



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

Master's Thesis

Monte-Carlo Dropout based Uncertainty Analysis
in Input Attributions of Multivariate Temporal
Neural Networks

Ginkyeng Lee

Department of Computer Science and Engineering

Graduate School of UNIST

2020

Monte-Carlo Dropout based Uncertainty Analysis
in Input Attributions of Multivariate Temporal
Neural Networks

Ginkyeng Lee

Department of Computer Science and Engineering

Graduate School of UNIST

Monte-Carlo Dropout based Uncertainty Analysis
in Input Attributions of Multivariate Temporal
Neural Networks

A thesis
submitted to the Graduate School of UNIST
in partial fulfillment of the
requirements for the degree of
Master of Science

Ginkyeng Lee

01/03/2020 of submission

Approved by

Advisor

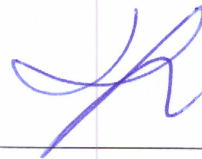
Kwang In Kim

Monte-Carlo Dropout based Uncertainty Analysis
in Input Attributions of Multivariate Temporal
Neural Networks

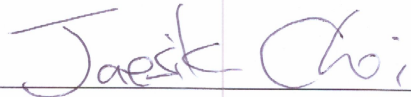
Ginkyeng Lee

This certifies that the thesis of Ginkyeng Lee is approved.


01/03/2020



Advisor: Kwang In Kim



Committee Member: Jaesik Choi



Committee Member: Sungahn Ko

Abstract

As deep learning has grown fast, so did the desire to interpret deep learning black boxes. As a result, many analysis tools have emerged to interpret it. Interpretation in deep learning has in fact popularized the use of deep learning in many areas including research, manufacturing, finance, and healthcare which needs relatively accurate and reliable decision making process. However, there is something we should not overlook. It is uncertainty. Uncertainties of models are directly reflected in the results of interpretations of model decision as explaining tools are dependent to models. Therefore, uncertainties of interpreting output from deep learning model should be also taken into account as quality and cost are directly impacted by measurement uncertainty. This attempt has not been made yet.

Therefore, we suggest Bayesian input attribution rather than discrete input attribution by approximating Bayesian inference in deep Gaussian process through dropout to input attribution in this paper. Then we extract candidates that can sufficiently affect the output of the model, taking into account both input attribution itself and uncertainty of it.

Contents

I	Introduction	1
II	Related Work	3
	2.1 Layer-wise Relevance Propagation	3
	2.2 Pattern net	4
	2.3 Bayesian Neural Networks and Monte-Carlo Dropout	6
III	Methods	8
	3.1 Input Attribution with Monte-Carlo dropout Network	8
	3.2 Input Attribution Analysis with Uncertainty	10
IV	Experiments	13
	4.1 Experimental Setup	13
	4.2 Experimental Result	14
V	Conclusion	17
	References	20
	Acknowledgements	22

List of Figures

1	Difficulty of interpreting and understanding of multivariate time series data. All input pixels will have as much relevance as they affect the output through LRP. Figure1a shows the result of LRP through MVLS GoogLeNet network with image as input, and figure 5b shows the part of result of LRP with sensors time sequence input. As mentioned in section 1, it is able to intuitively interpret the input attribution results for images, but it is not easy to interpret for multivariate temporal data when the input time is long or the number of features is large. . . .	5
2	Difference between standard dropout and MC dropout. Standard dropout in prediction or test time has single output from expected output from training. However, Monte-Carlo dropout in prediction time has multiple outputs from several model variations and averages stochastic forward passes through the model.	8
3	50 different input attributions for 3 sensors with 30 data through LRP with backward Monte-Carlo Dropout. Different input attributions are derived from randomly dropped out model structures.	10
4	Scatter plot of the mean and uncertainty of the input attribution. X axis represents mean, and Y axis represents uncertainty(standard deviation) of input attribution after MC dropout. The blue dots are the ones that have the possibility of affecting the output of the model, given uncertainty and mean of input attribution	12
5	Examples of input attributions with several sub-divided cases. X-axis represent time, Y-axis represent sensors, and Z-axis represent input attribution. The data points in the case are marked with a red dot and the uncertainty of the input attribution is represented with green shadow.	13
6	Box plot and histogram of single LRP values from existing method and box plot and histogram of mean and uncertainty values from multiple LRPs with MC dropout	18
7	Box plot and histogram of single LRP values from existing method and box plot and histogram of mean and uncertainty values from multiple LRPs with MC dropout	19

Glossary

DTD Deep Taylor Decomposition. 2, 6, 7, 9, 14

LRP Layer-wise Relevance Propagation. 1–3, 5, 7–11, 13–19

MC dropout Monte-Carlo Dropout. 2, 3, 8–13, 15, 17–19

MC dropout LRP Monte-Carlo Dropout Layer-wise Relevance Propagation. 10, 18, 19

XAI eXplainable Artificial Intelligence. 1

I Introduction

AI-based systems can perform very complex tasks and make good predictions on a wide range of topics, but these systems are often called as black box as it is hard to tracing which features influenced the prediction, and the user does not know how the decision is made from it. In these days, the direction to explain the cause of the decision of deep learning model to the user has emerged, which is called XAI, and many efforts are underway to interpret, explain, and visualize deep learning [1]. These explainable AI techniques has improved usability of deep learning in many areas including research, manufacturing, finance, and healthcare which needs relatively accurate, reliable and transparent decision making process. For example, LRP [2] explains the model's decision by decomposing the output of model to each input pixel with amount of contribution to model's output, and it is applied to assist clinicians in explaining neural network decisions for diagnosing Alzheimer's Disease(and potentially other disease) based on structural MRI data [3].

However, there is important issue we overlook : how reliable can we have this result? Uncertainties should be considered along with quality and cost as they are directly impacted by measurement uncertainty [4]. Product quality, experiment results, financial decisions, and medical diagnosis can all be directly impacted by errors introduced from the omission of measurement uncertainty. Ignoring the impact of the uncertainty might result in a higher probability of increased operating costs and failure rates.

Let's assume that a cancer diagnosis model with input (A,B,C,D) diagnosed that the patient has cancer. Let us assume we have contribution of each input to this cancer from XAI tools. If A is a major contributor to the decision to be cancer, but the uncertainty associated with this contribution is large, doctors should doubt that the A is actually important in determining cancer when reflecting the model's decisions in actual decisions. On the contrary, if B has a high contribution with a small uncertainty. doctor can be sure that B is the main decision reason for the model. Let's also consider a case with less input contribution with large uncertainty, C, and less uncertainty, D. If doctor only consider input attribution itself, he or she can conclude that only A and B were factors that influenced the outcome of cancer. But with uncertainty, physicians think that C is also likely to be a high impact candidate, so he or she can look carefully at C and incorporate the model's judgment into their actual judgment. Lastly, physicians can easily sure D as a less influential factor in model decisions.

As you can see from this representative example, it is important for model to make the right decision, but it is also important to be able to explain the obvious reason for its decision. It will help user see how acceptable the model makes decisions, and how much to reflect the results of this model in actual decisions. If there is uncertainty analysis on explaining reasons for output, it will reduce the cost and risk of doing this.

While there are many uncertainty analyses of deep learning model or output of deep learning itself, as shown in [5-7], there are still needs for uncertainty analysis of result of input attri-

bution methods. In this paper, we propose Monte-Carlo Dropout based Uncertainty Analysis in Input Attributions. We firstly use one of explainable method, called LRP using DTD for explaining output of model with input as before. But, we use MC dropout backwardly for approximation of uncertainty of input attribution in this process. Unlike the existing method which provide deterministic input attribution, the proposed method shows the distribution of input attribution, so we can see uncertainty about how each feature affect the results of the model. In addition, rather than ending with uncertainty analysis, we extract candidates that can sufficiently affect the output of the model, taking into account both input attribution itself and uncertainty of it. Although our focus is on multivariate time series data, the method is also applicable to a broad set of input data.

The rest of this paper is organized as follows: In chapter 2, we will briefly introduce background with explainable AI tools ,LRP using DTD, pattern attribution and uncertainty methods called MC dropout. In chapter 3, we will introduce our algorithm. Firstly, we will approximate Bayesian inference in deep Gaussian process using MC dropout to input attribution from LRP using DTD. Secondly, we will extract possible influential features to model's output with mean and standard deviation value of input attribution as criterion. In chapter4, we will introduce the experiment and results of 1 real-industrial data, and 1 open multi variate time series data. Lastly, we conclude our results in chapter 5.

Contributions

1. Provide a Bayesian approximate for input attribution through MC dropout.
2. Not only find input data point which has high input attribution, but also find all possible influential point which has the potential to affect the output of the model.
3. Visualize possible influential point in input attribution for users to understand and interpret easily.

Notation

We denote scalars by lowercase letters (e.g., i), column vectors by bold (e.g., \mathbf{u}), random variable by X , and an estimate of a random variable by lowercase letters with a hat (\hat{X}).

II Related Work

2.1 Layer-wise Relevance Propagation

Layer-wise Relevance Propagation

Reference [2] proposes a general solution to the problem of understanding classification decisions by pixel-wise decomposition of nonlinear classifiers using Taylor-type decomposition, and reference [8] extends the idea of this paper to be applied to deep neural networks by the divide-and-conquer paradigm, and exploits the property that the function learned by a deep network is structurally decomposed into a set of simpler sub-functions that relate quantities in adjacent layers.

Deep Taylor Decomposition

In order to decompose the prediction of deep neural network, [8] utilized Taylor series. Instead of considering the whole neural network function f like [2], they consider the mapping of a set of neurons x_i at a i layer to the relevance R_j which is the relevance of next layer at the output direction and assigned to a neuron x_j . Assuming that these two objects are functionally related by some function $R_j(x_i)$, they apply Taylor decomposition on this local function in order to redistribute relevance R_j onto lower-layer relevance R_i . Running this redistribution procedure in a backward pass leads eventually to the pixel-wise relevance R_p that forms the heatmap.

Equations below introduce the Taylor series for arbitrary smooth function and real number.

$$f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!} (x-a)^n. \quad (1)$$

$$= f(a) + \frac{f'(a)}{1!} (x-a) + \frac{f''(a)}{2!} (x-a)^2 + \frac{f'''(a)}{3!} (x-a)^3. \quad (2)$$

Using error term ϵ , we can express the first-order Taylor series as

$$f(x) = f(a) + \frac{d}{dx} f(x) \Big|_{x=a} (x-a) + \epsilon. \quad (3)$$

$$R_j = \sum_k \frac{z_{jk}}{\sum_j z_{jk}} R_k. \quad (4)$$

The propagation procedure implemented by LRP is subject to a conservation property, where what has been received by a neuron must be redistributed to the lower layer in equal amount.

That is, for the d dimensional input considering multivariate function, we can write the first-order Taylor series as follows

$$f(\mathbf{x}) = f(\mathbf{a}) + \sum_{p=1}^d \frac{\partial f}{\partial x_p} \Big|_{\mathbf{x}=\mathbf{a}} (\mathbf{x}-\mathbf{a}) + \epsilon. \quad (5)$$

In equation (5), $f(\mathbf{a})$ and ϵ are constant. The second term on the right side represents the change in $f(\mathbf{x})$ in x_p . Equation (5) helps us to decompose the output into a relevance score,

but there are 2 unnecessary terms, $f(a)$ and ϵ . We can find d dimensional input \mathbf{a} that makes $f(a) = 0$ called root point mathematically as introduced in [8] which depends on input domain as follow

$$[x_i \in \mathbb{R}^{t \times s}, x_i \in \mathbb{R}_+^{t \times s}, x_i \in [l_i; h_i], l_i \leq 0 \leq h_i].$$

. Finally, output can be resolvable only by the relevance score, approximating the function from the root point \mathbf{a} that makes $f(a) = 0$, and using the properties of the ReLu activation function to make ϵ as 0.

$$f(\mathbf{x}) = f(\mathbf{a}) + \sum_{i=1}^d \frac{\partial f}{\partial \mathbf{x}_i} \Big|_{\mathbf{a}_i = \mathbf{a}_i} (\mathbf{x}_i - \mathbf{a}_i) + \epsilon \quad (6)$$

$$= \sum_{i=1}^d \frac{\partial f}{\partial \mathbf{x}_i} \Big|_{\mathbf{x}_i = \mathbf{a}_i} (\mathbf{x}_i - \mathbf{a}_i) \quad (7)$$

$$= \sum_{i=1}^d R_i \quad (8)$$

There are 2 things we have to consider when we use this method. As Deep Taylor Decomposition has 3 different methods to find root point depends on input domain, we should consider the input domain, and check if all relevance satisfy certain properties defined by [8] at the same time when we utilize this method.

Definition 1 A heat-mapping $R(x)$ is conservative if the sum of assigned relevance in the pixel space corresponds to the total relevance detected by the model: $\forall x : f(x) = \sum_p R_p(x)$

Definition 2 A heat-mapping $R(x)$ is positive if all values forming the heat-map are greater or equal to zero, that is : $\forall x, p : R_p(x) \geq 0$

Definition 3 A heat-mapping $R(x)$ is consistent if it is conservative and positive. That is, it is consistent if it complies with [Definitions 1 and 2](#).

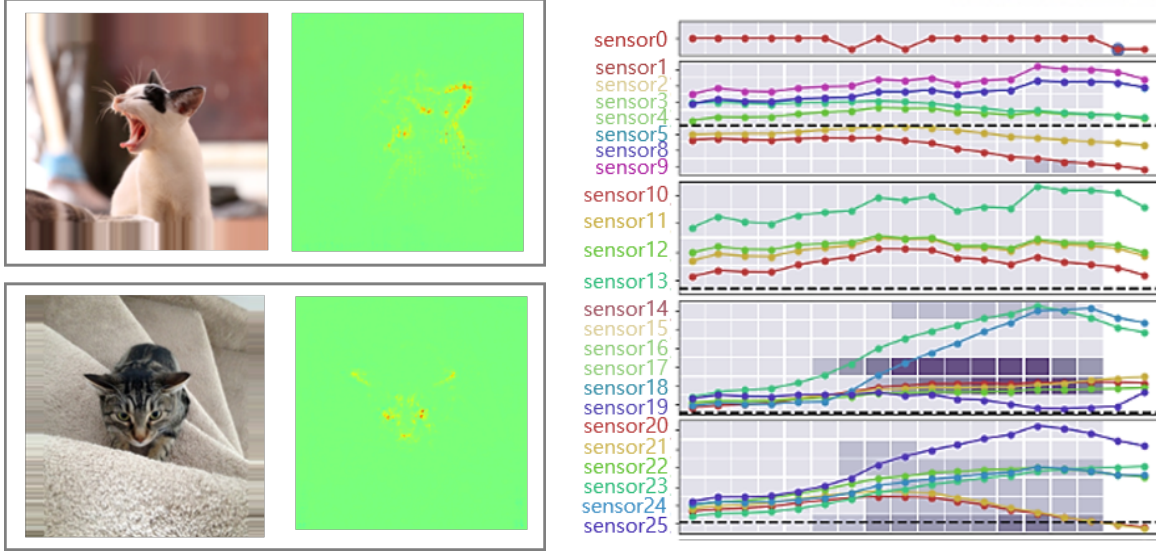
2.2 Pattern net

[9] proposed a generalization that yields two explanation techniques, PatternNet and PatternAttribution. These techniques are theoretically sound for linear models and produce improved explanations for deep networks.

A linear model can be represented as below

$$\mathbf{x} = \mathbf{s} + \mathbf{d}$$

$$\mathbf{x} = \mathbf{a}_s y + \mathbf{a}_d \epsilon$$



(a) 2 samples of LRP results with image data. The redder the heatmap, the greater the relevance. Each pixel in the heatmap represents a relevance of the pixels in the input image.

(b) Part of LRP results with multi-variate temporal data. The darker the heatmap, the greater the relevance, and each pixel in the heatmap represents the relevance of the same line feature at each input time. The lines of various colors indicate features of the label data of the corresponding color.

Figure 1: Difficulty of interpreting and understanding of multivariate time series data. All input pixels will have as much relevance as they affect the output through LRP. Figure 1a shows the result of LRP through MVLS GoogLeNet network with image as input, and figure 5b shows the part of result of LRP with sensors time sequence input. As mentioned in section 1, it is able to intuitively interpret the input attribution results for images, but it is not easy to interpret for multivariate temporal data when the input time is long or the number of features is large.

where \mathbf{x} is total data, \mathbf{s} is the signal in data, distractor \mathbf{d} is the component of the data that does not contain information about the desired output, and \mathbf{a}_s and \mathbf{a}_d are directions of spread information. Assuming the filter \mathbf{w} has been trained sufficiently well to extract y , we have

$$\mathbf{w}^T \mathbf{x} = y, \mathbf{w}^T \mathbf{s} = y, \mathbf{w}^T \mathbf{d} = 0$$

In addition, they introduce the following quality measure ρ for a signal estimator $S(\mathbf{x}) = \hat{\mathbf{s}}$, and suggest to learn the signal estimators S from data by optimizing this criterion with below equation (9) with additional constraints by measuring how much information about y can be reconstructed from the residual $\mathbf{x} - \hat{\mathbf{s}}$ using linear projection

$$\rho(S) = 1 - \max_v \text{cov}(\mathbf{w}^t \mathbf{x}, \mathbf{v}^T (\mathbf{x} - S(\mathbf{x}))) = 1 - \max_v \frac{\mathbf{v}^T \text{cov} [\hat{\mathbf{d}}, y]}{\sqrt{\sigma_{\mathbf{v}^T \hat{\mathbf{d}}}^2 \sigma_y^2}}. \quad (9)$$

The best signal estimators remove most of the information in the residuals and thus yield large $\rho(S)$. In addition, they present two possible solutions to this problem, the linear estimator as-

suming a linear dependency between \mathbf{s} and y , yielding a signal estimator, and the two-component estimator to move beyond the linear signal estimator, considering the gate of the ReLU closes for negative activations. Based on the presented analysis, they propose PatternNet and PatternAttribution. PatternNet yields a layer-wise back-projection of the estimated signal to input space. The signal estimator is approximated as a superposition of neuron-wise, nonlinear signal estimators in each layer. It is equal to the computation of the gradient where during the backward pass the weights of the network are replaced by the informative directions. PatternAttribution can be seen as a root point estimator for the Deep Taylor Decomposition. Here, the explanation consists of neuron-wise contributions of the estimated signal to the classification score. By ignoring the distractor, PatternAttribution can reduce the noise and produces much clearer heat maps. By working out the back-projection steps in the Deep-Taylor Decomposition with the proposed root point selection method, it becomes obvious that PatternAttribution is also analogous to the backpropagation operation.

Other approaches take weight vector \mathbf{w} as importance measure which highly depends on the distractor and this approach detect \mathbf{a}_s to be learned from data. It is important to recognize at this point that selecting a root point for the DTD corresponds to estimating the distractor $\mathbf{x}_0 = \mathbf{d}$ and, by that, the signal $\hat{\mathbf{s}} = \mathbf{x} - \mathbf{x}_0$. Pattern Attribution is a DTD extension that learns from data how to set the root point.

This method is an attempt to increase the reliability of the input attribution itself by setting the root point learned from data. In contrast, our direction is to inform the user of the existence of uncertainty by analyzing the uncertainty of the input attribution.

2.3 Bayesian Neural Networks and Monte-Carlo Dropout

Bayesian Neural Network

Bayesian Neural Networks (BNNs) introduce uncertainty to deep learning models from a Bayesian perspective. By giving a prior to the network parameters W , the network aims to find the posterior distribution of W , instead of a point estimation. Unfortunately, due to the complicated non-linearity and non-conjugacy in deep models, exact posterior inference is rarely available. In addition, most traditional algorithms for approximate Bayesian inference cannot scale to the large number of parameters in most neural networks. In order to perform inference about the distribution of the parameters of the deep learning model with the Bayesian approach, the necessary content is variational Inference. Because deep learning models have very high-level parameters, obtaining a posterior distribution is intractable. Therefore, we have to use an approximation method and the most commonly used method is variational inference. However, recent studies have demonstrated that this approach approximates the use of Monte-Carlo Dropout in network [6]

Monte-Carlo Dropout

Reference [6] show that the use of dropout in neural networks can be interpreted as a Bayesian approximation of a Gaussian process, a well known probabilistic model. Dropout is used in many models in deep learning as a way to avoid over-fitting [10], and [6] show that dropout approximately integrates over the models weights. This approach, called **Monte Carlo dropout**, will mitigate the problem of representing model uncertainty in deep learning without sacrificing either computational complexity or test accuracy and can be used for all kind of models trained with dropout.

Modelling uncertainty with Monte Carlo dropout works by running multiple forward passes through the model with a different dropout masks with probability p every time. Consider a trained neural network with dropout f_{nn} . To derive the uncertainty for one sample \mathbf{x} , we collect the predictions of B inferences with different dropout masks with probability p . Here $f_{nn}^{d_i}$ represents the model with dropout mask d_i . So we obtain a sample of the possible model outputs for sample \mathbf{x} as

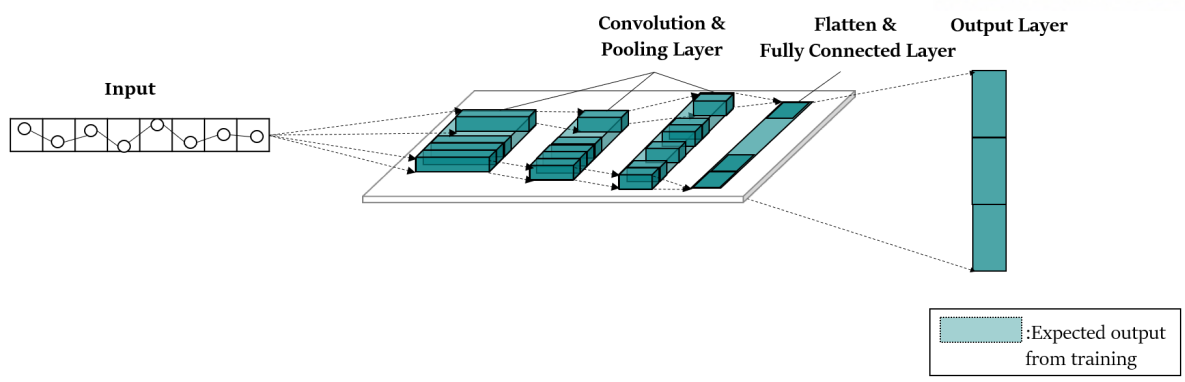
$$\left\{ f_{nn}^{d_0}(\mathbf{x}), \dots, f_{nn}^{d_B}(\mathbf{x}) \right\}.$$

By computing the average and the variance of this sample, we get an ensemble prediction, which is the mean of the models posterior distribution for this sample and an estimate of the uncertainty of the model regarding x .

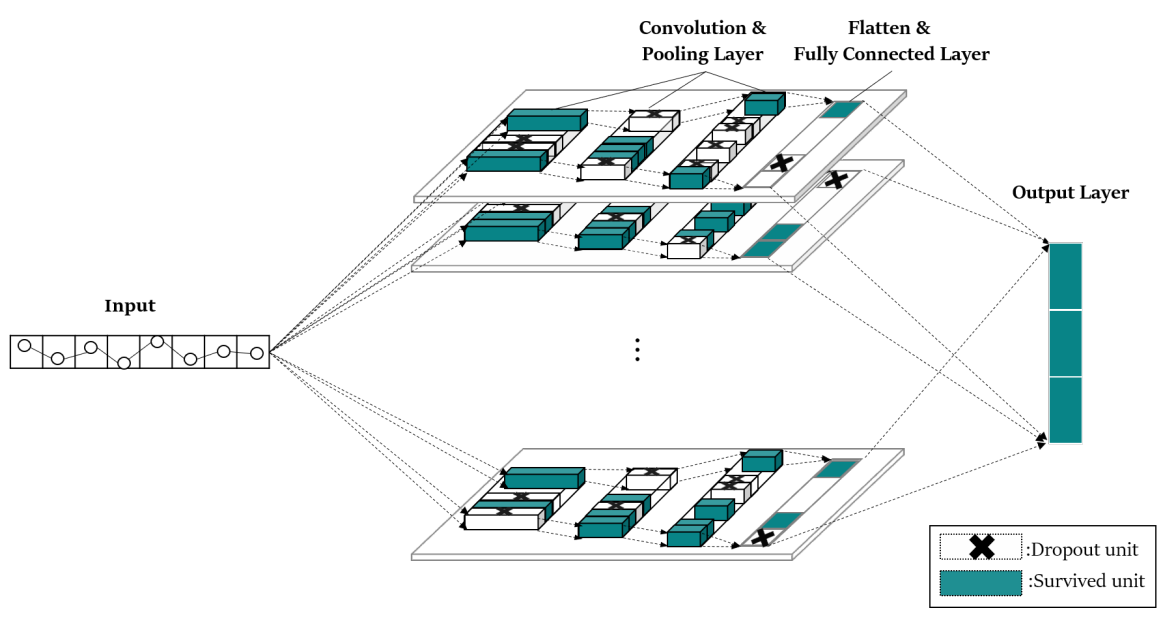
$$\text{predictive posterior mean : } p = \frac{1}{B} \sum_{i=0}^B f_{nn}^{d_i}(x).$$

$$\text{uncertainty : } c = \frac{1}{B} \sum_{i=0}^B \left[f_{nn}^{d_i}(x) - p \right]^2.$$

This method can be easily implemented using existing deep learning tools which means no change of the existing model architecture and provides uncertainty estimation almost for free. Specifically, stochastic dropouts are applied after each hidden layer, and the model output can be approximately viewed as a random sample generated from the posterior predictive distribution. As a result, the model uncertainty can be estimated by the sample variance of the model predictions in a few repetitions. We will use this method to get random sample generated from the posterior predictive distribution of input attribution with LRP using DTD.



(a) Standard dropout in prediction step



(b) MC dropout in prediction step

Figure 2: Difference between standard dropout and MC dropout. Standard dropout in prediction or test time has single output from expected output from training. However, Monte-Carlo dropout in prediction time has multiple outputs from several model variations and averages stochastic forward passes through the model.

III Methods

3.1 Input Attribution with Monte-Carlo dropout Network

In this section, we will do Monte Carlo dropout backward passes when calculating input attributions using LRP. We will be able to approximate the distribution of input attribution by running multiple backward passes through the model with a different dropout masks with probability p every time and obtaining input attribution through several modified model structures. In addition, we will see multiple input attribution from dropout network whether satisfy consistency defined in *Definition3* referred in section 3.2 when modelling uncertainty of input

attribution,

The algorithm for the input attribution with MC dropout network is described in Algorithm 1. In this algorithm, we apply MC dropout as a backward pass to different dropout models with probability p for getting several input attributes through LRP using DTD. We will introduce our methods by explaining each line of the algorithm in below.

Algorithm 1: Input Attribution with Monte-Carlo dropout Network

Input : data \mathbf{X} , prediction network $f_{nn}(\cdot)$, dropout probability p , number of iterations

B

Output: $MCDropoutRelevance$

```

1 for  $i \leftarrow 1$  to  $N$  do
2    $y_i \leftarrow f_{nn}(X_i)$ 
3   for  $b \leftarrow 1$  to  $B$  do
4      $f_{nn}^b \leftarrow \text{MCDropout}(f_{nn}(\cdot), p)$  // Apply MC dropout in trained network with
        probability  $p$  for model perturbation
5     input attribution  $\mathbf{r}_i^b \leftarrow \text{DeepTaylorDecomposition}(f_{nn}^b, y_i)$  // Calculate input
        attribution with dropout network and output using Deep Taylor
        Decomposition
6     if not ( CheckConservative ( $\mathbf{r}_i^b$ ) and CheckPositive ( $\mathbf{r}_i^b$ ) ) then
7       /* Check if consistency of relevance is maintained after dropout */
8       break
9     end if
10  end for
11 end for
12 return  $MCDropoutRelevance : \{ [\mathbf{r}_1^1, \dots, \mathbf{r}_1^B], [\mathbf{r}_2^1, \dots, \mathbf{r}_2^B], \dots, [\mathbf{r}_N^1, \dots, \mathbf{r}_N^B] \}$ 

```

For every input data \mathbf{X}_i with shape [length of time $t \times$ Number of features f] in $\mathbf{X}_{\{1, \dots, N\}}$, we can get corresponding output from trained prediction network $f_{nn}(\cdot)$ (line 1-2, Algorithm1). And based on the given network and output, we can get input attribution with several XAI methods. Here, we are using LRP with DTD method to get the input attribution, but we don't simply get a single input attribute over a given network $f_{nn}(\cdot)$. Instead, we will use MC dropout with backward pass to make B times of random sampling of each hidden layers with probability p . Then, we can obtain input attribution \mathbf{r}_i^b from each of the models f_{nn}^b that gave this change for each input data \mathbf{X}_i (line 3-5, Algorithm1). Figure 3 shows the multiple input attributions of 30 input sensors in real industrial data from models with randomly dropped node with probability p . Through the B input attribution samples, we can approximate the distribution of the input attribution. Importantly, as we utilize LRP in this step, we have to make sure if these relevance output has still maintained consistency. (line 6-9, Algorithm1). After we do all these procedure, we will be able to get input relevance with B number of MC dropout network for each data \mathbf{X}_i ,

Algorithm 2: Dropout

Input : prediction network $f_{nn}(\cdot)$, dropout probability p
Output: dropout network f_{nn}^*

```

1 /* Apply dropout to every hidden layers of network */
2 for layer in  $f_{nn}$  do
3   if isHiddenLayer (layer) then
4      $f_{nn}[layer] \leftarrow \text{SelectNode}(\text{layer}, p)$  // Apply dropout for hidden layer.
5   end if
6    $f_{nn}^* \leftarrow f_{nn}$ 
7 end for
8 return  $f_{nn}^*$ 

```

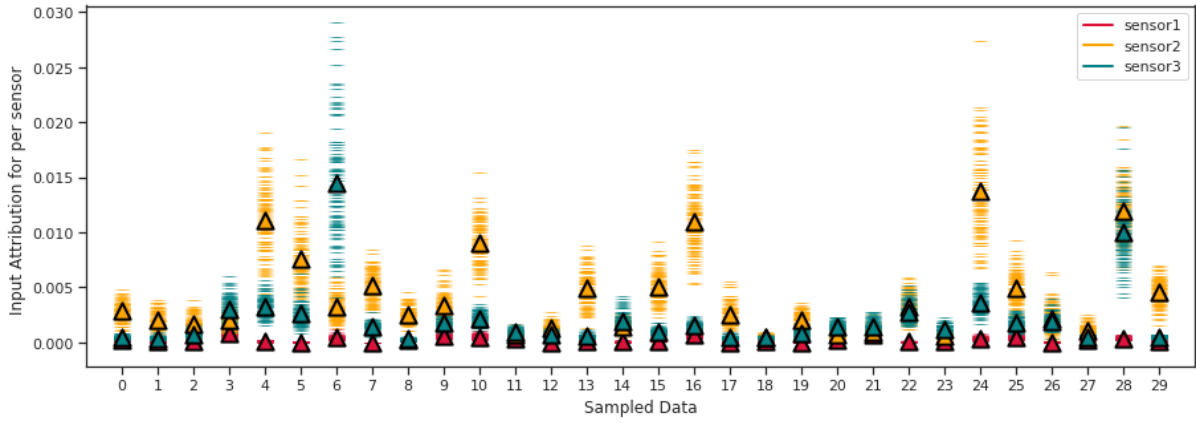


Figure 3: 50 different input attributions for 3 sensors with 30 data through LRP with backward Monte-Carlo Dropout. Different input attributions are derived from randomly dropped out model structures.

shaped $[B \times N \times t \times f]$, (line 12, Algorithm1).

3.2 Input Attribution Analysis with Uncertainty

In this section, we define **Monte-Carlo Layer-wise Relevance Decomposition, MC dropout LRP**, as LRP input attribution results from MC dropout network in section 3.1. We will extract possible influential features to model's output with mean and standard deviation value of MC dropout LRP as criterion.

$$\text{predictive posterior mean for } \mathbf{r}_i^{1,\dots,B} : p_i = \frac{1}{B} \sum_{j=0}^B \mathbf{r}_i^j$$

$$\text{uncertainty for } \mathbf{r}_i^{1,\dots,B} : c_i = \sqrt{\frac{1}{B} \sum_{j=0}^B [\mathbf{r}_i^j - p_i]^2}$$

So far, we have *MCDropoutRelevance*, which means multiple input attributions with LRP from B numbers of random sampled trained network using MC dropout backward pass from section 3.1. Now we will calculate the mean and uncertainty value for B relevance scores and will use them as the criterion to distinguish inputs that possibly affect the output in Algorithm 3.

Algorithm 3: Input Attribution Analysis with Uncertainty

Input : *MCDropoutRelevance*, data \mathbf{X}

Output : Possible Influential Data

```

1  $\mathbf{p} = \text{PredictivePosteriorMean}(\textit{MCDropoutRelevance})$ 
2  $\mathbf{c} = \text{Uncertainty}(\textit{MCDropoutRelevance})$ 
3  $\mathbf{Q1}_p, \mathbf{Q1}_c = \text{getQuantile}(\mathbf{p}, 1), \text{getQuantile}(\mathbf{c}, 1)$ 
4  $\mathbf{Q3}_p, \mathbf{Q3}_c = \text{getQuantile}(\mathbf{p}, 3), \text{getQuantile}(\mathbf{c}, 3)$ 
5  $\mathbf{IQR}_p, \mathbf{IQR}_c = \mathbf{Q3}_p - \mathbf{Q1}_p, \mathbf{Q3}_c - \mathbf{Q1}_c$ 
6  $\text{fence}_p, \text{fence}_c = \mathbf{Q3}_p + 1.5 \times \mathbf{IQR}_p, \mathbf{Q3}_c + 1.5 \times \mathbf{IQR}_c$ 
7  $\text{threshold}_p, \text{threshold}_c = \text{fence}_p + \varepsilon_p, \text{fence}_c + \varepsilon_c$ 
8  $\text{candidate1} \leftarrow \{\mathbf{X}_i \in \mathbf{X} \mid \mathbf{p}_i > \text{threshold}_{p_i} > \text{ and } \mathbf{c}_i > \text{threshold}_{c_i}\}$ 
9  $\text{candidate2} \leftarrow \{\mathbf{X}_i \in \mathbf{X} \mid \mathbf{p}_i > \text{threshold}_{p_i} > \text{ and } \mathbf{c}_i \leq \text{threshold}_{c_i}\}$ 
10  $\text{candidate3} \leftarrow \{\mathbf{X}_i \in \mathbf{X} \mid \mathbf{p}_i \leq \text{threshold}_{p_i} > \text{ and } \mathbf{c}_i > \text{threshold}_{c_i}\}$ .
11 return candidate1, candidate2, candidate3

```

In case of relevance, there are few inputs that affect the results to some degree, especially when data has many features and long input time as the output of model between 0 to 1 should be decomposed to all inputs. Therefore, we can think of high value of attribution as outlier which is an observation that lies an abnormal distance from other values in a random sample from a population. We use interquartile range(IQR) of mean and uncertainty value which is defined to be the spread of the middle of data values, and upper fence as $Q3 + 3 * IQR$ and call points beyond the outer fences as *High*, and within the outer fences as *Low*.(line 1-5, Algorithm3) and pick out data candidates of influential input (line 8-10, Algorithm3) as follow.

- Data with High uncertainty condition $\{x_i \in \mathbf{x} \mid c_i > \text{threshold}_{c_i}\}$
 - with condition of High mean $\{x_i \in \mathbf{x} \mid p_i > \text{threshold}_{p_i}\}$
 - with condition of with Low mean $\{x_i \in \mathbf{x} \mid p_i \leq \text{threshold}_{p_i}\}$
- Data with Low uncertainty condition $\{x_i \in \mathbf{x} \mid c_i \leq \text{threshold}_{c_i}\}$
 - with condition of High mean $\{x_i \in \mathbf{x} \mid p_i > \text{threshold}_{p_i}\}$

As data in attribution of [High Mean, High Uncertainty](upper-right) case have high attribution, but it also have a high uncertainty, we have to look carefully with some doubt when

analyzing this result with possibility that it does not have much effect to the output of model. In case of [High Mean, Low Uncertainty](lower-right), we can trust this result to some extent with little uncertainty. And lastly, we have to focus on the data with input attribution of [Low Mean, High Uncertainty](upper-left) since certainty is less for the less influence, the possibility of being sufficiently large should also be considered. Figure 4 shows the result after dividing the cases for input attribution with proposed method. Before considering the uncertainty, it can be said that only input data with high attribution is influential on the output, but considering uncertainty, data with low attribution but high uncertainty has the potential to have high attribution also and need to be considered.

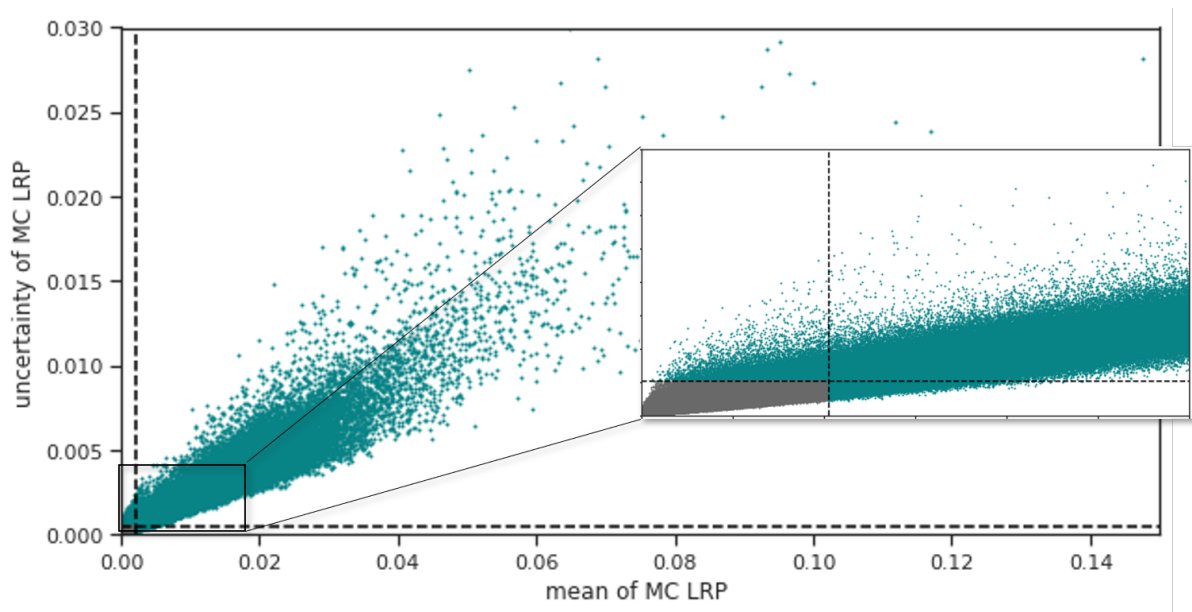
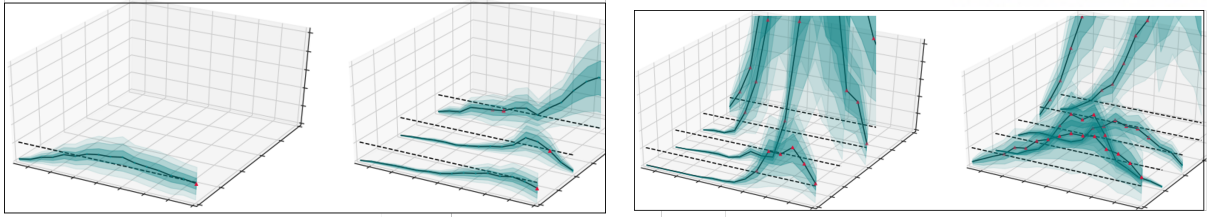


Figure 4: Scatter plot of the mean and uncertainty of the input attribution. X axis represents mean, and Y axis represents uncertainty(standard deviation) of input attribution after MC dropout. The blue dots are the ones that have the possibility of affecting the output of the model, given uncertainty and mean of input attribution



(a) Sample of input attribution that has low values but has high uncertainty at the same time.

(b) Sample of input attribution that has high values but has high uncertainty at the same time.

Figure 5: Examples of input attributions with several sub-divided cases. X-axis represent time, Y-axis represent sensors, and Z-axis represent input attribution. The data points in the case are marked with a red dot and the uncertainty of the input attribution is represented with green shadow.

IV Experiments

4.1 Experimental Setup

In this section, we will compare the results of proposed algorithm with those of the existing algorithm. For comparison, we use several evaluation method for XAI on time series domain suggested by [11]. They suggested four methods below with assumption. The assumption follows the time series $t = (t_0, t_1, t_2, \dots, t_m)$ and the relevance generated by the XAI method, LRP in this paper, as $r = (r_0, r_1, r_2, \dots, r_m)$ to get a worse result of the quality metric for the classifier if combined. Under this assumption, a time point t_i gets changed if r_i is in the set of influential input candidates. The time point t_i is set to zero or the inverse $(1 - t_i)$ (data is normalized) and leads to the new time series samples, called *zero perturbation* and *inverse perturbation*. Another method is that swap time points in consecutive time points in candidates of influential input, called *swap time points*. The last method they propose is *mean time points*. Instead of swapping the time points, the mean of the sub-sequence is taken to exchange the whole sub-sequence.

Based on these evaluation method, we did experiments using steel industrial data and a public multi variate time series data from UCI dataset [12]. In case of industrial data, we cannot reveal details about the data, so it is not specified numerically because of security concerns. However, we maintained the overall form of the result. In addition, as the proposed algorithm is applicable not only to time series data but also to various domains, we will simply show the difference between existing algorithm and proposed method by visualization.

The model we used in this experiment section is composed of three convolution layers, and two fully connected layers. ReLu function is used for the activation function in every hidden layer. In the training phase, dropout of 0.5 was applied after the activation function of every fully connected layer to prevent over-fitting. Batch size and epoch was set to 128 and 500 respectively. In the prediction phase, we use the model with the lowest validation loss and MC

Description/Data	Steel Industrial Data	Human Activity Recognition(HAR)
Number of instances	771,547	10,299
Input time	A	128
Number of Attributes	B	561
Number of Class	3	6

Table 1: Data Description for the experiments.

dropout was applied at a rate of 0.5 before the non-linear activation function of all hidden layers for Bayesian approximate. We have done 100 times of Monte-Carlo dropout, and then we have applied LRP method using DTD to get input attribution for each random sampled network.

4.2 Experimental Result

Steel Industrial Data

In the first experiment, real industry data from steel company were used and due to security issues, the values were randomly changed while maintaining the overall form of the result. We trained the classification model to classify whether to lower, maintain or raise the airflow with 300 epochs in this experiment.

The statistics of input attribution are well shown in figure 6. Dotted line represent threshold. Most of input data has low uncertainty and low input attribution at the same time as only a few points of input data directly affect the result.

Bayesian Input Attribution	Low Mean ($mean \leq 0.002055$)	High Mean ($mean > 0.002055$)
High uncertainty ($std > 0.000524$)	33,638	231,822
Low uncertainty ($std \leq 0.000524$)	2,295,859	29,817

(a) Table from steel industrial data representing numbers of each sub-divided cases with input attribution and its uncertainty.

Discrete Input Attribution	Low Mean ($mean \leq 0.00151$)	High Mean ($mean > 0.00151$)
	2,311,319	279,817

(b) Table from steel industrial data representing numbers of each sub-divided cases with input attribution.

Table 2: Result difference from steel industrial data between considering input attribution only and with uncertainty also.

Table 2a shows the number of subdivided cases with the criterion of mean and uncertainty of

multiple input attributions from MC dropout. The shaded area in blue is the number of input data that have the possibility of influencing the output of the model, given the uncertainty. We would simply think that input with a high attribution were candidates of influential data for the output which is $(2,301,822 + 29,817)$, but after considering uncertainty, we could found 33,638 more candidates with our proposed method. On the contrary, table 2b shows the number of subdivided cases with the single discrete input attribution. The shaded area in blue is the number of input data that have the possibility of influencing the output of the model without the uncertainty.

	Zero	Inverse	Swap	Mean
LRP with uncertainty	0.342	0.443	0.363	0.360
LRP	0.302	0.435	0.361	0.362
<i>Random</i>	0.039	0.203	0.108	0.094

Table 3: Result table from steel industrial data with the averaged changed accuracy from the different candidates of influential input over 1000 random sampled for each class.

We randomly sampled 1000 data in each class and evaluated the existing and proposed methods using mentioned methods. Table 3 shows averaged changed accuracy from the different candidates of influential input over 1000 random sampled for each class. In case of perturbation with zero, inverse and swap for influential candidates, our proposed method have most different result.

Human Activity Recognition

In the second experiment, dataset from UCI is used. Each person performed six activities (WALKING, WALKING UPSTAIRS, WALKING DOWNSTAIRS, SITTING, STANDING, LAYING) wearing a smartphone on the waist. Data is consist of 3-axial linear acceleration and 3-axial angular velocity at a constant rate of 50Hz.

The statistics of input attribution are well shown in figure 7. Dotted line represent threshold. Most of input data has low uncertainty and low input attribution at the same time as only a few points of input data directly affect the result.

Table 4a shows the number of subdivided cases with the criterion of mean and uncertainty of multiple input attributions from MC dropout. The shaded area in blue is the number of input data that have the possibility of influencing the output of the model, given the uncertainty. We would simply think that input with a high attribution were candidates of influential data for the output which is $(80,897 + 39,061)$, but after considering uncertainty, we could found 57,815 more candidates with our proposed method. On the contrary, table 4 shows the number of subdivided cases with the single discrete input attribution. The shaded area in blue is the number of input data that have the possibility of influencing the output of the model without the uncertainty.

Bayesian Input Attribution	Low Mean ($mean \leq 0.002455$)	High Mean ($mean > 0.002455$)
High uncertainty ($std > 0.000611$)	57,815	80,897
Low uncertainty ($std \leq 0.000611$)	1,519,123	39,061

(a) Table from Human Activity Recognition(HAR) representing numbers of each sub-divided cases with input attribution and its uncertainty.

Discrete Input Attribution	Low Mean ($mean \leq 0.002278$)	High Mean ($mean > 0.002278$)
	1,553,243	143,653

(b) Table representing numbers of each sub-divided cases with input attribution.

Table 4: Result difference from Human Activity Recognition(HAR) between considering only input attribution itself and with uncertainty also.

	Zero	Inverse	Swap	Mean
LRP with uncertainty	0.223	0.338	0.299	0.294
LRP	0.219	0.338	0.264	0.264
Random	0.048	0.122	0.098	0.085

Table 5: Result table from Human Activity Recognition(HAR) with the averaged changed accuracy from the different candidates of influential input over 1000 random sampled for each class.

We evaluated the existing and proposed methods as same as previous experiments. Table 5 shows changed accuracy from the different candidates of influential input over whole 1474 test time sequence points. In case of perturbation with zero, inverse and mean for influential candidates, our proposed method have most different result, and in case of swapping, existing method has same difference with our proposed one.

V Conclusion

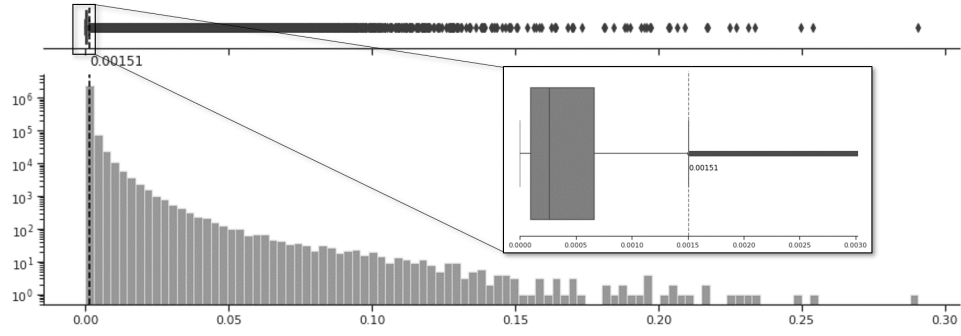
Other approaches focus on reducing uncertainty of input attribution, but we dealt with this uncertainty in a slightly different direction. We show the existence of uncertainty in input attribution from LRP with MC dropout, and suggest all possible influential input as considering both the uncertainty and input attribution itself at the same time. Lastly, we present visualization to make people understand and able to interpret better, especially for multi-variate time series data which is usually complicated to do so.

Here, we summarize the main results. In thesis, we

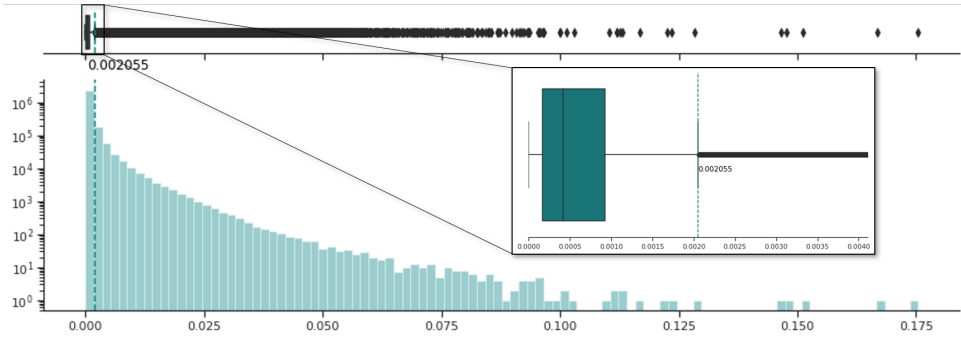
1. Provide a Bayesian approximate for input attribution through MC dropout.
2. Not only find input data point which has high input attribution, but also find all possible influential point which has the potential to affect the output of the model.
3. Proved proposed method is improved over the existing method with several evaluation methods for XAI on time series domain.

We address here some limitations in our current work and some potential future directions that we can extend our ideas.

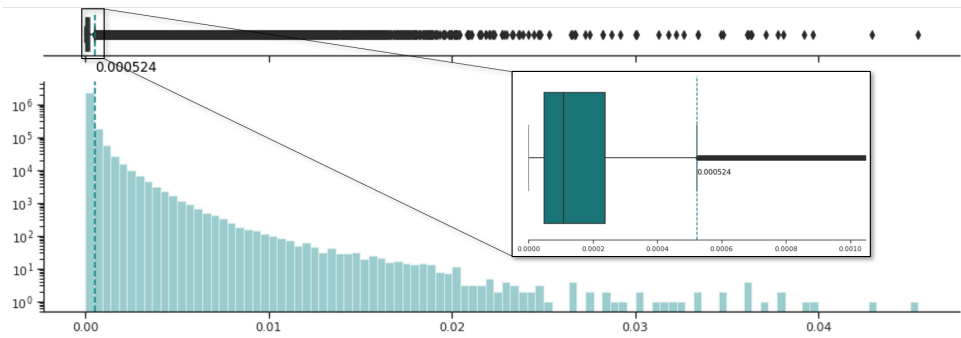
1. It depends heavily on how you set the criteria for dividing the input attribution on a case-by-case basis.
2. The procedure of MC dropout costs a lot in terms of space and time.



(a) Box plot and histogram of mean of LRP

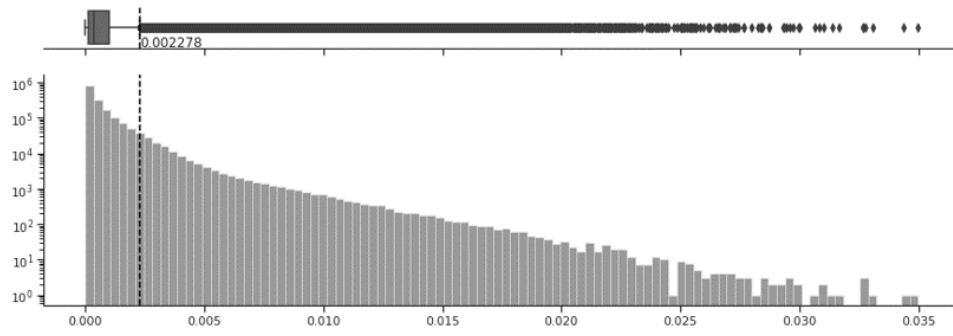


(b) Box plot and histogram of mean of MC dropout LRP

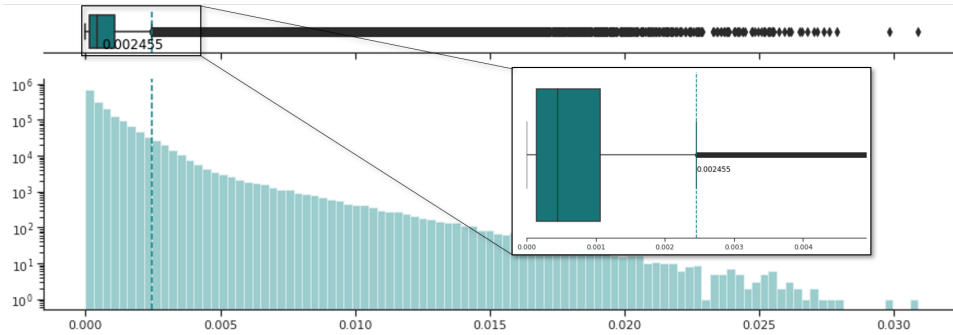


(c) Box plot and histogram of uncertainty of MC dropout LRP

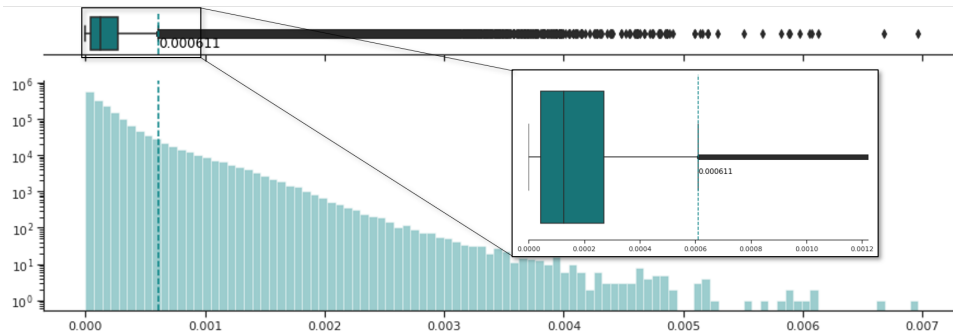
Figure 6: Box plot and histogram of single LRP values from existing method and box plot and histogram of mean and uncertainty values from multiple LRPs with MC dropout



(a) Box plot and histogram of mean of LRP



(b) Box plot and histogram of mean of MC dropout LRP



(c) Box plot and histogram of uncertainty of MC dropout LRP

Figure 7: Box plot and histogram of single LRP values from existing method and box plot and histogram of mean and uncertainty values from multiple LRPs with MC dropout

References

- [1] W. Samek, G. Montavon, A. Vedaldi, L. K. Hansen, and K.-R. Muller, *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. <https://doi.org/10.1007/978-3-030-28954-6>: Springer, Cham, 2019.
- [2] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Muller, and W. Samek, “On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation,” *PLOS ONE*, vol. 10, pp. 1–46, 07 2015. [Online]. Available: <https://doi.org/10.1371/journal.pone.0130140>
- [3] M. Bohle, F. Eitel, M. Weygandt, and K. Ritter, “Layer-wise relevance propagation for explaining deep neural network decisions in mri-based alzheimer’s disease classification,” *Frontiers in Aging Neuroscience*, vol. 11, p. 194, 2019. [Online]. Available: <https://www.frontiersin.org/article/10.3389/fnagi.2019.00194>
- [4] M. Krzywinski and N. Altman, “Points of significance: Importance of being uncertain,” *Nature Methods*, vol. 10, pp. 809–810, 2013.
- [5] N. Radzuan, Z. Othman, and A. Abu Bakar, “Uncertain time series in weather prediction,” *Procedia Technology*, vol. 11, pp. 557–564, 12 2013.
- [6] Y. Gal and Z. Ghahramani, “Dropout as a bayesian approximation: Representing model uncertainty in deep learning,” 2015.
- [7] B. Letham, C. Rudin, T. H. McCormick, and D. Madigan, “Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model,” *The Annals of Applied Statistics*, vol. 9, no. 3, p. 1350–1371, Sep 2015. [Online]. Available: <http://dx.doi.org/10.1214/15-AOAS848>
- [8] G. Montavon, S. Lapuschkin, A. Binder, W. Samek, and K.-R. Muller, “Explaining nonlinear classification decisions with deep taylor decomposition,” *Pattern Recognition*, vol. 65, pp. 211 – 222, 2017. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0031320316303582>
- [9] H. Li, J. G. Ellis, L. Zhang, and S.-F. Chang, “PatternNet: Visual Pattern Mining with Deep Neural Network,” *arXiv e-prints*, p. arXiv:1703.06339, Mar 2017.

- [10] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014. [Online]. Available: <http://jmlr.org/papers/v15/srivastava14a.html>
- [11] U. Schlegel, H. Arnout, M. El-Assady, D. Oelke, and D. A. Keim, “Towards a rigorous evaluation of xai methods on time series,” 2019.
- [12] D. Dua and C. Graff, “UCI machine learning repository,” 2017. [Online]. Available: <http://archive.ics.uci.edu/ml>

Acknowledgements

I would like to express my gratitude to my supervisor Jaesik Choi for the useful comments, remarks and engagement through the learning process of this master thesis.

Furthermore I would like to thank members of Statistical Artificial Intelligence Lab who have willingly shared their precious time during the process of discussing.

I would also like to express my gratitude to my friends who always make me smile and comforted me during difficult times.

Lastly, I would like to express my deepest appreciation to my family who have supported me throughout entire process both by keeping me harmonious and helping me putting pieces together. I will be grateful forever for your love.

