International Journal on
Advanced Science
Engineering
Information Technology

Vol.14 (2024) No. 3
ISSN: 2088-5334

Time-Series Data Augmentation for Improving Multi-Class Classification Performance

Woo-Hyeon Kim [a], Geon-Woo Kim [a], Jaeyoon Ahn [b], Kyungyong Chung [a,*]

[a] *Division of AI Computer Science and Engineering, Kyonggi University, Suwon, 16227, Republic of Korea*
[b] *Seoyoung Engineering Co., Ltd: Seongnam, 13595, Republic of Korea*
*Corresponding author: *dragonhci@gmail.com*

Abstract— **This paper proposes a new approach to classify and evaluate defects in concrete structures automatically. To overcome the limitations of defect detection methods that traditionally relied on expert visual observation, the reflection signal of electromagnetic pulses is extracted as time-series data and used to analyze the propagation characteristics of each defect. This study uses deep learning models to analyze these time-series data and classify defects. Since anomaly detection data has more normal data than anomaly data, data augmentation methods such as Time Warping, Noise Injection, Smoothing, Trend Shifting, etc., were applied to solve the problem of data imbalance and overfitting. Among them, Noise Injection showed the best performance. The generalization performance of the proposed method was evaluated through performance evaluation using LSTM, GRU, and TCN models, and LSTM models showed the highest performance. The study results show that the proposed method effectively classifies defect types in concrete structures and can solve the limitations of existing methods by automatic classification through deep learning models. In addition, it was confirmed that the model's performance could be improved by improving the amount and diversity of data by selecting and applying appropriate data augmentation methods. The contribution of the research is to present a new approach that automates the defect detection and classification task of concrete structures and provides high accuracy and efficiency.**

Keywords—**Time-series; data augmentation; multi-class classification; deep learning; signal reflection.**

Manuscript received 15 Nov. 2023; revised 14 Jan. 2024; accepted 22 Feb. 2024. Date of publication 30 Jun. 2024.

I. Introduction

Concrete structures can develop cracks over time or absorb moisture, potentially harming human life and property damage. Therefore, assessing the safety and durability of concrete structures and classifying types of defects are essential aspects of the construction field. Traditionally, experts primarily employed visual inspection methods to detect defects [1]. However, this approach requires a high level of expertise and experience and is disadvantageous due to its high cost and time consumption. Remarkably, the time required increases dramatically when many defects are detected compared to the limited number of experts available. This paper aims to overcome these limitations by extracting the propagation characteristics of each defect in concrete structures using the reflected signals of electromagnetic pulses as time-series data [2]. Subsequently, deep learning models are utilized to analyze the propagation characteristics of concrete structures and classify defects [3].

Time-series data, characterized by its continuous nature, is observed chronologically. Multiclass classification based on time-series data is employed in various fields, including finance, healthcare, weather forecast, and energy [4], [5], [6], [7]. As sensors and data-related tools advance and more data measurement systems are introduced, the importance of time-series data analysis is increasing with the growing volume of data [8]. However, in the real world, everyday situations occur more frequently than abnormal ones in time-series data, leading to a class imbalance where data for all classes are not equally represented. This can cause overfitting and lower the performance of models. To address this, the study employs time-series data augmentation techniques such as Time Warping, Noise Injection, Smoothing, and Trend Shifting to alleviate class imbalance and prevent overfitting while processing the data with deep learning models [9], [10], [11], [12], [13], [14]. Each augmentation method's analysis and performance evaluation reveal that noise injection is the most effective. Additionally, the performance of LSTM, GRU, and TCN models is evaluated to assess the generalization capability of the proposed method [15], [16], [17]. This approach aids in determining the type of defects in concrete and assessing the structure's safety and durability. The contributions of the proposed method are as follows:

- Automatically classifying concrete structure defects using deep learning models, thereby overcoming the limitations of traditional defect classification methods.
- The time-series data augmentation methods process insufficient or biased data for deep learning models, providing reliable results in detecting defects in concrete structures.
- The time-series data augmentation techniques used in this paper achieve significant performance improvements and can be applied to classify time-series data based on sensor data.

The structure of this paper is as follows:

- Section II describes the collection of time-series data, time-series data augmentation methods, and model training.
- Section III compares performance using the data augmentation methods and generalization performance comparison through LSTM, GRU, and TCN models.
- Finally, section IV provides the conclusion of this paper.

II. Materials and Method

The proposed method consists of three stages. The first stage involves collecting time-series data on defects in concrete structures. In the second stage, four time-series data augmentation techniques are applied: Time Warping, Noise Injection, Smoothing, and Trend Shifting for analysis. The third stage involves using the LSTM model to perform multi-class classification of the time-series data that was augmented in the second stage [18]. Fig. 1 shows a process of time-series data augmentation methods.

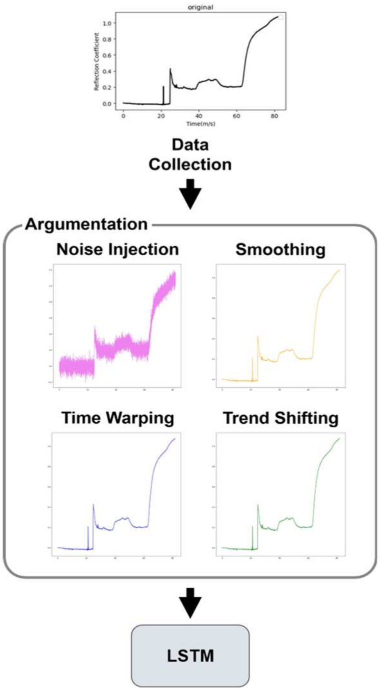

Fig. 1 Process of Time-Series Data Augmentation for Improving Multi-Class Classification Performance

A. Collection of Concrete Structure Defect Data

The data used in this study consists of time-series data obtained by intentionally creating defects in concrete structures and then capturing the characteristics of each defect through the reflected signals of electromagnetic pulses. The reflection patterns of electromagnetic waves passing through concrete and those passing through air, soil, and water exhibit distinct differences. Particularly notable are the variations in reflection patterns in soil, which are significantly influenced by its moisture content, i.e., the volume-moisture ratio. Accordingly, the defects are classified into eight categories.

The categories include Dry, representing completely oven-dried sand samples; Water, indicating the presence of water; and Void, signifying empty spaces. For defects caused by moisture, the volume-moisture ratio is used. This ratio is calculated as the water volume divided by the soil sample's total volume. Consequently, the data is classified based on volume moisture ratios of 6%, 12%, 18%, 24%, and 30%, labeled as S6, S12, S18, S24, and S30, respectively [19], [20]. Fig. 2 presents a visualization of the concrete structure defect data, illustrating these variations.

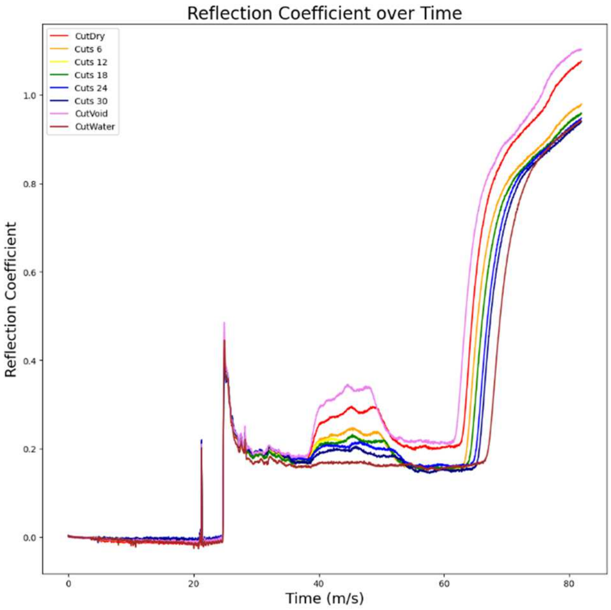

Fig. 2 Visualization Results of Concrete Structure Defect Data

B. Augmentation of Concrete Structure Defect Data

The time-series data collected in the previous phase predominantly comprises normal conditions instead of abnormal ones. Moreover, occurrences are not evenly distributed across all classes, resulting in class imbalance. This imbalance can lead to overfitting and subsequently lower the model's performance. Therefore, data augmentation is performed at this stage to address these issues. Data augmentation helps evenly distribute data across all classes and increase the data representing abnormal conditions, thereby adjusting the ratio with standard condition data. This process prevents overfitting during the model training phase and enhances the model's overall performance. Four data augmentation methods are applied and analyzed in this stage.

The first augmentation method is time-warping [21]. The equation applied to time-warping is as shown in Equation (1).

$$x'_t = x_{\varphi(t)} \quad (1)$$

In the augmented data, x'_t represents the value at time t. $x_{\phi(t)}$denotes the value of the original data at a time $\phi(t)$. Here, $\phi(t)$ is a function that transforms time, either stretching or compressing the time axis. This method involves dividing the entire dataset into different n segments and expanding them, thus aiding the model to learn better about variations in the time axis. However, a limitation of this approach is the potential increase in featureless data due to the uniform augmentation across all segments.

The second augmentation method is Noise Injection. This method preserves the characteristics of the original data while adding natural noise throughout the dataset. The equation applied for Noise Injection is described in Equation (2).

$$x' = x + N(0,\sigma^2) \quad (2)$$

The equation for Noise Injection is as follows: x represents the original data, and x' is the data with added noise [23]. $N(0,\sigma^2)$ signifies random noise generated from a normal distribution with mean 0 and variance σ^2. Through the noise injection process, the model can learn about various patterns. However, excessive noise addition interferes with the model's learning. In addition, a limitation adversely affects learning if it is augmented similarly to the class data that must be classified when noise is added.

The third augmentation method is Smoothing [22]. This technique reduces outliers in the data and smoothens the overall trend. In this study, the Smoothing technique employed is the Simple Moving Average (SMA), and its equation is described in Equation (3).

$$SMA_t = \frac{1}{N}\sum_{i=t-N+1}^{t} x_i \quad (3)$$

Here, N is the number of data points used to calculate the average, and x_i is the data value at time i. This process helps to reduce the effects of irregular changes in the data and increases its stability. However, there is a limitation in that excessive augmentation can erase the data's characteristics, hindering the model's learning.

The fourth augmentation method is Trend Shifting. This method increases the diversity of data trends by moving the time axis as specified by the user or changing the numerical value of the data as a whole. The equation for Trend Shifting is described in Equation (4).

$$x'_t = x_t \pm \Delta t \quad (4)$$

In Trend Shifting, x'_t represents the shifted data value at time t, and x_t is the value of the original data at time t. Here, Δt is the magnitude of the time shift, which can be either forward (+) or backward (-). This process assists the model in learning about various changes and trends. However, there is a limitation, as the reduction in data variability can lead to overfitting. Fig. 3 shows the four time-series data augmentation methods for enhancing multi-class classification performance.

Each graph in Fig. 3 represents the data before and after augmentation. The black dotted lines in these graphs depict the original data before augmentation, while solid lines in various colors represent the data post-augmentation. All the graphs are generated by augmenting data from the Dry class.

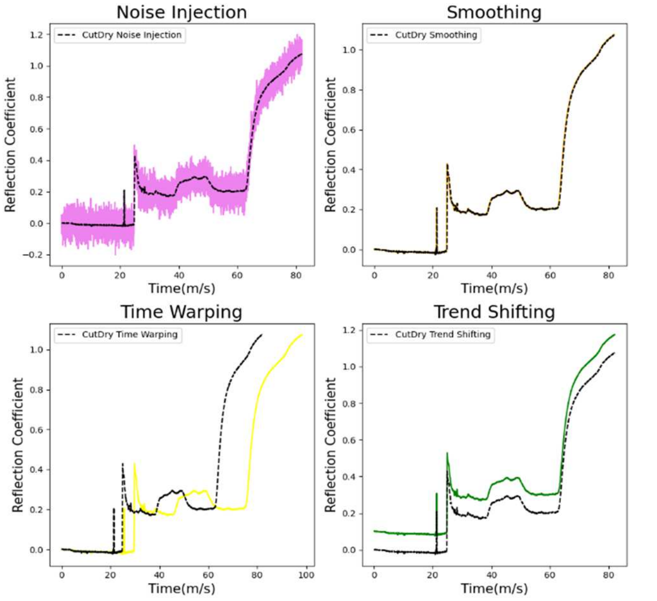

Fig. 3 Application of Four Time-Series Data Augmentation Methods for Enhancing Multiclass Classification Performance

In Fig. 3 Noise Injection graph, we can observe that the graph maintains its trend while exhibiting added noise, creating a noticeable fluctuation compared to the original data. This fluctuation can be controlled by adjusting the level of Gaussian noise. However, a limitation exists: adding too much noise can lead to excessive fluctuation, potentially hindering the learning process. Therefore, it's crucial to add an appropriate level of noise. The model is trained with various noise levels to determine the optimal noise value, and its performance is evaluated through experiments and validation. This method involves monitoring the model's accuracy and loss at each noise level to identify which noise value yields the most favorable results. The performance of each noise value will be further discussed in Section 3, providing a detailed analysis of their impact on the model. In the time-warping graph of Fig. 3, the time axis appears elongated, indicating an extension compared to the original data. This extension can be controlled based on the ratio of time stretching. However, excessively increasing or decreasing the time axis can lead to a geometric increase in the amount of data, thereby extending the learning time. Thus, an appropriate ratio should be determined.

The Smoothing graph in Fig. 3 shows a reduction in unstable trends in the graph, a result of averaging values over a specified window size. However, applying the average to significant fluctuations may lead to losing essential characteristics in the graph. Therefore, setting an appropriate window size is crucial. In the Trend Shifting graph of Fig. 3, an overall increase in the reflection coefficient values can be noticed. This shift can be adjusted based on the shifting value. However, applying a fixed value across the board results in a trend similar to the original data.

Noise Injection was selected as the time-series data augmentation technique in this study. Noise Injection involves injecting noise into the existing data without altering its time axis or trends. As a result, it preserves the characteristics of the original data, allowing the model to learn effectively, which often leads to higher performance than other augmentation techniques. This is evidenced in the performance evaluation section, Section 3, additionally, by intentionally introducing noise. Noise Injection has the advantage of enhancing the model's generalization performance.

C. Augmented Data Performance Analysis

LSTM (Long Short-Term Memory) [14] is a model developed to address the issue of long-term dependencies, a limitation inherent in traditional RNNs (Recurrent Neural Networks) [24]. Fig. 4 presents the process of the LSTM model used for multi-class classification performance evaluation.

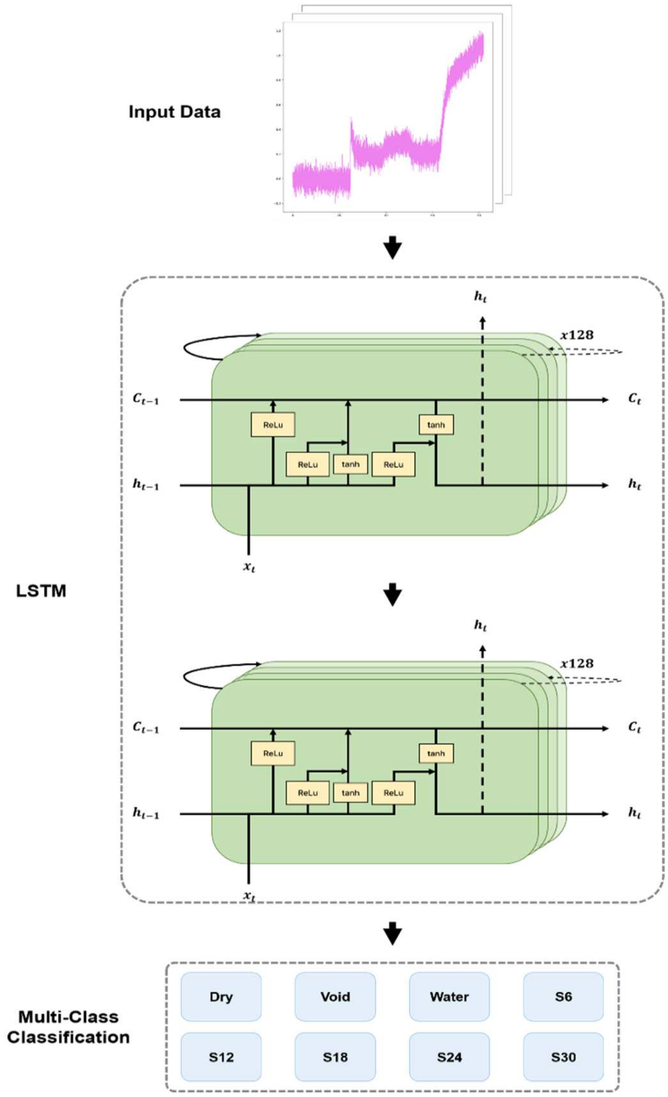

Fig. 4 Process of LSTM model for Multi-Class Classification

Its ability to learn from long sequences is a significant advantage. However, due to its numerous parameters, LSTM incurs high computational costs and is susceptible to overfitting, especially with simple and limited datasets. This study sets the input data feature to 1, indicating that only the reflection coefficient, excluding time, is used as input. The hidden layer's size in the LSTM layer is 128, and the total number of LSTM layers is set to 2. Additionally, a dropout rate of 0.5 is defined. The activation function used is ReLU (Rectified Linear Unit), and the loss function is CrossEntropy [25], [26], [27].

III. Results and Discussion

The performance evaluation of the time-series data augmentation proposed in this study uses the LSTM model. Additionally, to assess the generalization performance of the proposed method, performance evaluations are also carried out on GRU(Gated Recurrent Unit) [16] and TCN(Temporal Convolutional Network) models [17]. For an objective performance comparison, all training variables are set identically. Furthermore, training is conducted from scratch for 50 Epochs without pre-training. The metrics for evaluating the performance of class classification results are Precision, Recall, and F1-Score [28].

Precision is the ratio of correctly classified classes among the predicted classes, while Recall is the number of accurately detected classes among the actual classes. Since Precision and Recall have a trade-off relationship, the F1-Score is utilized. The F1-Score is the harmonic means of Precision and Recall, reflecting a balance between the two metrics. The F1 score is instrumental when class distribution is imbalanced. If one class has significantly more samples than another, using Accuracy alone might lead to a model bias towards the majority class. In such cases, the F1-Score better reflects the model's performance in an imbalanced class distribution. Indeed, by utilizing the F1-Score, it is possible to determine the effectiveness of the applied data augmentation methods in addressing the challenges inherent in time-series data, particularly the prevalence of standard data compared to abnormal data and the imbalance across different classes. The F1-Score is especially valuable in this context as it provides a more nuanced understanding of model performance in the face of class imbalance and varying data distributions. Equation (5) provides the formula for calculating the F1-Score.

$$F1 - Score = \frac{2 * Precision * Recall}{Precision+Recall} \tag{5}$$

Performance evaluation proceeds in four stages. The first is to evaluate the four time-series data augmentation techniques presented in the study using the LSTM model to confirm the optimal augmentation method. Second, the noise injection technique, which is the highest-performance augmentation technique, is evaluated using other models, such as the GRU and TCN, to confirm the generalization performance. Third, it compares and analyzes the LSTM, GRU, and TCN performance and evaluates its speed and accuracy. Finally, in Noise Injection, the performance of each noise value is checked by setting the noise value differently. In this paper, the implementation hardware and operating system used are as follows: Intel(R) Core (TM) i7-9700 Processor, NVIDIA RTX A4000, 64GB RAM, running on Ubuntu 20.04.6, with Python 3.8.0. Additionally, the libraries utilized include PyTorch 2.1.0, among others. This setup ensures a robust and efficient environment for executing the deep learning models and techniques discussed in the study.

A. Performance of Data Augmentation Methods

The performance of various data augmentation methods is assessed by applying different data augmentation techniques to evaluate the performance of multiple data augmentation methods. Parameters are set to commonly used values. In Noise Injection, the Noise Level is set to 0.05. Table 1 presents the performance evaluation of augmentation techniques based on LSTM.

TABLE I
Performance evaluation of augmentation techniques based on LSTM

	F1-Score	Precision	Recall
Noise Injection (Ours)	**0.87**	**0.8966**	**0.876**
Time Warping	0.83	0.81	0.88
Smoothing	0.84	0.81	0.87
Trend Shifting	0.83	0.81	0.88

The performance evaluation results in Table 1 show that the Noise Injection augmentation technique has higher F1-Score, Precision, and Recall compared to Time Warping, Smoothing, and Trend Shifting augmentation techniques. Therefore, the Noise Injection augmentation technique is more effective for class classification. Unlike other augmentation techniques, Noise Injection adds noise to the graph, altering the original data. There is an inherent limitation in the original data of the eight classes, which tend to have similar characteristics. By augmenting the data, it becomes different from the original, aiding the model in classifying classes more easily.

Consequently, it can be observed that the detection accuracy of defect types in the F1-Score is 0.87. Additionally, the paper conducts a performance evaluation using TCN (Temporal Convolutional Network) and GRU (Gated Recurrent Unit) models, applying the most effective augmentation method identified in this study, Noise Injection. The TCN model is a deep learning architecture suitable for processing sequential data. It is characterized by using the 1D convolution layer to learn information about time. The GRU model is also an architecture ideal for processing sequential data. It has a structure similar to that of the LSTM, but it has the characteristics of shortening the time consumed by the calculation speed by reducing the weight of the model structure. It further enhances the validity of the proposed augmentation technique by employing a different model that analyzes time-series data other than LSTM. All models were trained for 50 epochs.

B. *Comparison of Generalization Performance of Noise Injection*

This study observed that the defect detection accuracy F1-Score is 0.87. Table 2 shows the performance evaluation of noise injection augmentation techniques based on LSTM, GRU, and TCN models. According to the performance evaluation results in Table 2, the LSTM (Long Short-Term Memory) [14] model exhibits higher F1-Score, Precision, and Recall compared to the GRU (Gated Recurrent Unit) [15] and TCN (Temporal Convolutional Network) [16] models.

TABLE II
Performance evaluation of noise injection augmentation technique on LSTM, GRU, TCN models

	F1-Score	Precision	Recall
LSTM(Ours)	**0.87**	**0.8966**	**0.8760**
GRU	0.67	0.625	0.75
TCN	0.57	0.56	0.62

Compared to other models, the LSTM model has a complex structure and many parameters. Therefore, it is more effective in evaluating the performance of the Noise Injection augmentation technique. Additionally, the defect detection accuracy F1-Score is 0.87.

C. *Comparison of Computational Speeds of LSTM, GRU, TCN Models*

To understand the correlation between the F1-Score and the calculation speed of the data augmented by the noise injection enhancement technique, a comparison experiment of the calculation speed per width was conducted for each model. Through the calculation speed comparison experiment, the balance between model performance and efficiency can be understood. Table 3 shows the calculation time per 1 epoch per model.

TABLE III
Calculation time per 1 epoch per model

	LSTM(Ours)	GRU	TCN
Computational Speed	**6m 37s**	3m 36s	4m 04s

The computational speed comparison results in Table 3 show that the LSTM (Long Short-Term Memory) model, which had the highest F1-Score, Precision, and Recall, also has the slowest computational speed. Conversely, the GRU (Gated Recurrent Unit) model is observed to have the fastest computational speed. LSTM has a more complex structure than GRU, so the number of parameters is more significant. The more parameters, the larger the capacity of the model, and the slower the calculation speed is inevitable, although more complex patterns can be learned in the training data [29], [30]. In addition, the LSTM uses multiple gates such as update gate, input gate, and output gate. Still, GRU uses only an update gate and a reset gate, so the operation is relatively simpler than the LSTM model [31]. Thus, it is evident that while the LSTM model offers higher accuracy, it does so at the expense of slower computational speed than the GRU model.

D. *Comparison of Performance by Noise Level Based on the LSTM Model*

To find the best noise level, the experiment was conducted by increasing the noise level by 0.01 from 0.02 to 0.05. After that, the data were augmented, and the LSTM model was trained. The LSTM model's performance according to the noise level was confirmed by F1-Score, Precision, and Recall [32]. Table 4 shows the performance change of the LSTM model according to each noise level.

TABLE IV
Comparison of performance by noise level based on LSTM model

Noise Level	F1-Score	Precision	Recall
0.02	0.83	0.83	0.83
0.03(Ours)	**0.90**	**0.93**	**0.87**
0.04	0.58	0.60	0.58
0.05	0.87	0.89	0.87

As a result of the performance comparison in Table 4 shows that when the noise level is 0.03 in the LSTM model, F1-Score is 0.9, Precision is 0.93, and Recall is 0.87, which is the highest performance. In Table 4, when the noise level is 0.04, the F1-Score decreases sharply to 0.58. This phenomenon occurs because of augmented data when the noise level is 0.04. The model does not learn the pattern corresponding to noise, and the data shape is similar to that of other classes. Additionally, it can be observed that the recall for the defect types of detection is 0.87.

E. *Performance of Fault Detection Model*

In previous research, we studied anomaly detection based on PatchCore [33]. This study used data augmentation by adding Gaussian noise and converting time-series data into images using the Markov Transition Field (MTF) method to visualize patterns and sequential dependencies effectively.

Advancing the development of our model, we propose an enhanced anomaly detection model employing LSTM and applying a Noise Level of 0.03. The difference from the previous approach lies in the composition being the same but with a focus on different types of data within the dataset. Specifically, water and air are treated as anomalies since they indicate the presence of defects in concrete structures, while all other types are considered normal. The training and testing were conducted using water and air as the abnormal cases. The test results are presented in Table 5.

TABLE V
PERFORMANCE OF THE AI ANOMALY DETECTION MODEL

	F1-Score	Precision	Recall
Ours	**0.92**	**0.93**	**0.92**

The performance evaluation results in Table 5 show that the precision of the proposed anomaly detection model is 0.93, and the recall is 0.92, confirming that the AI anomaly detection accuracy, measured by the F1-Score, is 0.92.

IV. CONCLUSION

Concrete defects occur due to various factors. However, defects within concrete structures often cannot be detected by human eyes, and accurately classifying the types of defects poses a challenge. Furthermore, the addition of experts' subjective opinions can hinder objective judgment. This study presents a time-series data augmentation for multi-class classification based on the LSTM model to address these issues. The proposed method was to augment time-series data measuring the reflection coefficient of electromagnetic pulses by inserting wires into cracks, moisture-containing concrete, and concrete structures filled with water in the cracks through Noise Injection, Time Warping, Smoothing, and Trend Shift Augmentation techniques. The augmented data is then evaluated using the LSTM model. GRU and TCN models are also used to measure Precision, Recall, F1-Score, and computational speed for the validity of the results. Through the performance evaluation in this study, it has been observed that using the Noise Injection augmentation technique with a Noise Level of 0.03 and the LSTM model, the accuracy of the defect types of detection achieves an F1-Score of 0.9, and the recall for the defect types of detection is 0.87. It was also confirmed that the defect detection accuracy was F1-Score 0.87.

The proposed time-series data augmentation method, noise injection, adds Gaussian noise without altering the time axis or trends of the original data. This preserves the characteristics of the data and increases its quantity, facilitating better model training compared to other augmentation methods. The LSTM model, compared to the GRU model, possesses more parameters and a more complex memory structure. Additionally, compared to the TCN model, the LSTM model employs a gate mechanism to solve the problem of long-term dependencies, leading to superior performance. However, its complex structure results in the slowest computational speed.

The performance evaluation results show that the Noise Injection augmentation technique with the LSTM model yields the highest performance. In addition, data augmentation is best when the noise level is set to 0.03. However, due to its nature, the LSTM model has the slowest computational speed compared to other models. A trade-off between accuracy and speed necessitates selecting the appropriate model based on the situation.

Future research will focus on methods to maintain performance while improving computational speed. Additionally, further studies will explore other augmentation techniques and models to enhance performance.

ACKNOWLEDGMENT

This research was financially supported by the Ministry of Trade, Industry, and Energy (MOTIE), Korea, under the "Project for Research and Development with Middle Markets Enterprises and DNA (Data, Network, AI) Universities" (AI-based Safety Assessment and Management System for Concrete Structures) (reference number P0024559) supervised by the Korea Institute for Advancement of Technology (KIAT).

REFERENCES

[1] M. Munir, S. A. Siddiqui, A. Dengel, and S. Ahmed, "DeepAnT: A Deep Learning Approach for Unsupervised Anomaly Detection in Time Series," IEEE Access, vol. 7, pp. 1991–2005, 2019, doi:10.1109/access.2018.2886457.

[2] H. Ismail Fawaz et al., "InceptionTime: Finding AlexNet for time series classification," Data Mining and Knowledge Discovery, vol. 34, no. 6, pp. 1936–1962, Sep. 2020, doi: 10.1007/s10618-020-00710-y.

[3] L. Bergman and Y. Hoshen, "Classification-based anomaly detection for general data," arXiv preprint arXiv:2005.02359, 2020.

[4] M. Wiese, R. Knobloch, R. Korn, and P. Kretschmer, "Quant GANs: deep generation of financial time series," Quantitative Finance, vol. 20, no. 9, pp. 1419–1440, Apr. 2020, doi:10.1080/14697688.2020.1730426.

[5] C. Sun, S. Hong, M. Song and H. Li, "A review of deep learning methods for irregularly sampled medical time series data," arXiv preprint arXiv:2010.12493, 2020.

[6] P. Hewage et al., "Temporal convolutional neural (TCN) network for an effective weather forecasting using time-series data from the local weather station," Soft Computing, vol. 24, no. 21, pp. 16453–16482, Apr. 2020, doi: 10.1007/s00500-020-04954-0.

[7] A. Copiaco et al., "An innovative deep anomaly detection of building energy consumption using energy time-series images," Engineering Applications of Artificial Intelligence, vol. 119, p. 105775, Mar. 2023, doi: 10.1016/j.engappai.2022.105775.

[8] Y. Liu et al., "Deep Anomaly Detection for Time-Series Data in Industrial IoT: A Communication-Efficient On-Device Federated Learning Approach," IEEE Internet of Things Journal, vol. 8, no. 8, pp. 6348–6358, Apr. 2021, doi: 10.1109/jiot.2020.3011726.

[9] B. K. Iwana and S. Uchida, "An empirical survey of data augmentation for time series classification with neural networks," PLOS ONE, vol. 16, no. 7, p. e0254841, Jul. 2021, doi: 10.1371/journal.pone.0254841.

[10] H. Li, "Time works well: Dynamic time warping based on time weighting for time series data mining," Information Sciences, vol. 547, pp. 592–608, Feb. 2021, doi: 10.1016/j.ins.2020.08.089.

[11] A. Camuto, M. Willetts, U. Simsekli, S. J. Roberts and C. C. Holmes, "Explicit regularisation in gaussian noise injections," In Proceedings of the 34th International Conference on Neural Information Processing Systems (NIPS '20). Curran Associates Inc., Red Hook, NY, USA, Article 1393, 16603–16614, 2020.

[12] C.-B. Zhang et al., “Delving Deep Into Label Smoothing,” IEEE Transactions on Image Processing, vol. 30, pp. 5984–5996, 2021, doi:10.1109/tip.2021.3089942.

[13] M. Sayemuzzaman and M. K. Jha, “Seasonal and annual precipitation time series trend analysis in North Carolina, United States,” Atmospheric Research, vol. 137, pp. 183–194, Feb. 2014, doi:10.1016/j.atmosres.2013.10.012.

[14] Q. Wen, L. Sun, F. Yang, X. Song, J. Gao, X. Wang and H. Xu, "Time series data augmentation for deep learning: A survey," arXiv preprint arXiv:2002.12478, 2020.

[15] F. Karim, S. Majumdar, H. Darabi, and S. Chen, “LSTM Fully Convolutional Networks for Time Series Classification,” IEEE Access, vol. 6, pp. 1662–1669, 2018, doi: 10.1109/access.2017.2779939.

[16] W. Zheng and G. Chen, “An Accurate GRU-Based Power Time-Series Prediction Approach With Selective State Updating and Stochastic Optimization,” IEEE Transactions on Cybernetics, vol. 52, no. 12, pp. 13902–13914, Dec. 2022, doi: 10.1109/tcyb.2021.3121312.

[17] J. Fan, K. Zhang, Y. Huang, Y. Zhu, and B. Chen, “Parallel spatio-temporal attention-based TCN for multivariate time series prediction,” Neural Computing and Applications, vol. 35, no. 18, pp. 13109–13118, May 2021, doi: 10.1007/s00521-021-05958-z.

[18] F. Ramzan et al., “A Deep Learning Approach for Automated Diagnosis and Multi-Class Classification of Alzheimer’s Disease Stages Using Resting-State fMRI and Residual Neural Networks,” Journal of Medical Systems, vol. 44, no. 2, Dec. 2019, doi: 10.1007/s10916-019-1475-2.

[19] F. Wang, K. Li, and Y. Liu, “Optimal water-cement ratio of cement-stabilized soil,” Construction and Building Materials, vol. 320, p. 126211, Feb. 2022, doi: 10.1016/j.conbuildmat.2021.126211.

[20] O. Lotfi-Omran, A. Sadrmomtazi, and I. M. Nikbin, “A comprehensive study on the effect of water to cement ratio on the mechanical and radiation shielding properties of heavyweight concrete,” Construction and Building Materials, vol. 229, p. 116905, Dec. 2019, doi:10.1016/j.conbuildmat.2019.116905.

[21] X. Cai, T. Xu, J. Yi, J. Huang, and S. Rajasekaran, "Dtwnet: a dynamic time warping network," Proceedings of the 33rd International Conference on Neural Information Processing Systems. Curran Associates Inc., Red Hook, NY, USA, Article 1044, 11640–11650, 2019

[22] S. Hansun, “A new approach of moving average method in time series analysis,” 2013 Conference on New Media Studies (CoNMedia), Nov. 2013, doi: 10.1109/conmedia.2013.6708545.

[23] Z. Zhao, Z. Wu, Y. Zheng, and P. Ma, “Recurrent neural networks for atmospheric noise removal from InSAR time series with missing values,” ISPRS Journal of Photogrammetry and Remote Sensing, vol. 180, pp. 227–237, Oct. 2021, doi: 10.1016/j.isprsjprs.2021.08.009.

[24] W. Yu, I. Y. Kim, and C. Mechefske, “Analysis of different RNN autoencoder variants for time series classification and machine prognostics,” Mechanical Systems and Signal Processing, vol. 149, p. 107322, Feb. 2021, doi: 10.1016/j.ymssp.2020.107322.

[25] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," J. Mach. Learn. Res., vol. 15, no. 1, pp. 1929-1958, 2014.

[26] T. Szandała, “Review and Comparison of Commonly Used Activation Functions for Deep Neural Networks,” Studies in Computational Intelligence, pp. 203–224, Jul. 2020, doi: 10.1007/978-981-15-5495-7_11.

[27] L. Hui, M. Belkin, "Evaluation of neural architectures trained with square loss vs cross-entropy in classification tasks," arXiv preprint arXiv:2006.07322, 2020.

[28] H. J. Kwon, M. J. Kim, J. W. Baek, and K. Chung, "Voice Frequency Synthesis using VAW-GAN based Amplitude Scaling for Emotion Transformation,” KSII Transactions on Internet and Information Systems, vol. 16, no. 2, Feb. 2022, doi: 10.3837/tiis.2022.02.018.

[29] Y. Yu, X. Si, C. Hu, and J. Zhang, “A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures,” Neural Computation, vol. 31, no. 7, pp. 1235–1270, Jul. 2019, doi:10.1162/neco_a_01199.

[30] K. Smagulova and A. P. James, “A survey on LSTM memristive neural network architectures and applications,” The European Physical Journal Special Topics, vol. 228, no. 10, pp. 2313–2324, Oct. 2019, doi:10.1140/epjst/e2019-900046-x.

[31] B. U. Jeon, K. Chung, " “CutPaste-Based Anomaly Detection Model using Multi Scale Feature Extraction in Time Series Streaming Data,” KSII Transactions on Internet and Information Systems, vol. 16, no. 8, Aug. 2022, doi: 10.3837/tiis.2022.08.018, 2022

[32] H. Yoo, R. C. Park, K. Chung, “IoT-Based Health Big-Data Process Technologies: A Survey,” KSII Transactions on Internet and Information Systems, vol. 15, no. 3, Mar. 2021, doi:10.3837/tiis.2021.03.009.

[33] G.-I. Kim, H. Yoo, H.-J. Cho, and K. Chung, “Defect Detection Model Using Time Series Data Augmentation and Transformation,” Computers, Materials &amp; Continua, vol. 78, no. 2, pp. 1713–1730, 2024, doi: 10.32604/cmc.2023.046324.