Interpreting Deep Temporal Neural Networks by Selective Visualization of Internally Activated Nodes

Sohee Cho*
Wonjoon Chang
sohee.cho@kaist.ac.kr
one_jj@kaist.ac.kr
KAIST
Daejeon, Republic of Korea

Ginkyeng Lee*
gin908@unist.ac.kr
UNIST
Ulsan, Republic of Korea

Jaesik Choi†
jaesik.choi@kaist.ac.kr
KAIST
Daejeon, Republic of Korea

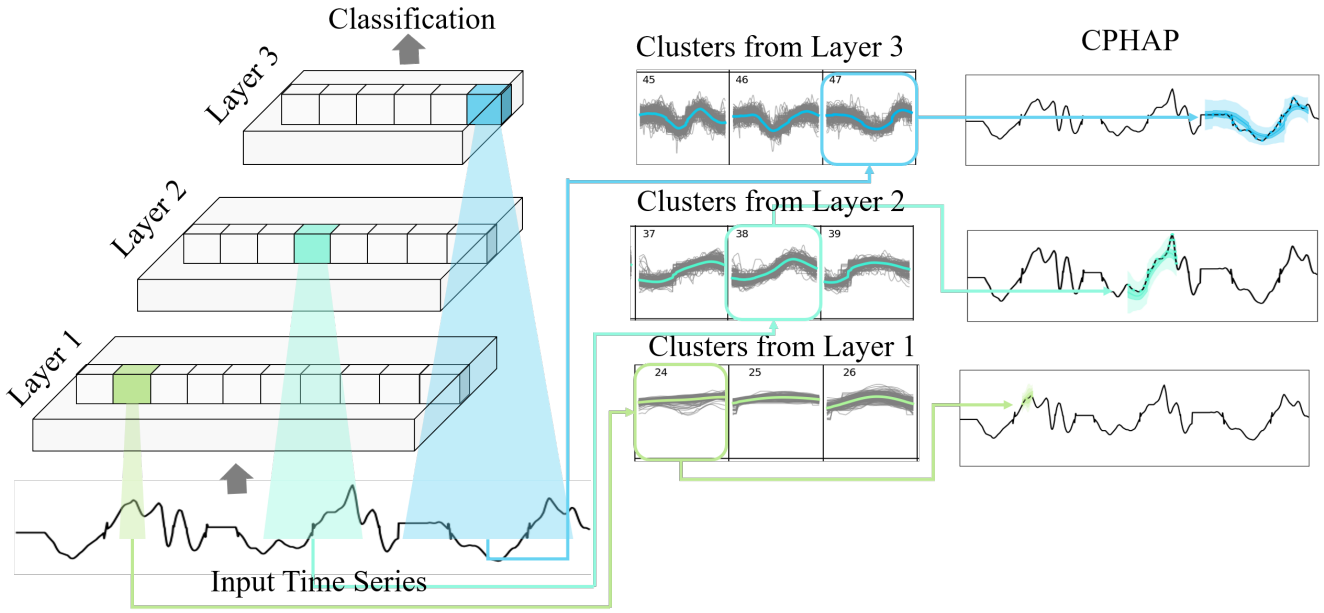

Figure 1: The process of CPHAP

ABSTRACT

Recently deep neural networks have demonstrated competitive performance in classification and regression tasks for sequential data. However, it is still hard to understand which temporal patterns the internal channels of deep neural networks see in sequential data. To address this issue, we propose a new framework to visualize temporal representations learned in deep neural networks without hand-crafted segmentation labels. Our framework extracts highly activated temporal regions and characterizes them as representative temporal patterns. Furthermore, our framework shows the representative temporal pattern with the uncertainty. It enables users to identify whether the input has been observed frequently in the training data.

CCS CONCEPTS

• **Computing methodologies** → **Temporal reasoning**.

KEYWORDS

Time Series, Clustering, Input Attribution, Deep Convolution Neural Network

ACM Reference Format:
Sohee Cho, Wonjoon Chang, Ginkyeng Lee, and Jaesik Choi. 2020. Interpreting Deep Temporal Neural Networks by Selective Visualization of Internally Activated Nodes. In *MileTS '20: 6th KDD Workshop on Mining and Learning from Time Series, August 24th, 2020, San Diego, Claifornia, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/1122445.1122456

*Both authors contributed equally to this research.
†Corresponding Author

MileTS '20, August 24th, 2020, San Diego, California, USA

ACM ISBN 978-1-4503-9999-9/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

The amount of temporal data has greatly increased due to the use of efficient and diverse automatic-information systems such as manufacturing sensors, stock database systems and healthcare wearable devices. Recent deep learning models enable users to extract appropriate features from such data and make decisions with high accuracy. As a result, to utilize such vast amounts of temporal data, demand for the application of deep learning models in the industry has grown rapidly. However, most industrial fields require transparency in decision-making when using deep learning models. Thus, they are still hesitant to adopt AI systems due to the lack of interpretability in their internal processes.

Nowadays many AI researchers have tried to understand the decision process of deep learning models. Interpretable artificial intelligence methods explain and interpret decisions of complex systems with illustrative or textual descriptions. These approaches for deep learning are classified into explaining input attribution methods based on relevance score [4, 9, 11, 12] , gradient-based methods [15–17] , explaining internal nodes methods [2, 3] , explaining through attention methods [5, 7] and generating explanations methods [1].

However, those methods mainly focus on the image domain and there have been few efforts to apply interpretation techniques to time series data. Objects in images can be easily recognized visually, since there is a lot of human-annotated segmentation information for image datasets, which are not provided for most time series datasets. Thus, it is hard to find semi-global shapes that a neural network is looking at in a time series input due to a lack of temporally segmented annotation data.

To address this issue, we suggest a new framework to visualize temporal representations by clustering temporal patterns of highly activated nodes. Our framework has the following contributions:

- Without hand-crafted segmentation labels, our framework identifies representative temporal patterns that activate each channel of convolutional neural networks most.
- Our framework matches perceived sub-sequences and the closest representative temporal patterns; It makes users easy to verify whether the patterns are commonly observed or deviated from trained data.
- Our framework provides the uncertainty of the representative temporal patterns. This uncertainty implies the potential variance of input shapes that activate the channel.

Consequently, our framework can provide insight into the decision-making process of internal channels in the neural network through visualization. This visualization shows general shapes that the network recognizes with uncertainty. To the best of our knowledge, this is first attempt to extract and visualize patterns that highly activate interval nodes. It helps users to understand how deep neural networks learn time series data intelligibly.

2 CLUSTERED PATTERN OF HIGHLY ACTIVATED PERIOD (CPHAP)

2.1 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) [8] can learn features from the input considering spatial correlation using filters. Channels from different learned filters reflect the features of the input with activation maps. A partial region of each channel receives information from only a restricted subset of the output of previous layers. This input subset is called a receptive field, which represents what the channel capture from the input.

The CNN architecture is widely known to be used for image domains, but it can also be utilized for time series data in order to extract local information from close data points which are sequentially correlated. In fact, Recurrent Neural Networks (RNNs), which share weight parameters over temporal steps, have been successfully used in natural language processing where data of interest has discrete values. However, CNN-based methods, such as ST-GCN [19] and WaveNet [13], have demonstrated outstanding performance on classification and regression tasks of time series data with continuous values. Furthermore, analyzing the role of hidden units in a RNN is difficult and complicated due to RNN's recursive structure. Therefore, we choose CNN-based models to figure out the roles of channels.

2.2 Extracting important input sub-sequences

We think that analyzing highly activated parts of activation maps play an important role to understand temporal patterns in neural networks. So, our algorithm interprets the decisions of neural networks by extracting highly activated nodes in a channel[1] and visualizing sub-sequences of the input data that contribute to the highly activated nodes.

Highly Activated Period (HAP): *Highly Activated* means that certain nodes in a channel have bigger values than the channel's threshold. We calculate a threshold $T_{j,k}$ satisfying $P(a_{j,k} > T_{j,k}) = 0.05$ to select the highly activated nodes for each channel k at layer j, where $T_{j,k}$ is a threshold of channel k at hidden layer j, and $a_{j,k}$ is the distribution of the kth channel activations at hidden layer j.

Then, we define Highly Activated Nodes (HAN) as a set of nodes of channel k at layer j that satisfy $A_{j,k}[i] > T_{j,k}$ where i is a node in channel k and $A_{j,k}[i]$ is the activation value of a node i. The union of the input receptive fields of nodes in HAN is **Highly Activated Period (HAP)**. This period is the important sub-sequence that we are looking for. The detailed process is described in Algorithm 1. Note that $N_{j,k}$ in Algorithm 1 denotes the number of nodes in channel k at hidden layer j.

2.3 Patternizing representative sub-sequences by clustering

So far, we have found important sub-sequences to figure out what the individual channels in CNNs are looking at in the time series data. Now, we characterize these sub-sequences and assign the general shapes to them. In this paper, we use Self Organizing Map (SOM) to characterize temporal patterns in a HAP [2].

SOM is an unsupervised learning method by mapping high-dimensional data to a low-dimensional map. During the training procedure, the weight vectors in the map are trained to move toward

[1]Network Dissection[2] and CAM call this channel as "unit". In this paper, we need to distinguish between a channel and the basic elements in a channel, so we call an activation map as a "channel" and an element of channel as a "node".

[2]We experimentally try various clustering methods, including comparing several clustering methods, including K-means clustering, Gaussian Mixture Models (GMMs), K-shape clustering [14] and SOM. The outputs of each method are provided in the appendix.

Algorithm 1 Extract important input sub-sequences

Input: data **X**, Trained CNN model
for j **do**
  for k **do**
    Compute $a_{j,k}$ = the distribution of the kth channel activations at hidden layer j in the CNN model for data **X**
    Find a threshold $T_{j,k}$ satisfying $P(a_{j,k} > T_{j,k}) = 0.05$
Define HAP_list$_j$ = [ ] for all j
for $x \in$ **X** **do**
  for j **do**
    for k **do**
      $A_{j,k}$ = the activation values of the kth channel at hidden layer j in the CNN model
      $\mathbf{HAN}_{j,k}(x) = \{i \in [1, N_{j,k}] \mid A_{j,k}[i] > T_{j,k}\}$ where i is a node in channel k
      $\mathbf{HAP}_{j,k}(x)$ = {Temporal indices in input receptive field(i) $\mid i \in \mathbf{HAN}_{j,k}(x)$}
      HAP_list$_j$.append($\mathbf{HAP}_{j,k}(x)$)
Train SOM clustering with HAP_list
Choose data x, layer j, channel k
Compute $p = \mathbf{HAP}_{j,k}(x)$
Compute cluster c_p = SOM(p)
$\mathbf{CPHAP}_p$ = mean($\{p' \in$ HAP_list$_j \mid$ SOM(p') = $c_p\}$)
$\mathbf{UNCERTAINTY}_p$ = variance($\{p' \in$ HAP_list$_j \mid$ SOM(p') = $c_p\}$)

the input data with keeping the topology of the map space. The trained weight vectors works as cluster groups. We use a 8×8 map, so a total of 64 groups show various patterns and each group has low variance among elements in the same group. Furthermore, SOM represents the relationship among cluster groups because near cluster groups show similar patterns on the map. After clustering, we compute an average of each cluster group to assign a pattern.

CPHAP: A Clustered Pattern of Highly Activated Period is an average over time axis of a cluster group C given HAP_list which means a list of temporal sequences that activates nodes.

$$\mathbf{CPHAP} = mean(\{p \in \text{HAP\_list} \mid SOM(p) = C\}) \quad (1)$$

Note that HAP_list should be defined by layer, since channels in different layer have different lengths of input receptive field.

Our framework also illustrates the uncertainty in CPHAP. The uncertainty in CPHAP is the variance for the posterior probability of the corresponding the input sub-sequences. It indicates the degree of certainty from the cluster that the detected sub-sequence belongs to. Thus, given new data, the user can determine whether the detected sub-sequence is a common pattern or an abnormal case. The detailed process is described in Algorithm 1 and shown in Figure 1.

3 EXPERIMENTAL RESULTS

Dataset We use three time series open dataset for experiments; *UWaveGestureLibraryAll* [10] is a set of eight simple gestures generated from accelerometers, *Smartphone Dataset for Human Activity Recognition (HAR)* [6] is a smartphone sensor dataset recording human perform eight different activities.

Model We use a temporal CNN which is composed of three convolution layers followed by pooling layers, and one fully connected layer. ReLU function is used as the activation function in each hidden layer. Batch size and training epoch are 64 and 500 respectively. We also apply our framework to ResNet [18] with 9 layers. Above figures show CPHAP results for various models and datasets. Further details are provided in the appendix.

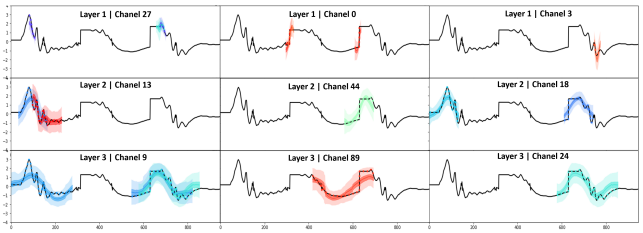

Figure 2: [CPHAP Result in CNN Model, UWave Data 1247] Our framework visualizes temporal representations learned from temporal deep neural networks.

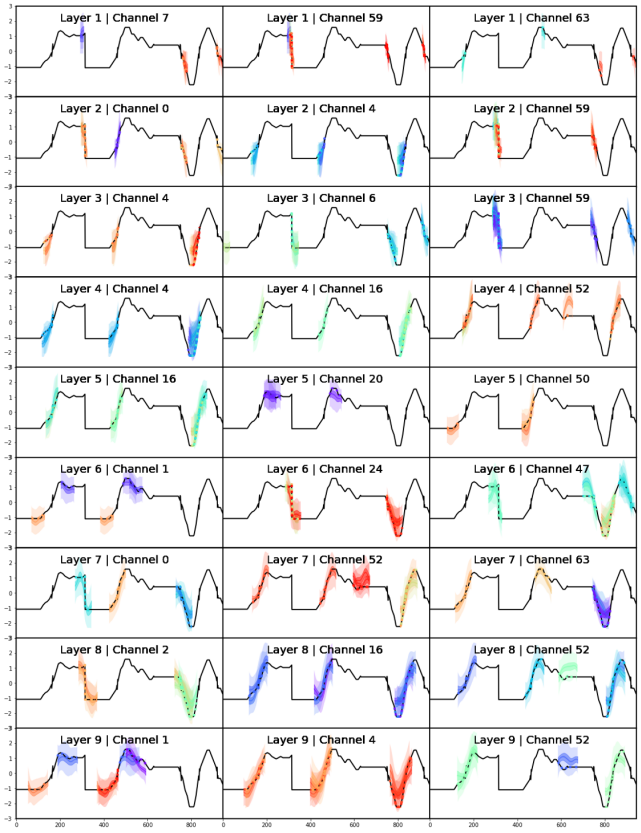

Figure 3: [CPHAP Result in ResNet Model, UWave Data 571, Sensor 0] Given the data sample, channels in lower layers tend to focus on rapid changes or inflection points. The channels in the higher layers recognize extreme changes in softer patterns.

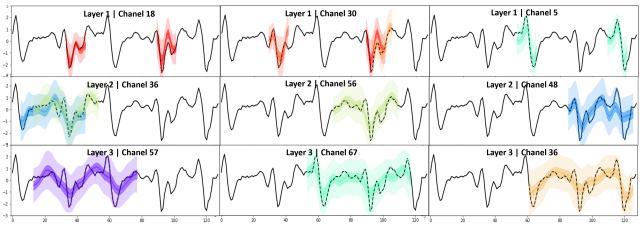

Figure 4: [CPHAP Result in CNN Model, HAR Data 1247 Sensor1] Since HAR data has a different length of sequence from Uwave data, the CPHAP for HAR has also the different lengths of patterns even the same model structure.

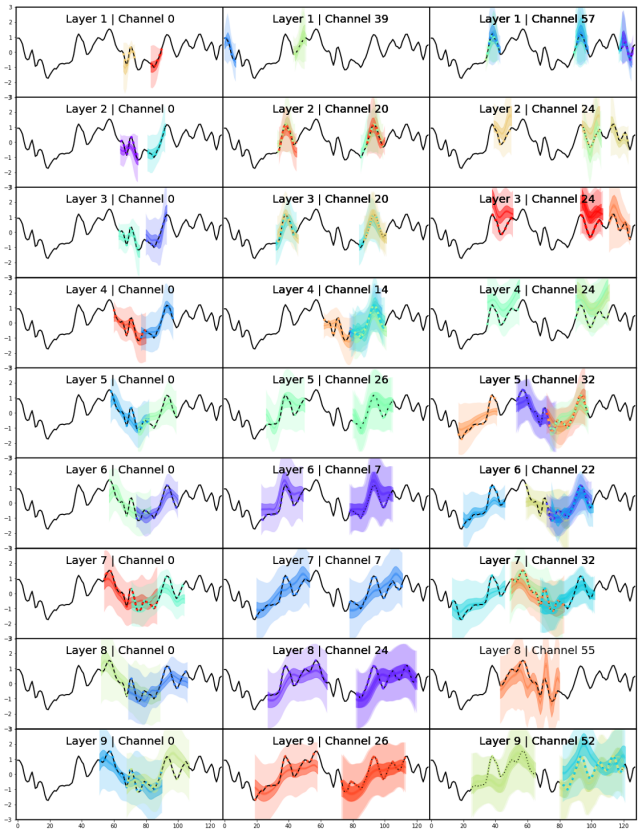

Figure 5: [CPHAP Result in ResNet Model, HAR Data 1613, Sensor8] Patterns from lower layers, such as layer 1, 2, and 3, reflect local changes like short concave shapes. On the other hands, patterns from higher layers, such as layer 7, 8, and 9, capture global changes like slow upward trends.

3.1 Comparisons with other interpretable methods

We try to compare our framework with Layer-wise Relevance Propagation (LRP) [11]. LRP explains the model's decision by decomposing an output of the model into individual pixels according to the amount of contribution to the output. Since LRP does not provide channel-specific analysis, we apply LRP to highly activated nodes of a specific channel which we call *Channel-LRP*.

Figure 6 depicts how each method visualize the role of the channel for time series data. The first plot in Figure 6 represent the upsampled activation map for the channel and the second plot represents thresholded regions. In the third plot, colorized points in Channel-LRP show importance of each point from the input with color variation. However, the results from these methods cannot explain how the neural networks recognize selected points or regions due to the lack of human-annotated segmentation data. On the other hand, CPHAP provides clear visualization for important sub-sequences and explains how each sub-sequence is perceived by neural networks. Furthermore, our framework shows the uncertainty for each perceived pattern, which implies the potential range of inputs activating corresponding channel. Note that each time point in the pattern has different degree of uncertainty.

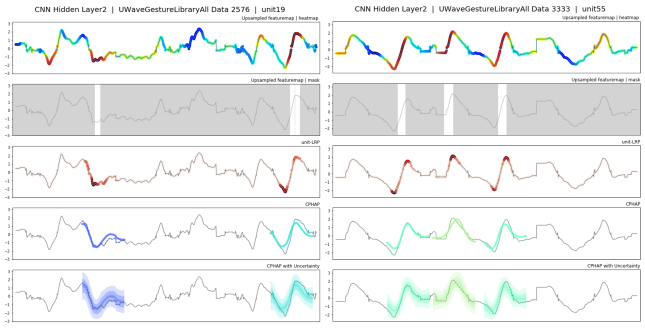

Figure 6: Comparison with other interpretable Methods

3.2 Perturbation analysis

We quantitatively evaluate the importance of these sub-sequences selected by our framework. Our assumption is that the activation values in the channel are robust to perturbations in the input with preserving the detected important sub-sequences.

We randomly apply perturbations while preserving specific regions selected by each method and observe changes in activations. The methods include CPHAP, LRP and Random. In the case of LRP, we calculate relevance scores and time points which have high relevance scores are selected to be preserved. No points are preserved when we use Random method. We apply three kinds of perturbations and observe changes in activations of the channel: Gaussian, Inverse, Zero. Each perturbation replaces the points with randomly sampled values from $Normal(0, 3)$, reversed values and zero values.

Table 1 shows how the results from CPHAP are less vulnerable to perturbations than the other methods on Uwave dataset. The first row of each table denotes the ratio of perturbations. Our performance is always better when using Gaussian perturbations. Though activation difference from CPHAP is slightly worse than LRP when using Zero perturbation with ratio of 1.0, the difference is very close to zero and CPHAP surpasses LRP method in most of the other cases. It implies that CPHAP can select critical regions from the input better than the other methods.

Table 1: The Robustness of internal activations when applying perturbations while preserving sub-sequences selected by CPHAP. The values in the table mean how less activations in channels change with CPHAP in perturbation compared to other methods (mean ± std).

Gaussian	0.2	0.6	1.0
vs Random	10.06±7.08%	11.4±7.15%	12.4±7.38%
vs LRP	5.72±3.96%	4.92±2.04%	5.11±1.91%
Inverse	0.2	0.6	1.0
vs Random	17.28±6.89%	10.44±10.91%	15.03±9.1%
vs LRP	4.93±3.72%	0.26±3.61%	2.45±2.86%
Zero	0.2	0.6	1.0
vs Random	20.27±6.08%	19.29±6.99%	21.2±6.92%
vs LRP	5.85±3.5%	0.33±4.23%	-0.41±3.23%

3.3 CPHAP for Test Dataset

Figure 7 illustrates how CPHAP works well for test dataset. The left column shows well-matched cases with new test data. On the other hands, the middle column shows examples of less-matched patterns through red-dashed circles. Note that there are certain points where the actual data deviate from the assigned pattern for each less-matched example. The positions of these points are reflected in the visualization of our framework. The right column shows that our visualization successfully capture the positions which have high uncertainty in the actual data.

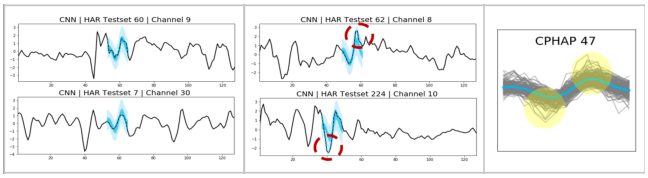

Figure 7: Comparisons between well-matched patterns and less-matched patterns

4 CONCLUSION

We propose a new method to visualize the representative temporal patterns in neural networks by breaking into the activations of channels and analyzing their receptive fields in the input space. Our framework uses highly activated temporal regions and applies clustering method to patternize these regions in order to obtain the general shapes without human-segmented information. Also, we calculate the uncertainty of each cluster, which enables users to gain insight into the type of the new input. Consequently, our work would provide intuitive visualizations for time series and inspire people to understand AI systems in real world problems.

ACKNOWLEDGMENTS

This work was supported by IITP grant funded by the Korea government(MSIT) (No.2017-0-01779, XAI) and Artificial Intelligence Graduate School Program(KAIST) (No, 2019-0-00075).

REFERENCES

[1] Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. 2016. Neural module networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 39–48.

[2] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. 2017. Network dissection: Quantifying interpretability of deep visual representations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 6541–6549.

[3] David Bau, Jun-Yan Zhu, Hendrik Strobelt, Bolei Zhou, Joshua B Tenenbaum, William T Freeman, and Antonio Torralba. 2018. Gan dissection: Visualizing and understanding generative adversarial networks. *arXiv preprint arXiv:1811.10597* (2018).

[4] Alexander Binder, Grégoire Montavon, Sebastian Lapuschkin, Klaus-Robert Müller, and Wojciech Samek. 2016. Layer-wise relevance propagation for neural networks with local renormalization layers. In *International Conference on Artificial Neural Networks*. Springer, 63–71.

[5] Edward Choi, Mohammad Taha Bahadori, Jimeng Sun, Joshua Kulas, Andy Schuetz, and Walter Stewart. 2016. Retain: An interpretable predictive model for healthcare using reverse time attention mechanism. In *Advances in Neural Information Processing Systems*. 3504–3512.

[6] Dheeru Dua and Casey Graff. 2017. UCI Machine Learning Repository. http://archive.ics.uci.edu/ml

[7] Jay Heo, Hae Beom Lee, Saehoon Kim, Juho Lee, Kwang Joon Kim, Eunho Yang, and Sung Ju Hwang. 2018. Uncertainty-aware attention for reliable interpretation and prediction. In *Advances in Neural Information Processing Systems*. 909–918.

[8] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11, 2278–2324.

[9] Hongzhi Li, Joseph G Ellis, Lei Zhang, and Shih-Fu Chang. 2018. Patternnet: Visual pattern mining with deep neural network. In *Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval*. 291–299.

[10] Jiayang Liu, Lin Zhong, Jehan Wickramasuriya, and Venu Vasudevan. 2009. uWave: Accelerometer-based personalized gesture recognition and its applications. *Pervasive and Mobile Computing* 5, 6 (2009), 657 – 675. https://doi.org/10.1016/j.pmcj.2009.07.007 PerCom 2009.

[11] Gregoire Montavon, Sebastian Lapuschkin, Alexander Binder, Wojciech Samek, and Klaus-Robert Muller. 2017. Explaining nonlinear classification decisions with deep Taylor decomposition. *Pattern Recognition* 65 (2017), 211 – 222. https://doi.org/10.1016/j.patcog.2016.11.008

[12] Woo-Jeoung Nam, Jaesik Choi, and Seong-Whan Lee. 2019. Relative Attributing Propagation: Interpreting the Comparative Contributions of Individual Units in Deep Neural Networks. *arXiv preprint arXiv:1904.00605* (2019).

[13] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. 2016. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499* (2016).

[14] John Paparrizos and Luis Gravano. 2016. k-Shape: Efficient and Accurate Clustering of Time Series. *ACM SIGMOD Record* 45, 69–76. https://doi.org/10.1145/2949741.2949758

[15] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*. 618–626.

[16] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. 2017. Learning important features through propagating activation differences. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 3145–3153.

[17] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. 2014. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806* (2014).

[18] Zhiguang Wang, Weizhong Yan, and Tim Oates. 2017. Time series classification from scratch with deep neural networks: A strong baseline. In *2017 International joint conference on neural networks (IJCNN)*. IEEE, 1578–1585.

[19] Sijie Yan, Yuanjun Xiong, and Dahua Lin. 2018. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Thirty-second AAAI conference on artificial intelligence*.

A VISUALIZATION OF CPHAP

In our figures, we visualize CPHAPs with colored lines; A black line is an input sequence, a colored dot line is an input receptive field, a colored line is CPHAP and a colored shadow means the uncertainty of the pattern.

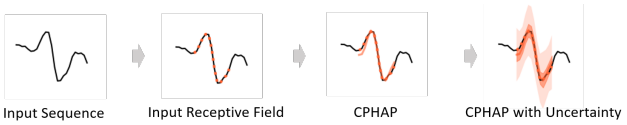

Figure 8: Visualization Process of CPHAP

Figure 9 shows that how the input receptive fields of the channel activations can have the overlapped areas. Then, some parts of patterns detected in receptive fields can also be overlapped with other patterns. As described in Figure 10, we remove some overlapped patterns for CPHAP to illustrate the important patterns more clearly.

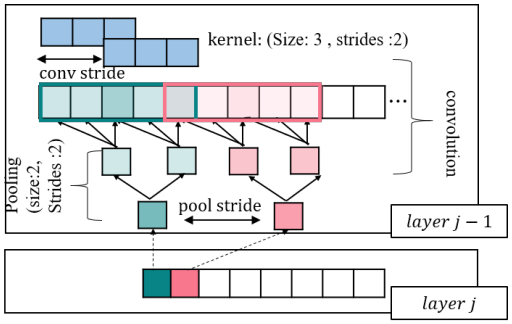

Figure 9: Receptive Field

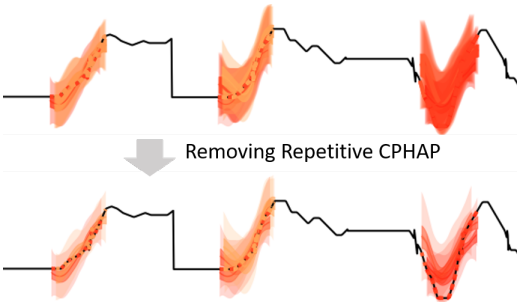

Figure 10: How to Handle Overlapped Patterns

B PATTERNIZING REPRESENTATIVE SUB-SEQUENCES BY CLUSTERING

We compare several clustering methods to suggest an optimal algorithm for characterizing input sub-sequences with a measure of shape and distribution. We applies various methods, including K-means, Gaussian Mixture Model (GMM), K-shape, and Self Organizing Map (SOM) methods. The results of clustering methods are below.

SOM is an unsupervised learning method by mapping high-dimensional data to a low-dimensional map with keeping the topology of the map space. GMM assumes that there is a certain number of Gaussian distributions, and that each of these distributions represents a cluster. K-means method classifies nearby sequences into identical clusters based on Euclidean distance only. K-shape method is similar to K-means, but specialized for time series: it compares sequences efficiently and computes centroids effectively under scaling and shift invariances.

We choose SOM for two reasons. First, the mean value of each SOM cluster seems to be an valid representation of the sub-sequences belonging to that cluster. Second, the map space of SOM represents spatial meanings. In other words, near cluster groups show similar patterns on the map. We expect this property to help to explain the role of channels in the convolutional neural network.

Figure 11 illustrate the clustering result of SOM, aligning with the map space of SOM. Also, we plot the marginalized clusters along horizontal axis and vertical axes. We can identify that near cluster groups have similar patterns. For the CNN model trained on Uwave dataset, we observe that the clusters detected by channel 1 in layer 2 are cluster 3, 9, 10, 11, 12, 16, 17 and 18 in Figure 11. These clusters are actually close together in the map space.

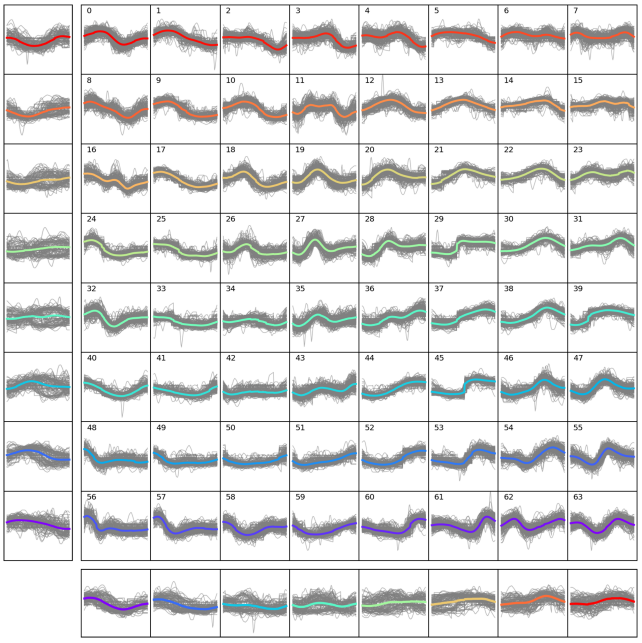

Figure 11: SOM clustering result

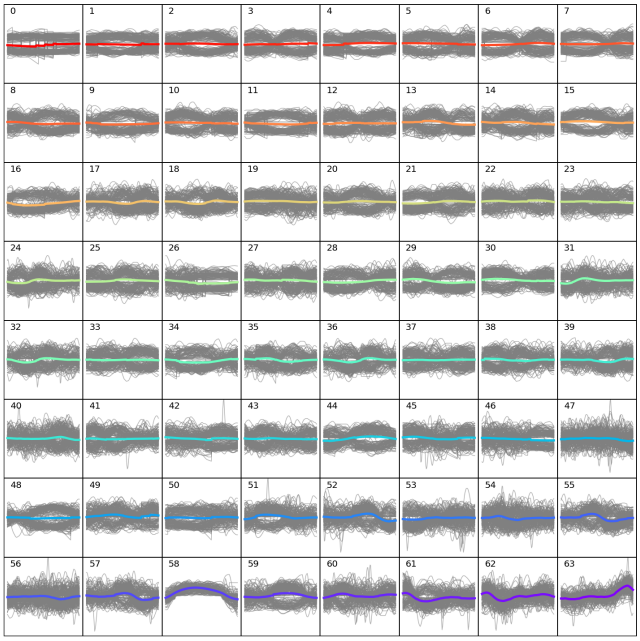

Figure 12: GMM clustering result

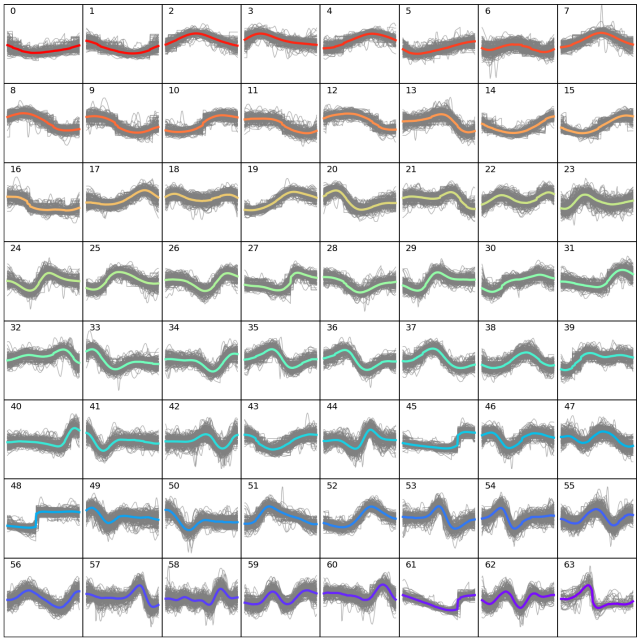

Figure 13: K-means clustering result

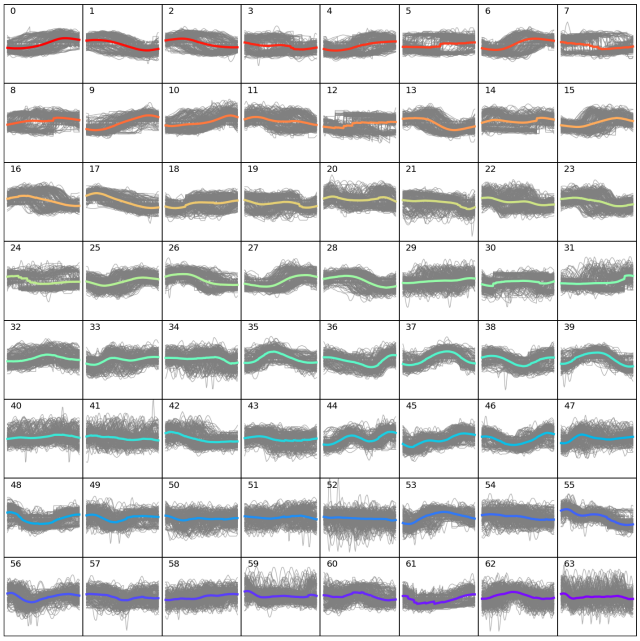

Figure 14: K-shape clustering result

C PERTURBATION ANALYSIS

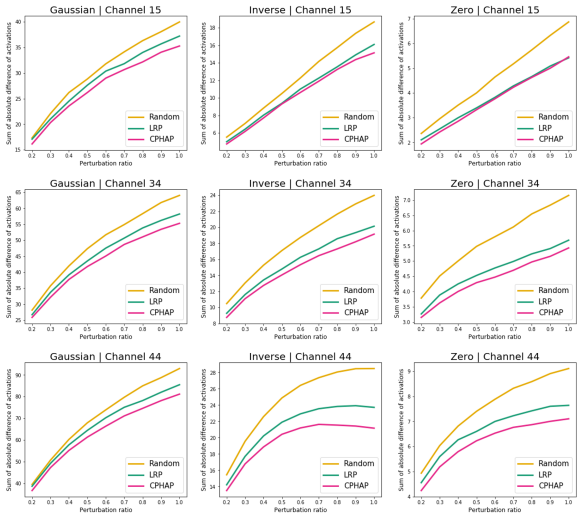

Figure 15: The results of perturbation analysis for sampled channels.

We appends samples of perturbation analysis. In Figure 15, the X-axis denotes the ratio of perturbations and the Y-axis denotes the change in the activation value according to perturbation. That is, the large change of Y value implies that the corresponding algorithm fails to select the appropriate points related to the channel.

In the case of Gaussian perturbation, preserving the temporal regions selected by CPHAP makes the channel activation more

robust for most channels. For some channels, such as channel 15, CPHAP has similar performance to LRP when using Inverse or Zero perturbation. However, CPHAP generally selects more important points than LRP to maintain the activation.

D MODEL STRUCTURE

Table 2: Datasets Information

	Time Length	Input Sensor	Class	Training Data	Test Data
UWave	945	1	9	3582	896
HAR	128	9	6	7352	2947

Table 3: CNN Structures

dataset	Test Accuracy	conv1 filter	pool1 size	conv2 filter	pool2 size	conv3 filter	pool3 size
UWave	98.2%	15	8	11	8	7	4
HAR	79.0%	7	4	5	4	3	2

Table 4: ResNet Structure

conv filter	1	2	3	4	5	6	7	8	9
filter size	15	11	9	15	11	9	15	11	9
pattern size	15	25	33	47	57	65	79	89	97