A novel intelligent health indicator using acoustic waves
CEEMDAN-driven semi-supervised ensemble deep learning

Moradi, Morteza; Galanopoulos, Georgios; Kuiters, Thyme; Zarouchas, Dimitrios

DOI
10.1016/j.ymssp.2024.112156

Publication date
2025

Document Version
Final published version

Published in
Mechanical Systems and Signal Processing

Citation (APA)
Moradi, M., Galanopoulos, G., Kuiters, T., & Zarouchas, D. (2025). A novel intelligent health indicator using acoustic waves: CEEMDAN-driven semi-supervised ensemble deep learning. *Mechanical Systems and Signal Processing*, *224*, Article 112156. https://doi.org/10.1016/j.ymssp.2024.112156

Mechanical Systems and Signal Processing 224 (2025) 112156

Contents lists available at ScienceDirect

Mechanical Systems and Signal Processing

journal homepage: www.elsevier.com/locate/ymssp

Full Length Article

A novel intelligent health indicator using acoustic waves: CEEMDAN-driven semi-supervised ensemble deep learning

Morteza Moradi [*], Georgios Galanopoulos, Thyme Kuiters, Dimitrios Zarouchas

Center of Excellence in Artificial Intelligence for Structures, Prognostics & Health Management, Aerospace Engineering Faculty, Delft University of Technology, Kluyverweg 1, Delft 2629 HS, the Netherlands

ARTICLE INFO

Keywords:
Prognostics and health management
Intelligent health indicator
Semi-supervised deep neural network
Ensemble learning
Acoustic emission

ABSTRACT

Designing health indicators (HIs) for aerospace composite structures that demonstrate their health comprehensively, including all types of damage that can be adaptively updated, is challenging, especially under complex conditions like impact and compression-fatigue loadings. This paper introduces a new AI-based approach to designing reliable HIs (fulfilling requirements—monotonicity, prognosability, and trendability—referred to as 'Fitness') for single-stiffener composite panels under fatigue loading using acoustic emission sensors. It incorporates complete ensemble empirical mode decomposition with adaptive noise for feature extraction, semi-supervised base deep learner models made of long short-term memory layers for information fusion, and a semi-supervised paradigm to simulate labels inspired by the physics of progressive damage. In this way, nondifferentiable prognostic criteria are implicitly implemented into the learning process. Ensemble learning, especially using a semi-supervised network built with bidirectional long short-term memory, improves HI quality while reducing deep learning randomness. The Fitness function equation has been modified to provide a more trustworthy foundation for comparison and enhance the practical reliability of the standard in prognostics and health management. Ablation experiments are conducted, including variations in dataset division and leave-one-out cross-validation, confirming the generalizability of the approach.

1. Introduction

Structural health monitoring (SHM) has been introduced in the last few decades as a way to provide in-situ measurements of the state of engineering structures in real time and identify adverse changes in their structural integrity [1,2]. Also, SHM is essential for the

Abbreviations: AE, Acoustic Emission; AI, Artificial Intelligence; ANN, Artificial Neural Network; BiLSTM, Bidirectional LSTM; C-C, Compression-Compression; CBM, Condition-Based Maintenance; CEEMDAN, Complete EEMD with Adaptive Noise; D, Dropout; DL, Deep Learning; EEMD, Ensemble Empirical Mode Decomposition; EL, Ensemble Learning; EMD, Empirical Mode Decomposition; EoL, End-of-Life; FC, Fully Connected; FE, Feature Extraction; FF, Feature Fusion; FFT, Fast Fourier Transform; FS, Feature Selection; HI, Health Indicator; HHT, Hilbert-Huang Transform; IMF, Intrinsic Mode Function; ReLU, Rectified Linear Unit; LOOCV, Leave-One-Out Cross-Validation; LSTM, Long Short-Term Memory; MMK, Modified Mann-Kendall; ML, Machine Learning; Mo, Monotonicity; MSE, Mean-Squared Error; PCA, Principal Component Analysis; PHM, Prognostics and Health Management; Pr, Prognosability; RMSE, Root-Mean-Square Error; RUL, Remaining Useful Life; SAE, Simple Averaging Ensemble; SHM, Structural Health Monitoring; SP, Signal Processing; SSDL, Semi-Supervised Deep Learning; SSL, Semi-Supervised Learning; SSP, Single-Stiffened composite Panel; Tr, Trendability; WAE, Weighted Averaging Ensemble.

* Corresponding author.
E-mail address: M.Moradi-1@tudelft.nl (M. Moradi).

https://doi.org/10.1016/j.ymssp.2024.112156
Received 26 July 2024; Received in revised form 15 October 2024; Accepted 12 November 2024
Available online 2 December 2024
0888-3270/

Nomenclature

m	Mean value
c_n	n^{th} IMF
r_n	n^{th} residual
$E_j[w_i(t)]$	particular noise
w_i	added Gaussian noise
β	noise amplitude
$M(\bullet)$	local means of data
β_n	signal-to-noise ratio
$x(t_p)$	HI at the time of t_p
$x(t_i)$	HI at the time of t_i
M	number of units (specimens)
x_j	HI for the j^{th} unit out of M units
x_k	HI for the k^{th} unit out of M units
N_j	number of measurements for the j^{th} unit
N_k	number of measurements for the k^{th} unit
$cov(x_j, x_k)$	covariance where x_j and x_k are HIs
$sgn(\bullet)$	signum function
σ_{x_j}	standard deviation of x_j
σ_{x_k}	standard deviation of x_k
τ	test unit
τ'	all units (except for the test unit)
$M^{\tau'}$	number of all units
T_i	Simulated HI as target value at time step i
HI_i	Network output for time step i
$\mathscr{H}\{\bullet\}$	Hilbert transform
j	imaginary part
β	shape parameter
t	operating time
N_a	Number of actuators
N_s	Number of sensors
f_k	k^{th} base learner model
ω_k	weight for k^{th} base learner model
$\overline{\omega}_k$	normalized weight for k^{th} base learner model
$L_{regress}$	Loss function of regression model

prognostics and health management (PHM) framework, which aims to direct the industry toward condition-based maintenance (CBM) policies that increase availability while lowering operating downtime and costs [3,4]. The shift into a CBM paradigm under a PHM framework requires reliable SHM techniques.

The final level of SHM is the accurate estimation of the remaining useful life (RUL) through a prognostic model [5,6]. Of imperative importance is the input to these models, which are features capable of correlating to degradation [7,8]. Such features are usually referred to as health indicators (HIs) [9] and are an indirect output of the SHM data [10]. HIs can be directly tied to the damage accumulation displaying a non-linear behavior, as damage progression is usually non-linear. For structural components, commonly employed HIs are based on acoustic emission (AE) [11], strain distribution [12], stiffness [13], crack size [14,15], or crack density [16]. However, the latter three cannot be directly measured in real-time by SHM systems. For the construction of capable HIs, researchers have turned their attention to advanced processing methodologies to create suitable HIs. Principal component analysis (PCA)-based algorithms [17,18,19], genetic programming [20,21], and other machine learning (ML) algorithms [10,22,23,24,25,26] are usually employed for this process. The favorable properties of such HIs make them more appropriate for use in more complex applications, like composite structures.

The quality and suitability of HIs are usually measured by three metrics: monotonicity (Mo), prognosability (Pr), and trendability (Tr) [27], each commenting on a specific property of the HI. All three are crucial for enhanced HI quality and increased prognostic accuracy [24], which can be considered as an objective function called "Fitness". However, extracting HIs fulfilling the Fitness metric is very challenging, even if we assume that informative SHM data is available, due to inhomogeneous material characteristics, variable operational conditions, stochastic activation and interaction of damage mechanisms, and uncertainties. Therefore, desirable information from SHM data should be effectively analyzed and then this information should be mapped to HI. Artificial intelligence (AI) and signal processing (SP) are two major fields that can be helpful in this regard.

For timeseries signals, like AE signals, common approaches for feature extraction (FE) in the SP field resort to frequency analysis using different methods such as fast Fourier transform (FFT) [28,29]. Such techniques are popular for analyzing stationary signals, which is not the case for complex structures under variable loading and environmental conditions. In real-world applications, signals are inherently non-stationary and require more complicated processing methodologies [30,31]. To address this issue, researchers are employing the Hilbert-Huang transform (HHT) [32,33], where the signal is decomposed into intrinsic mode functions (IMFs) using empirical mode decomposition (EMD). EMD has found application in SHM for damage detection purposes. For instance, EMD was used to determine the existence of damage in different types of structures, such as aluminum beams, reinforced concrete slabs, and plates [34]. This technique was also employed to detect faults in a satellite altitude control system [35]. The methodology decomposed the signal into IMFs to capture the fault trend and was proven to be able to diagnose both abrupt and early failures.

Despite the wide appeal of EMD, the methodology suffers from a significant drawback: mode mixing, which can lead to inaccurate SHM results. To solve this issue, variational mode decomposition (VMD) was proposed by Dragomiretskiy and Zosso [36]. VMD can efficiently differentiate distinct frequency band components of the signal and has an improved time-frequency distribution capacity in signal decomposition and assessment than EMD [37]. VMD decomposes a signal into a preset number of modes, each with unique sparsity characteristics, tackling the mode mixing problem more effectively than EMD. However, VMD has some drawbacks, including the requirement to predetermine the number of modes and the computational complexity. Another method to address the mode mixing issue is ensemble EMD (EEMD) proposed by Wu and Huang [38], which improved the methodology by adding white noise to the signal. Sarmadi et al. [39] combined EEMD and Mahalanobis squared distance to identify the presence of damage in a steel truss structure. The methodology was compared to the simple EMD and displayed superior damage localization performance. Amiri and Darvishan [40] also compared the EMD and EEMD capabilities for damage detection using vibrational signals from a steel moment frame. Although superior to the EMD, the EEMD has two significant drawbacks. Due to the added Gaussian white noise, there is residual noise in the IMFs and difficulty in averaging different numbers of IMFs. To overcome these drawbacks, the complete EEMD with adaptive noise (CEEMDAN) was proposed [41]. In CEEMDAN, the noise is added after each stage of the decomposition process, creating a unique residual. Thus, this SP technique is employed as a suitable candidate to process AE data in the present work.

As a powerful mathematical tool, ML and deep learning (DL) have garnered significant attention in recent years across the domains of SHM, CBM, and PHM. Within this context, a plethora of algorithms have emerged aimed at analyzing data for purposes such as feature extraction (FE), feature selection (FS), and feature fusion (FF) [31,42]. While there have been persistent efforts to replace traditional SP algorithms with ML and DL models in pursuit of higher accuracy and end-to-end automation, AI models have not yet reached that level of maturity [43]. In fact, both SP and AI (particularly DL) represent potent mathematical tools for information extraction from data, each with its own unique set of advantages and drawbacks. SP methods often follow explicit solutions, resulting in globally applicable and faster outcomes, whereas AI methods, including back-propagation-based approaches, tend to be implicit, introducing locality and randomness into their outputs. On the other hand, DL models can uncover deeper and more intricate relationships among input variables thanks to deep networks and parallel computation capabilities. In contrast, SP models cannot be conducted simultaneously; they typically require a sequential approach and cannot be applied until the previous step is completed. Both SP and AI methodologies demand ongoing parameter adjustments when operational conditions change, necessitating the fine-tuning of (hyper)parameters. Consequently, expertise in AI or SP is essential for navigating the (hyper)parameter tuning process. Regarding the (hyper)parameter tuning of the models and the relevant knowledge to do so, a data analyzer with skills in AI or SP for each case is needed. In terms of experts' involvement in data extraction and labeling, both AI and SP models need that step. Therefore, SP and AI can complement each other in order to provide a more efficient methodology. For instance, Mousavi et al. [44,45], paired CEEMDAN with HHT and an artificial neural network (ANN) to locate and classify the severity of damage in a model steel bridge structure. While the methodology provided effective indicators for detecting damage, first, dealing with composite structures is more challenging than with metallic ones; second, the task of constructing a comprehensive, adaptive HI poses greater difficulty compared to the damage detection.

To develop a capable HI, the common procedure involves the selection of the best extracted features from SHM data in terms of the prognostic metrics. In such cases, features with low Fitness values are discarded, despite the fact that they may prove useful in a feature fusion scenario. A more valid approach would be to fuse the available information and embed the prognostic criteria into the fusion process as a supervisory tool, rather than using them only as a measurement of the HI's quality at the final stage [23]. However, Fitness is not a differentiable function, making it challenging to incorporate into many optimization techniques, such as back-propagation-based ones. Furthermore, the entire timeseries of HIs, and as a result, SHM data up to the end of life (EoL), should be available in order to calculate Tr and Pr, which is not practical.

To address these challenges, different kernel functions (linear, polynomial, logarithmic, and exponential) in terms of operational time were investigated based on their suitability as functions to simulate hypothetical HIs, and it was found out that the polynomial kernel equation provided the highest prognostic criteria, inspired by damage accumulation [46]. Accordingly, a label simulation technique under the inductive semi-supervised learning (SSL) paradigm can be adopted to train a back-propagation-based model. Other shortcomings that should be addressed are: first, input features into the model could be more informative; and second, the inherent randomness and uncertainty of an ML-based model should be mitigated. In this study, CEEMDAN is used to deal with nonstationary signals and extract more informative features and ensemble learning (EL) is adopted to address the ML's randomness.

In this paper, a SSL framework, made up of semi-supervised (SS) base learners and ensemble learners, is proposed to develop HIs.

The composite structures experienced different real-world uncertainties, including impacts before or during operation and disbond defects during the manufacturing process. Ten different semi-supervised deep learning (SSDL) networks made of multilayers of long short-term memory (LSTM) are considered as the SS-base learner, and sixteen models, including four unsupervised averaged-based and twelve SSDL networks, are considered as the ensemble learner. Compression-compression (C-C) run-to-failure fatigue tests are conducted on single stiffened composite panels experiencing different damage scenarios, and AE is used to monitor the evolution of damage. First, acoustic emission waves are processed, and then CEEMDAN is applied to extract IMFs. Extracted features from IMFs are used as input to the SSL framework to derive HIs. The methodology is validated using a leave-one-out cross-validation (LOOCV) scheme, demonstrating the robustness and repeatability of the process.

The remainder of the paper is structured as follows: In Section 2, the experimental campaign is detailed, providing information on the specimens, fatigue tests, and acoustic emission monitoring. Section 3 presents and discusses the proposed methodologies in depth, covering data acquisition, pre-processing, signal processing (SP), feature extraction (FE), and feature fusion (FF). Section 4 provides the results from the base learner and ensemble models, comparing their performance with existing literature and discussing the findings. Finally, Section 5 concludes the paper.

2. Case study

Two test campaigns were launched at Delft University of Technology, comprising of constant-amplitude C-C fatigue experiments on single-stiffened composite panels (SSPs), representative of aerospace wing components, with a fatigue frequency of 2 Hz and a load ratio of 10 at 65% of the ultimate compressive strength. As can be seen in Fig. 1, an initial damage in the form of disbond (Teflon insert during manufacturing) in the skin//stiffener interface with different sizes or an impact damage of around 10 J located on the stiffener area is introduced to some panels. For the panels that do not experience impact prior to testing, after 5000 cycles, the impact is performed, even if a panel already had an artificial disbond defects. These factors simulate various realistic and uncertain phenomena in the experiments, resulting in a wide range of EoL from 48.7 K to 756.3 K cycles, which will make it more challenging to perform HI construction and RUL prediction. To monitor the damage growth, several SHM techniques were employed, including AE, distributed fiber optical sensors, fiber Bragg gratings, and lamb wave. The experimental campaigns and SHM systems have been discussed in depth in previous publications [47,48,49]. In this paper, we focus on data recorded by AE, which was constantly logged throughout the tests. All data are publicly available in [50,51]. More information on the panels and their fatigue life is presented in Table 1.

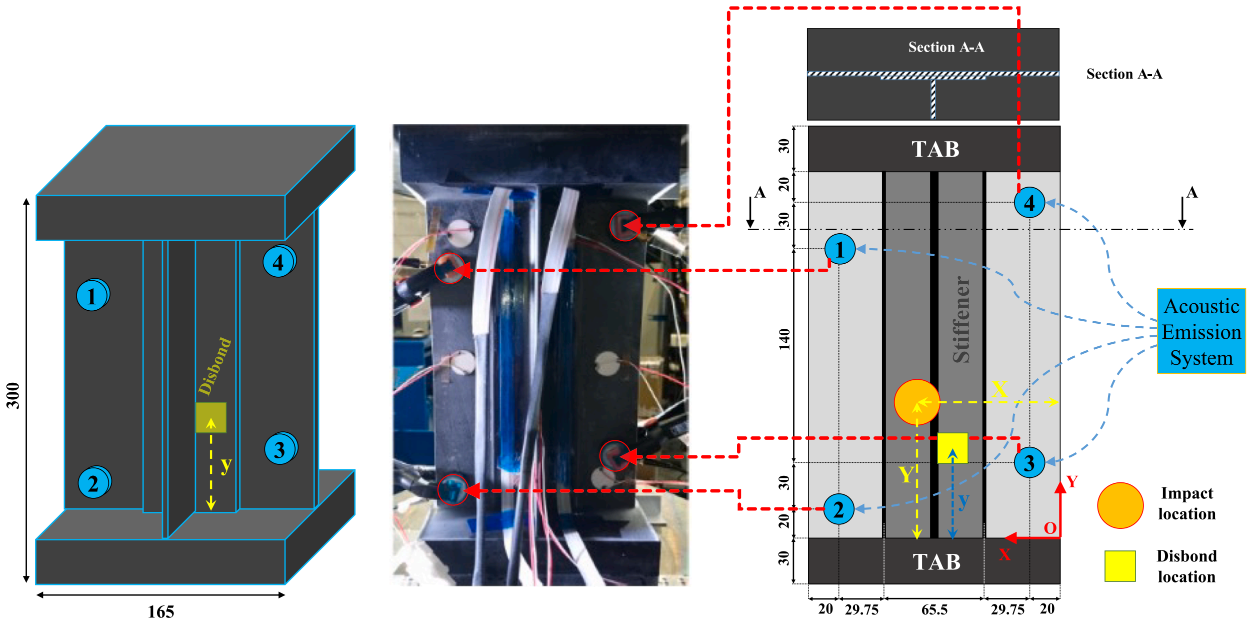

Fig. 1. Experimental setup, including a single stiffener panel (from the stiffener side) under monitoring by four acoustic emission sensors (all dimensions in [mm]).

Table 1
Information about SSPs' lifetimes, impact, disbond, and loading conditions.

Specimen No.	Impact time (cycle)	Disbond		Load [min, max]	Cycles to failure	Labeled cycles	Labeling error (cycle)
		size (mm × mm)	**location – y (mm)**				
	Campaign 2019						
SSP1	0			[−6.5, −65]	152,458	152,457	−1
SSP2	0			[−6.5, −65]	144,969	144,970	1
SSP3	0			[−6.5, −65]	133,281	133,283	1
	Campaign 2020						
SSP4	5000	15 × 20	60	[−6.5, −65]	48,702	48,703	1
SSP5	5000	20 × 20	60	[−6.5, −65]	65,500	65,502	2
SSP6	5000	20 × 25	60	[−6.5, −65]	94,431	94,437	6
SSP7	0			[−6.5, −65]	368,558	368,590	32
SSP8	0			[−6.5, −65]	510,961	510,982	21
SSP9	5000			[−6.5, −65]	226,356	226,361	5
SSP10	5000			[−6.5, −65]	756,226	756,264	38
SSP11	5000			[−6.5, −65]	110,137	110,185	48
SSP12	5000			[−6.5, −65]	170,884	170,898	14

3. Methodologies

In this section, the methodologies and step-by-step process from data collection to HI creation are discussed (see Fig. 2).

3.1. Data acquisition

A 4-channel Vallen AMSY-6 was used to record the acoustic emission waves. Paired with the acquisition system, four (4) AE sensors, also from Vallen Systeme GmbH, with a frequency range of 100–900 kHz (VS900-M)—located at strategically selected locations in order to monitor a broad area of the structure and provide the ability to localize potential damage events (Fig. 1)—and an external 34 dB pre-amplifier were used. In total, six low-level AE features are extracted from the events (directly from the acquisition system), including amplitude (A), rise time (R), energy (E), counts (CNTS), duration (D), and root mean square (RMS).

3.2. Data pre-processing

To simulate hypothetical, ideal HI with inherently high Fitness as the target of SSDL, operational time information, as will be explained in subsection 3.5, is required, and hence accurate cycle labeling is mandatory. The MTS machine cannot export the cycle count straight to the AE system; however, due to the time synchronization of the two systems and knowledge of load and displacement, the cycles each hit occurred are approximated through signal processing with a maximum error of 0.044% (see Table 1).

To limit the amount of data recorded that is not representative of the degradation process (e.g., noise from machine movement), an amplitude threshold of 60 dB is employed in the data acquisition phase. Regarding the further noise reduction of the original acoustic emission signal, the internal Vallen processor for planar localization was utilized. The localization of the AE events is performed using Geiger's method [52]. Pencil lead breaks tests were performed prior to each experiment to measure the wave propagation speed in the longitudinal and shear directions, which showed only slight variability between the different panels. Assuming an average wave velocity of 5265 m/s in both directions [47], Geiger's method (included in the internal Vallen processor) estimates the location based on the time-of-arrival at each sensor. Since we are dealing with an anisotropic structure where localization errors can be large, a filter is applied, excluding events whose location uncertainty is higher than 50 mm.

Windowing is a common practice to treat non-stationary signals. Additionally, because AE is a passive monitoring technique that records continuously, windowing aids in identifying highlighted trends in the data that may reveal possible structural degeneration. There are two main factors influencing the final windowing result: the window's length and interval. In our case, since every 500 fatigue cycles, quasi-static loadings are performed, we selected both length and interval to be 500 cycles.

To tackle potential missing values in our windowed data, either due to the pre-processing methodologies or the calculations of statistical features, linear interpolation is used to fill in those missing values.

3.3. Signal processing

Empirical mode decomposition (EMD) is an algorithm capable of decomposing non-linear, non-stationary signals into a set of orthogonal components. The basis of EMD is the Hilbert-Huang transform [32], which decomposes the original signal into simple intrinsic mode functions called IMFs. Each IMF needs to fulfill two conditions:

1. The number of extrema and zero crossings must be equal or differ by no more than one.
2. The mean value of the upper and lower envelopes is zero everywhere.

The basic steps of the EMD algorithms are explained below.

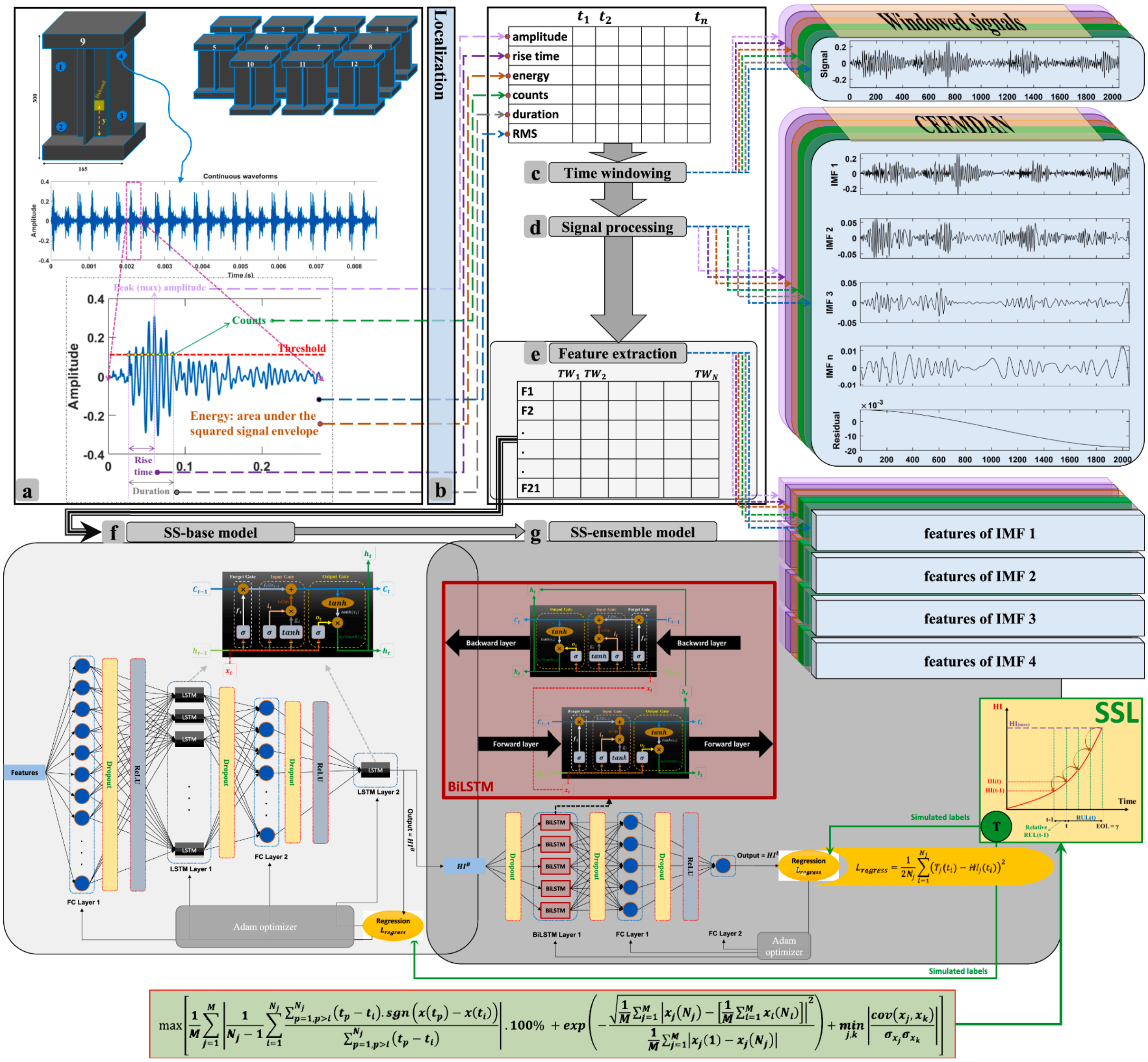

Fig. 2. The overall proposed framework: (a) AE monitoring and low-level feature extraction; (b) localization; (c) windowing; (d) signal processing (CEEMDAN); (e) statistical feature extraction; (f) semi-supervised base learner model; (g) semi-supervised ensemble learner model.

Firstly, we identify the local extrema of the data. By employing cubic splines, an upper and lower signal envelope are created. Then, the mean of the upper and lower envelopes is calculated. In the next step, the difference between the mean (m_1) and the data ($X(t)$) is calculated, which corresponds to the first IMF component h_1:

$$X(t) - m_1 = h_1 \tag{1}$$

In the case that the two conditions mentioned above are not validated, the previous steps are repeated until h_1 complies with those criteria (the sifting process). h_1 now corresponds to the new data, i.e., $X(t) = h_1$, and the previous steps are repeated to extract the next IMF as:

$$h_1 - m_{11} = h_{11} \tag{2}$$

After k siftings, IMF that adheres to the criteria is designated as $c_1 = h_{1k}$. Then, c_1 is the first IMF and is subtracted from the original data:

$$X(t) - c_1 = r_1 \tag{3}$$

where r_1 is the first residual and treated as the new data in Eq. (1).

The subsequent steps are to extract the n^{th} IMF (c_n) until r_n is a monotonic function. When a monotonic function is reached, all IMFs are extracted, and the process is completed. To reconstruct the original data after extracting the n^{th} IMF and leaving the residual r_n, Eq. (4) is used.

$$X(t) = \sum_{i=1}^{n} c_i + r_n \tag{4}$$

Despite the advantages of EMD, it suffers from a major drawback referred to as mode mixing. A solution to this drawback is ensemble EMD (EEMD) [38], which adds Gaussian white noise to the data with the appropriate scale. However, it creates new issues, such as the independence of the decomposition process due to the residual noise in the IMF and deficiencies in the decomposition process. Also, the EEMD makes it difficult to average different numbers of IMFs. These newfound issues are addressed by the complete ensemble empirical mode decomposition with adaptive noise (CEEMDAN) [41]. The CEEMDAN process, unlike the EEMD, adds particular noise $E_j[w_i(t)]$ at each step of the decomposition. The main process of CEEMDAN is described in the following steps:

$$x_i(t) = X(t) + \beta_0 E_1[w_i(t)] \tag{5}$$

where $E_1[w_i(t)]$ is added to the original signal $X(t)$ for $i = 1, 2, \cdots, N$. The parameters w_i, β, and N indicate the added Gaussian noise, the noise amplitude, and the ensemble size, respectively. The first IMF (c_1) is calculated through the first residual r_1 as:

$$c_1 = X(t) - r_1, \quad where \quad r_1 = \frac{1}{N}\sum_{i=1}^{N} M(x_i(t)) \tag{6}$$

where $M(\bullet)$ is the operator representing local means of data. The second IMF (c_2) is obtained through Eq. (7):

$$c_2 = r_1 - r_2, \quad where \quad r_2 = \frac{1}{N}\sum_{i=1}^{N} M(r_1 + \beta_1 E_2[w_i(t)]) \tag{7}$$

Here $E_2[w_i(t)]$ is the second IMF of EEMD. The n^{th} IMF of CEEMDAN is obtained through:

$$c_n = r_{n-1} - r_n, \quad where \quad r_n = \frac{1}{N}\sum_{i=1}^{N} M(r_{n-1} + \beta_{n-1} E_n[w_i(t)]) \tag{8}$$

where $\beta_n = \varepsilon_0 std(r_n)$ is the signal-to-noise ratio (SNR).

3.4. Feature extraction (FE)

As mentioned in Section 3.1. Data acquisition, six low-level features are obtained from each AE event, i.e., A, R, E, $CNTS$, D, *and RMS*. For the i^{th} SSP, there are k_i time windows. The CEEMDAN methodology is applied to each time window in order to extract the IMFs. Arbitrarily, four IMFs are extracted for each time window. Since the time windows have different numbers of data points and occasionally not enough data is available to decompose into IMFs, linear interpolation between the previous and next time windows is performed.

For each low-level feature, four IMFs are extracted, providing 24 new features. For each time window, statistical quantities are calculated to generate a single value, and the single value is assigned a time value of the mean time of the window. 21 statistical quantities are calculated, which are listed in Table 2. In total, 504 ($6x4x21$) new features are obtained, which are going to be the input to the DL algorithm that creates the HI.

3.5. Feature fusion (FF)

To achieve the optimal fusion of features, yielding a HI qualified as per prognostic metrics, a DL model referred to as the base learner is designed. Given the absence of labels for real, comprehensive HIs in composite laminates, a semi-supervised paradigm is adopted. This involves generating hypothetical ideal labels based on prognostic metrics. Subsequently, EL is employed to mitigate the inherent randomness within the base learner. This subsection begins by defining the prognostic metrics that will be updated in the

Table 2
Statistical features.

No.	Name	No.	Name	No.	Name
1	Mean	8	Kurtosis	15	Central moment for 4th order
2	Standard Deviation	9	Crest Factor	16	Central moment for 5th order
3	Root Amplitude	10	Clearance factor	17	Central moment for 6th order
4	Root Mean Square	11	Shape factor	18	FM4
5	Root sum of Squares	12	Impulse factor	19	Median value
6	Peak	13	Maximum to minimum difference	20	Signal Power
7	Skewness	14	Central moment for 3rd order	21	Entropy

current work for consideration of only the test unit (SSP). Following that, both the base and ensemble learner models are introduced.

3.5.1. Health indicator metrics

The evaluation of a prognostic predictor, or HI, is based upon three established criteria, namely Mo, Pr, and Tr [46,53,54]. These metrics are defined as follows:

$$Mo = \frac{1}{M}\sum_{j=1}^{M}\left|\frac{1}{N_j - 1}\sum_{i=1}^{N_j}\frac{\sum_{p=1,p>i}^{N_j}(t_p - t_i).sgn(x(t_p) - x(t_i))}{\sum_{p=1,p>i}^{N_j}(t_p - t_i)}\right|.100\% \quad (9)$$

$$Pr = exp\left(-\frac{\sqrt{\frac{1}{M}\sum_{j=1}^{M}\left|x_j(N_j) - \left[\frac{1}{M}\sum_{i=1}^{M}x_i(N_i)\right]\right|^2}}{\frac{1}{M}\sum_{j=1}^{M}|x_j(1) - x_j(N_j)|}\right) \quad (10)$$

$$Tr = \min_{j,k}\left|\frac{cov(x_j, x_k)}{\sigma_{x_j}\sigma_{x_k}}\right|, \quad j,k = 1,2,\cdots,M \quad (11)$$

where $x(t_p)$ and $x(t_i)$ denote the measurements (HIs in the context) at the times of t_p and t_i, respectively. The $sgn(\bullet)$ function represents the signum function. $cov(x_j, x_k)$ signifies the covariance, where x_j and x_k are vectors of measurements for the j^{th} and k^{th} unit (out of M units) with N_j and N_k measurements, respectively. The standard deviations of x_j and x_k are denoted by σ_{x_j} and σ_{x_k}, respectively. The evaluation metric selected for Mo in Eq. (9) is the modified Mann-Kendall (MMK) metric. In comparison to the Sign and Mann-Kendall versions, MMK is more resilient to noise and considers the correlation between data points with time gaps exceeding one unit [46,53]. All three HIs metrics (Mo, Pr, and Tr) are rated on a scale ranging from 0 to 1, where a score of 1 signifies optimal HI performance.

With consideration to these criteria, the Fitness metric is formulated as follows:

$$Fitness = a.Mo_{HI} + b.Pr_{HI} + c.Tr_{HI} \quad (12)$$

Assuming the control constants a, b, and c each equal 1, the Fitness metric spans from 0 (indicating minimal quality) to 3 (indicating maximal quality) for the evaluated HIs.

It is important to emphasize that the HIs' evaluation metrics mentioned above are devised to encompass all units (SSPs in the present work), specifically from their healthy state to their final failure status within the context of PHM. Without access to complete trajectories of HIs across all units, the assessment of HIs' quality lacks implication. Consequently, whether during the training or testing phase of ML-based models, the inclusion of all units becomes crucial to accurately measuring Fitness. However, a potential challenge arises due to the possibility of highly matched HIs during the training phase, which might result in a misleadingly high Fitness score when confronted with an unmatched (outlier) HI from a specific unit during the testing phase. Therefore, it is essential to evaluate HI metrics solely based on the test unit.

The choice of using the singular noun "test unit" instead of the plural "test units" is a reflection of the specific context within this work. In this study, we are focused on a single unit designated for testing the model. This decision is dictated by the limited number of units available, with only 12 SSPs allocated for training and testing in the sequence-to-sequence (seq2seq) problem. However, it is noteworthy that similar functions, as will be briefly explained for each, could be readily extended to accommodate multiple test units.

Metrics, particularly Pr, are refined to focus more on the test units rather than incorporating the entire set (train/validation/test) of units. This adjustment is aimed at preventing the false positive influence of training units on final scores, which might overshadow the low scores of the test unit. By primarily considering the test unit, these updated metrics aim to determine if it deviates significantly from the training units and to what extent. It is noteworthy that if the metrics yield higher values for only the test unit while remaining lower for the training units, the methodology could be rendered ineffective. Conversely, if the metrics are consistently high for the training units, then achieving high scores for the test unit becomes meaningful [55].

The definition of Mo remains consistent, and its computation for a single unit follows a straightforward process involving the internal summation of Eq. (9):

$$Mo^{\tau} = \left|\frac{1}{N_{\tau} - 1}\sum_{i=1}^{N_{\tau}}\frac{\sum_{p=1,p>i}^{N_{\tau}}(t_p - t_i).sgn(x(t_p) - x(t_i))}{\sum_{p=1,p>i}^{N_{\tau}}(t_p - t_i)}\right|.100\% \quad (13)$$

where the symbol τ represents the test unit. In cases where multiple units are being considered for testing, Eq. (13) will be similar to Eq. (9), with the difference being the inclusion of an averaging procedure across only the test units.

However, Pr requires redefinition. In this adaptation, rather than considering the standard deviation of HIs at EoL across all units in the numerator, the deviation of the HI at EoL for the test unit from the deviation basis (i.e., its corresponding value averaged over the training units) is computed:

$$Pr^{\tau} = exp\left(- \frac{\left| x_{\tau}(N_{\tau}) - \overbrace{\left[\frac{1}{M^{\tau'}} \sum_{i=1}^{M^{\tau'}} x_i(N_i) \right]}^{deviationbasis} \right|}{\underbrace{\frac{1}{M} \sum_{j=1}^{M} |x_j(1) - x_j(N_j)|}_{scalingfactor}} \right) \tag{14}$$

where τ' represents all units set except for the test ones and $M^{\tau'}$ signifies the count of those units, including the training ones (or even the validation ones). It should be noted that $x_j(1)$ and $x_j(N_j)$ denote the HI values of the jth unit at the initiation and EoL, respectively. The denominator serves as a scaling factor, which in this case corresponds to the mean value of the difference between HIs at the beginning and EoL across all units ($\tau \cup \tau'$) or the training units (τ'). After evaluating both options within this study, the mean value over all units is considered, as symbolized in Eq. (14). When dealing with multiple units (M^{τ}) under consideration for testing, the deviation basis of EoL could be established based on the training set (τ'), the test set (τ), or a combination of both ($\tau \cup \tau'$), among which the latest with M units is advisable:

$$Pr^{\tau} = exp\left(- \frac{\sqrt{\frac{1}{M^{\tau}} \sum_{j=1}^{M^{\tau}} \left| x_j(N_j) - \overbrace{\left[\frac{1}{M} \sum_{i=1}^{M} x_i(N_i) \right]}^{deviationbasis} \right|^2}}{\underbrace{\frac{1}{M} \sum_{j=1}^{M} |x_j(1) - x_j(N_j)|}_{scalingfactor}} \right) \tag{15}$$

Regarding Tr, it is crucial to note that the minimum correlation of HIs should be computed between two distinct units, which is not feasible when considering only a single test unit while excluding the training units. Additionally, if the correlation between the HI of the test unit and the HIs of the training units is calculated in a pairwise manner, followed by selecting the minimum as Tr, this value might surpass the correlation computed when considering all units' HIs pairwise. Consequently, to ensure a more stringent evaluation, the same formula as Eq. (11) is again applied to the test units. This approach is more rigorous and maintains consistency.

With considering the updated metrics, the Fitness metric for the test unit is as follows:

$$Fitness^{\tau} = a.Mo^{\tau}_{HI} + b.Pr^{\tau}_{HI} + c.Tr_{HI} \tag{16}$$

where Tr_{HI} is the same as before.

3.5.2. Base learner models

In continuation of the authors' previous work [46], a SSDNN approach is proposed to create the HI. A SSL paradigm is employed due to its ability to exploit unlabeled data to create more potent regressors. The Fitness function as well as the available lifetimes of the

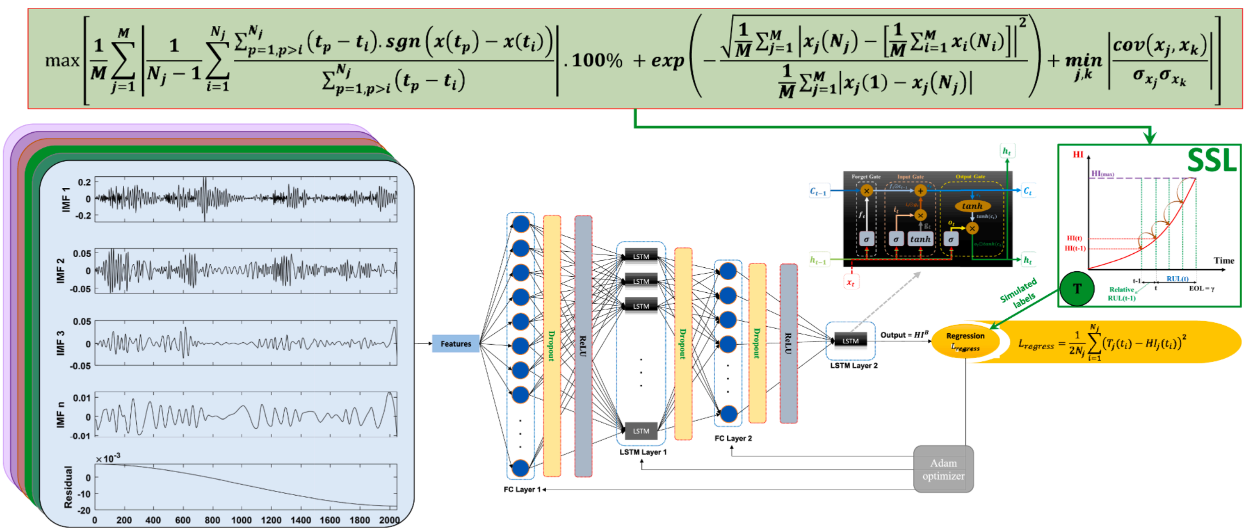

Fig. 3. The general architecture of the semi-supervised base learner models.

units are implemented in the network targets. First, a hypothetical HI is formulated that conforms to the prognostic metrics. It was found [46] that the hypothetical HI that best matches these assumptions is:

$$HI_{(t)} = \frac{t^2}{t_{EOL}^2} \tag{17}$$

where t is the running time and t_{EOL} is the EoL operational time of the available training SSPs. Eq. (17) provides a HI in the range [0, 1], which can also be scaled by simply a multiplication in a coefficient [10,24,56]. The normalization via t_{EOL}^2 is necessary to obtain the desired values for prognosability since each unit has a different lifetime. It is highlighted that this hypothetical HI is only proposed for the available training samples, where the lifetimes are known a priori.

A multi-layer LSTM network is proposed to perform the feature fusion task, and the general network architecture can be observed in Fig. 3.

The half-mean-squared error is regarded as the loss function of the seq2seq regression network:

$$L_{regress} = \frac{1}{2N_j}\sum_{i=1}^{N_j}\left(T_j(t_i) - HI_j(t_i)\right)^2 \tag{18}$$

where $T_j(t_i) = {t_i}^2/t_{EOL_j}^2$ denotes the target value and $HI_j(t_i)$ is the output of the last layer for the j$^{\text{th}}$ unit at time step i.

The general architecture of the network was fixed, while hyperparameters, including the number of neurons in the fully connected (FC) layers, the number of units in the first LSTM layer, dropout (D) value, and batch size, were tuned using Bayesian optimization [57] given the statistical features extracted from the time and frequency domains of FFT in [46]. To this end, a holdout validation scheme is employed, with 10 SSPs used for training, 1 for validation, and 1 for testing, to evaluate the hyperparameters of the network.

Ten top models, according to the lowest root-mean-square error (RMSE) (the maximum RMSE over all SSPs was regarded as the objective function), are considered the base learners in the current work, where the new features extracted from the IMFs of CEEMDAN are imported. Although the process of optimizing the hyperparameters based on the new feature inputs could be repeated, this step is skipped to save time. It should be noted that this section can be viewed as a type of transfer learning designed to cut down on time. The relevant hyperparameter information after Bayesian optimization can be seen in Table 3.

3.5.3. Ensemble learner models

Ensemble learning (EL) techniques can be used to build a *meta*-model that can address the inherent randomness in ML models and uncertainties in model structure after HIs have been extracted from base models. This is especially relevant when there is a small sample size, as there was in the present research. The reliability and accuracy of an ensemble prediction model can be improved by effectively leveraging the strengths of various single prediction models [58,59]. EL techniques fall into three main categories: bagging [60], boosting [61], and stacking [62]. Bagging includes bootstrapping (random sampling) and aggregation (averaging the outputs of the base learners). An example of this is random forest [63]. Boosting combines sequentially arranged base learners, such as in adaptive boosting (AdaBoost) [64] and extreme gradient boosting (XGB) [65]. Stacking uses heterogeneous base learner models through training a *meta*-model (blender) as opposed to the earlier methods' homogeneous base learner ones.

From another point of view, ensemble models that rely on averaging can be broadly divided into two types: simple averaging ensemble (SAE) [66] and weighted averaging ensemble (WAE) [67], with the first being a particular instance of the last. Predictions from various trained base models are equally combined by the EL model in SAE. The fact that each model contributes equally to the ensemble prediction, regardless of how well it performs, is a limitation. Contrarily, WAE allows for a more refined contribution by allocating weights to ensemble members based on confidence in their predictive qualities.

Despite the fact that a variety of EL methods and models (including random forest, least squares linear, support vector machine, boosting, Bayesian linear regression, Gaussian process regression, and Gaussian kernel regression) were investigated, the presented work concentrates on averaging ensemble models using various weighting techniques and DL ensemble models. The decision to use

Table 3
Base learner models' hyperparameters.

Model (rank)	**Hyperparameters** [search space] I: integer or D: decimal				
	Batch Size **I: [1, 5]**	Dropout **D: [0, 0.5]**	FCL1 **I: [1, 201]**	LSTM1 **I: [1, 256]**	FCL2 **I: [1, 50]**
1	4	0.3	110	154	50
2	5	0.4	124	83	48
3	5	0.5	201	79	36
4	5	0.4	152	81	27
5	5	0.5	41	142	43
6	2	0	30	256	45
7	5	0.4	124	56	48
8	5	0.1	137	20	39
9	5	0.4	161	92	48
10	5	0.4	53	120	50

these models over others was made based on their superior performance and effectiveness.

Initially, leave-one-out cross-validation (LOOCV) [68] is employed, where a single unit (SSP) is designated for testing, leaving 11 specimens for subsequent processing. With this in mind, three main dataset divisions can be made: (A) considering one test SSP without validation (training with a fixed number of epochs); (B) considering the test SSP itself as validation (intended to prevent overfitting); and (C) considering another SSP other than the test SSP as validation. Given the small number of specimens available in this study, Ccase B is less generalized yet nonetheless partially valid. Therefore, cases B and C are investigated in the current work, where in the latter, the validation SSP is randomly chosen from the 11 left SSPs. The base learner models are totally trained 100 times in such a way that, for Case C, the validation unit is randomly selected 10 times, and then the learning process is conducted with 10 different random seed numbers [69] for initializing weights and biases. The 100 HIs predicted by the base learner models are then ensembled using a process involving SAE, WAE, and finally DL models.

The overall WAE can be stated as follows:

$$f_{WAE} = \sum_{k=1}^{K} \overline{\omega}_k f_k \tag{19}$$

where f_k indicates the k^{th} individual model and $\overline{\omega}_k$ is its normalized weight:

$$\overline{\omega}_k = \frac{\omega_k}{\sum_{k=1}^{K} \omega_k} \tag{20}$$

where ω_i denotes the k^{th} individual base model's weight, which can be specified according to diverse error metrics. In the current study, the mean square deviation (MSE), RMSE, and the model's Fitness are considered as the error metrics. Both MSE and RMSE are computed by comparing the predictions of the k^{th} base model ($HI^{k(E)}$, where (E) denotes the ensemble HI) and the simulated HIs (T):

$$\omega_k^{MSE} = \frac{1}{MSE\left(T, HI^{k(E)}\right)} = \frac{1}{\frac{1}{M^{\tau'}} \sum_{j\in\tau'} \left[\frac{1}{N_j} \sum_{i=1}^{N_j} \left(T_j(t_i) - HI_j^{k(E)}(t_i) \right)^2 \right]} \tag{21}$$

$$\omega_k^{RMSE} = \frac{1}{RMSE\left(T, HI^{k(E)}\right)} = \frac{1}{\frac{1}{M^{\tau'}} \sum_{j\in\tau'} \sqrt{\left[\frac{1}{N_j} \sum_{i=1}^{N_j} \left(T_j(t_i) - HI_j^{k(E)}(t_i) \right)^2 \right]}} \tag{22}$$

$$\omega_k^{Fitness} = a.Mo_{HI} + b.Pr_{HI} + c.Tr_{HI} \ , \quad j \in \tau' \tag{23}$$

When all $\overline{\omega}_k$ are uniformly set to one, the SAE approach is implemented.

In addition to averaging ensemble models, another approach involves using a subsequent ML-based model to fuse the predictions. This ML-based EL model can be implemented using DL networks. In this study, we assess 12 networks comprising various types of layers, including FC, LSTM, bidirectional LSTM (BiLSTM), and dropout (D) layers. The architectures (hidden layers) of these EL models are summarized in Table 4, in which the value within parentheses indicates the number of neurons, units, or dropout percentage. For example, Fig. 4 shows the architecture of Model 16—Net(12), which is composed of a BiLSTM layer. The output layer is a seq2seq regression layer that relies on the loss function outlined in Eq. (18).

Table 4
Ensemble learner models.

Model num.	Model name	Architecture (hidden layers)						
1	**SAE**							
2	**WAE-MSE**							
3	**WAE-RMSE**							
4	**WAE-Fitness**							
5	**Net(1)**	FC(10)	D(0.5)	ReLU	FC(1)			
6	**Net(2)**	FC(100)	D(0.5)	ReLU	FC(1)			
7	**Net(3)**	FC(10)	D(0.5)	ReLU	FC(5)	D(0.5)	ReLU	FC(1)
8	**Net(4)**	FC(100)	D(0.5)	ReLU	FC(5)	D(0.5)	ReLU	FC(1)
9	**Net(5)**	LSTM(5)	D(0.5)	FC(5)	D(0.5)	ReLU	FC(1)	
10	**Net(6)**	LSTM(10)	D(0.5)	FC(5)	D(0.5)	ReLU	FC(1)	
11	**Net(7)**	FC(10)	D(0.5)	ReLU	LSTM(5)	D(0.5)	ReLU	FC(1)
12	**Net(8)**	FC(10)	D(0.5)	ReLU	BiLSTM(5)	D(0.5)	ReLU	FC(1)
13	**Net(9)**	BiLSTM(5)	D(0.5)	BiLSTM(1)	D(0.5)	FC(1)		
14	**Net(10)**	BiLSTM(10)	D(0.5)	BiLSTM(1)	D(0.5)	FC(1)		
15	**Net(11)**	BiLSTM(5)	D(0.5)	FC(5)	D(0.5)	ReLU	FC(1)	
16	**Net(12)**	D(0.5)	BiLSTM(5)	D(0.5)	FC(5)	D(0.5)	ReLU	FC(1)

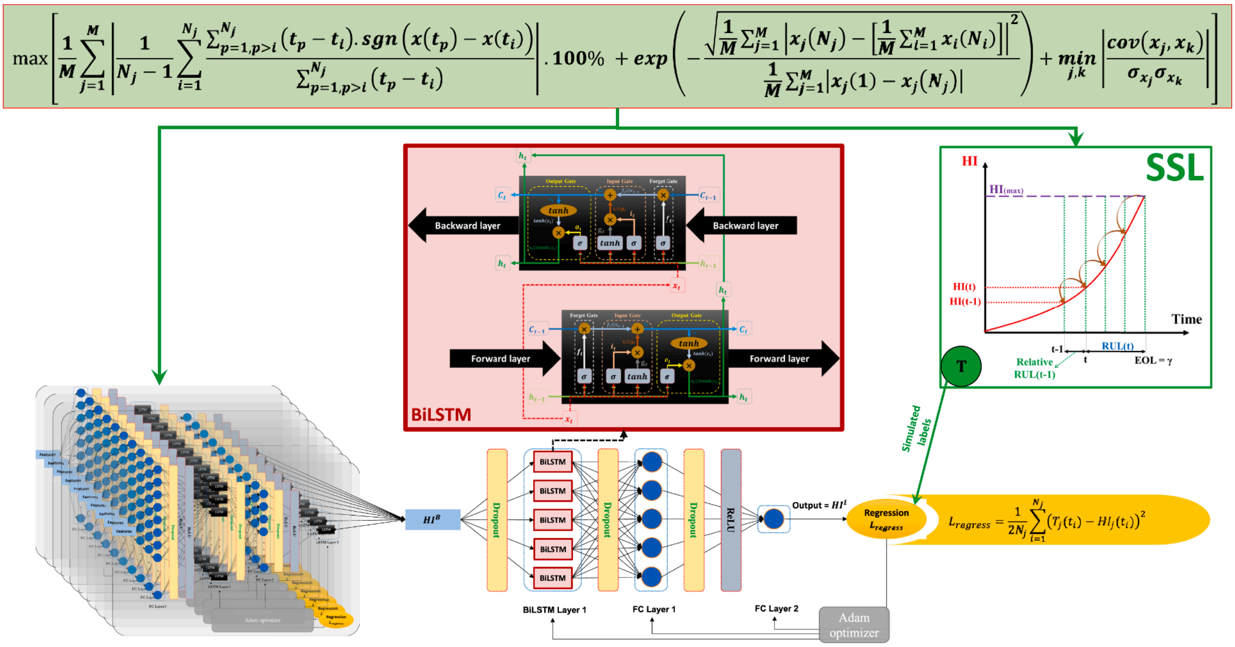

Fig. 4. The architecture of the semi-supervised ensemble learner Model 16 – Net(12).

4. Results and discussion

4.1. Base learner models

To demonstrate the effectiveness of the CEEMDAN features as input to the base learner models, the average Fitness values (and standard deviation) over the 100 repetitions are presented in Table 5. The best obtained value is 2.82 (± 0.24) for Fold 12 and the base learner Model 10. Table 6 summarizes the results over the 100 repetitions for the test units (Eq. (16)) to evaluate the methodology's adaptability to unknown data. The best result is observed for Model 10, Fold 12, with a Fitness 2.75 (± 0.21). In Fig. 5**,** a visual representation of Tables 5 and 6 for Model 10 is displayed for easier comparison between the two. What is observed is that Mo and Pr, and consequently Fitness, display slightly lower values, since the test units are unable to follow the ideal HI with the same proficiency, resulting in overall lower fitness values.

A drawback of the original methodology's evaluation via LOOCV, also presented in [46], is the use of the test unit as validation during training (Case B). This way, it is incorporated into the training step, thus it is not hidden during the application step. To overcome the limitation, a random SSP from the training set is used for model validation (Case C). By employing this methodology, the training dependency on the test unit is eliminated, thus enabling the possibility of real-time implementation.

The average Fitness scores over the 100 repetitions for both the entire and the test units are shown in Tables 7 and 8, respectively, while Fig. 6 presents both cases for the base learner Model 9. Comparing with data division Case B, it is evident that the values obtained are overall lower. However, the values remain high, providing HIs with great prognostic potential. The best values are observed for Fold 12 and Model 9 at 2.51 (± 0.43) for the entire set and 2.44 (± 0.4) for the test unit.

Model 10 (iteration 96) achieved the best HI for Case B, while Model 9 (iteration 97) did so for Case C, as shown in Fig. 7**(a) and (b)**. At first glance, it is observed that not every fold is able to create a HI close to the ideal simulated one. For instance, in Fold 3 of Fig. 7**(a)**, not even the training units are able to fit the ideal HI, although the different units display a similar trend. On the other hand, for Fold 10, the training SSPs display a behavior close to the ideal one; however, the test unit fails to reproduce this trend. Contrary to Fig. 7**(a)**, the test units in Case C deviate from the training units quite significantly and in more folds, as seen in Fig. 7**(b)**. This is also observed by the drop in average Fitness values in Table 8. This suggests that the base learners are struggling to generalize the good results of training to unseen data.

4.2. Ensemble learner models

In Table 9, the ensemble models' Fitness averaged over the 12 Folds is presented for the different ensemble models, where the base learner is trained with Case B (the test unit as validation). When considering the entire set (training and testing) it is observed that Fitness is mostly over 2.6. What is surprising is the performance of WAE-MSE and WAE-RMSE, which are underperforming even compared to SAE. The best Fitness result is obtained using the 16th EL model, Net(12), which is one of the more complex networks, where the Fitness value is 2.84 (± 0.13) and is obtained for base learner Model 9, while base learner Model 3 also achieves a similar

Table 5
Fitness values for base learner models averaged over the 100 repetitions using Eq.(12) considering all units and data division Case B.

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10	Fold 11	Fold 12
Model 1	2.07 (±0.67)	2.21 (±0.61)	1.82 (±0.65)	2.75 (±0.26)	2.58 (±0.48)	2.81 (±0.17)	2.45 (±0.5)	1.98 (±0.35)	2.26 (±0.57)	1.26 (±0.3)	2.48 (±0.48)	2.71 (±0.23)
Model 2	2.35 (±0.6)	2.24 (±0.64)	1.66 (±0.63)	2.74 (±0.26)	2.59 (±0.45)	2.69 (±0.35)	2.49 (±0.44)	2.07 (±0.38)	2.25 (±0.62)	1.34 (±0.35)	2.48 (±0.54)	2.6 (±0.38)
Model 3	2.58 (±0.41)	2.12 (±0.68)	1.67 (±0.61)	2.68 (±0.28)	2.43 (±0.47)	2.65 (±0.31)	2.56 (±0.35)	2.16 (±0.38)	1.97 (±0.71)	1.47 (±0.48)	2.49 (±0.49)	2.54 (±0.33)
Model 4	2.22 (±0.6)	2.09 (±0.66)	1.61 (±0.64)	2.71 (±0.31)	2.5 (±0.45)	2.72 (±0.38)	2.54 (±0.43)	2.1 (±0.37)	2.17 (±0.62)	1.42 (±0.43)	2.32 (±0.58)	2.58 (±0.32)
Model 5	2.56 (±0.65)	1.71 (±0.62)	2.35 (±0.57)	2.7 (±0.27)	2.75 (±0.3)	2.59 (±0.38)	2.48 (±0.46)	1.93 (±0.33)	2.5 (±0.44)	1.32 (±0.27)	2.42 (±0.56)	2.78 (±0.25)
Model 6	2.07 (±0.65)	1.82 (±0.58)	2.1 (±0.55)	2.39 (±0.5)	2.33 (±0.55)	2.67 (±0.26)	2.31 (±0.51)	1.82 (±0.34)	2.06 (±0.61)	1.39 (±0.28)	1.93 (±0.58)	2.51 (±0.39)
Model 7	2.38 (±0.65)	2.12 (±0.7)	1.72 (±0.63)	2.75 (±0.21)	2.54 (±0.54)	2.76 (±0.28)	2.33 (±0.59)	2.17 (±0.36)	2.01 (±0.69)	1.39 (±0.41)	2.53 (±0.47)	2.6 (±0.31)
Model 8	2.23 (±0.6)	2.12 (±0.58)	1.74 (±0.54)	2.53 (±0.53)	2.46 (±0.53)	2.63 (±0.47)	2.24 (±0.59)	2.13 (±0.49)	1.91 (±0.68)	1.75 (±0.51)	2.24 (±0.5)	2.51 (±0.38)
Model 9	2.34 (±0.62)	2.31 (±0.61)	1.7 (±0.67)	2.72 (±0.24)	2.46 (±0.54)	2.7 (±0.34)	2.51 (±0.41)	2.17 (±0.37)	2.33 (±0.54)	1.34 (±0.42)	2.58 (±0.4)	2.61 (±0.33)
Model 10	2.36 (±0.76)	1.95 (±0.69)	1.98 (±0.7)	2.75 (±0.19)	2.74 (±0.35)	2.64 (±0.38)	2.47 (±0.49)	1.98 (±0.26)	2.56 (±0.36)	1.35 (±0.3)	2.48 (±0.5)	**2.82 (±0.24)**

Table 6
Fitness values for base learner models averaged over the 100 repetitions using Eq.(16) considering only test unit and data division Case B.

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10	Fold 11	Fold 12
Model 1	1.82 (±0.69)	1.82 (±0.52)	1.51 (±0.56)	2.51 (±0.37)	2.53 (±0.39)	2.63 (±0.26)	2.33 (±0.45)	1.61 (±0.31)	1.9 (±0.51)	1.26 (±0.25)	2.2 (±0.39)	2.66 (±0.2)
Model 2	2.13 (±0.66)	1.93 (±0.59)	1.5 (±0.49)	2.56 (±0.3)	2.54 (±0.38)	2.53 (±0.44)	2.34 (±0.43)	1.65 (±0.35)	1.99 (±0.48)	1.28 (±0.24)	2.32 (±0.45)	2.58 (±0.34)
Model 3	2.44 (±0.48)	1.89 (±0.57)	1.42 (±0.55)	2.48 (±0.34)	2.46 (±0.37)	2.54 (±0.34)	2.46 (±0.35)	1.81 (±0.38)	1.82 (±0.61)	1.39 (±0.31)	2.35 (±0.47)	2.55 (±0.3)
Model 4	1.96 (±0.63)	1.79 (±0.5)	1.44 (±0.54)	2.49 (±0.39)	2.45 (±0.44)	2.56 (±0.37)	2.41 (±0.4)	1.7 (±0.33)	1.88 (±0.49)	1.3 (±0.27)	2.11 (±0.49)	2.55 (±0.28)
Model 5	2.51 (±0.7)	1.46 (±0.47)	2.03 (±0.5)	2.48 (±0.35)	2.7 (±0.24)	2.43 (±0.42)	2.25 (±0.44)	1.59 (±0.29)	2.07 (±0.36)	1.26 (±0.21)	2.19 (±0.45)	2.72 (±0.22)
Model 6	1.84 (±0.71)	1.49 (±0.47)	1.86 (±0.46)	2.1 (±0.54)	2.37 (±0.46)	2.45 (±0.27)	2.2 (±0.45)	1.48 (±0.28)	1.75 (±0.49)	1.27 (±0.26)	1.79 (±0.43)	2.43 (±0.37)
Model 7	2.2 (±0.69)	1.86 (±0.54)	1.52 (±0.5)	2.57 (±0.29)	2.49 (±0.45)	2.64 (±0.29)	2.23 (±0.51)	1.76 (±0.33)	1.8 (±0.56)	1.27 (±0.25)	2.33 (±0.42)	2.59 (±0.28)
Model 8	1.86 (±0.56)	1.66 (±0.51)	1.33 (±0.46)	2.24 (±0.5)	2.41 (±0.45)	2.44 (±0.4)	2.06 (±0.5)	1.69 (±0.42)	1.7 (±0.54)	1.43 (±0.38)	2.05 (±0.44)	2.48 (±0.34)
Model 9	2.19 (±0.64)	2.03 (±0.52)	1.48 (±0.53)	2.51 (±0.29)	2.45 (±0.45)	2.55 (±0.38)	2.38 (±0.42)	1.75 (±0.36)	2.06 (±0.48)	1.3 (±0.25)	2.37 (±0.38)	2.6 (±0.3)
Model 10	2.26 (±0.76)	1.7 (±0.54)	1.76 (±0.58)	2.52 (±0.31)	2.68 (±0.31)	2.41 (±0.42)	2.24 (±0.46)	1.63 (±0.22)	2.08 (±0.4)	1.34 (±0.22)	2.21 (±0.4)	**2.75 (±0.21)**

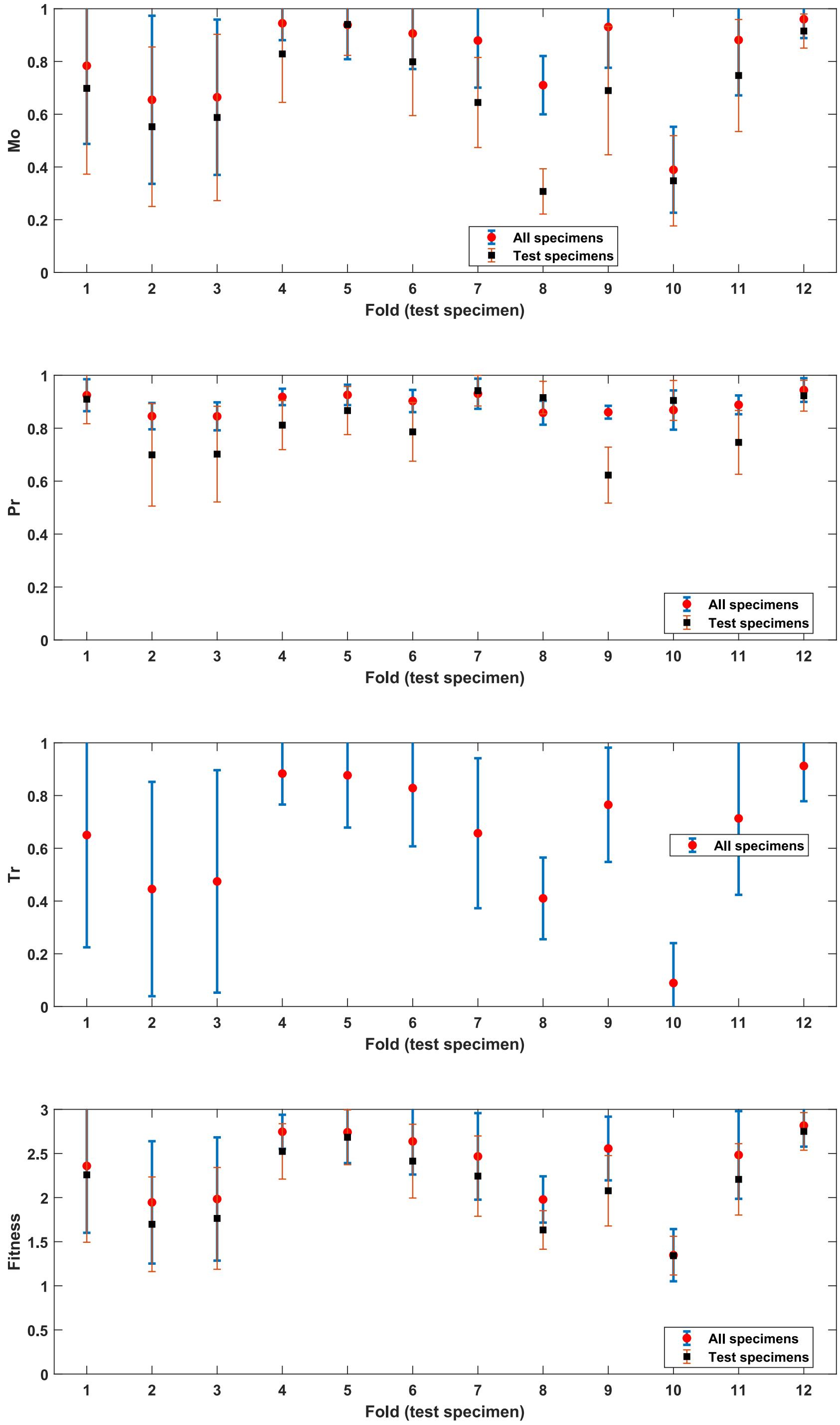

Fig. 5. The prognostic metrics distribution based on all and test units for the base learner Model 10, considering Case B for the dataset division (the test SSP itself as validation).

value of 2.84 (±0.15).

To evaluate the performance of the models only on unknown data of the test unit, Table 10 demonstrates the Fitness values calculated using Eq. (16). The values have overall decreased, which is expected since the test unit does not fit the ideal behavior perfectly and diverges from the average behavior of the training units. The best Fitness is again obtained for the Net(12) ensemble upon Model 3, with a value of 2.74 (±0.33). The results of Tables 9 and 10 are also visually summarized for Model 3 in Fig. 8. It is evident that the Fitness has only slightly reduced overall when only considering the test units, which demonstrates the effectiveness of the

Table 7
Fitness values for base learner models averaged over the 100 repetitions using Eq.(12) considering all units and data division Case C.

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10	Fold 11	Fold 12
Model 1	1.8 (±0.55)	2 (±0.59)	2.08 (±0.61)	2.51 (±0.38)	2.18 (±0.62)	2.03 (±0.65)	1.67 (±0.5)	1.93 (±0.54)	2.02 (±0.49)	1.96 (±0.36)	2.06 (±0.51)	2.43 (±0.46)
Model 2	1.72 (±0.56)	2 (±0.6)	2.07 (±0.55)	2 (±0.67)	2.21 (±0.55)	2.09 (±0.6)	1.85 (±0.57)	2.21 (±0.49)	2.2 (±0.41)	1.94 (±0.45)	1.92 (±0.57)	2.38 (±0.56)
Model 3	2.07 (±0.49)	2.03 (±0.54)	2.12 (±0.54)	2.19 (±0.55)	2.19 (±0.59)	2.08 (±0.65)	1.95 (±0.59)	1.65 (±0.52)	2.1 (±0.5)	1.99 (±0.45)	2.09 (±0.53)	2.18 (±0.61)
Model 4	1.97 (±0.46)	1.96 (±0.64)	1.86 (±0.55)	2.19 (±0.55)	1.98 (±0.67)	2.21 (±0.69)	1.85 (±0.61)	1.99 (±0.55)	2.03 (±0.45)	1.83 (±0.47)	2.02 (±0.53)	2.27 (±0.55)
Model 5	1.6 (±0.52)	2.17 (±0.5)	2.01 (±0.6)	2.17 (±0.59)	2.2 (±0.61)	2.04 (±0.55)	1.56 (±0.4)	2.02 (±0.37)	1.95 (±0.58)	1.8 (±0.42)	2 (±0.54)	2.07 (±0.55)
Model 6	1.71 (±0.4)	1.88 (±0.53)	1.84 (±0.51)	2.02 (±0.53)	2.11 (±0.52)	1.92 (±0.48)	1.73 (±0.4)	1.79 (±0.41)	1.87 (±0.4)	1.86 (±0.41)	1.83 (±0.42)	1.92 (±0.57)
Model 7	2.09 (±0.51)	2.15 (±0.58)	2.05 (±0.52)	2.26 (±0.57)	2.35 (±0.53)	2.3 (±0.57)	1.75 (±0.53)	2.2 (±0.53)	1.97 (±0.54)	2.1 (±0.39)	2.08 (±0.53)	2.33 (±0.55)
Model 8	1.8 (±0.45)	2.02 (±0.55)	1.7 (±0.45)	2.21 (±0.51)	2.01 (±0.59)	2.17 (±0.56)	1.75 (±0.48)	2.11 (±0.49)	1.97 (±0.47)	1.82 (±0.39)	1.94 (±0.51)	2.17 (±0.58)
Model 9	1.95 (±0.55)	2.22 (±0.51)	2.12 (±0.56)	2.27 (±0.56)	2.36 (±0.5)	2.21 (±0.67)	1.91 (±0.53)	1.86 (±0.62)	2.03 (±0.51)	1.93 (±0.5)	2.15 (±0.56)	**2.51 (±0.43)**
Model 10	1.97 (±0.56)	2.18 (±0.51)	2.05 (±0.5)	1.93 (±0.55)	2.1 (±0.65)	2.1 (±0.58)	1.78 (±0.5)	1.92 (±0.47)	1.92 (±0.45)	1.8 (±0.42)	1.96 (±0.48)	2.25 (±0.57)

Table 8
Fitness values for base learner models averaged over the 100 repetitions using Eq.(16) considering only test unit and data division Case C.

	Fold 1	**Fold 2**	**Fold 3**	**Fold 4**	**Fold 5**	**Fold 6**	**Fold 7**	**Fold 8**	**Fold 9**	**Fold 10**	**Fold 11**	**Fold 12**
Model 1	1.59 (±0.54)	1.62 (±0.52)	1.68 (±0.55)	2.15 (±0.55)	2 (±0.59)	1.91 (±0.55)	1.5 (±0.41)	1.49 (±0.52)	1.47 (±0.53)	1.33 (±0.26)	1.75 (±0.43)	2.38 (±0.46)
Model 2	1.45 (±0.56)	1.68 (±0.53)	1.57 (±0.6)	1.81 (±0.64)	2.03 (±0.57)	1.94 (±0.6)	1.58 (±0.51)	1.75 (±0.5)	1.58 (±0.65)	1.43 (±0.34)	1.7 (±0.5)	2.32 (±0.55)
Model 3	1.77 (±0.63)	1.63 (±0.51)	1.64 (±0.57)	1.85 (±0.6)	2.08 (±0.54)	1.94 (±0.61)	1.79 (±0.58)	1.37 (±0.48)	1.62 (±0.6)	1.45 (±0.32)	1.84 (±0.49)	2.16 (±0.6)
Model 4	1.58 (±0.56)	1.6 (±0.57)	1.41 (±0.52)	1.89 (±0.58)	1.89 (±0.6)	2.12 (±0.62)	1.64 (±0.59)	1.59 (±0.5)	1.5 (±0.57)	1.37 (±0.28)	1.69 (±0.46)	2.2 (±0.55)
Model 5	1.38 (±0.5)	1.77 (±0.41)	1.63 (±0.54)	1.89 (±0.62)	2.05 (±0.6)	1.78 (±0.53)	1.3 (±0.28)	1.33 (±0.53)	1.52 (±0.53)	1.3 (±0.29)	1.73 (±0.46)	1.99 (±0.55)
Model 6	1.38 (±0.42)	1.47 (±0.48)	1.42 (±0.54)	1.62 (±0.55)	2.05 (±0.51)	1.67 (±0.49)	1.5 (±0.37)	1.31 (±0.39)	1.31 (±0.48)	1.36 (±0.37)	1.49 (±0.35)	1.8 (±0.55)
Model 7	1.76 (±0.59)	1.8 (±0.55)	1.52 (±0.58)	1.95 (±0.64)	2.12 (±0.55)	2.07 (±0.53)	1.49 (±0.47)	1.81 (±0.52)	1.48 (±0.57)	1.53 (±0.38)	1.79 (±0.52)	2.25 (±0.52)
Model 8	1.29 (±0.45)	1.59 (±0.5)	1.15 (±0.44)	1.76 (±0.54)	1.85 (±0.52)	1.94 (±0.56)	1.48 (±0.44)	1.62 (±0.48)	1.46 (±0.54)	1.23 (±0.33)	1.63 (±0.5)	2.03 (±0.64)
Model 9	1.67 (±0.59)	1.77 (±0.55)	1.62 (±0.65)	1.98 (±0.59)	2.12 (±0.54)	2.06 (±0.63)	1.64 (±0.51)	1.5 (±0.58)	1.49 (±0.6)	1.46 (±0.3)	1.84 (±0.54)	**2.44 (±0.4)**
Model 10	1.68 (±0.65)	1.75 (±0.46)	1.58 (±0.54)	1.59 (±0.59)	1.94 (±0.65)	1.89 (±0.51)	1.49 (±0.42)	1.37 (±0.46)	1.29 (±0.51)	1.25 (±0.25)	1.63 (±0.41)	2.19 (±0.57)

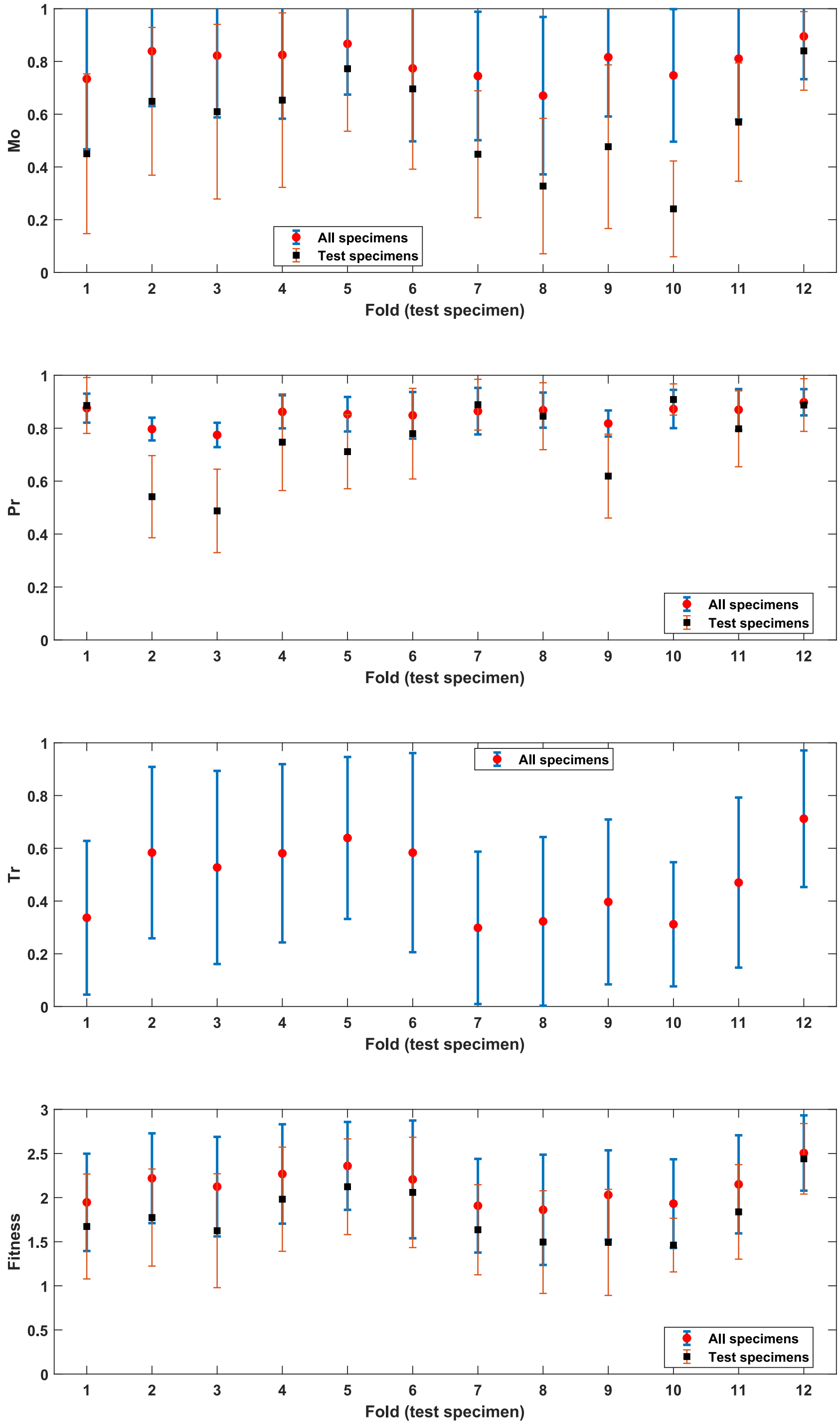

Fig. 6. The prognostic metrics distribution based on all and test units for the base learner Model 10, considering Case C for the dataset division (another SSP other than the test SSP as validation).

methodology.

Using a random SSP as the validation unit (data division Case C) slightly affects the Fitness values of the HIs compared to Case B. Tables 11 and 12 show the average Fitness values (±standard deviation) over 12 Folds, calculated for the entire set (using Eq. (12)) and only for the test unit (using Eq. (16)), respectively. The highest Fitness value is obtained by Net(12) upon Model 9 in both cases with values 2.74 (±0.19) and 2.59 (±0.24), respectively. In Fig. 9, the results of Tables 11 and 12 for base learner Model 9 are visually summarized and compared. The values of the test-only-based Fitness are slightly lower than those for the entire set, especially for the

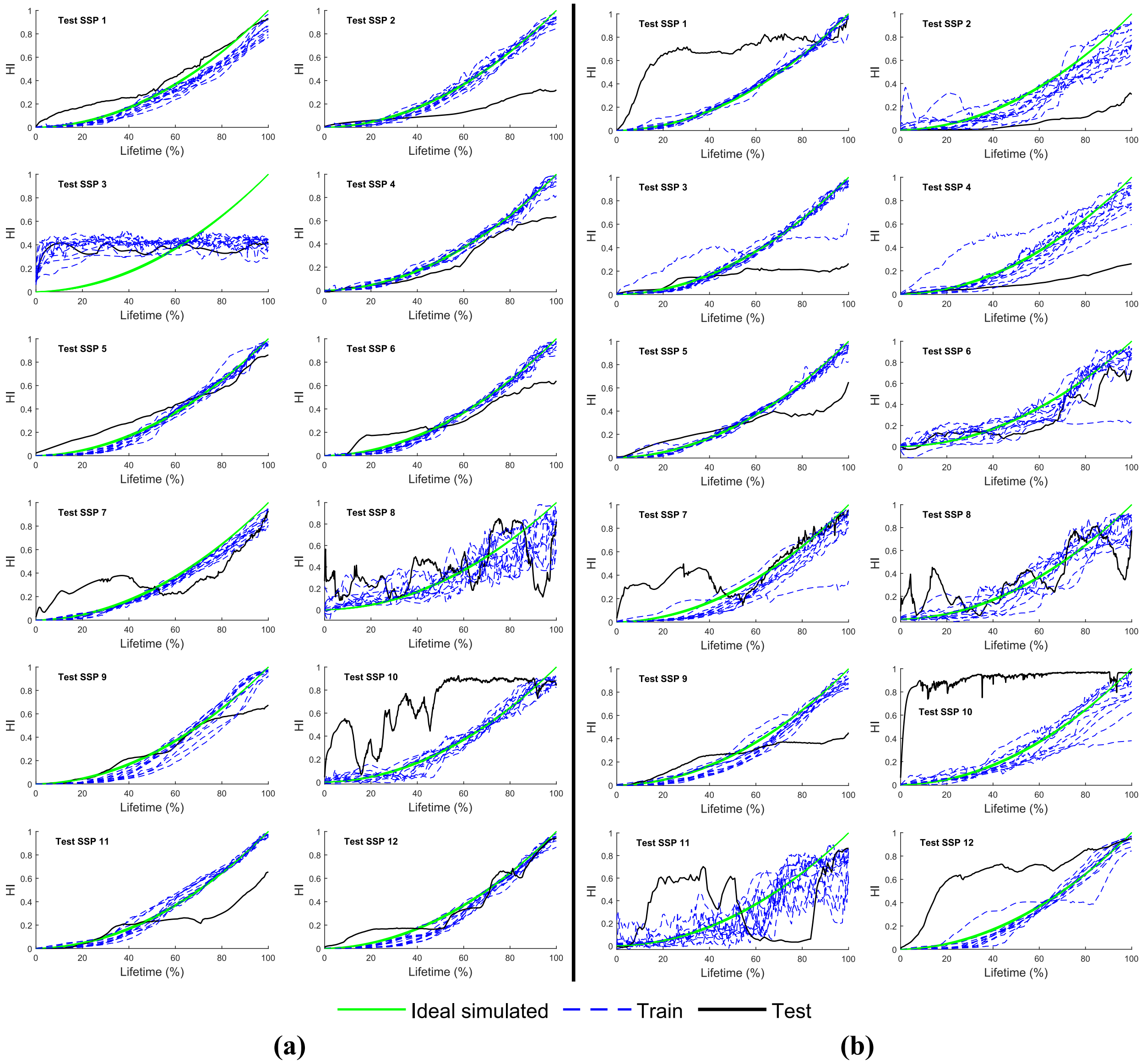

Fig. 7. (a) The HIs constructed by the base Model 10, considering Case B for the dataset division (the test SSP itself as validation), with Fitness 2.42 (±0.52) based on Eq. (16); (b) The HIs constructed by the base Model 9, considering Case C for the dataset division (another SSP other than the test SSP as validation), with Fitness 2.21 (±0.39) based on Eq. (16).

Net(12) ensemble, which displays promising results.

Fig. 10**(a)** displays HIs for different Folds using base learner Model 9 and data division Case B. Each Fold denotes the respective unit as the test. The solid gray line represents the ideal HI and the dotted colored lines represent the training unit results for each ensemble model. The solid-colored lines denote the test unit. It is observed that in some Folds, like Folds 2, 3, 9, and 10, the test unit demonstrates a different behavior than the training units and diverges from both the constructed HIs of the training units and the ideal HI. In the other Folds, the test units display a similar trend to the training. When data division Case C is applied, the difference between training and test units increases, diverging from the ideal HI in most Folds, as displayed in Fig. 10**(b)**. Diverging from the ideal HI is not necessarily a negative attribute as long as training and test units display similar overall trends. The ideal HI is mostly used as a baseline for the training and not a panacea for the HIs to follow. However, there are Folds, like Folds 1 and 12, where the constructed HIs are similar between test and training as well as the ideal HI.

4.3. Discussion

The advantage of not using the test unit as validation in LOOCV lies in enabling the ability to validate the methodology in real

Table 9
Fitness values for Ensemble learning models averaged over the 12 Folds using Eq.(12) considering all units and data division Case B.

	SAE	WAE			DL-based: Net											
		MSE	*RMSE*	*Fitness*	*(1)*	*(2)*	*(3)*	*(4)*	*(5)*	*(6)*	*(7)*	*(8)*	*(9)*	*(10)*	*(11)*	*(12)*
Model 1	2.69 (±0.26)	2.37 (±0.38)	2.61 (±0.29)	2.7 (±0.24)	2.39 (±0.39)	2.75 (±0.19)	2.68 (±0.13)	2.72 (±0.24)	2.72 (±0.31)	2.74 (±0.22)	2.71 (±0.25)	2.63 (±0.18)	2.66 (±0.17)	2.7 (±0.18)	2.76 (±0.22)	2.78 (±0.24)
Model 2	2.75 (±0.19)	2.52 (±0.27)	2.68 (±0.21)	2.75 (±0.18)	2.49 (±0.34)	2.76 (±0.18)	2.74 (±0.12)	2.72 (±0.14)	2.79 (±0.17)	2.75 (±0.2)	2.73 (±0.19)	2.57 (±0.23)	2.66 (±0.13)	2.72 (±0.13)	2.78 (±0.19)	2.81 (±0.18)
Model 3	2.75 (±0.14)	2.51 (±0.34)	2.67 (±0.2)	2.76 (±0.13)	2.53 (±0.39)	2.78 (±0.12)	2.63 (±0.19)	2.68 (±0.17)	2.82 (±0.11)	2.79 (±0.13)	2.67 (±0.19)	2.65 (±0.13)	2.7 (±0.09)	2.75 (±0.11)	2.81 (±0.14)	**2.84 (±0.15)**
Model 4	2.73 (±0.17)	2.44 (±0.27)	2.64 (±0.22)	2.73 (±0.16)	2.69 (±0.21)	2.79 (±0.12)	2.7 (±0.15)	2.72 (±0.12)	2.78 (±0.13)	2.77 (±0.14)	2.72 (±0.14)	2.64 (±0.11)	2.67 (±0.11)	2.72 (±0.11)	2.71 (±0.32)	2.82 (±0.14)
Model 5	2.74 (±0.22)	2.57 (±0.24)	2.7 (±0.21)	2.74 (±0.22)	2.57 (±0.2)	2.76 (±0.17)	2.63 (±0.18)	2.71 (±0.22)	2.8 (±0.14)	2.77 (±0.17)	2.76 (±0.17)	2.63 (±0.16)	2.64 (±0.15)	2.74 (±0.09)	2.78 (±0.2)	2.8 (±0.19)
Model 6	2.62 (±0.27)	2.32 (±0.31)	2.47 (±0.27)	2.63 (±0.26)	2.41 (±0.38)	2.73 (±0.24)	2.46 (±0.34)	2.66 (±0.28)	2.74 (±0.21)	2.74 (±0.23)	2.7 (±0.21)	2.54 (±0.25)	2.63 (±0.18)	2.7 (±0.14)	2.77 (±0.13)	2.77 (±0.17)
Model 7	2.75 (±0.2)	2.41 (±0.38)	2.64 (±0.26)	2.76 (±0.17)	2.66 (±0.15)	2.78 (±0.13)	2.58 (±0.22)	2.66 (±0.37)	2.77 (±0.14)	2.81 (±0.14)	2.74 (±0.18)	2.62 (±0.12)	2.68 (±0.09)	2.73 (±0.11)	2.81 (±0.14)	2.84 (±0.14)
Model 8	2.74 (±0.13)	2.15 (±0.28)	2.59 (±0.15)	2.74 (±0.13)	2.63 (±0.23)	2.75 (±0.14)	2.68 (±0.15)	2.72 (±0.14)	2.77 (±0.13)	2.75 (±0.15)	2.7 (±0.21)	2.66 (±0.16)	2.65 (±0.12)	2.72 (±0.09)	2.77 (±0.15)	2.77 (±0.17)
Model 9	2.76 (±0.15)	2.6 (±0.19)	2.71 (±0.15)	2.76 (±0.15)	2.47 (±0.38)	2.75 (±0.17)	2.62 (±0.23)	2.62 (±0.34)	2.79 (±0.13)	2.75 (±0.16)	2.71 (±0.2)	2.62 (±0.18)	2.66 (±0.15)	2.72 (±0.13)	2.76 (±0.19)	**2.84 (±0.13)**
Model 10	2.75 (±0.24)	2.53 (±0.24)	2.7 (±0.23)	2.75 (±0.23)	2.64 (±0.26)	2.77 (±0.15)	2.61 (±0.21)	2.63 (±0.36)	2.77 (±0.15)	2.76 (±0.14)	2.7 (±0.3)	2.61 (±0.17)	2.67 (±0.13)	2.72 (±0.1)	2.8 (±0.15)	2.79 (±0.21)

Table 10
Fitness values for Ensemble learning models averaged over the 12 Folds using Eq.(16) considering only test unit and data division Case B.

	SAE	WAE			DL-based: Net											
		MSE	*RMSE*	*Fitness*	*(1)*	*(2)*	*(3)*	*(4)*	*(5)*	*(6)*	*(7)*	*(8)*	*(9)*	*(10)*	*(11)*	*(12)*
Model 1	2.44 (±0.39)	2.21 (±0.43)	2.39 (±0.41)	2.45 (±0.38)	2.09 (±0.66)	2.44 (±0.41)	2.42 (±0.29)	2.44 (±0.44)	2.48 (±0.51)	2.47 (±0.44)	2.41 (±0.52)	2.33 (±0.37)	2.43 (±0.3)	2.5 (±0.37)	2.5 (±0.45)	2.6 (±0.44)
Model 2	2.53 (±0.33)	2.39 (±0.34)	2.49 (±0.34)	2.53 (±0.33)	2.2 (±0.53)	2.48 (±0.4)	2.51 (±0.31)	2.47 (±0.26)	2.55 (±0.4)	2.51 (±0.4)	2.48 (±0.45)	2.3 (±0.41)	2.44 (±0.31)	2.56 (±0.29)	2.54 (±0.39)	2.67 (±0.37)
Model 3	2.58 (±0.29)	2.38 (±0.42)	2.51 (±0.33)	2.58 (±0.29)	2.3 (±0.62)	2.56 (±0.32)	2.44 (±0.33)	2.42 (±0.4)	2.63 (±0.32)	2.57 (±0.33)	2.47 (±0.36)	2.48 (±0.27)	2.52 (±0.25)	2.59 (±0.28)	2.62 (±0.33)	**2.74 (±0.33)**
Model 4	2.5 (±0.32)	2.29 (±0.35)	2.44 (±0.34)	2.49 (±0.31)	2.44 (±0.4)	2.49 (±0.32)	2.35 (±0.39)	2.44 (±0.35)	2.5 (±0.36)	2.5 (±0.36)	2.45 (±0.36)	2.4 (±0.28)	2.44 (±0.29)	2.53 (±0.29)	2.4 (±0.63)	2.71 (±0.21)
Model 5	2.5 (±0.36)	2.39 (±0.33)	2.48 (±0.36)	2.5 (±0.37)	2.24 (±0.47)	2.44 (±0.43)	2.27 (±0.41)	2.41 (±0.47)	2.52 (±0.4)	2.48 (±0.42)	2.46 (±0.43)	2.37 (±0.34)	2.39 (±0.36)	2.55 (±0.23)	2.54 (±0.44)	2.61 (±0.35)
Model 6	2.39 (±0.38)	2.15 (±0.35)	2.27 (±0.34)	2.4 (±0.37)	2.16 (±0.41)	2.41 (±0.43)	2.05 (±0.65)	2.38 (±0.41)	2.45 (±0.46)	2.47 (±0.45)	2.4 (±0.4)	2.25 (±0.32)	2.39 (±0.36)	2.52 (±0.23)	2.55 (±0.25)	2.66 (±0.26)
Model 7	2.54 (±0.37)	2.29 (±0.43)	2.47 (±0.41)	2.55 (±0.36)	2.42 (±0.31)	2.51 (±0.35)	2.33 (±0.39)	2.45 (±0.47)	2.55 (±0.4)	2.61 (±0.33)	2.51 (±0.41)	2.36 (±0.27)	2.5 (±0.21)	2.55 (±0.29)	2.59 (±0.32)	2.72 (±0.29)
Model 8	2.46 (±0.27)	2.1 (±0.28)	2.4 (±0.25)	2.46 (±0.27)	2.3 (±0.39)	2.39 (±0.35)	2.32 (±0.33)	2.48 (±0.28)	2.5 (±0.32)	2.46 (±0.33)	2.44 (±0.43)	2.36 (±0.28)	2.42 (±0.25)	2.52 (±0.2)	2.48 (±0.34)	2.57 (±0.33)
Model 9	2.55 (±0.3)	2.46 (±0.25)	2.52 (±0.27)	2.54 (±0.32)	2.19 (±0.38)	2.47 (±0.42)	2.34 (±0.45)	2.37 (±0.49)	2.56 (±0.37)	2.51 (±0.37)	2.46 (±0.43)	2.37 (±0.32)	2.45 (±0.36)	2.54 (±0.31)	2.53 (±0.43)	2.69 (±0.3)
Model 10	2.51 (±0.36)	2.37 (±0.34)	2.48 (±0.35)	2.51 (±0.36)	2.34 (±0.46)	2.47 (±0.37)	2.3 (±0.42)	2.34 (±0.52)	2.55 (±0.34)	2.51 (±0.34)	2.46 (±0.5)	2.33 (±0.37)	2.44 (±0.3)	2.52 (±0.27)	2.57 (±0.36)	2.61 (±0.38)

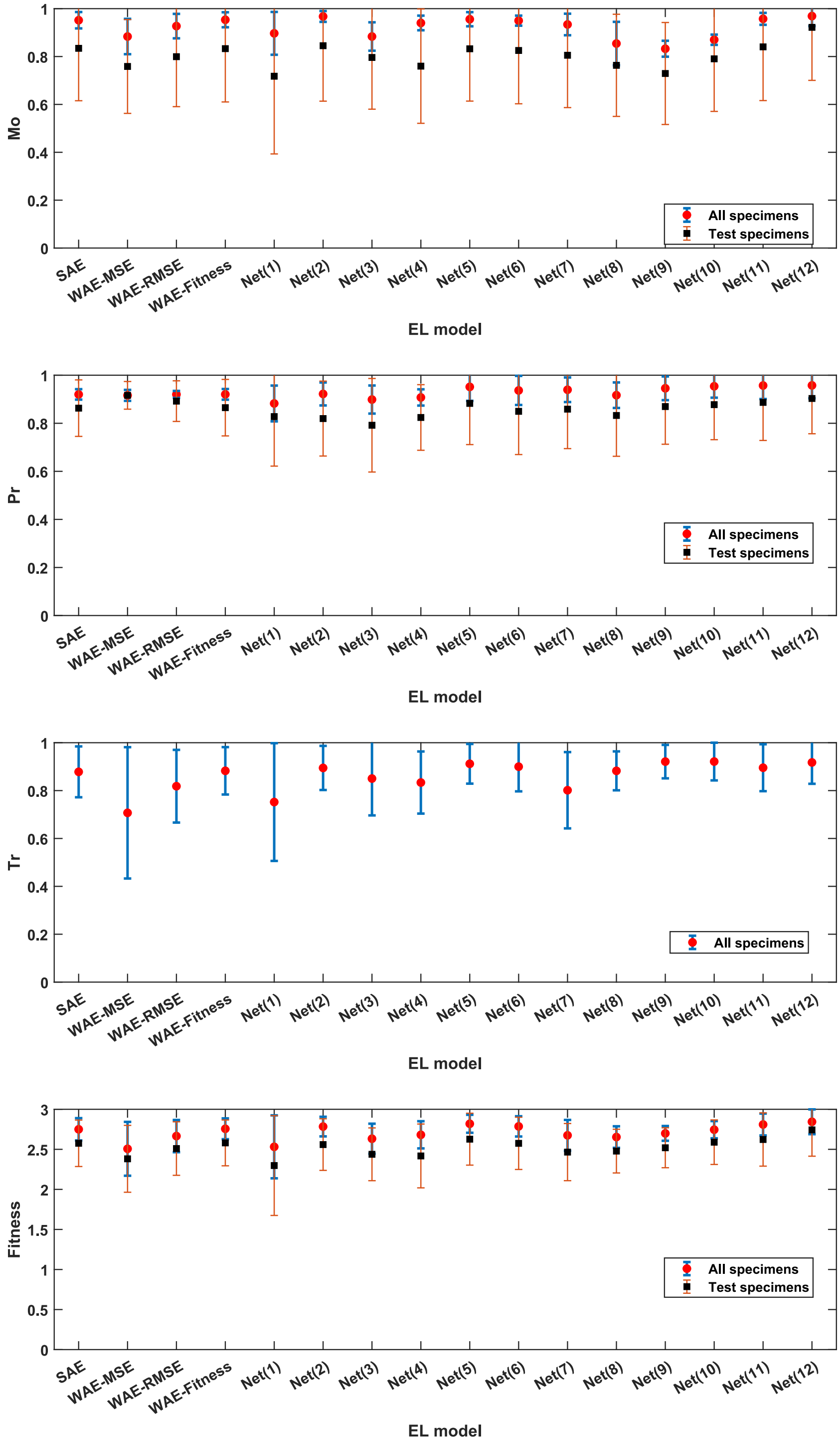

Fig. 8. The prognostic metrics distribution based on all and test units for the base learner Model 3, considering Case B for the dataset division (the test SSP itself as validation).

application scenarios where the test units are new and unknown data. Despite the overall reduction in Fitness of the HI, it remains high, providing a HI that can be useful in prognostic tasks. The best result, while considering only the test unit for the Fitness calculation (Eq. (16)), is obtained for SSP 12 using the base learner Model 9 incorporated with the Net(12) ensemble with a value of 2.94. The extension to calculate test set-specific prognostic standards provides a clear overview of the methodology's performance on unknown data, bringing us closer to evaluating real-time implementation. To the best of the authors' knowledge, this separation between training and

Table 11
Fitness values for Ensemble learning models averaged over the 12 Folds using Eq.(12) considering all units and data division Case C.

	SAE	**WAE**			**DL-based: Net**											
		MSE	***RMSE***	***Fitness***	***(1)***	***(2)***	***(3)***	***(4)***	***(5)***	***(6)***	***(7)***	***(8)***	***(9)***	***(10)***	***(11)***	***(12)***
Model 1	2.62 (±0.2)	2.42 (±0.22)	2.54 (±0.2)	2.62 (±0.19)	2.43 (±0.35)	2.61 (±0.19)	2.52 (±0.28)	2.56 (±0.19)	2.64 (±0.2)	2.59 (±0.19)	2.52 (±0.28)	2.41 (±0.34)	2.5 (±0.21)	2.56 (±0.2)	2.64 (±0.19)	2.63 (±0.2)
Model 2	2.63 (±0.17)	2.48 (±0.23)	2.58 (±0.19)	2.63 (±0.17)	2.35 (±0.28)	2.65 (±0.16)	2.5 (±0.15)	2.55 (±0.24)	2.63 (±0.19)	2.58 (±0.2)	2.61 (±0.18)	2.48 (±0.19)	2.52 (±0.21)	2.59 (±0.17)	2.61 (±0.19)	2.67 (±0.27)
Model 3	2.67 (±0.12)	2.53 (±0.11)	2.62 (±0.12)	2.67 (±0.12)	2.25 (±0.49)	2.69 (±0.11)	2.5 (±0.34)	2.6 (±0.19)	2.69 (±0.14)	2.62 (±0.17)	2.6 (±0.18)	2.58 (±0.13)	2.58 (±0.13)	2.62 (±0.15)	2.67 (±0.18)	2.7 (±0.17)
Model 4	2.64 (±0.17)	2.41 (±0.24)	2.55 (±0.19)	2.64 (±0.17)	2.24 (±0.42)	2.64 (±0.16)	2.52 (±0.23)	2.54 (±0.21)	2.62 (±0.21)	2.6 (±0.22)	2.53 (±0.35)	2.42 (±0.33)	2.54 (±0.16)	2.59 (±0.16)	2.59 (±0.23)	2.64 (±0.31)
Model 5	2.55 (±0.3)	2.47 (±0.18)	2.5 (±0.24)	2.54 (±0.3)	2.37 (±0.36)	2.52 (±0.31)	2.34 (±0.29)	2.44 (±0.34)	2.55 (±0.3)	2.51 (±0.29)	2.52 (±0.27)	2.44 (±0.3)	2.47 (±0.32)	2.52 (±0.29)	2.58 (±0.28)	2.59 (±0.26)
Model 6	2.43 (±0.24)	2.18 (±0.24)	2.28 (±0.25)	2.44 (±0.24)	1.94 (±0.41)	2.53 (±0.23)	2.31 (±0.28)	2.36 (±0.3)	2.54 (±0.27)	2.51 (±0.23)	2.36 (±0.39)	2.32 (±0.29)	2.43 (±0.29)	2.49 (±0.25)	2.6 (±0.23)	2.61 (±0.21)
Model 7	2.65 (±0.15)	2.42 (±0.23)	2.57 (±0.18)	2.65 (±0.16)	2.42 (±0.38)	2.66 (±0.16)	2.37 (±0.25)	2.53 (±0.26)	2.64 (±0.21)	2.63 (±0.17)	2.56 (±0.19)	2.49 (±0.28)	2.54 (±0.17)	2.6 (±0.16)	2.67 (±0.18)	2.69 (±0.16)
Model 8	2.56 (±0.22)	2.17 (±0.27)	2.44 (±0.25)	2.56 (±0.21)	2.4 (±0.29)	2.57 (±0.25)	2.37 (±0.37)	2.45 (±0.24)	2.57 (±0.24)	2.56 (±0.23)	2.45 (±0.34)	2.52 (±0.23)	2.47 (±0.24)	2.56 (±0.2)	2.59 (±0.22)	2.61 (±0.34)
Model 9	2.67 (±0.15)	2.47 (±0.24)	2.6 (±0.17)	2.67 (±0.15)	2.55 (±0.22)	2.67 (±0.16)	2.53 (±0.23)	2.6 (±0.23)	2.64 (±0.19)	2.6 (±0.21)	2.61 (±0.22)	2.43 (±0.35)	2.53 (±0.19)	2.62 (±0.16)	2.62 (±0.25)	**2.74 (±0.19)**
Model 10	2.52 (±0.26)	2.41 (±0.23)	2.45 (±0.25)	2.52 (±0.26)	2.38 (±0.31)	2.56 (±0.26)	2.31 (±0.37)	2.49 (±0.29)	2.58 (±0.22)	2.54 (±0.26)	2.53 (±0.27)	2.4 (±0.3)	2.46 (±0.28)	2.51 (±0.24)	2.53 (±0.27)	2.63 (±0.22)

Table 12
Fitness values for Ensemble learning models averaged over the 12 Folds using Eq.(16) considering only test unit and data division Case C.

	SAE	WAE			DL-based: Net											
		MSE	*RMSE*	*Fitness*	*(1)*	*(2)*	*(3)*	*(4)*	*(5)*	*(6)*	*(7)*	*(8)*	*(9)*	*(10)*	*(11)*	*(12)*
Model 1	2.22 (±0.41)	2.05 (±0.44)	2.16 (±0.38)	2.23 (±0.4)	2.03 (±0.54)	2.18 (±0.41)	2.04 (±0.6)	2.13 (±0.39)	2.33 (±0.35)	2.25 (±0.35)	2.12 (±0.47)	2.02 (±0.55)	2.2 (±0.34)	2.27 (±0.34)	2.3 (±0.37)	2.43 (±0.31)
Model 2	2.23 (±0.34)	2.09 (±0.46)	2.16 (±0.4)	2.22 (±0.34)	1.86 (±0.51)	2.26 (±0.33)	2.12 (±0.34)	2.12 (±0.45)	2.23 (±0.36)	2.22 (±0.37)	2.28 (±0.35)	2.13 (±0.26)	2.22 (±0.34)	2.33 (±0.32)	2.26 (±0.37)	2.43 (±0.51)
Model 3	2.32 (±0.33)	2.18 (±0.33)	2.27 (±0.34)	2.32 (±0.33)	1.84 (±0.64)	2.38 (±0.28)	2.07 (±0.68)	2.23 (±0.41)	2.42 (±0.23)	2.32 (±0.32)	2.32 (±0.32)	2.23 (±0.3)	2.34 (±0.24)	2.41 (±0.26)	2.42 (±0.31)	2.56 (±0.29)
Model 4	2.24 (±0.39)	2.07 (±0.48)	2.18 (±0.4)	2.23 (±0.39)	1.77 (±0.55)	2.23 (±0.38)	2.02 (±0.5)	2.17 (±0.47)	2.22 (±0.43)	2.2 (±0.41)	2.07 (±0.69)	2.06 (±0.4)	2.21 (±0.35)	2.29 (±0.36)	2.23 (±0.43)	2.4 (±0.48)
Model 5	2.1 (±0.5)	2.12 (±0.36)	2.09 (±0.42)	2.09 (±0.51)	1.86 (±0.67)	2 (±0.54)	1.67 (±0.54)	1.93 (±0.56)	2.06 (±0.52)	2.02 (±0.52)	2.04 (±0.53)	2.01 (±0.46)	2.09 (±0.52)	2.15 (±0.47)	2.13 (±0.52)	2.3 (±0.48)
Model 6	1.97 (±0.45)	1.83 (±0.42)	1.89 (±0.43)	1.98 (±0.44)	1.56 (±0.49)	2 (±0.44)	1.73 (±0.56)	1.81 (±0.57)	2.07 (±0.47)	2.05 (±0.42)	1.84 (±0.7)	1.83 (±0.5)	2.05 (±0.46)	2.08 (±0.43)	2.18 (±0.46)	2.29 (±0.43)
Model 7	2.27 (±0.3)	2.07 (±0.42)	2.19 (±0.35)	2.27 (±0.31)	1.95 (±0.56)	2.27 (±0.31)	1.92 (±0.4)	2.06 (±0.55)	2.33 (±0.35)	2.32 (±0.29)	2.23 (±0.32)	2.07 (±0.46)	2.25 (±0.31)	2.36 (±0.19)	2.36 (±0.32)	2.49 (±0.23)
Model 8	2.08 (±0.44)	1.85 (±0.46)	2.02 (±0.47)	2.08 (±0.44)	1.97 (±0.49)	2.06 (±0.49)	1.9 (±0.6)	1.97 (±0.4)	2.13 (±0.47)	2.08 (±0.47)	1.91 (±0.7)	2.07 (±0.47)	2.06 (±0.45)	2.23 (±0.38)	2.16 (±0.47)	2.28 (±0.61)
Model 9	2.3 (±0.32)	2.1 (±0.47)	2.22 (±0.36)	2.3 (±0.32)	2.22 (±0.31)	2.31 (±0.32)	2.13 (±0.44)	2.23 (±0.41)	2.31 (±0.33)	2.29 (±0.36)	2.23 (±0.41)	2.03 (±0.54)	2.25 (±0.28)	2.36 (±0.29)	2.3 (±0.4)	**2.59 (±0.24)**
Model 10	2.02 (±0.5)	2 (±0.46)	1.99 (±0.5)	2.02 (±0.49)	1.94 (±0.6)	2.08 (±0.51)	1.68 (±0.73)	1.99 (±0.54)	2.14 (±0.46)	2.09 (±0.52)	2.11 (±0.53)	1.96 (±0.49)	2.07 (±0.48)	2.11 (±0.43)	2.04 (±0.57)	2.25 (±0.49)

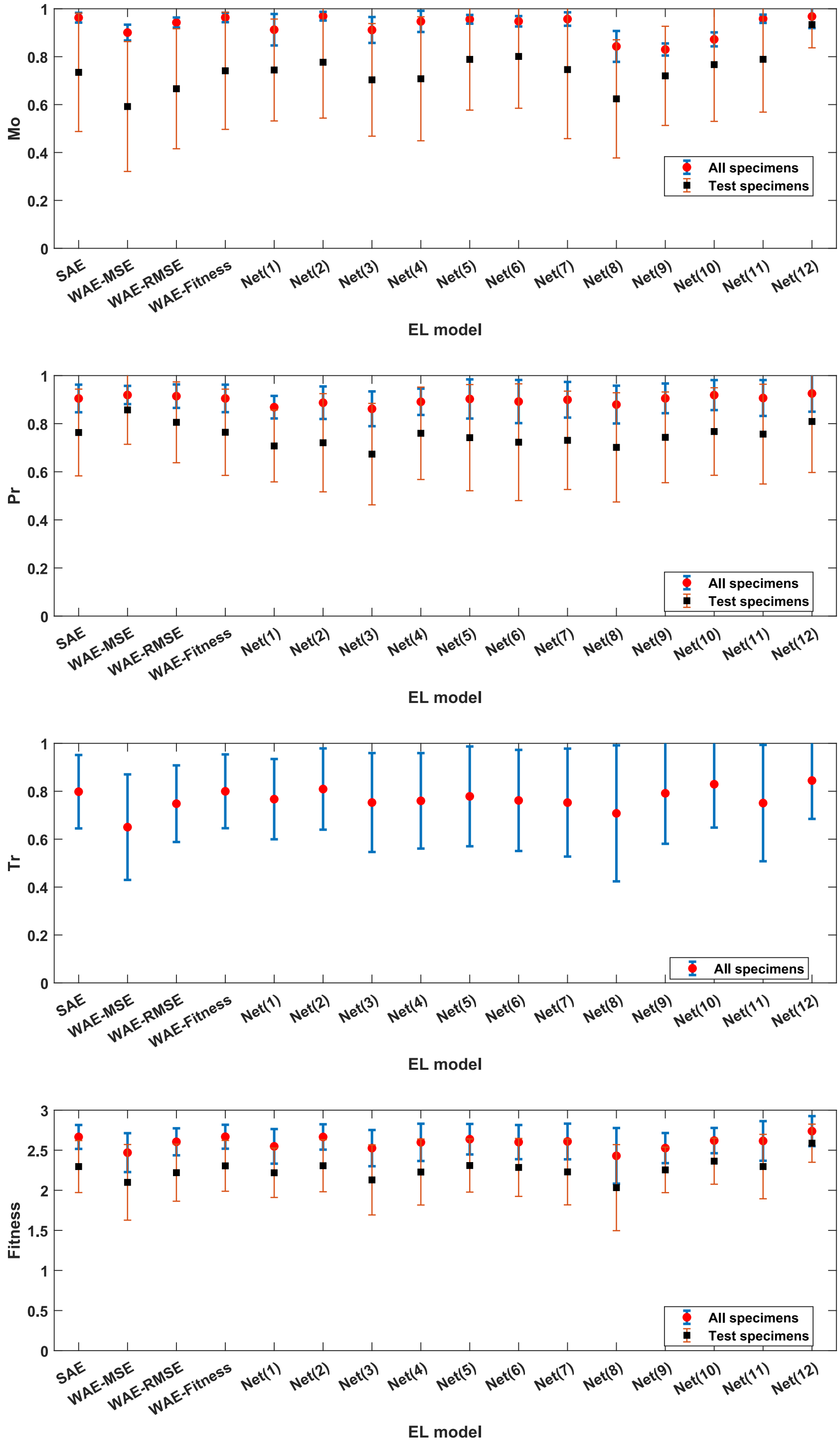

Fig. 9. The prognostic metrics distribution based on all and test units for the base learner Model 9, considering Case C for the dataset division (another SSP other than the test SSP as validation).

testing has not been previously performed and provides a first look at a new way to calculate prognosability, taking into account only the test units.

There have been other attempts to develop HIs for stiffened composite panels. AE-based HIs from complex composite structures have been presented in [70]. In that research, statistical features were extracted from the windowed AE data and used as input into multi-input ML algorithms for RUL estimation. The best-discovered feature displayed average Mo and Pr of 0.693 and 0.871,

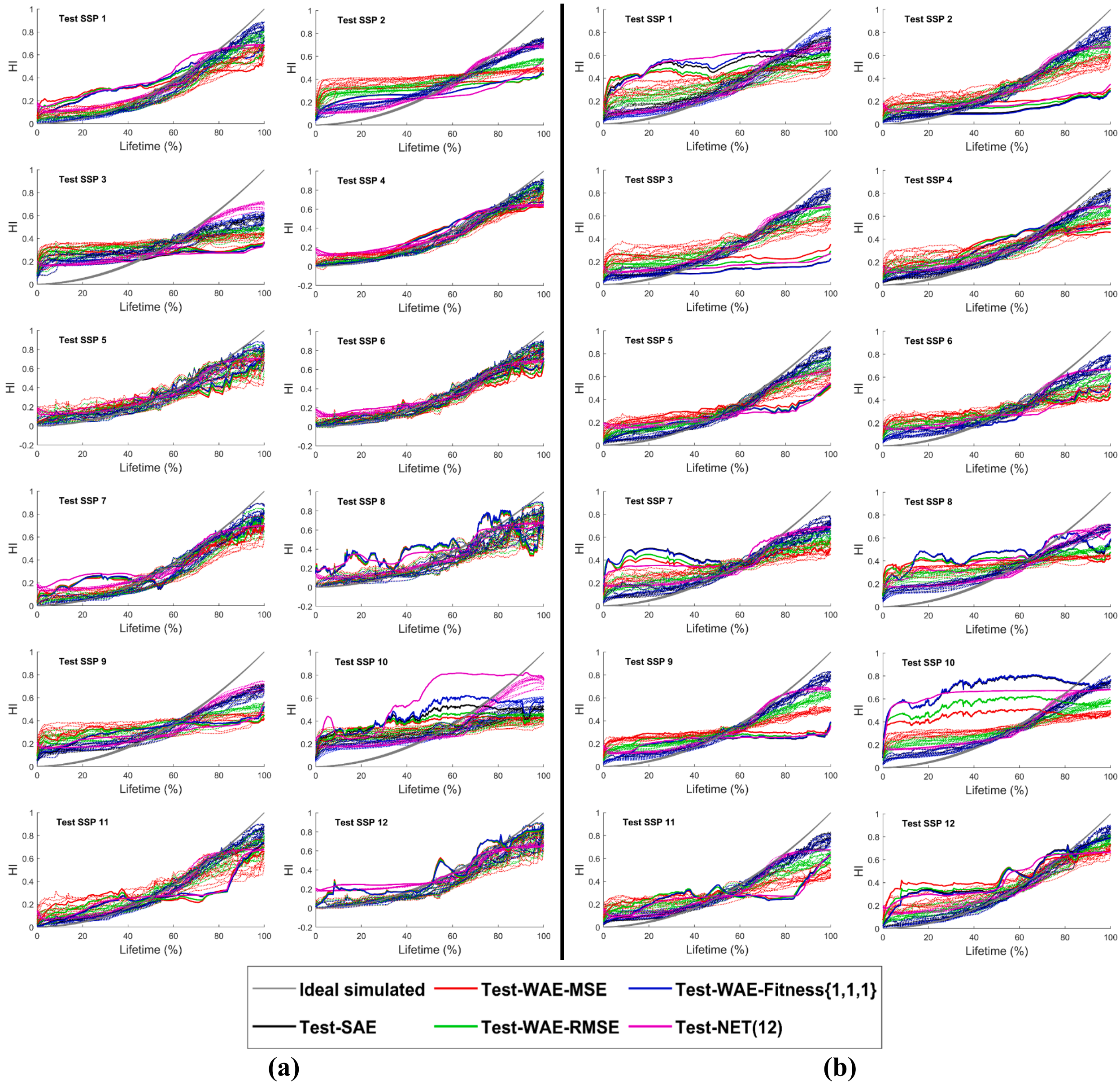

Fig. 10. (a) The HIs constructed by the base Model 3, considering Case B for the dataset division (the test SSP itself as validation), with best average Fitness 2.74 (±0.33) based on Eq. (16) for ensemble Net(12); (b) The HIs constructed by the base Model 9, considering Case C for the dataset division (another SSP other than the test SSP as validation), with best average Fitness 2.59 (±0.24) based on Eq. (16) for ensemble Net(12). Dotted lines are related to the training units.

respectively. Using strain data and genetic programming, Galanopoulos et al. [21] developed a HI with Mo and Pr of 0.89 and 0.95, respectively. Tr is not reported in either of these papers. In Yue et al. [13], stiffness was estimated through guided waves and used as a HI. However, only average Mo and Pr were achieved with values of 0.66 and 0.84, respectively. Also, they did not consider the MMK Mo which is more rigorous. Comparing with the previous references, the SS-DNN made of LSTM layers in the first step and the EL model made of BiLSTM layers in the second step, proposed in this paper, provide a HI with average Mo and Pr (including training set, over 12 Folds) of 0.968 and 0.925, respectively, for the best case and 0.843 and 0.879 for the worst case. Thus, for the HI construction of SSP under C-C fatigue loading, the high performance and generalizability of the developed approach are both confirmed.

5. Conclusions

In this paper, a methodology for creating a reliable HI for the remaining useful life estimation of composite structures is presented. Single-stiffened composite panels with disbond/impact damage are subjected to C–C fatigue loading, and acoustic emission (AE) is used to monitor degradation.

The complete ensemble empirical mode decomposition with adaptive noise (CEEMDAN) is first applied to low-level AE features in order to extract information with better monotonicity, prognosability, and trendability. Statistical quantities are calculated from the first four intrinsic mode functions, and a total of 504 high-level features are used as input to a semi-supervised deep learning framework based on a multilayer LSTM network (referred to as the semi-supervised base model) triggered by simulated HIs due to the lack of labels. The leave-one-out cross-validation scheme is implemented to evaluate the generalized performance, and a total of 10 different model configurations are investigated and compared.

To further improve the developed HIs, an ensemble learning (EL) step is implemented. Different EL models are investigated, including simple and weighted averaging as well as semi-supervised deep learning networks. The EL models, particularly the one composed of BiLSTM units, achieved over 20% increased performance compared to the semi-supervised base learner models, which had a performance of 71% (2.13/3.00), as demonstrated by the higher fitness values.

To evaluate the suitability of the developed HIs, the prognostic metrics are refined to account for the performance of only the test unit instead of the entirety of the available units. The new evaluation metrics indicated that the Fitness results were slightly lower (~5%) than the complete set's performance of 91.3% (2.74/3.00). Yet, the high values (2.59/3.00) demonstrated the framework's effectiveness in creating suitable HIs.

The main limitation of the proposed methodology is its interpretability, which lies in its lack of actual degradation information over fatigue loading regarding the composite structure. A significant improvement would include physics-informed labels, e.g., knowledge of the damage size and type, to provide a more interpretable HI. The next step of this research involves employing the developed HIs in capable prognostic models for RUL estimation of composite structures. Future research can take into account retraining the models and applying the suggested approach to multi-stiffener panels through transfer learning.

CRediT authorship contribution statement

Morteza Moradi: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Resources, Methodology, Investigation, Formal analysis, Conceptualization. **Georgios Galanopoulos:** Writing – review & editing, Writing – original draft, Validation, Software, Investigation. **Thyme Kuiters:** Writing – review & editing, Software. **Dimitrios Zarouchas:** Writing – review & editing, Supervision, Conceptualization.

Funding

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No 859957 "ENHAnCE, European training Network in intelligent prognostics and Health mAnagement in Composite structurEs".

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.

Data availability

The data is publicly available.

References

[1] V. Giurgiutiu, Structural health monitoring (SHM) of aerospace composites, Polym. Compos. Aerosp. Ind. (2020) 491–558, https://doi.org/10.1016/B978-0-08-102679-3.00017-4.
[2] Y. He, et al., An overview of acoustic emission inspection and monitoring technology in the key components of renewable energy systems, Mech. Syst. Signal Process 148 (2021) 107146, https://doi.org/10.1016/J.YMSSP.2020.107146.
[3] J. Contreras Lopez, J. Chiachío, A. Saleh, M. Chiachío, A. Kolios, A cross-sectoral review of the current and potential maintenance strategies for composite structures, SN Appl. Sci. 4 (6) (2022) 180, https://doi.org/10.1007/s42452-022-05063-3.
[4] J.J. Montero Jimenez, S. Schwartz, R. Vingerhoeds, B. Grabot, M. Salaün, Towards multi-model approaches to predictive maintenance: A systematic literature survey on diagnostics and prognostics, J. Manuf. Syst. 56 (2020) 539–557, https://doi.org/10.1016/J.JMSY.2020.07.008.
[5] J.Z. Sikorska, M. Hodkiewicz, L. Ma, Prognostic modelling options for remaining useful life estimation by industry, Mech. Syst. Signal Process 25 (5) (2011) 1803–1836, https://doi.org/10.1016/j.ymssp.2010.11.018.
[6] T. Li, J. Chen, S. Yuan, F. Cadini, C. Sbarufatti, Particle filter-based damage prognosis using online feature fusion and selection, Mech. Syst. Signal Process 203 (2023) 110713, https://doi.org/10.1016/J.YMSSP.2023.110713.
[7] K. Javed, R. Gouriveau, N. Zerhouni, P. Nectoux, Enabling health monitoring approach based on vibration data for accurate prognostics, IEEE Trans. Ind. Electron. 62 (1) (2014) 647–656.
[8] A. Behnia, H.K. Chai, A.A. Mousa, S.A. Ravanfar, A novel damage index for online monitoring of RC slabs under monotonic loading by integration of process controlling into acoustic emission technique, Mech. Syst. Signal Process 119 (2019) 547–560, https://doi.org/10.1016/J.YMSSP.2018.09.030.
[9] L. Guo, L. Naipeng, J. Feng, L. Yaguo, L. Jing, A recurrent neural network based health indicator for remaining useful life prediction of bearings, *Neurocomputing* 240 (2017) 98–109, https://doi.org/10.1016/j.neucom.2017.02.045.
[10] M. Moradi, J. Chiachío, D. Zarouchas, "Developing health indicators for composite structures based on a two-stage semi-supervised machine learning model using acoustic emission data," 2023. Accessed: Sep. 15, 2023. [Online]. Available: https://research.tudelft.nl/en/publications/developing-health-indicators-for-composite-structures-based-on-a-.

[11] T. Loutas, N. Eleftheroglou, D. Zarouchas, A data-driven probabilistic framework towards the in-situ prognostics of fatigue life of composites based on acoustic emission data, Compos. Struct. 161 (2017) 522–529, https://doi.org/10.1016/j.compstruct.2016.10.109.
[12] D.P. Milanoski, T.H. Loutas, Strain-based health indicators for the structural health monitoring of stiffened composite panels, J. Intell. Mater. Syst. Struct. 32 (3) (2020) 1045389X20924822., https://doi.org/10.1177/1045389X20924822.
[13] N. Yue, A. Broer, W. Briand, M. Rébillat, T. Loutas, D. Zarouchas, Assessing stiffness degradation of stiffened composite panels in post-buckling compression-compression fatigue using guided waves, Compos. Struct. 293 (2022) 115751, https://doi.org/10.1016/J.COMPSTRUCT.2022.115751.
[14] J. Sun, H. Zuo, W. Wang, M.G. Pecht, Prognostics uncertainty reduction by fusing on-line monitoring data based on a state-space-based degradation model, Mech. Syst. Signal Process. 45 (2) (2014) 396–407, https://doi.org/10.1016/J.YMSSP.2013.08.022.
[15] J. Chen, S. Yuan, H. Wang, On-line updating Gaussian process measurement model for crack prognosis using the particle filter, Mech. Syst. Signal Process. 140 (2020) 106646, https://doi.org/10.1016/J.YMSSP.2020.106646.
[16] J. Chiachío, M. Chiachío, S. Sankararaman, A. Saxena, K. Goebel, Condition-based prediction of time-dependent reliability in composites, Reliab. Eng. Syst. Saf. 142 (2015) 134–147, https://doi.org/10.1016/J.RESS.2015.04.018.
[17] P. Loukopoulos, et al., Reciprocating compressor prognostics of an instantaneous failure mode utilising temperature only measurements, Appl. Acoust. 147 (2017) 77–86, https://doi.org/10.1016/j.apacoust.2017.12.003.
[18] Y. Qin, S. Xiang, Y. Chai, H. Chen, Macroscopic–microscopic attention in LSTM networks based on fusion features for gear remaining life prediction, IEEE Trans. Ind. Electron. 67 (12) (2019) 10865–10875.
[19] T. Song, C. Liu, R. Wu, Y. Jin, D. Jiang, A hierarchical scheme for remaining useful life prediction with long short-term memory networks, Neurocomputing 487 (2022) 22–33, https://doi.org/10.1016/j.neucom.2022.02.032.
[20] K.T.P. Nguyen, K. Medjaher, An automated health indicator construction methodology for prognostics based on multi-criteria optimization, ISA Trans. 113 (2021) 81–96, https://doi.org/10.1016/j.isatra.2020.03.017.
[21] G. Galanopoulos, N. Eleftheroglou, D. Milanoski, A. Broer, D. Zarouchas, T. Loutas, A novel strain-based health indicator for the remaining useful life estimation of degrading composite structures, Compos. Struct. 306 (2023) 116579, https://doi.org/10.1016/J.COMPSTRUCT.2022.116579.
[22] C. Song, K. Liu, X. Zhang, Integration of Data-Level Fusion Model and Kernel Methods for Degradation Modeling and Prognostic Analysis, IEEE Trans. Reliab. 67 (2) (2018) 640–650, https://doi.org/10.1109/TR.2017.2715180.
[23] M. Moradi, A. Broer, J. Chiachío, R. Benedictus, D. Zarouchas, Intelligent health indicators based on semi-supervised learning utilizing acoustic emission data, Lecture Notes Civ. Eng. vol. 270 LNCE (2023) 419–428, https://doi.org/10.1007/978-3-031-07322-9_43.
[24] M. Moradi, P. Komninos, R. Benedictus, D. Zarouchas, Interpretable neural network with limited weights for constructing simple and explainable HI using SHM data, in: Annual Conference of the PHM Society, vol. 14, no. 1. PHM Society, 2022.
[25] Y. Li, X. Huang, T. Gao, C. Zhao, S. Li, A wiener-based remaining useful life prediction method with multiple degradation patterns, Adv. Eng. Inf. 57 (2023) 102066, https://doi.org/10.1016/J.AEI.2023.102066.
[26] L. Guo, Y. Lei, N. Li, T. Yan, N. Li, Machinery health indicator construction based on convolutional neural networks considering trend burr, Neurocomputing 292 (2018) 142–150, https://doi.org/10.1016/J.NEUCOM.2018.02.083.
[27] J.B. Coble, “Merging Data Sources to Predict Remaining Useful Life – An Automated Method to Identify Prognostic Parameters,” *Doctoral Dissertations*, May 2010, Accessed: Sep. 12, 2023. [Online]. Available: https://trace.tennessee.edu/utk_graddiss/683.
[28] F.L. Zhang, Y.P. Yang, H.B. Xiong, J.H. Yang, Z. Yu, Structural health monitoring of a 250-m super-tall building and operational modal analysis using the fast Bayesian FFT method, Struct. Control Health Monit. 26 (8) (2019) e2383.
[29] G. Ruocci, G. Cumunel, T. Le, P. Argoul, N. Point, L. Dieng, Damage assessment of pre-stressed structures: A SVD-based approach to deal with time-varying loading, Mech. Syst. Signal Process. 47 (1–2) (2014) 50–65, https://doi.org/10.1016/J.YMSSP.2013.09.011.
[30] R. Yan, X. Chen, S.C. Mukhopadhyay, Advanced signal processing for structural health monitoring, Smart Sensors, Measur. Instrumen. 26 (2017) 1–11, https://doi.org/10.1007/978-3-319-56126-4_1/FIGURES/5.
[31] A. Khan, D.K. Ko, S.C. Lim, H.S. Kim, Structural vibration-based classification and prediction of delamination in smart composite laminates using deep learning neural network, Compos. B Eng. 161 (2019) 586–594, https://doi.org/10.1016/J.COMPOSITESB.2018.12.118.
[32] N.E. Huang, et al., The empirical mode decomposition and the Hubert spectrum for nonlinear and non-stationary time series analysis, Proc. R. Soc. A: Math. Phys. Eng. Sci. 454 (1971) (1998) 903–995, https://doi.org/10.1098/RSPA.1998.0193.
[33] N.E. Huang, S.R. Long, Z. Shen, The mechanism for frequency downshift in nonlinear wave evolution, Adv. Appl. Mech. 32 (1996).
[34] S.T. Quek, P.S. Tua, Q. Wang, Detecting anomalies in beams and plate based on the Hilbert-Huang transform of real signals, 2003.
[35] Y. Shen, Y. Zhang, Z. Wang, Satellite fault diagnosis method based on predictive filter and empirical mode decomposition, J. Syst. Eng. Electron. 22 (1) (2011) 83–87, https://doi.org/10.3969/j.issn.1004-4132.2011.01.010.
[36] K. Dragomiretskiy, D. Zosso, Variational Mode Decomposition, IEEE Trans. Signal Process. 62 (3) (2014) 531–544, https://doi.org/10.1109/TSP.2013.2288675.
[37] Y. Li, F. Xu, Acoustic emission sources localization of laser cladding metallic panels using improved fruit fly optimization algorithm-based independent variational mode decomposition, Mech. Syst. Signal Process. 166 (2022) 108514, https://doi.org/10.1016/j.ymssp.2021.108514.
[38] Z. Wu, N.E. Huang, Ensemble empirical mode decomposition: a noise-assisted data analysis method, Adv. Adapt. Data Anal. 1 (1) (2011) 1–41, https://doi.org/10.1142/S1793536909000047.
[39] H. Sarmadi, A. Entezami, M. Daneshvar Khorram, Energy-based damage localization under ambient vibration and non-stationary signals by ensemble empirical mode decomposition and Mahalanobis-squared distance, JVC/J. Vib. Control 26 (11–12) (2020) 1012–1027, https://doi.org/10.1177/1077546319891306/ASSET/IMAGES/LARGE/10.1177_1077546319891306-FIG10.JPEG.
[40] G.G. Amiri, E. Darvishan, Damage detection of moment frames using ensemble empirical mode decomposition and clustering techniques, KSCE J. Civ. Eng. 19 (5) (2015) 1302–1311, https://doi.org/10.1007/s12205-015-0415-z.
[41] M.E. Torres, M.A. Colominas, G. Schlotthauer, P. Flandrin, “A COMPLETE ENSEMBLE EMPIRICAL MODE DECOMPOSITION WITH ADAPTIVE NOISE”, Accessed: Jul. 24, 2023. [Online]. Available: http://www.physionet.org/cgi-bin/ATM.
[42] H.S. Kim, D.W. Park, S.Il Kim, G.H. Oh, H.S. Kim, Non-destructive detection of thin micro-defects in glass reinforced polymer composites using a terahertz electro–magnetic wave based on a convolution neural network, Compos. B Eng. 257 (2023) 110694, https://doi.org/10.1016/J.COMPOSITESB.2023.110694.
[43] M.O.K. Mendonça, I.F. Apolinário, P.S.R. Diniz, Introduction to signal processing and machine learning theory, Signal Process. Mach. Learn. Theory (2024) 1–34, https://doi.org/10.1016/B978-0-32-391772-8.00007-7.
[44] A.A. Mousavi, C. Zhang, S.F. Masri, G. Gholipour, Structural Damage Localization and Quantification Based on a CEEMDAN Hilbert Transform Neural Network Approach: A Model Steel Truss Bridge Case Study, Sensors 20 (5) (2020) 1271, https://doi.org/10.3390/S20051271.
[45] A.A. Mousavi, C. Zhang, S.F. Masri, G. Gholipour, Structural damage detection method based on the complete ensemble empirical mode decomposition with adaptive noise: a model steel truss bridge case study, Struct. Health Monit. 21 (3) (2022) 887–912, https://doi.org/10.1177/14759217211013535/ASSET/IMAGES/LARGE/10.1177_14759217211013535-FIG20.JPEG.
[46] M. Moradi, A. Broer, J. Chiachío, R. Benedictus, T.H. Loutas, D. Zarouchas, Intelligent health indicator construction for prognostics of composite structures utilizing a semi-supervised deep neural network and SHM data, Eng. Appl. Artif. Intell. 117 (2023) 105502, https://doi.org/10.1016/J.ENGAPPAI.2022.105502.
[47] A. Broer, G. Galanopoulos, R. Benedictus, T. Loutas, D. Zarouchas, Fusion-based damage diagnostics for stiffened composite panels, Struct. Health Monit. 21 (2) (2022) 613–639, https://doi.org/10.1177/14759217211007127/ASSET/IMAGES/LARGE/10.1177_14759217211007127-FIG17.JPEG.
[48] G. Galanopoulos, D. Milanoski, A. Broer, D. Zarouchas, T. Loutas, Health Monitoring of Aerospace Structures Utilizing Novel Health Indicators Extracted from Complex Strain and Acoustic Emission Data, Sensors 21 (17) (2021) 5701, https://doi.org/10.3390/S21175701.
[49] D. Zarouchas, A. Broer, G. Galanopoulos, W. Briand, R. Benedictus, T. Loutas, Compression Compression fatigue tests on single stiffener aerospace structures, 2021. https://doi.org/10.34894/QNURER.

[50] D. Zarouchas, A. Broer, G. Galanopoulos, W. Briand, R. Benedictus, T. Loutas, Compression Compression fatigue tests on single stiffener aerospace structures, *DataverseNL*. https://doi.org/10.34894/QNURER.
[51] M. Moradi, A. Broer, D. Zarouchas, Acoustic emission dataset of single-stiffener composite panels subjected to impact and run-to-failure fatigue loading, vol. 2, 2022, https://doi.org/10.17632/YS8R8M7BX2.2.
[52] M. Ge, Analysis of source location algorithMS Part II: Iterative methods, J. Acoustic Emission 21 (2003) 29.
[53] N. Eleftheroglou, D. Zarouchas, T. Loutas, R. Alderliesten, R. Benedictus, Structural health monitoring data fusion for in-situ life prognosis of composite structures, Reliab. Eng. Syst. Saf. 178 (2018) 40–54, https://doi.org/10.1016/J.RESS.2018.04.031.
[54] J. Coble, J.W. Hines, Identifying Optimal Prognostic Parameters from Data: A Genetic Algorithms Approach, *Annual Conference of the PHM Society*, vol. 1, no. 1, 2009, Accessed: Sep. 16, 2023. [Online]. Available: http://www.papers.phmsociety.org/index.php/phmconf/article/view/1404.
[55] M. Moradi, Designing health indicators for aerospace structures by intelligent information fusion, TU Delft (2024), https://doi.org/10.4233/uuid:7ac03701-b97a-427d-990c-e6c696d1254b.
[56] M. Moradi, P. Komninos, D. Zarouchas, Constructing explainable health indicators for aircraft engines by developing an interpretable neural network with discretized weights, Appl. Intell. (2024).
[57] J. Snoek, H. Larochelle, R.P. Adams, Practical Bayesian Optimization of Machine Learning Algorithms, Adv. Neural Inf. Process Syst. 25 (2012).
[58] D. Opitz, R. Maclin, Popular ensemble methods: An empirical study, J. Artif. Intell. Res. 11 (1999) 169–198, https://doi.org/10.1613/JAIR.614.
[59] R. Polikar, Ensemble based systems in decision making, IEEE Circuits Syst. Mag. 6 (3) (2006) 21–44, https://doi.org/10.1109/MCAS.2006.1688199.
[60] L. Breiman, Bagging predictors, Mach. Learn. 24 (2) (1996) 123–140, https://doi.org/10.1007/BF00058655/METRICS.
[61] J. Friedman, T. Hastie, R. Tibshirani, Additive logistic regression: a statistical view of boosting (With discussion and a rejoinder by the authors), Ann. Statist. 28 (2) (2000) 337–407, https://doi.org/10.1214/AOS/1016218223.
[62] D.H. Wolpert, Stacked generalization, Neural Netw. 5 (2) (1992) 241–259, https://doi.org/10.1016/S0893-6080(05)80023-1.
[63] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32, https://doi.org/10.1023/A:1010933404324/METRICS.
[64] Y. Freund, R.E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, Lecture Notes in Computer Science (including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 904 (1995) 23–37, https://doi.org/10.1007/3-540-59119-2_166/COVER.
[65] T. Chen, C. Guestrin, XGBoost: A Scalable Tree Boosting System, *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, https://doi.org/10.1145/2939672.
[66] S. Haykin, *Neural networks: A comprehensive foundation*, Prentice Hall PTR (1998), https://doi.org/10.1016/0967-0661(95)90080-2.
[67] S. Hashem, Optimal linear combinations of neural networks, Neural Networks 10 (4) (1997) 599–614, https://doi.org/10.1016/S0893-6080(96)00098-6.
[68] T. Hastie, J. Friedman, R. Tibshirani, *The Elements of Statistical Learning*, vol. 45, no. 3. in Springer Series in Statistics, vol. 45. New York, NY: Springer New York, 2001. https://doi.org/10.1007/978-0-387-21606-5.
[69] M. Matsumoto, T. Nishimura, Mersenne twister, ACM Transactions on Modeling and Computer Simulation (TOMACS) 8 (1) (1998) 3–30, https://doi.org/10.1145/272991.272995.
[70] G. Galanopoulos, D. Milanoski, N. Eleftheroglou, A. Broer, D. Zarouchas, T. Loutas, Acoustic emission-based remaining useful life prognosis of aeronautical structures subjected to compressive fatigue loading, Eng. Struct. 290 (2023) 116391, https://doi.org/10.1016/J.ENGSTRUCT.2023.116391.