State Graph based Explanation Approach for Black-box Time Series Model

Yiran Huang*
Chaofan Li*
yhuang@teco.edu
chaofan.li@kit.edu
Karlsruhe Institute of Technology
Karlsruhe, Baden-Wuerttemberg, Germany

Hansen Lu
Karlsruhe Institute of Technology
Karlsruhe, Baden-Wuerttemberg, Germany

Till Riedel
Karlsruhe Institute of Technology
Karlsruhe, Baden-Wuerttemberg, Germany

Michael Beigl
Karlsruhe Institute of Technology
Karlsruhe, Baden-Wuerttemberg, Germany

ABSTRACT

In recent years, there has been a growing trend in the utilization of Artificial Intelligence (AI) technology to construct human-centered systems that are based on implicit time series information, ranging from contextual recommendations on smartwatches to human activity recognition on production workshop. Despite the advantages of these systems, the opaqueness and unpredictability of these systems for users has elicited concerns. To mitigate these issues, time-series explanation methods have been proposed. However, existing methods only focus on the segment importance of the instance to be explained and ignore its chronological nature. In this paper, we propose a novel explanation method named State-graph Based eXplanation Artificial Intelligent (SBXAI), which exhibits the sequential relationship between time periods through directed circular graphs while emphasizing the importance of each time period in an instance. Our proposed method was evaluated on 20 time-series datasets, and the results showed that the explanations provided by SBXAI are consistent with the behavior of the AI model in making predictions.

KEYWORDS

Explainability in IR, Time series processing, Markov Model

1 INTRODUCTION

With the rapid development of the always-on network technology and micro-/nano-electromechanical systems, the use of live sensor information presents an increasingly important role in human-centered Artificial Intelligent (AI) system in our daily life.Schilit [17] already described in 1994 the different types of implicit interactions with computer systems stemming from such contextual information, which since then has led to a wide range of context-aware recommender systems from context-aware advertising to adjusting music playlists based on the user behaviors [11].

Typical examples of devices enabling such implicit interactions are smartwatches and mobile phones equipped with numerous sensors. These devices collect time-series data on daily human activities and provide relevant recommendations based on the insight derived from the collected data. However, since the models used by these devices are becoming increasingly complex, it is becoming more challenging to explain the underlying logic behind these recommendations. Therefore, there is a requirement for explanatory methods for time-series AI models, more precisely, for methods that clarify how AI models process the sequential relationships inherent in the data to derive a context-based recommendation.

To this day, explainability research has gained much attention and progress in the computer vision domain [23]. However, the unique characteristics of time series data, which are the foundation for many context-aware information retrieval models, make it challenging to directly apply these advancements to time-series explanations. There are several reasons for this, with the two most notable being: *(i)* for humans, images often possess inherent semantics, whereas time-series data is incomprehensible without domain knowledge, *(ii)* the features of the image data are typically related to the values and their numerical differences, while the features of time series data are usually characterized by the values and their chronological order.

Several methodologies have emerged for interpreting time series models in recent years. Approaches [3, 14] evaluate the significance of input data by introducing perturbations. Schlegel et al. [19] adapts the LIME approach to the time series domain, utilizing six distinct segmentation methods and elucidating the target model through the training of local models with the generated segmentations. Doddaiah et al. [6] broadens this method to encompass multi-class forecasting issues. While these approaches offer model explanation, they are restricted to the importance of input segments. This constraint appears incongruous with the predictive foundations inherent to numerous time series models. For instance, in predicting a head nod using data from a gravitational acceleration sensor affixed to the head, the assessment should encompass the entire sequence of acceleration increasing in the direction of gravity, returning to zero, increasing in the opposite direction, and returning to zero once more. Although the value at any given moment might exhibit a linear correlation with the predicted outcome, it cannot be deemed a comprehensive explanation for the predicted result.

*Both authors contributed equally to this research.

To this end, we proposed a novel time series explanation method named the State-graph Based eXplanation Artificial Intelligent (SBXAI). This method utilizes Bayesian optimization to aggregate data that are adjacent to each other in the given example to create multiple, more understandable data units (state). Furthermore, it applies Directed Circular Graphs (DCG) to visualize the sequential relationships between states and used it to explain the model decision.

Our contributions can be summarized as *(i)* We applied DCG to explain the decision of an AI model, which demonstrates how the sequential relationship within the given example determines the decision made by the AI model. To the best of our knowledge, this is the first approach to explain a time-series model in terms of chronological order. *(ii)* We utilized the Bayesian optimization algorithm to group adjacent data, creating states that are more understandable by humans. *(iii)* We provided scalable algorithm architecture in GitHub [1], which facilitates the subsequent research and development.

2 RELATED WORKS

Theissler et al. [23] divided the existing time series explanation methods into three main categories: Time Points-Based Explanations, Subsequences-Based Explanations, and Instance-Based Explanations. Among them, Time Points-Based Explanations usually assign a weight for each time point in the input time series data, reflecting how much the value at this time point contributes to the final decisions of the model [9, 13, 18, 25]. Subsequences-Based Explanations explain the model by figuring out the input sub-segments most representative of the model's decisions. Such sub-segments could either be real-valued subsequences directly extracted from the raw time series [2, 12, 20] or discretized representations obtained through an aggregate algorithm [15, 16, 24]. Instance-Based Explanations, on the other hand, rely on the whole time series instance to show the reasons why the model makes a judgment, such as features extracted from the entire time series instance [7, 21], the most exemplifying examples of a particular classification made by the model [8, 22], counter-examples that lead to changes in classification through minimal modification [5, 10], etc.

It is not difficult to see that Time Points-Based Explanations and Subsequences-Based Explanations have difficulty showing the effect of chronological order when presenting explanations. On the other hand, Instance-Based Explanations are based on implicit assumptions, e.g., the feature or exemplifying examples they found are themselves explainable. And this assumption, as stated earlier, is not always true. Time-series data (especially longer or with more complex trends ones) are often difficult to understand, sometimes even for experts with domain knowledge.

3 METHODOLOGY

The preceding discourse led to the inspiration that a successful explanation method for time series models should prioritize the following two elements: *(i)* the ability to effectively analyze and visually demonstrate the impact of the chronological order of the input value on the model prediction, *(ii)* the avoidance of presenting overly length time series segments that are difficult for humans to understand and single value that contains little information. The proposed explanation method is constructed based on these two principles, and its structure is depicted in Figure 1(a). The framework consists of three modules: the Segment & Clustering Module, the Perturbation Module, and the Explanation Module.

3.1 Segment & Clustering Module

The Segment & Clustering Module is responsible for dividing all the time series from given data into small segments and categorizing them according to their similarity. This module aims to decompose the whole time series into a series of smaller, more straightforward patterns that are easily understandable to humans by avoiding showing explanations that contain long segments. This module includes two essential hyperparameters, namely the length of the segment and the number of clusters. They are fine-tuned using the Bayesian optimization algorithm. As shown in Figure 1(b), two processes are presented to address the two common scenarios in the time series Black-box explanation, respectively.

The process on the left side of Figure 1(b) pertains to the scenario where the original dataset used to train the Black-box model is available during the explanation stage. For example, the model's trainer wants to get an explanation of the misclassified samples, to optimize the accuracy of the trained model further or to enhance its robustness against adversarial attacks. In this case, the original dataset is processed through the Segment & Clustering Module to generate the state sequence representation of the data. At the same time, the original dataset is also fed to the target Black-box model to get its prediction. Subsequently, the state sequence representation of the original data set serves as the input, while the output of the Black-box model serves as the target to train a Model. The model is then evaluated using the accuracy metric. The evaluation result is used as the objective of the Bayesian optimization algorithm. This is because the optimal hyperparameter combination is expected to result in the highest accuracy, thereby minimizing the information loss during segmentation and clustering.

The process on the right side of Figure 1(b) pertains to the scenario where the original dataset used to train the Black-box model is not available during the explanation stage. For example, when users of a Black-box model have concerns regarding the output. In this case, we process the time series data (hereinafter referred to as Interested Data Entry) that the user wants to explain through the Segment & Clustering Module to obtain the state sequence representation of this entry. Then, the state sequence representation and the original data are shuffled in the same way to get two perturbation datasets: the state sequence perturbation dataset and the original value perturbation dataset. The latter dataset is fed into the Black-box model to produce the perturbation prediction. Finally, The state sequence perturbation and the generated perturbation prediction are used to train a classification model. The model's accuracy is the target of Bayesian optimization, as in the other process.

3.2 Perturbation Module

The task of the perturbation module is to shuffle the state sequence representation (exchange position of two states) of the Interested

[1] http:

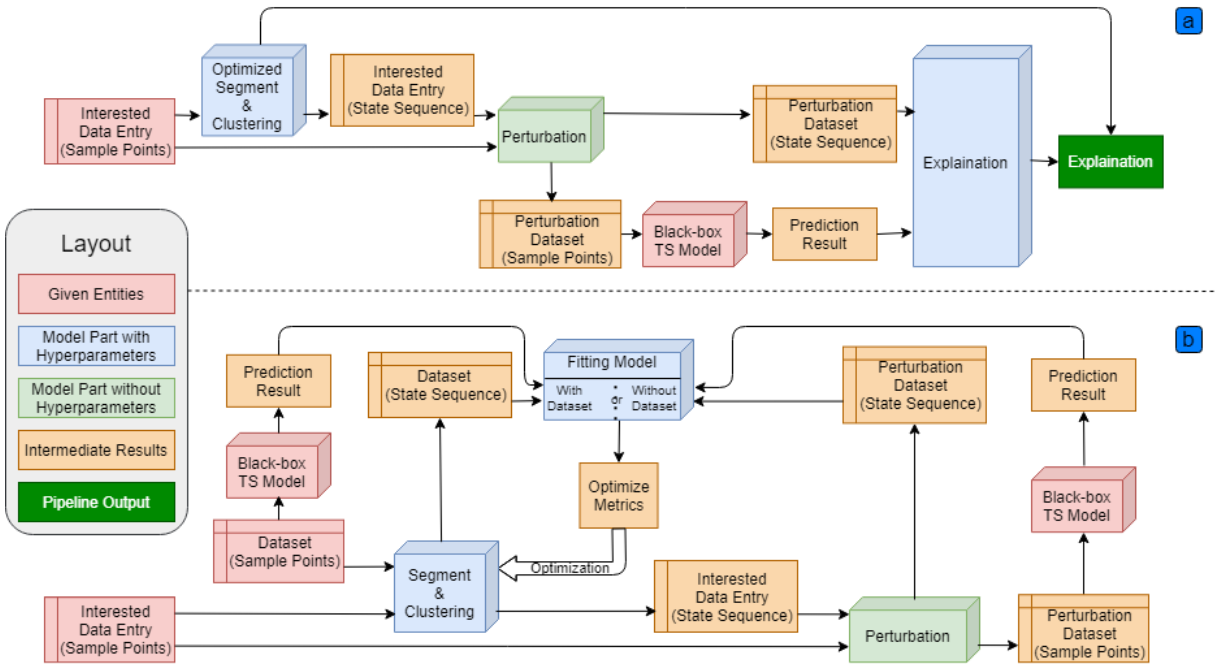

Figure 1: (a). Description of the Whole Model Pipeline. (b). Details of the Hyperparameter Optimization Procedure for the Segment & Clustering Module.

Data Entry obtained from the previous module (Segment & Clustering Module) to get a perturbation dataset. At the same time, with the help of the one-to-one correspondence between the state and the original data, the original representation of the Interested Data Entry is also shuffled in the same way. Thus, we could get the prediction of the Black-box model to the perturbation dataset. These two will serve as inputs to the next module.

3.3 Explanation Module

As the output of the previous module, we performed various chronological perturbations to the Interested Data Entry and obtained the prediction results of the Black-box model for these perturbations. By analyzing the response of the Black-box model to different perturbations, we can explain the behavior of the Black-box model. That's the task of the Explanation Module. In this module, we fit an explainable model to the behavior of the Black-box model. Thus, the explanation of the fitted model can be used to explain the Black-box model.

In our study, we choose Hidden Markov Model (HMM) as the explainable model because its visualization is straightforward and clear. Taking an instance from 'Allgesturewiimote' dataset 'Pick up' Class and a Long Short Term Memory (LSTM) black box as an example, the final presented explanation consists of three parts. The first is the correspondence between the states generated by the Segment & Clustering Module and the original data features (see Figure 2(c)). Each state describes a simple trend of data change, thus making it easy to understand. For example, state 0 here indicates no change, while states 1 and 3 distinguish between getting smaller and getting larger. Each state consists of multiple single values, and its complexity is determined by the length of the state. The second part of the explanation pertains to the importance of different features. This importance is obtained at the state level by using an existing counterfactual-based explanation method [19] and displayed through a histogram superimposed on the state transition curve of the Interested Data Entry, as shown in Figure 2(b). The explanation identifies the initial segment of the instance (state 1) as crucial and assigns different importance values to it. This assignment of importance is likely a result of the "deactivating segment" operation, or perturbation, in the counterfactual-based explanation method. Typically, this operation is carried out by substituting the value of the selected segments with non-informative values. However, when applied to time series data, this operation can introduce new trends in data variation, leading to potential errors in explanation. The final part of the explanation centers on the significance of various state transitions, which is demonstrated by the transitions graph of the Hidden Markov Model (HMM) presented in Figure 2(a). In this graph, each node represents a state, and each edge represents a transition, with each edge corresponding to a value that describes the importance of the corresponding transition. In this specific example, the edge from state 3 to state 1 is assigned a value of 1, whereas the value from state 1 to itself is 0.8. Notably, there is no edge from state 3 to itself, indicating that the corresponding value is 0. This leads us to conclude that the classification of this example is based on the sensor value being maintained in a stable state for an extended duration following a brief upward trend.

Dataset	Number Class	Sequence Length	Random-Fe	Random-Seq	Fe	SBXAI-Seq
AllGestureWiimote	10	vary	10	7	86	**89**
Car	4	577	16	10	63	**68**
YoGA	2	426	19	12	76	**81**
ShapesAll	60	512	13	17	50	**72**
PigAirwayPressure	50	2000	3	9	18	**95**
Mallat	8	1024	10	5	**98**	88
InlineSkate	7	1882	20	17	42	**85**
CricketY	12	300	11	16	69	**83**
RefrigerationDevices	3	720	11	10	46	**64**
MixedShapesRegularTrain	5	1024	3	2	37	**65**
BirdChicken	2	512	13	28	79	**89**
WordSynonyms	25	270	12	12	57	**64**
DodgerLoopGame	2	288	0	20	42	**72**
FreezerRegularTrain	2	301	16	15	45	**96**
EthanolLevel	4	1751	11	8	96	**100**
LargeKitchenAppliances	3	720	6	8	56	**72**
FiftyWords	50	270	8	17	74	**75**
ArrowHead	3	251	16	20	74	**84**
EOGHorizontalSignal	12	1250	18	14	70	**80**
ACSF1	10	1460	0	1	17	**90**

Table 1: Attack Success Rate (ASR) of different modification methods.

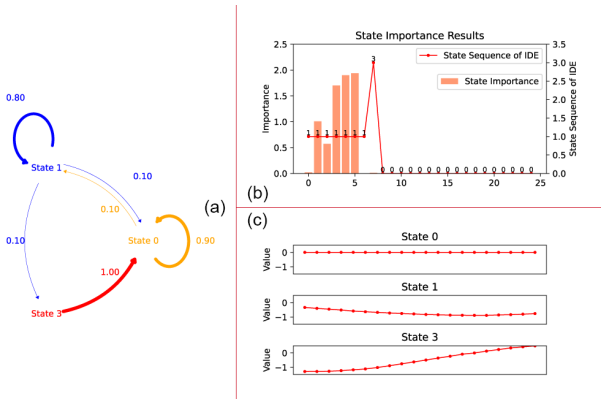

Figure 2: (a). Transitions graph of HMM, showing the importance of different state transitions. (b). The state sequence of the interested data entry (IDE) and the histogram showing the feature importanc. (c). The Correspondence between the state and the original data. The x-coordinates in Figures (b), (c) both represent timestamp.

4 EXPERIMENTS & EVALUATION

The proposed method seeks to explain the decision of the Black-box model by considering two elements: *(i)* the content of the individual generated state, and *(ii)* the sequential relationships between states. To show how accurately the explanation found by the proposed method fits the Black-box model's behavior toward the prediction, we design an experiment. Given a dataset, we randomly sample items from the dataset and explain them with the selected methods. Then we modify the item according to the explanation. Suppose the Black-box prediction changed after the modification. We record the modification as a success. For each dataset, we repeat this process 100 times to get an average attack success rate (ASR), which indicates the importance of the rules broken by the modification. The performance of the following modification methods are compared: *(i)* replace the value of a random position in the item with the least important state found by the proposed method in the given item (Random-Fe); *(ii)* exchange the position of random two states in the given sequence (Random-Seq); *(iii)* remove important feature found by TS-MULE [19] with proposed Segment Module (TS-MULE-Fe); *(iv)* exchange the position of the state pair considered to be the most important by the proposed method (SBXAI-Seq). For example, given state sequence [abcba], the proposed method found out that the sequence relationship 'ab' is important to identify the class of the item. We exchange the position of the states and transform the original sequence to [bacba]. We try to emphasize the meaning of the successful attack by limiting the strength of the applied modification. During the experiment, we utilize the Hyperopt [1] package to do the optimization and set its hyperparameter max_iter to 100 and the optimization algorithm to 'tpe.suggest'.

For the reliability of the results, we conducted experiments on the UCR datasets [4] with various numbers of classes and sequence lengths. The Black-box model used in the experiment is constructed with Long Short Term Memory layers. To the best of our knowledge, no heuristic method employing sequential relations has been used to explain black-box models. Therefore, we did not compare the proposed method with other sequential-based explanatory methods in the experiment. The search space and the Black-box models are presented in *githublink*.

As summarized in Table 1, the element found by the two different modification methods, TS-MULE-Fe and SBXAI-Seq, have a substantial impact on the Black-box prediction. The average attack success rate for modifying the state order is 79.9%, while that for eliminating important status is 60.4%. These results suggest that the explanation generated by the proposed method is reasonable. And we can conclude that the prediction of a time series depends not

only on its states, but also on the sequential relationship between them. For time series data, sequential changes between states have a more pronounced effect on decisions than the removal of individual states. In addition, we find that some models are strongly influenced by the sequential relationship and are almost unaffected by the individual states, e.g., 'PigAirwayPressure'. This is likely because the states considered important appear multiple times in the same item. Besides, no significant correlation was found between the attack success rate, number of classes, and sequence length.

5 CONCLUSION

In this paper, we propose a method to explain the time series Black-box model in terms of sequential relations and experimentally demonstrate the correctness of the explanation obtained by this method. However, there is still room for improvement, both in experiments and in the further development of the algorithm. For example, all the Black-box models have the same structure in the experiment. The impact of the proposed method on models of different architectures should be verified. Besides, we believe that further improvements could be made to the algorithms by enlarging the search space, e.g., adding more segmentation and clustering methods.

ACKNOWLEDGMENTS

REFERENCES

[1] James Bergstra, Dan Yamins, David D Cox, et al. 2013. Hyperopt: A python library for optimizing the hyperparameters of machine learning algorithms.

[2] Sohee Cho, Wonjoon Chang, Ginkyeng Lee, and Jaesik Choi. 2021. Interpreting internal activation patterns in deep temporal neural networks by finding prototypes. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 158–166.

[3] Jonathan Crabbé and Mihaela Van Der Schaar. 2021. Explaining time series predictions with dynamic masks. In *International Conference on Machine Learning*. PMLR, 2166–2177.

[4] Hoang Anh Dau, Anthony Bagnall, Kaveh Kamgar, Chin-Chia Michael Yeh, Yan Zhu, Shaghayegh Gharghabi, Chotirat Ann Ratanamahatana, and Eamonn Keogh. 2019. The UCR time series archive. *IEEE/CAA Journal of Automatica Sinica* 6, 6 (2019), 1293–1305.

[5] Eoin Delaney, Derek Greene, and Mark T Keane. 2021. Instance-based counterfactual explanations for time series classification. In *International Conference on Case-Based Reasoning*. Springer, 32–47.

[6] Ramesh Doddaiah, Prathyush Parvatharaju, Elke Rundensteiner, and Thomas Hartvigsen. 2023. Explaining Deep Multi-Class Time Series Classifiers. (2023).

[7] Dominique Gay, Romain Guigourès, Marc Boullé, and Fabrice Clérot. 2013. Feature extraction over multiple representations for time series classification. In *International Workshop on New Frontiers in Mining Complex Patterns*. Springer, 18–34.

[8] Alan H Gee, Diego Garcia-Olano, Joydeep Ghosh, and David Paydarfar. 2019. Explaining deep classification of time-series data with learned prototypes. In *CEUR workshop proceedings*, Vol. 2429. NIH Public Access, 15.

[9] Yiran Huang, Nicole Schaal, Michael Hefenbrock, Yexu Zhou, Till Riedel, Likun Fang, and Michael Beigl. 2022. McXai: Local model-agnostic explanation as two games. *arXiv preprint arXiv:2201.01044* (2022).

[10] Isak Karlsson, Jonathan Rebane, Panagiotis Papapetrou, and Aristides Gionis. 2020. Locally and globally explainable time series tweaking. *Knowledge and Information Systems* 62, 5 (2020), 1671–1700.

[11] Álvaro Lozano Murciego, Diego M Jiménez-Bravo, Adrián Valera Román, Juan F De Paz Santana, and María N Moreno-García. 2021. Context-aware recommender systems in the music domain: A systematic literature review. *Electronics* 10, 13 (2021), 1555.

[12] Dominique Mercier, Andreas Dengel, and Sheraz Ahmed. 2021. PatchX: Explaining Deep Models by Intelligible Pattern Patches for Time-series Classification. In *2021 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–8.

[13] Mohsin Munir, Shoaib Ahmed Siddiqui, Ferdinand Küsters, Dominique Mercier, Andreas Dengel, and Sheraz Ahmed. 2019. Tsxplain: Demystification of dnn decisions for time-series using natural language and statistical features. In *International conference on artificial neural networks*. Springer, 426–439.

[14] Prathyush S Parvatharaju, Ramesh Doddaiah, Thomas Hartvigsen, and Elke A Rundensteiner. 2021. Learning saliency maps to explain deep time series classifiers. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 1406–1415.

[15] Om P Patri, Anand V Panangadan, Charalampos Chelmis, and Viktor K Prasanna. 2014. Extracting discriminative features for event-based electricity disaggregation. In *2014 IEEE Conference on Technologies for Sustainability (SusTech)*. IEEE, 232–238.

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

[17] Bill Schilit, Norman Adams, and Roy Want. 1994. Context-aware computing applications. In *1994 first workshop on mobile computing systems and applications*. IEEE, 85–90.

[18] Udo Schlegel, Hiba Arnout, Mennatallah El-Assady, Daniela Oelke, and Daniel A Keim. 2019. Towards a rigorous evaluation of xai methods on time series. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*. IEEE, 4197–4201.

[19] Udo Schlegel, Duy Lam Vo, Daniel A Keim, and Daniel Seebacher. 2022. Ts-mule: Local interpretable model-agnostic explanations for time series forecast models. In *Machine Learning and Principles and Practice of Knowledge Discovery in Databases: International Workshops of ECML PKDD 2021, Virtual Event, September 13-17, 2021, Proceedings, Part I*. Springer, 5–14.

[20] Pavel Senin and Sergey Malinchik. 2013. Sax-vsm: Interpretable time series classification using sax and vector space model. In *2013 IEEE 13th international conference on data mining*. IEEE, 1175–1180.

[21] Vera Shalaeva, Sami Alkhoury, Julien Marinescu, Cécile Amblard, and Gilles Bisson. 2018. Multi-operator decision trees for explainable time-series classification. In *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*. Springer, 86–99.

[22] Wensi Tang, Lu Liu, and Guodong Long. 2020. Interpretable time-series classification on few-shot samples. In *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–8.

[23] Andreas Theissler, Francesco Spinnato, Udo Schlegel, and Riccardo Guidotti. 2022. Explainable AI for Time Series Classification: A review, taxonomy and research directions. *IEEE Access* (2022).

[24] Lexiang Ye and Eamonn Keogh. 2009. Time series shapelets: a new primitive for data mining. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. 947–956.

[25] Linjiang Zhou, Chao Ma, Xiaochuan Shi, Dian Zhang, Wei Li, and Libing Wu. 2021. Salience-cam: Visual explanations from convolutional neural networks via salience score. In *2021 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–8.