
C-SHAP for time series: An approach to high-level temporal explanations

Annemarie Jutte^{*1,2}, **Faizan Ahmed**^{1,2}, **Jeroen Linssen**¹, and **Maurice van Keulen**²

¹Saxion University of Applied Sciences, Enschede, The Netherlands

²University of Twente, Enschede, The Netherlands

Abstract

Time series are ubiquitous in domains such as energy forecasting, healthcare, and industry. Using AI systems, some tasks within these domains can be efficiently handled. Explainable AI (XAI) aims to increase the reliability of AI solutions by explaining model reasoning. For time series, many XAI methods provide point- or sequence-based attribution maps. These methods explain model reasoning in terms of low-level patterns. However, they do not capture high-level patterns that may also influence model reasoning. We propose a concept-based method to provide explanations in terms of these high-level patterns. In this paper, we present C-SHAP for time series, an approach which determines the contribution of concepts to a model outcome. We provide a general definition of C-SHAP and present an example implementation using time series decomposition. Additionally, we demonstrate the effectiveness of the methodology through a use case from the energy domain.

Keywords Explainable AI · Time series · Concept-based XAI · SHAP

1 Introduction

As the adoption of artificial intelligence (AI) systems grows, awareness of the opaque nature of many of the underlying models is becoming more widespread [1, 19]. This lack of transparency prevents the reliable use of AI and can lead to user distrust, specifically in high-stakes environments. To increase trust in AI models, users need insight into the models’ reasoning [6]. The field of explainable AI (XAI) aims to uncover the reasoning of AI models by providing explanations [1].

This paper considers XAI for time series. AI has been applied successfully in a wide range of tasks involving time series, in domains such as energy forecasting [7], healthcare [16], and industry [10]. Traditionally, most XAI methods for time series are point- and subsequence-based [23], they aim to uncover which specific points or subsequences in a temporal sequence contribute most towards a model prediction.

However, a model might not only leverage local patterns, but also higher-level patterns for decision making. These higher-level patterns will be referred to as ‘concepts’ in this article. While the term ‘concept’ is often used in literature on XAI [18], a universal definition does not yet exist. In this paper, we follow the interpretation by Goyal et al. [9], that a concept is a “higher level unit than low-level individual input features”.

It has been shown for time series that in some cases high-level concepts, such as trend and frequency, may better explain model behaviour than point-based attribution [12]. In addition to the technical possibilities of concept-based explanations, using high-level human interpretable concepts may better match user understanding [18].

*a.m.p.jutte@saxion.nl

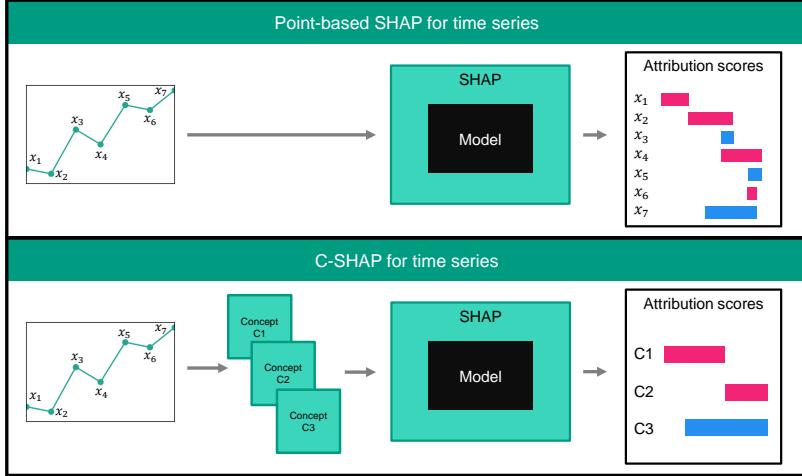


Figure 1: Whereas point-based attribution determines the attribution of individual points, concept-based attribution considers higher-level features: concepts.

This article presents C-SHAP for time series. The approach is based on SHAP, which is a model-agnostic method to measure the contribution of features to a model outcome [13]. These features are usually defined as low-level model input features, but we define the SHAP features as concepts. Although various XAI methods for time series using SHAP have been proposed [2, 15, 17], these are all point- or sequence-based attribution methods. C-SHAP instead provides attributions for concepts, see Figure 1.

An alternative approach to SHAP, for concept-based explanations, is Testing with Concept Activation Vectors (TCAV) [11]. TCAV constructs Concept Activation Vectors (CAVs), which represent a model’s activation response to predefined concepts. CAVs are based on (linear) classifiers, which are trained to distinguish between the model activations for samples containing a concept and random other samples. The explanation for a sample is generated using the similarity between its model activations and the CAVs.

ConceptSHAP [24] combines TCAV and SHAP, Yeh et al. [24] applied it to image and language data. A completeness score is defined to evaluate the concepts used in TCAV. ConceptSHAP estimates the contribution of individual concepts to the completeness score. We instead propose to directly measure the contribution of concepts to the model output.

While TCAV has shown to be successful for time series tasks [3, 14], C-SHAP offers advantages. SHAP is model-agnostic and works regardless of model and data selection. TCAV requires the training of classifiers on the model activations, which requires knowledge of the model and complicates its application. Additionally, the training of these classifiers introduces uncertainty. Furthermore, TCAV assumes linear separability of activations of samples with and without a concept. Methods may be introduced to expand to non-linear cases [5], but this introduces opaqueness to the explanation itself.

For image data, a concept-based SHAP approach has been successfully applied to determine the attribution of concepts, in the form of image segments [21]. In this article, we generalize the concept-based definition for SHAP for images and apply it to the time series domain. The main contribution of this article is the conceptualization of C-SHAP for time series.

Additionally, we provide an implementation of C-SHAP. To implement C-SHAP, concepts need to be selected and constructed for which the attribution scores are determined. In this paper, time series decomposition is used. However, we emphasize that C-SHAP works regardless of the method of concept construction. Finally, we also provide a proof of concept in which C-SHAP is applied to a black-box model for energy consumption forecasting.

Due to the novelty of concept-based explanations for time series, the state of the art is limited. Therefore, the discussion of the state of the art is merged with the introduction. The rest of the paper is structured as follows: Section 2 provides the definition of C-SHAP for time series, Section 3 presents the implementation of

C-SHAP using time series decomposition, Section 4 provides the proof of concept for the energy use case, and Section 5 describes the findings of this research and discusses the implications for future work.

2 Conceptualization of C-SHAP

Shapley values [20] represent the contribution of variables to a function output. Given a black box data-driven model, SHAP [13] is a method that estimates Shapley values for data features as their contribution to the model output. These estimated Shapley values are called SHAP values. In this section, we briefly discuss SHAP and present the definition of C-SHAP for time series.

For univariate time series, consider an input sample $\mathbf{y}(t) = (x_1, x_2, \dots, x_n)$. Shapley values are based on coalitions, if $G = \{g_1, g_2, \dots, g_m\}$ is the full set of features, a coalition is defined as a subset of features ($S \subseteq G$). For each coalition S , the model is applied to a masked model input \mathbf{z}_S , where any features in $\mathbf{y}(t)$ which are not in coalition S are masked, i.e. ‘toggled off’. Features can be masked by replacing their value with an uninformative background value. We denote the model output of the masked input as $f_{\mathbf{y}}(\mathbf{z}_S)$.

The SHAP value ϕ_i for feature g_i is calculated as a weighted average, over all coalitions, of the difference in model output when g_i is either included or excluded in the coalition [13]:

$$\phi_i(f, \mathbf{y}) = \sum_{S \subseteq G \setminus g_i} \frac{|S|(m - |S| - 1)!}{m!} [f_{\mathbf{y}}(\mathbf{z}_S) - f_{\mathbf{y}}(\mathbf{z}_{S \cup g_i})]. \quad (1)$$

Previous methods applying SHAP to time series consider features g_i to be data points or subsequences [2, 15, 17]. In this paper, we follow Sun et al. [21] and instead consider a set of high-level concepts C as features, i.e. $G = C$.

We define a concept c_i as a feature of a higher-level of abstraction than input features [9]. Define $C = \{c_1, c_2, \dots, c_m\}$ to be a set of concepts, with coalition S defined to be a subset of concepts, i.e. $S \subseteq C$. Furthermore, \mathbf{z}_S denotes the masked model input given S . From (1), it follows that the SHAP values ϕ_i for concept c_i are given by:

$$\phi_i(f, \mathbf{y}) = \sum_{S \subseteq C \setminus c_i} \frac{|S|(m - |S| - 1)!}{m!} [f_{\mathbf{y}}(\mathbf{z}_S) - f_{\mathbf{y}}(\mathbf{z}_{S \cup c_i})]. \quad (2)$$

Note, that while this article will only consider univariate time series, the above definition is applicable to any data for which a set of concepts can be defined.

To implement C-SHAP, two algorithms are needed: a method for concept construction to obtain the set of concepts C and a masking algorithm suitable for these concepts. The masking algorithm is needed to obtain the masked input \mathbf{z}_S . Features are masked by setting them to an uninformative value. Point-based attribution methods mask features by changing the values of specific points [2, 15, 17]. C-SHAP masks features by transforming concepts. As a result, the global series may be transformed. This will be illustrated in Section 3.2.

3 Implementation

In this section, we present an implementation of the conceptualization of C-SHAP. As previously discussed, for this we need to select a method for concept construction and an algorithm for masking concepts.

3.1 Concept construction

In literature, concept-based XAI methods automatically detect and construct concepts [8, 21], or work with manually constructed concepts [11]. Time series are less visually intuitive than images, complicating the automatic construction of human-interpretable concepts. Therefore, we choose manual construction in this demonstration. Specifically, we use time series decomposition. The advantage of decomposition is that components can be independently adjusted. As a result, the SHAP values do not depend on the order in which concepts are masked.

3.1.1 Decomposition

Given a time series $\mathbf{y}(t) \in \mathbb{F}^n$, we define its decomposition into a set of components $C = \{\mathbf{y}_1(t), \mathbf{y}_2(t), \dots, \mathbf{y}_n(t) : \mathbf{y}_i(t) \in \mathbb{F}^n\}$ such that:

$$\mathbf{y}(t) = \bigotimes_{i=1}^n \alpha_i \mathbf{y}_i(t), \quad (3)$$

where \otimes is an aggregation operator and α_i are the component weights. For this implementation of C-SHAP, the set of components is considered to be our concept set C in (2).

3.1.2 Prophet

To finalize the concept construction, a method for component decomposition needs to be selected. In this paper, the decomposition is based on Prophet [22], due to its suitability for the experiment in Section 4. Prophet is a forecasting model based on time series decomposition. In this article, we repurpose the underlying decomposition.

Given a time series $\mathbf{y}(t)$, we define the aggregation operator \otimes as the sum operator and we define all weights as unit weights, i.e. $\alpha_i = 1$, then the Prophet decomposition is defined as follows [22]:

$$\mathbf{y}(t) = \mathbf{y}_{\text{growth}}(t) + \mathbf{y}_{\text{season}}(t) + \mathbf{y}_{\text{holiday}}(t) + \varepsilon_t. \quad (4)$$

The growth component $\mathbf{y}_{\text{growth}}(t)$ captures trends in the data. It is modelled either as non-linear or linear growth. Non-linear growth is modelled as a system with a maximum capacity for its application at Facebook [22]. For both types of growth, Prophet accounts for changes in distribution by selecting automatic change points after which the growth parameters may be shifted.

The seasonal component $\mathbf{y}_{\text{season}}(t)$ captures periodic seasonality. Prophet uses Fourier series to approximate periodic components of the data. To account for multi-period seasonality, subcomponents can be defined. For each selected period P_i , where $i \in \{1, 2, \dots, M\}$, a subcomponent $\mathbf{y}_{\text{season}}^{P_i}$ is defined, such that:

$$\mathbf{y}_{\text{season}}(t) = \sum_{i=1}^M \mathbf{y}_{\text{season}}^{P_i}.$$

Each component $\mathbf{y}_{\text{season}}^{P_i}(t)$ is modelled as:

$$\mathbf{y}_{\text{season}}^{P_i}(t) = \sum_{n=1}^{N_i} \left(a_n \cos \left(\frac{2\pi n t}{P_i} \right) + b_n \sin \left(\frac{2\pi n t}{P_i} \right) \right),$$

where N_i specifies the number of Fourier components to include. Increasing N_i results in more higher-frequency components being included.

The holiday component $\mathbf{y}_{\text{holiday}}(t)$ takes care of non-periodically recurring events. Specifically, in case of financial data, holidays. In this paper this component is not considered, since it requires prior knowledge on special events.

Prophet assumes the component ε_t to be noise; this requires that all patterns in $\mathbf{y}(t)$ are captured by the other components. However, in case of an imperfect decomposition, ε_t will contain leftover patterns. Therefore, we do not assume ε_t to be noise and instead refer to the component as $\mathbf{y}_{\text{other}}(t)$. When calculating the SHAP values, the SHAP value for $\mathbf{y}_{\text{other}}(t)$ will show how much leftover patterns contribute to the model prediction. In conclusion, we adapt (4) to:

$$\mathbf{y}(t) = \mathbf{y}_{\text{growth}}(t) + \sum_{i=1}^M \mathbf{y}_{\text{season}}^{P_i}(t) + \mathbf{y}_{\text{other}}(t). \quad (5)$$

3.2 Component masking

Besides the concept construction, a masking algorithm is needed to obtain the SHAP values. In this section, we propose a masking algorithm for concepts constructed using time series decomposition. This method of masking is not limited to Prophet, but works for any decomposition as defined in (3).

For the set of concepts $C = \{c_1, c_2, \dots, c_n\}$, the concept c_i can be masked by replacing the associated component $\mathbf{y}_i(t)$ in (3) by an uninformative value. In this paper, uninformative values are simulated by repeatedly replacing $\mathbf{y}_i(t)$ by components sampled from the training data².

Concretely, N samples are randomly sampled from the training data as masking samples. For each masking sample $\mathbf{y}^j(t)$, where $j \in \{1, 2, \dots, N\}$, the same procedure is repeated. The decomposition in (3) is applied to obtain the decomposition $\{\mathbf{y}_2^j(t), \dots, \mathbf{y}_n^j(t)\}$. Then, each component $\mathbf{y}_i(t)$ in (3) that is not in coalition S is replaced by $\mathbf{y}_i^j(t)$. The model is applied to the result. After repeating this for all masking samples, the masked output is approximated by averaging the resulting model outputs:

$$f_{\mathbf{y}}(\mathbf{z}_S) \approx \frac{1}{N} \sum_{j=1}^N f \left(\sum_{\{i: \mathbf{y}_i \in S\}} \mathbf{y}_i(t) + \sum_{\{i: \mathbf{y}_i \notin S\}} \mathbf{y}_i^j(t) \right). \quad (6)$$

Pseudocode for the algorithm is given in Algorithm 1.

Algorithm 1 Component masking algorithm

```

1: maskingSamples  $\leftarrow$  sample(trainingSamples,  $N$ )
2: componentsInput  $\leftarrow$  decompose( $\mathbf{y}(t)$ ) ▷  $\mathbf{y}(t)$  is the input sample
3: initialize maskedOutputArray
4: for  $j = 1, 2, \dots, N$  do
5:   componentsMask  $\leftarrow$  decompose(maskingSamples[ $j$ ])
6:   initialize sampleMasked
7:   for  $i = 1, 2, \dots, n$  do ▷  $\otimes$  is an aggregation operator cf. (3)
8:     if  $i$  such that componentsInput [ $i$ ]  $\in S$  then ▷  $S$  is a coalition
9:       sampleMasked  $=$  sampleMasked  $\otimes$  componentsInput [ $i$ ]
10:    else
11:      sampleMasked  $=$  sampleMasked  $\otimes$  componentsMask [ $i$ ]
12:    maskedOutputArray[ $j$ ]  $=$  model(sampleMasked)
13: maskedOutput  $=$  mean(maskedOutputArray)

```

4 Demonstration

In this section, we provide a proof of concept to show how C-SHAP can be applied to time series data. For the demonstration, the implementation described in the previous section is applied to a use case from the energy forecasting domain.

4.1 Experimental setup

4.1.1 Data

The dataset used is the Hourly Energy Consumption dataset³. The data set contains the hourly energy consumption of PJM Interconnection LLC, an organization that coordinates the movement of electricity in regions of the United States of America. The data from the PJM West region for 1 April 2001 to 3 August 2018 is used. Consumption is given per hour in megawatts (MW).

The first 80% of the data, until 27 April 2015 14:00 is used as training data, the other 20% is used as test data. Both datasets are segmented into windows of 169 hours, where consecutive windows overlap 25 hours. This results in 4575 training samples and 1138 test samples. The task for the AI model is to use the consumption of the first 168 hours (7 days) to forecast the consumption at hour 169. The data is scaled by the mean and standard deviation of the training data.

²<https://shap.readthedocs.io>

³<https://www.kaggle.com/datasets/robikscube/hourly-energy-consumption>

4.1.2 Model

For the forecasting task, a GRU-based regression model [4] was trained. The model consists of two unidirectional GRU layers with a hidden dimension of 64, followed by a single fully connected linear layer. The model has a root mean squared error (RMSE) of 56.01 MW on the training data and 62.24 MW on the test data. For reference, the mean value of the training data is 5616.06 MW.

4.1.3 SHAP

For the calculation of the SHAP values, we construct four concepts using the Prophet decomposition given in (5). The concept “Growth” is defined as the $y_{\text{growth}}(t)$ component, “Other” is defined as the $y_{\text{other}}(t)$ component. The “Daily” and “Weekly” concepts are defined as seasonal subcomponents, we define $y_{\text{daily}}(t)$ with $P_{\text{daily}} = 1$ day and $y_{\text{weekly}}(t)$ with $P_{\text{weekly}} = 7$ days.

The decomposition is implemented using the Prophet package⁴. For extraction of the growth component, piecewise linear growth is fitted using the automatic parameters. For the seasonal components, the number of components for the Fourier series are set to $N_{\text{daily}} = 3$ and $N_{\text{weekly}} = 3$.

For the masking of concepts, component masking is applied as described in Section 3.2. For the algorithm, we choose to sample $N = 100$ training samples without replacement as masking samples.

The computational cost of SHAP scales exponentially with the number of features. This is why generally methods for estimation [13, 21] are used to calculate the SHAP values. Since only four concepts are used, the SHAP values are calculated directly.

4.2 Results

The SHAP values were calculated for the samples in the test data. In Figure 2, the mean absolute SHAP value for each concept is shown as a global explanation of the model. The explanation indicates that the “Growth” concept has the largest influence on the global predictions of the model.

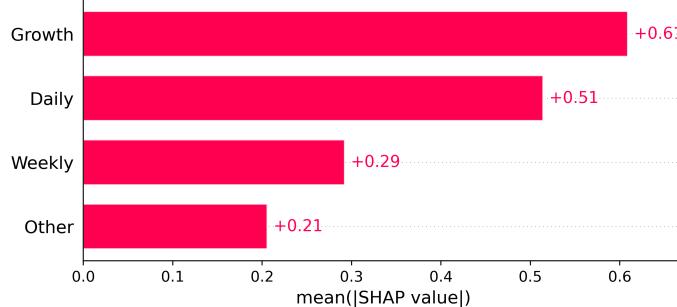


Figure 2: Mean absolute SHAP values over all test data.

Selected local explanations are presented in Figure 3, Figure 4 and Figure 5. For the visualization of the local explanations, the decomposition of the sample is shown on the right. The original signal is shown on top, its decomposition into concepts is shown below. On the left, the corresponding SHAP values are visualized by arrows [13]. From top to bottom, the SHAP value visualization starts at the base value, the expected output value of the featureless model, and shows how each concept additively contributes to the final prediction value.

In Figure 4, the explanation for the sample with the highest value for the concept “Growth” is given. The high value for the concept “Growth” corresponds to the substantial growth present in the signal. In Figure 5, the explanation for the sample with the lowest value for the concept “Other” is given.

⁴<https://facebook.github.io/prophet>

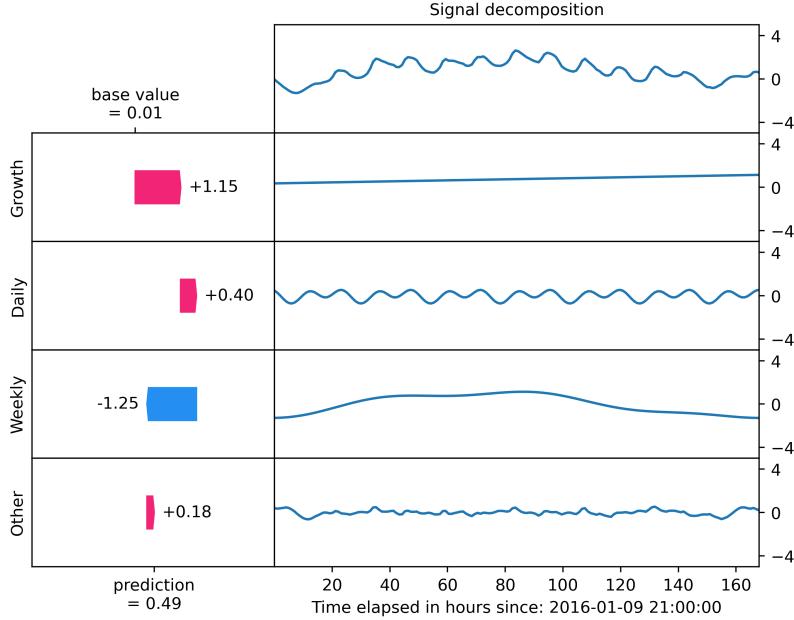


Figure 3: Local explanation of a forecast for a sample from January 2016. The “Growth” and “Weekly” components have the highest contribution.

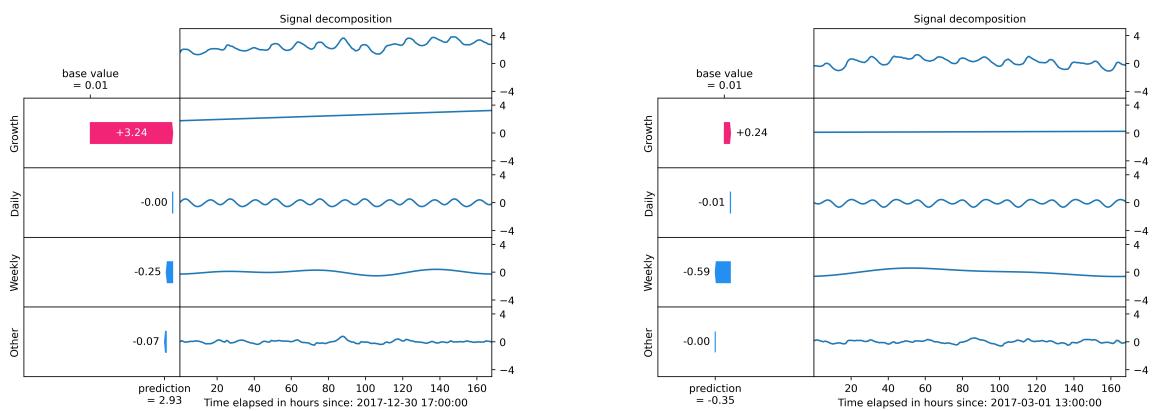


Figure 4: Local explanation of the sample with the highest SHAP value for “Growth”.

Figure 5: Local explanation of the sample with the lowest SHAP value for “Other”.

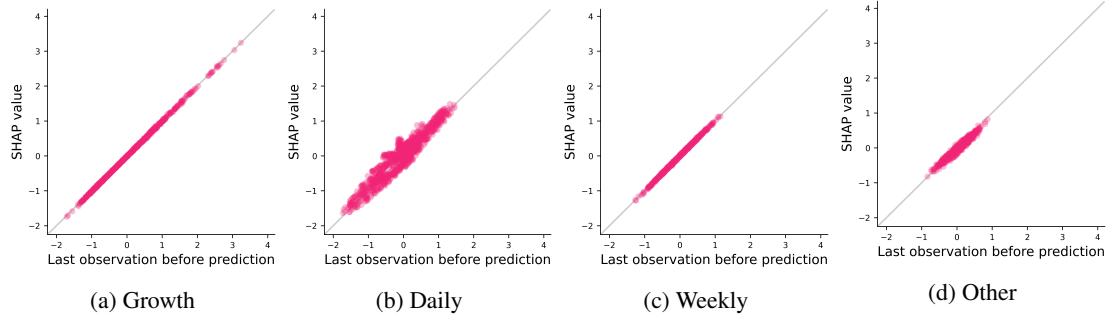


Figure 6: Relation between the SHAP value of each concept and the last value of the component before model prediction for all test samples.

5 Discussion

Looking closely at the local explanations presented in Section 4.2, there tends to be a correlation between the final observed value for each component and their respective SHAP value. In Figure 6, for each concept, the last observed value (at hour 168) is shown in relation to the corresponding SHAP value. This figure shows a relation between the last observed values and the SHAP values.

This behaviour can be explained due to the nature of the use case. For a high-quality forecasting model, where the underlying data shows non-erratic behaviour, the predicted value of the model will lie in the neighbourhood of the last observed value. As a consequence, given a masked input $\mathbf{z}_S = (z_1^S, z_2^S, \dots, z_n^S)$, the model output $f_y(\mathbf{z}_S)$ will lie in the neighbourhood of z_n^S . Due to the additive nature of the decomposition (see (6)), if component $\mathbf{y}_i(t) = (y_1^i, y_2^i, \dots, y_n^i)$ for concept c_i has a relatively large value y_n^i compared to the masking samples, $z_n^{S \cup c_i}$ will be larger than z_n^S . Therefore, due to the model output lying in the neighbourhood of the input, $f_y(\mathbf{z}_{S \cup c_i})$ will be larger than $f_y(\mathbf{z}_S)$. As a consequence, cf. (2), the corresponding SHAP value ϕ_i will be relatively large.

In other words, for this use case, the SHAP value ϕ_i correlates with the last observed component value y_n^i due to the nature of the data and the simple decomposition used for the components. It does illustrate that the model works as expected. Similar correlation is not expected when moving to more complex datasets, different concepts, or other tasks such as classification, non-forecasting regression or long-term forecasting.

Yeh et al. [24] define a completeness score to verify whether a set of components cover model reasoning. For our approach this is not necessary, since any information not captured by the first components will end up in the “Other” component. As a result, the SHAP value for the “Other” concept represents the completeness of the other components.

In the global explanation, see Figure 2, the “Other” concept has a non-negligible value. Additionally, the correlation in Figure 6d for the “Other” component should not be present if the initial components capture all model reasoning. Therefore, we conclude that the model is using more information than is captured by the considered components. In future research, the decomposition could be expanded with more components. Examples of such components could be outliers or variance shifts.

Another limitation is that the Prophet decomposition (5) only captures repetitions with fixed periodicity, e.g. seasonality. This seasonality does not exist in all time series data. Specifically in sensor data, for repeating patterns, there are often fluctuations in period over time. Such repetitions cannot be captured by Prophet. Future work will aim to expand the concept construction to sensor data. This construction will not necessarily limit itself to decomposition. Decomposition is one viable approach to concept construction, however other approaches should also be considered. For example, filters may have the potential to capture concepts such as frequency variability.

We argue that the method proposed in this paper does not alleviate the need for point- or subsequence-based explanations [23], rather the two should be used complementary. Point- or subsequence-based explanations show which regions of the time series are used in model reasoning, concept-based explanations show which

aspects of the patterns are used. In future research, a hybrid approach could be explored, in which the influence of concepts over time is determined.

The conceptualization presented in this paper is not necessarily restricted to univariate time series. The C-SHAP for definition can be expanded to multivariate series. Additionally, C-SHAP can be expanded to other modalities, Sun et al. [21] applied a similar method for image data. Other potential modalities are point cloud and video data.

6 Conclusion

Concept-based explanations can uncover global patterns used by black-box AI models for time series tasks. Furthermore, they may provide explanations in terms that match human understanding. In this article, we defined C-SHAP as an approach for concept-based explanations for time series. To illustrate how C-SHAP can be applied, an implementation using time series decomposition was presented. To demonstrate its effectiveness, C-SHAP was applied to an energy forecasting use case. The implementation in this article offers a starting point for universal high-level time series explanations. Concept-based SHAP is generally applicable to any time series use case and method of concept construction. In future work, more advanced approaches to concept construction should be explored to increase the impact of this method. Furthermore, the approach can be refined by investigating methods to capture shifts in concepts over time.

Acknowledgements

This publication is part of the project ZORRO with project number KIC1.ST02.21.003 of the research programme Key Enabling Technologies (KIC) which is partly financed by the Dutch Research Council (NWO). This research is part of the SPRONG DEMAND. This research is partly financed by Taskforce for Applied Research SIA, part of the Dutch Research Council (NWO).

References

- [1] Amina Adadi and Mohammed Berrada. Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI). *IEEE Access*, 6:52138–52160, 2018.
- [2] João Bento, Pedro Saleiro, André F Cruz, Mário AT Figueiredo, and Pedro Bizarro. TimeSHAP: Explaining Recurrent Models through Sequence Perturbations. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pages 2565–2573, 2021.
- [3] Alexander Brenner, Felix Knispel, Florian P Fischer, Peter Rossmanith, Yvonne Weber, Henner Koch, Rainer Röhrlig, Julian Varghese, and Ekaterina Kutafina. Concept-based AI interpretability in physiological time-series data: Example of abnormality detection in electroencephalography. *Computer Methods and Programs in Biomedicine*, 257:108448, 2024.
- [4] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734. Association for Computational Linguistics, 2014.
- [5] Jonathan Crabbé and Mihaela van der Schaar. Concept Activation Regions: A Generalized Framework For Concept-Based Explanations. *Advances in Neural Information Processing Systems*, 35:2590–2607, 2022.
- [6] Arun Das and Paul Rad. Opportunities and Challenges in Explainable Artificial Intelligence (XAI): A Survey, 2020.
- [7] Chirag Deb, Fan Zhang, Junjing Yang, Siew Eang Lee, and Kwok Wei Shah. A review on time series forecasting techniques for building energy consumption. *Renewable and Sustainable Energy Reviews*, 74:902–924, 2017.

- [8] Amirata Ghorbani, James Wexler, James Y Zou, and Been Kim. Towards automatic concept-based explanations. *Advances in neural information processing systems*, 32, 2019.
- [9] Yash Goyal, Amir Feder, Uri Shalit, and Been Kim. Explaining Classifiers with Causal Concept Effect (CaCE). *arXiv preprint arXiv:1907.07165*, 2019.
- [10] Zohaib Jan, Farhad Ahamed, Wolfgang Mayer, Niki Patel, Georg Grossmann, Markus Stumptner, and Ana Kuusk. Artificial intelligence for industry 4.0: Systematic review of applications, challenges, and opportunities. *Expert Systems with Applications*, 216:119456, 2023.
- [11] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV). In *International conference on machine learning*, pages 2668–2677. PMLR, 2018.
- [12] Ferdinand Küsters, Peter Schichtel, Sheraz Ahmed, and Andreas Dengel. Conceptual Explanations of Neural Network Prediction for Time Series. In *2020 International joint conference on neural networks (IJCNN)*, pages 1–6. IEEE, 2020.
- [13] Scott M Lundberg and Su-In Lee. A Unified Approach to Interpreting Model Predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 4768–4777, 2017.
- [14] Diana Mincu, Eric Loreaux, Shaobo Hou, Sebastien Baur, Ivan Protsyuk, Martin Seneviratne, Anne Mottram, Nenad Tomasev, Alan Karthikesalingam, and Jessica Schrouff. Concept-based model explanations for electronic health records. In *Proceedings of the Conference on Health, Inference, and Learning*, pages 36–46, 2021.
- [15] Karim El Mokhtari, Ben Peachey Higdon, and Ayşe Başar. Interpreting financial time series with SHAP values. In *Proceedings of the 29th annual international conference on computer science and software engineering*, pages 166–172, 2019.
- [16] Mohammad Amin Morid, Olivia R Liu Sheng, and Joseph Dunbar. Time Series Prediction Using Deep Learning Methods in Healthcare. *ACM Transactions on Management Information Systems*, 14(1):1–29, 2023.
- [17] Amin Nayebi, Sindhu Tipirneni, Chandan K. Reddy, Brandon Foreman, and Vignesh Subbian. Window-SHAP: An efficient framework for explaining time-series classifiers based on Shapley values. *Journal of Biomedical Informatics*, 144:104438, 2023.
- [18] Eleonora Poeta, Gabriele Ciravegna, Eliana Pastor, Tania Cerquitelli, and Elena Baralis. Concept-based Explainable Artificial Intelligence: A Survey, 2023.
- [19] Gesina Schwalbe and Bettina Finzel. A comprehensive taxonomy for explainable artificial intelligence: a systematic survey of surveys on methods and concepts. *Data Mining and Knowledge Discovery*, 38(5):3043–3101, 2024.
- [20] Lloyd S. Shapley. A Value for N-Person Games. *Contribution to the Theory of Games*, 2, 1952.
- [21] Ao Sun, Pingchuan Ma, Yuanyuan Yuan, and Shuai Wang. Explain any concept: Segment anything meets concept-based explanation. *Advances in Neural Information Processing Systems*, 36, 2024.
- [22] Sean J Taylor and Benjamin Letham. Forecasting at scale. *The American Statistician*, 72(1):37–45, 2018.
- [23] Andreas Theissler, Francesco Spinnato, Udo Schlegel, and Riccardo Guidotti. Explainable AI for time series classification: a review, taxonomy and research directions. *IEEE Access*, 10:100700–100724, 2022.
- [24] Chih-Kuan Yeh, Been Kim, Sercan Arik, Chun-Liang Li, Tomas Pfister, and Pradeep Ravikumar. On Completeness-aware Concept-Based Explanations in Deep Neural Networks. *Advances in neural information processing systems*, 33:20554–20565, 2020.