2021). A total of 5 target parameters for each station are predicted, namely peak ground acceleration (PGA), peak ground velocity (PGV) and spectral acceleration (SA) at 0.3s, 1s and 3s periods, cf. Jozinović et al. (2020). The average mean squared error (MSE) was used to evaluate the results of all models. To compare the overall performance of all models, the MSE of all target values is averaged into one error score.

For both datasets, an 80%-20% train-test split was used. The shuffled train set was further split using a five-fold cross-validation. The train-test split is repeated 5 times to guarantee stable results, since the results depend on the distribution of larger earthquakes in the validation and test sets.

Method

Below, we present our approach integrating symbolic abstraction with graph modeling, which we evaluate in different instantiations. Figure 3 exemplifies the applied process.

Symbolic Fourier Approximation

Using the Discrete Fourier Transformation (DFT) (Winograd 1978), time series data can be decomposed into base functions with different frequencies and amplitudes. It is a typical method for time series pre-processing and approximation (Nalmpantis and Vrakas 2020). To further abstract and reduce the data load, Schäfer and Högqvist (2012) introduced the Symbolic Fourier Approximation (SFA) which discretizes the Fourier coefficients into symbols using intervals and approximates the original sample with only a subset of Fourier coefficients, i. e., reducing noise from higher frequencies (Schäfer and Högqvist 2012; Nalmpantis and Vrakas 2020). With this approach, we can avoid using windowing while building an informative and more accessible data structure, i. e., trained models are smaller and faster, while performing possible better on high frequent data, cf. Schwenke and Atzmueller (2023). An example process can be seen in Figure 2. SFA has three important parameters: the number of intervals/symbols, number of Fourier coefficients and the interval building strategy. Schwenke and Atzmueller (2023) showed that a uniform interval distribution works nearly always best on time series data. The smaller the other two parameters, the simpler the data is. For our models, we explored and selected a fixed set of well performing values for better comparison.

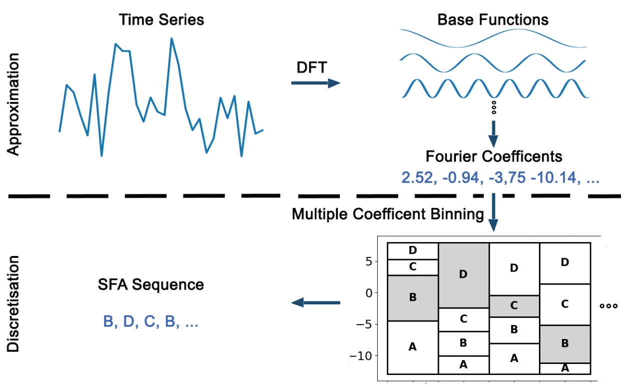


Figure 2: Example pipeline for SFA, based on and adapted from Schäfer and Högqvist (2012).

Local Attention-based Symbolic Abstraction

In our application case, we aim to reduce the number of sensors providing information in our sensor network of seismic stations while aiming to keep the overall performance. In the general graph modeling approach, we thus want to identify more informative nodes (corresponding to the stations). Schwenke and Atzmueller (2021c) introduced a local attention-based abstraction method, with the goal to locally simplify time series data using a human-in-the-loop to promote interpretability; thus, this is quite suitable for our approach. For the sake of readability, we refer to this method as Local Attention-based Symbolic Abstraction (LASA).

In the original method, the abstraction process is done in the time dimension of the respective time series, such that only specific time points or time intervals are considered in the abstraction. This then targets the abstraction of *trend* information. However, this approach struggles to maintain frequency based information, as we have in our current case. Thus, to apply LASA, we first replace SAX with SFA to enable symbolification on the frequency based data, while reducing the model load with a smaller input sequence. By restructuring the data format w.r.t. the nodal information $(node, attributes)$, the abstraction process can be applied on the nodes rather than the measurement dimension to reduce attributed information of the nodes, cf. Figure 3. It is important to note, that LASA has not been tested on Graph-like structures before, e. g., for extracting informative node information to simplify the graph; nor has it been applied when a Transformer acts as an encoder of another model.

We train and validate our model afterwards with the mapped symbolic data — to [-1,1] w.r.t. the vocabulary relations, as suggested by Schwenke and Atzmueller (2021c) — to ensure that our selected parameters still capture the task information. Afterwards, we aggregate all attention matrices from the Transformer into a vector, by reducing the dimensions using the maximum operation. We refer to Schwenke and Atzmueller (2021a) for a detailed discussion about this aggregation process. Afterwards, locally for each sample, all vector entries below a certain threshold get masked; compared to Schwenke and Atzmueller (2021c) we use only one threshold due to the graph structured data. By re-training the model with the masked values, we verify that the information needed for the task is still included. In a human-in-the-loop approach, this threshold can now be fine-tuned to cope for a suited performance-to-data-reduction ratio. Figure 3 visualizes an example abstraction process.

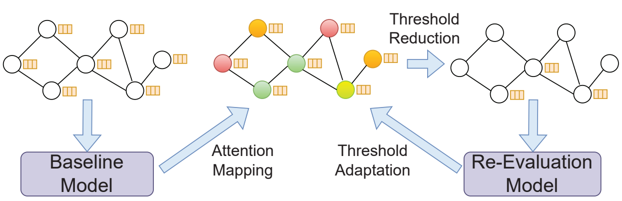


Figure 3: Local Attention-based Symbolic Abstraction process on graphs, i. e., keeping the SFA coefficient supplement information on high attended nodes (green).

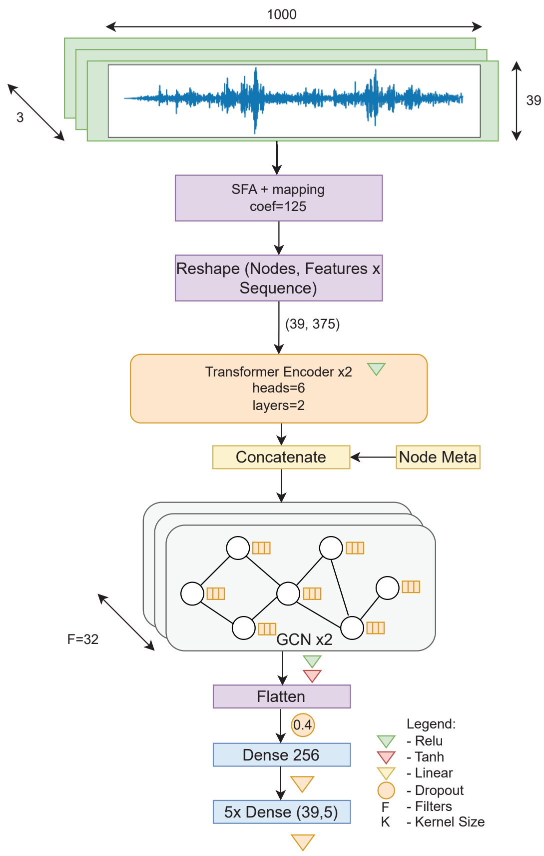


Figure 4: Overview of our proposed TISER-TGCN model.

Models

We compare three models to analyze the effect of the SFA and the Transformer encoder. As a first baseline we use the original CNN model from Jozinović et al. (2020), which is the previous version of the state-of-the-art TISER-GCN model (our second baseline) from Bloemheuvel et al. (2022); which includes an GCN to preprocess the graph structure. As the third option, we propose the novel TISER-TGCN model: it replaces the CNN layer from TISER-GCN with a Transformer encoder to process the measured time series information per station, as shown in Figure 4. Compared to the normal Transformer encoder, we removed the positional encoding due to the graph-like data structure with no sequential relation. It is important to note, that due to the memory limitations of the Transformer, TISER-TGCN could only be run when applying the SFA transformation. Figure 4 depicts our architecture. First, we apply SFA to (1) extract meaningful frequency data, (2) abstract the inputs to discrete symbolic values to enhance interpretability, while enabling the application of LASA and (3) reduce the input size and thus the model size. The mapped data gets afterwards reshaped to match the transformer input. We concatenate the symbolified sensor data into one sequence to do so. The data is then fed into two Transformer encoder layers. The encoded data afterwards flows into a GCN, which also gets the graph structure and node metadata as additional input. At the end, the five outputs are generated via a flattened dropout in combination with some Dense layers. Each model is trained for up to 200 epochs, with 15 epochs patience. The CNN models uses the RMSprop optimizer and the TISER-TGCN uses the Adam optimizer with warm-up steps.

To compare the effect of our model modifications, we calculate for all baseline models (1) the proposed *original* configuration, (2) a modified parameter set (abbreviated as *ModParams*) we optimized for the TISER-TGCN model and (3) the models with SFA as input (including ModParams, denoted as *SFA*) to highlight the effect of SFA between TISER-TGCN and the other models. The new parameter set ModParams was necessary, due to the reduced input size, which made the previous kernel size not applicable on the SFA models. For our model parameters, in the context of our implementation[1], we refer to Figure 4.

To evaluate the LASA method, we train three LASA models with the thresholds t0.5, t1.0 and t1.4. For e. g., t0.5 this means we take the average of all attention values divided by 0.5. Those thresholds can be fine-tuned in a human-in-the-loop process for optimizing the reduction-performance ratio. Here, for a comprehensive overview on the threshold effects, we considered a broad range of the thresholds.

Results

This section presents our results. First, we analyze the performance of our presented approach, after that we discuss the impact on identifying informative nodes in our applied sensor network context.

Performance

In Table 1, we compare the MSE performance of the models for the CI and CW dataset on a 10-second window. We can see that the TISER-GCN SFA and TISER-TGCN model (which both use the SFA) clearly outperform the previous state-of-the-art TISER-GCN model on both datasets, thus showing the effect of the SFA. Interesting are also the TISER-TGCN LASA results, especially with the t1.0 threshold, which – for the LASA models – performed best on both datasets. For the CI dataset, the performance decreases for each threshold, even if we only reduce 50% of the data; indicating that close stations enhance the information pool e. g., by ensuring measurements. On the other hand, for the CW dataset LASA improves the results, indicating that distant stations provide more noise than benefits for the regression.

By comparing the results for smaller time windows, see Figures 6 and 7, we can see that SFA outperforms the baseline at each time window; even providing a more stable linear loss in performance, while the baseline seems more exponential. This shows that the SFA can successfully approximate the crucial regression information, while removing noise and represent the data in a small fixed sequence regardless of the initial data length. Due to this smaller input size, we could also decrease the prediction time of the test set to $0.33s$ for TISER-GCN and $0.58s$ for TISER-TGCN, compared to $1.15s$ of the TISER-GCN.

[1] https://github.com/lschwenke/GraphNodeAttention

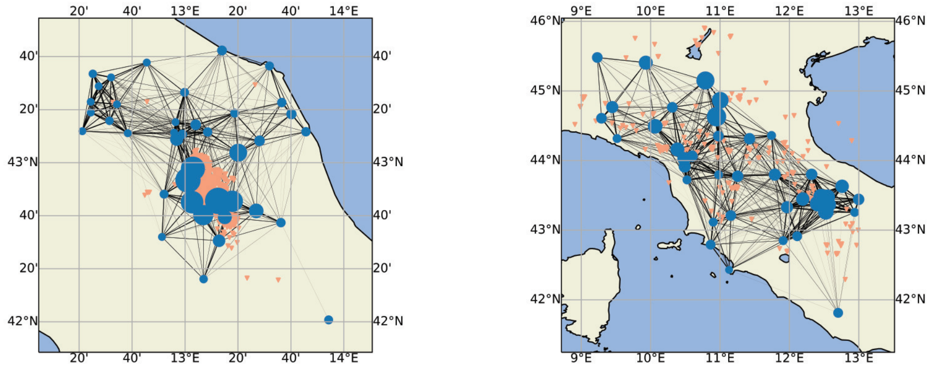
Table 1: MSE (best performing in bold, second best in underscore) of all models for both datasets with a window length of 10s, and the reduction amount for the LASA technique.

Model	Average	Reduction
CI Network		
TISER-GCN Original	0.2007 ± 0.0273	-
TISER-GCN ModParams	0.2098 ± 0.0269	-
TISER-GCN SFA	$\mathbf{0.1722 \pm 0.0076}$	-
CNN Original	0.2543 ± 0.0317	-
CNN ModParams	0.2809 ± 0.0285	-
CNN SFA	0.4840 ± 0.0415	-
TISER-TGCN	$\underline{0.1728} \pm 0.0134$	-
TISER-TGCN LASA t0.5	0.2013 ± 0.0184	$95.24\% \pm 01.18\%$
TISER-TGCN LASA t1.0	0.1885 ± 0.0204	$91.09\% \pm 00.80\%$
TISER-TGCN LASA t1.4	0.1909 ± 0.0217	$52.72\% \pm 18.45\%$
TISER-TGCN LASA^2	0.1941 ± 0.0155	$94.42\% \pm 03.25\%$
CW Network		
TISER-GCN Original	0.2929 ± 0.0523	-
TISER-GCN ModParams	0.3184 ± 0.0664	-
TISER-GCN SFA	0.2597 ± 0.0338	-
CNN Original	0.3514 ± 0.0488	-
CNN ModParams	0.4305 ± 0.0544	-
CNN SFA	0.6617 ± 0.0535	-
TISER-TGCN	0.2685 ± 0.0219	-
TISER-TGCN LASA t0.5	0.2629 ± 0.0554	$94.29\% \pm 01.59\%$
TISER-TGCN LASA t1.0	$\mathbf{0.2332 \pm 0.0329}$	$92.01\% \pm 01.56\%$
TISER-TGCN LASA t1.4	$\underline{0.2421} \pm 0.0290$	$74.17\% \pm 17.68\%$
TISER-TGCN LASA^2	0.2556 ± 0.0483	$95.01\% \pm 01.96\%$

With smaller windows, most of the time the TISER-GCN SFA model outperformed even the TISER-TGCN clearly. We further observe that the LASA 1.0 model approximates the original model, for both datasets. This could however also be due to a suboptimal threshold, which is not always simple to fine-tune due to the sometimes noisy attention values (Schwenke and Atzmueller 2021c). In our experimentation, we only tested three thresholds as an approximation, where overall t1.0 performed best.

Analysis on Identifying Informative Nodes

While the Transformer encoder model TISER-TGCN performed very similar to the TISER-GCN SFA on the 10s window, the attention mechanism in the Transformer allows us to apply LASA. Table 1 also shows the performance and reduction of the data for three different thresholds. The reductions of $>90\%$ shows that with only 1-5 stations, we can already quite well predict most samples; i. e., LASA can act as an abstraction method on graphs, even when the Transformer is only an encoder. The LASA-selected stations were typically close to the epicenters of the earthquakes, with sometimes an additional more distant station (maybe to reduce noise over distance). Figure 5 emphasizes this by showing how often each station (blue) is selected. Especially in the CI network, where most earthquakes are clustered in the center (orange), it is easy to see that mostly the very



(a) CI Network

(b) CW Network

Figure 5: Summary of LASA selected stations (blue) in relation to earthquakes (orange). Bigger nodes are more often selected.

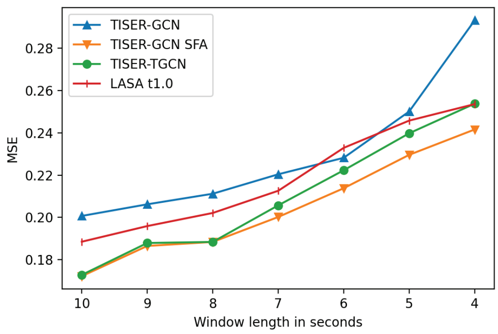


Figure 6: CI dataset's MSE performance development of multiple models over different window lengths.

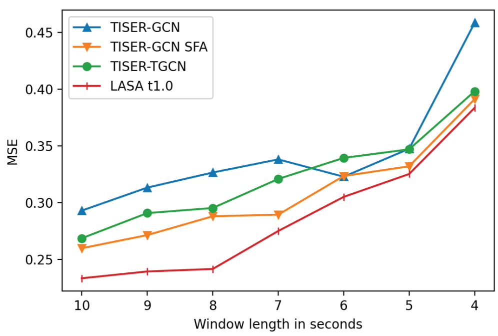


Figure 7: CW dataset's MSE performance development of multiple models over different window lengths.

close stations are selected over LASA. We observe, that due to the redundant information inside the data, multiple valid reductions exist and our results depend heavily on the trained model. As an alternative test, we also successfully reduced less important Fourier coefficients, by changing the first sample dimension. However, in this setting the overall results were significantly worse, which is why we do not report them directly. As a further addition, we tested to apply LASA twice; first on the coefficient dimension (t1.4) and afterwards on the stations (t1.0). Using this double reduction, we reduce information in two dimensions and can even better pinpoint informative features. In contrast to just reducing the coefficients, we received comparable results, as can be seen at LASA^2 in Table 1.

In summary, using LASA we can explore the data and find more informative stations over a human-in-the-loop. This process can give insight into the model's decision w.r.t interpretability. Here we showed that the model mostly detects nodes closer to the epicenter to be more informative, which matches the human intuition. Seismologists could e. g., use our abstraction approach to find an informative set/range of stations inside a larger area to reduce noise and computation time, cf. CW network. In addition, they could also identify defective stations with noisy or unreliable data indicating that more stations are needed for a reliable application.

Conclusion and Future Work

In this paper, we presented an approach combining the modeling power of graphs with an abstraction method to identify informative nodes embedded into an extrinsic regression task. By adapting an attention-based abstraction method to our context, we demonstrated how to improve local interpretability by focusing on the more informative nodes. Hereby, the graph provided structural spatial information on the measured information, enabling meaningful predictions on the whole graph. In our seismic application, the relevant information mostly focused on stations close to the epicenter, leading even to a noise reduction effect on the scattered network. Also, by introducing a symbolic representation into the models, we could increase the interpretability, speed and performance of the TISER-GCN, while enabling our Transformer model. This indicates the importance of such data representations. In future work, a comparison to other Graph XAI methods (Yuan et al. 2022; Li et al. 2022), as well as higher-order pattern structures (Interdonato et al. 2019; Atzmueller et al. 2019) could be considered.

References

Albert, R., and Barabási, A. 2002. Statistical Mechanics of Complex Networks. *Reviews of modern physics* 74(1):47.

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.

Atzmueller, M.; Soldano, H.; Santini, G.; and Bouthinon, D. 2019. MinerLSD: Efficient Mining of Local Patterns on Attributed Networks. *Applied Network Science* 4(43).

Baldassarre, F., and Azizpour, H. 2019. Explainability techniques for graph convolutional networks. *arXiv preprint arXiv:1905.13686*.

Bloemheuvel, S.; van den Hoogen, J.; Jozinović, D.; Michelini, A.; and Atzmueller, M. 2022. Graph neural networks for multivariate time series regression with application to seismic data. *International Journal of Data Science and Analytics* 1–16.

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.

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*.

Holzinger, A.; Malle, B.; Saranti, A.; and Pfeifer, B. 2021. Towards multi-modal causability with graph neural networks enabling information fusion for explainable AI. *Information Fusion* 71:28–37.

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).

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.

Jozinović, D.; Lomax, A.; Štajduhar, I.; and Michelini, A. 2020. Rapid prediction of earthquake ground shaking intensity using raw waveform data and a convolutional neural network. *Geophysical Journal International* 222(2):1379–1389.

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.

Kipf, T. N., and Welling, M. 2017. Semi-supervised classification with graph convolutional networks. In *Proc. International Conference on Learning Representations*.

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

Li, Y.; Yu, R.; Shahabi, C.; and Liu, Y. 2017. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv preprint arXiv:1707.01926*.

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, P.; Yang, Y.; Pagnucco, M.; and Song, Y. 2022. Explainability in graph neural networks: An experimental survey. *arXiv preprint arXiv:2203.09258*.

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. SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, 2–11.

Michelini, A.; Margheriti, L.; Cattaneo, M.; Cecere, G.; D'Anna, G.; Delladio, A.; et al. 2016. The Italian National Seismic Network and the earthquake and tsunami monitoring and surveillance systems. *Advances in Geosciences* 43:31–38.

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 *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.

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.

Schwenke, L., and Atzmueller, M. 2021a. Abstracting local transformer attention for enhancing interpretability on time series data. In *LWDA*, 205–218.

Schwenke, L., and Atzmueller, M. 2021b. Constructing global coherence representations: Identifying interpretability and coherences of transformer attention in time series data. In *2021 IEEE 8th International Conference on Data Science and Advanced Analytics (DSAA)*, 1–12. 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 *The International FLAIRS Conference Proceedings*, volume 34.

Schwenke, L., and Atzmueller, M. 2023. Making time series embeddings more interpretable in deep learning: Extracting higher-level features via symbolic approximation representations. In *The International FLAIRS Conference Proceedings*, volume 36.

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

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*.

Strogatz, S. H. 2001. Exploring Complex Networks. *Nature* 410(6825):268–276.

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.

Tan, C. W.; Bergmeir, C.; Petitjean, F.; and Webb, G. I. 2021. Time series extrinsic regression. *Data Mining and Knowledge Discovery* 35(3):1032–1060.

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

Theissler, A.; Spinnato, F.; Schlegel, U.; and Guidotti, R. 2022. Explainable AI for time series classification: A review, taxonomy and research directions. *IEEE Access*.

Tubaishat, M., and Madria, S. 2003. Sensor networks: an overview. *IEEE potentials* 22(2):20–23.

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.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph attention networks. *International Conference on Learning Representations*.

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

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.

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.

Ying, Z.; Bourgeois, D.; You, J.; Zitnik, M.; and Leskovec, J. 2019. Gnnexplainer: Generating explanations for graph neural networks. *Advances in neural information processing systems* 32.

Yuan, H.; Yu, H.; Gui, S.; and Ji, S. 2022. Explainability in graph neural networks: A taxonomic survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

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.