

Data-fused and concatenated-ensemble learning for in-situ anomaly detection in wire and arc-based direct energy deposition

Duck Bong Kim^a, Hamin Chong^b, Mohammad Mahruf Mahdi^c, Seung-Jun Shin^{d,*}

^a Department of Manufacturing and Engineering Technology, Tennessee Technological University, United States

^b Department of Industrial Data Engineering, Hanyang University, Republic of Korea

^c Department of Electrical and Computer Engineering, Tennessee Technological University, United States

^d School of Interdisciplinary Industrial Studies, Hanyang University, Republic of Korea

ARTICLE INFO

Keywords:

Direct energy deposition
Anomaly detection
Wire + arc additive manufacturing
Concatenated ensemble learning

ABSTRACT

Convolutional neural network (CNN), a type of deep learning algorithm, is a powerful tool for analyzing visual images. It has been actively investigated to monitor metal additive manufacturing (AM) processes for quality control and has been proven effective. However, typical CNN algorithms inherently have two issues when used in metal AM processes. First, in many cases, acquiring datasets with sufficient quantity and quality, as well as necessary information, is challenging because of technical difficulties and/or cost intensiveness. Second, determining a near-optimal CNN model takes considerable effort and is time-consuming. This is because the types and quality of datasets can be significantly different with respect to different AM processes and materials. The study proposes a novel concatenated ensemble learning method to obtain a flexible and robust algorithm for in-situ anomaly detection in wire + arc additive manufacturing (WAAM), a type of wire-based direct energy deposition (DED) process. For this, data, as well as machine learning models, were seamlessly integrated to overcome the limitations and difficulties in acquiring sufficient data and finding a near-optimal machine learning model. Using inexpensively obtainable and comprehensive datasets from the WAAM process, the proposed method was investigated and validated. In contrast to the one-dimensional and two-dimensional CNN models' accuracies of 81.6 % and 88.6 %, respectively, the proposed concatenated ensemble model achieved an accuracy of 98 %.

1. Introduction

Metal additive manufacturing (AM) can be categorized into powder-bed fusion (PBF) and direct-energy deposition (DED) processes. DED can be further classified into powder-fed and wire-fed processes, which in turn can be classified according to energy source electron beam-, laser-, ultrasonic-, and arc-based systems [1]. This study focuses on an arc-based system known as wire-arc additive manufacturing (WAAM), which utilizes the wire as the feeding stock, welding arc as the energy source, and robot arms or computer numerical control (CNC) router for movement [2]. WAAM has the following advantages: (1) inexpensive setup, (2) high deposition rate (up to 10 kg/h), and (3) wide selection of materials [2,3]. In addition, the energy efficiency of WAAM is >90 %, whereas that of the laser/electron beams is 5–20 % [4–6]. WAAM can be further classified into gas metal arc welding (GMAW), gas tungsten arc welding (GTAW), and plasma-based processes [7].

Metal AM has inherent uncertainties and complexities because it is a multi-scale, multiphysics process [8]. For instance, non-equilibrium WAAM thermal cycles and layer-by-layer stacking mechanisms induce defects that negatively affect the geometry, surface quality, microstructure, and mechanical properties [9]. This contrasts with the ultimate goal of metal AM, which is to fabricate defect-free parts with desirable structures applicable as end-user products. Numerous factors, such as process complexities and instabilities, contribute to the quality and microstructure of an AM part, which can be controlled by aligning process parameters with optimized values through in-situ monitoring and control. Although the critical factor in a successful AM process is monitoring and feedback control [10], in-situ data acquisition for closed-loop control and detecting material discontinuities have been highlighted as crucial barriers to AM implementation as well as a priority area for research and development [11].

Design of experiments (DOE) is a commonly used approach for

* Corresponding author at: 2nd Engineering Building No. 113, 222 Wangsimni-ro, Seongdong-gu, Seoul 04763, Republic of Korea.

E-mail addresses: dkim@tntech.edu (D.B. Kim), gsosun@hanyang.ac.kr (H. Chong), mmahdi42@tntech.edu (M.M. Mahdi), sjshin@hanyang.ac.kr (S.-J. Shin).

<https://doi.org/10.1016/j.jmapro.2024.01.041>

Received 22 August 2023; Received in revised form 14 November 2023; Accepted 11 January 2024

Available online 23 January 2024

1526-6125/© 2024 The Society of Manufacturing Engineers. Published by Elsevier Ltd. All rights reserved.

process control and window-map generation for metal AM processes because of its success rate and efficiency. However, it cannot be used for in-situ monitoring and control since several experiments must be conducted, which requires considerable resources and time. Machine learning (ML) approaches have gained increasing attention for overcoming the limitations of DOE [12]. ML algorithms, in general, can be divided into supervised, unsupervised, and reinforcement learning [13]. In supervised learning, each input datum is labeled with an output, and the training set consists of many input-output pairs [14]. Unsupervised learning infers unlabelled data and can find hidden patterns or similar group data (i.e., clustering) in a given dataset and perform anomaly detection [15,16]. In contrast, reinforcement learning is a semi-supervised ML paradigm that employs agents to make decisions to maximize the cumulative reward. It can be implemented for anomaly detection when the problem is formulated as a Markov decision process (MDP) [17].

ML techniques are considered a viable solution to establish the relationships among process, structure, and property for AM control, called the “design rule.” Since process parameters significantly affect microstructures and mechanical properties, reliable and extensive datasets for training are required. The vast number of process parameters results in the “curse of dimensionality”. For instance, since the number of process parameters in a powder bed fusion process is >100, the possible combinations of process parameters exponentially increase, as shown by Mani et al. [18]. Moreover, acquiring datasets for high-performance materials (e.g., nickel- or titanium-based alloys) would be cost-intensive. A flexible and robust ML algorithm can work around the need for large datasets and thus is necessary for anomaly detection in metal AM processes.

To achieve real-time anomaly detection, convolutional neural networks (CNN), a form of supervised machine learning algorithms, have been widely studied [12]. For example, two CNN algorithms that could obtain accuracies of 95.5 % [19] and 97.5 % [20], respectively, were successfully developed for real-time anomaly detection. However, these typical CNN algorithms have two inherent issues. First, in many cases, acquiring datasets of reliable quality after sensitivity analysis is time-consuming and cost-intensive due to the requirement of expensive resources. Second, although necessary datasets are obtained, finding a near-optimal CNN model is time-consuming because CNN models significantly depend on datasets’ types and quality, which can differ considerably with variations in AM processes and materials.

Ensemble learning (EL) is a viable solution because it uses multiple learning algorithms to obtain better predictive performance than any constituent learning algorithms alone [21]. It compensates for poor learning algorithms by performing additional computations. Although EL algorithms are widely used in other manufacturing processes, to our knowledge, they have neither been investigated nor applied in metal AM processes because they lack available datasets [12]. For example, in the case of the powder bed process to generate a dataset, a high-resolution, high-speed thermal camera with an appropriate field of view is required to capture clear melt-pool images because of the rapid solidification [22], and AM-grade Inconel 625 powder, costs double the price of its wire counterpart [23]. For these reasons, datasets for EL investigations are lacking in the case of laser- and powder-based AM processes.

This study proposes a novel data-fused, concatenated EL method to develop a flexible and robust algorithm for in-situ anomaly detection in AM. This algorithm involves four stages: data pre-processing, data preparation, pretraining, predictions, and concatenated ensemble. In the data pre-processing stage, voltage and WAAM video data are collected, transformed into 1D (one-dimensional) and 2D (two-dimensional) formats, resized, and labeled. Data preparation involves synchronizing and splitting the dataset, training CNN models, and validating their accuracy. In the pretraining and predictions stage, pre-trained CNN models are used to achieve varying accuracies based on hyperparameters and training time. Finally, in the concatenated ensemble stage, prediction results from these models are combined, and six binary classifiers are

applied. The final prediction is made through a voting-based ensemble, incorporating K-fold cross-validation to mitigate model bias.

The proposed model can effectively detect anomalies in the WAAM process, namely, bead cut and balling effects. This model takes advantage of the 1D and 2D process signatures to detect the anomalies which is expected to enhance the predictive performance of the model. The bead cut anomaly signifies a separated bead formation caused by discontinuous deposition of a feeding material on a substrate, and the balling anomaly indicates an irregular bead surface contour owing to the separation of small spherical balls from the melt pool [19]. Moreover, data and ML models were readily integrated to overcome the limitations and difficulties in acquiring sufficient data and finding a near-optimal ML model. The proposed method is investigated and validated by taking advantage of the inexpensive and comprehensive datasets from a GTAW-based WAAM process. In contrast to the 1D and 2D CNN models’ accuracies of 81.6 % and 88.6 %, respectively, the data-fused, concatenated ensemble model obtained an accuracy of 98 %. The remainder of this paper is organized as follows. Section 2 discusses related work on the state of ML in AM. The data-fused, concatenated ensemble algorithm is proposed in Section 3, followed by the deliberation of its results and validations in Section 4. Finally, concluding remarks are presented in Section 5.

2. Related work

The studies conducted in the field of ML for anomaly detection can be categorized into four main categories: (1) non-real-time, data-driven ML methods, (2) non-real-time CNN-based methods, (3) real-time CNN-based methods, and (4) EL-based methods. This section discusses the performance and limitations of the state-of-the-art methods.

Non-real-time ML methods were applied to design experiments with large labeled datasets that used artificial neural networks (ANN) for bead geometry predictions and parameter optimizations. Manan introduces a Federated Learning (FL) approach with U-Net architecture for privacy and data availability, showcasing its superior defect detection compared to individual learning and highlighting FL’s promise in privacy-preserving collaborative ML for AM process control with a mean intersection over union of 0.807. [24]. Mahmoudi et al. devised a 99.6 % accurate anomaly detection framework for L-PBF AM by analyzing thermal signatures of melt pools, employing image segmentation, clustering, spatial statistics, and classification techniques. [25]. Tapia et al. create a Gaussian process-based model to predict porosity in metallic parts produced via selective laser melting, demonstrating its effectiveness through a case study on 17–4 PH stainless steel manufacturing. [26]. Khanzadeh et al. developed a porosity prediction method in additive manufacturing by analyzing melt pool characteristics using supervised machine learning, with K-Nearest Neighbor (KNN) achieving the highest accuracy (98.44 %) in classifying melt pools, outperforming traditional metrics and enabling broader applications in similar AM processes. [27]. Aminzadeh et al. develop a monitoring system for laser powder-bed fusion quality using Bayesian inference, creating a unique dataset for in-situ visual images of the process, which is then used to train a Bayesian classifier to detect defective build layers or regions in real-time with an accuracy of 89.5 % [28]. Poudel et al. use high-resolution X-ray computed tomography to analyze and classify volumetric defects in laser powder bed fused Ti-6Al-4 V by quantifying nine morphological parameters, achieving high accuracy in defect classification through a decision tree (>98 %) [29]. Moreno et al. present a three-stage approach using random forests to automatically classify porosity in metallographic data, achieving an accuracy of 94.41 % and a low out-of-bag error, demonstrating high precision for porosity classification in metallic additively manufactured components. [30]. Although these approaches effectively predict weld bead geometry, they are generally limited to using 1D process signatures (e.g., power and acoustic). Furthermore, they could not provide a scheme for real-time monitoring and control.

Non-real-time CNN-based methods use the capability of CNN to process images that tend to show good performance with offline images. For instance, Khumaidi et al. developed a Gaussian kernel-based classifier that could detect four defect types with an accuracy of 95.83 % [31]. Hou et al. implemented a three-stage deep neural network (DNN) that used the pre-processed GDxray data set [32] as input and a sliding window for the final classification, achieving an accuracy of 91.84 % (90.27 % precision and 92.78 % recall) [33]. Zhu et al. used a combination of CNN for high-level feature extraction and random forest (RF) for final classification, which achieved an accuracy of 98.75 % [34]. Park et al. used a two-stage model to classify non-pattern weld defects using 32,000 sample images, wherein the first stage, the region of interest (RoI) was detected, followed by the classification of these RoIs [35]. Jiang et al. used the improved pooling strategy and feature selection for weld defect recognition using the ReliefF algorithm, which achieved an accuracy of 91.71 % [36]. Liu et al. developed an anchor-free region-based convolutional neural network (AF-RCNN), which resulted in a prediction accuracy of 95 % compared to Faster-RCNN [37]. Although these algorithms utilize cutting-edge CNN techniques, they have not been investigated for real-time monitoring and control. Experimental samples were used to classify the weld defects and predict the bead geometry after welding.

In contrast, CNN models can be effective for the real-time monitoring and control of AM. Scime et al. proposed a multi-scale patch detector for anomaly detection fidelity that was trained with 70,000 images from a PBF process and attained 85 % and 93 % differentiation and detection accuracy, respectively [38]. Caggiano et al. suggested a bi-stream deep convolutional network for fault detection in PBF with an accuracy of 99.4 % [39]. Reisch et al. proposed an unsupervised multivariate predictor using 1D and 2D process signatures to reduce labeled dataset deficiency [40]. Lyu et al. developed a CNN model to extract features for a physically labeled dataset that contained 17,595 images and recorded an accuracy of 90.08 % [41]. Cho et al. proposed a real-time anomaly detection algorithm that used weld pool images as inputs and detected bead cuts and balling abnormalities with a prediction accuracy of 98 % [19]. Lee et al. developed a real-time CNN monitoring console that converts voltage images into time-series waveshape data to classify normal and abnormal beads [42]. Kim et al. developed CNN-based real-time monitoring that predicts anomalies for the WAAM process using molybdenum, consisting of three modules: image conversion, CNN prediction, and real-time tracking [20]. Despite the improved performance of these models, acquiring sufficient and reliable data and determining a near-optimal model proved time-consuming and cost-intensive. Thus, robust and flexible algorithms are necessary to overcome these limitations in the current CNN models.

EL methods can address the adequately labeled dataset deficiency challenge by implementing transfer learning. The main advantage of EL is that it can improve the performance of an ML base learner. Although some studies have included EL as the basis for anomaly detection, this area remains mostly unexplored in AM. For instance, Li et al. developed an EL model to predict the surface roughness in extruded AM processes comprising RF, AdaBoost, CART, SVR, RR, and RVFL as the base learners, which showed better accuracy than the ensemble [43]. Khan et al. developed a RF classifier model for layer-wise monitoring in metal additive manufacturing, achieving a 99.98 % detection accuracy for anomalies, which were validated using optical tomography (OT) imaging and correlated with defects from computerized tomography (CT) data. [44]. Zhang et al. developed an enhanced ensemble learning prediction model for predicting the yield stress of lattice structures in additive manufacturing by using data from finite element simulation, integrating a Boosting module and feature transformation methods to improve prediction accuracy (R-squared of 0.844) and generalization, reducing preparation time and testing costs, and offering valuable insights for industrial inspection and evaluation of such structures. [45]. Li et al. introduced a cost-efficient EL approach (Bagging of Trees, Gradient Boosting, Random Forest) with accuracies around 99 % using synthetic

3D point clouds to detect defects in additively manufactured objects, outperforming existing methods and demonstrating the applicability of their scheme to in-situ defect detection during additive manufacturing with the aid of 3D data acquisition [46]. However, these studies mainly used rudimentary EL models that operated 1D process signatures. Thus, a robust and flexible EL algorithm that uses both 1D and 2D process signatures and multiple classifiers simultaneously can be a better design for prediction.

3. Proposed methodology

Fig. 1 shows the proposed data-fused, concatenated EL algorithm that can flexibly and robustly detect anomalies from insufficient datasets, less accurate ML models, and classifiers. The procedure for implementing the proposed algorithm in the WAAM process is discussed in this section. It consists of four stages: data pre-processing, data preparation, pretraining and predictions, and concatenated ensemble detailed as under:

- **Step 1 (Data pre-processing):** First, voltage data and WAAM videos were collected from specially designed experiments. Second, the voltage data were converted into voltage image data, and the WAAM video was partitioned into 2D frame data. Third, these data were resized and cleaned according to the requirements for uniformity. Fourth, both data types were labeled with predefined classes. As a result, the raw WAAM data were converted into 1D and 2D pre-processed data.
- **Step 2 (Data preparation):** Each pre-processed voltage image data was synchronized to the corresponding 2D frame data to prepare a unified dataset. The unified dataset was consequently divided into training and test data with a ratio of 90:10, respectively. 30 % of the training data was set aside for the validation dataset. With the remaining 70 % of training data, the CNN models were trained. The validation data with matching labels was used to check the model's training accuracy. On the other hand, the test data without labels was used to test the models to obtain the predicted labels. The predicted labels thus were compared to the actual test labels to determine the final accuracy of the models.
- **Step 3 (Pretraining and predictions):** Three pre-trained CNN models were individually trained from the voltage dataset, whereas four CNN models were trained from the 2D dataset. The models can achieve different accuracies depending on the hyperparameters and train time. For investigation, four ranges of accuracies were selected for each model, and their corresponding weights were preserved.
- **Step 4 (Concatenated ensemble):** The prediction results of the pre-trained CNN models for 1D and 2D data were concatenated together, forming a vector of probabilities. Six distinct binary classifiers were employed in this stage that uses the concatenated probability vector as the input. Among the six classifiers, the four most accurate classifiers were selected for the final prediction using a voting-based ensemble. Second-order voting was also performed to select the final concatenated ensemble prediction outcome. K-fold cross-validation was incorporated to reshuffle the dataset into train and validation to avoid model bias.

3.1. Step 1: Data pre-processing

A GTAW-based WAAM process was used to deposit weld beads on a low-carbon steel substrate fed from a wire feeder; the feeder used Inconel 625 wire to create the weld beads. The experimental setup consisted of a Fanuc ARC Mate 120ic robot arm with a Fanuc R-30iA controller, tungsten inert gas (TIG) torch, and Miller Dynasty 400 GTA welding power source. The detailed setups can be found in previous studies [19,20,47]. The following stages describe the entire data-preparation process, as shown in Fig. 4.

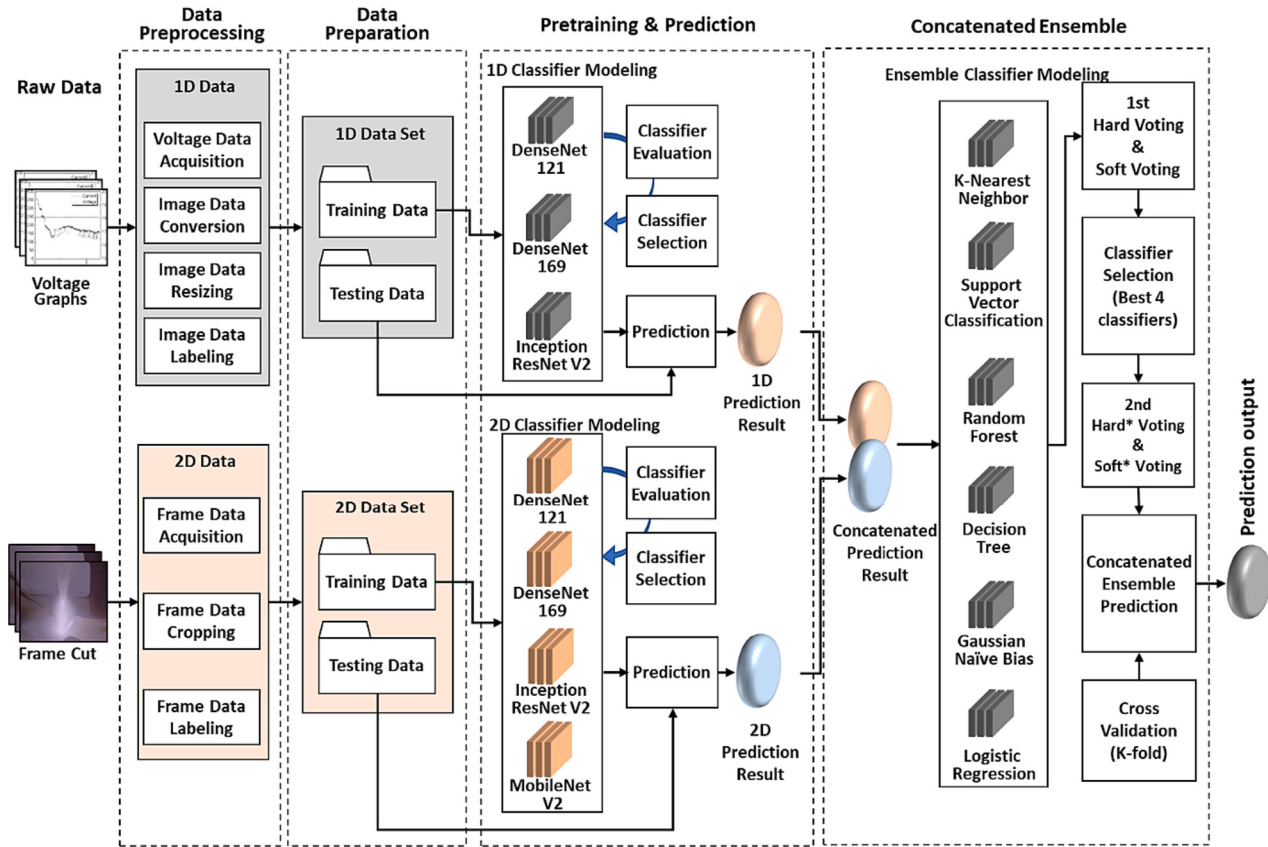


Fig. 1. Implemented data-fused, concatenated EL algorithm.

Bead deposition on the substrate was initially designed as an experimental trial controlled by experimental parameters. In this case, current (A), wire feed rate (cpm), and welding/travel speed (cpm) were considered variable parameters. Two process parameters—a Wire feed rate of 70–300 cm per minute (cpm) and a travel speed of 10–100 cpm with steps of 25 and 10, respectively—were changed in order to create the experiments. A constant current of 200 A (A) was used. As indicated in Table 1, each set of settings created 100 distinct trials for each material. As a result, 300 testing for the three materials LCS, STS, and INC were conducted. The experimental setup's design is seen in Fig. 2. The central composite design (CCD) suggested by the National Institute of Standards and Technology (NIST) Handbook was used to design the experiments. Experimental parameters, such as shield gas, which used 70 % argon and 30 % helium, and arc length set to 5 mm, were kept constant throughout the experiment. The detailed experimental conditions can be found in Lee et al. [47].

The creation of the voltage dataset consisted of four steps: (1) voltage data acquisition, (2) voltage image data conversion, (3) voltage image

data resizing, and (4) voltage image data labeling. Each step is described in detail below:

- **Voltage data acquisition:** The voltage data were measured by the Miller Insight ArcAgent Auto current voltage sensor and collected by the Miller Insight Centerpoint current voltage interface. These data were collected as numerical time-series data at a 1 kHz sampling rate that was stored in a .txt file. All 25 trials generated 25 .txt files.
- **Voltage Image data conversion:** Each voltage data point was plotted as a graph showing the change in voltage values with respect to time, which essentially converted the numerical data to voltage image data. This process was performed because feature extractions become difficult in numerical form owing to extensive fluctuations. In comparison, image-based detection often employs advanced image classification networks such as CNN. The voltage image data represents a chart displaying variations in voltage values over time, reflecting the time-series nature of the voltage data. The voltage values were first defined to perform image conversion. Suppose, v_m be the m^{th} voltage value for a particular time instant T . Hence when $T = 1/1000$, the first voltage is observed that is v_1 . Hence, v_m is the voltage value observed at $T = m/1000$. Assuming $V_{w,i,k}$ to be the k^{th} set of voltage values, i the interval between the current and the next snapshot, and w the bandwidth defined as the duration between the initial and final time point in a particular snapshot, then $V_{w,i,k}$ can be defined as Eq. 1. For instance, $V_{3,1,1}$ indicates 3000 voltages from $T = (1/1000) \rightarrow (3000/1000)$, similarly, $V_{3,1,2}$ and $V_{3,1,3}$ indicate 3000 voltages from $T = (1001/1000) \rightarrow (4000/1000)$ and $T = (2001/1000) \rightarrow (5000/1000)$, respectively. The bandwidth and interval must be decided logically because too small a bandwidth or interval will result in an overlap in the values of v_m . In this case, the bandwidth and intervals are maintained at 3 and 0.5, respectively.

Table 1
Process parameters and bead numbers considered for the experiments.

Wire feed rate (cpm)	Travel speed (cpm)									
	10	20	30	40	50	60	70	80	90	100
75	1	11	21	31	41	51	61	71	81	91
100	2	12	22	32	42	52	62	72	82	92
125	3	13	23	33	43	53	63	73	83	93
150	4	14	24	34	44	54	64	74	84	94
175	5	15	25	35	45	55	65	75	85	95
200	6	16	26	36	46	56	66	76	86	96
225	7	17	27	37	47	57	67	77	87	97
250	8	18	28	38	48	58	68	78	88	98
275	9	19	29	39	49	59	69	79	89	99
300	10	20	30	40	50	60	70	80	90	100

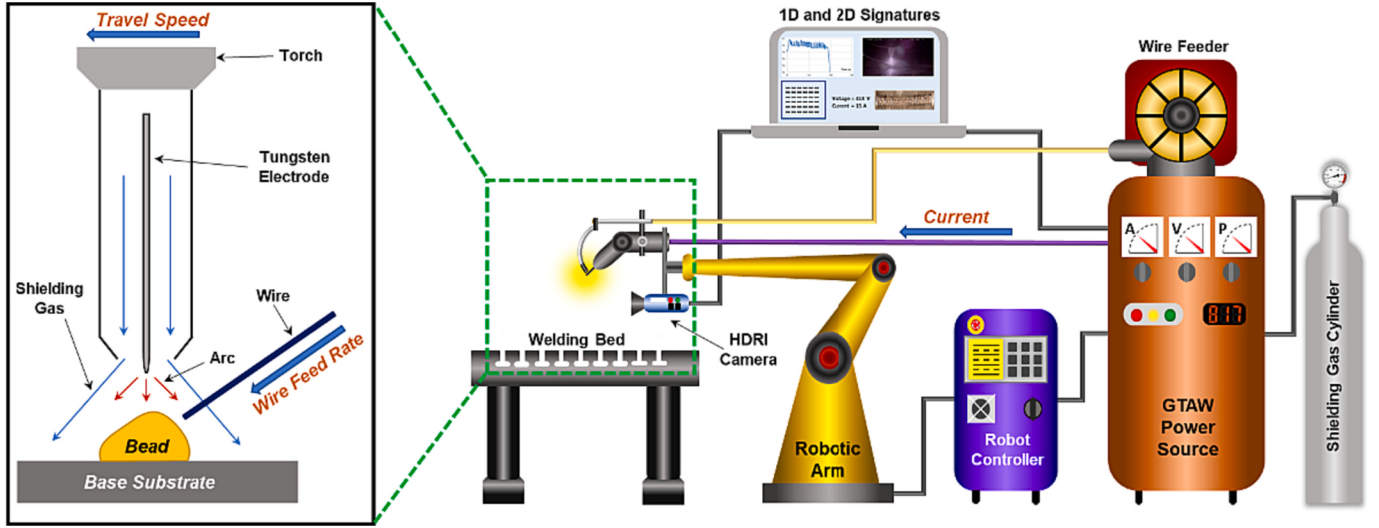


Fig. 2. Gas tungsten arc welding-based WAAM experimental setup.

$$V_{w,i,k} = \{v_m : m \in M_{w,i,k}\} \quad (1)$$

where $M_{w,i,k}$ is the set of m instances, and consequently, can be denoted by Eq. 2.

$$M_{w,i,k} = \{x + (k-1)(1000i) : x \in \mathbb{N}, 1 \leq x \leq 1000w\} \quad (2)$$

where x is a certain time point.

Fig. 3 pertains to $V_{w,i,k}$ with $w = 3$ and $i = 0.5$. In this context, the horizontal and vertical axes represent T and v_m , respectively, at time T . Since $w = 3$, each element of $V_{w,i,k}$ encompasses a set of 9000 voltage values. For example, $V_{3,0.5,1}$ is the initial dataset and encompasses $v_1, v_2, \dots, v_{9000}$. Given $i = 0.5$, $V_{3,0.5,2}$ comprises $v_{1501}, v_2, \dots, v_{10500}$. It's noteworthy that any two consecutive voltage value datasets share 7500

voltage values, owing to the parameters $w = 3$ and $i = 0.5$.

- **Voltage Image data resizing:** Upon defining the $V_{w,i,k}$ values, each $V_{w,i,k}$ undergoes transformation into voltage image data, representing time-series waveforms. Notably, each $V_{w,i,k}$ component exhibits varying minimum and maximum v_m values, leading to differences in image sizes based on these extrema. However, since CNN necessitates uniform image sizes so that, features can be extracted effortlessly, adjustments are made to the vertical axes of the images to ensure that they all share the consistent dimensions of 224×224

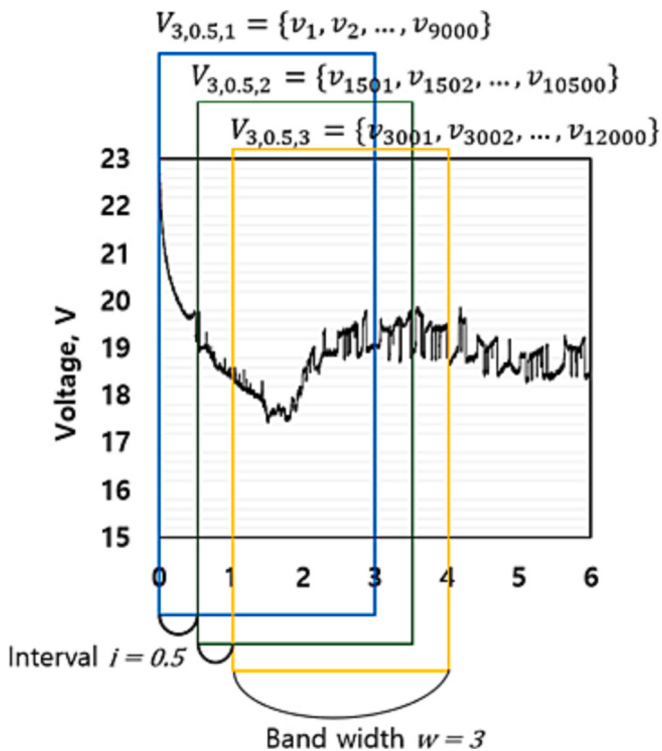
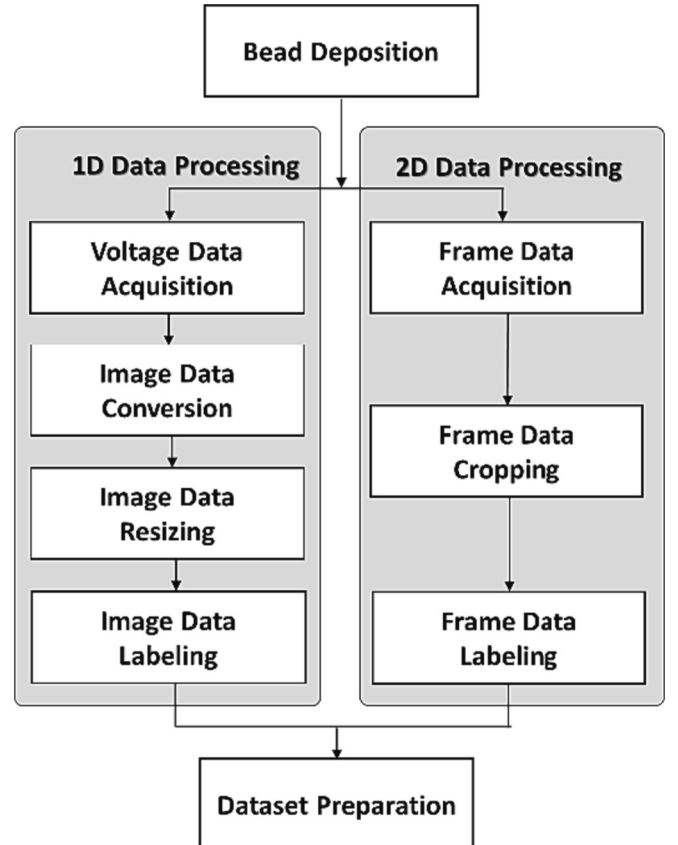
Fig. 3. Voltage image data conversion process with $w = 3$ and $i = 0.5$ [20].

Fig. 4. Steps of 1D and 2D data pre-processes for the dataset preparation.

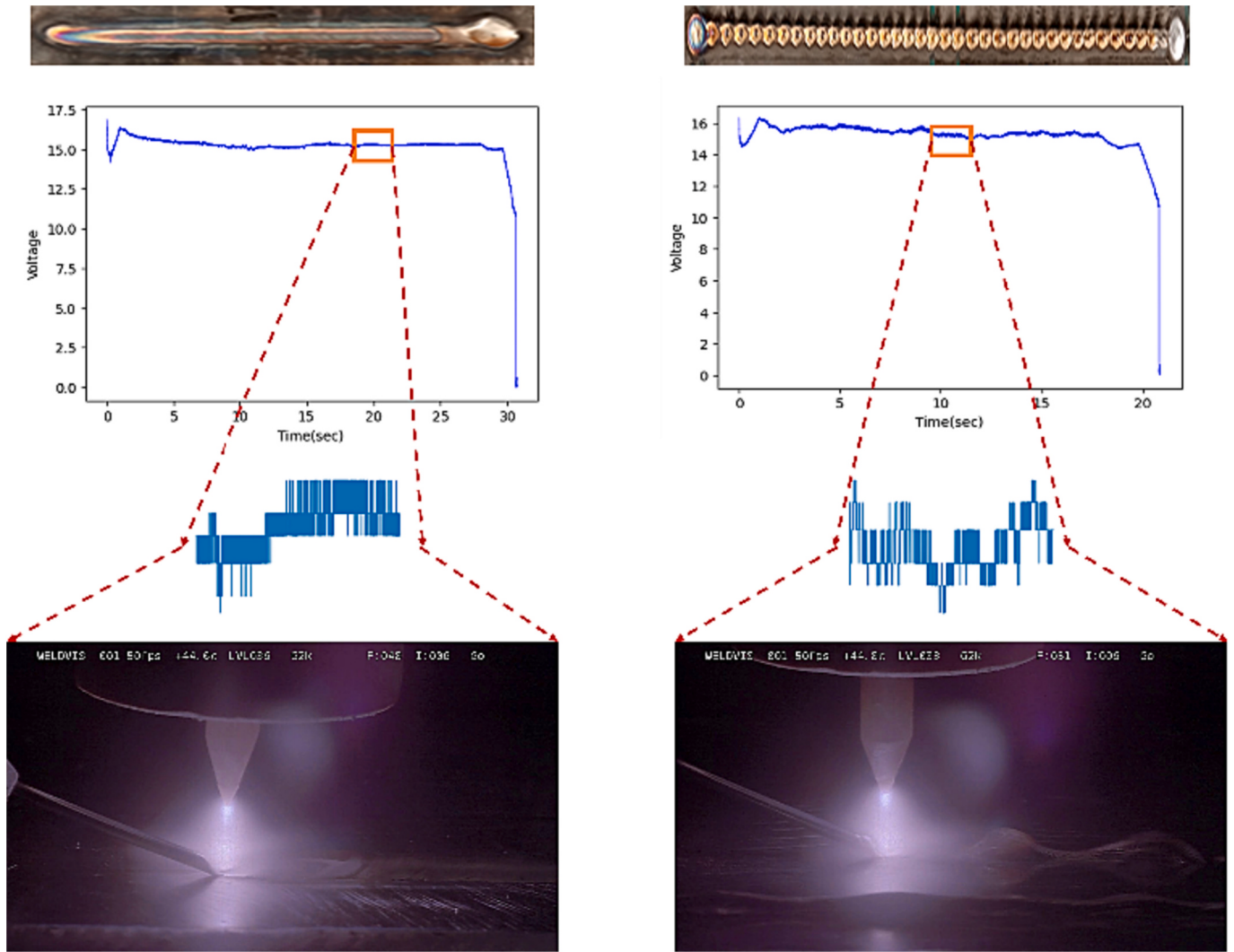


Fig. 5. Voltage and HDR images in the cases of normal (left) and abnormal (right) beads.

$$\begin{bmatrix}
 \text{Img}_1 & l_1 & S_{\text{class Normal}_1}^{1D} & S_{\text{class Abnormal}_1}^{1D} & S_{\text{class Normal}_1}^{2D} & S_{\text{class Abnormal}_1}^{2D} \\
 \text{Img}_2 & l_2 & S_{\text{class Normal}_2}^{1D} & S_{\text{class Abnormal}_2}^{1D} & S_{\text{class Normal}_2}^{2D} & S_{\text{class Abnormal}_2}^{2D} \\
 \text{Img}_3 & l_3 & S_{\text{class Normal}_3}^{1D} & S_{\text{class Abnormal}_3}^{1D} & S_{\text{class Normal}_3}^{2D} & S_{\text{class Abnormal}_3}^{2D} \\
 & & & \vdots & & \\
 \text{Img}_n & l_n & S_{\text{class Normal}_n}^{1D} & S_{\text{class Abnormal}_n}^{1D} & S_{\text{class Normal}_n}^{2D} & S_{\text{class Abnormal}_n}^{2D}
 \end{bmatrix}$$

Fig. 6. Concatenated predictions.

pixels. The size of 224×224 was selected primarily for two reasons. First, most CNN architectures for pre-trained models uses this size for feature extraction and classification process. Second, it is considered an optimal size that sufficiently contains time and spatial context of an image for the purpose of prediction. Hence, resizing should always be done such that no meaningful information is lost from the concerned images, which was done by adaptive resizing for each $V_{w,i,k}$ snapshot. Adaptive resizing was done by setting a range, that is, $V_{\text{range}} = V_{\text{max}} - V_{\text{min}}$. Ranges for each snapshot are variable in nature. If a snapshot has a small range compared to another, it automatically re-adjusts to incorporate the larger voltage value, maintaining the exact image size.

- **Voltage Image data labeling:** All the image snapshots were labeled manually by two professionals using their expert judgment into three distinct classes Normal, Abnormal, and Unclassified. Beads with stable and unimpeded voltage patterns were classified as Normal, whereas fluctuating and irregular data patterns were labeled as Abnormal. Unclassified types were patterns that started as normal but transitioned to abnormal or vice-versa. Any bead deposition's starting and ending points were also labeled as unclassified because the welding speed was zero in those positions. For proper training, the output labels need to be balanced. If a particular class of data becomes very high for some reason, the network tends to get biased toward that class. This can be avoided by implementing under-

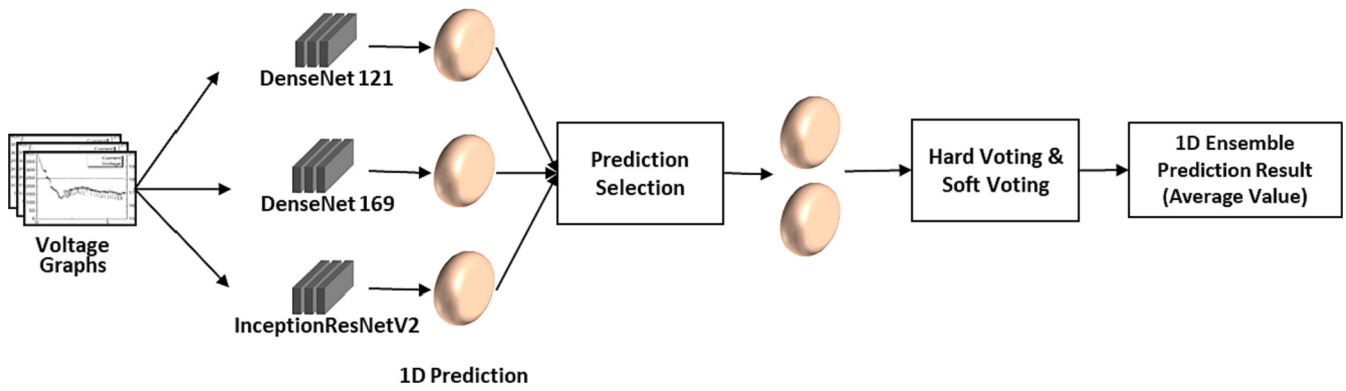


Fig. 7. 1D model ensemble.

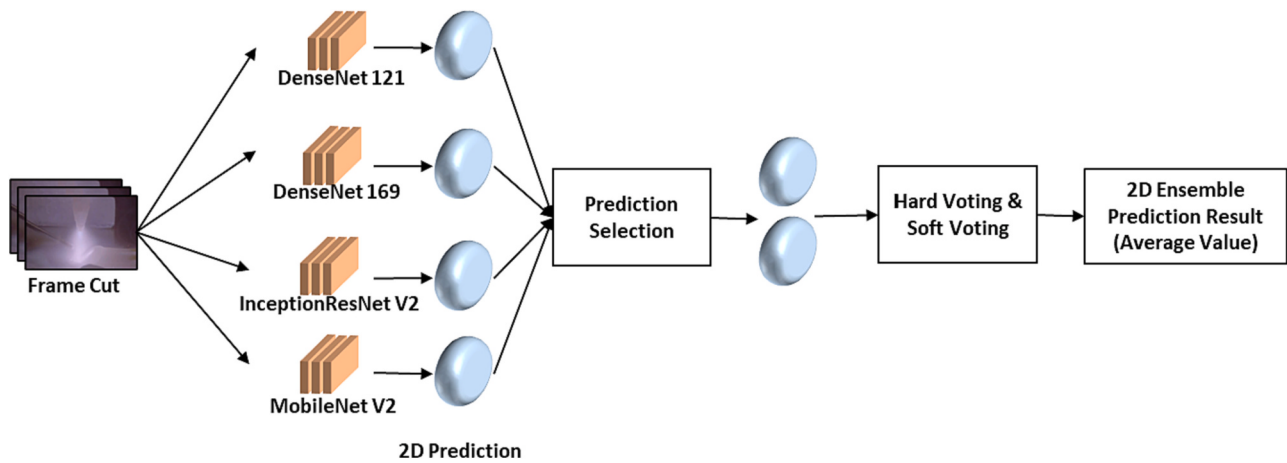


Fig. 8. 2D model ensemble.

sampling, which reduces the size of the majority class to balance the minority class [50]. The final voltage image data set was prepared with only image snapshots labeled as Normal and Abnormal, discarding the unclassified class.

The 2D datasets consisted of three steps: (1) frame data acquisition, (2) frame data cropping, and (3) frame data labeling. Each step is described in detail below:

- **Frame data acquisition:** The WAAM process was captured by a Weldvis WL2-H7ML-M35 high dynamic range (HDR) camera at a sampling rate of 50 frames per second (*fps*), tracking the movement of the TIG torch from start to end. In this case, each DoE generated one *mp4* file that is partitioned into *.jpg* while maintaining the same sampling rate. This *.jpg* frame data are subsequently cleansed, disposing of any image obstructed by shadow, smoke, or any form of signal noise.
- **Frame data cropping:** Cropping was performed on frame images to focus on the presently associated melt-pool image, dismissing any form of noise and irregularities in the raw data. Each frame cut was obtained as an image of size 224×224 pixels. Because the 224×224 image contains unnecessary areas and noises unrelated to the anomaly detection in the WAAM process, cropping was performed to remove these. Cropping was performed based on the average coordinates obtained using the You Only Look Once (YOLO) model, which is an object detection technique.
- **Frame data labeling:** Data labeling for frame data was performed in a manner similar to that for voltage data. Two professionals also labeled the frame data with an unbiased expert perception. Each

frame was labeled as normal, abnormal, or unclassified. In the normal case, the shape of the bead made through WAAM is normal, whereas in the abnormal case, the shape of the bead is cut (e.g., balling effect) or thin. The unclassified images include several issues, such as the mixing of normal and abnormal images, preheating, and processing start/stop images.

3.2. Step 2: Data preparation

The final dataset was prepared by synchronizing both the image and frame data to a single metadata set. Synchronization was performed after each type of voltage data image (1D) and frame (2D) data were individually labeled. Because each frame data is sampled every 1/50 seconds and the voltage data are sampled at 1/1000 s, each frame data consists of 20 voltage values which was synchronized using considering each time stamp of the frame data. Hence, any form of latency was avoided all together. All 20 voltage values were labeled as those of the frame data label. The same was also the case for the voltage image data, where the each data were labeled as normal only if all the voltage values corresponding to the voltage image data were labeled normal. The same was the case for the abnormal class; otherwise, the voltage image data are removed from the dataset. In doing so, it was found that the voltage image data label tended to coincide with the corresponding frame data label. The total number of frame cuts used in this process was 29,089. Among them, 10,032 frame cuts were classified as normal, whereas 12,596 and 6461 frame cuts were classified as abnormal and unclassified, respectively. The total number of data points in the training sets was 25,666, consisting of normal (8853), abnormal (11,136), and unclassified (5677) images. The test set consisted of 3423

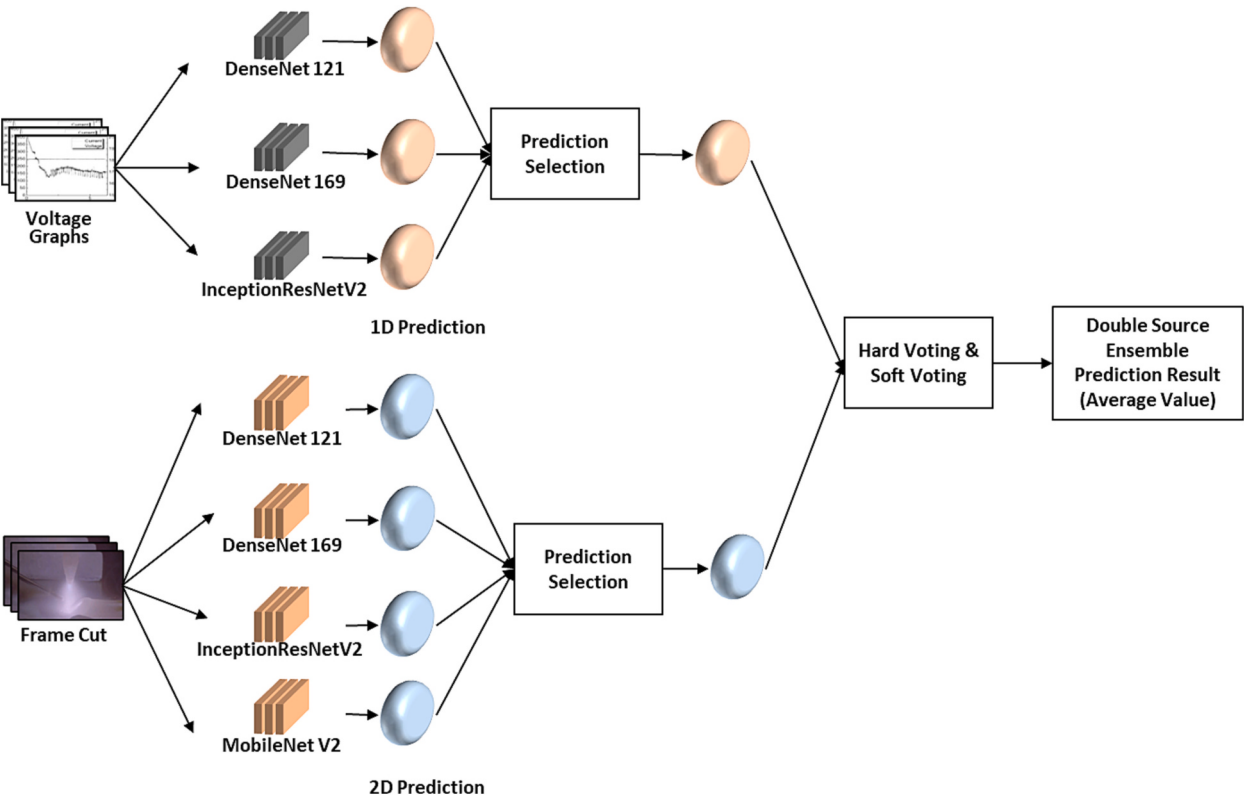


Fig. 9. Data fusion EL.

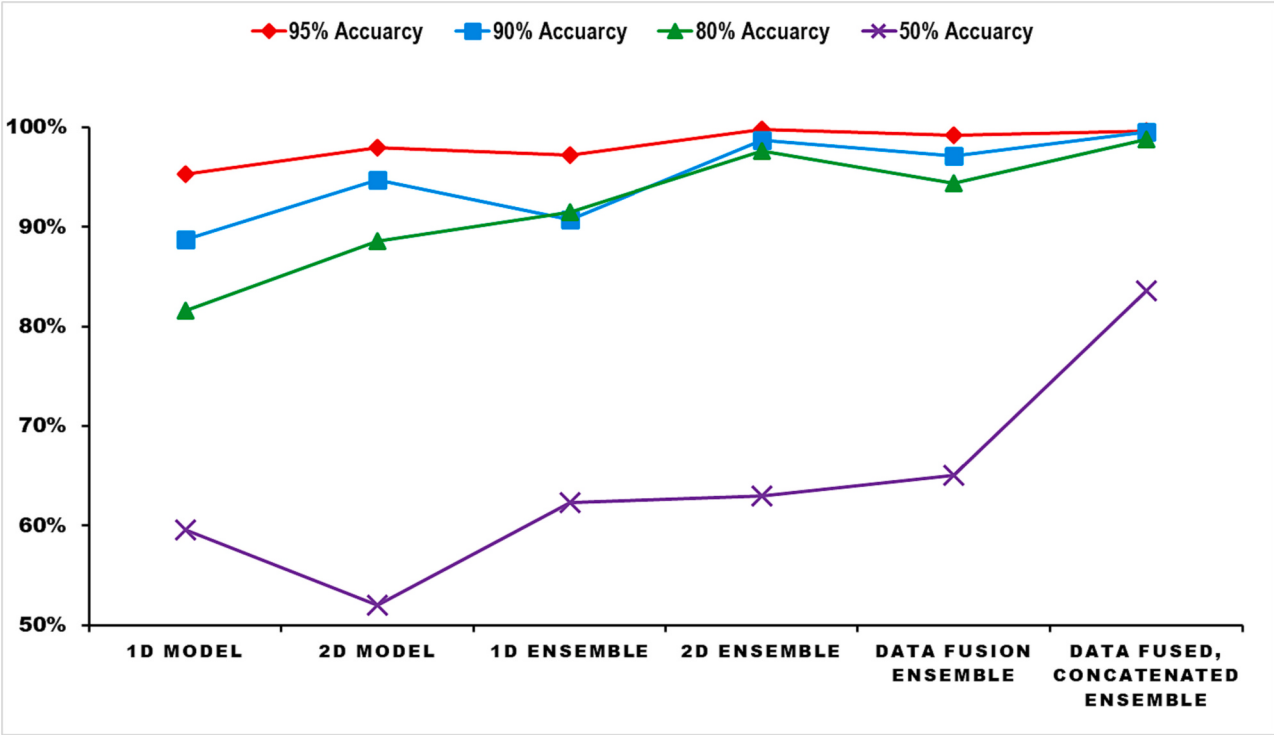


Fig. 10. Accuracy of 1D CNN, 2D CNN, 1D ensemble, 2D ensemble, Data fusion ensemble, and data-fused, concatenated ensemble.

images: normal (1179), abnormal (1460), and unclassified (784) images.

Fig. 5 shows two typical examples of “Normal” and “Abnormal” classes of input metadata. In the case of the metadata labeled normal,

the bead was found to be stable and monotonic, and the voltage image pattern was smooth and unfluctuating. The 2D frame data also show no signs of balling or bead-cut defects. In contrast, for the abnormal class of metadata, the bead formed was unstable and full of irregularities, the

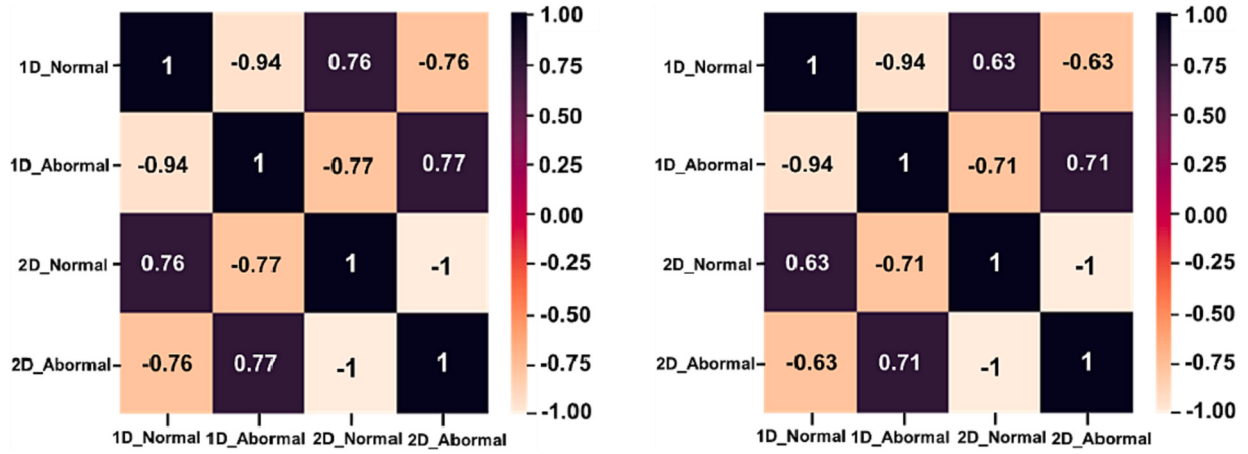


Fig. 11. Correlation between input variables of data set through concatenation using low-accuracy (right) and high-accuracy (left) models.

voltage (1D) profile showed significant fluctuations, and the 2D frame data showed signs of balling.

3.3. Step 3: Pretraining and predictions

The difference between the proposed and previous methods is that data fusion was used with two types of data (i.e., voltage images and 2D HDR frame cuts). In contrast, existing methods use single data in an ensemble architecture. Prior to the ensemble, the CNN models were trained to create a new input variable. From the voltage data set, DenseNet121, DenseNet169, and InceptionResNetV2 were individually trained. DenseNet achieved the highest prediction accuracy and prediction time [20]. CNN models, such as DenseNet121 (DN121), DenseNet169 (DN169), MobileNetV2 (MN2), and InceptionResNetV2 (IRV2), were pretrained on a large dataset consisting of numerous images and output classes for the 2D dataset. MobileNetV2 for 2D data showed the best accuracy with the minimum prediction time [19]. In addition to storing the highest accuracy weights (e.g., 99 %), weights of 50 %, 80 %, 90 %, and 95 % were also preserved for comparison and validation.

3.4. Step 4: Concatenated ensemble

This step aimed to create a new dataset that could be used for the concatenated ensemble for better prediction. The CNN models in the previous steps used voltage (1D) images and 2D frame cuts as the input datasets for classification. The concatenated dataset defined in this subsection is different because instead of using images, it uses numeric values for prediction. This resulted in two inherent advantages. First, because computation on numeric data is far more relaxed than that of images, it substantially reduces the processing time. Second, the sheer amount of data in the voltage image dataset is much more than the numeric dataset, which is also computationally taxing.

In the previous steps, it was seen that voltage (1D) images and 2D frame cuts data provide the probability of certain data occurring in a particular class rather than giving the final classification. Once the pretrained model is completed, a set of these probabilities is obtained for each image. Then, an ensemble is performed based on the prediction probability through CNN. The probability that a specific dataset belongs to a particular class was calculated. The dataset was used to create a new link matrix, that is, the concatenated dataset. A concatenated dataset was created using a concatenated matrix of these probabilities. Suppose a particular image Img_1 has a set of voltage images probabilities as $S_{1D} = \{S_{class Normal}^{1D}, S_{class Abnormal}^{1D}\}$ and the corresponding 2D voting probabilities as $S_{2D} = \{S_{class Normal}^{2D}, S_{class Abnormal}^{2D}\}$ and if the original class label of Img_1 is I_1 then the concatenated data point for Img_1 will be,

$$[Img_1 \ I_1 \ S_{class Normal}^{1D} \ S_{class Abnormal}^{1D} \ S_{class Normal}^{2D} \ S_{class Abnormal}^{2D}]$$

Consequently, for the other images, the dataset follows the format shown in Fig. 6, where all the soft voting probabilities become the input of the new dataset.

$$\begin{bmatrix} Img_1 & I_1 & S_{class Normal}^{1D} & S_{class Abnormal}^{1D} & S_{class Normal}^{2D} & S_{class Abnormal}^{2D} \\ Img_2 & I_2 & S_{class Normal}^{1D} & S_{class Abnormal}^{1D} & S_{class Normal}^{2D} & S_{class Abnormal}^{2D} \\ Img_3 & I_3 & S_{class Normal}^{1D} & S_{class Abnormal}^{1D} & S_{class Normal}^{2D} & S_{class Abnormal}^{2D} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ Img_n & I_n & S_{class Normal}^{1D} & S_{class Abnormal}^{1D} & S_{class Normal}^{2D} & S_{class Abnormal}^{2D} \end{bmatrix}$$

Because the probability is concatenated, it is based on soft voting cases. For instance, for the same Img_1 if the original label is I_1 and the probability of belonging to the normal class; the abnormal class is $S_{class Normal}^{1D}, S_{class Abnormal}^{1D}$ for the voltage image data, and the probability of belonging to the normal class, the abnormal class is $S_{class Normal}^{2D}, S_{class Abnormal}^{2D}$ for the 2D data. The dataset is a concatenation of the results of different data types. This newly created dataset can be used for concatenating ensembles. As seen above, concatenation also creates synchronization between the two data types.

The concatenated ensemble is based on the stacking technique, where a group of base learners is trained, and the stacking result of the base learner, instead of showing the final classification, is driven as input to another group of classifiers. The output of these classifiers was considered as the final classification. Note that the ensemble uses several classifiers after concatenating the prediction results and proceeds with the ensemble, whereas the stacking method uses a classifier after stacking the results for the final prediction.

In this study, the voltage image data and 2D frame cut data were the initial inputs that were then passed on to the CNN networks, which functioned as base learners. The output of the CNN was then concatenated to a new dataset and driven to the next layer of the ensemble, which in our case are six distinct ML classifiers: logistic regression (LR), k-nearest neighbor (KNN), decision tree (DT), Gaussian naive bias (GNB), support vector classification (SVC), and random forest (RF). These classifiers were chosen because they tend to exhibit better performance in binary classification. After classification, the results of the top four classifiers with high accuracy were selected from among the six. The voting-based ensemble was performed on the results of the selected classifier to derive the final prediction class and measure the accuracy.

We compared four models: (1) Voltage image ensemble model and Voltage image CNN model, (2) Frame cut image ensemble model and

Frame cut image CNN model, (3) a data fusion ensemble that uses voltage image data and 2D frame cut image data simultaneously, and (4) the proposed data fused, concatenated ensemble. In addition, the data-fused, concatenated ensemble model was verified by the k-fold cross-validation method, and the value of k was set to 10. When all four models were compared, it was observed that the increase in accuracy of the proposed method, that is, the data-fused, concatenated ensemble, was the highest.

4. Results and validation

In this section, we compared the results obtained from four model configurations: single source-based CNN: 1D (voltage image)-based CNN and 2D (frame cut image)-based CNN, and single source-based ensemble: 1D (voltage image) model ensemble and 2D (frame cut image) model ensemble, 1D (voltage image)-2D (framecut image) double source-based ensemble (Data-fused ensemble), and the proposed 1D (voltage image)-2D (frame cut image) double source-based concatenated ensemble (Data-fused, concatenated ensemble). From here onwards, each model will be referred to by the name in the parentheses for ease of nomenclature: 1D-based CNN (1D-CNN), 2D-based CNN (2D-CNN), 1D model ensemble (1D-Ensemble), 2D model ensemble (2D-Ensemble), 1D–2D double source-based ensemble (Data-fused Ensemble), and 1D–2D concatenated ensemble (Data-fused, concatenated ensemble). As mentioned earlier, the models can perform at different accuracies by tuning the hyperparameters. Hence, for ease of contrast, the models were divided into four ranges of accuracies:

- 0 % – 60 % referred to as ≈ 50 %
- 60 % – 89 %, referred to as ≈ 80 %
- 90 % – 95 % referred to as ≈ 90 %
- 95 % – 100 % referred to as ≈ 95 %

The development environment used Google's Colab pro-environment, including GPU T4, RAM 25.51 GB, CPU Intel(R) Xeon(R) CPU @ 2.30GHz. The Google TensorFlow library defined the CNN models for training and testing.

4.1. Ensemble models for comparisons

Each procedure and its accuracy results for the 1D-Ensemble, 2D-Ensemble, and Data Fusion Ensemble will be explained.

4.1.1. 1D model ensemble

Fig. 7 shows the 1D ensemble model. Here the voltage image data are individually classified using the three pre-trained CNN models: DenseNet121 (DN121), DenseNet 169 (DN169), and InceptionResNetV2 (IRV2). Consequently, the predicted outcomes of each model are equated through a voting classification approach for hard and weighted

soft voting as defined in Eqs. 3–5. Two models are considered at a time for the voting classification, and the subsequent accuracy for each voting class is shown in Table 2.

For Eqs. 3–5, suppose there are T base learners and N classes, and the base learner obtains $\{h_1, h_2, \dots, h_T\}$ as the probability value for $h_i(x)$. In the case of Eq. 3, if the majority predicts that the results of the classifiers $h_i^N(x)$ is the class of C_j , it becomes the output class of the base learner. By contrast, the probability itself is compared, and classification is performed in the class with a higher probability value in Eq. 4. In the case of Eq. (5), the classification progresses by adding the weights in Eq. 4.

$$H(x) = \begin{cases} C_j, & \text{if } \sum_{i=1}^T h_i^j(x) > 0.5 \sum_{k=1}^N \sum_{i=1}^T h_i^k(x) \\ \text{reject, otherwise} \end{cases} \quad (3)$$

$$H(x) = C_{\arg, \max} \sum_{i=1}^T h_i^i(x) \quad (4)$$

$$H(x) = C_{\arg, \max} \sum_{i=1}^T w_i h_i^i(x) \quad (5)$$

The final predictions of each single and double-source ensemble model were made using these hard and weighted soft voting classifiers. As shown in Table 2, seven distinct voting conditions were used. The first two conditions are based on hard voting classification. In both Condition 1 and Condition 2, when the individual classifications were the same, the final classification followed the individual classification. For instance, if classification and classification* both predict “normal” then the final prediction is also normal. However, when both the models' classification differs, then in Condition 1, the final prediction followed the prediction of ‘classification*’ in Condition 2, followed the prediction of ‘classification’. Conditions 3–7 were based on a soft voting approach. Suppose that the probability of a model classifying an instant as “normal” is $P(N)$ and that of another model is $P(N^*)$. The soft voting approach takes the aggregate of $P(N)$ and $P(N^*)$ multiplied by a pre-defined weight (that is different for each condition) and compares it to 0.5, which, if greater, gives the final prediction of ‘normal’ otherwise ‘abnormal’.

In Table 3, the 1D-Ensemble results are shown in terms of the seven voting conditions. The best accuracy is increased for the ensemble pairs marked as the 1D model and 1D model*. The underlined cells in the table indicate the best accuracy.

4.1.2. 2D model ensemble

The overall classification process of the 2D model ensemble is similar to that of the 1D ensemble model. Although in this case, instead of voltage image data, 2D frame cut data was used for classification by four pre-trained CNN models, namely: DenseNet121 (DN121), DenseNet 169 (DN169), InceptionResNetV2 (IRV2), and MobileNetV2 (MNV2), as shown in Fig. 8. The same voting technique (given by Eqs. 3–5) and

Table 2
Voting conditions for EL.

Case	Classification	Classification*	Final prediction
Condition 1	Normal	Normal	Normal
	Normal	Abnormal	Normal
	Abnormal	Normal	Normal
	Abnormal	Abnormal	Abnormal
Condition 2	Normal	Normal	Normal
	Normal	Abnormal	Abnormal
	Abnormal	Normal	Abnormal
	Abnormal	Abnormal	Abnormal
Condition 3 ($w_{1D} = 0.4, w_{2D} = 0.6$)			
Condition 4 ($w_{1D} = 0.45, w_{2D} = 0.55$)	$P(\text{Normal})$	$P(\text{Normal}^*)$	if $(P(N) \times w_{1D} + P(N^*) \times w_{2D}) \geq 0.5$ →normal else→abnormal
Condition 5 ($w_{1D} = 0.5, w_{2D} = 0.5$)			
Condition 6 ($w_{1D} = 0.55, w_{2D} = 0.45$)			
Condition 7 ($w_{1D} = 0.6, w_{2D} = 0.4$)			

Table 3
1D model EL results.

Accuracy	1D model	1D* model	1D acc	1D* acc	Voting Conditions						
					1	2	3	4	5	6	7
≈ 95 %	DN121	IRV2	0.96	0.92	<u>0.95</u>	0.93	0.94	0.94	0.93	0.92	0.92
	DN121	DN169	0.96	0.98	<u>0.97</u>	0.97	0.98	<u>0.99</u>	0.98	0.98	0.98
	DN169	IRV2	0.98	0.92	<u>0.98</u>	0.92	0.97	0.96	0.95	0.93	0.93
Average			0.97	0.94	<u>0.97</u>	0.94	<u>0.97</u>	0.94	0.97	0.96	0.95
≈ 90 %	DN121	IRV2	0.90	0.87	0.91	0.87	<u>0.91</u>	0.91	0.90	0.88	0.88
	DN121	DN169	0.90	0.89	0.88	0.91	<u>0.91</u>	0.91	<u>0.91</u>	0.90	0.90
	DN169	IRV2	0.89	0.87	0.89	0.87	0.90	<u>0.90</u>	0.90	0.89	0.88
Average			0.9	0.88	0.90	0.88	0.89	0.88	0.91	0.91	0.90
≈ 80 %	DN121	IRV2	0.86	0.71	0.66	<u>0.90</u>	0.84	0.81	0.72	0.71	0.71
	DN121	DN169	0.86	0.88	0.86	0.88	0.91	<u>0.92</u>	0.92	0.90	0.90
	DN169	IRV2	0.88	0.71	0.71	0.88	0.90	0.90	<u>0.92</u>	0.73	0.72
Average			0.87	0.76	0.74	0.89	0.88	0.88	0.85	0.78	0.78
≈ 50 %	DN121	IRV2	0.56	0.64	<u>0.64</u>	0.56	0.56	0.56	0.59	0.63	0.63
	DN121	DN169	0.56	0.58	<u>0.58</u>	0.56	0.56	0.56	0.56	0.58	0.58
	DN169	IRV2	0.58	0.64	<u>0.64</u>	0.58	0.59	0.59	0.62	0.63	0.64
Average			0.57	0.62	0.57	0.62	0.62	0.57	0.57	0.57	0.59

voting conditions (given in Table 2) were employed to determine the final prediction of the 2D model ensemble.

Table 4 provides the accuracy comparison of the 2D ensemble model of the six ensemble pairs for the four CNN models. Similar to the 1D model ensemble, seven voting condition cases, as shown in Table 2, were used to determine the output classification probabilities. The best-case accuracies ranged from 0.44 to 1.00 for the ensemble pairs. Compared with the 1D single-source ensemble, the results were reasonably improved (Table 4).

4.1.3. Data fusion ensemble (1D and 2D double source-based ensemble)

The data fusion ensemble, otherwise called the double source ensemble, uses both voltage (1D) images and 2D frame cuts together for the ensemble. First, the CNN model was trained for each data type. Three models were trained on the voltage image dataset, whereas four models were trained on the 2D dataset. But instead of ensembling pairs of CNN models trained using the same data type, ensemble pairs were

created with CNN models trained using different data types, as shown in Fig. 9. This resulted in 12 combinations of ensemble pairs. Comparable to the single-source ensemble, each pair's prediction result was determined using the voting conditions in Table 2.

The 1D–2D data fusion ensemble results are presented in Table 5. This generates 12 ensemble pair combinations that yield better results than a 1D single-source ensemble. However, compared with the 2D single-source ensemble, the accuracies deteriorated. Overall, the soft-voting method yielded better results than the hard-voting method.

4.2. Proposed data-fused, concatenated ensemble

Table 6 shows the model results taking advantage of the entire notion of this study. Here, the data-fused and concatenated ensemble results are described. The concatenated dataset comprises the prediction outcomes of both 1D and 2D prediction pairs. The average accuracy is shown for each ML model. As expected, this technique yields the best results. Soft*

Table 4
2D model EL results.

Accuracy	2D model	2D* model	2D acc	2D* acc	Voting Conditions						
					1	2	3	4	5	6	7
≈ 95 %	DN121	IRV2	0.995	0.989	0.989	0.995	0.995	0.997	<u>1</u>	0.999	0.998
	DN121	DN169	0.995	0.994	0.994	0.995	<u>0.999</u>	0.999	0.999	0.998	0.998
	DN169	IRV2	0.994	0.989	0.984	<u>0.999</u>	0.995	0.997	0.997	0.998	0.998
	DN121	MNV2	0.995	0.941	0.948	0.988	0.95	0.954	<u>0.997</u>	0.997	0.997
	DN169	MNV2	0.994	0.941	0.945	0.99	0.951	0.955	0.969	0.991	<u>0.993</u>
	MNV2	IRV2	0.941	0.989	0.939	0.991	0.952	0.957	0.995	<u>0.997</u>	0.995
Average			0.986	0.974	0.967	0.993	0.974	0.976	0.993	0.997	0.997
≈ 90 %	DN121	IRV2	0.986	0.967	<u>0.99</u>	0.963	0.97	0.97	0.982	0.986	0.986
	DN121	DN169	0.986	0.916	<u>0.986</u>	0.916	0.941	0.946	0.955	0.964	0.97
	DN169	IRV2	0.916	0.967	<u>0.981</u>	0.902	0.96	0.955	0.952	0.945	0.94
	DN121	MNV2	0.986	0.919	0.919	0.986	0.933	0.941	0.994	<u>0.995</u>	0.994
	DN169	MNV2	0.916	0.919	0.919	0.916	0.933	0.941	<u>0.982</u>	0.949	0.949
	MNV2	IRV2	0.919	0.967	0.919	0.967	0.933	0.941	<u>0.996</u>	0.978	0.978
Average			0.951	0.942	0.952	0.941	0.945	0.949	0.977	0.97	0.97
≈ 80 %	DN121	IRV2	0.77	0.97	0.77	0.97	0.98	0.98	<u>0.99</u>	0.94	0.9
	DN121	DN169	0.77	0.92	0.77	0.92	0.94	0.95	<u>1</u>	0.94	0.9
	DN169	IRV2	0.92	0.97	<u>0.98</u>	0.9	0.96	0.95	0.95	0.95	0.94
	DN121	MNV2	0.77	0.89	0.73	<u>0.93</u>	0.91	0.91	0.92	0.89	0.86
	DN169	MNV2	0.92	0.89	0.89	0.91	0.92	0.93	<u>0.96</u>	0.95	0.94
	MNV2	IRV2	0.89	0.97	0.91	0.95	0.92	0.93	0.96	0.97	<u>0.97</u>
Average			0.84	0.93	0.84	0.93	0.94	0.94	0.96	0.94	0.92
≈ 50 %	DN121	IRV2	0.445	0.555	0.445	0.555	0.555	0.555	<u>0.727</u>	0.445	0.445
	DN121	DN169	0.445	0.487	0.445	0.487	0.485	0.479	<u>0.639</u>	0.445	0.445
	DN169	IRV2	0.487	0.555	0.487	0.555	0.555	<u>0.555</u>	0.534	0.502	0.496
	DN121	MNV2	0.445	0.592	0.445	0.592	0.601	0.599	<u>0.662</u>	0.445	0.445
	DN169	MNV2	0.487	0.592	0.531	0.549	<u>0.583</u>	0.579	0.574	0.505	0.5
	MNV2	IRV2	0.592	0.555	<u>0.592</u>	0.555	0.583	0.581	0.558	0.555	0.555
Average			0.484	0.556	0.491	0.549	0.56	0.558	0.616	0.483	0.481

Table 5
Data fusion EL results.

Accuracy	1D model	2D model	1D acc	2D acc	Case						
					1	2	3	4	5	6	7
≈ 95 %	DN121	DN121	0.961	0.995	0.985	0.931	<u>0.997</u>	0.998	0.993	0.974	0.967
		DN169		0.994	0.980	0.934	<u>0.999</u>	0.999	0.998	0.974	0.967
		IRV2		0.989	0.974	0.935	0.995	0.997	0.993	0.974	0.967
		MNV2		0.941	0.933	0.933	0.949	0.953	0.954	<u>0.961</u>	0.959
	DN169	DN121	0.976	0.995	0.980	0.953	<u>0.998</u>	0.997	0.992	0.974	0.973
		DN169		0.994	0.975	0.957	<u>0.999</u>	0.999	0.994	0.974	0.973
		IRV2		0.989	0.970	0.957	<u>0.994</u>	0.994	0.989	0.973	0.973
		MNV2		0.941	0.929	0.955	0.950	0.954	<u>0.991</u>	0.973	0.972
	IRV2	DN121	0.921	0.995	0.988	0.910	<u>0.998</u>	0.997	0.984	0.942	0.938
		DN169		0.994	0.984	0.914	<u>0.999</u>	0.999	0.992	0.944	0.939
		IRV2		0.989	0.978	0.914	<u>0.994</u>	0.994	0.983	0.944	0.939
		MNV2		0.941	0.937	0.912	0.950	0.953	<u>0.975</u>	0.942	0.938
Average		0.953	0.980	0.953	0.980	0.968	0.934	0.985	0.986	0.987	
			0.986	<u>0.984</u>	0.885	0.982	0.978	0.951	0.924	0.918	
			0.916	<u>0.960</u>	0.842	0.932	0.932	0.931	0.918	0.915	
≈ 90 %	DN121	DN169	0.905	0.967	0.966	0.884	<u>0.967</u>	0.964	0.932	0.918	0.915
		IRV2		0.919	0.908	0.894	0.933	0.941	<u>0.983</u>	0.929	0.921
		DN121		0.986	0.936	0.918	<u>0.989</u>	0.988	0.986	0.960	0.945
		DN169		0.916	0.920	0.868	0.937	0.941	<u>0.958</u>	0.941	0.935
	DN169	IRV2	0.885	0.967	0.924	0.911	0.969	<u>0.969</u>	0.968	0.949	0.939
		MNV2		0.919	0.859	0.929	0.931	0.939	<u>0.970</u>	0.961	0.948
		DN121		0.986	<u>0.985</u>	0.854	0.981	0.977	0.946	0.896	0.890
		DN169		0.916	<u>0.951</u>	0.822	0.928	0.926	0.905	0.886	0.883
	IRV2	IRV2	0.872	0.967	<u>0.967</u>	0.854	0.967	0.963	0.934	0.897	0.890
		MNV2		0.919	0.910	0.863	0.933	0.941	<u>0.965</u>	0.901	0.893
				0.939	0.947	0.939	0.877	0.954	0.955	0.952	0.923
				0.771	0.699	<u>0.903</u>	0.825	0.839	0.866	0.865	0.872
≈ 80 %	DN121	DN169	0.860	0.916	0.876	0.874	0.937	0.941	<u>0.958</u>	0.928	0.919
		IRV2		0.967	0.876	0.921	0.973	0.974	<u>0.976</u>	0.933	0.922
		MNV2		0.888	0.830	0.888	0.901	0.907	<u>0.915</u>	0.900	0.898
		DN121		0.771	0.766	0.866	0.901	0.942	<u>0.973</u>	0.903	0.896
	DN169	DN169	0.882	0.916	<u>0.956</u>	0.824	0.927	0.924	0.898	0.886	0.884
		IRV2		0.967	<u>0.971</u>	0.857	0.966	0.962	0.914	0.901	0.895
		MNV2		0.888	0.890	0.860	0.919	0.928	<u>0.936</u>	0.901	0.895
		DN121		0.771	0.526	<u>0.919</u>	0.842	0.853	0.734	0.704	0.701
	IRV2	DN169	0.706	0.916	0.698	0.894	0.943	<u>0.948</u>	0.927	0.721	0.711
		IRV2		0.967	0.698	0.942	0.974	<u>0.975</u>	0.861	0.721	0.711
		MNV2		0.888	0.675	0.886	<u>0.895</u>	0.894	0.787	0.714	0.707
				0.816	0.82	0.816	0.886	0.788	0.886	0.917	0.924
≈ 50 %	DN121	DN121	0.820	0.445	0.445	0.555	0.445	0.445	<u>0.901</u>	0.571	0.571
		DN169		0.487	0.503	0.555	0.501	0.509	0.559	<u>0.571</u>	0.571
		IRV2		0.555	0.571	0.555	0.555	0.555	0.571	<u>0.571</u>	0.571
		MNV2		0.592	<u>0.588</u>	0.555	0.584	0.582	0.567	0.571	0.571
	DN169	DN121	0.563	0.445	0.445	0.573	0.445	0.445	<u>0.691</u>	0.603	0.598
		DN169		0.487	0.520	0.555	0.502	0.510	0.576	0.586	<u>0.587</u>
		IRV2		0.555	<u>0.590</u>	0.555	0.555	0.555	0.571	0.585	0.586
		MNV2		0.592	0.591	0.570	0.586	0.584	<u>0.599</u>	0.590	0.590
	IRV2	DN121	0.644	0.445	0.445	0.634	0.445	0.445	0.766	0.661	0.658
		DN169		0.487	0.578	0.559	0.505	0.517	0.610	0.636	<u>0.640</u>
		IRV2		0.555	<u>0.650</u>	0.555	0.555	0.555	0.592	0.637	0.640
		MNV2		0.592	0.620	0.601	0.595	0.596	0.626	0.652	<u>0.652</u>
Average			0.596	0.520	0.596	0.520	0.545	0.569	0.523	0.525	0.636

and hard * voting gave a prediction accuracy ranging from 0.72 to 1.00, with a standard deviation as low as 0.002 for some cases.

5. Discussion

To compare the accuracy improvement, the increase in accuracy after the ensemble was calculated and compared with the average accuracy of the two existing models. The increase in accuracy is defined by Eq. 6.

$$\text{Increase of accuracy} = \frac{\text{Ensemble acc} - \text{average}(\text{model acc}, \text{model}^* \text{acc})}{\text{average}(\text{model acc}, \text{model}^* \text{acc})} \quad (6)$$

Fig. 10 shows the accuracy of the 1D CNN, 2D CNN, 1D model ensemble, 2D model ensemble, data fusion ensemble, and data-fused, concatenated ensemble. It is observed that significant accuracy

improvement was observed when the base CNN model performed at the range of ≈50 %. In that case, both the single source ensemble provided better accuracy by a 5–10 % margin compared to the 1D and 2D base CNN models. The double source model demonstrated further improvement in accuracy, whereas the proposed concatenated ensemble showed the highest accuracy, >80 %. However, it is also observed that with the increase in accuracy ranges of the base model (i.e., ≈80 %, ≈90 %, and ≈95 %), the increase in accuracy compared to the double source ensemble and the proposed concatenated ensemble is considerably reduced. This is because the model becomes overfitted to the data set with increased CNN base classifier accuracies. As such, the scope of accuracy improvement is lessened.

Moreover, to check the proposed data fused concatenated ensemble, averages of the probability of belonging to a class (i.e., normal and abnormal) from the CNN models in the ranges ≈50 % and ≈95 % cases are shown in Table 9 where the test data consists of 1280 normal and

Table 6
1D–2D data-fused, concatenated EL results.

Accuracy $\approx$ 95 %													
	Case												Avg.
	1*	2*	3*	4*	5*	6*	7*	8*	9*	10*	11*	12*	
LR	0.997	1.000	0.998	0.967	1.000	1.000	0.993	0.977	0.998	0.998	0.995	0.962	0.990
KNN	0.998	1.000	1.000	0.969	1.000	1.000	0.998	0.995	0.998	0.998	0.998	0.981	0.995
DT	1.000	1.000	0.998	0.969	1.000	0.998	1.000	1.000	1.000	0.998	0.998	0.981	0.995
GNB	1.000	0.998	0.997	0.960	1.000	1.000	0.995	0.974	1.000	0.998	0.995	0.951	0.989
SVC	0.995	1.000	0.998	0.958	0.997	1.000	0.993	0.974	0.997	0.998	0.995	0.960	0.989
RF	1.000	1.000	1.000	0.972	1.000	1.000	1.000	1.000	1.000	0.998	0.998	0.984	0.996
VH	1.000	1.000	0.998	0.972	1.000	1.000	0.997	0.995	1.000	0.998	0.997	0.971	0.994
VS	0.998	1.000	0.998	0.967	1.000	1.000	0.997	0.993	1.000	0.998	0.997	0.984	0.994
VH*	1.000	1.000	1.000	0.972	1.000	1.000	0.998	1.000	1.000	0.998	0.998	0.986	0.996
VS*	0.998	1.000	1.000	0.971	1.000	1.000	1.000	1.000	1.000	0.998	0.998	0.986	0.996
Accuracy $\approx$ 90 %													
	Case												Avg.
	1*	2*	3*	4*	5*	6*	7*	8*	9*	10*	11*	12*	
LR	0.997	0.981	0.984	0.946	0.990	0.960	0.967	0.984	0.993	0.946	0.974	0.938	0.972
KNN	1.000	0.962	0.993	0.998	0.995	0.962	0.976	0.990	0.995	0.991	0.979	0.981	0.985
DT	0.998	1.000	0.995	1.000	0.997	0.997	0.990	0.991	0.998	1.000	0.979	0.988	0.994
GNB	0.995	0.986	0.974	0.964	0.993	0.995	0.974	0.979	0.993	0.991	0.972	0.972	0.982
SVC	1.000	0.977	0.986	0.934	0.993	0.964	0.967	0.979	0.991	0.950	0.979	0.936	0.971
RF	0.998	1.000	0.995	0.998	0.998	0.997	0.991	0.990	0.998	1.000	0.983	0.990	0.995
VH	0.998	0.993	0.991	0.964	0.997	0.993	0.979	0.993	0.997	0.993	0.983	0.972	0.988
VS	1.000	0.993	0.986	1.000	0.993	0.993	0.979	0.991	0.998	0.991	0.984	0.988	0.991
VH*	0.998	0.997	0.995	0.998	0.998	0.997	0.990	0.991	0.997	0.998	0.981	0.986	0.994
VS*	1.000	0.998	0.993	1.000	0.995	0.997	0.990	0.991	0.998	0.997	0.984	0.988	0.994
Accuracy $\approx$ 80 %													
	Case												Avg.
	1*	2*	3*	4*	5*	6*	7*	8*	9*	10*	11*	12*	
LR	0.971	0.965	0.969	0.918	0.986	0.976	0.977	0.917	0.962	0.943	0.976	0.918	0.957
KNN	0.984	0.969	0.976	0.957	0.993	0.995	0.977	0.953	0.993	0.995	0.986	0.915	0.974
DT	0.998	0.998	0.983	0.944	0.997	1.000	0.988	0.948	1.000	1.000	0.993	0.976	0.985
GNB	0.997	0.995	0.971	0.912	0.993	0.984	0.971	0.936	0.993	0.997	0.977	0.912	0.970
SVC	0.977	0.950	0.971	0.913	0.967	0.977	0.977	0.899	0.981	0.946	0.976	0.918	0.954
RF	0.998	0.998	0.986	0.969	0.995	0.998	0.990	0.974	1.000	1.000	0.995	0.971	0.990
VH	0.995	0.997	0.979	0.941	0.993	0.997	0.981	0.938	0.998	1.000	0.991	0.922	0.978
VS	0.998	0.997	0.979	0.944	0.993	0.997	0.981	0.953	0.998	1.000	0.991	0.948	0.982
VH*	0.995	0.998	0.984	0.953	0.993	0.998	0.990	0.962	1.000	1.000	0.995	0.965	0.986
VS*	0.997	0.998	0.983	0.957	0.995	0.998	0.988	0.960	1.000	1.000	0.995	0.974	0.987
Accuracy $\approx$ 50 %													
	Case												Avg.
	1*	2*	3*	4*	5*	6*	7*	8*	9*	10*	11*	12*	
LR	0.569	0.554	0.589	0.550	0.606	0.594	0.620	0.606	0.693	0.649	0.648	0.637	0.610
KNN	0.920	0.589	0.828	0.670	0.938	0.660	0.792	0.729	0.932	0.707	0.781	0.745	0.774
DT	0.931	0.736	0.745	0.691	0.941	0.743	0.799	0.778	0.913	0.741	0.809	0.797	0.802
GNB	0.917	0.554	0.589	0.580	0.731	0.668	0.693	0.665	0.917	0.646	0.648	0.635	0.687
SVC	0.569	0.554	0.589	0.563	0.599	0.587	0.608	0.589	0.689	0.635	0.639	0.635	0.605
RF	0.938	0.752	0.722	0.693	0.944	0.778	0.839	0.788	0.938	0.797	0.839	0.814	0.820
VH	0.941	0.679	0.752	0.689	0.922	0.731	0.818	0.754	0.931	0.747	0.802	0.748	0.793
VS	0.941	0.729	0.821	0.688	0.912	0.769	0.802	0.797	0.941	0.780	0.816	0.783	0.815
VH*	0.941	0.731	0.783	0.705	0.939	0.748	0.825	0.781	0.936	0.769	0.830	0.806	0.816
VS*	0.924	0.726	0.821	0.729	0.936	0.747	0.800	0.785	0.929	0.754	0.816	0.809	0.815

1597 abnormal images, leading to a ratio of 0.44 and 0.56. It is hypothesized that the classification was successful only when the probability of belonging to each class was 0.44 and 0.56. It was observed that the classification in the range of $\approx$ 50 % case was biased because the model’s accuracy was low. For example, all models were biased in classifying images as abnormal in voltage image data. In contrast, the three models (DenseNet121, DenseNet169, and InceptionResNetV2)

were biased toward classifying images as normal or abnormal, and MobileNet was unbiased in 2D. In contrast, the ratio is similar to the composition of 0.44 and 0.56 in the range of $\approx$ 95 % case because it is a high-accuracy model. It can be concluded that classification with high accuracy can be performed using the biased 1D and 2D models through the proposed data-fused, concatenated ensemble.

In the case of the $\approx$ 50 % accuracy range, classification is biased

toward the abnormal class for all three models of the 1D classification. In addition, in the case of 2D, such an inclination is seen toward the Abnormal class, except for DenseNet121. The proposed data fused, concatenated ensemble, shown in Table 6, on the other hand, for $\approx 50\%$ accuracy range using DenseNet121 for 2D, the final accuracies were higher in Cases 1, 5, and 9 than that of the other cases. Suppose that both the models used for the proposed EL classify different classes in a biased manner. In this case, accuracy can be increased by learning according to a specific pattern and classifying the final result into the correct class. It is known that suitable models that can classify “Normal” and “Abnormal” are different depending on the characteristics of models and data. Even if each model does not perform well, the pattern can achieve the final classification. In addition, both models are inherently biased toward abnormal classes. However, their data were intrinsically normal. In such a case, the accuracy is also improved by recognizing the pattern when both models are incorrect. Ultimately, in this study, even if classification using specific data or models is incorrect, the classification accuracy can be increased by learning through pattern recognition and

feedback from the results as new inputs.

For a better understanding of the empirical values, the analysis of the average accuracies is presented in Table 7, and the percentage improvement in the accuracy is shown in Table 8. The average accuracy of the data-fused, concatenated ensemble was found to be the highest, which is approximately 98.8 %. The average percentage increase in accuracy was also the highest at approximately 16.3 % compared with the average original CNN accuracy.

Three specific reasons can be attributed to the high-performance accuracy of the proposed concatenated ensemble model in particular and the ensemble models in general. First is overfitting avoidance, which means that the learning algorithm finds several hypotheses for small data that seamlessly predict the training data. However, when unseen data are provided, the model fails to classify it correctly. In our case, for the low accuracy range of the base CNN models, overfitting is avoided because by taking the voting approach to different model predictions, reducing the probability of choosing the wrong model, and increasing overall performance. Nevertheless, with the increase in the

Table 7

Average accuracies of models tested with CNN weights.

Acc.	Cases	1D Model	2D Model	1D Ensemble	2D Ensemble	Data fusion ensemble	Data fused concatenated ensemble
$\approx 95\%$	1*	0.961	0.995	–	–	0.997	1.000
	2*		0.994	0.986	0.999	0.999	1.000
	3*		0.989	0.955	1.000	0.997	1.000
	4*	0.976	0.941	–	0.997	0.961	0.972
	5*		0.995	0.986	0.999	0.998	1.000
	6*		0.994	–	–	0.999	1.000
	7*	0.921	0.989	0.975	0.999	0.994	1.000
	8*		0.941	–	0.993	0.991	1.000
	9*		0.995	0.955	1.000	0.998	1.000
	10*	0.953	0.994	0.975	0.999	0.999	0.998
	11*		0.989	–	–	0.994	0.998
	12*		0.941	–	0.997	0.975	0.986
Avg.			0.980	0.972	0.998	0.992	0.996
$\approx 90\%$	1*	0.905	0.986	–	–	0.984	1.000
	2*		0.916	0.913	0.986	0.960	0.998
	3*		0.967	0.906	0.990	0.967	0.995
	4*	0.885	0.919	–	0.995	0.983	1.000
	5*		0.986	0.913	0.986	0.989	0.998
	6*		0.916	–	–	0.958	0.997
	7*	0.872	0.967	0.903	0.981	0.969	0.990
	8*		0.919	–	0.982	0.970	0.993
	9*		0.986	0.906	0.990	0.985	0.998
	10*	0.887	0.916	0.903	0.981	0.951	0.998
	11*		0.967	–	–	0.967	0.984
	12*		0.919	–	0.996	0.965	0.988
Avg.			0.947	0.907	0.987	0.971	0.995
$\approx 80\%$	1*	0.860	0.771	–	–	0.903	0.998
	2*		0.916	0.920	0.997	0.958	0.998
	3*		0.967	0.903	0.986	0.976	0.984
	4*	0.882	0.888	–	0.931	0.915	0.957
	5*		0.771	0.920	0.997	0.973	0.995
	6*		0.916	–	–	0.956	0.998
	7*	0.706	0.967	0.922	0.981	0.971	0.990
	8*		0.888	–	0.958	0.936	0.962
	9*		0.771	0.903	0.986	0.919	1.000
	10*	0.816	0.916	0.922	0.981	0.948	1.000
	11*		0.967	–	–	0.975	0.995
	12*		0.888	–	0.966	0.895	0.974
Avg.			0.886	0.915	0.976	0.944	0.988
$\approx 50\%$	1*	0.563	0.445	–	–	0.901	0.941
	2*		0.487	0.580	0.639	0.571	0.731
	3*		0.555	0.644	0.727	0.571	0.941
	4*	0.580	0.592	–	0.662	0.588	0.731
	5*		0.445	0.580	0.639	0.691	0.939
	6*		0.487	–	–	0.587	0.769
	7*	0.644	0.555	0.644	0.555	0.590	0.825
	8*		0.592	–	0.579	0.599	0.797
	9*		0.445	0.644	0.727	0.766	0.941
	10*	0.596	0.487	0.644	0.555	0.640	0.780
	11*		0.555	–	–	0.650	0.830
	12*		0.592	–	0.583	0.652	0.809
Avg.			0.520	0.623	0.630	0.651	0.836

Table 8
Increase in percentage accuracies for various CNN pre-trained models.

Acc.	Data	Original	1D Ensemble	2D Ensemble	Data fusion ensemble	Data-fused concatenated Ensemble
≈ 95 %	1D	0.953	2 %	–	4 %	4 %
	2D	0.98	–	2 %	1 %	2 %
	Avg.	0.966	2 %	2 %	3 %	3 %
≈ 90 %	1D	0.887	2 %	–	9 %	11 %
	2D	0.995	–	4 %	2 %	5 %
	Avg.	0.917	2 %	4 %	6 %	8 %
≈ 80 %	1D	0.816	12 %	–	16 %	21 %
	2D	0.886	–	9 %	7 %	11 %
	Avg.	0.851	12 %	9 %	11 %	16 %
≈ 50 %	1D	0.596	5 %	–	9 %	40 %
	2D	0.52	–	20 %	25 %	61 %
	Avg.	0.558	5 %	20 %	17 %	50 %

Table 9
Probability distribution of classes of 95 % accuracy model (left) and 50 % accuracy model (right).

Model	50 % accuracy model				95 % accuracy model			
	1D		2D		1D		2D	
	Normal	Abnormal	Normal	Abnormal	Normal	Abnormal	Normal	Abnormal
DenseNet121	0.03	0.97	0.99	0.01	0.44	0.56	0.44	0.56
DenseNet169	0.05	0.95	0.15	0.85	0.44	0.56	0.45	0.55
InceptionResNetV2	0.11	0.89	0.01	0.99	0.39	0.61	0.47	0.53
MobileNet	–	–	0.42	0.58	–	–	0.48	0.52

Table 10
Training times for different accuracy models.

Range of model accuracy	Ensemble		Concatenated ensemble	
	Acc	Time (s)	Acc	Time (s)
≈ 50 %	0.6505	0.52	0.836	63.66
≈ 80 %	0.9438	0.69	0.988	28.48
≈ 90 %	0.9706	0.76	0.995	23.89
≈ 95 %	0.9922	0.85	0.996	16.14

Table 11
Result of classification for total images and specific images.

		Number of images			
		Low accuracy		High accuracy	
1D	2D	Total	Test	Total	Test
Incorrect	Correct	2722 (10.6 %)	353 (12.3 %)	1915 (7.4 %)	230 (8.0 %)
Correct	Incorrect	4768 (18.4 %)	955 (33.2 %)	352 (1.4 %)	13 (0.4 %)
Incorrect	Incorrect	<u>1202 (4.6 %)</u>	<u>135 (4.7 %)</u>	<u>152 (0.6 %)</u>	<u>1 (0.1 %)</u>
Correct	Correct	17,201 (66.4 %)	1434 (49.8 %)	23,474 (90.6 %)	2633 (91.5 %)
Total		25,893 (100 %)	2877 (100 %)	25,893 (100 %)	2877 (100 %)

Table 12
Result of classification before and after ensemble.

		Number of images			
		Low accuracy model		High accuracy model	
		Before	After	Before	After
Incorrect		1443 (50.2 %)	<u>121 (4.2 %)</u>	244(9.5 %)	<u>7 (0.3 %)</u>
Correct		1434 (49.8 %)	2756 (95.8 %)	2633 (91.5 %)	2870 (99.7 %)
Total		2877 (100 %)	2877 (100 %)	2877 (100 %)	2877 (100 %)

accuracy range of the base CNN models, the room for improvement becomes significantly narrow; thus, some overfitting is bound to happen.

Second, base models, when individually used for classification, have the tendency to be trapped in the local minimum accuracy. Whereas, in the case of ensemble models, multiple base models are trained for the prediction purpose. By the voting approach, the global lowest optimum is selected. Third, the representation indicates that a single model's search space cannot often enclose the optimal hypothesis. However, merging several models using an ensemble can better fit the data space [48]. Moreover, the data-fused, concatenated ensemble used both voltage (1D) images and 2D frame cuts data in conjunction with one another for the prediction, which resulted in a more extensive dataset for improving prediction. Second, as seen in [47], the prediction outcomes were concatenated instead of the image features, generating a simple numeric matrix of prediction probabilities. Matrix was used as the input for the double ensemble to reduce the overall training time. Ultimately, the results improved due to the stacking ensemble approach, which used six ML classifiers.

When comparing training time with accuracy, the data-fused, concatenated ensemble algorithm scored the best prediction; the training time was increased by 20 to 60 s. The reason is that the initialization and weight selections took longer when the proposed ensemble operation was done for the lower accuracy range base CNN model. In the higher accuracy range, CNN models continuously performed backpropagation, increasing overall accuracy but increasing the time for weight selection. Nevertheless, the mentioned models maintained a lower training prediction time, which decreased the overall training time. The percentage increase in accuracy for the said models is found to be low. As such, a trade-off exists between the training time and the percentage increase in prediction accuracy. The trade-off cannot be eradicated completely; however, a near optimal approach is suggested. For example, if prediction time is disregarded, then substantial improvement is observed from ≈50 % accuracy to 83.6 % concatenated accuracy. However, considering training time, the ≈80 % accuracy range model can be considered optimal, with a prediction improvement of 16 %, an average training time of 28.48 s, and 98.8 % accuracy. Accordingly, it can be concluded that the high-accuracy range model showed an increase in accuracy from 90 % to 99.5 %, with a training

Table 13

Results of detecting time for the anomaly detections with respect to different models.

Required time (seconds)	1D	2D	1D ensemble	2D ensemble	Data fusion ensemble	Data-fused concatenated ensemble
1D	0.075	–	0.075	–	0.075	Max (1D: 0.075,
2D	–	0.033	–	0.033	0.033	2D: 0.033)
Ensemble	–	–	0.005	0.005	0.005	0.005
Total	0.075	0.033	0.08	0.038	0.113	0.08

time of 23.89 s. In contrast, the low-accuracy model's growth in accuracy was from $\approx 50\%$ to 83.6% , with a training time of 63.66 s. The near-optimal model reduced this trade-off between time and accuracy by increasing average accuracy from $\approx 80\%$ to 98.8% with a reasonable training time of 28.48 s. The relationship between the accuracies and training times is presented in Table 10.

A 'heatmap' was used to determine the correlation between 1D_Normal, 1D_Abnormal, 2D_Normal, and 2D_Abnormal input variables of the concatenated data set. Fig. 11 shows a heatmap showing the correlation in an ensemble using a low-accuracy model on the right and a high-accuracy model on the left. Here, the input variables of 1D_Normal and 2D_Normal of the probabilities are the prediction results of the 1D and 2D models, respectively, being the normal class. Hence, if the accuracy of the two models is high, then the correlation between the input variables should be high. In addition, the input variables of 1D_Abnormal and 2D_Abnormal of probabilities are the prediction results of the 1D and 2D models, respectively, representing the abnormal class. Hence, if the accuracy of the two models is high, then the correlation between the input variables should be high. 1D_Normal and 2D_Normal, which represent the probability of belonging to the same normal class, have a positive correlation, whereas 1D_Abnormal and 2D_Abnormal, which represent the probability of belonging to the abnormal class, have a positive correlation. In contrast, Normal, which indicates the probability of belonging to the normal class, and Abnormal, which shows the probability of belonging to the abnormal class, have a negative correlation.

Table 11 lists the classification results for the input images. The total number of images that the 1D and 2D low-accuracy models misclassified was 1202, accounting for 4.6 % of the total images. The test data prepared with 2877 images also showed a similar percentage of misclassification. Conversely, the percentage of misclassification was noticeably reduced using the high-accuracy models by 0.6 % for total data and 0.1 % for test data. In Table 12, the misclassified cases were reduced by 46 % and 9.2 % after performing ensemble for the low- and high-accuracy models, respectively.

As shown in Table 13, comparisons were made based on the detection time of prediction for each model. In anomaly detection (1D model) using existing voltage image data, the prediction time per image was 0.075 s, which was relatively large because it was performed after converting the voltage value into a graph image. In contrast, anomaly detection using frame cut images (2D model) required a prediction time of 0.033 s per image. Since an additional 0.005 s is generated when the ensemble is performed, it takes 0.08 s and 0.038 s for 1D ensemble and 2D ensemble models, respectively. Also, an additional time of 0.005 s is required for data fusion ensemble and data-fused concatenated ensemble models. As a result, if real-time prediction is performed during the bead deposition process, a prediction time of 0.113 s (1D: 0.075 + 2D: 0.033 + ensemble: 0.005) per image is required for the data fusion ensemble. In the case of data-fused concatenated ensemble, predictions of 1D and 2D models are made simultaneously, requiring 0.08 s (1D: 0.075 + ensemble: 0.005) per image. Since the 1D model prediction time (0.075 s) is longer than the 2D one (0.033 s), the 2D model prediction result is already available. Thus, the frame per second (fps) is 12.5 in the case of the data-fused concatenated ensemble.

6. Conclusion

We proposed a novel data-fused, concatenated EL algorithm to obtain a flexible and robust methodology for in-situ anomaly detection in WAAM. To achieve this, voltage (1D) images and 2D frame cuts datasets and ML models were seamlessly integrated to overcome the limitations and difficulties in acquiring sufficient data and finding a near-optimal ML algorithm. The proposed method was investigated and validated on inexpensive and comprehensive datasets from the WAAM process. The data were pre-trained using CNN models (i.e., DenseNet121, DenseNet169, MobileNetV2, and InceptionResNetV2), achieving accuracies ranging from 50 % to 95 %. Upon implementing the proposed EL algorithm, the accuracy increased by 3 %, 8 %, 16 %, and even 50 % in CNN models with accuracy ranges of 95 %, 90 %, 80 %, and 50 %, respectively. Furthermore, as shown in Table 8, the proposed EL algorithm achieves an accuracy of 98 %, compared with 81.6 % and 88.6 % of 1D and 2D CNN models, respectively. Results also showed that in addition to improving the prediction accuracy, the proposed ensemble model increases the overall prediction time. Considering the trade-off between the accuracy and prediction time, it can be concluded that the proposed method is suitable and effective when the accuracy of each CNN model is approximately 80 %, whose accuracy was finally increased to 98 %.

Although in this work, the process of Data-fused concatenated ensemble prediction of the anomaly was done for the WAAM process, in essence, the concept can also be applied to the prediction of anomalies in other DED powder-fed and wire-fed processes. Moreover, in this work, we used two types of data that can be expanded to a multi-source approach that can incorporate a wide range of data collection methods such as 3D scanner, dual camera setup, and laser profilometry, among others, as the data set can be enhanced by the usage of the 3D data which is expected to improve the prediction accuracy. In our case, as 1D data, only voltage is considered; however, the incorporation of other 1D signals, such as acoustics, temperature, and pyrometry data, can further improve the model performance. Similarly, in the case of 2D data, x-ray and infrared images can also result in better prediction results. Furthermore, the work can also be modified for other applications in the WAAM field, such as bead geometry prediction and surface roughness determination. Using the vast number of process parameters can result in optimization problems, in which case reinforcement learning can be used for the optimization and process planning. The mentioned cases are yet to be explored and are taken into consideration for the future expansion of the work.

Funding

This research was supported by the Ministry of Science, ICT (MSIT), Korea, under the High Potential Individuals Global Training Program (No. 2021-0-01566) supervised by the Institute for Information & Communications Technology Planning & Evaluation (IITP). This work was supported by the National Science Foundation under Grant No. 2015693.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence

the work reported in this paper.

Acknowledgment

The authors appreciate the continuous support provided by the Center for Manufacturing Research (CMR) and the Department of Manufacturing and Engineering Technology at Tennessee Technological University.

References

- [1] Frazier WE. Metal additive manufacturing: a review. *J Mater Eng Perform* 2014;23: 1917–28.
- [2] Williams SW, Martina F, Addison AC, Ding J, Pardal G, Colegrove P. Wire + arc additive manufacturing. *Mater Sci Technol* 2016;32:641–7.
- [3] Cunningham CR, Flynn JM, Shokrani A, Dhokia V, Newman ST. Invited review article: strategies and processes for high quality wire arc additive manufacturing. *Addit Manuf* 2018;22:672–86.
- [4] Rannar LE, Glad A, Gustafson CG. Efficient cooling with tool inserts manufactured by electron beam melting. *Rapid Prototyp J* 2007;13:128–35.
- [5] Unocic RR, DuPont JN. Process efficiency measurements in the laser engineered net shaping process. *Metall Mater Trans B* 2004;35:143–52.
- [6] DuPont JN, Marder AR. Thermal efficiency of arc welding processes. *Welding Journal - Welding Research Supplement* 1995;74:406s.
- [7] Ding D, Pan Z, Cuiuri D, Li H. Wire-feed additive manufacturing of metal components: technologies, developments and future interests. *Int J Adv Manuf Technol* 2015;81:465–81.
- [8] Bayat M, Dong W, Thorborg J, To, A.C, Hattel JH. A review of multi-scale and multiphysics simulations of metal additive manufacturing processes with focus on modeling strategies. *Addit Manuf* 2021;47:102278.
- [9] Khanzadeh M, Chowdhury S, Tschopp MA, Doude HR, Marufuzzaman M, Bian L. In-situ monitoring of melt pool images for porosity prediction in directed energy deposition processes. *IJSE Transactions* 2019;51:437–55.
- [10] Xia C, Pan Z, Polden J, Li H, Xu Y, Chen S, et al. A review on wire arc additive manufacturing: monitoring, control and a framework of automated system. *J Manuf Syst* 2020;57:31–45.
- [11] NIST. Measurement science roadmap for metal-based additive manufacturing. Gaithersburg, U.S.A: Workshop Summary Report; 2012. p. 4–5.
- [12] Qin J, Hu F, Liu Y, Witherell P, Wang CCL, Rosen DW, et al. Research and application of machine learning for additive manufacturing. *Addit Manuf* 2022;52: 102691.
- [13] Alpaydin E. Introduction to machine learning. 3rd ed. Cambridge: The MIT Press; 2014.
- [14] Meng L, McWilliams B, Jarosinski W, Park HY, Jung YG, Lee J, et al. Machine learning in additive manufacturing: a review. *JOM* 2020;72:2363–77.
- [15] Alabi MO, Nixon K, Botef I. A survey on recent applications of machine learning with big data in additive manufacturing industry. *American Journal of Engineering and Applied Sciences* 2018;11:1114–24.
- [16] Omar S, Ngadi A, Jebur HH. Machine learning techniques for anomaly detection: An overview. *Int J Comput Appl* 2013;79(2):33–41.
- [17] Wu T, Ortiz J. RLAD: Time series anomaly detection through reinforcement learning and active learning. *Arxiv* 2021:00543. <http://arxiv.org/abs/2104.00543>.
- [18] Mani M, Lane B, Donmez A, Feng S, Moylan S, Fesperman R. Measurement science needs for real-time control of additive manufacturing powder bed fusion processes. Gaithersburg, U.S.A: NIST Internal Report; 2015. <https://doi.org/10.6028/NIST.IR.8036>.
- [19] Cho HW, Shin SJ, Seo GJ, Kim DB, Lee DH. Real-time anomaly detection using convolutional neural network in wire arc additive manufacturing: molybdenum material. *J Mater Process Technol* 2022;302:117495.
- [20] Kim ES, Seo GJ, Kim DB, Lee DH. Development of a CNN-based real-time monitoring algorithm for additively manufactured molybdenum. *Sensors & Actuators: A Physical* 2023;352:114205.
- [21] Vanerio J, Casas P. Ensemble-learning approaches for network security and anomaly detection. Paper presented at: Proceedings of the 2017 Workshop on big data analytics and machine learning for data communication networks, part of SIGCOMM 2017. Association for Computing Machinery, Inc; 2017. <https://doi.org/10.1145/3098593.3098594>.
- [22] Scipioni Bertoli U, Guss G, Wu S, Matthews MJ, Schoenung JM. In-situ characterization of laser-powder interaction and cooling rates through high-speed imaging of powder bed fusion additive manufacturing. *Mater Des* 2017;135: 385–96.
- [23] Gibson I, Rosen D, Stucker B, Khorasani. Materials for additive manufacturing. Additive manufacturing technologies 2021:379–428.
- [24] Mehta M, Shao C. Federated learning-based semantic segmentation for pixel-wise defect detection in additive manufacturing. *J Manuf Syst* 2022;64:197–210.
- [25] Mahmoudi M, Ezzat AA, Elwany A. Layerwise anomaly detection in laser powder-bed fusion metal additive manufacturing. *J Manuf Sci Eng* 2019;141(3):031002.
- [26] Tapia G, Elwany AH, Sang H. Prediction of porosity in metal-based additive manufacturing using spatial Gaussian process models. *Addit Manuf* 2016;12: 282–90.
- [27] Khanzadeh M, Chowdhury S, Marufuzzaman M, Tschopp MA, Bian L. Porosity prediction: supervised-learning of thermal history for direct laser deposition. *J Manuf Syst* 2018;47:69–82.
- [28] Aminzadeh M, Kurfess TR. Online quality inspection using Bayesian classification in powder-bed additive manufacturing from high-resolution visual camera images. *J Intell Manuf* 2019;30:2505–23.
- [29] Poudel A, Yasin MS, Ye J, Liu J, Vinel A, Shao S, et al. Feature-based volumetric defect classification in metal additive manufacturing. *Nat Commun* 2022;13(1): 6369.
- [30] García-Moreno AI, Alvarado-Orozco JM, Ibarra-Medina J, Martínez-Franco E. Image-based porosity classification in Al-alloys by laser metal deposition using random forests. *Int J Adv Manuf Technol* 2020;110:2827–45.
- [31] Khumaidi A, Yuniarno EM, Purnomo MH. Welding defect classification based on convolution neural network (CNN) and Gaussian kernel. In: Paper presented at: 2017 international seminar on intelligent technology and its applications (ISITIA); Aug 28–29; Surabaya, Indonesia; 2017.
- [32] Mery D, Rizzo V, Zscherpel U, Mondragón G, Lillo I, Zuccar I, et al. GDxray: the database of X-ray images for nondestructive testing. *J Nondestruct Eval* 2015;34: 1–12.
- [33] Hou W, Wei Y, Guo J, Jin Y, Zhu C. Automatic detection of welding defects using deep neural network. *J Phys Conf Ser* 2018;933:012006. <https://doi.org/10.1088/1742-6596/933/1/012006>.
- [34] Zhu H, Ge W, Liu Z. Deep learning-based classification of weld surface defects. *Appl Sci* 2019;9(16):3312.
- [35] Park JK, An WH, Kang DJ. Convolutional neural network based surface inspection system for non-patterned welding defects. *Int J Precis Eng Manuf* 2019;20:363–74.
- [36] Jiang H, Hu Q, Zhi Z, Gao J, Gao Z, Wang R, et al. Convolution neural network model with improved pooling strategy and feature selection for weld defect recognition. *Welding in the World* 2021;65:731–44.
- [37] Liu W, Shan S, Chen H, Wang R, Sun J, Zhou Z. X-ray weld defect detection based on AF-RCNN. *Welding in the World* 2022;66:1165–77.
- [38] Scime L, Beuth J. A multi-scale convolutional neural network for autonomous anomaly detection and classification in a laser powder bed fusion additive manufacturing process. *Addit Manuf* 2018;24:273–86.
- [39] Caggiano A, Zhang J, Alfieri V, Caiazza F, Gao R, Teti R. Machine learning-based image processing for on-line defect recognition in additive manufacturing. *CIRP Ann* 2019;68:451–4.
- [40] Reich R, Hauser T, Lutz B, Pantano M, Kamps T, Knoll A. Distance-based multivariate anomaly detection in wire arc additive manufacturing. In: Paper presented at: 19th IEEE international conference on machine learning and applications. Dec 14–17; Miami, U.S.A; 2020.
- [41] Lyu J, Manoochehri S. Online convolutional neural network-based anomaly detection and quality control for fused filament fabrication process. *Virtual and Physical Prototyping* 2021;16:160–77.
- [42] Lee C, Seo G, Kim D, Kim M, Shin JH. Development of defect detection AI model for wire+arc additive manufacturing using high dynamic range images. *Appl Sci* 2021; 11(16):7541.
- [43] Li Z, Zhang Z, Shi J, Wu D. Prediction of surface roughness in extrusion-based additive manufacturing with machine learning. *Robot Comput Integr Manuf* 2019; 57:488–95.
- [44] Khan IA, Birkhofer H, Kunz D, Lukas D, Ploshikhin V. A random Forest classifier for anomaly detection in laser-powder bed fusion using optical monitoring. *Materials* 2023;16(19):6470.
- [45] Zhang Z, Zhang Y, Wen Y, Ren Y, Liang X, Cheng J, et al. An improved stacking ensemble learning model for predicting the effect of lattice structure defects on yield stress. *Comput Ind* 2023;151:103986.
- [46] Li R, Jin M, Paquit VC. Geometrical defect detection for additive manufacturing with machine learning models. *Mater Des* 2021;206:109726.
- [47] Shin, S. J., Lee, J. H., Sainand, J., & Kim, D. B. (2022). Material-Adaptive Anomaly Detection Using Property-Concatenated Transfer Learning in Wire Arc Additive Manufacturing, Available at SSRN 4242808.
- [48] Sagi O, Rokach L. Ensemble learning: a survey. *WIREs – Data Mining and Knowledge Discovery* 2018;8:e1249. <https://doi.org/10.1002/widm.1249>.