Deep Learning-based SBOM Defect Detection for Medical Devices

Heeyeon Kim
dept. Knowledge Information Engineering
Ajou University
Suwon, Republic of Korea
heey08@ajou.ac.kr

Ki-Hyung Kim
dept. Cyber Security
Ajou University
Suwon, Republic of Korea
kkim86@ajou.ac.kr

***Abstract*—This paper focuses on the digital innovation brought about by the advancement of ICT technology in the healthcare field. In particular, the introduction of cutting-edge technologies such as AI are contributing to the improvement of health care services, but at the same time, the risk of cybersecurity threats and software security problems in medical devices are emerging as well. Therefore, the importance of SBOM (Software Bill of Materials) to respond to such problems has now been emphasized, and this paper focuses on strengthening the cybersecurity of medical devices through the creation and application of SBOM. This study develops an AI-based software defect detection method and seeks to ensure the reliability and safety of medical devices by proposing an efficient method compared to existing rule-based approaches. This study aims to contribute to strengthening security by detecting flaws in medical devices through AI and establishing a technical foundation for providing ultimate quality healthcare services.**

***Index Terms*—SBOM, Deep Learning, Medical Device, Software Defect Detection, Cybersecurity, Artificial Intelligence (AI)**

I. Introduction

Recently, the rapid development of ICT technology has revolutionized the healthcare sector by combining it with digital technology. In particular, technological advances in digital healthcare have improved the efficiency of patient management and healthcare services [1]. In particular, the introduction of Artificial Intelligence (AI) technology is increasing in importance in various fields by helping to analyze health and medical data, support diagnosis, and establish treatment plans. AI technology is contributing greatly to early detection of diseases and providing customized treatment plans by quickly and accurately analyzing large amounts of data [2]. This is helping to improve the quality of life for both patients and healthcare providers.

However, despite these technological advances, the security of the software of medical devices remains a significant challenge. Software security of medical devices is a key factor in protecting patients' personal information and health data, and security is especially emphasized as cyber attacks and data leaks are possible. To address these issues, the concept of a Software Bill of Materials (SBOM) was introduced. SBOM can clearly identify all components inside software and help identify potential security vulnerabilities, contributing to increasing the safety and reliability of software. In particular, the application of SBOM in the digital healthcare field is essential to protect patient data and ensure the stability of medical devices. SBOM can improve the overall security level in the digital healthcare field by managing software security vulnerabilities more effectively and contributing to increasing the reliability and safety of the digital healthcare system. Additionally, it is expected to play an important role in providing a safe digital environment for both healthcare providers (HCPs) and patients [3].

Therefore, this research aims to provide new insights and technical contributions to the fields of digital healthcare and cybersecurity. By focusing on flaw detection and security vulnerabilities in digital healthcare software, the AI-based SBOM approach presents a new methodology to effectively identify and address security vulnerabilities in digital healthcare software, which aims to focus on cybersecurity management in healthcare. In addition, by developing and evaluating effective ways to enhance the security of digital healthcare software by utilizing SBOM based on AI, we aim to provide specific solutions to improve the overall level of cybersecurity.

Chapter 2 of this thesis analyzes related research on which this paper is based. Additionally, Chapter 3 of this thesis explains the concepts and research methodology related to the study and proposes the method of this study. Chapter 4 presents the implementation and evaluation of the research and experiment. Finally, Chapter 5 concludes with the conclusion.

II. Related Work

Modern healthcare is advancing through digital transformation with AI and cybersecurity at its core. This study analyzes the related research on SBOM, which is an important component of digital healthcare software, and presents related research to utilize AI. Also we explore the application of AI to SBOM by analyzing existing research to propose an ultimate deep learning-based software defect detection model.

A. SBOM (Software Bill of Materials)

SBOM has recently been gaining importance in many fields to ensure software transparency and strengthen security. However, in order to use SBOM more safely and effectively, there are limitations to simply using SBOM itself. Therefore, studies dealing with techniques for efficient use of SBOM

are as follows. L. Camp [4] aims to improve the overall cybersecurity state to use SBOM more safely and effectively.

In addition, studies that have dealt with existing SBOM to use the efficiency of AI are as follows. B. Xia [5] proposed a method utilizing blockchain for enhancing the reliability of software supply chains and employing AI, introducing a new concept called AIBOM. B. P. Radanliev [6] mentioned the importance of SBOM and proposed a framework that can create and share VEX data and combine it with AI models. Therefore, based on existing research presented in related research, this study also would like to propose a technique that combines SBOM.

B. Software Defect Detection

Methods for detecting software defects include existing rule-based research. However, with the advancement of technology, unlike existing software, it has become increasingly complex. Fragmented software systems require the identification of complex patterns and relationships. Therefore, it is necessary to process large amounts of data and quickly detect defects through automated software detection based on AI.

Recently, research is being conducted on techniques to predict defects by combining various algorithms of AI models with software defect detection technology. G. Esteves [7] focused on defect prediction using machine learning to detect known software, defects, and M. Cetiner [8] used 10 different machine learning algorithms and then compared them to predict software defects. We sought to improve the quality of software through prediction. In addition, J. Li [9] attempted to increase the reliability of software by predicting defects and proposing a framework using the DP-CNN model. N. Shakhovska [10] presents an improved software defect prediction algorithm based on hierarchical clustering combination. Therefore, in this study, we attempted to detect defects in the SBOM creation process of medical devices based on deep learning by referring to existing research.

III. Proposed Method

A. Deep Learning-based Software Defect Detection

This study seeks to develop a software defect detection model using open source-based software defect detection data. As a specific model, we plan to use supervised learning using two deep learning models, DNN and Conv1D. The model construction procedure first preprocesses the data, learns using two models, and then measures and analyzes performance through the use of performance evaluation indicators. In the preprocessing process, the model is divided into training, validation, and test data sets, and then down sampling and outlier removal are performed on the training data set. Afterwards, we plan to go through scaling and feature extraction on the training, verification, and test datasets then reduce the dimension through PCA, use the preprocessed dataset, apply it to the model, learn it, and evaluate performance. The specific work flow diagram of this study is shown in Fig. 1.

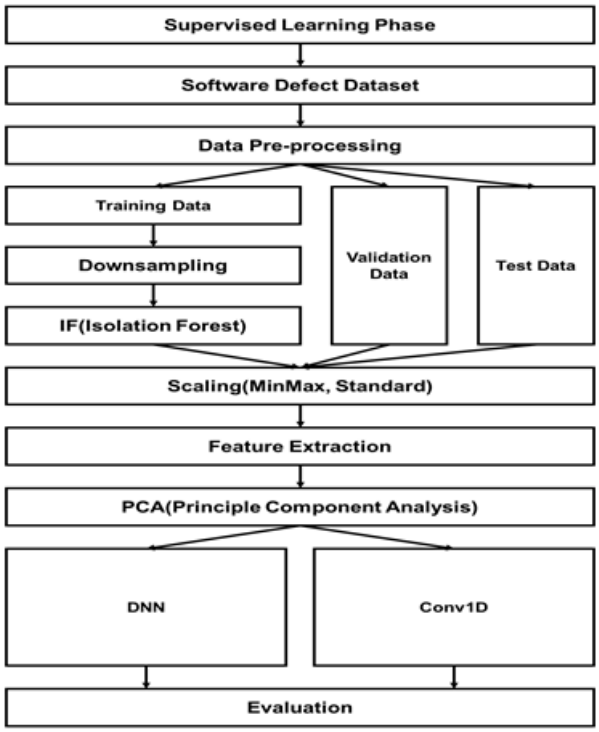

Fig. 1. Workflow diagram of deep learning-based software defect detection

a) Dataset: The dataset for this study is Binary Classification with a Software Defects Dataset is used, and it is an open dataset that allows predicting defects in C programs by considering various properties of the code [11]. It consists of a dataset and a test dataset. The train dataset has a label, while the test dataset has no label. The dataset is 101,763 in total, consisting of 78,699 normal data and 23,064 attack data, and has a total of 23 characteristics.

b) Data Preprocessing: Before constructing a model, the data preprocessing process is an important step to secure high quality data and optimize model performance. In this paper, we would like to propose a model that can effectively process data and generalize through a preprocessing process, which is the basis for binary classification using the 'software defect dataset'. First, the dataset is divided into training, validation and test datasets then used for model learning and evaluation. The training dataset ensures quality data through downsampling and outlier removal. Next, the training, validation, and test datasets go through scaling and feature extraction processes to enable the model to produce important information through meaningful features. Fig. 2 is a graphic visualization of the results after feature extraction based on mutual information. Finally, principal component analysis (Principle Component Analysis, hereinafter referred to as PCA) reduces the complexity of data and increases efficiency through dimensionality reduction.

c) Constructing Deep Learning Models (DNN and Conv1D): In this study, we aim to develop a deep learning

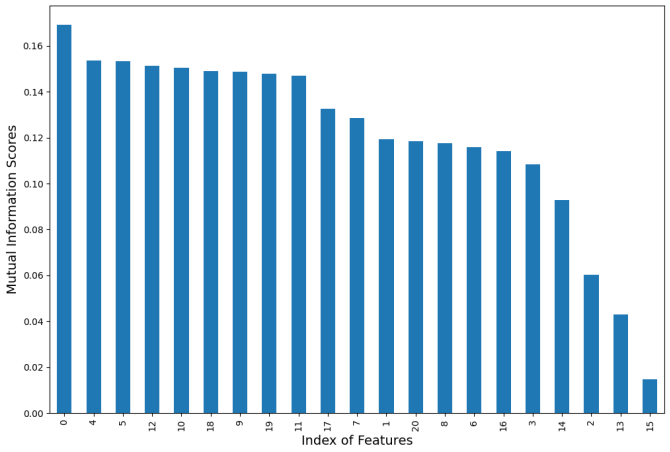

Fig. 2. Example of feature extraction results

model using a preprocessed dataset. We want to utilize the deep learning model structures of DNN and Conv1D, which are suitable for binary classification. First, DNN [12] is an artificial neural network structure composed of deep neural networks. It generally uses a fully connected layer and can learn complex patterns, producing high performance on various types of data such as images, voices, and texts. Next, Conv1D [13] is a one-dimensional synthetic neural network that can effectively utilize sequential data such as time series data or text. It has the characteristic of using a kernel to scan input data and find patterns and features.

Based on the selected model, we want to construct a model that can prevent overfitting and generalize by adjusting hyperparameters such as activation and dropout when designing the model. In addition, in the process of training the model, hyperparameters such as loss, optimizer, learning rate, batch size, and epochs are adjusted to find the optimal combination to improve performance.

d) Performance evaluation indicators: In this study, accuracy, precision, recall, F1-score, and AUC (Area Under the Curve) were used as performance evaluation indicators. Performance evaluation indicators are intended to evaluate the model's classification performance from various angles. Accuracy refers to the proportion of the total data that the model predicts correctly, and is used as a measure of general performance. Precision measures the ratio of actual positive data among data predicted to be positive, and evaluates how accurately the model makes positive predictions. Recall rate indicates the proportion of actual positive data that the model correctly predicts as positive, and measures whether the model predicts well without missing actual positive data. F1-score takes the harmonic average of precision and recall, providing a performance indicator that considers the balance between the two indicators. Lastly, AUC probabilistically represents the model's ability to actually distinguish positives from negatives, and refers to the area under the ROC curve (Receiver Operating Characteristic curve). This is an indicator of how better the model is at distinguishing positives than randomly predicting positives. The closer AUC is to 1, the better the model's performance. In this way, the performance evaluation indicator is an indicator that can confirm whether the model demonstrates consistent and reliable performance in various situations and conditions.

B. Deep Learning-based Software Defect Detection when Generating SBOM

This paper proposes a technique to detect software defects using deep learning when generating SBOM of medical devices. Specifically, manufacturers of medical devices use multiple software to manufacture medical devices. Therefore, there is a need to manage the software components that make up medical devices. SBOM is needed to manage this more efficiently and transparently. However, there is a risk of vulnerabilities or threats in configuring SBOM. Therefore, a technique for detecting software defects is needed when creating SBOM. This paper proposes a method to detect software defects using deep learning.

Fig. 3. is an illustration of a deep learning-based software defect detection technique when generating SBOM. After detecting software defects in the software components that make up medical devices using deep learning techniques, a method was presented to configure SBOM only with defect-free software.

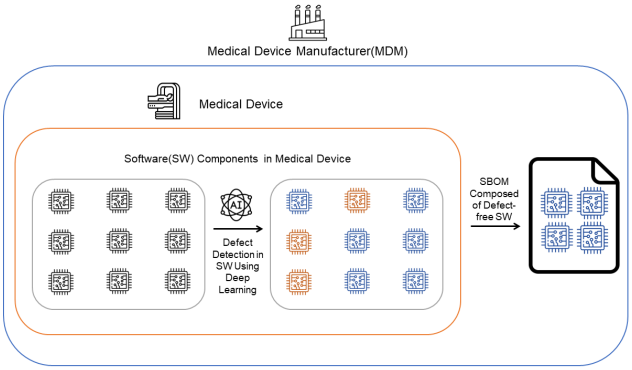

Fig. 3. Deep learning-based software defect detection method when generating SBOM

IV. Implementation and Performance Evaluation

A. Implementation of Deep Learning-based Software Defect Detection Models

Through the results of the model construction process discussed in Chapter 3, this study developed DNN and Conv1D deep learning models. Hyperparameters of loss function Binary Crossentropy, epoch 500, batch size 256, and learning rate 0.001 were commonly applied to both models. Table I presents the hyperparameters of the DNN and Conv1D models.

In this paper, we attempted to develop a software defect detection model and evaluate its performance using DNN and Conv1D models. Table II and Table III respectively show the architecture of the DNN model and Conv1D model in detail.

TABLE I
HYPERPARAMETERS OF DNN AND CONV1D MODELS

Model	Hyperparameters	
DNN	Layers	100, 30, 1
	Dense	1
	Activation	tanh, relu, sigmoid
	Optimizer	nAdam
	Loss	Binary crossentropy
Conv1D	Layers	16, 128
	Dense	10, 1
	Activation	relu, sigmoid
	Optimizer	nAdam
	Loss	Binary crossentropy

TABLE II
DNN MODEL ARCHITECTURE

Layer	Units	Output Shape
DNN_1	100	(None, 100)
DNN_2	30	(None, 30)
DNN_3	15	(None, 1)

B. Deep Learning-based Software Defect Detection Performance Evaluations

a) t-SNE: For deep learning-based software defect detection, t-SNE was used to check the distribution of the dataset. t-SNE (t-distributed Stochastic Neighbor Embedding) is a technique for visualizing high-dimensional data by reducing it to low-dimensional data. Fig. 4. shows the distribution of normal data and attack data through t-SNE. Additionally, Fig. 5. is a diagram visualizing the distribution of data changing with the downsampling rate using t-SNE.

b) Performance Evaluation Indicators: In this paper, two deep learning models, DNN (Deep Neural Network) and Conv1D (1D Convolutional Neural Network), were used, and the performance evaluation indicators include Accuracy, Precision, Recall, and F1. -score and AUC were used.

Table IV is a table comparing the performance using DNN and Conv1D models. The DNN model produced performance of accuracy 0.736, precision 0.77, recall 0.672, F1-score 0.717, and AUC 0.736. On the other hand, the Conv1D model showed performance of accuracy 0.722, precision 0.744, recall 0.677, F1-score 0.708, and AUC 0.722. These results indicate that the DNN model showed overall better performance than the Conv1D model. Fig. 6. is an example of an AUROC curve to visually evaluate model performance.

TABLE III
DNN MODEL ARCHITECTURE

Layer	Units	Output Shape
$Conv1D_1$	100	(None, 4, 16)
MaxPool1D	2	(None, 2, 16)
$Conv1D_2$	15	(None, 2, 128)
Flatten	-	(None, 256, 16)
$Dense_1$	10	(None, 10)
$Dense_2$	1	(None, 1)

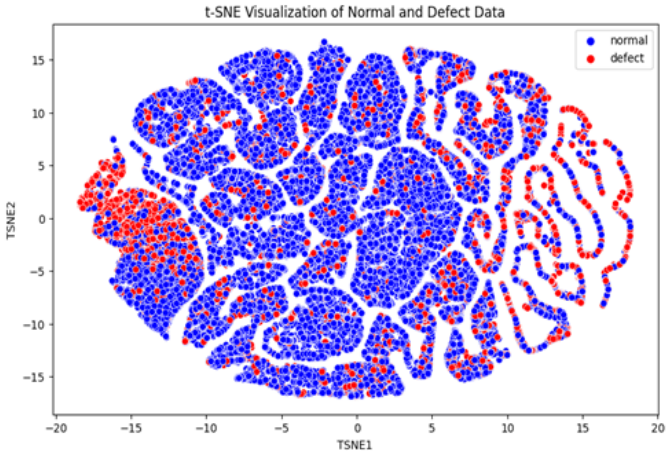

Fig. 4. Visualizing normal and attack data with t-SNE

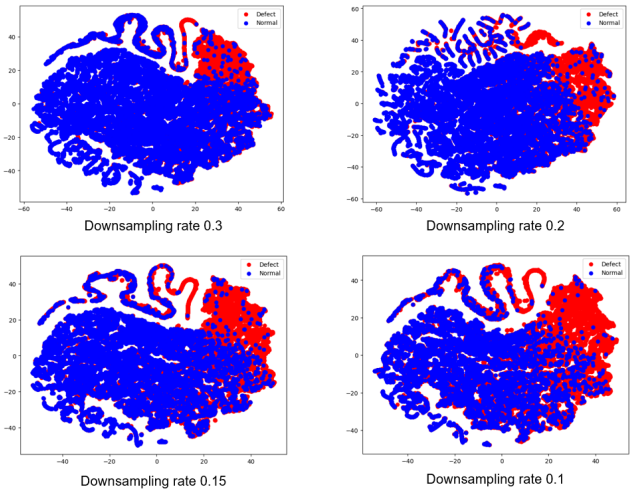

Fig. 5. Visualization of data distribution according to downsampling rate through t-SNE

V. CONCLUSION

This paper applied AI techniques to SBOM generation and software defect detection to strengthen the cybersecurity of medical devices and ensure safety and reliability. The AI model developed in this study used a deep learning-based model, unlike existing rule-based defect detection methods. In particular, it secured the stability of medical devices by analyzing complex interactions of software to identify potential vulnerabilities made it possible to do so. In addition, we aim to increase the transparency of software components contained in devices by distinguishing SBOM components

TABLE IV
DEEP LEARNING-BASED SOFTWARE DEFECT DETECTION PERFORMANCE

Model	Accuracy	Precision	Recall	F1-score	AUC
DNN	0.736	0.77	0.672	0.717	0.736
Conv1D	0.722	0.744	0.677	0.709	0.722

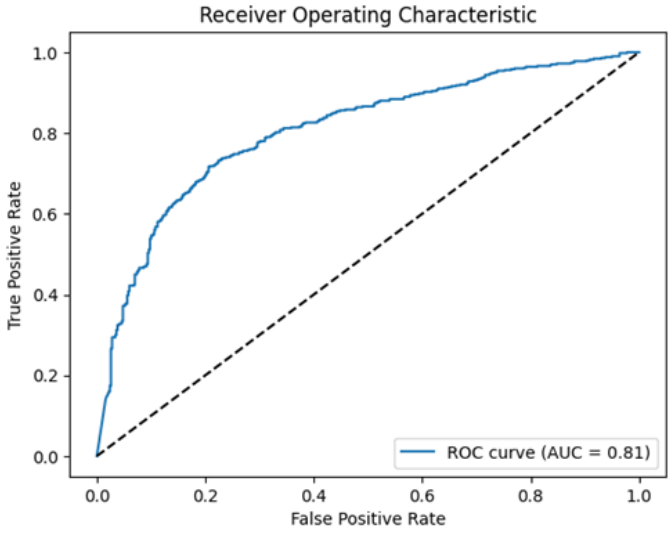

Fig. 6. AUROC curve example for performance evaluation

based on deep learning and protect medical devices from cybersecurity threats. Therefore, deep learning-based defect detection techniques seek to improve the long-term reliability of medical devices by identifying and distinguishing software defects at the SBOM generation stage.

As future research in this paper, we will first apply semi-supervised learning techniques to improve the performance of AI models. Using semi-supervised learning, you can expand the applicability of the model to various datasets and improve performance by increasing the number of training data. In addition to the SBOM creation stage, also we can explore the application of AI in other software life cycle stages. This allows strengthening of security by creating overall cybersecurity protocols for software across multiple software life cycles. Lastly, this study has limitations because it did not develop a deep learning model using the SBOM dataset of actual medical devices. Therefore, in future research, use data collected in the actual healthcare field or datasets that constitute the SBOM of actual medical devices to conduct highly usable research so that the AI model can be utilized more appropriately and applied to real life. Through this, reliability of quality will be secured through the development of safe medical devices, which is expected to contribute to the healthcare system by serving as the basis for providing ultimate quality healthcare services.

ACKNOWLEDGMENT

This research was supported in part by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2024-2021-0-01835) supervised by the IITP (Institute for Information Communications Technology Planning Evaluation). This research was supported by Korea Institute for Advancement of Technology (KIAT) grant funded by the Korea Government (MOTIE) (P0008703, HRD Program for Industrial Innovation). This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (2021R1F1A1045861). This research was supported by Institute of Information communications Technology Planning Evaluation (IITP) grant funded by the Korea government (MSIT) (2021-0-00590, Decentralized High Performance Consensus for Large-Scale Blockchains). This research was supported by Korea Research Institute for defense Technology Planning and advancement (KRIT) grant funded by the Korea government (DAPA(Defense Acquisition Program Administration)) (KRIT-CT-23-041, LiDAR/RADAR Supported Edge AI-based Highly Reliable IR/UV FSO/OCC Specialized Laboratory, 2024).

REFERENCES

[1] R. Sharma and N. Kshetri, "Digital healthcare: Historical development, applications, and future research directions," *International Journal of Information Management*, vol. 53. Elsevier BV, p. 102105, Aug. 2020

[2] A. Arora, "Conceptualising Artificial Intelligence as a Digital Healthcare Innovation: An Introductory Review," *Medical Devices: Evidence and Research*, vol. Volume 13. Informa UK Limited, pp. 223–230, Aug. 2020.

[3] S. Carmody et al., "Building resilient medical technology supply chains with a software bill of materials," *npj Digital Medicine*, vol. 4, no. 1. Springer Science and Business Media LLC, Feb. 23, 2021.

[4] L. J. Camp and V. Andalibi, "SBOM vulnerability assessment & corresponding requirements," in NTIA Response to Notice and Request for Comments on Software Bill of Materials Elements and Considerations, 2021.

[5] B. Xia, D. Zhang, Y. Liu, Q. Lu, Z. Xing, and L. Zhu, "Trust in Software Supply Chains: Blockchain-Enabled SBOM and the AIBOM Future." *arXiv*, 2023, doi: 10.48550/arXiv.2307.02088.

[6] P. Radanliev, D. De Roure, and O. Santos, "Generative Pre-Trained Transformers, Natural Language Processing and Artificial Intelligence and Machine Learning (AI/ML) in Software Vulnerability Management: automations in the Software Bill of Materials (SBOM) and the Vulnerability-Exploitability eXchange (VEX)." MDPI AG, Jul. 19, 2023.

[7] G. Esteves, E. Figueiredo, A. Veloso, M. Viggiato, and N. Ziviani, "Understanding machine learning software defect predictions," *Automated Software Engineering*, vol. 27, no. 3–4. Springer Science and Business Media LLC, pp. 369–392, Oct. 12, 2020.

[8] M. Cetiner and O. K. Sahingoz, "A Comparative Analysis for Machine Learning based Software Defect Prediction Systems," *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*. IEEE, Jul. 2020.

[9] J. Li, P. He, J. Zhu, and M. R. Lyu, "Software Defect Prediction via Convolutional Neural Network," *2017 IEEE International Conference on Software Quality, Reliability and Security (QRS)*. IEEE, Jul. 2017.

[10] N. Shakhovska, V. Yakovyna, and N. Kryvinska, "An Improved Software Defect Prediction Algorithm Using Self-organizing Maps Combined with Hierarchical Clustering and Data Preprocessing," *Lecture Notes in Computer Science*. Springer International Publishing, pp. 414–424, 2020.

[11] W. Reade and A. Chow, "Binary Classification with a Software Defects Dataset," Kaggle, 2023. [Online]. Available: https://kaggle.com/competitions/playground-series-s3e23.

[12] J.-T. Chien, *Source Separation and Machine Learning*, Elsevier, 2019.

[13] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, and D. J. Inman, "1D convolutional neural networks and applications: A survey," *Mechanical Systems and Signal Processing*, vol. 151. Elsevier BV, p. 107398, Apr. 2021.