Transactions of the Korean Nuclear Society Spring Meeting
Jeju, Korea, May 9-10, 2024

Autoregressive Multivariate Time Series Forecasting of Total Loss of Component Cooling Water Accident Sequences Calculated by Modular Accident Analysis Program, Inspired by Video Prediction Methods Using Deep Learning

Joon Young Bae [a], JinHo Song [a], Chang Hyun Song [a], Injae Kim [a], Sung Joong Kim [a, b*] Jeong Ik Lee [c], Miro Seo [d], Tae-Young Shin [d]
[a]*Department of Nuclear Engineering, Hanyang University, 222, Wangsimni-ro, Seongdong-gu, Seoul, Korea*
[b]*Institute of Nano Science and Technology, Hanyang University, 222, Wangsimni-ro, Seongdong-gu, Seoul, Korea*
[c]*Department of Nuclear and Quantum Engineering, KAIST, 291, Daehak-ro, Yuseong-gu, Daejeon, Korea*
[d] *KOREA Hydro & Nuclear Power Co., Ltd., Central Research Institute, 701312-gil, Yuseong-daero, Yuseong-gu, Daejeon, Korea*
[*]*Corresponding author: sungjkim@hanyang.ac.kr*

Keywords : MAAP, surrogate model, deep learning, multivariate time series, autoregressive model

1. Introduction

Research has been underway to predict the thermal hydraulic behaviors of Nuclear Power Plants (NPP) and to derive an optimal combination of mitigation strategies to support operator activities during severe accidents [1-11]. As an extension of this ongoing research, this paper aims to predict the behavior of nuclear power plants in real-time during severe accidents using deep learning methods. This approach seeks to replace existing integrated severe accident analysis codes, which are associated with high computational costs.

The proposed surrogate model, designed to predict thermal hydraulic behavior in real-time, is trained using extensive analysis results from integrated severe accident analysis codes. However, ensuring accuracy poses an inherent challenge, as predictions for the next time step depend on a restricted subset of thermal hydraulic variables from the analysis results, and the interrelationships between these variables may not be fully understood.

To address this challenge, this paper explores the use of deep learning-based video prediction methods. Leveraging the similarity between multivariate time series and video data, these methods aim to rapidly predict thermal hydraulic variables during severe accidents without significantly compromising accuracy. The methodology and outcomes of predicting such variables are discussed herein.

2. Methodologies

In this section, dataset used to train surrogate models, some of the techniques used to construct surrogate models, and results quantification method are explained. Surrogate models introduced in this paper are a fully connected (linear) layer model as a baseline model, and a convolutional neural network (CNN)-based model inspired by video prediction methods. A quantitative comparison was also conducted between the prediction results from surrogate models and the results of the integrated severe accident interpretation code.

2.1 Dataset Used to Train Surrogate Models

In order for the surrogate model to approximate the Modular Accident Analysis Program (MAAP) 5.03, an integrated severe accident analysis code, using deep learning techniques, a comprehensive dataset was constructed. This dataset encompasses 12,000 instances of Total Loss of Component Cooling Water (TLOCCW) scenarios specific to the Optimized Power Reactor 1000 (OPR1000) power plant. Instead of all components failing as soon as the accident begins, as in the static probabilistic safety assessment, the components involved in the accident were sequenced to fail at random times [12]. The occurrences of Reactor Coolant Pump Seal Failures followed a lognormal distribution with a mean time to failure of 5 hours, alongside 6 component failures and 3 mitigation strategies derived from the Severe Accident Management Guidelines (SAMG). Additionally, recirculation operation was not considered in the scenario. Component failures and implementation of mitigation strategies were randomized within a 72-hour timeframe, adhering to uniform distributions. Once a component failure or mitigation strategy was initiated, it remained in effect until the conclusion of the scenario. And even if the component is not malfunctioning, it may not work if it is affected by another component. For example, in the case of High Pressure Safety Injection (HPSI), a scenario was simulated so that even if it operates normally, it will not operate when the water level of Refueling Water Storage Tank is zero. Table I presents a comprehensive overview of components involved in the Total Loss of Component Cooling Water (TLOCCW) scenario, along with corresponding mitigation strategies. All variables are binary, represented as 0 or 1 depending on their operating states.

Table I. List of Related Components and Mitigation Strategies used as Surrogate Model Input

Component Failures	Reactor Coolant Pump Seal Failures
	Heat Exchanger
	High Pressure Safety Injection
	Low Pressure Safety Injection

	Containment Spray System
	Motor Driven Auxiliary Feedwater
	Charging Pump
Mitigation Strategies	Atmospheric Dump Valves Open Steam Generator External Injection
	Reactor Coolant System Depressurization
	Reactor Coolant System External Injection

Thermal hydraulic variables used as input and output to the model were also selected using SAMG. All thermal hydraulic variables except Reactor Pressure Vessel (RPV) status were selected only as variables that could be observed in the main control room of the plant. These variables were min-max scaled to values between 0 and 1, and the RPV status was composed of a binary form of 0 (RPV failure) and 1 (RPV not failure) at each time step. Table II is a list of variables used as input and output of the surrogate model.

Table II. List of variables used as both input and output of the surrogate model

Elapsed Time (Embedded)
Safety Injection Tank Pressure
Refueling Water Storage Tank Water Level
Cavity Pressure
Hot Leg Gas Temperature
Cold Leg Gas Temperature
Steam Generator 1 Secondary Pressure
Reactor Coolant System Pressure
RPV Water Level
Maximum Core Exit Temperature
Steam Generator 1 Downcomer Water Level
RPV Integrity

The dataset was divided into training set (9600 scenarios), validation set (1200 scenarios), and test set (1200 scenarios) at a ratio of 8:1:1. The validation set was used to prevent overfitting during training, and the test set is new data that the model has never encountered for the continuous inference phase.

The type of data injected varies depending on the model type used. For example, in a baseline model, only data from one-time point is injected into the model, while in another model, data from multiple consecutive time points can be injected into the model. Descriptions of the models continue below.

2.2 Problem Definition

The Dataset, $\mathcal{D} = \left(x_i^{C\times T\times H\times W}, y_i^{C\times T`\times H\times W`}\right)_{i=1}^{N}$ consists of input x, output y, channel $C = 1$, sequence lengths T, $T`$, the number of input features $W = 22$, the number of output features $W` = 12$, and the number of piece of data N. T may vary depending on the characteristics of the surrogate model, and since the surrogate models used in this study all produce a single output, $T`$ is 1. The goal of a surrogate model $\mathcal{F}_\Theta$ with learnable parameters $\Theta$ is to map $\mathcal{F}_\Theta : x_i^{C\times T\times H\times W} \mapsto y_i^{C\times T`\times H\times W`}$ by optimizing learnable parameters. Optimal parameters $\Theta^*$are as follows.

$$\Theta^* = \arg\min_{\Theta} \mathcal{L}(\mathcal{F}_\Theta(x), y)$$
$$x, y \in \mathcal{D}$$

Loss function $\mathcal{L}$ evaluates differences between predicted values $\mathcal{F}_\Theta(x)$ from the surrogate model, and desired values y from the dataset. Mean squared error was used in the baseline model, and Huber loss and binary cross-entropy loss were used together in the CNN-based model.

2.3 Baseline Model

2.3.1 Model Architecture

To compare the performance of surrogate models, a baseline model which is composed of linear blocks structure was constructed. One linear block consists of four parts [Linear layer, Batch Normalization, Gaussian Error Linear Units Function (GELU), and Dropout]. The linear layer is a fully connected layer, basic form of a neural network, and batch normalization is one of the methods to help the model converge better by normalizing the input data to the mean and standard deviation. GELU is a type of activation function injecting nonlinearity into a neural network, and dropout is a technique to improve the generalization performance of a neural network by randomly deleting some of the connections between neural network layers in the train stage [13, 14, 15]. Figure 1 is a schematic diagram of this model. '×7' means that 7 blocks are connected in series. The numbers 22 and 12 are the number of input and output features, respectively, and 1024 is the number of nodes in the linear block.

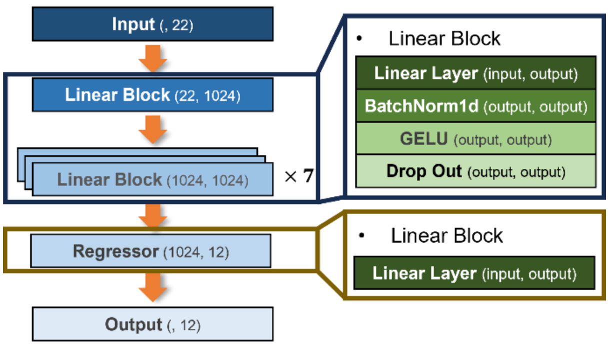

Fig. 1. Schematic Diagram of Baseline Model Consists of Fully Connected Layers

2.3.2 Train and Inference Phases

Table III. Hyperparameters of Baseline Model

Batch Size	32
Learning Rate	1e-4 w/ Cosine Annealing Warm Restarts
Optimizer	AdamW
Criterion	Mean Squared Error

Some hyperparameters used in the train phase of baseline model are shown in Table III. In train phase, the thermal hydraulic variables, component states (binary type), and mitigation strategy implementation states (binary type) at time t are input to the baseline model, the model outputs the thermal hydraulic variables at time $t+1$. Then, the loss between the predicted thermal hydraulic variables and the thermal hydraulic variables calculated with MAAP is calculated as the mean squared error, and the weights and biases of the surrogate model are updated in the direction of decreasing this loss using AdamW optimizer with designated learning rate scheduler [16, 17]. Therefore, baseline model is Single Input Single Output (SISO) model [18]. Training process of a SISO model is shown in Figure 2. The loss function used in the train phase of the baseline model is mean squared error (MSE) as follows.

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}(\mathcal{F}_{\theta}(x_i) - y_i)^2$$

In inference phase, the model uses autoregressive method as shown in Figure 3, the continuous inference processes of the SISO model. When the thermal hydraulic variables, components failure states, and mitigation strategies implementation states at the first time retrieved from the MAAP dataset are input to the model, the model infers the thermal hydraulic variables of the next time step as learned. Then, the components failure states, mitigation strategies implementation states, and inferred thermal hydraulic variables are merged and used as input to the model, and this inference-concatenation process is repeated until the time step reaches 72 hours.

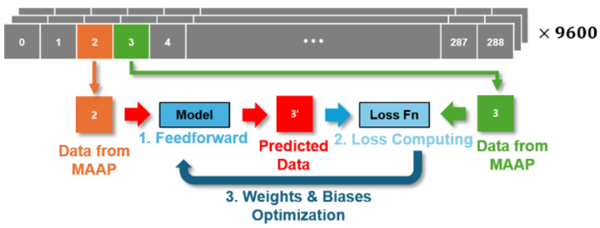

Fig. 2. Training process of Baseline Model (SISO type). The superscript indicates the value predicted by the surrogate model.

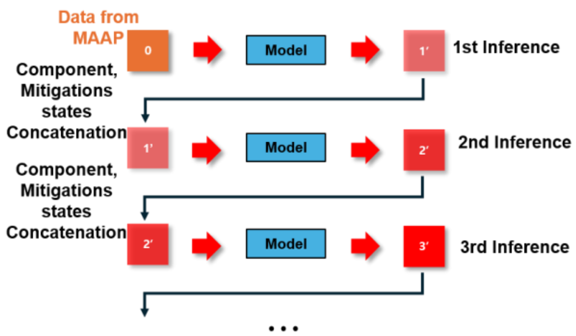

Fig. 3. Continuous inference process of Baseline Model (SISO type). The superscript indicates the value predicted by the surrogate model.

The surrogate model is engineered to deduce information gradually, focusing on individual time steps, rather than attempting to predict thermal hydraulic variables for extended periods like 72 hours. This approach is necessitated by the dynamic nature of reinforcement learning, which aims to optimize mitigation strategies while considering real-time fluctuations in components failures and strategies implementation. Adopting this method ensures effective interaction between the surrogate model and the reinforcement learning agent, which would otherwise be hindered if attempting to process 72 hours' worth of data in a single prediction (multiple output). This single output method was used for the same reason in the other surrogate model that will be introduced later.

2.4 Convolutional Neural Network-based Model

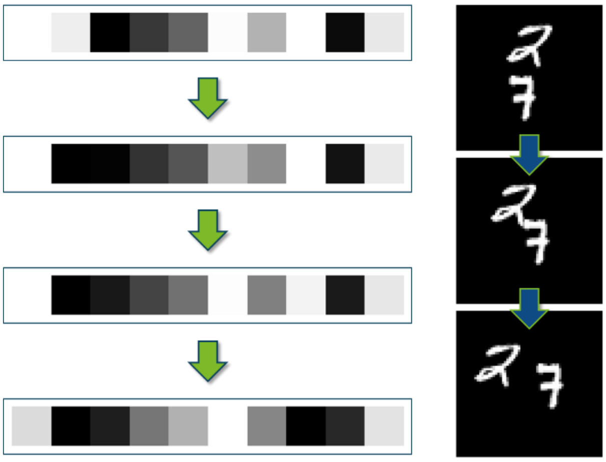

Fig 4. Multivariate time series example (left) and Moving MNIST example (right), used as a demonstration in video prediction [19].

Video prediction and multivariate thermal hydraulic time series forecasting of a NPP are similar in that they deal with spatio-temporal data. Various thermal hydraulic variables of a NPP affect each other (spatial domain) and change over time (temporal domain), so it can be said to be spatio-temporal data. In the context of video prediction, it entails forecasting a series of forthcoming frames by analyzing previous input frames [18]. In one frame, each pixel has a spatial context with surrounding pixels, and temporal contexts exist between frames. Therefore, predicting thermal hydraulic variables, and predicting video involve learning representations or features that capture both spatial and temporal information from sequences. In other words, the time series used as input in this study can have the dimensions of batch, sequence length, channel, height, and width, just like video. These similarities motivated us to appropriately improve deep learning architectures for video prediction and suit them for continuous inference of multivariate time series.

2.4.1 Macro Design

In the domain of video prediction, the prevailing architectural framework involves an encoder, translator, and decoder structure [18, 20]. The encoder captures features from observed frames, and transmits them to the translator, which produces prospective features. Subsequently, the decoder interprets these prospective features to reconstruct the forthcoming frame [18]. This method of allocating spatial information processing to the encoder and decoder and temporal information processing to the translator has become a common framework in video prediction because it shows high performance [21]. As this architecture is state-of-the-art in the field of video prediction, it was also adopted as a macro structure in this study. The overall structure of the CNN-based model used in this study is shown in Figure 5. Looking at the overall structure, the encoder, translator, and decoder blocks are made up of several ConvNeXt blocks [22]. In the encoder and decoder, 4 ConvNeXt blocks are connected in series, and in the translator, 7 ConvNeXt blocks are connected in series. The encoder and decoder are connected through long skip connections, and ConvNeXt blocks in the translator are also designed to be connected to each other through long skip connections [23]. In front of the encoder, there is a data captor block that accepts data, and behind the decoder, there is a readout block that outputs thermal hydraulics variables of the next time step.

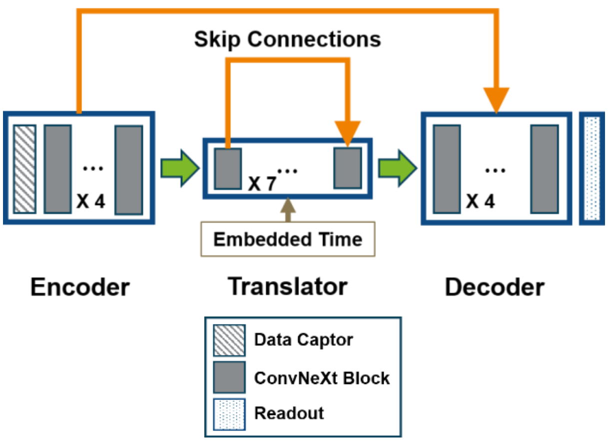

Fig. 5. Macro Structure of CNN-based Surrogate Model

Unlike the ConvNeXt blocks that make up the encoder and decoder, the ConvNeXt blocks that make up the translator are not only connected through Skip Connections, but also input the embedded prediction time that passes through the fully connected layers. The approach, drawing inspiration from [24] and implemented in [18], was termed the 'implicit architecture,' involving the embedding of the target time step and its subsequent input to the translator. As per [18], optimal performance entails establishing distinct neural networks for predicting each target time step. However, implementing such a method in this study would necessitate the creation of up to 288 neural networks to process a 72-hour scenario. Hence, in [18], an alternative approach was adopted, consolidating multiple neural networks into a singular entity. This was achieved by incorporating the embedded time as an input to the translator and adjusting the neural network's weights accordingly. This same methodology was also employed in the present study.

2.4.2 Micro Design

Data Captor

As explained in 2.4, a multivariate time series is a unity-height video, that is, a video with frames in which pixels are stretched only horizontally. Data captor block was designed with the intention of creating a height dimension like a typical video so that the kernel can move up and down as well as left and right during the convolution operation to obtain more spatial dimension information. This block consists of two stages: [Pointwise Conv2D, Pixel Shuffle]. First, Pointwise Conv2D layer increases the channels of the time series [25]. Afterwards, the Pixel Shuffle layer converts the stretched channels to the height dimension, turning the horizontally elongated time series into a frame of video with height and width. The functioning of pixel shuffling is detailed in the reference [26]. Figure 6 provides a simple diagram of how this block works.

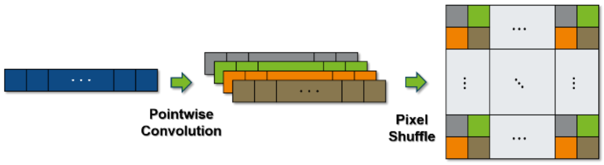

Fig. 6. Mechanism of data captor block

ConvNeXt Block

ConvNeXt is modernized version of ResNet [22]. ConvNeXt improves performance compared to ResNet by introducing some transformer design decisions. The structure of the ConvNeXt block used in this study is shown in Figure 7. In this study, ConvNeXt blocks were arranged in series to process temporal or spatial information.

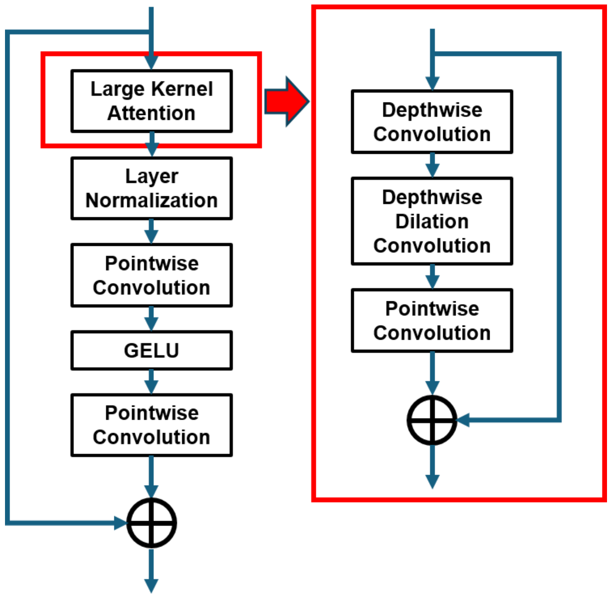

Fig. 7. Schematic of ConvNeXt Block

Large kernel attention (LKA) is based on the belief that the human visual system selectively processes certain stimuli in detail while allocating less processing resources to others, resulting in partial processing of potential visual inputs [27]. Based on this belief, LKA extracts some features from a relatively large kernel. The combination of depthwise convolution and pointwise convolution to reduce the computational amount of existing convolution is described in reference [28].

Readout

The readout block comprises bundles of [Linear, GELU, Layer Normalization, and Dropout] [29]. The readout block plays a role in adjusting the shape so that it can output the thermal hydraulic variables and RPV status of the next future step.

2.4.3 Add noise to data

Before the input data was injected into the model, Gaussian noise with a mean of 0, and a standard deviation varying from 0 to 0.03 was randomly added. The method of improving the generalization performance of a model by training the model with data with added noise has already been proven in [30, 31, 32]. The figure 8 shows the data before adding noise and after adding noise with a standard deviation of 0.03.

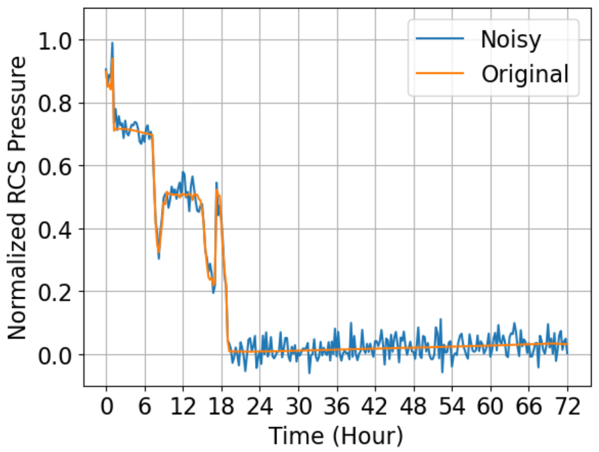

Fig. 8. Normalized RCS pressure example and noise added data

2.4.4 Train and Inference Phases of CNN-Based Model

Table IV. Hyperparameters used in CNN-based Model

Batch Size	32
Learning Rate	1e-4 w/ Cosine Annealing Warm Restarts
Optimizer	AdamW
Criterion	Huber Loss, Binary Cross Entropy Loss

The hyperparameters employed during the training phase of the CNN-based model are detailed in Table IV. Unlike the baseline model (SISO Model), in the CNN-based model, the thermal hydraulic variables, component failure states, and mitigation strategies implementation states of the previous six time steps are simultaneously injected into the model. The model then outputs the thermal hydraulic variables for the next single time step. Afterwards, it goes through an optimization process of weights and biases like the SISO model. Therefore, this model can be said to be a Multiple Input Single Output (MISO) model (The necessity for a singular output was elucidated in section 2.3.2.). Figure 9 schematically illustrates the training process of the MISO model. For weights and biases update, Huber loss was used for thermal hydraulics variables losses, and binary cross entropy (BCE) was used for RPV status loss which is indicated as 0 or 1 [33].

$$Huber\ Loss = \begin{cases} \frac{1}{N}\sum_{i=1}^{N}\frac{1}{2}(\mathcal{F}_{\Theta}(x_i)-y_i)^2 \ \ for\ |\mathcal{F}_{\Theta}(x_i)-y_i| \le \delta, \\ \frac{1}{N}\sum_{i=1}^{N}\delta\cdot\left(|\mathcal{F}_{\Theta}(x_i)-y_i|-\frac{1}{2}\delta\right), otherwise. \end{cases}$$

$$BCE = -\frac{1}{N}\sum_{i=1}^{N}[\mathcal{F}_{\Theta}(x_i)\cdot log\, y_i + \left(1-\mathcal{F}_{\Theta}(x_i)\right)\cdot log(1-y_i)]$$

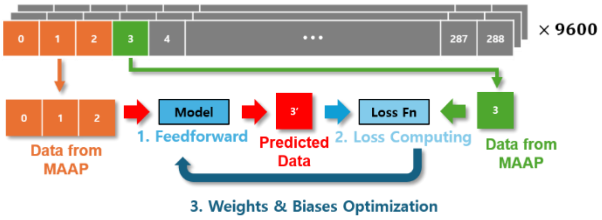

Fig. 9. Training process of CNN-Based Model (MISO type). The superscript indicates the value predicted by the surrogate model.

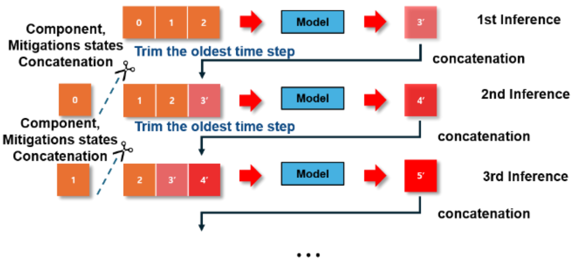

Fig. 10. Continuous inference process of CNN-based Model (MISO type). The superscript indicates the value predicted by the surrogate model

Figure 10 illustrates the continuous inference procedure of the CNN-based model. In the figure 10, the autoregressive continuous inference approach was employed, mirroring the methodology of the baseline model. However, as a MISO model, at the beginning of inference, six time steps were simultaneously injected into the model, resulting in one time step in the future. Just because six steps are injected at once at the start of inference does not mean that there is any difference in the amount of information the baseline model receives at the start of inference. This is because the first six steps that the CNN-based model takes during inference are in a steady state.

Since the input shape of the model is fixed, to continue inference at the next time step, the newly inferred output was concatenated with the component failure states and mitigation strategies implementation states at that time. Then the oldest time step is removed from the existing input, and the existing input is merged with newly inferred, (and concatenated) output. As with the baseline model's inference phase, the same procedure is repeated until the time step reaches 72 hours.

2.5 Evaluation Measure

In the realm of time series analysis, dynamic time warping (DTW) serves as an algorithm for assessing the resemblance between two sequential temporal datasets [34]. The Sakoe-Chiba band reduces computational cost by limiting comparison operations in DTW. This band defines a diagonal band of constant width, and comparisons are performed only within this band at each time step. This reduces computational cost by focusing the comparison on areas where the two time series are likely to match each other.

However, the useful thing about the Sakoe-Chiba band is that the DTW distance increases when the time alignment of the two time series is excessively misaligned. In the case of unconstrained DTW, even if the time alignment of the two time series is too different, if the shape is similar, the distance is low. Therefore, in the application of this study, where both the trend and value of the time series are important, it is more advantageous to introduce the Sakoe-Chiba band. [35]

3. Results and Discussion

To evaluate the models' performances, continuous inference was performed using a test set that models did not encounter during training and validation. The test set consists of 1200 72-hour scenarios, and as explained in the methodology for continuous inference, only the first part of the time series, component failure states, and mitigation strategies implementation states are input to the models to infer thermal-hydraulic variables. For the baseline model, the inference time was about 6 seconds on an NVIDIA® A100 GPU to infer 1200 72-hour scenarios, and for the CNN-based model, it took about 5 minutes in the same environment. Then, DTW was used to measure the similarity between the predicted values and the values of the test dataset.

3.1 Continuous Inference Results from Baseline Model

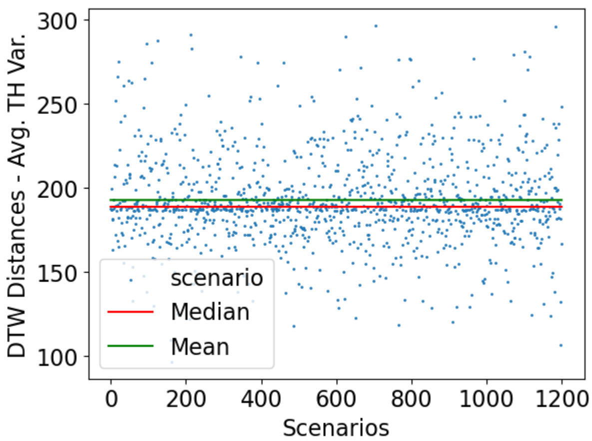

Fig. 11. Scatter plot of the average value of the DTW distance between the thermal hydraulic variables predicted by the baseline model and the thermal hydraulic variables computed with MAAP.

Table V. Representative values of DTW distances between values predicted by baseline model and values computed by MAAP.

Max.	Min.	Median	Mean	Std.
296.41	96.38	192.60	188.57	26.77

Figure 11 shows a scatter plot of the average value of the DTW distance between the thermal hydraulic variables of the 1200 scenarios predicted by the baseline model and the thermal hydraulic variables of the scenarios computed with MAAP. Additionally, Table V shows the maximum, minimum, median, average, and standard deviation of the DTW distances for the scenarios predicted by the baseline model.

The scenario closest to the median of the average has the component failure states and mitigation strategies implementation states as shown in the following table VI. Figures 12 to 15 show the continuous inference results and MAAP calculation results of the thermal hydraulic variables of the scenario closest to the median value of the average.

Table VI. Component Failure States and Mitigation Strategies Implementation States from the Scenario Closest to the Median of the Average. '×' indicates that it did not occur.

RCP Seal Fail	Hx Fail	HPSI Fail	LPSI Fail	CSS Fail
1	×	7	×	×
MDAFW Fail	CP Fail	Mit 1	Mit 2	Mit 3
×	40	×	×	×

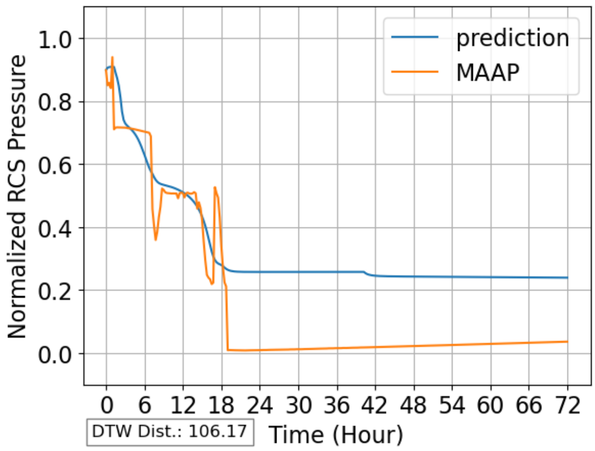

Fig. 12. Baseline model inference result and MAAP calculation result for Normalized reactor coolant system pressure for average DTW distance scenario.

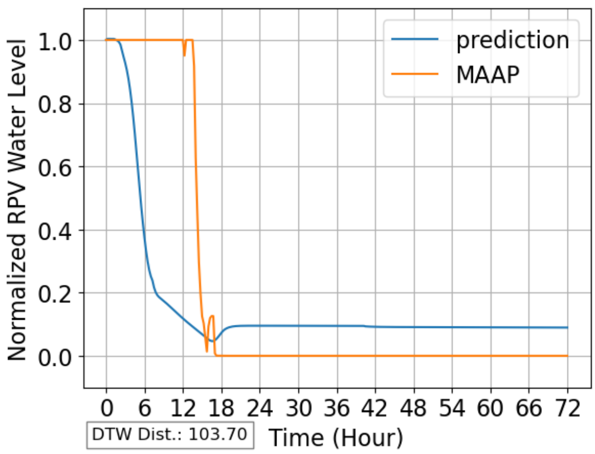

Fig. 13. Baseline model inference result and MAAP calculation result for Normalized RPV water level for average DTW distance scenario.

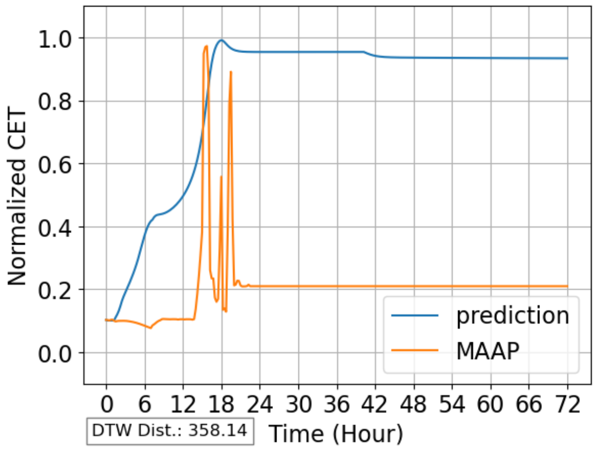

Fig. 14. Baseline model inference result and MAAP calculation result for Normalized maximum core exit temperature for average DTW distance scenario.

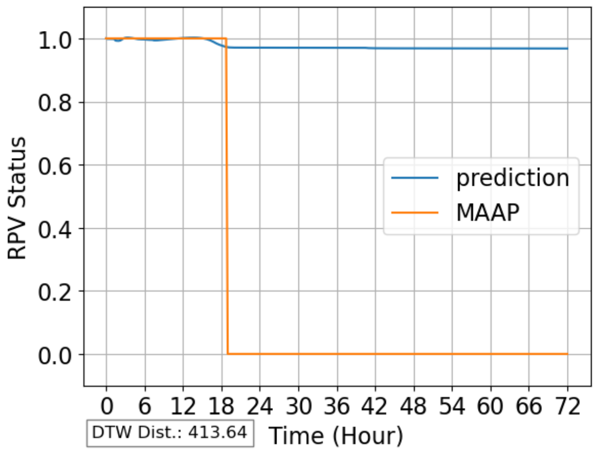

Fig. 15. Baseline model inference result and MAAP calculation result for RPV Status for average DTW distance scenario. 0 indicates RPV failure, 1 indicates RPV did not fail.

3.2 Continuous Inference Results from CNN-Based Model

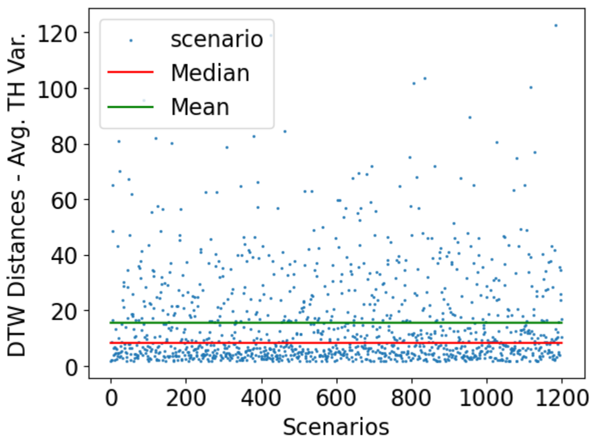

Figure 16. Scatter plot of the average value of the DTW distance between the thermal hydraulic variables predicted by the baseline model and the thermal hydraulic variables computed with MAAP.

Table VII. Representative values of DTW distances between values predicted by CNN-based model and values computed by MAAP.

Max.	Min.	Median	Mean	Std.
122.71	1.69	8.19	15.45	17.14

Figure 16 depicts a scatter plot illustrating the mean DTW (Dynamic Time Warping) distance between the thermal hydraulic variables of 1200 scenarios predicted by the CNN-based model and those computed using MAAP. Furthermore, Table VII provides statistical summaries including the maximum, minimum, median, mean, and standard deviation of DTW distances for scenarios predicted by the CNN-based model.

The scenario with thermal hydraulic variables closest to the median of the average is accompanied by a description of component failure states and implemented mitigation strategies, outlined in Table VIII. In Table VIII, the mitigation strategies are being implemented after the RPV failure. This is due to the following characteristics of the dataset constructed in this study. The surrogate model will be used in reinforcement learning later, and the reinforcement learning agent uses mitigation strategies at random times and gradually optimizes the use time. In order to implement the initial random mitigation strategies implementation time, in-vessel mitigation strategies were implemented after RPV failure. Figures 17 to 20 present the continuous inference outcomes and MAAP calculations pertaining to the thermal hydraulic variables of this particular scenario, positioned nearest to the median average value.

Table VIII. Component Failure States and Mitigation Strategies Implementation States from the Scenario Closest to the Median of the Average. '×' indicates that it did not occur.

RCP Seal Fail	Hx Fail	HPSI Fail	LPSI Fail	CSS Fail
1	1	15	×	×
MDAFW Fail	CP Fail	Mit 1	Mit 2	Mit 3
19	50	29	×	49

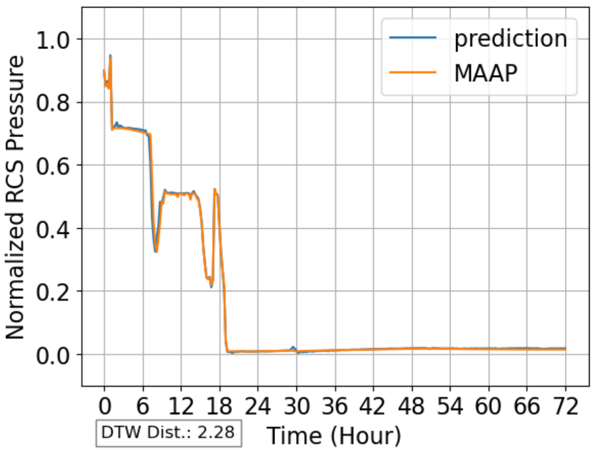

Fig. 17. CNN-based model inference result and MAAP calculation result for Normalized reactor coolant system pressure for average DTW distance scenario.

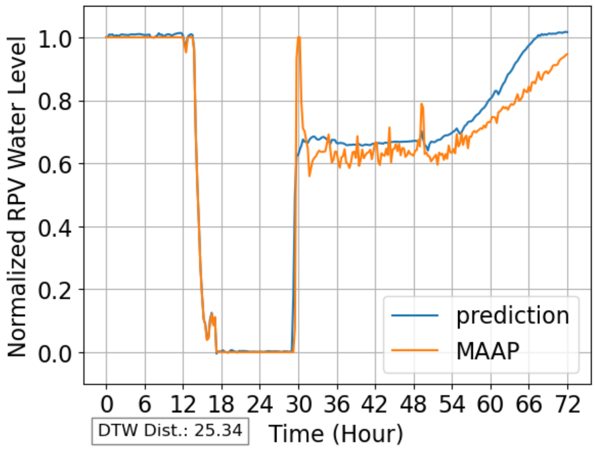

Fig. 18. CNN-based model inference result and MAAP calculation result for Normalized RPV water level for average DTW distance scenario.

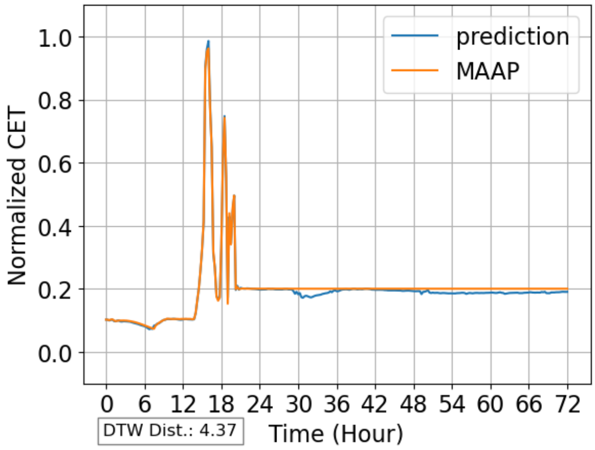

Fig. 19. CNN-based model inference result and MAAP calculation result for Normalized maximum core exit temperature for average DTW distance scenario.

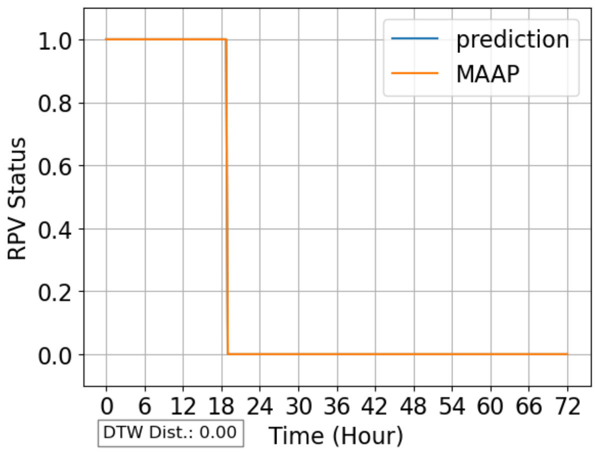

Fig. 20. CNN-based model inference result and MAAP calculation result for RPV Status for average DTW distance scenario. 0 indicates RPV failure, 1 indicates RPV did not fail.

3.3 Discussion

First, in section 3.1., Overall, it can be seen that the predicted results of thermal hydraulic variables and the MAAP calculation results do not match at all. Although quantitative analysis was not published in this paper, it is assumed that the reason these non-physical results were derived from the baseline model was because the following Markov property was introduced into the baseline model.

$$P(X_{t+n} = x | X_t, X_{t-1}, \cdots, X_{t-k}) = P(X_{t+n} = x | X_t)$$

As shown in the equation above, the Markov property is a stochastic process in which the past and future are independent of each other, conditional on the present. The continuous inference process of the baseline model can be said to be a Markov process because only the current input values affect the values of the next time step. In the training process of the baseline model, only the values of the variables were input and the model did not learn the change trends of the variables, so it is believed that the inference results do not match the MAAP calculation results.

On the other hand, in the case of the CNN-based model in which six time steps were input simultaneously, this Markov property was somewhat relaxed, so it appears to have shown relatively better inference results compared to the baseline model.

Originally, the ConvNeXt technique was used for object edge detection, video prediction, etc., but it was confirmed to work well for predicting multivariate time series. Although the recent trend in deep learning model development is to remove bias, CNN has been confirmed to perform better than the fully connected layer model if applied to an appropriate dataset due to its strong inductive bias.

4. Conclusions

The purpose of this study is to quickly and accurately predict thermal hydraulic variables and RPV failure points based on artificial neural networks, replacing the existing integrated severe accident analysis codes, which has relatively high computational costs. As a baseline model, a fully connected layer SISO model and a MISO model inspired by a video prediction model were developed, and the performance of surrogate models was quantified using DTW, a measure of the similarity of time series. The performance of the MISO model, which injected inductive biases to reflect the characteristics of the time series, was shown to be improved compared to the performance of the baseline model.

Artificial neural network-based surrogate models are fast, but (1) they attempt extrapolation in areas where data learning is insufficient, and (2) they select only specific thermal hydraulic variables for learning, so their accuracy is bound to be less accurate than existing severe accident computational codes. This inaccuracy leads to unphysical results in the predictions of the surrogate model, and as a result, the reliability of optimizing a severe accident management strategies based on this inevitably decreases.

As a future work, we would like to introduce a physics-informed neural network methodology to the surrogate model to prevent the surrogate model from producing non-physical results. Subsequent research will continue to attempt to maintain the inference speed of the surrogate model and increase accuracy by injecting physical constraints consisting of several mathematical models into the neural network that makes up the surrogate model.

Acknowledgement

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT: Ministry of Science and ICT) (No. RS-2022-00144202). This work was also supported by the Innovative Small Modular Reactor Development Agency grant funded by the Korea Government (MSIT) (No. RS-2023-00259516).

REFERENCES

[1] Lee et al., "Applications of Neural Network to Predict Reactor Vessel Failure Time for Various Component Failures during Severe Accident", *Transactions of the Korean Nuclear Society Spring Meeting*, Jeju, Korea, May 19-20, 2022.

[2] Song et al., "Feasibility Estimation of Development of Real-time Interactive Nuclear Accident Simulation Using Supervised Learning", *Transactions of the Korean Nuclear Society Spring Meeting*, Jeju, Korea, May 19-20, 2022.

[3] Lee et al., "Applications of Supervised Machine Learning to Diagnose Reactor Vessel Failure", *Transactions of the Korean Nuclear Society Autumn Meeting*, Changwon, Korea, October 20-21, 2022.

[4] S. H. Song and J. I. Lee, "Applying Supervised Learning Algorithm to Nuclear Power Plant Severe Accident Data Generated from MAAP Code", *Transactions of the Korean Nuclear Society Autumn Meeting*, Changwon, Korea, October 20-21, 2022.

[5] J. Y. Bae and S. J. Kim, "Applicability Study of Deep Reinforcement Learning to Severe Accident Analysis", *Transactions of the Korean Nuclear Society Spring Meeting*, Jeju, Korea, May 18-19, 2023.

[6] Lee et al., "Inferring Severe Accident Scenarios in Nuclear Power Plants with Reinforcement Learning (RL) and Supervised Learning (SL) Approaches: Part 1 SL Development", *Transactions of the Korean Nuclear Society Spring Meeting*, Jeju, Korea, May 18-19, 2023.

[7] Song et al., "Inferring Severe Accident Scenarios in Nuclear Power Plants with Reinforcement Learning (RL) and Supervised Learning (SL) Approaches: Part 2 RL Development", *Transactions of the Korean Nuclear Society Spring Meeting*, Jeju, Korea, May 18-19, 2023.

[8] Joo et al., "Inferring Severe Accident Scenarios in Nuclear Power Plants with Reinforcement Learning (RL) and Supervised Learning (SL) Approaches: Part 3 Sensitivity of RL to SL" *Transactions of the Korean Nuclear Society Spring Meeting*, Jeju, Korea, May 18-19, 2023.

[9] Lee et al., "Detecting Core Uncovery with Limited Information during Severe Accident Using Machine Learning

Methods”, *Transactions of the Korean Nuclear Society Autumn Meeting*, Gyeongju, Korea, October 26-27, 2023.

[10] Song et al., “Utilizing Artificial Neural Networks to Forecast Remaining Time to Reactor Pressure Vessel Failure during Severe Accident”, *Transactions of the Korean Nuclear Society Autumn Meeting*, Gyeongju, Korea, October 26-27, 2023.

[11] Joo et al., “Accelerated Prediction of Severe Accident Progression: Sensitivity of Deep Neural Network Performance to Time Resolution”, *Transactions of the Korean Nuclear Society Autumn Meeting*, Gyeongju, Korea, October 26-27, 2023.

[12] KAERI., “Development of Site Risk Assessment & Management Technology including Extreme External Events – Development of the Integrated Risk Assessment Technology for Multiple Units”, KAERI/RR-3924/2014, KAERI, 2015.

[13] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in International conference on machine learning, 2015: pmlr, pp. 448-456.

[14] D. Hendrycks and K. Gimpel, "Gaussian error linear units (gelus)," arXiv preprint arXiv:1606.08415, 2016.

[15] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," arXiv preprint arXiv:1207.0580, 2012.

[16] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," arXiv preprint arXiv:1711.05101, 2017.

[17] I. Loshchilov and F. Hutter, "Sgdr: Stochastic gradient descent with warm restarts," arXiv preprint arXiv:1608.03983, 2016.

[18] M. Seo, H. Lee, D. Kim, and J. Seo, "Implicit Stacked Autoregressive Model for Video Prediction," arXiv preprint arXiv:2303.07849, 2023.

[19] N. Srivastava, E. Mansimov, and R. Salakhudinov, "Unsupervised learning of video representations using lstms," in International conference on machine learning, 2015: PMLR, pp. 843-852.

[20] Z. Gao, C. Tan, L. Wu, and S. Z. Li, "Simvp: Simpler yet better video prediction," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 3170-3180.

[21] C. Tan et al., "Temporal attention unit: Towards efficient spatiotemporal predictive learning," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 18770-18782.

[22] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2022, pp. 11976-11986.

[23] M. Drozdzal, E. Vorontsov, G. Chartrand, S. Kadoury, and C. Pal, "The importance of skip connections in biomedical image segmentation," in International Workshop on Deep Learning in Medical Image Analysis, International Workshop on Large-Scale Annotation of Biomedical Data and Expert Label Synthesis, 2016: Springer, pp. 179-187.

[24] V. Sitzmann, J. Martel, A. Bergman, D. Lindell, and G. Wetzstein, "Implicit neural representations with periodic activation functions," Advances in neural information processing systems, vol. 33, pp. 7462-7473, 2020.

[25] B.-S. Hua, M.-K. Tran, and S.-K. Yeung, "Pointwise convolutional neural networks," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 984-993.

[26] W. Shi et al., "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 1874-1883.

[27] M.-H. Guo, C.-Z. Lu, Z.-N. Liu, M.-M. Cheng, and S.-M. Hu, "Visual attention network," Computational Visual Media, vol. 9, no. 4, pp. 733-752, 2023.

[28] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 1251-1258.

[29] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," arXiv preprint arXiv:1607.06450, 2016.

[30] Z. Hussain, F. Gimenez, D. Yi, and D. Rubin, "Differential Data Augmentation Techniques for Medical Imaging Classification Tasks," AMIA ... Annual Symposium proceedings. AMIA Symposium, vol. 2017, pp. 979-984, 04/16 2018.

[31] I. Goodfellow, Y. Bengio, and A. Courville, Deep learning. MIT press, 2016.

[32] S. Huq, P. Xi, R. Goubran, J. J. Valdés, F. Knoefel, and J. R. Green, "Data Augmentation using Reverb and Noise in Deep Learning Implementation of Cough Classification," in 2023 IEEE International Symposium on Medical Measurements and Applications (MeMeA), 14-16 June 2023 2023, pp. 1-6, doi: 10.1109/MeMeA57477.2023.10171862.

[33] P. J. Huber, "Robust Estimation of a Location Parameter," The Annals of Mathematical Statistics, vol. 35, no. 1, pp. 73-101, 29, 1964. [Online]. Available: https://doi.org/10.1214/aoms/1177703732.

[34] D. J. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series," presented at the Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining, Seattle, WA, 1994.

[35] Z. Geler, V. Kurbalija, M. Ivanović, M. Radovanović, and W. Dai, "Dynamic Time Warping: Itakura vs Sakoe-Chiba," in 2019 IEEE International Symposium on INnovations in Intelligent SysTems and Applications (INISTA), 3-5 July 2019 2019, pp. 1-6, doi: 10.1109/INISTA.2019.8778300.