Quality-Discriminative Localization of Multisensor Signals for Root Cause Analysis

Yoon Sang Cho and Seoung Bum Kim

***Abstract*—Root cause analysis (RCA) methods for effectively identifying the critical causes of abnormal processes have attracted attention because manufacturing processes have grown in scale and complexity. However, the existing methods for building automatic RCA models suffer from the disadvantage of typically requiring expert knowledge. In addition, without a dataset representing the causal relationship of multivariate processes, it is difficult to provide useful information for RCA. Although data-driven RCA methods have been proposed, most are based on classification models. Given that product quality is defined as a continuous variable in many manufacturing industries, classification models are limited in deriving root causes affecting the product quality level. In this article, we propose a regression model-based RCA method, which we call quality-discriminative localization, consisting of a convolutional neural network (CNN)-based activation mapping of multisensor signal data. In our proposed method, the CNN predicts the product quality of a continuous variable. Activation mapping then extracts causal maps that highlight significant sensor signals for each product. To identify the root causes, we generate a root cause map from the weighted sum of quality and causal maps. We consider root causes as locations of abnormal processes and processing times from localized activation scores on the root cause map. We experimentally demonstrate the usefulness of the proposed method with simulated data and real process data from a steel manufacturing process. Our results show that the proposed method successfully identifies root causes with distinct sensor signal patterns.**

***Index Terms*—Activation mapping, convolutional neural network (CNN), multisensor signal data, quality-discriminative localization, root cause analysis (RCA), steel manufacturing.**

I. Introduction

ROOT cause analysis (RCA) plays a key role in maintaining stable manufacturing processes. An RCA aims to determine the critical causes of abnormal processes and product defects [1]. Recently, RCA methods have drawn attention because modern manufacturing processes have become more automated and interlinked [2]. Once a process has an abnormal event with an unknown root cause, it adversely affects other processes. In particular, in large-scale and complex manufacturing systems, failing to detect root causes leads to recurrent problems, which can, unchecked, engender machine breakdowns and decrease productivity [3]. Therefore, an RCA model that appropriately explains the relationship between process states and product quality is required.

This article addresses an RCA problem for understanding root causes, considering the following issues. First, abnormal signals, such as noisy symptoms, should be detected because they are directly associated with abnormal process states and decreased product quality. Second, the RCA model must identify multilevel causes because problems occur in relation to multivariate processes and processing times. Third, the RCA method must detect not only temporary causes but also the unknown root causes that intrinsically lead to abnormal processes.

With advanced sensor technology, real-time multisensor signal data can be collected in many industries. Real-world examples include human activity recognition [4], machinery health monitoring [5], and manufacturing processes [6]. Sensor signals have been used to recognize human activity; the detection of abnormal signals of human activity has been utilized in healthcare areas for providing patient information, such as fainting, falling, and headaches. In machinery health monitoring, sensor signals can represent the process state of equipment, and the detection of abnormal signals has been considered a crucial task for preventing equipment faults. In manufacturing systems, monitoring abnormal sensor signals, such as fluctuations, is necessary for process control in a normal state. Thus, the identification and explanation of abnormal signals are important issues.

In manufacturing systems, such sensor data represent sequential processes and processing time information, which determine product quality. When the datasets include the causal factor, RCA involves two steps: 1) construction of a predictive model and 2) identification of the root cause. The prediction model explains the relationship between process states and product quality, and then it detects the root cause from the sensor data with the highlighted parameters of the predictive model [1], [7].

In general, RCA models are either probabilistic or deterministic. In probabilistic model-based RCA, Bayesian networks that can achieve probabilistic reasoning have been widely

Manuscript received March 6, 2021; accepted June 28, 2021. Date of publication July 26, 2021; date of current version June 16, 2022. This work was supported in part by the Brain Korea 21 FOUR, Ministry of Science and ICT (MSIT) in Korea through the ITRC Support Program supervised by the IITP under Grant IITP-2020-0-01749; in part by the National Research Foundation of Korea grant funded by the MSIT under Grant NRF-2019R1A4A1024732; and in part by the Ministry of Culture, Sports and Tourism and Korea Creative Content Agency under Grant R2019020067. This article was recommended by Associate Editor L. Chen. *(Corresponding author: Seoung Bum Kim.)*

The authors are with the School of Industrial and Management Engineering, Korea University, Seoul 02841, Republic of Korea (e-mail: yscho187@korea.ac.kr; sbkim1@korea.ac.kr).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TSMC.2021.3096529.

Digital Object Identifier 10.1109/TSMC.2021.3096529

used. These Bayesian networks derive the posterior probabilities and can detect changes in the sensor data [3], [8]–[10]. Weidl *et al.* [3] proposed an object-oriented Bayesian network, which is a probabilistic graphical model that performs reasoning under uncertainty. Nawaz *et al.* [8] and Liu *et al.* [9] considered the cause-and-effect relationships between root causes, equipment, and process parameters using a Bayesian network. Furthermore, Wee *et al.* [10] proposed a Bayesian belief network-based causal knowledge model that provides causal strength using a fuzzy cognitive map. However, the above-mentioned RCA models are knowledge based and depend on expert knowledge [11]. Although they are useful for identifying immediate causes, requiring expert knowledge is disadvantageous in building automatic RCA methods for large-scale systems.

Data-driven RCA methods based on deterministic models have become more attractive in modern industries because they can derive root causes from observations without model uncertainty [6]. Chien *et al.* [7] used Hotelling's T^2 for sensor variable selection and analyzed the association between faulty products and sensor variables using decision trees. Mahadevan and Shah [1] proposed a one-class support vector machine (SVM) to identify abnormal processes and used SVM recursive feature elimination to determine root causes. Safizadeh and Latifi [12] proposed feature extraction using principal component analysis and data-level, feature-level, and decision-level fusion of multiple sensors. They then used *k*-nearest neighbor algorithms to perform classification in bearing fault diagnosis. Although these methods perform reasonably well within the industrial realms for which they were designed, there is still much scope for improvement. Many feature selection methods suffer from computational complexity in large-volume datasets. Furthermore, rule-based models, such as decision trees, facilitate the interpretation of results, but they cannot be readily used with raw sensor signals.

Recently, with the increasing popularity of deep learning owing to its computational and predictive performance, deep learning-based RCA models have become prominent in various fields [2], [13]–[15]. Deep neural networks directly use multiple sensor signals as the input and automatically learn the desired information from the input data. Chen and Li [16] proposed sparse autoencoder-based feature extraction for multiple accelerometer sensors and performed classification using a deep belief network for bearing fault diagnosis. Jing *et al.* [17] used deep convolutional neural networks (CNNs) for multiple fault classification by constructing two-dimensional (2-D) multisensor data.

CNNs have also been widely used for the diagnosis of defect causes in manufacturing processes. For example, Lee *et al.* [11] proposed a CNN structure, in which a receptive field tailored to multisensor signals slides along the time axis, to extract fault feature maps providing abnormal process variables and time information. Azamfar *et al.* [13] proposed the multisensor data fusion for gearbox fault diagnosis using 2-D CNN for motor current signature analysis. Yang *et al.* [18] proposed a Spearman rank correlation-based CNN architecture to extract useful features of multiple time-series signals and recognize fault features' locations. In addition, Yao *et al.* [19] attempted fault diagnosis using a CNN and temporal attention mechanism to extract meaningful temporal parts from sensor signals. Assaf and Schumann [14] also proposed explainable deep neural networks for multivariate time series. They designed a two-stage CNN architecture and used a gradient-based class activation mapping (grad-CAM) for interpreting the prediction results of average energy production.

Recently, the model-agnostic methods, including the local interpretable model agnostic explanation (LIME) [20] and Shapley additive explanation (SHAP) [21], have gained much interest. They provide local interpretability by quantifying the contribution of features for the individual prediction. The LIME perturbs the input data and observes the resulting impact of the prediction. The SHAP assigns each feature importance by exploiting the Shapely values from game theory. Schlegel *et al.* [22] used LIME and SHAP for time-series explanation and compared their performances with various machine learning models. As a similar approach, the counterfactual explanation [23] describes a causal situation by examining how the features should change to obtain a different prediction [24]. Ates *et al.* [25] used the counterfactual explanation method to explain multivariate time series and showed its better performance compared to LIME and SHAP. All the aforementioned methods were proposed for classification purposes. Although such classification model-based RCAs exhibit good performance, they cannot be applied when continuous values represent the output variable. For example, in semiconductor manufacturing processes, quality is determined by defect rates. In steel-making processes, quality can be determined by numerical values, such as the weight deviation between the target and output products.

A few RCA methods to handle regression problems have been proposed. Xia *et al.* [26] used spectral regression for fault feature extraction based on multisensor signal data. Borchert *et al.* [27] attempted to use a partial least squares regression model that can derive variable importance; they compared its performances when using raw sensor signals and using features extracted by principal component analysis. However, when using high-dimensional sensor signals, such approaches require feature selection or an extraction step, which are cumbersome for users. In addition, such approaches have difficulty deriving local explainability, indicating observation-wise causes because they focus on selecting significant variables. Granger causality [28] and transfer entropy [29] measuring statistical causality between two time-series data can be used for an RCA, but this study addresses time-series data as an input and one response variable as an output.

Schockaert *et al.* [15] attempted to derive local and global interpretability explaining the multivariate time-series data for the regression problem. They proposed a method combining CNN-based guided backpropagation and long short-term memory (LSTM)-based attention mechanism for a blast furnace process in steel manufacturing. The method predicts the hot metal temperature and derives local and global saliency maps that visualize temporal and spatial interpretability. However, the RCA was not performed in a testing process and, thus, its performance cannot be verified. They only

showed the potential of their proposed method without proper validations of RCA results. Schockaert *et al.* [30] used the variational autoencoder and LIME for local interpretability of data-driven models that forecast the hot metal temperature for the blast furnace process in steel manufacturing. However, they also missed proper validations to evaluate the RCA results.

This article proposes a CNN architecture-based RCA approach that combines a 2-D CNN model and activation mapping for deriving causal sensor signals in a regression system. Our method can visually explain which parts of the sensor signal cause abnormal quality. Thus, we call the proposed method is a quality-discriminative localization. The main contributions of this study can be summarized as follows.

1) Our study presents an RCA approach for making a CNN explainable for predicting a continuous output with multivariate time-series inputs. The RCA based on class activation mapping (CAM) has been used for classification purposes in most of the existing literature, but our study presents CAM for regression problems.
2) Our study presents the detailed and proper validation procedure in both simulation and real data to demonstrate the usefulness and applicability of the proposed method.

The remainder of this article is organized as follows. Section II introduces the discriminative localization technique using a CNN. Section III describes the details of the proposed method. Section IV presents a simulation study to examine the performance of the proposed method and compare it with other methods under different scenarios. Section V presents a case study to demonstrate the applicability of the proposed method using real data from a steel manufacturing process. Finally, Section VI presents our concluding remarks.

II. Discriminative Localization

While CNNs have been widely used in visual recognition problems [31], a number of previous works have proposed visualizing the causality of CNN predictions by highlighting pixels that play an important role in prediction [32]–[34]. The most relevant to our study is the CAM approach to class-discriminative localization. Zhou *et al.* [32] proposed a CAM method for identifying discriminative regions using image classification with restricted classes. They replaced fully connected layers with convolutional layers and a global average pooling (GAP) layer to produce class-specific feature maps. The GAP layer performs a downsampling operation, and it calculates the average values of each feature map as follows:

$$F_k = \frac{1}{(W \cdot H)} \cdot \sum_{x,y} f_k(x, y) \tag{1}$$

where $f_k(x,\ y)$ is the kth feature map, $(x,\ y)$ represents the x-axis and y-axis information of the feature map, and W and H represent the width and height of the feature maps, respectively. This yields the global average value F_k. A GAP layer is typically added between a prediction network and a feature extraction network. Instead of adding a fully connected layer for flattened features, the summarized vector F_k is fed directly into the activation function. One advantage of a GAP layer over a fully connected layer is that it makes it easier to construct an interpretable CNN by enforcing correspondences between feature maps and labels.

After training the CNN, the CAM derives an activation map by a weighted combination of the resulting feature values of the GAP and weights of a soft-max activation function. The activation map then indicates which parts of an input image were considered by the CNN for assigning labels. The activation map $M_c(x,y)$ for class c is derived as follows:

$$M_c(x, y) = \sum_{k=1}^{K} f_k(x, y) \cdot w_k^c \tag{2}$$

where $f_k(x,y)$ and $w_k{}^c$ are the kth feature map and learned weights of the last dense layer, respectively. The CAM considers $w_k{}^c$ as the importance of $f_k(x,y)$. In this study, we built a regression version of the CNN-based activation mapping that can be used to predict the quality label of a continuous variable and to highlight the critical regions of 2-D multisensor signal data.

III. Quality-Discriminative Localization

Our goal is to derive visually explainable root causes from multisensor signal data. In this section, we present the architecture of the CNN, followed by the proposed quality-discriminative localization method.

Fig. 1 presents an overview of the quality-discriminative localization using a CNN. The CNN consists of a feature extraction network and a prediction network [31]. The feature extraction network aims to construct feature maps that contain distinctive features of input data. As the CNN's essential components, the convolutional layers extract local features of the input, and then the output is typically fed into a downsampling layer, such as a max-pooling layer, to achieve translation invariance over small spatial shifts in the input data [35]. These layers form a convolution block and generate feature maps according to the number of blocks. It is noteworthy that because we aim to localize the root cause location of multisensor signals in a pixelwise manner, the final feature maps must be formed with the same size as that of the input data. Thus, we must add an upsampling operation for reduced feature maps.

In this study, to investigate the performance of the quality-discriminative localization in various feature extraction architectures, we adopted several structures that have been widely used in the image segmentation field because they simply and effectively construct final feature maps of the same size as the input. We detail the adopted structures and compare their performances in Section IV.

The prediction network comprises a 2-D GAP layer and a dense layer. The GAP layer averages the final convolutional feature map $f_k(t,s)$, and the resulting values F_k are fed into the dense layer. The dense layer has one neuron with a linear activation function that computes a linear combination of flattened feature values and learnable weight parameters [31]. The linear activation function directly passes the input value and then enables the CNN to perform a regression. Thus,

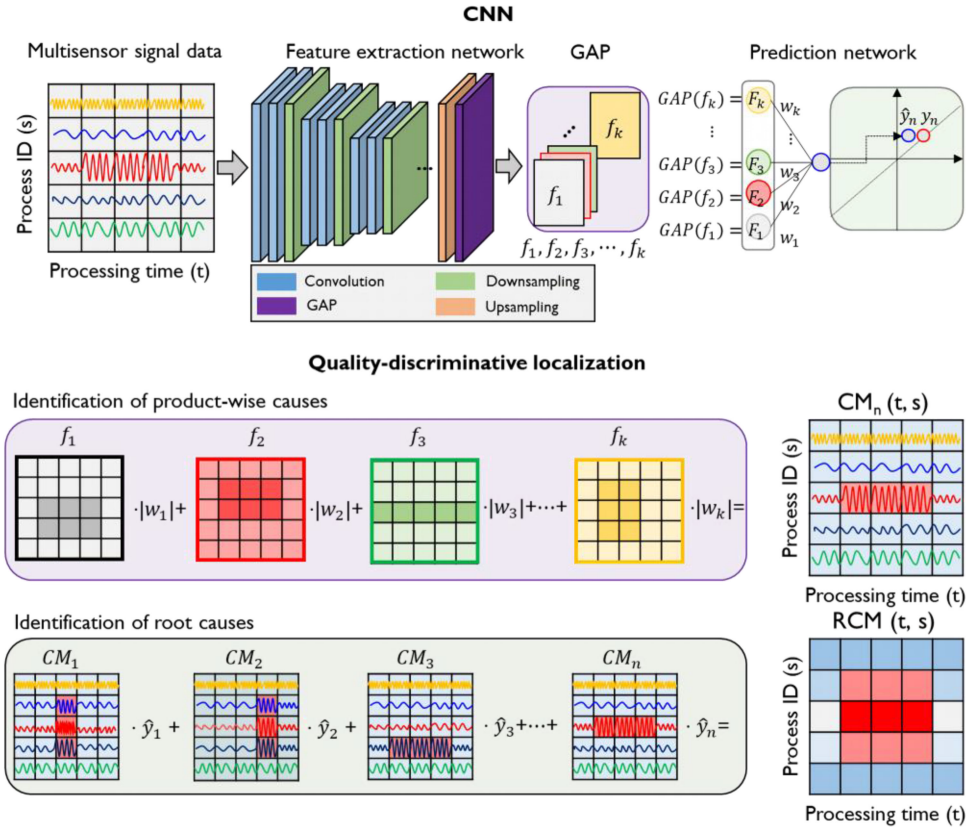

Fig. 1. Overview of the proposed CNN-based quality-discriminative localization.

the formulation of the CNN regression can be summarized as follows:

$$\mathbf{Y} = \mathbf{W} \cdot \mathbf{F} = w_1 \cdot F_1 + w_2 \cdot F_2 + \cdots + w_k \cdot F_k \tag{3}$$

where $\boldsymbol{Y}$ is a response variable, described by the averaged feature map $\boldsymbol{F}$ of extracted feature maps and weight parameter $\boldsymbol{W}$. Note that $\boldsymbol{W}$ can be considered as the coefficients that enable us to identify the significant variables in the regression model. After training the model, we use the trained weight $\boldsymbol{W}^*$ as the importance for each F_k when deriving the activation map.

We train the CNN model such that the following cost function L (mean squared error) is minimized:

$$L = \frac{1}{N} \sum_{n=1}^{N} \left(y_n - \hat{y}_n\right)^2 \tag{4}$$

where N is the number of observations, y_n is the nth response variable, and $\hat{y}_n$ is the predicted value, which is the output of the CNN. The CNN is trained using an Adam optimizer, which is an algorithm for the first-order gradient-based stochastic objective functions.

We now present the proposed quality-discriminative localization for RCA. The purpose of the proposed method is to visually emphasize the sensor signals that represent the root causes of abnormal quality. As shown in Fig. 1, we first derive causal maps with activation mapping for all products, and then we identify the root causes with the localized sensor signals in the RCM. Having trained the CNN, we conduct the activation mapping to produce the following causal maps from the weighted sum of $|w_k^*|$ and $f_k(t, s)$:

$$CM_n(t, s) = \sum_{k=1}^{K} f_k^n(t, s) \cdot \left|w_k^*\right| \tag{5}$$

where $|w_k^*|$ is the absolute value of trained weights w_k^*, which can be considered the importance of each feature map $f_k(t, s)$ [36]. The $CM_n(t, s)$ score highlights the important regions of the input data corresponding to the label. Considering that the regression model determines important variables based on the magnitude of weights, we use the absolute value of w_k^*. $CM_n(t, s)$ explains the causes of a product's predicted quality by localizing sensor signals. If a region contains high $CM_n(t, s)$ scores, the corresponding region's sensor signals can be considered significant causes.

In the present study, we aim to identify the root causes, whose processes and processing times indicate an increase in weight deviation. Thus, we obtain a quality-weighted activation map, called the RCM. By computing the weighted sum of weight deviation and causal maps, the RCM represents the most critical causes of abnormal quality. The RCM is calculated as follows:

$$\mathrm{RCM}(t, s) = \sum_{n}^{N} CM_n(t, s) \cdot \hat{y}_n \tag{6}$$

where $CM_n(t, s)$ is an activated map of the nth observation, and $\hat{y}_n$ is a predicted value. Having constructed the RCM, we examine the localized positions that have scores exceeding percentile h. Thus, we interpret that the

TABLE I
LOCATIONS OF ROOT CAUSES

Dataset		Root Causes Types								
1	Process ID	13	14	15	-	-	-	-	-	-
	Processing time	40~70	40~70	40~70	-	-	-	-	-	-
	Cause type	A	B	C	-	-	-	-	-	-
2	Process ID	13	13	14	14	15	15	-	-	-
	Processing time	10~30	70~90	10~30	70~90	10~30	70~90	-	-	-
	Cause type	A	A	B	B	C	C	-	-	-
3	Process ID	0	1	2	3	4	26	27	28	29
	Processing time	10~40	10~40	10~40	10~40	10~40	60~90	60~90	60~90	60~90
	Cause type	A	A	A	B	B	C	C	C	C
4	Process ID	10	10	11	11	12	12	-	-	-
	Processing time	10~30	80~100	10~30	80~100	10~30	80~100	-	-	-
	Cause type	A	A	B	B	C	C	-	-	-
5	Process ID	4	5	6	14	15	16	24	25	26
	Processing time	0~50	10~60	20~70	0~50	10~60	20~70	0~50	10~60	20~70
	Cause type	A	A	A	B	B	B	C	C	C
6	Process ID	0	1	2	10	11	12	22	21	22
	Processing time	30~60	30~60	30~60	40~70	40~70	40~70	50~80	50~80	50~80
	Cause type	A	A	A	B	B	B	C	C	C
7	Process ID	5	6	7	8	9	10	-	-	-
	Processing time	0~50	50~100	10~60	40~90	20~70	30~80	-	-	-
	Cause type	A	A	B	B	C	C	-	-	-
8	Process ID	11	12	13	24	25	26	-	-	-
	Processing time	0~50	50~100	20~70	0~50	50~100	20~70	-	-	-
	Cause type	A	A	B	B	C	C	-	-	-
9	Process ID	6	7	8	19	20	21	27	27	28
	Processing time	0~30	70~100	30~60	0~30	70~100	30~60	0~30	70~100	30~60
	Cause type	A	A	A	B	B	B	C	C	C

Algorithm 1 Quality-Discriminative Localization

Input: Data $\mathbf{X}$ and $\mathbf{y}$;
Output: $RCM^*(t, s)$
▷ Train the model
$w \leftarrow$ Initialize the parameters
Repeat
  $\hat{\mathrm{y}} \leftarrow \mathrm{Model(X)} = F_k \cdot w_k$
  $L = \frac{1}{N} \cdot \sum_{n=1}^{N} \left(y_n - \hat{y}_n\right)^2$
  $w \leftarrow$ Update the parameters using the gradients of L
until convergence of the parameters using L
▷ Construct the RCM
$RCM(t, s) \leftarrow$ Initialize elements of the t by s matrix to zero
for n=1 to N do
  $CM_{\mathrm{n}}(t, s) \leftarrow \sum_{k=1}^{K} f_k^n(t, s) \cdot |w_k^*|$
  $RCM(t, s) \leftarrow RCM(t, s) + CM_{\mathrm{n}}(t, s) \cdot \hat{y}_n$
end for
▷ Identify the root causes
Let h be a hyperparameter that indicates a threshold as a percentile.
  If $RCM(t, s) > h$ **then** $RCM(t, s)$ = root cause's location
$RCM^*(t, s) \leftarrow$ root cause's location

localized sensor signal is an indicator of the cause of $\hat{y}_n$. The causal maps show the causes of defects in the products, and the RCM explains the root causes accounting for all product quality. Algorithm 1 shows the procedure of quality-discriminative localization.

IV. SIMULATION STUDY

We evaluated the quality-discriminative localization on simulated datasets in terms of its predictive performance and localization ability.

A. Simulated Data

We generated nine datasets composed of 1000 observations, which consist of multiple sensor signals of 30 × 100 size, representing the process states of a manufacturing system with 30 processes and 100 processing times. We first sampled the sensor signals from a normal distribution with mean 0 and standard deviation 0.1, representing the normal operation of processes. We also sampled the response variable, indicating a manufacturing system's defect rates by sampling 1000 values from a normal distribution with mean 60 and standard deviation 20. We then generated abnormal signals of three cause types by sampling values from a sine wave according to the defect rates. The cause types (A), (B), and (C) were generated from defect rates × sine wave of causal time length, defect rates + sine wave of causal time length, and defect rates × sine wave (from $-\pi$ to π) of causal time length, respectively, which represent many fluctuations, a broad-scale peak, and a large-scale fluctuation, as shown in Fig. 2.

We then associated them with root cause locations. Table I lists the root cause locations for each dataset under different scenarios: 1) consecutive causal processes have the same causal processing times (see Datasets 1, 2, and 3); 2) nonconsecutive causal processes have different causal processing times (see Datasets 4, 5, and 6); and 3) all of the causal processes have different causal processing times (see Datasets 7, 8, and 9). They indicate that causal processes and processing times can occur from different abnormal signal patterns and multiple causal locations in a manufacturing system. Using these datasets, we examined the predictive performance and localization ability.

We conducted experiments with five CNN architectures to investigate the quality-discriminative localization ability of various feature extraction architectures, including a fully convolutional network (FCN), U-Net, a deconvolutional network (DeconvNet), and SegNet. These are popular CNN architectures for image segmentation that perform pixelwise classification and can extract feature maps of the same size

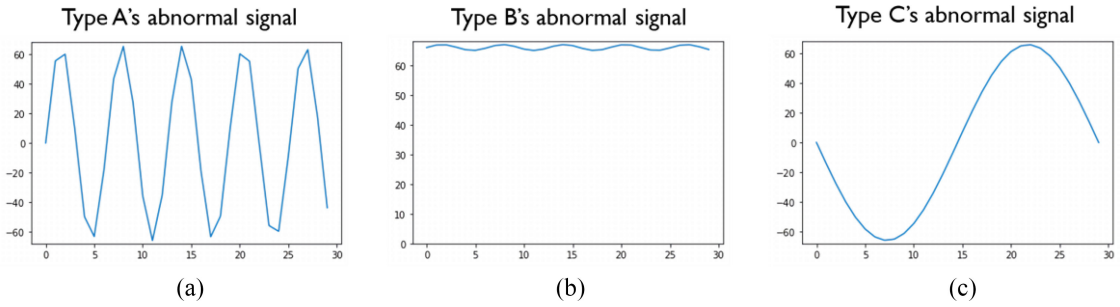

Fig. 2. Examples of three types of abnormal signals with 30 causal time lengths and 66 defect rates. Each type of abnormal signal is generated from (a) 66 × sine wave of length 30, (b) 66 + sine wave of length 30, and (c) 66 × sine wave (from $-\pi$ to π) of length 30.

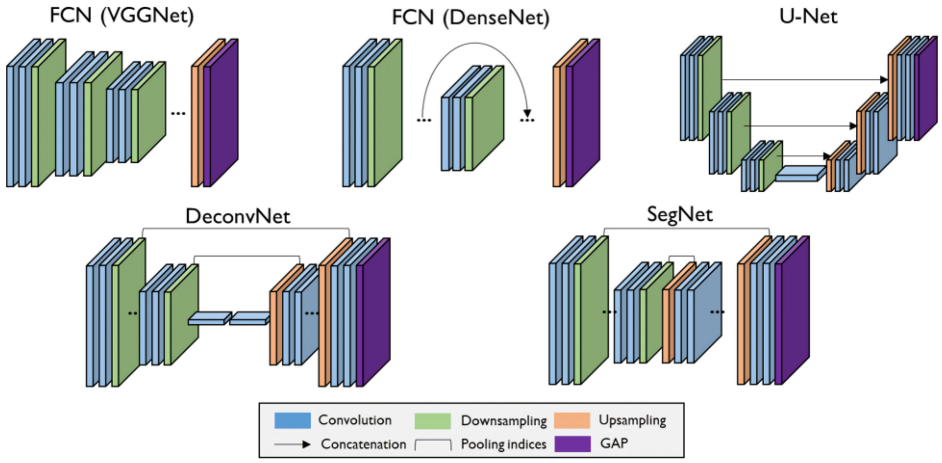

Fig. 3. Structures of feature extraction networks.

as the input data. Fig. 3 shows the structures of these feature extraction networks.

The FCN consists of a convolutional network classifier followed by an upsampling layer, performing bilinear interpolation [37]. We used VGGNet and DenseNet for the FCN [38], [39]. VGGNet has convolution blocks, each containing two convolution layers and a max-pooling layer. DenseNet is composed of dense blocks, performing concatenation of the previous feature maps and subsequent feature maps. A dense block also has two convolution layers, and each convolution layer is fed into a batch normalization (BN) layer with a rectified linear unit (ReLU) activation function. A dense block is followed by a transition layer consisting of a BN layer and a 1 × 1 convolutional layer, followed by an average pooling layer.

U-Net has a U-shaped network with a contracting path and an expansive path. The contracting path consists of convolution blocks, where each block has two convolutional and BN layers, followed by a max-pooling layer. The expansive path consists of feature map upsampling, followed by convolution blocks. The upsampled feature maps are then concatenated with the corresponding feature map of the convolution blocks in the contracting path [40].

DeconvNet has an encoder-decoder structure, consisting of convolution blocks and deconvolution blocks, respectively, [41]. The deconvolution operation is identical to the convolution operation but is hierarchically opposite. Each block has two consecutive convolution operations, followed by a BN layer with the ReLU activation function. Before a deconvolution block, two fully connected layers are added, and then an unpooling layer, which restores the max-pool indices in every step of the decoder's convolution operation, is applied.

As with DeconvNet, SegNet has an encoder–decoder structure, but it uses only convolution blocks in both the encoder and decoder and does not use any fully connected layers [42].

We employed a smaller version of each model to consider the size of the simulated data and to minimize the difference in the numbers of parameters between models. We used only one convolution block in the FCNs. We also used one convolution block for each downsampling and upsampling operation in U-Net, DeconvNet, and SegNet. Each convolutional layer was set to generate 32 feature maps with 2-D convolutional kernels of 3 × 3 size. Moreover, we used 2-D filters of 2 × 2 size for the downsampling and upsampling layers. Then, to enable them to perform the regression model-based RCA, we reshaped their prediction networks to consist of a GAP layer and a dense layer with a linear activation function.

B. Evaluation of the Predictive Performance

We performed fivefold cross-validation using 80% and 20% as the training and testing data, respectively. We trained the CNN with a batch size of eight and epoch size of 100 for all datasets. We also implemented an early stopping criterion, where if the validation loss did not improve after 30 epochs, the training would terminate. All models

TABLE II
AVERAGE PREDICTIVE PERFORMANCES IN FIVEFOLD CROSS-VALIDATIONS. STANDARD DEVIATIONS ARE PRESENTED IN PARENTHESES

Dataset	FCN-VGGNet+GAP			FCN-DenseNet+GAP			U-Net+GAP			DeconvNet+GAP			SegNet+GAP		
	R^2	MAE	RMSE	R^2	MAE	RMSE	R^2	MAE	RMSE	R^2	MAE	RMSE	R^2	MAE	RMSE
1	0.95 (0.01)	3.13 (0.61)	4.16 (0.45)	0.99 (0.00)	1.41 (0.27)	1.95 (0.63)	0.50 (0.08)	10.55 (1.09)	13.93 (1.08)	0.92 (0.06)	4.22 (2.03)	5.4 (1.76)	0.92 (0.02)	3.7 (0.36)	5.39 (0.72)
2	0.93 (0.03)	4.48 (0.83)	5.35 (1.07)	0.98 (0.01)	1.99 (0.59)	2.79 (1.06)	0.74 (0.05)	7.51 (1.09)	10.09 (1.06)	0.92 (0.01)	3.53 (0.19)	5.47 (0.56)	0.94 (0.03)	3.17 (0.65)	4.68 (1.10)
3	0.92 (0.06)	4.56 (2.11)	5.19 (2.23)	0.99 (0.00)	0.92 (0.28)	1.18 (0.29)	0.71 (0.06)	7.74 (1.27)	10.71 (1.41)	0.97 (0.01)	2.05 (0.30)	3.19 (0.62)	0.97 (0.04)	2.24 (2.01)	2.8 (1.96)
4	0.95 (0.03)	3.15 (1.02)	4.29 (1.32)	0.98 (0.02)	2.39 (1.26)	2.92 (1.26)	0.69 (0.11)	7.69 (1.83)	10.95 (2.00)	0.92 (0.02)	3.91 (0.49)	5.47 (0.85)	0.94 (0.02)	3.45 (0.6)	4.71 (0.72)
5	0.94 (0.01)	4.08 (0.51)	4.91 (0.54)	0.99 (0.00)	1.64 (0.26)	2.15 (0.37)	0.94 (0.03)	2.98 (1.08)	4.87 (1.08)	0.98 (0.01)	2.07 (0.62)	2.72 (0.81)	0.99 (0.01)	1.55 (1.06)	2.14 (0.96)
6	0.94 (0.04)	3.94 (1.74)	4.54 (1.73)	0.99 (0.00)	1.53 (0.23)	1.98 (0.14)	0.81 (0.03)	5.68 (0.62)	8.57 (0.89)	0.98 (0.01)	2.07 (0.15)	2.94 (0.55)	0.99 (0.00)	1.33 (0.39)	2.01 (0.52)
7	0.95 (0.01)	3.21 (0.68)	4.29 (0.46)	0.96 (0.05)	3.03 (2.12)	3.54 (2.07)	0.84 (0.09)	5.54 (2.31)	7.57 (1.78)	0.95 (0.03)	3.24 (1.02)	4.22 (1.34)	0.98 (0.02)	1.81 (0.93)	2.64 (0.98)
8	0.93 (0.02)	4.29 (0.84)	5.09 (0.69)	0.98 (0.01)	2.12 (0.31)	2.88 (0.69)	0.97 (0.01)	1.36 (0.18)	3.03 (0.48)	0.96 (0.01)	2.78 (0.41)	3.72 (0.47)	0.99 (0.00)	1.33 (0.19)	1.79 (0.35)
9	0.95 (0.04)	3.65 (1.64)	4.32 (1.46)	0.99 (0.01)	1.32 (0.30)	1.72 (0.41)	0.98 (0.01)	1.33 (0.20)	2.79 (0.47)	0.98 (0.01)	2.28 (0.52)	3.17 (0.56)	0.99 (0.01)	1.46 (0.71)	1.96 (0.70)
Average	0.94 (0.03)	3.83 (1.11)	4.68 (1.11)	**0.98 (0.01)**	**1.81 (0.62)**	**2.35 (0.77)**	0.80 (0.05)	5.60 (1.07)	8.06 (1.14)	0.95 (0.02)	2.91 (0.64)	4.03 (0.84)	0.97 (0.02)	2.23 (0.77)	3.12 (0.89)

were programmed in Python using Keras and a TensorFlow 2.0 backend. We conducted all experiments on a workstation equipped with an Intel Core I9-9900 CPU @ 3.10 GHz, 64-GB DDR4 RAM 3200 MHz, NVIDIA GeForce RTX 2080 Ti with 4352 CUDA cores, and Windows 10 Enterprise operating system.

We used performance measures of R^2, mean absolute error (MAE), and root mean squared error (RMSE) as follows:

$$R^2 = \frac{\sum_{n=1}^{N} \left(\hat{y}_n - \bar{y}\right)^2}{\sum_{n=1}^{N} \left(y_n - \bar{y}\right)^2} \tag{7}$$

$$\text{MAE} = \frac{1}{N} \sum_{n=1}^{N} \left|y_n - \hat{y}_n\right| \tag{8}$$

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{n=1}^{N} \left(y_n - \hat{y}_n\right)^2} \tag{9}$$

where y_n and $\hat{y}_n$ are the actual and predicted values of the nth observation, respectively. R^2 represents the coefficient of determination calculated by the square of the correlation between y and $\hat{y}$. MAE is the average value over the validation data of the absolute differences between the actual and predicted values, where all individual differences have equal weights. RMSE is the square root of the mean squared difference between the actual and predicted values for the validation data.

Table II lists the averages and standard deviations of the fivefold cross-validation results. The average R^2, MAE, and RMSE values of all datasets are listed in the last row. Although the FCN-DenseNet network showed the most accurate performance, with average R^2, MAE, and RMSE values of 0.98, 1.81, and 2.35, respectively, the performances of the other methods are comparable. We can conclude that our simulated sensor data explain a response variable well under various scenarios, and this opens the possibility of performing RCA based on such models.

TABLE III
AUC SCORES OF THE QUALITY-DISCRIMINATIVE LOCALIZATION

Dataset	FCN-VGGNet +GAP	FCN-DenseNet +GAP	U-Net +GAP	DeconvNet +GAP	SegNet +GAP
1	0.36	0.50	0.29	0.97	0.99
2	0.54	0.46	0.31	0.95	0.99
3	0.39	0.54	0.24	0.98	1.00
4	0.51	0.25	0.01	0.97	0.99
5	0.61	0.67	0.44	0.77	0.67
6	0.26	0.26	0.38	0.95	0.98
7	0.55	0.80	0.19	0.90	0.92
8	0.48	0.58	0.40	0.77	0.82
9	0.44	0.50	0.53	0.65	0.71
Average	0.46	0.51	0.31	0.88	0.90
S.D.	0.10	0.16	0.15	0.11	0.12

C. Evaluation of the Localization Ability

In the simulation study, the specific threshold h, determining how many causal pixels would be localized, has not been set because we compare the localization abilities of CNN architectures for all ranges of thresholding values based on the area under the curve (AUC). The AUC indicates the area under the receiver operating characteristic (ROC) curve. The ROC curve is created by plotting the true-positive rate (TPR) against the false-positive rate (FPR) under various threshold settings. The TPR is the rate of correctly predicted locations of root causes based on a specific threshold h, and FPR has the opposite meaning. The AUC can be defined as follows:

$$\text{AUC} = \int_{h=0}^{1} \text{TPR}\left(\text{FPR}^{-1}(h)\right) dh \tag{10}$$

where the h is the threshold of quality-weighted activation score in an RCM; we generate causal maps and RCMs using fivefold test data for

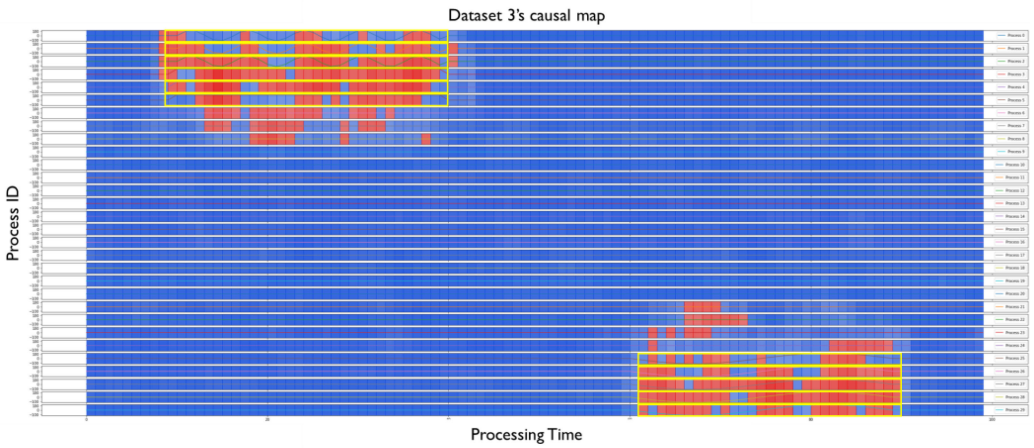

Fig. 4. Illustration of a causal maps: the red regions indicate more significant regions than blue regions, and the yellow bounding boxes indicate regions of abnormal signals.

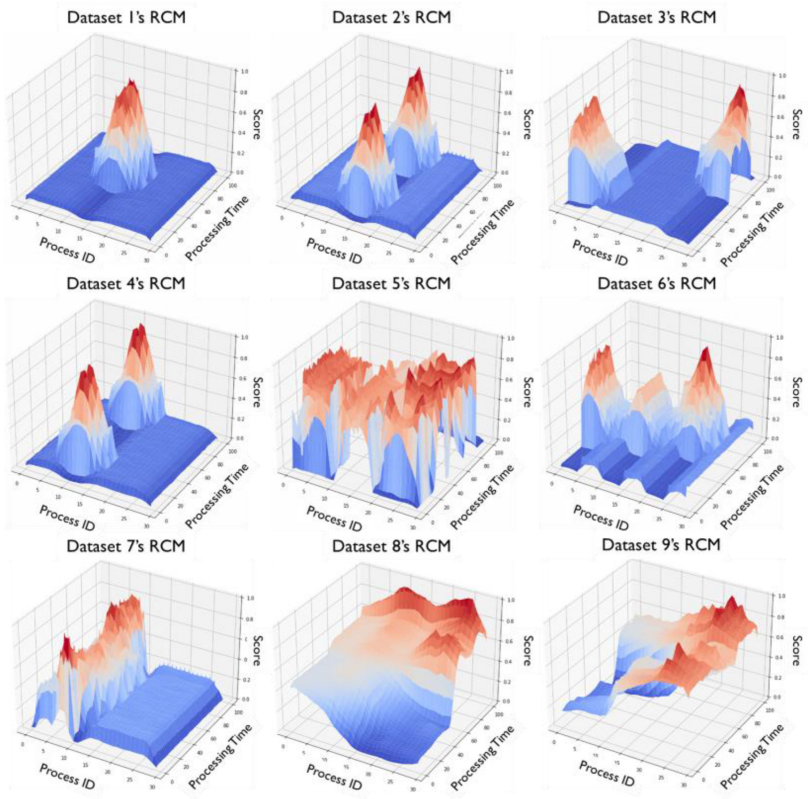

Fig. 5. Illustration of RCMs generated from the SegNet-based quality-discriminative localization.

each dataset, and we report the AUC scores in Table III.

The SegNet architecture outperformed the other methods, and we found that the encoder–decoder structure, performing more convolution operations during upsampling with max-pool indices, shows higher accuracy in root cause localization. Fig. 4 jointly illustrates the multisensor signals and a causal map, highlighting abnormal signals of the highest defect rate product in the Dataset 3 in which the SegNet architecture showed the most accurate AUC score. Fig. 5 illustrates RCMs generated from the quality-discriminative localization for all datasets, highlighting the root cause regions. We can confirm that the proposed method correctly emphasizes causal regions.

V. Case Study

A. Steel Manufacturing Process Data

To evaluate the applicability of the proposed method in the real world, we conducted experiments on nine datasets collected from a steel manufacturing process. The steel manufacturing process produces steel from iron ore and scrap.

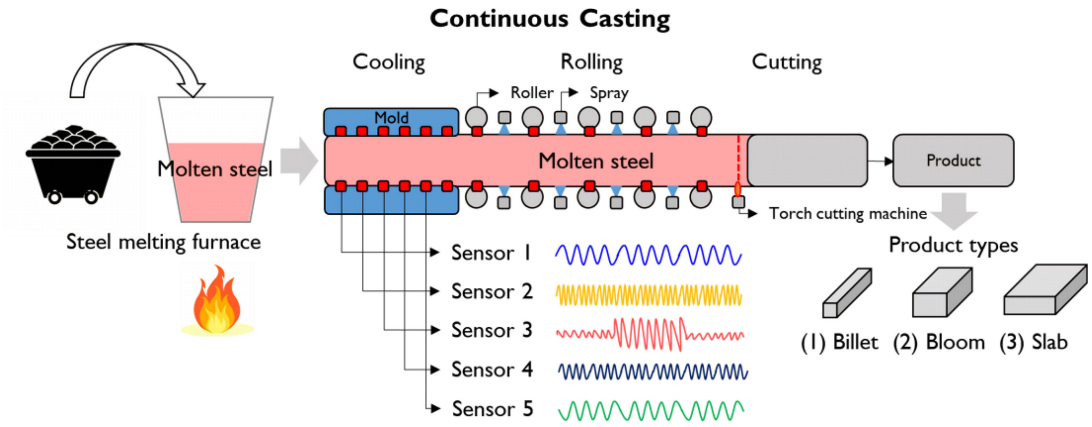

Fig. 6. Illustration of data collection from the continuous casting in the steel manufacturing process.

TABLE IV
CONFIGURATION OF SENSOR VARIABLES OF CONTINUOUS CASTING PROCESS

Process	Sensor types
Cooling process	Mold's oscillation, mold's cooling water temperature, pressure, inflow rate, etc.
Rolling process	Cooling water spray's set / actual / control value, cooling air spray's set / actual / control value, roller's set / actual pressure, roller's set / actual speed, etc.
Cutting process	Torch cutting machine's speed, length, etc.

TABLE V
SUMMARY OF THE 2-D MULTISENSOR SIGNAL DATASETS

Dataset	Product type	Number of products	Number of processes	Processing time
1	Billet	397	154	100
2	Billet	253	148	100
3	Bloom	532	170	100
4	Bloom	906	153	100
5	Bloom	1,017	152	100
6	Bloom	916	153	100
7	Bloom	771	201	100
8	Slab	949	183	100
9	Slab	934	177	100

The entire process is divided into four subprocesses: 1) steelmaking; 2) refining; 3) continuous casting; and 4) forming. Steelmaking involves the input of raw materials that are melted in a blast furnace. Refining reduces the impurities that can make the resulting molten steel brittle. Next, continuous casting places the molten steel into a cooled mold, causing it to solidify into a thin steel shell. It is then made into an intermediate-stage product, such as a billet, bloom, or slab. The steel is formed into various shapes, often by hot rolling, eliminating cast defects, and achieving the required shape.

In this study, we focused on RCA for the continuous casting process, which plays an important role in determining the quality of steel. Fig. 6 illustrates a procedure for collecting multisensor signal datasets in continuous casting, consisting of the cooling, rolling, and cutting processes. Once the molten steel enters the continuous casting process, the cooling process decreases its temperature using mold's cooling water. It is then cooled and rolled into the shape of an intermediate-stage product, billet, bloom, and slab, using cooling sprays and rollers in the rolling process. In the cutting process, the molten steel is cut to the desired length using a torch cutting machine. We measure the difference between the intermediate-stage product's weight and its target weight after the cutting process. This study defines this weight deviation as indicating the final steel product's quality, enabling us to address a regression problem.

We collected nine datasets from multiple sensors attached to each piece of processing equipment. Table IV shows the sensor types according to the procedure of the continuous casting process: 1) cooling process sensors attached to the rectangular mold's four axes (east, west, south, and north), and the sensors measure the temperature, pressure, inflow rate of cooling water and mold's oscillation; 2) rolling process sensors measure the pressure and speed of the roller, values of cooling water sprays and cooling air sprays; and 3) cutting process sensors measure the length and speed of torch cutting machine during process operation.

Using these sensors, we obtained sensor data of three product types, billet, bloom, and slap, having different widths each other, to verify the robustness and applicability of the proposed method under various process states. Table V shows a summary of the datasets. We then transformed the multiple sensor signals into 2-D data. As shown in Fig. 7, we first collect sensor data for all processes, and then we sampled the time steps of 100 according to the process sequence to form the CNN's input data.

Consequently, each product had a 2-D dataset consisting of the process ID on the y-axis and processing time on the x-axis. The response variable is the weight deviation, measured after the cutting process, and its range is -25 to 120, as shown in Fig. 8. The larger the value of the response variable (weight deviation), the lower the quality. Using these datasets, we first evaluate the predictive performance and then examine the ability of quality-discriminative localization.

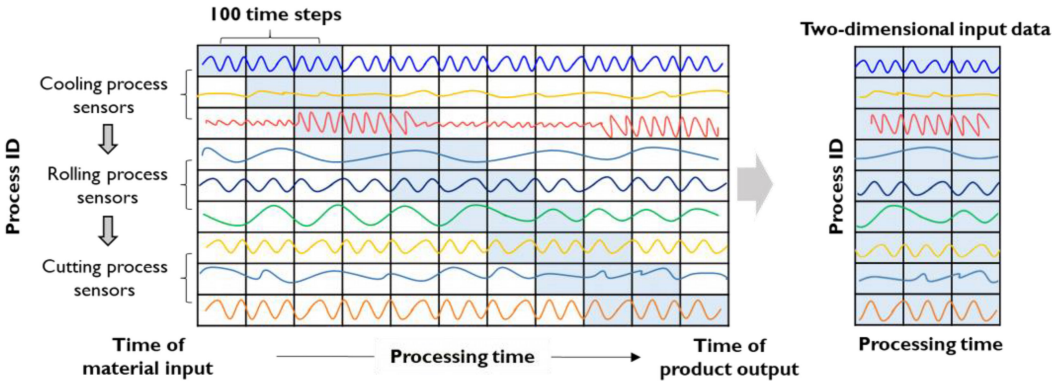

Fig. 7. Illustration of the transformation of multisensor signals to 2-D input data.

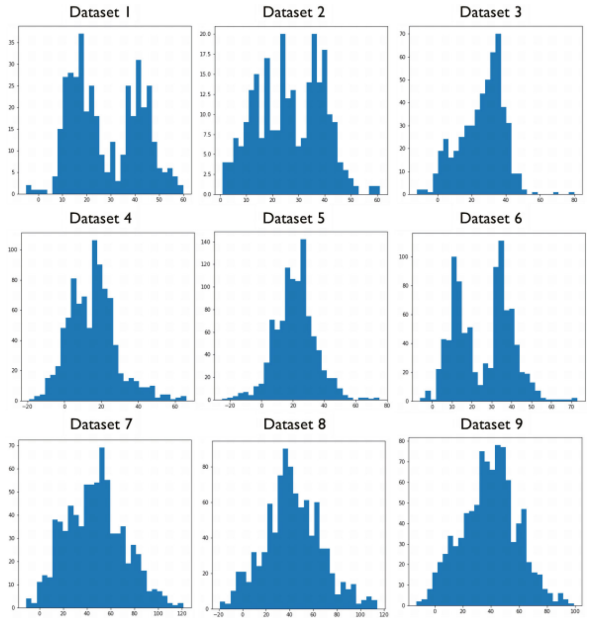

Fig. 8. Histograms of response variables (weight deviations) for each dataset.

TABLE VI
AVERAGE PREDICTIVE PERFORMANCES OF SEGNET+GAP IN FIVEFOLD CROSS-VALIDATIONS. STANDARD DEVIATIONS ARE PRESENTED IN PARENTHESES

Dataset	SegNet+GAP		
	R^2	MAE	RMSE
1	0.69 (0.05)	5.45 (0.34)	7.88 (0.57)
2	0.68 (0.04)	5.11 (0.32)	7.07 (0.62)
3	0.75 (0.02)	4.81 (0.23)	6.49 (0.36)
4	0.84 (0.04)	3.93 (0.19)	5.31 (0.45)
5	0.78 (0.02)	4.66 (0.17)	6.04 (0.18)
6	0.88 (0.01)	3.52 (0.15)	4.83 (0.16)
7	0.75 (0.03)	9.19 (0.29)	12.23 (0.74)
8	0.81 (0.03)	8.05 (0.71)	10.62 (0.71)
9	0.75 (0.03)	7.48 (0.49)	9.90 (0.76)
Average	0.77 (0.03)	5.80 (0.32)	7.82 (0.51)

B. Evaluation of the Predictive Performance

We evaluated the predictive performance of SegNet-based CNN, containing three convolution blocks for each downsampling and upsampling operation. We performed five-fold cross-validation using 80% and 20% as the training and testing data, respectively. We also trained the CNN with a batch size of 8 and epoch size of 100 for all datasets and implemented an early stopping criterion, where if the validation loss did not improve after 30 epochs, the training would terminate.

Table VI presents the averages and standard deviations of the results of fivefold cross-validation. The average R^2, MAE, and RMSE values of all datasets are listed in the last row. The average R^2, MAE, and RMSE were 0.77, 5.80, and 7.82, respectively, indicating that the SegNet structure is sufficiently robust to explain the relationship between the sensor signals and product quality in various processes states.

C. Evaluation of the Localization Ability

With the regression-trained CNN, we performed quality-discriminative localization for RCA. We first extracted causal maps, which can provide temporary causes for each product, and then we generated an RCM to identify the root causes. Fig. 9 shows two examples of causal maps, where the (left) causal map has the lowest weight deviation, and the (right) causal map has the highest weight deviation on Dataset 1. The red highlighted regions exhibit more critical factors in the causal maps than the blue regions in predicting weight deviation. We considered that the red highlighted sensor signals represent the causal process states of predicted quality; thus, these regions enabled us to derive the location of abnormal sensor signals for the process ID and processing time.

As shown in Fig. 10, we generated RCMs based on the weighted sum of quality and causal maps and derived three-dimensional (3-D) activation maps for each dataset. In each RCM, the x-, y-, and z-axes indicate the process ID, processing time, and quality-weighted activation score, respectively. Min-max normalization was performed to represent the quality-weighted activation scores between 0 and 1. We arbitrarily set the threshold h as 90th percentile to derive the most significant sensor signals. It depends on the number of causal factors that need to be analyzed. If the threshold h is set at a lower percentile, more root cause sensors and processing time are detected. Our empirical results indicate that 90th or 95th works well in most cases. The red highlighted regions of the RCMs in Fig. 10 indicate major causes of abnormal quality.

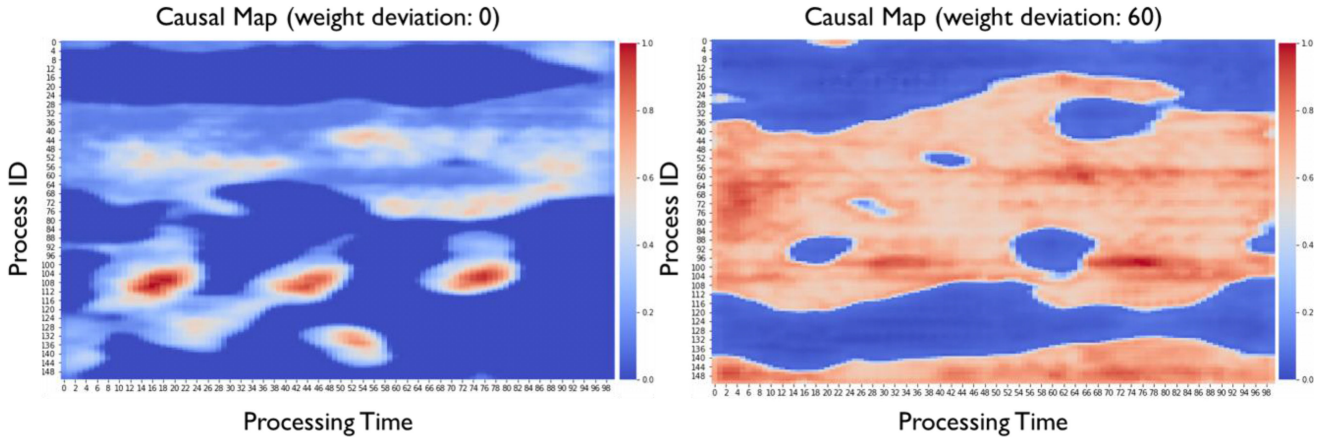

Fig. 9. Causal maps: (left) causal map of a steel product with the lowest weight deviation and (right) causal map of a steel product with the highest weight deviation. The red regions indicate more important regions for predicting weight deviations.

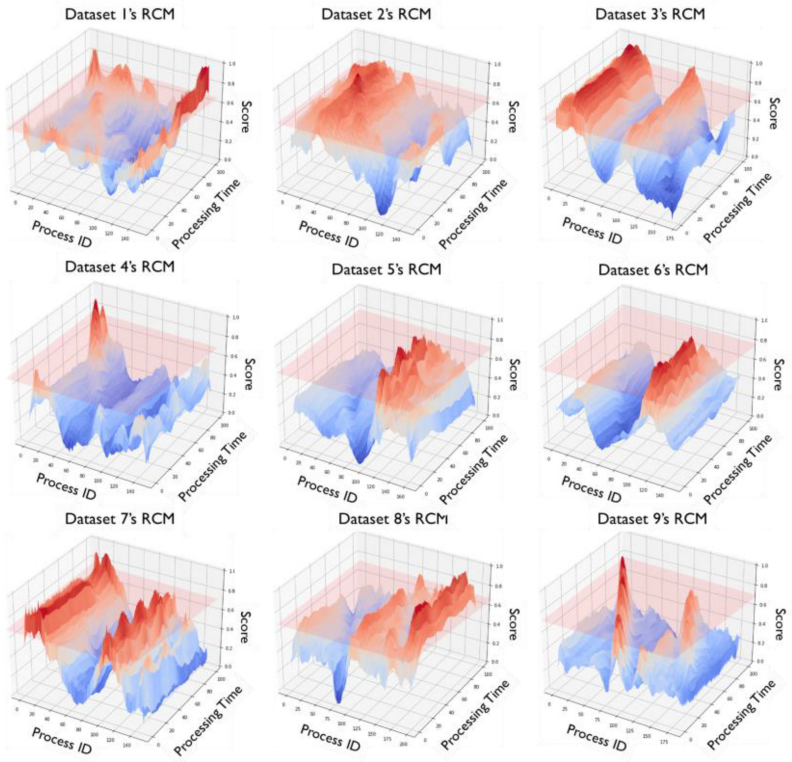

Fig. 10. Results of the RCMs. The red plane indicates the 90th percentile threshold of quality-weighted activation scores.

To statistically verify that the localized regions of process ID and processing time exhibit significant causal sensor signals, we conducted a two-sample t-test to compare the sensor signals. We grouped the 100 signals into normal and abnormal classes, with 50 signals in each class. We summarized each sensor signal using eight statistical categories: 1) mean; 2) variance; 3) min; 4) max; 5) median; 6) range; 7) area; and 8) correlation coefficient. The area represents the summation of a signal for all time steps. We expected that if each class had distinctive signal patterns, such statistics could represent the characteristics of the sensor signals. We also conducted a correlation analysis to derive a correlation coefficient for considering the overall time information. Correlation analysis was performed between different sensor signals, from which we obtained a correlation coefficient matrix. We used an upper triangular matrix, excluding diagonal elements, as the input to the two-sample t-test and expected that the correlation coefficients of the sensor signals in the same class would be higher than those from different classes.

Using these summarized statistics, we performed a two-sample t-test to confirm the differences between the normal and abnormal groups. The t statistic was calculated using the following equation:

$$t = \frac{\overline{X}_{\text{normal}} - \overline{X}_{\text{abnormal}}}{\sqrt{(S\_\text{normal}\text{\^{}}2)/N\_\text{normal} - (S\_\text{abnormal}\text{\^{}}2)/N\_\text{abnormal}}} \tag{11}$$

TABLE VII
p-VALUES OF THE TWO-SAMPLE *t*-TEST FOR THE SIGNIFICANT SENSORS; THOSE IN BOLDFACE ARE LESS THAN 0.05

Dataset	Process ID	*p*-value							
		Mean	Min	Max	Variance	Median	Range	Area	Correlation Coefficient
1	151	**0.03**	-	**0.01**	**0.01**	-	**0.01**	**0.03**	-
	150	**0.03**	**0.01**	-	**0.01**	-	**0.01**	**0.03**	-
	149	0.56	-	0.70	0.93	-	0.70	0.56	-
2	44	**0.00**	**0.00**	**0.01**	**0.01**	**0.00**	**0.04**	**0.00**	0.79
	48	-	-	-	**0.00**	-	**0.00**	-	0.48
	47	0.07	0.81	**0.00**	-	-	-	**0.07**	0.59
3	28	**0.00**	**0.00**	**0.00**	**0.04**	**0.00**	**0.01**	**0.00**	0.69
	29	**0.01**	**0.01**	**0.01**	-	**0.01**	-	**0.01**	-
	27	**0.00**	**0.00**	**0.00**	**0.03**	**0.00**	**0.01**	**0.00**	0.89
4	16	**0.02**	0.22	**0.01**	**0.00**	**0.01**	**0.00**	**0.02**	0.37
	15	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.09
	14	0.08	0.13	**0.03**	**0.00**	0.09	0.07	0.08	0.44
5	119	0.88	-	-	0.60	-	-	0.88	**0.00**
	118	0.39	-	-	0.63	-	-	0.39	**0.00**
	117	**0.00**	**0.00**	**0.00**	**0.03**	**0.00**	**0.03**	**0.00**	-
6	20	**0.00**	**0.00**	**0.00**	0.62	**0.00**	0.67	**0.00**	0.50
	23	0.65	0.40	0.90	**0.05**	0.64	**0.02**	0.65	0.25
	24	0.65	0.43	0.91	**0.05**	0.64	**0.03**	0.65	0.30
7	181	0.22	0.80	0.15	0.25	0.17	0.41	0.22	0.97
	180	**0.00**	**0.00**	**0.00**	0.36	**0.01**	**0.00**	**0.00**	**0.00**
	179	**0.00**	-	-	**0.01**	**0.00**	-	**0.00**	0.17
8	52	**0.00**	**0.00**	**0.00**	0.97	**0.00**	0.39	**0.00**	0.81
	51	**0.00**	**0.00**	**0.00**	0.13	**0.00**	**0.06**	**0.00**	0.08
	125	-	-	-	-	-	-	-	-
9	66	**0.00**	**0.00**	**0.00**	**0.03**	**0.00**	**0.04**	**0.00**	0.33
	65	**0.00**	**0.00**	**0.00**	**0.01**	**0.00**	**0.02**	**0.00**	0.86
	63	**0.00**	**0.00**	**0.00**	**0.02**	**0.00**	**0.02**	**0.00**	0.27

TABLE VIII
PREDICTIVE PERFORMANCES OF SEGNET+GAP FOR THE LOCALIZED REGIONS LEARNED BY THE RCM

Dataset	SegNet+GAP		
	R^2	MAE	RMSE
1	0.71 (0.05)	5.29 (0.40)	7.59 (0.67)
2	0.65 (0.06)	5.45 (0.54)	7.38 (0.91)
3	0.74 (0.04)	4.75 (0.35)	6.47 (0.5)
4	0.82 (0.05)	4.09 (0.39)	5.57 (0.62)
5	0.71 (0.05)	5.16 (0.25)	6.84 (0.48)
6	0.86 (0.02)	3.79 (0.26)	5.13 (0.35)
7	0.73 (0.03)	9.46 (0.63)	12.63 (0.80)
8	0.74 (0.04)	9.5 (0.67)	12.42 (0.88)
9	0.65 (0.08)	8.79 (1.28)	11.62 (1.66)
Average	0.73 (0.05)	6.25 (0.53)	8.41 (0.76)

where N_{normal} and N_{abnormal} are the sample sizes, $S^2{}_{\text{normal}}$ and $S^2{}_{\text{abnormal}}$ are the sample variances, and $\overline{X}_{\text{normal}}$ and $\overline{X}_{\text{abnormal}}$ indicate the average values of the statistics from the sensor signals. We assumed that the statistics do not have equal variance.

Table VII shows the resulting *p*-values of the two-sample *t*-tests. In each dataset, we listed three sensors that rendered high activation scores. The *p*-values, which are less than 0.05, are highlighted in bold. In some cases, we could not obtain the *p*-values because of the inflated *t* statistics caused by zero variance (indicated by the hyphens in Table VII). Most sensors exhibit *p*-values less than 0.05, implying that the selected sensors have distinct features of the signals. However, we observed some cases with nonsignificant *p*-values (e.g., process 149 of dataset 1, process 181 of dataset 7, and process 125 of dataset 8). We believe that these cases cannot be sufficiently explained based on the statistics used in this study. Nevertheless, the results showed that most localized signals exhibited significant differences.

We also evaluated the CNN's predictive performance using the multisensor signals located at the localized regions in the RCM. We selected 50 sensors with the highest quality-weighted activation scores for each dataset, and then we conducted fivefold cross-validation for each selected dataset. We found that the average R^2, MAE, and RMSE were 0.73, 6.25, and 8.41, respectively, (Table VIII). These results are comparable with the predictive CNN results using all regions (see Table VI), indicating that the localized regions learned by the RCM are significant for characterizing a root cause.

VI. CONCLUSION

We proposed an RCA method, called quality-discriminative localization. With multisensor signal data, a regression-trained CNN provides productwise causal maps and generates an RCM to identify the most significant causal regions in multivariate processes. The causal maps can be used for monitoring the causal processes and processing times for all products. Then, the RCM can be applied for identifying the root causes as processes that frequently cause abnormal quality. Consequently, the proposed method enables us to interpret multisensor signals by highlighting distinctive patterns. In addition, using simulated and real-world process sensor datasets, we demonstrated that the proposed

method exhibits applicability and robustness with satisfactory predictive performance and localization ability.

Although the proposed method shows promising results, the activation mapping using the output of a GAP layer has a limitation in that it leads to information loss because the GAP layer summarizes the final feature maps. Nevertheless, the CNN learns the parameters for prediction, even when using the summarized values of the GAP, and verifying the predictive performance before conducting the activation mapping is essential. The proposed CNN also has certain limitations posed by the 2-D kernels in the convolution process. The 2-D kernel extracts a pixelwise element of the local input. When we perform root cause identification, the pixelwise activation scores involve risk because each pixel element can contain peripheral information from the previous input; in other words, the localized regions may represent not only the location of the target pixel but also the surrounding locations. However, because the interaction relationship is the main factor yielding powerful performance in the CNN, we expect that we can minimize this concern if we can show good predictive performance using a 1-D kernel.

Using the discriminative localization via activation mapping with the proposed method for identifying optimal processes is an interesting direction for future work. Despite the abnormal sensor signals, the detection of optimal sensor signals that explain the normal quality can be used to identify the conditions of stable processes. We believe that the proposed method can be a cornerstone for the use of activation mapping in multiple sensor data-based root cause monitoring and is a useful tool for various manufacturing industries that require multilevel causal analysis to examine the characteristics of sequential processes.

ACKNOWLEDGMENT

The authors would like to thank the editor and reviewers for their useful comments and suggestions, which greatly helped in improving the quality of this article.

REFERENCES

[1] S. Mahadevan and S. L. Shah, "Fault detection and diagnosis in process data using one-class support vector machines," *J. Process Control*, vol. 19, no. 10, pp. 1627–1639, 2009.

[2] F. Jia, Y. Lei, L. Guo, J. Lin, and S. Xing, "A neural network constructed by deep learning technique and its application to intelligent fault diagnosis of machines," *Neurocomputing*, vol. 272, pp. 619–628, Jan. 2018.

[3] G. Weidl, A. L. Madsen, and S. Israelson, "Applications of object-oriented Bayesian networks for condition monitoring, root cause analysis and decision support on operation of complex continuous processes," *Comput. Chem. Eng.*, vol. 29, no. 9, pp. 1996–2009, 2005.

[4] C. A. Ronao and S.-B. Cho, "Human activity recognition with smartphone sensors using deep learning neural networks," *Expert Syst. Appl.*, vol. 59, pp. 235–244, Oct. 2016.

[5] Y. Lei, N. Li, L. Guo, N. Li, T. Yan, and J. Lin, "Machinery health prognostics: A systematic review from data acquisition to RUL prediction," *Mech. Syst. Signal Process.*, vol. 104, pp. 799–834, May 2018.

[6] G. Li, S. J. Qin, and T. Yuan, "Data-driven root cause diagnosis of faults in process industries," *Chemom. Intell. Lab. Syst.*, vol. 159, pp. 1–11, Dec. 2016.

[7] C.-F. Chien, C.-Y. Hsu, and P.-N. Chen, "Semiconductor fault detection and classification for yield enhancement and manufacturing intelligence," *Flex. Serv. Manuf. J.*, vol. 25, no. 3, pp. 367–388, 2013.

[8] J. M. Nawaz, M. Z. Arshad, and S. J. Hong, "Fault diagnosis in semiconductor etch equipment using Bayesian networks," *J. Semicond. Technol. Sci.*, vol. 14, no. 2, pp. 252–261, 2014.

[9] Z. Liu, Y. Liu, B. Cai, and C. Zheng, "An approach for developing diagnostic Bayesian network based on operation procedures," *Expert Syst. Appl.*, vol. 42, no. 4, pp. 1917–1926, 2015.

[10] Y. Y. Wee, W. P. Cheah, S. C. Tan, and K. Wee, "A method for root cause analysis with a Bayesian belief network and fuzzy cognitive map," *Expert Syst. Appl.*, vol. 42, no. 1, pp. 468–487, 2015.

[11] K. B. Lee, S. Cheon, and C. O. Kim, "A convolutional neural network for fault classification and diagnosis in semiconductor manufacturing processes," *IEEE Trans. Semicond. Manuf.*, vol. 30, no. 2, pp. 135–142, May 2017.

[12] M. S. Safizadeh and S. K. Latifi, "Using multi-sensor data fusion for vibration fault diagnosis of rolling element bearings by accelerometer and load cell," *Inf. Fusion*, vol. 18, pp. 1–8, Jul. 2014.

[13] M. Azamfar, J. Singh, I. Bravo-Imaz, and J. Lee, "Multisensor data fusion for gearbox fault diagnosis using 2-D convolutional neural network and motor current signature analysis," *Mech. Syst. Signal Process.*, vol. 144, Oct. 2020, Art. no. 106861.

[14] R. Assaf and A. Schumann, "Explainable deep neural networks for multivariate time series predictions," in *Proc. 28th Int. Joint Conf. Artif. Intell. (IJCAI)*, 2019, pp. 6488–6490.

[15] C. Schockaert, R. Leperlier, and A. Moawad, "Attention mechanism for multivariate time series recurrent model interpretability applied to the ironmaking industry," 2020. [Online]. Available: arXiv:2007.12617.

[16] Z. Chen and W. Li, "Multisensor feature fusion for bearing fault diagnosis using sparse autoencoder and deep belief network," *IEEE Trans. Instrum. Meas.*, vol. 66, no. 7, pp. 1693–1702, Jul. 2017.

[17] L. Jing, T. Wang, M. Zhao, and P. Wang, "An adaptive multi-sensor data fusion method based on deep convolutional neural networks for fault diagnosis of planetary gearbox," *Sensors*, vol. 17, no. 2, p. 414, 2017.

[18] D. Yang, Y. Pang, B. Zhou, and K. Li, "Fault diagnosis for energy Internet using correlation processing-based convolutional neural networks," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 49, no. 8, pp. 1739–1748, Aug. 2019.

[19] Y. Yao, S. Zhang, S. Yang, and G. Gui, "Learning attention representation with a multi-scale CNN for gear fault diagnosis under different working conditions," *Sensors*, vol. 20, no. 4, p. 1233, 2020.

[20] S. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," 2017. [Online]. Available: arXiv:1705.07874.

[21] M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why should i trust you?' Explaining the predictions of any classifier," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, 2016, pp. 1135–1144.

[22] U. Schlegel, H. Arnout, M. El-Assady, D. Oelke, and D. A. Keim, "Towards a rigorous evaluation of XAI methods on time series," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Seoul, South Korea, 2019, pp. 4197–4201, doi: 10.1109/ICCVW.2019.00516.

[23] S. Wachter, B. Mittelstadt, and C. Russell, "Counterfactual explanations without opening the black box: Automated decisions and the GDPR," *Harvard J. Law Technol.*, vol. 31, no. 2, pp. 841–887, 2018.

[24] S. Verma, J. Dickerson, and K. Hines, "Counterfactual explanations for machine learning: A review," 2020, [Online]. Available: arXiv: 2010.10596.

[25] E. Ates, B. Aksar, V. J. Leung, and A. K. Coskun, "Counterfactual explanations for machine learning on multivariate time series data," 2020. [Online]. Available: arXiv: 2008.10781.

[26] Z. Xia, S. Xia, L. Wan, and S. Cai, "Spectral regression based fault feature extraction for bearing accelerometer sensor signals," *Sensors*, vol. 12, no. 10, pp. 13694–13719, 2012.

[27] D. Borchert, D. A. Suarez-Zuluaga, P. Sagmeister, Y. E. Thomassen, and C. Herwig, "Comparison of data science workflows for root cause analysis of bioprocesses," *Bioprocess Biosyst. Eng.*, vol. 42, no. 2, pp. 245–256, 2019.

[28] H.-S. Chen, Z. Yan, Y. Yao, T.-B. Huang, and Y.-S. Wong, "Systematic procedure for granger-causality-based root cause diagnosis of chemical process faults," *Ind. Eng. Chem. Res.*, vol. 57, no. 29, pp. 9500–9512, 2018, doi: 10.1021/acs.iecr.8b00697.

[29] L. Ma, J. Dong, and K. Peng, "Root cause diagnosis of quality-related faults in industrial multimode processes using robust Gaussian mixture model and transfer entropy," *Neurocomputing*, vol. 285, pp. 60–73, Apr. 2018, doi: 10.1016/j.neucom.2018.01.028.

[30] C. Schockaert, V. Macher, and A. Schmitz, "VAE-LIME: Deep generative model based approach for local data-Driven model interpretability applied to the ironmaking industry," 2020. [Online]. Available: arXiv:2007.10256.

[31] J. Gu *et al.*, "Recent advances in convolutional neural networks," *Pattern Recognit.*, vol. 77, pp. 354–377, May 2018.
[32] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Las Vegas, NV, USA, 2016, pp. 2921–2929.
[33] K. Muhammad, J. Ahmad, Z. Lv, P. Bellavista, P. Yang, and S. W. Baik, "Efficient deep CNN-based fire detection and localization in video surveillance applications," *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 49, no. 7, pp. 1419–1434, Jul. 2019.
[34] X. Tao, D. Zhang, Z. Wang, X. Liu, H. Zhang, and D. Xu, "Detection of power line insulator defects using aerial images analyzed with convolutional neural networks," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 50, no. 4, pp. 1486–1498, Apr. 2020.
[35] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
[36] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Roy. Stat. Soc. B, Methodol.*, vol. 58, no. 1, pp. 267–288, 1996, doi: 10.1111/j.2517-6161.1996.tb02080.x.
[37] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Boston, MA, USA, 2015, pp. 3431–3440.
[38] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014. [Online]. Available: arXiv: 1409.1556.
[39] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Honolulu, HI, USA, 2017, pp. 4700–4708.
[40] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput. Assist. Intervention*, 2015, pp. 234–241.
[41] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis.*, Santiago, Chile, 2015, pp. 1520–1528.
[42] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.

Yoon Sang Cho received the B.S. degree in industrial and management engineering from the Department of Industrial and Management Engineering, Hankuk University of Foreign Studies, Seoul, South Korea, in 2017. He is currently pursuing the Ph.D. in industrial and management engineering degree with the Department of Industrial and Management Engineering, Korea University, Seoul, Republic of Korea.

His research interests include explainable artificial intelligence algorithms for sequential patterns.

Seoung Bum Kim received the M.S. degree in industrial and systems engineering and the Ph.D. degree in industrial and systems engineering from the Georgia Institute of Technology, Atlanta, GA, USA, in 2001 and 2005, respectively.

He is a Professor with the Department of Industrial and Management Engineering, Korea University, Seoul, Republic of Korea. His research interests utilize machine learning algorithms to create new methods for various problems appearing in engineering and science.