IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 10 September 2023, accepted 28 November 2023, date of publication 5 December 2023, date of current version 13 December 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3339500

RESEARCH ARTICLE

Shapelet-Based Sensor Fault Detection and Human-Centered Explanations in Industrial Control System

SUENGBUM LIM, JINGANG KIM, AND TAEJIN LEE
Department of Information Security, Hoseo University, Asan-si 31499, South Korea

Corresponding author: Taejin Lee (kinjecs0@gmail.com)

This work was supported in part by the Institute of Information & Communications Technology Planning & Evaluation (IITP) funded by the Korean Government (MSIT) through the Development of Security Monitoring Technology-Based Network Behavior Against Encrypted Cyber Threats in ICT Convergence Environment under Grant RS-2023-00235509; and in part by the Institute of Civil-Military Technology Cooperation funded by the Defense Acquisition Program Administration and Ministry of Trade, Industry, and Energy of Korean Government under Grant 21-CM-EC-07.

ABSTRACT With the development of information and communication technology, industrial control systems (ICSs) that operate in closed environments are now operating in smart environments, and external threats are increasing. To predict failure and respond to threats, anomaly detection and fault detection using artificial intelligence (AI) are being introduced, but the issue of the reliability of AI prediction is emerging. For anomaly detection, the operator must check thousands of sensors. In addition, practical operational constraints exist because AI predictions are not always accurate. This study proposes shapelet-based anomaly detection and automatic fault sensor description technology to overcome these limitations. Through intuitive abnormality detection and interpretation based on these representative patterns, when an abnormal situation occurs, operators can immediately intuitively determine which sensor causes the problem and how much the sensor differs from the pattern. This was verified with the HIL-based Augmented ICS Security Dataset (HAI) and Secure Water Treatment (SWaT) dataset, which is widely used in the ICS field. In the case of the HAI Dataset, 95.12% of the failed sensors were analyzed by extracting and inspecting only 4% of the total sensors. In the case of the SWaT Dataset, only 7% of the sensors were extracted and inspected, confirming that 84% of the failed sensors could be analyzed. We expect that intuitive explanations and anomaly detection will enable more effective technological operations in industrial environments.

INDEX TERMS Anomaly detection, efficient explanations, effective operation, fault sensor, shapelet.

I. INTRODUCTION

Industrial control systems (ICSs) monitor and control work processes, such as important national infrastructure facilities, and industrial processes, such as gas, power, water and sewage, transportation, nuclear power, and manufacturing. Initially, the ICS was an isolated system implemented using an operating system in the form of proprietary control protocols, with little resemblance to traditional information technology (IT) systems. They also used protocols developed by system manufacturers with availability as the top priority.

The associate editor coordinating the review of this manuscript and approving it for publication was Mehrdad Saif.

Because the programmable logic controller (PLC), the main element of the control system, is not connected to the network, there are almost no other threats besides those caused by physical sabotage or natural disasters. Therefore, when designing a system operating in a closed network, ICS manufacturers can operate the system without considering security. However, owing to the recent development of information and communication technology, industries that operated in a closed environment in the past are now operating in a smart environment. In a closed environment, there are almost no external threats. However, by introducing a smart environment for supervisory control and data acquisition (SCADA), ICSs, and operational technology (OT) such as factories and power plants, cyberattacks targeting industrial facilities and infrastructure that operate in such environments regularly occur [1], [2]. Systematic security technology is required to respond to and prevent such threats [3]. These security technologies have been studied for intrusion detection and failure prediction using artificial intelligence (AI) in various environments [4], [5], [6], [7], [8]. However, there are few studies on the fault analysis of predictions based on time series flow and the prediction of AI models, and the reliability is insufficient. Also, when an anomaly is detected, there is a problem in that all features must be analyzed to check it.

To solve this problem, this study extracts representative data patterns using shapelets and proposes an abnormal inquiry and incorrect data analysis method based on the representative patterns. This approach detects anomalies and supports decision-making so that field experts can make quick judgments and responses by providing evidence of the faults that have caused the abnormalities.

Contributions. This study makes the following contributions:

- We propose a method for improving the interpretation of shapelet-based detection and its interpretation in security applications. The framework consists of two main goals. Abnormal detection and interpretation from the point of view, abnormal detection and interpretation of features that cause abnormalities.
- Abnormal detection based on shapelets provides abnormal detection from a specific point of view and a detailed feature that exceeds the threshold. Based on shapelets, interpreters provide a powerful interpretation of human understanding of abnormal detection results.
- We provided abnormal detection and interpretation in two aspects to identify targets that required a quick response and inspection. It can be used to start a quick response by identifying the point in time and providing a detailed interpretation of the inspection target that causes the attack to identify the targets that need to be inspected.
- It also provides the actual value shown from an abnormal point of view, the actual value that appears at any normal point in time, and the representative pattern values. This allowed us to compare the data flow in terms of the data flow in the usual feature.

Section II introduces related work on anomaly detection, interpretation, and evaluation. Section III presents the proposed model that uses a shapelet. Section IV describes the results of the experiments using the proposed model and introduces operational examples in real environments. Section V discusses areas of improvement in the research conducted in this study. Section VI introduces the research results shown in this paper and their contributions.

II. RELATED WORK

Anomaly detection has been widely used in various fields [9], [10]. Anomaly detection identifies outliers that do not follow a normal pattern in large datasets. This section introduces several research cases for detecting anomalies in a time series, studies on how to improve the performance of these anomaly detections, how to interpret the detected anomalies, and how to evaluate the results of the interpretation.

Among the various methods introduced in this section, shapelets are used in this study. The reason for using shapelets is to detect anomalies in time-series data. In addition, it was determined to be advantageous for intuitive interpretation and quantification using the distance mechanism. Therefore, this study aims to enable experts to quickly recognize problems through intuitive visualization and quantification and respond to causes with little effort.

A. STUDY ON ANOMALY DETECTION IN VARIOUS ENVIRONMENTS

1) ANOMALY DETECTION IN ENVIRONMENTS UTILIZING MULTIPLE SENSORS

Owing to recent technological developments, environments utilizing multiple sensors are increasing. Therefore, anomaly detection using multiple sensors is necessary. Canizo et al. proposed a deep learning-based approach for supervised multi-time series anomaly detection that combines a Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) in different ways [11]. Unlike other approaches, this approach uses independent CNNs to perform anomaly detection in multisensor systems. They experimented with a real industrial scenario, in which anomalies were effectively detected on a service elevator based on multiple sensor data. The features from each sensor data are extracted completely independently using a multi-head CNN. Accordingly, heterogeneous data could be processed.

2) ANOMALY DETECTION IN MEDICAL ENVIRONMENTS

The need for anomaly detection is also increasing in environments that require the identification of unusual points, such as the medical or security industries. Liu et al. proposed the arrhythmia classification of an Long Short-Term Memory (LSTM) autoencoder based on time-series anomaly detection [12]. This study highlights the need for anomaly detection in this environment. They used five different types of ECG data from the MIT-BIH arrhythmia and MIT-BIH supraventricular arrhythmia databases: atrial premature beats (APB), left bundle branch block (LBBB), normal heartbeat (NSR), right bundle branch block (RBBB) and ventricular premature beats (PVC).

A model based on the LSTM autoencoder was created for each dataset, and comprehensive classification was performed for the input data. In this way, there is also a way to create multiple models for each important piece of data and perform comprehensive anomaly detection.

B. STUDY ON PERFORMANCE IMPROVEMENT OF TIME SERIES MODELS

1) FULLY CONVOLUTIONAL NETWORKS (FCNS)

An FCN is a variant of existing CNN-based models (such as Visual Geometry Group 16) for semantic segmentation models.

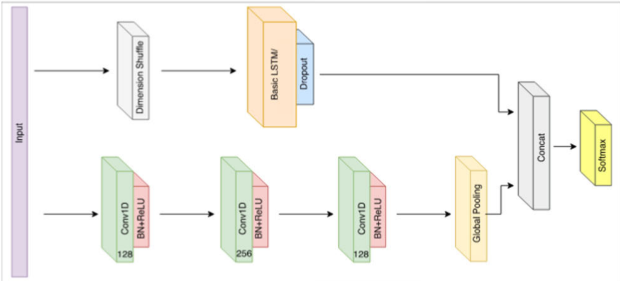


FIGURE 1. Example of the FCN & FCN+FCN model architecture [14].

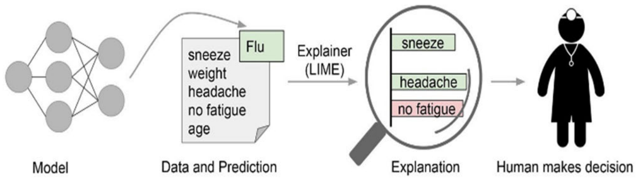


FIGURE 2. LIME example for cause judged by the model as a result of the input value of patient [18].

FIGURE 3. SHAP example of feature contribution in classifier model [21].

The fully connected layer (FCL) architecture has three limitations: there are points where the number of parameters is too large, the location information of the image feature disappears, or the size of the input image is fixed. The FCN model replaces all the FCLs with convolutional layers to compensate for these problems. Karim et al. proposed a novel LSTM + FCN model that combines an FCN with an existing long short-term memory (LSTM) model. Through the FCN process, the convolutional layer and global pooling, LSTM dropout, concatenation, and SoftMax classification are performed to create a model. Fig. 1 shows the structures of the FCN and LSTM+FCN models [13], [14].

2) ATTENTION+LSTM MODEL

Hao et al. proposed a new model in which CA-SFCN, compared to GA (Global Attention)-SFCN, RA (Recurrent Attention)-SFCN, and SFCN, achieved high performance in classification using mostly time series data in 14 datasets. This model uses the CA-SFCN (cross-caution) for multivariate time-series classification. We reuse the output of the last convolutional layer of the FCN to measure the attention scores for the entire time series (past–present) and then proceed with matrix addition between the extracted score values. In other words, the goal was to improve the model's performance by measuring attention multiple times at a full-time point. On average, using attention yields a higher performance [15], [16], [17].

C. STUDY ON EXPLAINABLE ARTIFICIAL INTELLIGENCE (XAI)

1) LOCAL INTERPRETABLE MODEL-AGNOSTIC EXPLANATION (LIME)

LIME was proposed by Marco Tulio Ribeiro in 2016 to address two confidence problems: trusting a prediction of individual values, and trusting a model. The description of the individual predictions identifies which model presents the results and which input influences them.

When the model for predicting influenza concludes that the patient (input value) has the flu (result value), LIME weighs the input value and informs the conclusion that the patient has the flu [18]. Fig. 2 shows an explanation of the individual predictions. The operating principle of LIME is to generates random data around the input data by partially modifying the value of the input data and then using it to train the surrogate model. Equation (1) yields the following formula:

$$explanation\,(x) = argmin_{g\in G} L\,(f, g, \pi_x) + \Omega\,(g) \quad (1)$$

In the formula, f is the black box model to be explained, and the explanation of the input value x selects the model g whose function L has the minimum value from among the set of explanatory models. G is the complexity of Model g.

2) SHAPLEY ADDITIVE EXPLANATIONS (SHAP)

The SHAP was first proposed by Shapley in 1953 [19]. It is a solution game theory that computes a model's contribution to the subset prediction of all data features using m features [20]. SHAP creates a dataset that adds and removes features, is composed of a linear model, and measures how much the prediction changes when a specific variable is removed by analyzing the weights of the linear model constructed in this manner. Fig. 3 shows an example of SHAP for a model classifying obesity and normal weight.

The classifier classified in this manner has a positive(negative) shapley value if it contributes to determining each feature as abnormal(normal) [21]. The Z_i' value in Equation (2) indicates whether the i-th feature occurs, whereas Φ_i is the contribution value of the i-th feature.

$$g\left(z'\right) = \Phi_0 + \sum\nolimits_{i=1}^{M} \Phi_i {z'}_i \quad (2)$$

In Equation (3), F is the number of input features. The difference between the model output value f () in all possible cases when attribute i is included in input data x and all possible cases when i is not is calculated and used as the contribution Φ_i of the i -th feature.

$$\phi_i = \sum\nolimits_{S\subseteq F\{i\}} \frac{|S|!\,(|F| - |S| - 1)!}{|F|!} \left[f_{S\cup\{i\}}\left(x_{S\cup\{i\}}\right) - f_S\,(x_S)\right] \quad (3)$$

3) SHAPELETS

Ye and Keogh first proposed shapelets in 2009. A shapelet explores all subsequences (partial time series) present in

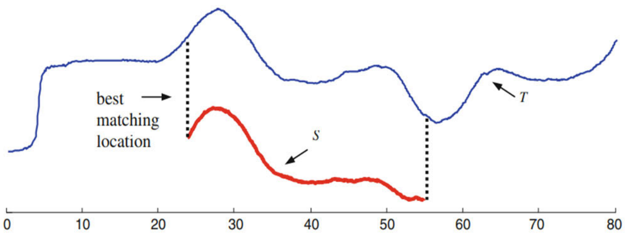


FIGURE 4. Example of applying Euclidean distance-based algorithm to shapelet and dataset [22].

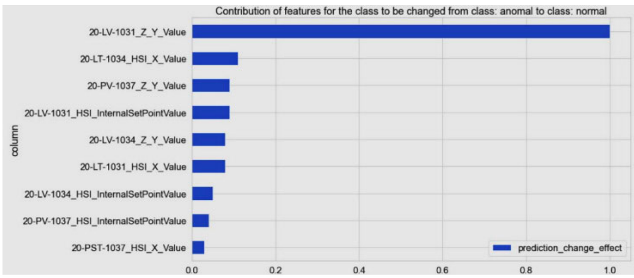


FIGURE 5. CLE: Feature importance plot [25].

the dataset. It extracts a partial time series as a representative pattern in which the dataset and distance belonging to each class significantly improve the model performance [22].

Several methods have been proposed to calculate the distance between extracted representative patterns and datasets. Euclidian distance–based similarity measurements explore the section where the extracted representative patterns and datasets are mapped 1:1 in a manner that maps sequences with the most similar values [23], [24]. Fig. 4 presents an example of applying Euclidean distance calculation to a shapelet.

4) CUSTOM LOCAL EXPLAINER (CLE)

The approach is to perturb the data points of the transformed anomalous window for several iterations and check the new perturbed or permuted time-series window against the original anomaly detection model for the prediction outcome [25]. This approach detects the normal points in the case of a maximum prediction drop from an anomalous window and observes and analyzes the features contributing to such a change. The feature importance chart in Fig. 5 was prepared to identify the feature that contributed the most to normalizing the anomalous window.

5) SIMILARITYEXPLAINER (SIMEX)

SimEx aims to compare the anomalous window with all normal training windows and find the most similar match [25]. After matching similar data, a comparison with the feature level was performed to determine the difference from the similar data. The least similar features were identified as probable faults that caused the anomaly. The plot in Fig. 6 is a line chart that compares the features of the abnormal window (in red) and similar-looking example window (in blue).

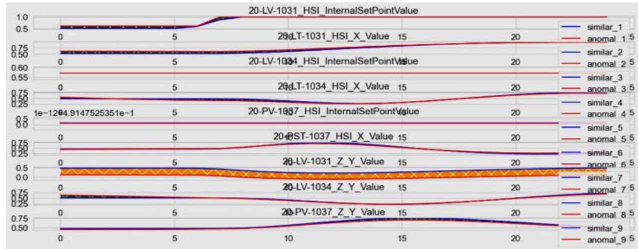


FIGURE 6. SimEx: Signal comparison plot [25].

D. STUDY ON EVALUATION OF XAI

1) ACCURACY-BASED XAI EVALUATION

Descriptive accuracy (DA) reflects the accuracy of the relevant features of the prediction. Because it is difficult to evaluate the relationship between features and predictions directly, we measure how different the predictions of neural networks will be if highly relevant features are removed through indirectly less accurate figures. Removing related features from sample data results in less information for neural networks to make accurate predictions, and consequently, faster accuracy drops. Therefore, an explanation method with a sharp decline in technical accuracy provides a better explanation than a progressively decreasing method [26]. Equation (4) provides the DA calculation formula:

$$DA_k\left(x, f_N\right) = f_N(x|x_1 = 0, \ldots, x_k = 0)_c \tag{4}$$

2) SPARSITY-BASED XAI EVALUATION

Descriptive sparsity is evaluated as a prerequisite for a case in which a good explanation assigns high relevance to a feature that influences the prediction. It was calculated using the importance value determined by XAI and scaled to the same size for comparison. Subsequently, a mass around zero (MAZ) was calculated by dividing the importance value sum by the importance value of each feature. The value is then displayed by accumulating from the first importance value. A sparse interpretation has a sharp rise close to zero, a reasonable interpretation is flat and close to one, and various other interpretations show a smaller slope and a more extensive set of features relative to zero. Therefore, a method in which the MAZ distribution peaks at 0 is better [27]. Equation (5) provides the MAZ calculation formula:

$$MAZ\left(r\right) = \int_{-r}^{r} h\left(x\right) dx \; for \; r \in [0, 1] \tag{5}$$

3) CUMULATIVE DISTRIBUTION FUNCTION (CDF) BASED XAI EVALUATION

To evaluate the reliability of the judgment of the AI model, authentication based on the CDF was performed. Let the samples of the model inference property values $\alpha \in [0, \infty)$ come from the distribution PA. The CDF was defined for the probability measure PA using Equation (6) [28].

$$CDF(\alpha) = \int_{0}^{\alpha} dP_A \tag{6}$$

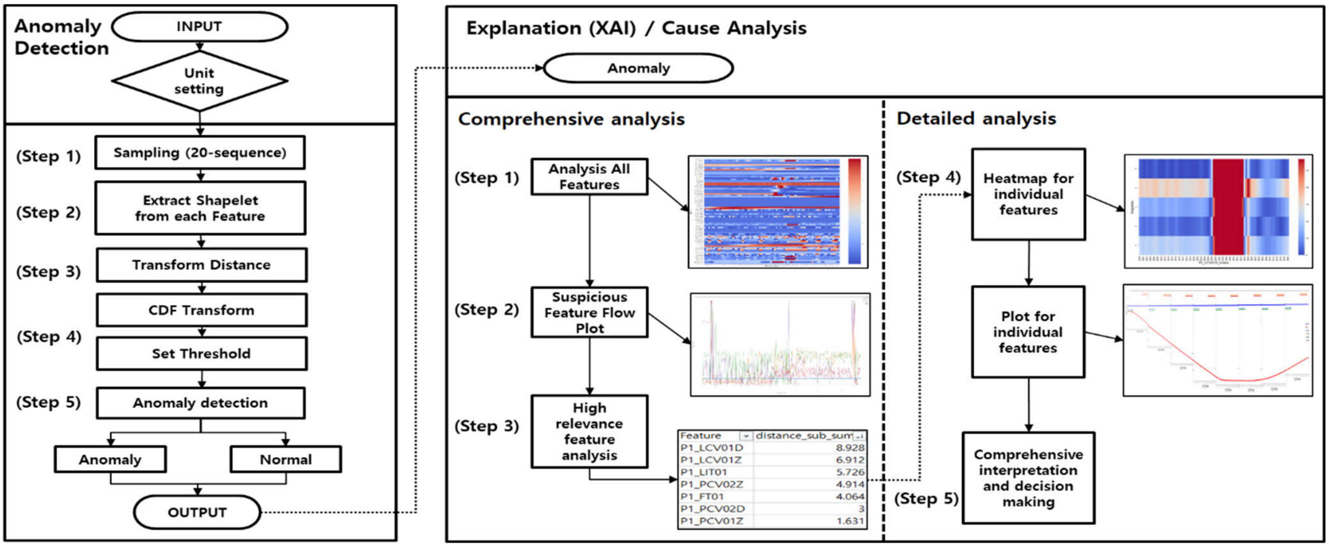


FIGURE 7. Proposed model architecture (shapelet-based anomaly detection/fault data analysis).

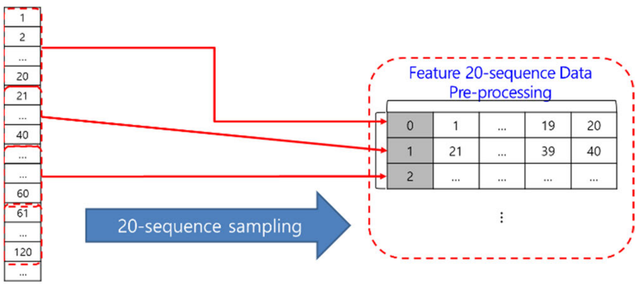


FIGURE 8. Data pre-processing according to set sequence.

III. PROPOSED MODEL

The model for anomaly detection and data analysis proposed in this study is shown in Fig. 7. The main steps of the proposed model are anomaly detection and cause analysis. To detect anomalies, normal representative patterns for each sensor were calculated and similarity was measured. Subsequently, an arbitrary threshold was set to detect the anomalies. To interpret the cause, numerical values and visualizations are made through information on the representative pattern, abnormal time point, and normal time point of the cause sensor. These data allow experts to respond immediately and make appropriate decisions.

A. DATA PRE-PROCESSING

To convert multi-dimensional time-series data to 1-dimensional time series data, separate data by each attribute. Data preprocessing was performed according to the sequence size set to extract the representative pattern for each separated feature. If is set to 20 sequences, the data are cut at intervals of 20 s, and shapelets are extracted. Fig. 8 shows an example of data preprocessing according to the set sequence. This sample was pre-processed using a sequence of 20. The extracted representative patterns differed depending on the size of the set sequence. If the sequence size is too small compared with the attack duration, detecting anomalies with a representative pattern is difficult.

Algorithm 1 GENDIS(T, y, pop_size, max_gen, patience, $p_{mutation}$, $p_{crossover}$, max_len) [29]

*Population = initialize_population(**T**, pop_size, max_len)*
current_gen, best_gen, best_fitness = 0, 0, 0

while current_gen < max_gen **and** current_gen – best_gen < patience:
  for(child1, child2) **in** *zip*(population[::2], population[1::2]):
    if *random*() $P_{crossover}$:
      population.*append*(*crossover*(child1, child2))
    if *random*() $P_{mutation}$:
    population.*append*(*mutate*(child1, child2))

  fittest = *select_fitteset*(population)
  population = *tournament_selection*(population, pop_size)
  population.*append*(fittest)
if *fitness*(**T**, **y**, fittest) > best_fitness:
  best_fitness = *fitness*(**T**, **y**, fittest)
  best_gen = current_gen
currnet_gen +=1

B. SHAPELET EXTRACTION

The GENDIS algorithm is used to extract the shapelet Algorithm 1 and presents the GENDIS algorithm, which uses a random extraction method [29]. A representative pattern similar to the original pattern was extracted for each feature by repeating a random value in length within the set sequence. The number, length, and value of the shapelets extracted for each feature are different.

C. CONVERTING DATA

The similarity between the extracted shapelet for each feature and the original feature data was measured using an improved Euclidean distance-based algorithm. The improved Euclidean formula for calculating the similarity d between

index	Feature	distance	value	threshold
108	P1_PCV02Z	0.644	0.889	0.245
108	P1_PCV01Z	0.016	0.389	0.373
108	P1_PIT01	0.398	0.997	0.599
index	Feature	distance	value	threshold
109	P1_PCV01Z	0.079	0.452	0.373
109	P1_FT01	0.127	0.815	0.688
109	P1_PIT01	0.401	1	0.599
109	P1_FT01Z	0.016	1	0.984
109	P1_PCV02Z	0.755	1	0.245
index	Feature	distance	value	threshold
110	P1_FT01	0.117	0.805	0.688
110	P1_LCV01D	0.108	0.418	0.31
110	P1_PCV02Z	0.755	1	0.245
110	P1_PIT01	0.401	1	0.599
110	P1_PCV01Z	0.099	0.472	0.373
110	P1_FT01Z	0.016	1	0.984

index	Feature	distance	value	threshold
111	P1_LCV01Z	0.121	0.563	0.442
111	P1_LCV01D	0.513	0.823	0.31
111	P1_PIT01	0.401	1	0.599
index	Feature	distance	value	threshold
112	P1_LCV01Z	0.295	0.737	0.442
112	P1_LCV01D	0.622	0.932	0.31
112	P1_PIT01	0.401	1	0.599
index	Feature	distance	value	threshold
113	P1_LCV01D	0.538	0.848	0.31
113	P1_LCV01Z	0.256	0.698	0.442
113	P1_FT01	0.307	0.995	0.688
113	P1_PIT01	0.401	1	0.599
index	Feature	distance	value	threshold
118	P1_FT01Z	0.013	0.997	0.984
118	P1_PIT01	0.401	1	0.599
118	P1_FT01	0.281	0.969	0.688

FIGURE 9. Example of anomaly detection/attack range search algorithm application.

the original data and shapelet is shown in Equation (7).

$$D = \sqrt{\sum_{i=1}^{n} (a_i - b_i)^2 \div len(s)} \tag{7}$$

a is the original data value, b is the shapelet value, and $len(s)$ is the shapelet length. A smaller calculated value is more similar to the shapelet, whereas a larger value is less similar. A value close to the normal representative pattern can be considered normal at a specific time point. By contrast, a value close to an abnormal pattern can be determined as abnormal.

The CDF was applied to each data value to detect anomalies. It also performs point-in-time integration analysis on distance data (similarity data converted per feature) with CDF applied. The original and test data were applied using the average value and standard deviation of the original distance data converted from the original data value. An integrated analysis can be performed because the minimum value of the applied data is fixed at 0 and the maximum value at 1. Most converted CDF values are distributed around 0.5 when normal; those closer to 1 are farther away from the normal value. The original distance data, the value to which CDF is applied to the original data, are sorted in descending order, and the top 1% is set to the 10% value as a threshold. An image can be judged as abnormal if it exceeds the corresponding value.

D. SHAPELET-BASED ANOMALY DETECTION AND ANOMALY RANGE SETTING

For the test data to which CDF is applied, a value greater than the threshold set for each feature is judged to be abnormal.

If the time points determined to be abnormal were continuous, they were set as abnormal periods. It is set as an abnormal section to analyze the fault data for a section that is determined to be abnormal. Because 20 sequences are converted into one unit for the original data, if the index that appears as an anomaly is multiplied by 20, it is also known that the original time is abnormal. Algorithm 2 proposes a method to set the attack range for the detected anomalies. Fig. 9 shows an example of the application of this algorithm.

If the result calculated using Algorithm 2 appears at consecutive points in time, as in the example in Fig. 9, it is regarded as the same attack. In addition, the table on the right of the figure provides information on the features that contribute to the anomaly by index. The red and blue values represent the CDF and threshold values of the feature, respectively. The difference between the two values is expressed as the distance; the larger the distance value, the higher the value contributing to the anomaly.

Algorithm 2 Anomaly Detection/Attack Range Search

SET Anomaly_score: Input Test Data Anomaly score
*SET Threshold: Input Train Data Anomaly score * ratio*
SET Attack_list: Save Attack Lists
SET start, end: Attack Start Point, Attack End Point
SET Continue: Continuous Index setting

```
Attack_list = []
for k in range(len(Anomaly_score)):
  if Anomlay_score[k] > Threshold:
      Attack_list.append(k)      # Attack List append
start, end = []
count, round = 0
for k in range(len(Attack_list)):
  if count == round:
    if count == 0:               # Set First Attack
       start.insert(count, Attack_list[k])
       end.append(Attack_list[k])
       round + = 1
    else:                        # Set attack after the First Attack
       start.insert(count, Attack_list[k-1])
       end.insert(count, Attack_list[k])
  if abs(start[count] – Attack_list[k]) > Continue:
     end[k] = Attack_list[k-1]
     count + = 1
     round = count

Return start, end
```

E. SHAPELET-BASED ANOMALY/FAULT SENSOR IDENTIFICATION AND INDIVIDUAL INTERPRETATION (XAI)

The abnormal time points calculated in this section were visualized for an integrated analysis. Algorithm 3 proposes a method for visualizing all features for an integrated analysis. Fig. 10 shows an example of the application of the algorithm. The X-axis represents the set time index, while the y-axis represents individual features. The red data that

Algorithm 3 XAI All Features

SET Feature_List: Input Features List
SET Feature_Threshold: Individual Thresholds for Input Features
SET Time_Threshold: Time Index Threshold
SET start, end: Attack Start Point, Attack End Point

```
1. anomlay_time_index = []
2. score_li = []
3. Time_score_li = []
4. score_df = pd.DataFrame()
5. for index in range(start[k], 1, end[k]):
6.    for k in range(len(Feature_List)):
7.       score = anomaly_score[Feature_List[index]].loc[index].min()
8.       score_li.append(score)
9.       if score > Feature_Threshold[Feature_List[index]]:
10.          acc_score + = score
11.      temp_df = pd.DataFrame(score_li)
12.      score_df = pd.concat(score_df, temp_df, axis=1)
13.   if acc_score > Time_Threshold:
14.      anomaly_time_index.append(index)
15.      Time_score_li.append(acc_score)
16. plt.subplots(figsize=(100,50))
17. ax = sns.heatmap(score_df.T, cmap='coolwarm', vmin=0, vmax=
Feature_Threshold.max())
18. Return Time_score_li, anomaly_time_index, score_df
```

appear when there is a significant difference from the normal representative pattern appear continuously in the indicated red box. If the similarity value for a feature is close to normal, it appears in blue; if it is far from normal, it appears in red.

For an integrated analysis, the value that minimizes the distance between the feature value and shapelet at the corresponding point in time for each feature is calculated. Suppose that the calculated minimum value is greater than the threshold value of the corresponding feature. In this case, it is selected as an abnormal feature and the excess value is added to the cumulative abnormal value.

If the attack section is visualized with a heatmap for each calculated minimum value, it can be observed that the attack section shows a larger value than the normal section.

The degree of the anomaly was checked at the time point by visualizing the outlier values accumulated and summed from the individual outliers. The abnormal features calculated through this process are visualized as targets to support decision making. Algorithm 4 proposes a method to visualize the previously computed features to yield specific features that contribute to the anomaly. Fig. 11 shows an example of calculating individual heatmaps for specific features contributing to the anomaly by applying Algorithm 4. These detailed visualizations allowed us to judge the anomaly contributions of specific features. In the case of the normal state on the left, the actual data value (blue) appears to be similar to the representative pattern (other colors). In the case of an abnormal state on the right, the actual data value (red) shows a large difference from the representative pattern (other colors). Fig. 12 shows an example of calculating individual for specific features contributing to the anomaly by applying

Algorithm 4 XAI Specific Features

SET Anomaly_score: Input Test Data Anomaly score
SET start, end: Attack Start Point, Attack End Point
SET Anomaly_Feature: Extracted Anomaly Feature List
SET individual_score: The distance score of each Feature from each shapelet

```
1. fig, ax = plt.subplots(figsize=(15,10))
2. for k in range(len(Anomaly_Feature)): # Anomaly Feature Flow
3.    ax.plot(anomaly_score[Anomaly_Feature[k]],
label='Anomaly_Feature[k]')
4. plt.show()
5. for k in range(len(Anomaly_Feature)):
# Individual score heatmap by Feature
6.    ax = sns.heatmap(individual_score[Anomaly_Feature[k],
cmap='coolwarm', vmin=0, vmax= Feature_Threshold.max())
7. for k in range(len(Anomaly_Feature)):
# Distance from Shapelets by Feature
8.    ax.plot(anomaly_score[Anomaly_Feature[k]],
label='Anomaly_Feature[k]')
9.    shapelet_df = pd.read_csv('shapelet_df_'+str(Anomaly_Feature[k]))
10.      for i in range(len(shapelet_df)):
11.         ax.plot(shapelet_df[i], label='Anomaly_Feature[k]')
```

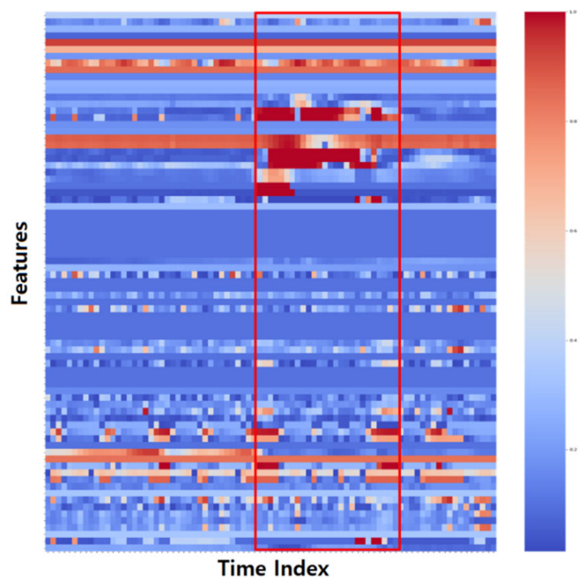


FIGURE 10. Example of XAI all-features algorithm application. This is an all-feature visualization for the period, including attack #7 in the HAI dataset.

Algorithm 4. In the example, blue flow indicates normal data and red indicates abnormal data. The remaining colors represent the normal representative patterns. The x-axis of the visualization is the set sequence size and the y-axis represents the actual data value. Therefore, if the value difference from the normal representative pattern is large, it can be judged that the feature is abnormal.

Moreover, it is possible to check the flow through which an abnormality occurs. In the case of normality, it can be confirmed that the data are similar to a normal representative pattern. However, in the case of an abnormality, it can be confirmed that it is not similar to the normal representative

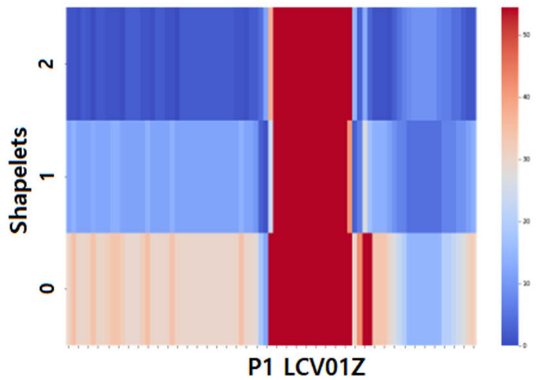


FIGURE 11. Examples of individual heatmaps for a specific feature This is an example of individual visualization for the feature "P1_LCV01Z" calculated as an anomaly at the time of attack#7 in the HAI dataset.

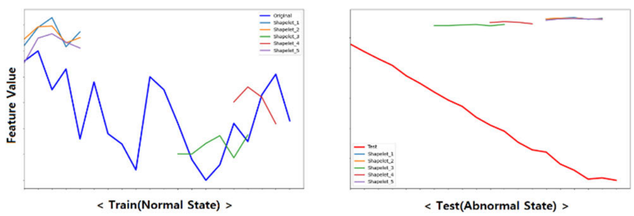


FIGURE 12. Examples of individual flows for the specific feature.

TABLE 1. HAI 2.0 dataset features by process.

Dataset	P1	P2	P3	P4	Attack / Total
HAI 21.03 (2.0)	38	22	7	12	50/79

pattern. Visualizations such as those shown in Figs. 11 and 12 can provide a detailed analysis and reliability of individual features.

IV. EXPERIMENT

Training and testing were performed on an Intel Xeon Gold 6226 2.7G server (128 GB of RAM) using an NVIDIA 16 GB Tesla T4 GPU. The development environment used the Python 3 programming language in the Anaconda 3 Jupyter Notebook.

A. DATASET

The experiment was conducted using two datasets. We used HAI 2.0 and SWaT.

The HAI dataset was collected from a realistic ICS testbed augmented with a Hardware-In-the-Loop (HIL) simulator that emulates steam-turbine power generation and pumped-storage hydropower generation [30], [31].

HAI 21.03 satisfies time continuity and contains 84 columns. The first column represents the observed time, and the next 79 columns provide the recorded SCADA data points. The last four columns provided data labels for the occurrence of an attack. Table 1 lists the numbers of features and attacks for each process. It consisted of four processes and 79 recorded SCADA data points. The structures of the

TABLE 2. HAI 2.0 dataset composition.

	Files	Interval(hours)	Size (MB)	
HAI 21.03 (2.0) Train (Normal Dataset)	train1.csv	60	110	
	train2.csv	63	116	
	train3.csv	229	245	
	SUM	**352**	**471**	
	Files	**Attack Counts**	Interval (hours)	Size (MB)
HAI 21.03 (2.0) Test (Abnormal Dataset)	test1.csv	**5**	12	22
	test2.csv	**20**	33	61
	test3.csv	**8**	30	55
	test4.csv	**5**	11	20
	test5.csv	**12**	26	47
	SUM	**50**	**112**	**205**

TABLE 3. SWaT dataset features by process.

Dataset	P1	P2	P3	P4	P5	P6	Attack / Total
SWaT	5	11	9	9	13	4	36/51

TABLE 4. SWaT dataset composition.

	Files	Interval(hours)	Size (MB)	
SWaT Train (Normal Dataset)	train0.csv	138	142	
	train1.csv	137.5	145	
	SUM	**275.5**	**287**	
SWaT Test (Abnormal Dataset)	Files	**Attack Counts**	Interval (hours)	Size (MB)
	test0.csv	**36**	125	130
	SUM	**36**	**125**	**130**

training data and test data are shown in Table 2. The training data consisted of three files, and the test data consisted of five files. The training data consisted of all normal data, and the test data contained 50 attacks, as listed in Table 1.

Secure Water Treatment (SWaT) is a water treatment testbed for research cyber security. This dataset targets the protection of Cyber-Physical Systems (CPS) such as those for water treatment, power generation and distribution, and oil and natural gas refinement [33].

SWaT satisfies time continuity and contains 53 columns. The first column represents the observed time, and the next 51 columns provide the recorded SCADA data points. The last columns provide data labels for whether an attack occurred. Table 3 lists the numbers of features and attacks for each process. It consisted of six processes and 51 recorded SCADA data points. The structures of the training data and test data are shown in Table 4. The training data consisted of two files, and the test data consisted of a total of one file. The training data consisted of all normal data, and the test data contained 36 attacks, as listed in Table 3.

B. DATA PRE-PROCESSING

In the case of a short attack time in the pre-processing data stage, detection was impossible when the sequence length was increased. The sequence used in this experiment was

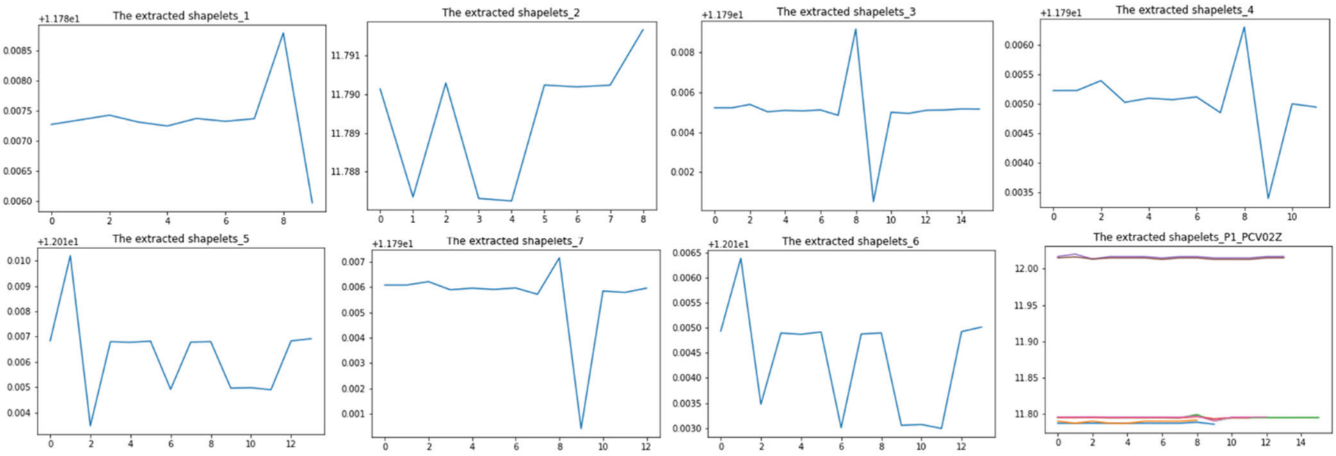


FIGURE 13. Specific feature "P1_PCV02Z" shapelets plot in the HAI dataset.

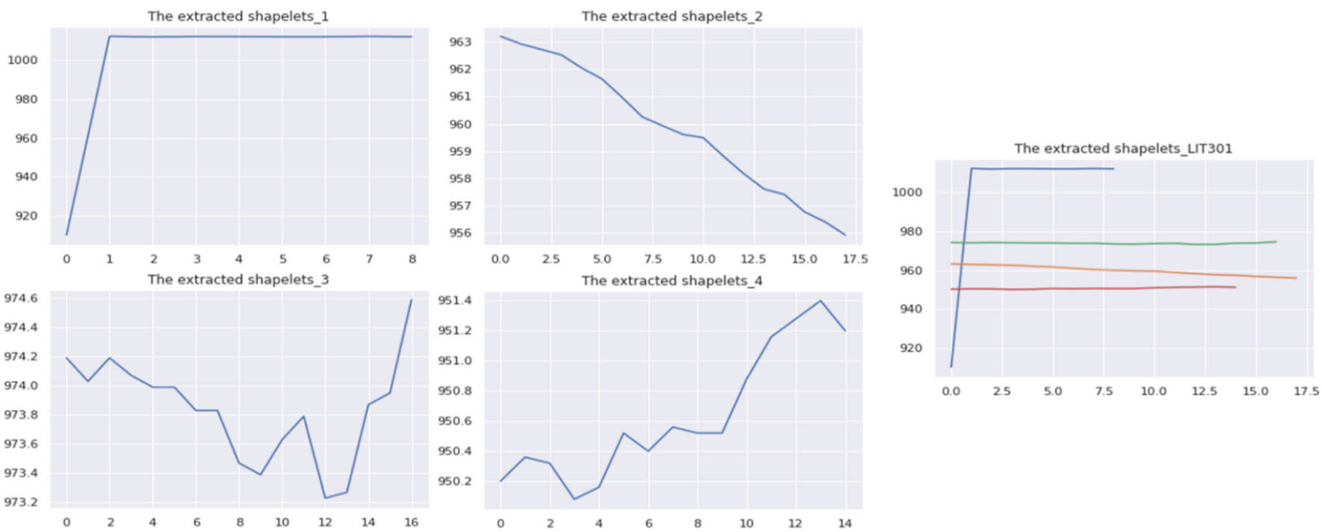


FIGURE 14. Specific feature "LIT301" shapelets plot in the SWaT dataset.

tested by setting it to 20, considering the attack time of data. In the HAI, each feature's data were pre-processed with 20 sequences and composed of 46,079 indexes. In SWaT, each feature's data were pre-processed with 20 sequences and composed of 49,590 indexes.

C. SHAPELET EXTRACTION

Shapelets were extracted for 20 sequences from 46,079 and 49,590 indexes for each feature data item. For each feature, the number of extracted shapelets, the length of the shapelet, and the shapelet value were extracted differently. Because the original data were all normal, the extracted representative pattern was a shapelet in the normal state. Fig. 13 and 14 show an example of a specific feature. Fig. 13 is the "P1_PCV02Z" feature in the HAI dataset, and Fig. 14 is the "LIT301" feature in the SWaT dataset.

In the case of "P1_PCV02Z", a total of seven shapelets were extracted, and the value in the normal range was calculated to be about 11.8 to 12.2. In the case of "LIT301", four shapelets were extracted, and the value in the normal range was calculated to be about 910 to 1,010. If the similarity between the corresponding normal representative pattern and the test data was measured to be different from the normal pattern, it could be judged as abnormal.

D. CONVERTING DATA

Distances were measured using the improved Euclidean algorithm to measure the similarity between the extracted shapelets, training data, and test data. Because all the training data were normal, almost all the data appeared close to at least one shapelet. In other words, data far from all shapelets can be considered abnormal.

The mean and standard deviation of the data of each training distance feature were extracted. The CDF was applied to the training distance feature data and the test distance using the extracted mean and standard deviation. The train distance

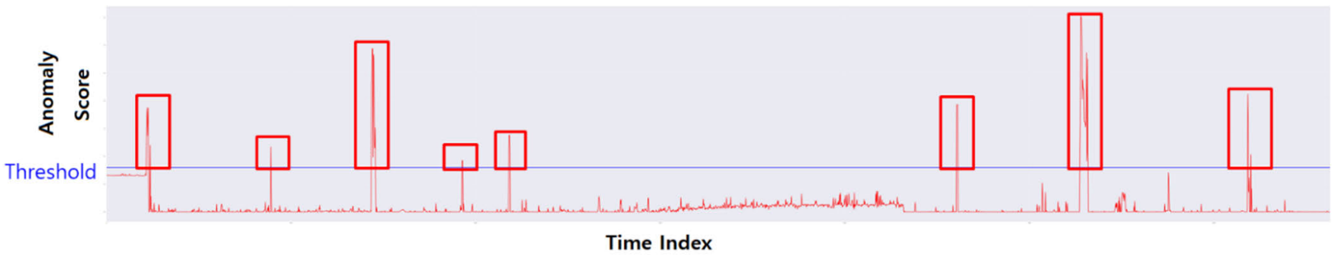


FIGURE 15. Anomaly score plot including attacks #1 to #10 in the HAI dataset. After arranging the calculated train data anomaly scores in descending order, the "Threshold" indicated by the blue line was set to the top 5%. Of the ten attacks, eight attacks indicated by red boxes were detected.

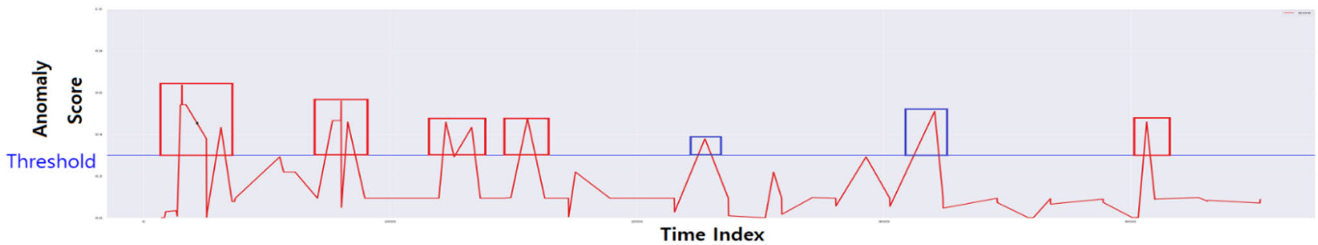


FIGURE 16. Anomaly score plot including attacks #1 to #10 in the SWaT dataset. After arranging the calculated train data anomaly scores in descending order, the "Threshold" indicated by the blue line was set to the top 10%. Of the ten attacks, red boxes indicated detected eight attacks, and blue boxes indicated two false alarms.

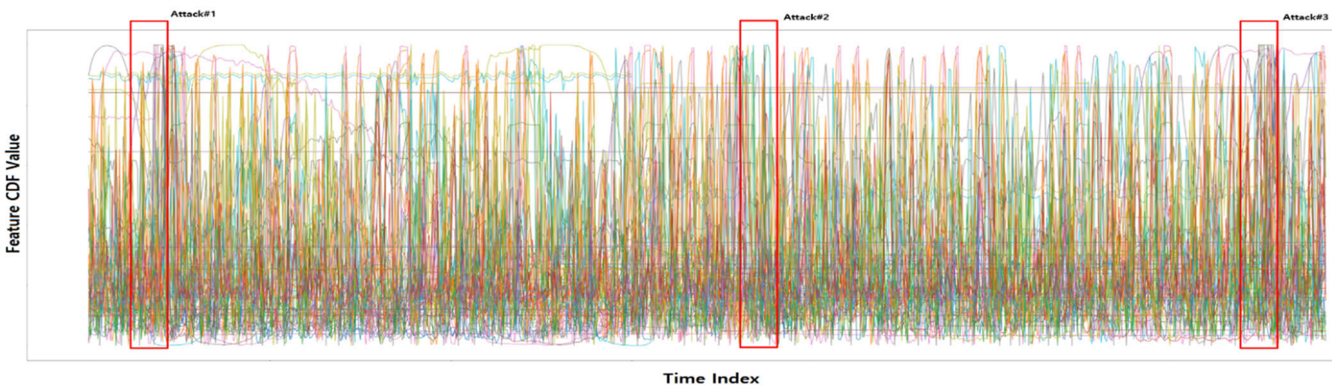


FIGURE 17. Visualization of all features for attack sections #1 to #3 in the HAI dataset. The section marked with red boxes are Attack 1, Attack 2, and Attack 3, respectively. Displaying all 79 features makes it difficult to determine whether or not there is an anomaly intuitively.

features CDF values were sorted in descending order, and because they were all normal data, the top 1–10% value can be set as the threshold value.

With the detailed percentage setting, the threshold can be set according to the distribution of data that is different from the normal in the training data. In the HAI dataset, 5% was set as the threshold because the distribution of data different from the normal was small, and in the case of the SWaT dataset, 10% was set as the threshold because the distribution of data different from the normal was greater than that in the HAI dataset. If the test distance feature value was greater than the threshold value, it was considered to be abnormal.

E. ANOMALY/FAULT DETECTION

This section discusses the identification of the abnormal time point, setting of the abnormal section, and identification of the sensor to be analyzed for XAI.

1) SETTING OF TIME INDEX THRESHOLD

The time index threshold was set similar to the individual feature threshold settings. For the training distance data to which CDF was applied, the time index summed the minimum distance between 79 features and shapelets in the HAI dataset and the minimum distances between 51 features and shapelets in the SWaT dataset. The summed values were sorted in descending order, and the top 5% values were set as the threshold for the HAI dataset and the top 10% values for the SWaT dataset were set as the threshold.

2) ANOMALY DETECTION

For each point in the test data to which the CDF was applied, the HAI dataset cumulatively summed 79 individual feature values and the values with the smallest distance from the feature shapelet for values exceeding the threshold of each feature. Similarly, the SWaT dataset was cumulatively

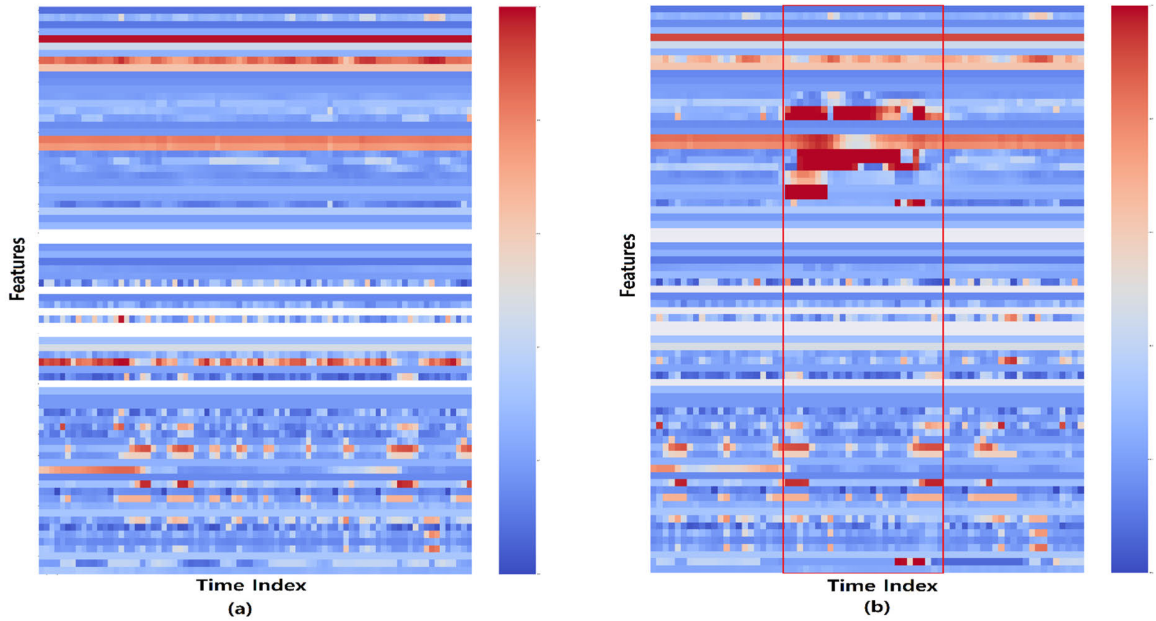


FIGURE 18. Heatmap visualization of all features in the HAI dataset. (a) Heatmap visualization for all features for an arbitrary normal-state section. (b) Heatmap visualization for all features targeting the section containing attack 1. The X-axis is the time index, and the Y-axis is 79 features. As in section III, E) Fig. 10, the anomaly is displayed in red.

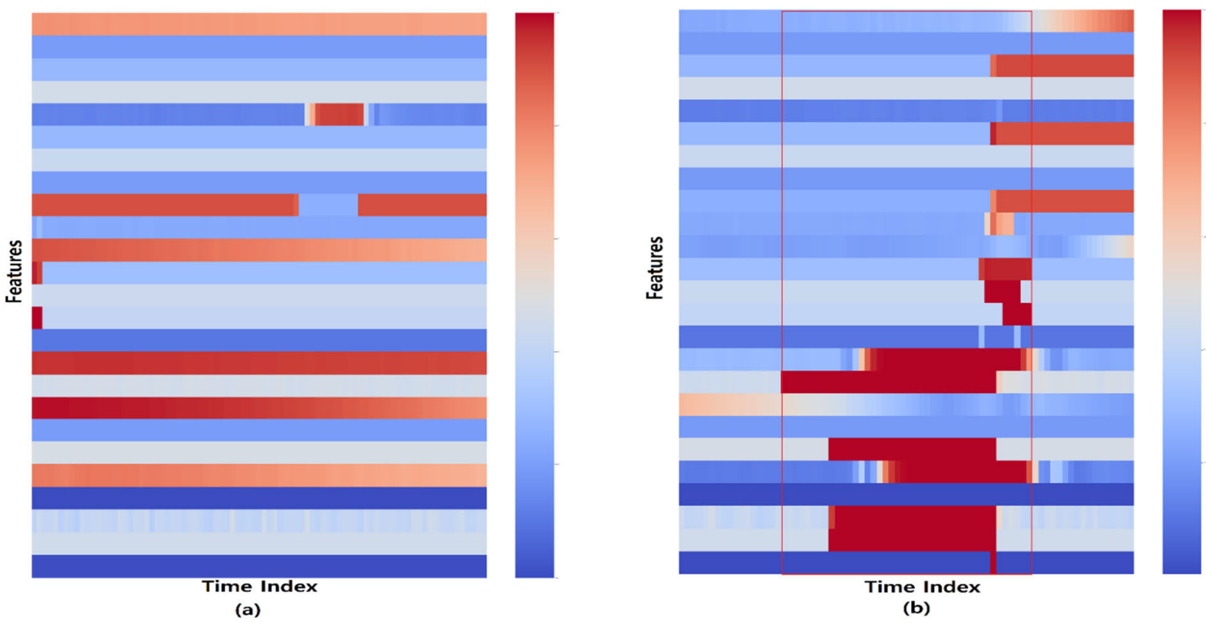


FIGURE 19. Heatmap visualization of all features in the SWaT dataset. (a) Heatmap visualization for all features for an arbitrary normal-state section. (b) Heatmap visualization for all features targeting the section containing attack #8. The X-axis is the time index, and the Y-axis is 51 features. As in section III, E) Fig. 10, the anomaly is displayed in red.

summed up for 51 features. A value higher than the set time index threshold was considered an abnormal time point. Fig. 15 and 16 show examples of calculating the anomaly score for the test data, including attack sections #1 to #10 of the HAI and SWaT datasets.

3) SETTING OF ATTACK SECTION

If abnormal points were consecutive, they were judged as one attack section and set as the attack section to be interpreted. The features to be analyzed individually in the set attack section were calculated. The features contributing to the attack were calculated using an anomaly value exceeding the threshold value of each feature.

F. SHAPELET-BASED ANOMALY FAULT SENSOR: IDENTIFICATION AND INDIVIDUAL INTERPRETATION (XAI)

This section discusses the abnormal time point XAI, abnormal section XAI, and abnormal sensor XAI.

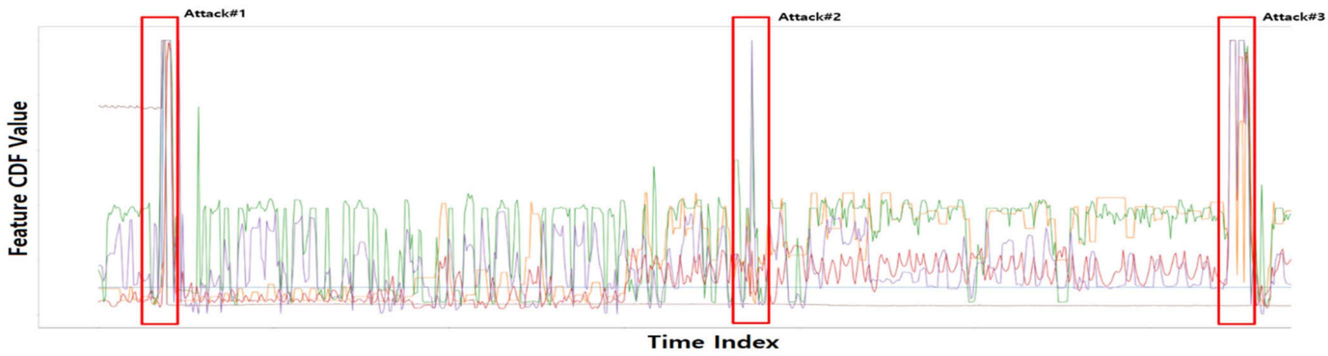


FIGURE 20. Specific anomaly features: flow plot. Visualization of specific anomaly features for attack sections #1 to #3 in the HAI dataset. The sections marked with red boxes are Attack 1, Attack 2, and Attack 3, respectively. Contrary to Fig. 17, which visualizes all features, it is possible to determine whether there is an abnormality intuitively.

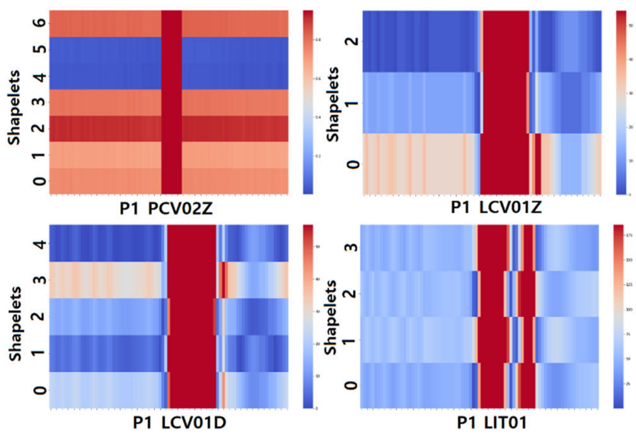


FIGURE 21. Specific anomaly features: 'P1_PCV02Z', 'P1_LCV01Z', 'P1_LCV01D', and 'P1_LIT01' features heatmap.

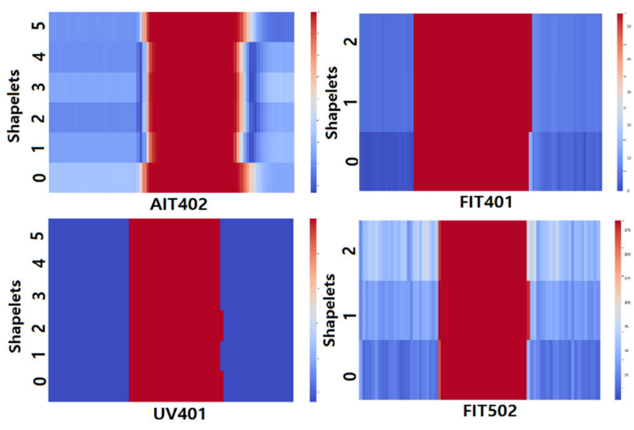


FIGURE 22. Specific anomaly features: 'AIT402', 'FIT401', 'UV401', and 'FIT502' features heatmap.

1) VISUALIZATION OF ALL FEATURES FOR A SPECIFIC ATTACK SECTION

Fig. 17 presents a visualization of all the features in the flow form for attack sections #1 to #3 in the HAI dataset. In a real-time operational environment, these can be expressed in flow form, as shown in Fig. 17.

However, the threshold value for each feature is different, and specific features have high values; therefore, it is better to mark only the features that are judged to be anomalies rather than all features.

Fig. 18 shows a heatmap visualization example for the normal section and an example of attack section #1 in the HAI dataset. Fig. 19 shows an example of heatmap visualization example for the normal section and an example for attack section #8 in the SWaT dataset. Fig. 18 (a) and 19 (a) show that some features have high values and appear as red anomalies, but the time point does not exceed the threshold and is in a normal state. Fig. 18 (b) and 19 (b) show many features as anomalies in red in the red box, and the time point also exceeds the threshold.

2) VISUALIZATION OF ANOMALY FEATURES FOR SPECIFIC ATTACK SECTION

Fig. 20 presents a flow plot of the features judged to be abnormal for the section, including attack sections #1 to #3 in the HAI dataset. By visualizing the specific features that affect an attack, one can immediately respond to abnormalities in a real-time operating environment.

Fig. 21 shows an example of heatmap visualization for some anomaly sensors for attack section I, which is visualized in red in Fig. 18 (b). As in Section III, E) in Fig. 11, the x-axis is the time index for the section including attack 1, and the Y-axis is the representative pattern of each feature.

The area where the color of the heatmap is red represents an attack. Fig. 22 shows an example of heatmap visualization for some anomaly sensors for attack section 8, which is visualized in red in Fig. 19 (b). The X-axis is the time index for the section including Attack 8, and the y-axis is the representative pattern of each feature. The area where the color of the heatmap is red represents an attack. According to the heatmap, it can be seen that the values of the features appear in red for the section judged to be an attack.

Fig. 23 presents an example of the visualization of the "P1_PCV02D" feature among the above features as a shapelet, a normal value, and a value for attack section #1 in the HAI dataset. Blue line, which is a normal value, clearly

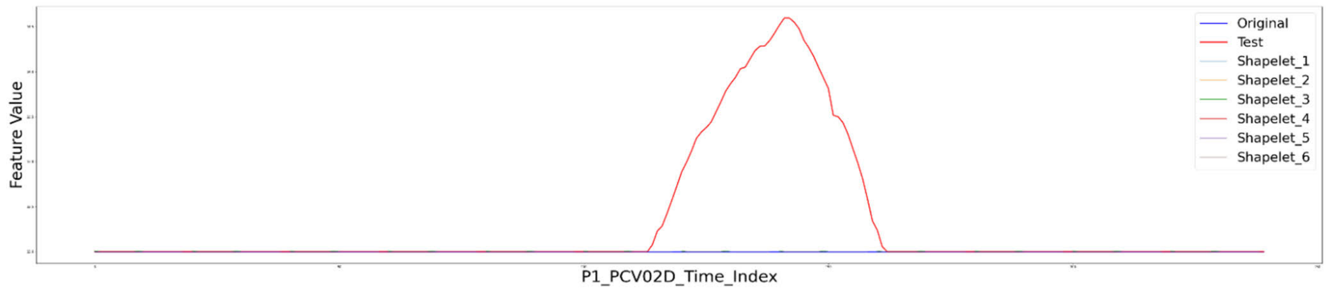


FIGURE 23. "P1_PCV02D" feature normal/abnormal/shapelet values plot in the HAI dataset.

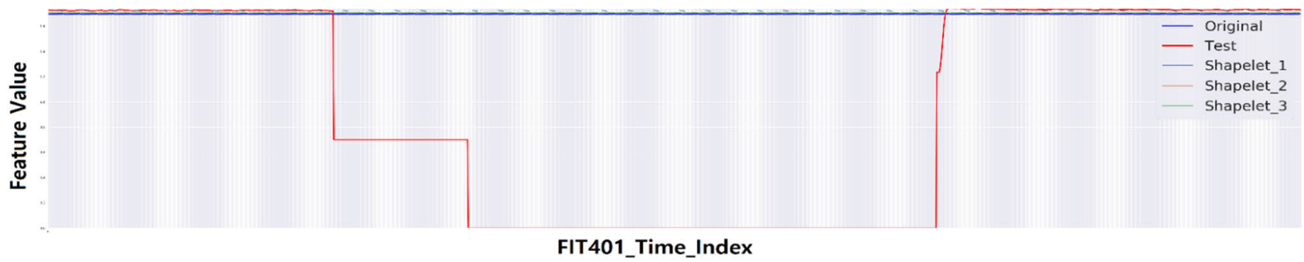


FIGURE 24. "FIT401" feature normal/abnormal/shapelet values plot in the SWaT dataset.

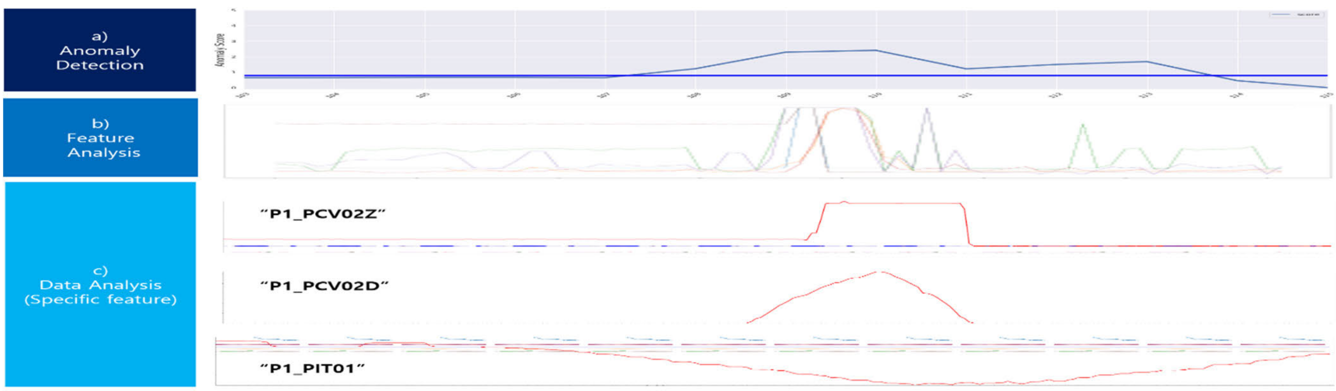


FIGURE 25. Example of a real operating environment with the HAI dataset. It is constructed through visual and numerical data calculated in Section IV. a) Anomaly detection example: anomaly score plot for attack 1 section (same as Fig. 15 method) b) Example of specific feature analysis: specific anomaly feature flow plot for the same attack 1 section (same as Fig. 20 method) c) Detailed data analysis example: "P1_PCV02Z", "P1_PCV02D", and "P1_PIT01" feature that appeared as an anomaly in the same attack 1 section Representative pattern and actual value flow plot for each target feature (same as Fig. 23 method).

forms a value in a category similar to that of the shapelet. However, the red line, indicating an anomalous value, shows a significant difference from the shapelet. Fig. 24 presents an example of the visualization of the "FIT401" feature among the above features as a shapelet, a normal value, and a value for attack section #8 in the SWaT dataset. The blue line, which is the normal value, clearly forms a value similar to that of the shapelet. However, the red line, indicating an anomalous value, shows a significant difference from the shapelet.

3) EXAMPLE OF APPLICATION IN A REAL OPERATING ENVIRONMENT

Fig. 25 and 26 are examples of an application in a real operating environment using the visual and numerical data calculated in Section IV, the experiment section. Fig. 25 and 26 a) indicate the light blue outlier score that exceeds the threshold indicated by the blue X-axis for the attack section. Experts can a) Utilize "Anomaly Detection" to immediately control an anomaly for a point in time. Fig. 25 and Fig. 26 b) indicate the CDF values for sensors that appear abnormal in some areas in the attack 1 section. Experts can also use Fig. 25 and 26 b) "Feature Analysis" and Fig. 25 and 26 c) "Data Analysis (Specific feature)" to determine which sensor has a problem. Fig. 25 and 26 c) represent the identified abnormal time point in red, the normal random time point in blue, and the normal representative pattern of the corresponding sensor in a different color. The normal pattern shows values similar to the representative pattern, but the identified abnormal points show a large difference, allowing the identification of abnormalities. Furthermore,

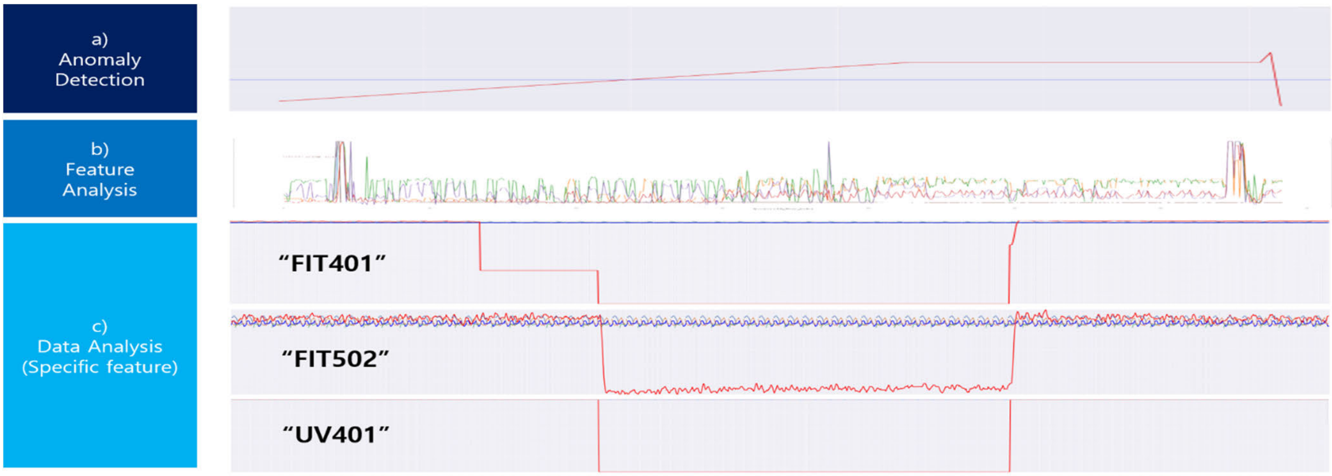


FIGURE 26. Example of a real operating environment with the SWaT dataset. It is constructed through visual and numerical data calculated in Section IV. a) Anomaly detection example: anomaly score plot for attack 8 section (same as Fig. 16 method) b) Example of specific feature analysis: specific anomaly feature flow plot for the same attack 8 section c) Detailed data analysis example: "FIT401", "FIT502", and "UV401" feature that appeared as an anomaly in the same attack 8 section Representative pattern and actual value flow plot for each target feature (same as Fig. 24 method).

the progress of an attack can be analyzed in detail. In the example shown in Fig. 25, in the HAI dataset, the sequence of abnormal values for sensor "P1_PCV02Z" begins first. As a result, the "P1_PCV02D" and "P1_PIT01" sensors record abnormal values, so "P1_PCV02Z" can be specified as the attack launch point sensor. Similar to the example in Fig. 26 for the SWaT dataset, the sequence of abnormal values for sensor "FIT401" starts first. As a result, the "FIT502" and "UV401" sensors record abnormal values, so "FIT401" can be specified as the attack launch point sensor.

4) CONTROL SYSTEM STRUCTURE AND FEATURES THAT CAN AFFECT ATTACKS

As a result of anomaly detection for the entire section, 41 of 50 attacks were detected in the HAI dataset. The structure of each attack provided by the Korea National Security Research Institute, which created and published the data, is shown in Fig. 27 and 28. If an attack or malfunction occurs in a specific sensor among the sensors that constitute the system, other nearby sensors may be affected.

Fig. 27 a) "Pressure control of the boiler (P1-PC)" It consists of sensors "PCV01," "PCV02," and "PIT01."

Fig. 27 b) "Level control of the boiler (P1-LC)" It consists of sensors "FCV03," "LCV01," and "LIT01.". Also, Fig. 27 c) Since "Flow rate control of boiler (P1-FC)" is also connected, the "FIT03" sensor may also be affected.

Fig. 27 d) "Speed control of a turbine (P2-SC)" It consists of sensors "SIT01" and "CO_rpm."

Fig. 28 a) "Turbine process control architecture (P2-TC)" It consists of sensors "OnOff" and "HiLout.". Also, Fig. 27 d) Since "Speed control of a turbine (P2-SC)" is also connected, "SIT01" and "CO_rpm" sensors may also be affected.

Fig. 28 b) "Water level control in the water treatment plant (P3-LC)" It consists of sensors "LCV01," "LCP01," and "LT01.".

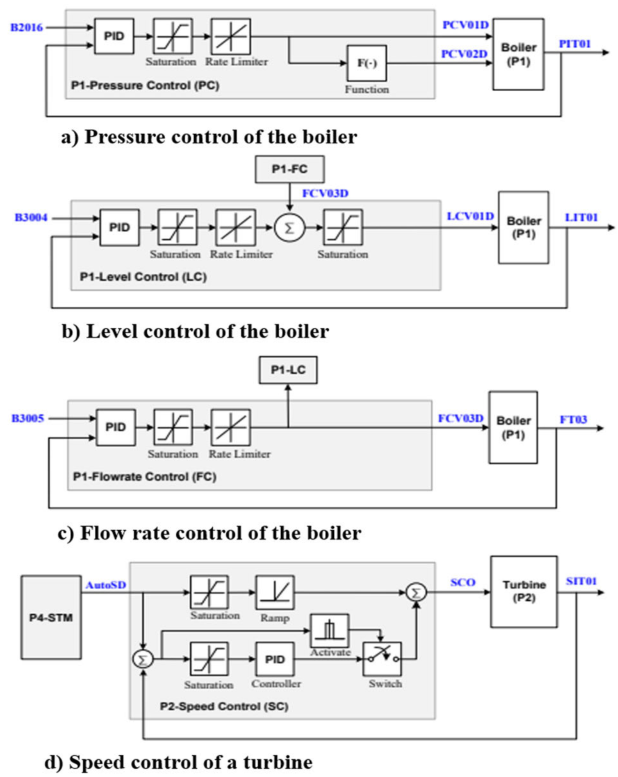


FIGURE 27. Detailed system structure and influencing features.

5) SHAPELET-BASED FEATURES CONTRIBUTING TO THE ATTACK (XAI RESULTS)

The results of extracting all the features that contribute significantly to the 41 detected attacks are shown in Appendix Table 6.

Attack sections #1 to #25 are single attacks, whereas sections #26–#50 are compound attacks. The features marked in red and blue indicate features that can be affected by the

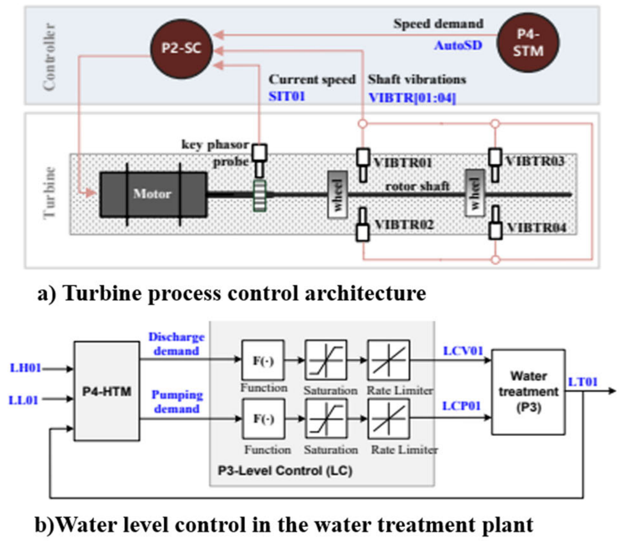


FIGURE 28. Water treatment, turbine process system structure, and influencing features.

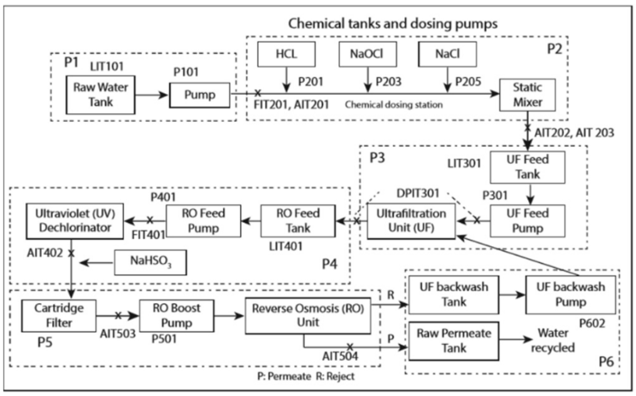


FIGURE 29. SWaT testbed processes overview [32].

TABLE 5. Performance composition.

	Model Performance	Interpretation Method (XAI)	Interpretation Performance
Bi LSTM-Based [31]	HAI Dataset Acc: 0.98	SHAP Based	-
One-class Neural network-based [34]	SWaT Dataset F1: 0.87	-	-
Proposed Method Based	HAI Dataset Acc: 0.82 SWaT Dataset Acc: 0.80	Shapelet Based	HAI Dataset: 0.95 SWaT Daataset: 0.88

structure of the control system. Nine undetected attacks had an orange background, and when the fault sensor detected only one attack during a complex attack, it was indicated by a gray background.

As a result of anomaly detection for the entire section, 25 of 36 attacks were detected in the SWaT dataset. The structure for each attack provided by iTrust, Center for Research in Cyber Security, Singapore University of Technology and Design, which created and published the data, is shown in Fig. 29 [32]. The process in Fig. 29 consists of six sub-processes, as shown in Table 3, which consists of P1: 5 features, P2: 11 features, P3: 9 features, P4: 9 features, P5: 13 features, and P6: 4 features.

P1 is the physical stage of raw water supply and storage, P2 is the chemical dosing stage, P3 is the filtering stage called Ultrafiltration (UF), P4 is dechlorination using Ultraviolet (UV) lamps, P5 is the feeding stage using a Reverse Osmosis (RO) system, and P6 is a backwash step that cleans the membranes using RO permeate.

In addition, descriptions of the attack time, attack sensor point, and impact of each attack are presented in Appendix Table 7.

The results of extracting all the features that contribute significantly to the 25 detected attacks are shown in Appendix Table 8.

The features marked in red indicate those that can be affected by the structure of the control system. Eleven undetected attacks had orange backgrounds.

V. DISCUSSION

Compared to the paper "E-SFD: Explainable Sensor Fault Detection in the ICS Anomaly Detection System" by Hwang and Lee. Hwang and Lee used the same HAI dataset using the Bi-LSTM model to achieve 98% accuracy and heatmap analysis through SHAP and Feature Importance [31].

Compared to the paper "Anomaly detection for a water treatment system based on one-class Neural network" [34], who used the same SWaT dataset, compared the performance using various models, and claimed a method through the NN-one class. Using this method, an 87% f1-score was achieved. However, the cause of the detected attack was not analyzed.

Compared with this study, the performance is lower than that of the deep learning model in terms of accuracy in detecting anomalies. However, deep learning methods cannot determine the exact cause in the form of a black box. Even if an anomaly is detected in terms of actual use, if it is impossible to find and analyze the exact cause, the operator must review all the sensors. Therefore, the time required to take action and the workload of the operator inevitably increases.

Boateng et al., using the SWaT dataset, did not analyze the cause of the detected attack.

Hwang and Lee analyzed the cause using the HAI dataset through a Heatmap and Feature Importance using SHAP. However, these methods cannot be interpreted. In this study, because the patterns of real data and actual feature values are visualized and used for comparison with real-time data and analysis of causes, the possibility of interpretation is higher than that of analysis using SHAP.

TABLE 6. Detected attacks and features contributing to attacks in the HAI dataset.

	Attack#1	Attack#2	Attack#3	Attack#4	Attack#5	Attack#6	Attack#7	Attack#8	Attack#9	Attack#10
0	P1_PCV02Z	P1_PIT01	P1_LCV01D	P2_SIT01	P2_SIT01	P2_OnOff	P1_PCV02Z	undetected	P1_LIT01	undetected
1	P1_PCV02D	P1_FT01	P1_LIT01	P2_CO_rpm	P2_CO_rpm	P2_SIT01	P1_PCV02D		P1_PIT01	
2	P1_LCV01D	P1_FCV03Z	P1_LCV01Z	P2_HILout		P2_CO_rpm	P1_LCV01D		P1_LCV01D	
3	P1_PIT01	P1_LCV01D	P1_PIT01	P1_FCV01Z		P2_HILout	P1_LIT01		P1_FCV03D	
4	P1_FT01	P1_FT03Z	P1_FT01	P3_PIT01		P1_TIT02	P1_LCV01Z		P1_FT01	
5	P1_LCV01Z	P1_FT03	P1_TIT01			P3_FIT01	P1_PIT01		P1_FCV03Z	
6	P1_PCV01Z	P1_FCV03D	P1_FT01Z				P1_PCV01Z		P1_LCV01Z	

	Attack#11	Attack#12	Attack#13	Attack#14	Attack#15	Attack#16	Attack#17	Attack#18	Attack#19	Attack#20
0	P1_FCV03D	P2_OnOff	P1_LCV01D	P1_LCV01D	P1_LCV01D	undetected	P2_CO_rpm	P1_LCV01D	undetected	undetected
1	P1_FCV03Z	P2_SIT01	P1_LCV01Z	P1_LIT01	P1_LCV01Z		P2_SIT01	P1_LCV01Z		
2	P1_PIT02	P2_CO_rpm	P1_PIT01	P1_LCV01Z	P1_FT01			P1_PIT01		
3		P2_HILout	P1_FT01	P1_PIT01	P3_FIT01			P1_FT01		
4		P3_FIT01	P1_FT01Z	P1_FT01	P1_TIT02			P1_FT01Z		
5		P1_PCV01Z	P1_PCV01Z	P2_HILout						
6				P1_FT01Z						

	Attack#21	Attack#22	Attack#23	Attack#24	Attack#25	Attack#26	Attack#27	Attack#28	Attack#29	Attack#30
0	undetected	P2_OnOff	P1_LCV01D	P2_CO_rpm	undetected	**P1_LCV01D**	P1_PCV02Z	P1_LCV01D	P2_OnOff	P2_OnOff
1		P2_SIT01	P1_LCV01Z	P3_FIT01		P1_PIT01	P1_PCV02D	P1_LIT01	**P2_SIT01**	**P2_SIT01**
2		P2_CO_rpm	P1_LIT01	P2_SIT01		P1_FT01	P1_LCV01D	P1_LCV01Z	**P2_CO_rpm**	**P2_CO_rpm**
3		P2_HILout	P1_PIT01	P3_LCP01D		**P1_LCV01Z**	P1_LIT01	P1_PIT01	P4_ST_TT01	P2_HILout
4			P1_FT01	P3_PIT01		**P1_FT03**	P1_LCV01Z	P1_FT01	P2_HILout	P1_PCV02Z
5			P2_SIT01	P1_B4022		**P1_FCV03Z**	P1_FCV03D	P1_FCV01Z	P1_FT02	
6			P1_B4022				P1_PIT01	P1_FT01Z	P1_PCV02Z	

	Attack#31	Attack#32	Attack#33	Attack#34	Attack#35	Attack#36	Attack#37	Attack#38	Attack#39	Attack#40
0	**P2_SIT01**	P2_OnOff	P1_LCV01D	P1_LCV01D	P1_PCV02D	P1_LCV01D	P1_LCV01D	P2_SIT01	P2_OnOff	P2_SIT01
1	**P2_CO_rpm**	P1_LCV01D	P1_LCV01Z	P1_LCV01Z	P1_PCV02Z	P1_LIT01	P1_LCV01Z	P2_CO_rpm	P2_SIT01	P2_CO_rpm
2	P1_PCV02Z	P1_FCV03D	P1_PIT01	P1_PIT01	P1_FCV03D	P1_LCV01Z	P2_CO_rpm	P3_LCV01D	P2_CO_rpm	P1_FCV03D
3	P1_TIT02	P2_SIT01	P1_FT01	P1_FT01	P1_LCV01D	P1_PIT01	P1_PIT01	P1_PCV01Z	P1_LCV01D	P1_FCV03Z
4		P2_CO_rpm	P1_FCV02D	P1_PCV02Z	P1_LCV01Z	P2_CO_rpm	P4_ST_TT01	P3_LCP01D	P2_HILout	P1_PIT01
5		P1_PIT01	P1_FT01Z	P2_SIT01	P1_PIT01	P1_FT01	P1_FT01		P1_TIT01	P4_ST_LD
6		P1_FT01	P1_PCV02Z	P2_CO_rpm	P1_FCV03Z	P2_SIT01	P2_SIT01		P1_FCV01Z	P1_LCV01D

	Attack#41	Attack#42	Attack#43	Attack#44	Attack#45	Attack#46	Attack#47	Attack#48	Attack#49	Attack#50
0	P1_LCV01D	P1_LCV01D	P1_LCV01D	P2_OnOff	undetected	**P1_LCV01D**	undetected	**P1_LCV01D**	P2_OnOff	P1_PCV02Z
1	P1_LCV01Z	P1_LCV01Z	P1_LIT01	P2_SIT01		**P1_FCV03D**		**P1_LIT01**	P2_SIT01	P1_PCV02D
2	P2_SIT01	P1_PIT01	P1_LCV01Z	P2_CO_rpm		P1_PIT01		**P1_LCV01Z**	P2_CO_rpm	**P1_FCV03D**
3	P2_CO_rpm	P1_FT01	P1_PIT01	P2_HILout		P1_FT01		P1_PIT01	P2_HILout	**P1_LIT01**
4	P1_PIT01	P3_LCV01D	P1_FT01	P4_ST_LD		P4_ST_TT01		P1_FT01	P1_FT01	P1_PIT01
5	P1_FT01	P1_FCV03D	P1_FT01Z	P1_FCV02D				**P1_FCV03Z**	P3_LCV01D	**P1_LCV01D**
6	P1_LIT01	P1_FT01Z	P1_FCV03D	P3_LCP01D				P2_CO_rpm	P1_FT01Z	**P1_FCV03Z**

As the above comparison is in contrast to existing XAI methodologies, this study improved interpretability by showing an example of a steady state through real data. In addition, analysis information on individual fault sensors contributing to an abnormal state that could not be calculated in the existing black-box model was provided based on actual data. Based on this, the analysis provides the necessary information for operator decision making. It supports an environment in

TABLE 7. Attack descriptions: attack time, attack sensor point, and impact of each attack.

	Start Time	End Time	Attack point	Expected Impact or attacker intent
1	10:29:14	10:58:30	MV-101	Tank overflow
2	10:51:08	11:28:22	P-102	Pipe bursts
3	11:22:00	11:54:08	LIT-101	Tank Underflow; Damage P-101
4	11:47:39	12:04:10	MV-504	Halt RO shut down sequence; Reduce the life of RO
5	12:00:55	12:15:33	AIT-202	P-203 turns off; Change in water quality
6	12:08:25	13:26:13	LIT-301	Stop of inflow; Tank underflow; Damage P-301
7	13:10:10	14:28:20	DPIT-301	The backwash process is started again and again; Normal operation stops; Decrease in the water level of tank 401. Increase in water level of tank 301
8	14:16:20	14:28:20	FIT-401	UV shutdown; P-501 turns off;
9	14:19:00	11:15:17	FIT-401	UV shutdown; P-501 turns off;
10	11:11:25	11:42:50	MV-304	Halt of stage 3 because of change in the backwash process
11	11:35:40	12:02:00	Mv-303	Halt of stage 3 because of change in the backwash process
12	11:57:25	14:50:08	LIT-301	Tank Overflow
13	14:38:12	18:15:01	MV-303	Halt of stage 3 because change in the backwash process
14	18:10:43	18:22:17	AIT-504	RO shutdown sequence starts after 30 minutes. Water should go to the drain.
15	18:15:43	18:42:00	AIT-504	RO shutdown sequence starts after 30 minutes. Water should go to the drain.
16	18:30:00	23:03:00	MV-101, LIT-101	Tank overflow
17	22:55:18	01:54:10	UV-401, AIT-502, P-501	Possible damage to RO
18	01:42:34	09:56:28	P-602, DIT-301, MV-302	System freeze
19	09:51:08	10:12:01	P-203, P-205	Change in water quality
20	10:01:50	17:29:00	LIT-401, P-401	Tank underflow
21	17:04:56	01:45:18	P-101, LIT-301	Tank 101 underflow; Tank 301 overflow
22	01:17:08	11:15:27	P-302, LIT-401	Tank overflow
23	01:45:19	15:34:00	P-302	Stop inflow of tank T-401
24	15:32:00	16:07:10	P-201, P-203, P-205	Wastage of chemicals
25	15:47:40	22:11:40	LIT-101, P-101, MV-201	Tank 101 underflow; Tank 301 overflow
26	22:05:34	10:46:00	LIT-401	Tank overflow
27	10:36:00	14:28:35	LIT-301	Tank underflow; Damage P-302
28	14:21:12	17:14:20	LIT-101	Tank underflow; Damage P-101
29	17:12:40	17:26:56	P-101	Stops outflow
30	17:18:56	22:25:00	P-101; P-102	Stops outflow
31	22:16:01	11:24:50	LIT-101	Tank overflow
32	11:17:02	11:36:18	P-501, FIT-502	Reduced output
33	11:31:38	11:50:28	AIT-402, AIT-502	Water goes to drain because of overdosing
34	11:43:48	11:56:38	FIT-401, AIT-502	UV will shut down and water will go to RO
35	11:51:42	13:40:56	FIT-401	UV will shut down and water will go to RO
36	13:13:02	10:58:30	LIT-301	Tank overflow
Single Stage Single Point Attacks				
Single Stage Multi Point Attacks				
Multi Stage Single Point Attacks				
Multi Stage Multi Point Attacks				

which a response action can be quickly taken by calculating the priority when a fault sensor occurs.

However, further improvements in detection rates are needed. Detecting anomalies based on current Euclidean distance. For a more advanced detection, it is necessary to establish a mathematical algorithm, and future research is planned. The results of the performance comparison are listed in Table 5. The compared methodologies either did not conduct analyses or, even if they did, did not verify the accuracy of the interpretation. However, the method proposed in this paper ultimately achieved a performance of over 95% for the HAI dataset and over 85% for the SWaT dataset.

VI. CONCLUSION

With the development of information and communication technology, research on AI and the introduction of smart environments is being conducted to respond to various attacks. However, as AI performance improves, internal interpretability becomes more complex and must rely only on AI prediction, which cannot be interpreted. As a result, the reliability issues are emerging, and operators need to check all possible sensor faults.

This study enhances credibility by providing information about detection results and detecting fault sensors to operators who monitor, analyze, and act on ICSs operating in a time series environment.

In a real operating environment, a large amount of data is provided in real time, but the number of experts who can analyze or act on it is limited. Moreover, if the detailed internal structure is unknown, appropriate actions cannot be performed. The method proposed in this study solves these problems by providing information about the detected fault sensor, information on the corresponding sensor in normal times, and representative patterns.

TABLE 8. Detected attacks and features contributing to attacks in the SWaT dataset.

	Attack#1	Attack#2	Attack#3	Attack#4	Attack#5	Attack#6	Attack#7	Attack#8	Attack#9	Attack#10
0	MV201	P602	MV101	undetected	P602	MV201	P602	P602	P602	undetected
1	LIT101	P102	P302		P302	LIT301	P302	UV401	UV401	
2	MV101	P302	LIT01		DPIT301		DPIT301	FIT502	FIT502	
3					AIT202		MV201	FIT401	FIT401	

	Attack#11	Attack#12	Attack#13	Attack#14	Attack#15	Attack#16	Attack#17	Attack#18	Attack#19	Attack#20
0	undetected	MV201	undetected	AIT504	AIT504	undetected	undetected	P602	P602	undetected
1		MV302		P602	P602			AIT504	DPIT301	
2		LIT301		DPIT301	DPIT301			P501	P205	
3				MV302	MV302			MV302	MV201	

	Attack#21	Attack#22	Attack#23	Attack#24	Attack#25	Attack#26	Attack#27	Attack#28	Attack#29	Attack#30
0	P101	P102	undetected	undetected	P102	undetected	P302	P602	P101	undetected
1	MV101	P302			MV101		MV302	MV101	P102	
2	LIT301	LIT401			LIT101		LIT301	LIT101		
3	P302	P602			P101			MV302		

	Attack#31	Attack#32	Attack#33	Attack#34	Attack#35	Attack#36
0	MV201	FIT502	AIT502	FIT401	UV401	MV201
1	LIT101	FIT401	AIT402	AIT502	FIT401	LIT301
2	MV101	P302		UV401	P501	
3		P501			AIT502	

The information calculated using the methodology confirmed the following results through reliable visual and quantitative values for abnormal signs.

In the HAI dataset, operators could respond to 39 of the 41 detected attacks by checking only the top three sensors (approximately 4%). We were able to respond to all attacks detected through the proposed methodology when we checked the top five sensors (approximately 6%). In the SWaT dataset, operators responded to 22 of the 25 detected attacks by checking only the top four sensors (approximately 7%).

In conclusion, if the operator confirms the key information (approximately 4% to 7%) of the attack, as shown in Appendix Tables 6 and 7 of the verification results for the two datasets, the operator can detect and interpret more than 85% to 95% of the attacks. Therefore, experts who previously had to work on many sensors could respond quickly to threats by only working on a few sensors. This is expected to improve efficiency and availability because experts who need to respond can take immediate action.

APPENDIX

See Tables 6, 7, and 8.

REFERENCES

[1] J. Sakhnini, H. Karimipour, A. Dehghantanha, R. M. Parizi, and G. Srivastava, "Security aspects of Internet of Things aided smart grids: A bibliometric survey," *Internet Things*, vol. 14, Jun. 2021, Art. no. 100111.

[2] H. HaddadPajouh, A. Dehghantanha, R. M. Parizi, M. Aledhari, and H. Karimipour, "A survey on Internet of Things security: Requirements, challenges, and solutions," *Internet Things*, vol. 14, Jun. 2021, Art. no. 100129.

[3] H. Karimipour and V. Dinavahi, "Robust massively parallel dynamic state estimation of power systems against cyber-attack," *IEEE Access*, vol. 6, pp. 2984–2995, 2017.

[4] Y. Zhao, T. Li, X. Zhang, and C. Zhang, "Artificial intelligence-based fault detection and diagnosis methods for building energy systems: Advantages, challenges and the future," *Renew. Sustain. Energy Rev.*, vol. 109, pp. 85–101, Jul. 2019.

[5] K. Michail, K. M. Deliparaschos, S. G. Tzafestas, and A. C. Zolotas, "AI-based actuator/sensor fault detection with low computational cost for industrial applications," *IEEE Trans. Control Syst. Technol.*, vol. 24, no. 1, pp. 293–301, Jan. 2016.

[6] A. Mellit and S. Kalogirou, "Artificial intelligence and Internet of Things to improve efficacy of diagnosis and remote sensing of solar photovoltaic systems: Challenges, recommendations and future directions," *Renew. Sustain. Energy Rev.*, vol. 143, Jun. 2021, Art. no. 110889.

[7] W. Lang, Y. Hu, C. Gong, X. Zhang, H. Xu, and J. Deng, "Artificial intelligence-based technique for fault detection and diagnosis of EV motors: A review," *IEEE Trans. Transport. Electrific.*, vol. 8, no. 1, pp. 384–406, Mar. 2022.

[8] Ö. Gültekin, E. Cinar, K. Özkan, and A. Yazıcı, "Real-time fault detection and condition monitoring for industrial autonomous transfer vehicles utilizing edge artificial intelligence," *Sensors*, vol. 22, no. 9, p. 3208, Apr. 2022.

[9] A. Abdallah, M. A. Maarof, and A. Zainal, "Fraud detection system: A survey," *J. Netw. Comput. Appl.*, vol. 68, pp. 90–113, Jun. 2016.

[10] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: Techniques, datasets and challenges," *Cybersecurity*, vol. 2, no. 1, Dec. 2019.

[11] M. Canizo, I. Triguero, A. Conde, and E. Onieva, "Multi-head CNN–RNN for multi-time series anomaly detection: An industrial case study," *Neurocomputing*, vol. 363, pp. 246–260, Oct. 2019.

[12] P. Liu, X. Sun, Y. Han, Z. He, W. Zhang, and C. Wu, "Arrhythmia classification of LSTM autoencoder based on time series anomaly detection," *Biomed. Signal Process. Control*, vol. 71, Jan. 2022, Art. no. 103228.

[13] S. M. Lundberg and S. I. Lee, "A unified approach to interpreting model predictions," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.

[14] F. Karim, S. Majumdar, H. Darabi, and S. Harford, "Multivariate LSTM-FCNs for time series classification," *Neural Netw.*, vol. 116, pp. 237–245, Aug. 2019.

[15] Y. Hao and H. Cao, "A new attention mechanism to classify multivariate time series," in *Proc. 29th Int. Joint Conf. Artif. Intell.*, Jul. 2020.

[16] X.-M. Yu, W.-Z. Feng, H. Wang, Q. Chu, and Q. Chen, "An attention mechanism and multi-granularity-based bi-LSTM model for Chinese Q&A system," *Soft Comput.*, vol. 24, no. 8, pp. 5831–5845, Apr. 2020.

[17] Y. Li, Z. Zhu, D. Kong, H. Han, and Y. Zhao, "EA-LSTM: Evolutionary attention-based LSTM for time series prediction," *Knowl.-Based Syst.*, vol. 181, Oct. 2019, Art. no. 104785.

[18] 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. Discovery Data Mining*, 2016, pp. 1–10.

[19] L. S. Shapley, *A Value for N-Person Games*, 1953.

[20] W. E. Marcílio and D. M. Eler, "From explanations to feature selection: Assessing SHAP values as feature selection mechanism," in *Proc. 33rd SIBGRAPI Conf. Graph., Patterns Images (SIBGRAPI)*, Nov. 2020, pp. 340–347.

[21] S. M. Lundberg, G. Erion, H. Chen, A. DeGrave, J. M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal, and S.-I. Lee, "From local explanations to global understanding with explainable AI for trees," *Nature Mach. Intell.*, vol. 2, no. 1, pp. 56–67, Jan. 2020.

[22] L. Ye and E. Keogh, "Time series shapelets: A new primitive for data mining," in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jun. 2009, pp. 947–956.

[23] J. Zakaria, A. Mueen, and E. Keogh, "Clustering time series using unsupervised-shapelets," in *Proc. IEEE 12th Int. Conf. Data Mining*, Dec. 2012, pp. 785–794.

[24] A. Yamaguchi and T. Nishikawa, "One-class learning time-series shapelets," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2018, pp. 2365–2372.

[25] S. M. Tripathy, A. Chouhan, M. Dix, A. Kotriwala, B. Klöpper, and A. Prabhune, "Explaining anomalies in industrial multivariate time-series data with the help of explainable AI," in *Proc. IEEE Int. Conf. Big Data Smart Comput. (BigComp)*, Jan. 2022, pp. 226–233.

[26] A. Warnecke, D. Arp, C. Wressnegger, and K. Rieck, "Evaluating explanation methods for deep learning in security," in *Proc. IEEE Eur. Symp. Secur. Privacy (EuroS&P)*, Sep. 2020, pp. 158–174.

[27] J. G. Kim, C. W. Hwang, and T. J. Lee, "A study on evaluation methods for interpreting AI results in malware analysis," *J. Korea Inst. Inf. Secur. Cryptol.*, vol. 31, no. 6, pp. 1193–1204, 2021.

[28] H. Heaton and S. W. Fung, "Explainable AI via learning to optimize," *Sci. Rep.*, vol. 13, no. 1, p. 10103, 2023.

[29] G. Vandewiele, F. Ongenae, and F. De Turck, "GENDIS: Genetic discovery of shapelets," *Sensors*, vol. 21, no. 4, p. 1059, 2021.

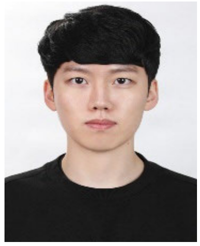
[30] H. K. Shin, W. Lee, J. H. Yun, and H. Kim, "HAI 1.0: HIL-based augmented ICS security dataset," in *Proc. 13th USENIX Conf. Cyber Secur. Experim. Test*, Aug. 2020, pp. 1–5.

[31] H.-K. Shin, W. Lee, J.-H. Yun, and B.-G. Min, "Two ICS security datasets and anomaly detection contest on the HIL-based augmented ICS testbed," in *Proc. Cyber Secur. Experim. Test Workshop*, Aug. 2021, pp. 36–40.

[32] C. Hwang and T. Lee, "E-SFD: Explainable sensor fault detection in the ICS anomaly detection system," *IEEE Access*, vol. 9, pp. 140470–140486, 2021.

[33] J. Goh, S. Adepu, K. N. Junejo, and A. Mathur, "A dataset to support research in the design of secure water treatment systems," in *Proc. 11th Int. Conf. Crit. Inf. Infrastruct. Secur. (CRITIS)*, Paris, France, Springer, Oct. 2016, pp. 88–99.

[34] E. A. Boateng, J. W. Bruce, and D. A. Talbert, "Anomaly detection for a water treatment system based on one-class neural network," *IEEE Access*, vol. 10, pp. 115179–115191, 2022.



SUENGBUM LIM received the B.S. degree from the Computer Engineering Department, Hoseo University, Asan-si, South Korea, in 2023, where he is currently pursuing the M.Sc. degree with the Information Security Department.

His research interests include malware analysis, machine learning, and anomaly detection.

Prof. Lim won the Excellence Prize at the Cyber Security Thesis Contest hosted by the Korea Internet & Security Agency (KISA), in 2022. In 2022, he won the Grand Prize at the Cyber Security Idea Challenge hosted by KISA.

JINGANG KIM received the B.S. degree in information security and the M.Sc. degree from the Information Security Department, Hoseo University, Asan-si, South Korea, in 2022. His research interests include artificial intelligence, intrusion detection, and anomaly detection.

In 2022, he won the Excellence Prize at the Cyber Security Thesis Contest hosted by the Korea Internet & Security Agency (KISA). In 2022, he won the Grand Prize at the Cyber Security Idea Challenge hosted by KISA.

TAEJIN LEE received the Graduate degree from the Postech Computer Engineering Department, in 2003, the degree from Yonsei University, in 2008, and the degree from Ajou University, in 2017.

He was with Korea Internet Security Agency, from 2003 to 2017, and has been with Hoseo University, since 2017. His research interests include artificial intelligence decision support, false alarm detection, explainable AI, and cyber securities, such as intrusion detection, malware analysis, attack profiling, fraud detection, artificial intelligence security, and trustworthy AI.