Sensor Fault Detection in Smart Extraterrestrial Habitats Using Unsupervised Learning

Zixin Wang * and Mohammad R. Jahanshahi †
Purdue University, West Lafayette, IN 47907, USA.

Mohsen Azimi ‡
Mississippi State University, Starkville, MS 39762, USA.

Shirley J. Dyke §
Purdue University, West Lafayette, IN 47907, USA.

Various types of sensors are needed to monitor the health state of smart deep-space habitats. However, measured data can be affected by sensor faults, which influence the health management system (HMS) and consequently the decision-making. In this paper, an unsupervised learning approach based on convolutional autoencoders (CAEs) is developed to detect anomalies in temperature and pressure sensors. The proposed method is systematically investigated using a habitat simulator (HabSim). Several illustrative examples are demonstrated in nominal and hazardous states of the habitat, including micrometeorite impact and fire scenarios. The performance of the proposed method using CAEs is compared with that of existing methods using auto-associative neural networks (AANNs) and variational autoencoders (VAEs). This comparison is based on typical evaluation metrics, including precision, recall, F1 score, training time, and testing time. The effect of temperature-pressure coupling on the detection performance of CAEs and AANNs is explored by training different data-driven models, including one with temperature sensors, one with pressure sensors, and one with both temperature and pressure sensors. The effect of the number of faulty sensors on the performance of CAEs is studied as with an increase in the number of faulty sensors, redundant information among the sensors is reduced. The capability of CAEs to change the number of sensors without re-designing the network architecture and re-training the neural network is investigated and demonstrated. The capabilities and limitations of the proposed solution are discussed.

Nomenclature

*Ph.D. Student, Lyles School of Civil Engineering; wang4591@purdue.edu.
†Associate Professor, Lyles School of Civil Engineering, Elmore Family School of Electrical and Computer Engineering; jahansha@purdue.edu (Corresponding Author).
‡Assistant Professor, Department of Mechanical Engineering; azimi@me.msstate.edu.
§Professor, School of Mechanical Engineering, Lyles School of Civil Engineering; sdyke@purdue.edu.

a	=	row index in the kernel
b	=	column index in the kernel
$\boldsymbol{b}$	=	trainable parameters (bias)
$c_0, c_1, ..., c_r$	=	thermocouple-type-dependent polynomial coefficients
C_0	=	fixed bias constant
d	=	stride size of the sliding window
E_P	=	output voltage of the pressure sensor corresponding to the actual pressure P
E_T	=	output voltage of the thermocouple corresponding to the actual temperature T
$f(\cdot)$	=	mapping function
f_x	=	measurement deviation caused by sensor faults
F	=	output of the convolutional layer
F_s	=	output size of the convolutional layer
H	=	length of the sliding window
I	=	input of the convolutional layer
I_s	=	input size of the convolutional layer
$g(\cdot)$	=	demapping function
K	=	convolution kernel
K_s	=	kernel size
L	=	length of the original signal
$\mathcal{L}(\cdot)$	=	loss function
N	=	number of segments obtained by slicing the original signal using a sliding window
p	=	row index in the output of the convolutional layer
P	=	actual pressure (Pa)
$\bar{P}$	=	measured pressure (Pa)
P_l	=	pressure lower limit (Pa)
P_u	=	pressure upper limit (Pa)
P_R	=	pressure range of the pressure sensor (Pa)
P_s	=	padding size
q	=	column index in the output of the convolutional layer
S	=	stride size of the kernel
t	=	current simulation time
t_0	=	initial simulation time

t_f = final simulation time

T = actual temperature (K)

$\bar{T}$ = measured temperature (K)

v_x = measurement noise

V_l = voltage lower limit (V)

V_u = voltage upper limit (V)

V_R = voltage range of the pressure sensor (V)

$\boldsymbol{W}$ = trainable parameters (weights)

x = actual value

$\bar{x}$ = measured value

$\boldsymbol{x}^{n\times m}$ = network input (measured data)

$\hat{\boldsymbol{x}}^{n\times m}$ = network output (reconstructed data)

$\boldsymbol{x}_{train}$ = observations in the training dataset

$\boldsymbol{z}^{n\times k}$ = latent information

$\sigma(\cdot)$ = activation function

τ = threshold for anomaly detection

$\Theta_j^{(i)}$ = predicted state for the j^{th} sensor and i^{th} observation

α = bias percentage of the faulty sensor

β = drift percentage of the faulty sensor

Subscripts

j = j^{th} sensor

r = order of polynomial for a thermocouple using the standard polynomial parameterization

Superscripts

i = i^{th} observation

k = latent dimension

m = number of variables or sensors

n = number of observations

I. Introduction

To monitor and manage the health state of a deep-space habitat, various types of sensors are necessary, such as accelerometers, temperature sensors, pressure sensors, etc. Recently, advanced sensing techniques have been implemented for the health monitoring of aerospace vehicles and structures, for instance, the real-time modal analysis of launch vehicles using fiber Bragg grating sensor arrays [1], the impact identification in sandwich structures using granular crystal sensors [2], and the flight state awareness of aerial vehicles based on multimodal sensing [3]. However, due to the harsh environment in extraterrestrial habitats, including radiation, dust, seismic excitation, extreme temperature, meteorite impact, etc., sensors may be affected or damaged by these disturbances. It should be noted that good-quality data is essential for the HMS to perform fault detection and diagnosis, and further make robust decisions.

Methods for sensor fault detection can be divided into two categories: model-based methods and data-driven methods. The model-based methods require manual construction of a mathematical model of the system, which is a time-consuming and knowledge-extensive process [4]. The predicted measurements can be calculated by mathematical models. The difference between the actual measurements and the predicted ones, which is called residual, can be utilized as the fault indexes to detect sensor anomalies [5]. For instance, Shang *et al.* [6] developed an extended Kalman filter to detect and isolate the sensor and actuator faults in an aircraft bleed air temperature control system. Yang *et al.* [7] built an air handling unit (AHU) process model to detect faults in the supply air temperature sensor of the AHU system. Dey *et al.* [8] constructed a battery cell model for the detection, isolation, and estimation of the temperature, voltage, and current sensor faults.

The data-driven methods, on the other hand, do not rely on physics-based models, since they can automatically encode system knowledge, and learn the intrinsic relations among variables and parameters through the measurement data. Data-driven techniques have been utilized in diverse safety-critical applications in the aerospace industry [9–13]. Data-driven approaches can be classified into supervised learning and unsupervised learning. Supervised learning requires access to data under both normal and faulty conditions to train the algorithm. Typical supervised learning methods include convolutional neural network (CNN) [14–18], stacked autoencoder [19], long short-term memory autoencoder [20], AdaBoost [21], random forest [22], wavelet neural network [5], and multi-layer perceptron [23]. However, it is difficult to acquire sufficient data under various faulty conditions of the system in the real world. Hence, unsupervised learning that only requires the data in the normal state has been widely applied in sensor fault detection. Unsupervised learning methods include probabilistic neural network [24], AANN [25], autoencoder (AE) [26–29], variational autoencoder (VAE) [30–35], long short-term memory network [36], principal component analysis [37–39], autoregressive integrated moving average model [40], Bayesian dynamic linear model [41], Bayesian estimation [42], and temporal forecasting driven approach [43]. Among various unsupervised learning methods, the neural network-based approach is more popularly used in sensor fault detection due to its high pattern recognition and self-learning capabilities, as well as fast inference time.

The National Aeronautics and Space Administration (NASA) has investigated sensor fault detection in aerospace applications for decades. The strategies for sensor validation or sensor data qualification that NASA developed can be divided into two categories: hardware redundancy check and analytical redundancy check. A hardware redundancy check compares the measurement data concurrently collected by a group of hardware redundant sensors that are of a homogeneous sensor type and co-located at a place to enable them to measure the same physical property at the same sensing location. A faulty sensor can be detected if the measurements collected by this sensor are dissimilar from the ones collected by other sensors within the redundancy group [44]. It should be noted that the hardware redundancy check has several limitations. In reality, it is difficult to install multiple sensors at the same location given the weight reduction requirements and geometric space restrictions in aerospace applications. Moreover, if there are only two redundant sensors, it is not possible to identify which sensor is faulty. Analytical redundancy check, on the other hand, estimates the sensor measurements by leveraging the relationships among a group of sensors, and the sensor fault can be detected by comparing the estimated and actual measurements of each sensor [45, 46]. With this approach, the redundant sensors can be disparately located and non-homogeneous. Additionally, the analytical redundancy check can detect subtle sensor failure modes. The analytical redundancy check can be implemented by the model-based methods (e.g., Kalman filters [47, 48]) and the data-driven methods (e.g., AANNs [49, 50] and SureSense Software [51, 52]). Fault tolerance, as an essential ability of safety-critical systems, can compensate for faults in such a way that they do not lead to system failures. A fault-tolerant sensor configuration can be obtained by applying hardware redundancy or analytical redundancy [53].

AANN, as an unsupervised data-driven method, was first proposed by Kramer for nonlinear principal component analysis [54]. Later, he developed AANNs for sensor fault detection [55]. Inspired by Kramer's work, several researchers also applied AANNs to the sensor validation for the health monitoring of engines [49, 50, 56–60], nuclear power plants [61], high flux isotope reactors [61], boiling water reactors [62], as well as heating, ventilation, and air conditioning (HVAC) systems [25], etc. If the measurement information is analytically redundant, the lost data caused by the faulty sensor can be accurately reconstructed by AANN using the information from the remaining normal sensors. However, existing studies based on fully connected AANNs use the measurement to predict the sensor data at each time step, which does not consider the temporal information of the signals. The coupling between the temperature and pressure data is not considered in the reported studies. Moreover, it should be noted that CNN, as one of the deep learning techniques, is widely utilized in computer vision, speech recognition, and natural language processing [63]. CNNs have the capability of extracting the features from the raw data automatically [64]. Since CNNs are designed to deal with data with multiple channels (e.g., RGB image), the correlation between the temperature and pressure data can be learned by CNNs using two separate channels for each of them. Additionally, there are no records pertinent to using fully convolutional networks, which have the potential to be adaptable to different numbers of sensors, for unsupervised sensor fault detection.

CAE has been utilized for fault detection in various applications, including fault detection of multivariate process [65], high-impedance fault detection [66], hydraulic solenoid valve fault detection [67], hyperspectral anomaly detection [68], visual inspection [69], abnormal event detection in time series[70–72] and in videos [73]. However, there have been limited studies applying CAE to sensor fault detection. Unlike the existing studies on fault detection using CAE, our proposed approach leverages the redundant information within a sensor system by utilizing multiple sensor signals from heterogeneous sensors as input. The trained CAE can then reconstruct the correct sensor data in the output, even when the corresponding input is faulty sensor data, thus enhancing the fault tolerance capability of a sensor system. Another contribution of this work is sensor fault detection in the presence of abnormal events such as micrometeorite impact and fire in space habitats. In this context, the proposed approach should be capable of distinguishing between sensor faults and abnormal events, which poses a more complex problem than simply detecting sensor faults in the nominal state of the habitat. To investigate the robustness of the approach subjected to abnormal events, we have compared the performance of CAE with AANN and VAE in terms of sensor fault detection in micrometeorite impact and fire scenarios. The CAE approach we developed has several unique features. For instance, we have implemented a fully convolutional CAE for sensor fault detection, enabling the CAE to adapt to varying numbers of sensors without re-training the neural network, which is conducive to the resilient design of space habitats. Additionally, the CAE architecture is designed to exploit the information coupling between heterogeneous sensors, such as temperature and pressure sensors.

In this study, a novel sensor fault detection method for extraterrestrial habitats using CAEs is developed. The proposed approach is employed to detect faults in temperature and pressure sensors. The proposed method is systematically investigated using HabSim. Several illustrative examples are presented: (1) the comparative study between AANNs, CAEs, and VAEs; (2) the effect of the temperature-pressure coupling on the performance of CAEs and AANNs; (3) the effect of the number of faulty sensors on the performance of CAEs and AANNs; and (4) the capability of CAEs to change the number of sensors without re-designing the network architecture and re-training the neural network.

II. Methodology

A. Auto-associative Neural Network

AANN is designed to perform nonlinear principal component analysis [54]. The ability of AANN to fit arbitrary nonlinear functions is achieved by incorporating nonlinear activation functions into hidden layers. AANN contains a mapping function f and a demapping function g, which are defined in Eqs. (1) and (2):

$$z = f(\boldsymbol{x}) = \sigma_f(\boldsymbol{W}_f \boldsymbol{x} + \boldsymbol{b}_f) \tag{1}$$

$$\hat{\boldsymbol{x}} = g(z) = \sigma_g(\boldsymbol{W}_g z + \boldsymbol{b}_g) \tag{2}$$

where $\boldsymbol{x}^{n\times m}$ is the network input; n is the number of observations; m is the number of variables, which is equivalent to the total number of temperature and pressure sensors. $\boldsymbol{z}^{n\times k}$ represents the latent information in a lower dimension

($m > k$) calculated by the mapping function f. The demapping function g restores the original dimensionality of the data using the latent information, and $\hat{\boldsymbol{x}}^{n\times m}$ is the reconstructed data calculated by the demapping function g. $\boldsymbol{W}$ and $\boldsymbol{b}$ are the trainable weights and biases of the network. $\sigma(\cdot)$ represents the activation function. In this study, the ReLU activation function is adopted to introduce nonlinearities to hidden layers. The Sigmoid activation function is utilized in the output layer such that the output values are between 0 and 1. The mapping and demapping functions in AANN can be implemented using five layers, including the input layer, mapping layer, bottleneck layer, demapping layer, and output layer, as shown in Fig. 1. To avoid exploding gradient problems, the input data is normalized to the range between 0 and 1 using Eq. (3):

$$\boldsymbol{x} = \frac{\boldsymbol{x} - \min\{\boldsymbol{x}_{train}\}}{\max\{\boldsymbol{x}_{train}\} - \min\{\boldsymbol{x}_{train}\}} \tag{3}$$

where $\boldsymbol{x}_{train}$ represents the observations in the training dataset which only contains the sensor data in the normal state of the sensor. Specifically, temperature and pressure data are normalized separately due to their different units.

The loss function used to train AANN is defined as the mean squared error (MSE) between the network input and the corresponding network output, as shown in Eq. (4):

$$\mathcal{L}(\boldsymbol{x}, \hat{\boldsymbol{x}}) = \frac{1}{m}\sum_{i=1}^{n}\sum_{j=1}^{m}\left(\boldsymbol{x}_j^{(i)} - \hat{\boldsymbol{x}}_j^{(i)}\right)^2 \tag{4}$$

where i and j represent the ith observation and the jth sensor, respectively. Therefore, AANN is trained to reconstruct the sensor data in the normal state of the sensors. For a redundant sensor set, one or more lost sensor data could be reconstructed using the correlation between the faulty sensors and normal sensors. Since the bottleneck layer can extract the dominant features from the input data, it captures the features in the normal state and filters out the features caused by the sensor fault. Therefore, a well-trained AANN can estimate the sensor data in the normal state even though the corresponding input has been distorted by the sensor fault. This characteristic allows AANN to detect a sensor fault by quantifying the MSE between the measured data and the corresponding reconstructed data output by AANN. If MSE is larger than a predefined threshold τ, then a sensor fault has likely taken place, which can be expressed by Eq. (5):

$$\Theta_j^{(i)} = \begin{cases} 1, & \text{if } \mathrm{MSE}\left(\boldsymbol{x}_j^{(i)}, \hat{\boldsymbol{x}}_j^{(i)}\right) > \tau, \\ 0, & \text{otherwise}, \end{cases} \tag{5}$$

where 1 and 0 denote the abnormal and normal states of the sensor, respectively.

B. Convolutional Autoencoder

CNNs specialize in processing data that has a known grid-like topology [74]. Autoencoder is a type of unsupervised neural network that is utilized for data compression, dimensionality reduction, and feature extraction [75]. If autoencoders

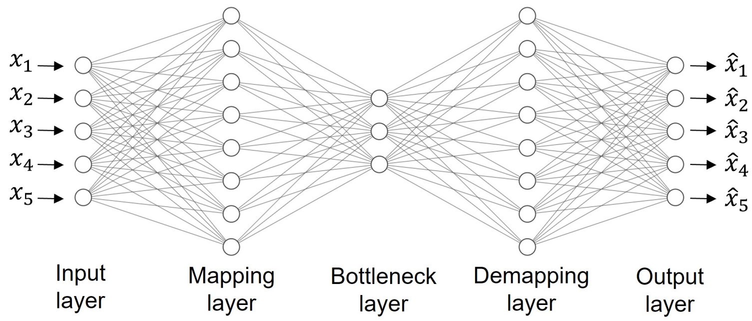


Fig. 1 Schematic diagram of AANN architecture.

use convolution operation in at least one of their layers, they are referred to as CAEs. In this study, fully convolutional layers are utilized in CAEs, as shown in Fig. 2, which is different from AANNs that contain fully connected layers. Convolution, also known as cross-correlation, is an operation on two functions (i.e., the input and kernel). Suppose the input and kernel are two-dimensional, the output of the convolutional layer, which is referred to as the feature map, can be obtained by Eq. (6):

$$F(p,q) = (K * I)(p,q) = \sum_a \sum_b I(p+a, q+b)K(a,b) \tag{6}$$

where I and F are the input and output of the convolutional layer, respectively. K is the kernel that slides across the input to extract features. a and b are the row and column indices in the kernel K. p and q are the row and column indices in the output F. Convolution provides a means for working with inputs of variable size [74]. The weight-sharing and local receptive field strategies in CNN make the model less prone to overfitting [76]. The size of the output of the convolutional layer can be determined by the size of the input, kernel, padding, and stride using Eq. (7):

$$F_s = \frac{I_s - K_s + 2P_s}{S_s} + 1 \tag{7}$$

where F_s, I_s, K_s, P_s, S_s denote the size of the output, input, kernel, padding, and stride in the corresponding dimension (i.e., height or width), respectively. CAE consists of an encoder and a decoder. The encoder compresses the input data to extract high-level features. The decoder reconstructs the input data using the latent representation generated by the encoder. As shown in Fig. 2, the height and width of the feature map decrease in the encoder, while the depth increases. To this end, the size of the kernel and stride decreases in the encoder, while the number of kernels increases. The decoder has a symmetric architecture of the encoder.

To mitigate internal covariance shift and enable faster training of deep neural networks, batch normalization through re-centering and re-scaling operations [77] is implemented in CAE. Moreover, the leaky ReLU activation function is employed to perform the nonlinear transformation, and the Sigmoid activation function is used to constrain the output of CAE to the range between 0 and 1. Similar to AANN, the input of CAE is the measured data and the output is the reconstructed data. The loss function used to train CAE is the MSE between the input and output of the network. A faulty sensor can be detected if the MSE between the measured data and the reconstructed data is larger than a predefined threshold τ (Fig. 2). Since CNNs are good at dealing with multi-channel data, such as RGB images, the coupling between the temperature and pressure can be captured by setting them into two separate channels in CAE. Each channel consists of a two-dimensional matrix with a size of $H \times W$, where W refers to the number of temperature or pressure sensors in the habitat, and H refers to the length of the segmented signal corresponding to each sensor. Therefore, by combining temperature and pressure channels, the size of the input and output of CAE is $H \times W \times 2$. To prepare the data used to train and test CAE, the sensor signal is divided into multiple segments using a sliding window mechanism. Specifically, the length of the window is H, which is set to 100 or 200, and the distance the window moves at each step (i.e., stride) is d, which is set to 50. The number of segments N obtained by slicing the original signal using a sliding window can be calculated as:

$$N = \frac{L - H}{d} + 1 \tag{8}$$

where L is the length of the original signal.

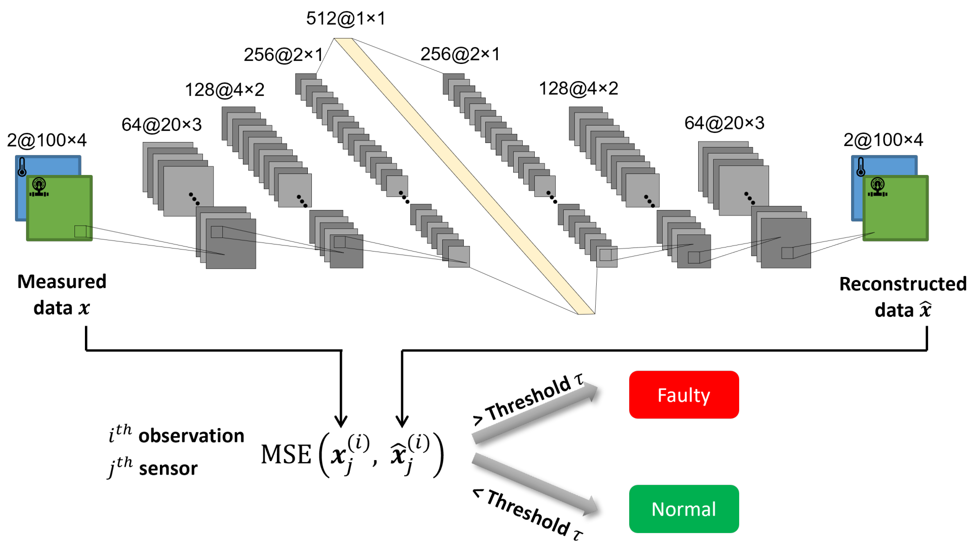


Fig. 2 Schematic diagram of CAE architecture. The layer labels represent 'the number of channels @ the size of each channel.'

III. Data Collection Using HabSim

A. Habitat Scenarios

HabSim is a plug-and-play environment developed in MATLAB Simulink to simulate a space habitat. It consists of five physics-based subsystems, including the structural protective layer (SPL), structure (ST), interior environment (IE), environmental control and life support system (ECLSS), and power system (PW). The layout of the one-fifth scale reference habitat concept is shown in Fig. 3 The IE model represents a multi-zonal lunar habitat. Specifically, the interior habitat is divided into two zones by a pocket door. Normally, the pocket door is open for crew access. It can be closed for safety purposes to isolate a zone that has been damaged as a result of a disruption scenario (e.g., a micrometeorite impact and fire) from another healthy zone. To reduce computational cost and conduct real-time simulation, a quasi-static zero-dimensional lumped model approach has been developed to predict temperature and pressure within each zone. In this study, four temperature sensors and four pressure sensors are placed in Zone #1, and they measure the temperature and pressure at a single node. Both micrometeorite impact and fire scenarios are assumed to occur in Zone #1, and after the disruption scenario happens, the pocket door is closed to isolate Zone #1 from Zone #2.

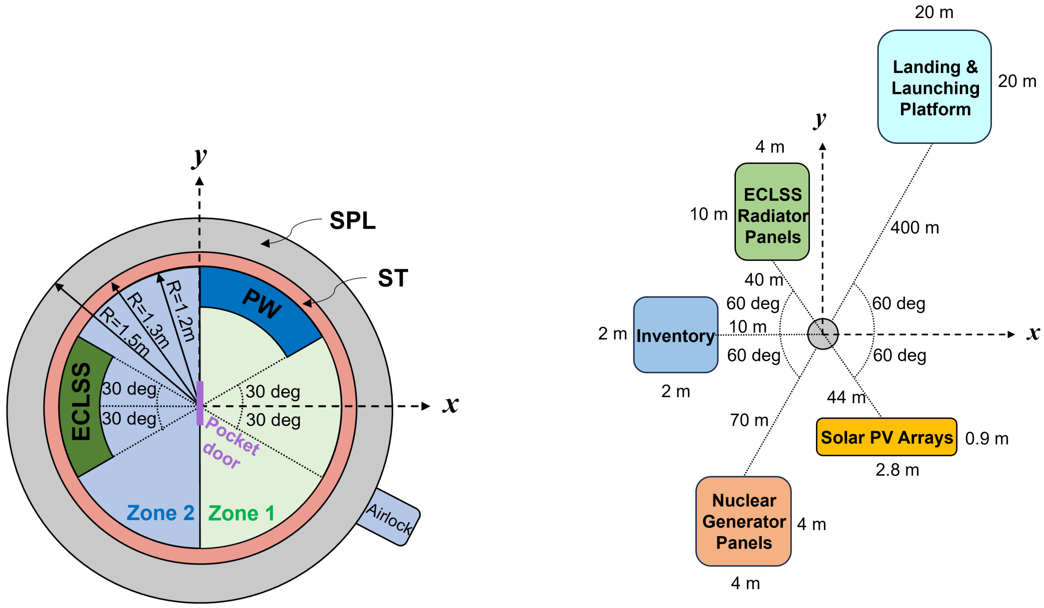
HabSim can simulate a space habitat in the nominal state, as well as hazardous states caused by physics-based disruption scenarios (e.g., micrometeorite impact) and phenomenological disruption scenarios (e.g., fire). As shown in Fig. 4, all the subsystems of HabSim are interconnected, including disturbance block (DB), agent (AG), SPL, ST, IE, ECLSS, PW, communication network (CN), command and control (C2), data repository and data service (DRDS), and human computer interface (HCI). The interdependencies between the subsystems can be categorized into four types: physical interdependencies, cyber interdependencies, intervention interdependencies, and disruption interdependencies. To simulate disruption scenarios, the DB sends physics-based signals (e.g., micrometeorite mass and velocity) and phenomenological signals (e.g., disruption intensity levels) to physics-based subsystems to model the effect of the disruptive hazard events on the subsystem components. The sensor signals utilized in this study are the cyber interdependencies between the IE and CN. In this study, we investigate the fault detection of temperature and pressure sensors in the IE by analyzing the data generated by HabSim in both the nominal state and hazardous states caused by micrometeorite impact and fire scenarios.

B. Sensor Models

Temperature and pressure sensor models are built in HabSim to collect realistic temperature and pressure measurements. Regarding the temperature sensor, the type T thermocouple with a measurement range of 3.15 to 643.15 K is selected. Thermocouples can be modeled using the standard polynomial parameterization defined in the NIST ITS-90 thermocouple database [79]. The output voltage E_T of the thermocouple corresponding to the actual temperature T is obtained by:

$$E_T = c_0 + \sum_{i=1}^{r} c_i (T - 273.15)^i \tag{9}$$

where $c_0, c_1, ..., c_r$ are thermocouple-type-dependent polynomial coefficients, T is the temperature to be measured (K), and r is the order of the polynomial. A saturation block is implemented to ensure the output voltage is in the range of 0 to 10 V. The quantizer block is utilized to simulate an A/D converter. To convert the output voltage E_T to the measured



(a) Interior habitat layout

(b) Exterior habitat layout

Fig. 3 The layout of the one-fifth scale reference habitat concept [78].

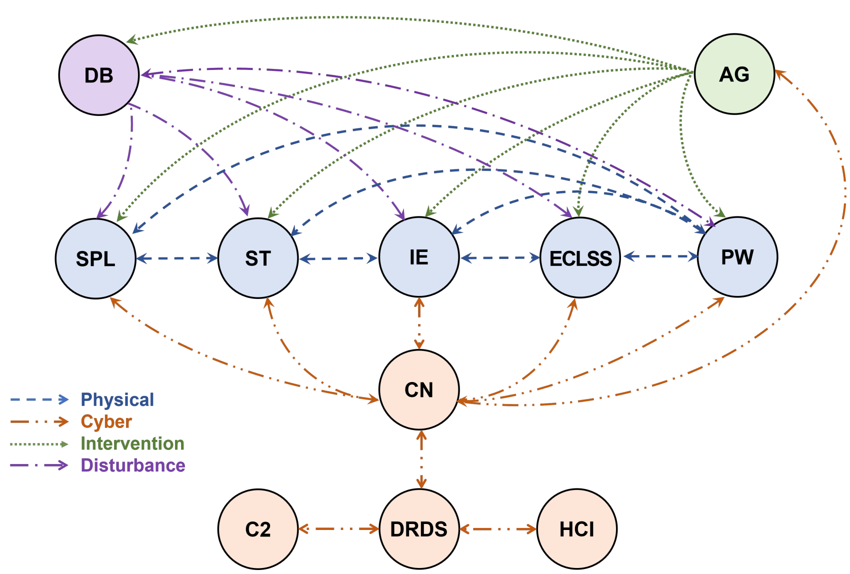


Fig. 4 Schematic diagram of interactions between different subsystems of HabSim[78].

temperature $\bar{T}$, a table search method is employed.

Regarding the pressure sensor, the absolute pressure sensor with an output voltage range of 0 to 10 V is selected, and its measurement range is 0 to 344,738 Pa. The output voltage is linearly proportional to the pressure. Therefore, the output voltage E_P of the pressure sensor corresponding to the actual pressure P can be obtained by:

$$E_P = V_R \times (P - P_l)/(P_u - P_l) \tag{10}$$

where V_R is the voltage range (V), P is the pressure to be measured (Pa), P_l is the pressure lower limit (Pa), and P_u is the pressure upper limit (Pa). The output voltage E_P can be converted to the measured pressure $\bar{P}$ using the following formula:

$$\bar{P} = P_R \times (E_P - V_l)/(V_u - V_l) \tag{11}$$

where P_R is the pressure range (Pa), V_l is the voltage lower limit (V), and V_u is the voltage upper limit (V). The saturation and quantizer blocks are also implemented in the pressure sensor model.

Sensor faults and measurement noise are added to the measured temperature and pressure which can be denoted as:

$$\bar{x} = x + f_x + v_x \tag{12}$$

where $\bar{x}$ is the measured value, x is the actual value, f_x is the measurement deviation caused by sensor faults, and v_x is the measurement noise. The measurement deviation in this study has two forms: fixed bias (Eq. (13)) and drift (Eq. (14)). Fixed bias means the difference between the measured value $\bar{x}$ and the actual value x is a constant C_0 while drift means this difference changes with time.

$$f_x = C_0 = \alpha \times x \tag{13}$$

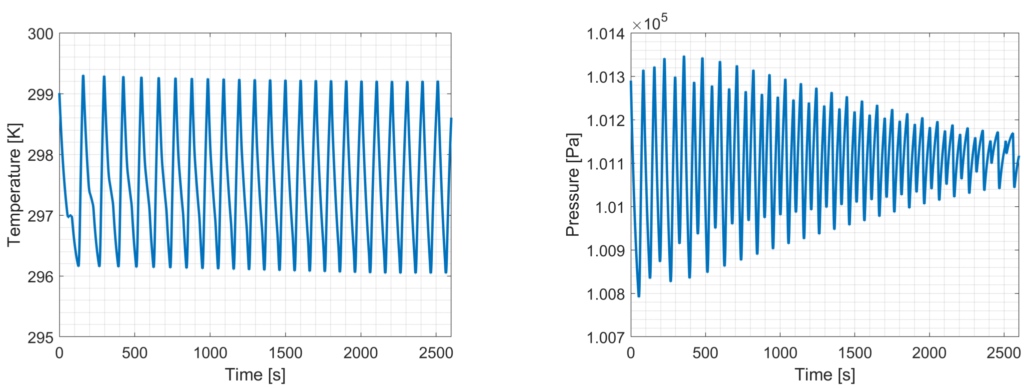
$$f_x = \beta \times \frac{t - t_0}{t_f - t_0} \times x \tag{14}$$

where α, β are the bias and drift percentages of the faulty sensor, respectively, t is the current simulation time, t_0 is the initial simulation time when the sensor starts to drift, and t_f is the final simulation time. Fixed bias and drift, as the faults in the temperature and pressure sensors, have been investigated in numerous studies [5, 48, 80–83], including the ones carried out by NASA [48, 80].

C. Data Collection

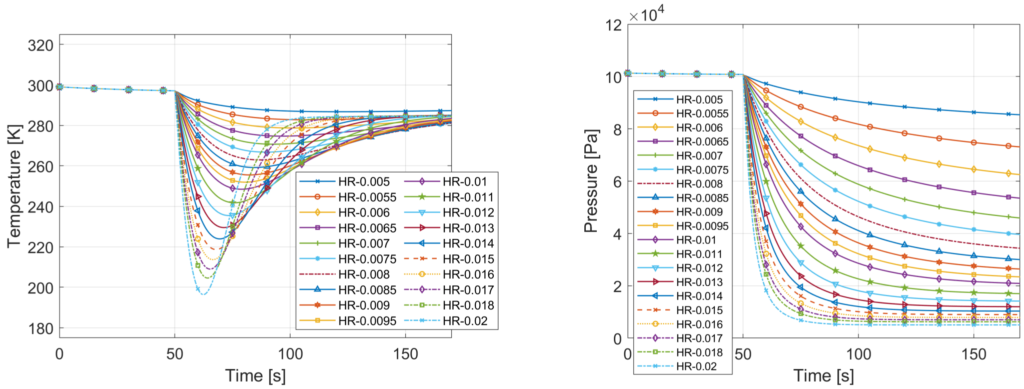
HabSim is employed to simulate the nominal state of the habitat, as well as micrometeorite and fire scenarios under different circumstances. Fig. 5 demonstrates the simulated data in the nominal state for the duration of 2600 sec, where the temperature and pressure data are collected with a sampling rate of 50 Hz. In the nominal state, the temperature and pressure in the IE fluctuate within a relatively small dead band. Micrometeorite strikes are set to penetrate both the SPL and the ST, creating a hole in the wall and thus causing air leakage of the habitat. HabSim is used to calculate the temperature and pressure inside the habitat for 20 different hole radii, which range from 0.005 to 0.02 m. Specifically, for each micrometeorite impact scenario, the impact occurs at 50 seconds, and the total simulation time is 170 seconds. As shown in Fig. 6, the drops in the temperature and pressure, which are proportional to the size of the hole radius, occur right after the micrometeorite impact. Despite the air leakage, ECLSS can regulate the temperature inside the habitat and keep it within the dead-band set points, mainly due to the small remaining amount of air. Meanwhile, ECLSS can only compensate for the pressure and keep it below the dead-band set points, depending on the radius of the hole. The temperature and pressure data in fire scenarios are collected with fire intensity levels (ILs) of 2, 3, 4, and 5, while the IL of 1 indicates the nominal state. For each fire IL, different fire spread rates ranging from 0.0035 to 0.00475 m/s are selected. Similarly, for each fire scenario, the fire starts at 50 sec, and the total simulation time is 170 sec. As shown in Fig. 7, both temperature and pressure increase when a fire incident takes place. The larger fire IL contributes to the larger temperature and pressure rises. Due to the significant amount of heat caused by the fire, ECLSS fails to regulate the IE, which leads to a monotonous increase in both temperature and pressure.

The collected data in each habitat scenario is split into training and testing datasets, which are summarized in Tables 1 - 3. In this study, four temperature sensors and four pressure sensors are deployed inside the habitat. The sensor noise with a signal-to-noise ratio of 70 is added to all of the sensors. For each testing case, a target sensor is randomly selected from eight sensors inside the habitat. The state of the target sensor is randomly assigned as either normal, fixed bias, or drift. The sensor faults with variable parameters are added to the target sensor using Eqs. (12) - (14) if it has a faulty state. Specifically, the fault start time, fault duration, bias percentage, and drift percentage are randomly selected for each faulty sensor. Bias and drift percentages are within the range between 5% and 50%. In the end, there are 5,280 testing cases generated in each of the habitat scenarios, including the nominal state, micrometeorite impact, and fire scenarios. Fig. 8 demonstrates sample testing cases with randomized sensor faults in the nominal state. Additionally, 800 validation cases that are different from the testing cases are generated in each habitat scenario. The validation dataset is used to select appropriate thresholds for sensor fault detection.



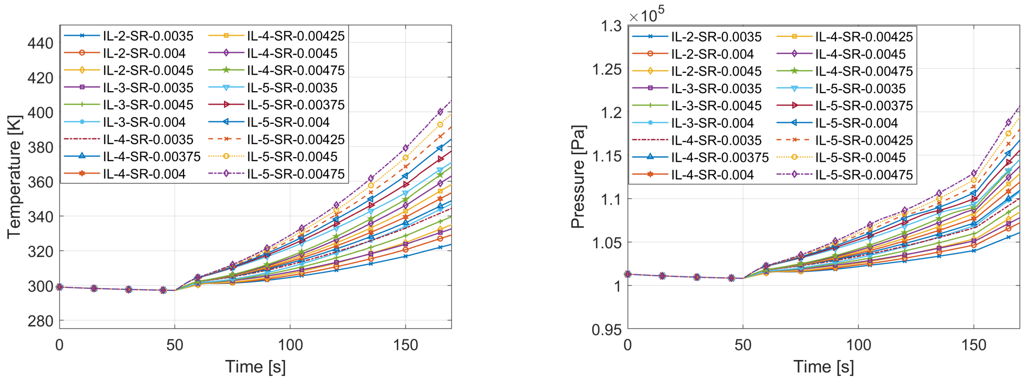
(a) Temperature **(b) Pressure**

Fig. 5 Temperature and pressure data collected by running HabSim in the nominal state.



(a) Temperature **(b) Pressure**

Fig. 6 Temperature and pressure variations in micrometeorite impact scenarios. 'HR' denotes the hole radius in meters.



(a) Temperature **(b) Pressure**

Fig. 7 Temperature and pressure variations in fire scenarios. 'IL' denotes the fire intensity level. 'SR' denotes the fire spread rate in m/s.

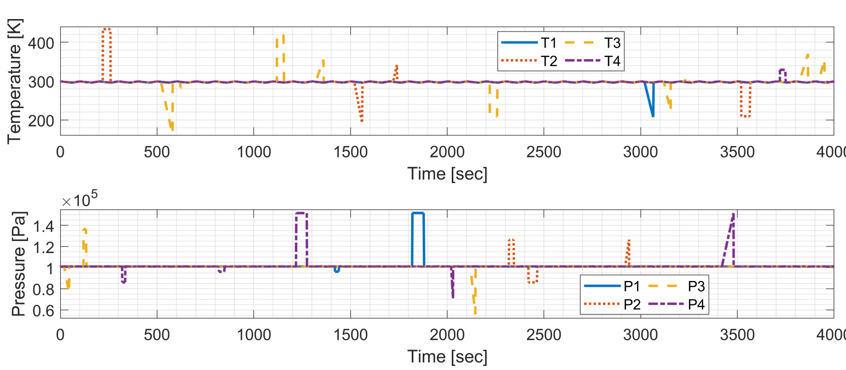


Fig. 8 Sample testing cases with randomized sensor faults in the nominal state.

Table 1 Training and testing datasets in the nominal state

Simulation time	Total 2600 sec
Training data	First 2000 out of 2600 sec
Testing data	Last 600 out of 2600 sec 5280 testing cases with randomized sensor faults

Table 2 Training and testing datasets in micrometeorite impact scenarios

Simulation time	170 sec per scenario
Training data	14 scenarios with HR [m]: 0.005, 0.006, 0.007, 0.008, 0.009, 0.01, 0.011, 0.012, 0.013, 0.014, 0.015,0.017,0.018, 0.02
Testing data	6 scenarios with HR [m]: 0.0055, 0.0065, 0.0075, 0.0085, 0.0095, 0.016 5280 testing cases with randomized sensor faults

Table 3 Training and testing datasets in fire scenarios

Simulation time	170 sec per scenario
Training data	14 scenarios with IL/SR [m/s]: 2/0.0035, 2/0.0045, 3/0.0035, 3/0.0045, 4/0.0035, 4/0.00375, 4/0.00425, 4/0.0045, 4/0.00475, 5/0.0035, 5/0.00375, 5/0.00425, 5/0.0045, 5/0.00475
Testing data	4 scenarios with IL/SR [m/s]: 2/0.004, 3/0.004, 4/0.004, 5/0.004 5280 testing cases with randomized sensor faults

IV. Illustrative Examples: Results and Discussion

A. AANN versus CAE versus VAE

AANN has been proposed and utilized for sensor fault detection for over 30 years, and its applications are focused on engine health monitoring [49, 50, 56–60]. It should be noted that AANN has been utilized by NASA for sensor fault detection in aircraft engines since 1998 [49, 50]. Therefore, it is meaningful to compare our approach with this well-established sensor fault detection approach in aerospace. Recently, VAE, as a probabilistic generative model, has been applied to anomaly detection in time series [31–33, 35]. In this study, a fully convolutional autoencoder is developed and applied to detect sensor faults in space habitats. Unlike AANN and CAE, which have a deterministic bottleneck layer, VAE has a probabilistic bottleneck layer. To fairly compare the performance of these three methods, the optimal architectures of AANN, CAE, and VAE should be determined. To this end, 9 architectures of AANN, 12 architectures of CAE, and 11 architectures of VAE are implemented to find the optimal architectures, as shown in Tables 4, 5, and 6, respectively. Since the encoder (mapping function) and the decoder (demapping function) are symmetric networks, only the configurations of encoders (mapping functions) are provided in these tables. For each architecture, AANN, CAE, and VAE models are trained and tested in different habitat scenarios using the datasets summarized in Tables 1-3. The thresholds used for sensor fault detection in each trained AANN, CAE, and VAE model are determined based on the validation dataset, and the ones that contribute to a comparable number of false positives and false negatives are selected. The performance of each architecture is evaluated by the F1 score, which is the harmonic mean of precision and recall. The precision, recall, and F1 score can be calculated using Eqs. (15) - (17). The training and testing time of AANN, CAE, and VAE are compared on both CPU (Intel Core i7-9700K Processor) and GPU (NVIDIA GeForce RTX 2080 Ti). The results obtained by using different architectures of AANN, CAE, and VAE are summarized in Tables 4, 5, and 6respectively.

$$Precision = \frac{TP}{TP + FP} \tag{15}$$

$$Recall = \frac{TP}{TP + FN} \tag{16}$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{17}$$

where TP denotes true positive, FP denotes false positive, and FN denotes false negative.

In this study, the HMS is assumed to be able to detect abnormal events (i.e., micrometeorite impact and fire) prior to sensor fault detection. Under this assumption, individual AANN, CAE, and VAE models are trained in each habitat scenario. The corresponding trained AANN, CAE, and VAE models can be selected accordingly based on the current habitat state. In the nominal state, the optimal AANN-3, CAE-1, and VAE-1 can achieve an F1 score of 1.00. This is because temperature and pressure remain in a small range in the nominal state, as shown in Fig. 5, which enables AANN, CAE, and VAE to reconstruct the temperature and pressure signals precisely. However, the F1 score is less than 1.00 using either AANN, CAE, or VAE in micrometeorite impact and fire scenarios, which is mainly due to the large fluctuation in temperature and pressure after events happen, as shown in Figs. 6 and 7. Moreover, in micrometeorite impact scenarios, the optimal AANN-4, CAE-11, and VAE-6 have F1 scores of 0.60, 0.65, and 0.64, respectively. It is indicated that CAE performs better than both AANN and VAE in micrometeorite impact scenarios. However, in fire scenarios, the optimal AANN-1, CAE-6, and VAE-11 have F1 scores of 0.86, 0.80, and 0.79. It is observed that AANN outperforms CAE and VAE in fire scenarios. The detailed results obtained by the optimal AANN, CAE, and VAE models in different habitat scenarios are summarized in Table 7. Regarding the computational costs, AANN has a faster training time than CAE and VAE, but CAE and VAE have a faster testing time than AANN. Given the constrained computational resources in space habitats, the CPU can be utilized to evaluate computational costs. For example, in the nominal state, the training times for AANN-3, CAE-1, and VAE-1 are 1.0 min, 1.8 min, and 6.1 min, respectively. The corresponding testing times are 9×10^{-4} sec, 1×10^{-5} sec, and 9×10^{-6} sec, respectively. This difference is attributed to AANN having fewer layers and trainable parameters, which contributes to a faster training time. However, AANN needs to take every single data point to make predictions, while CAE and VAE utilize a sequence of data points to reconstruct the signal, which makes their testing time faster than AANN's.

To further investigate the performance of AANN and CAE, sample testing cases in different habitat scenarios are

Table 4 AANN architectures

AANN model	Number of neurons in each layer (mapping function)	Nominal state			Micrometeorite impact			Fire		
		F1 score	Train time	Test time	F1 score	Train time	Test time	F1 score	Train time	Test time
AANN-1	8-64-3	0.99	1.0 (5.7)	800 (900)	0.58	1.2 (6.6)	900 (900)	**0.86**	**1.2 (6.9)**	**1800 (1800)**
AANN-2	8-64-4	0.99	1.0 (5.4)	800 (900)	0.59	1.2 (6.6)	900 (900)	0.79	1.2 (6.8)	1800 (1800)
AANN-3	8-64-5	**1.00**	**1.0 (5.5)**	**900 (900)**	0.58	1.2 (6.5)	900 (900)	0.82	1.2 (6.7)	1800 (1800)
AANN-4	8-128-3	1.00	1.0 (5.5)	900 (900)	**0.60**	**1.2 (6.8)**	**900 (900)**	0.82	1.2 (6.5)	1800 (1800)
AANN-5	8-128-4	1.00	1.0 (5.4)	900 (900)	0.57	1.2 (6.7)	900 (900)	0.79	1.2 (6.4)	1800 (1800)
AANN-6	8-128-5	1.00	1.1 (5.4)	900 (900)	0.59	1.2 (6.8)	900 (900)	0.80	1.2 (6.6)	1800 (1800)
AANN-7	8-256-3	0.99	1.1 (5.5)	900 (900)	0.58	1.3 (6.4)	900 (700)	0.81	1.3 (6.2)	1800 (1400)
AANN-8	8-256-4	0.99	1.2 (5.6)	900 (900)	0.59	1.4 (6.8)	900 (700)	0.76	1.4 (6.4)	1800 (1400)
AANN-9	8-256-5	0.99	1.2 (5.5)	900 (900)	0.58	1.4 (6.6)	900 (700)	0.80	1.4 (6.6)	1800 (1400)

CPU: Intel Core i7-9700K CPU.
GPU: NVIDIA GeForce RTX 2080 Ti.
The computational times shown outside the parentheses are evaluated on the CPU, and the ones shown inside the parentheses are evaluated on the GPU.
The unit of training time is in minutes, and the unit of testing time is in microseconds (i.e., 10^{-6} sec).
The bolded results represent the ones obtained by the optimal architecture.

Table 5 CAE architectures

CAE model	Number of channels in each layer (encoder)	Nominal state			Micrometeorite impact			Fire		
		F1 score	Train time	Test time	F1 score	Train time	Test time	F1 score	Train time	Test time
CAE-1	2-16-32	**1.00**	**1.8 (1.6)**	**10 (9)**	0.52	2.6 (2.5)	10 (9)	0.78	3.6 (3.2)	13 (10)
CAE-2	2-16-32-64	1.00	1.8 (1.6)	11 (9)	0.57	3.2 (3.0)	12 (9)	0.75	3.8 (3.2)	14 (10)
CAE-3	2-16-32-64-128	1.00	2.0 (1.6)	14 (9)	0.60	4.1 (3.6)	15 (10)	0.75	4.5 (3.0)	18 (10)
CAE-4	2-32-64	1.00	1.8 (1.6)	11 (9)	0.55	2.9 (2.6)	12 (10)	0.79	4.0 (3.2)	15 (11)
CAE-5	2-32-64-128	1.00	2.1 (1.6)	14 (9)	0.58	4.0 (3.1)	15 (9)	0.76	4.6 (3.1)	20 (11)
CAE-6	2-32-64-128-256	1.00	2.5 (1.7)	20 (9)	0.62	5.7 (3.8)	25 (10)	**0.80**	**5.9 (3.0)**	**26 (11)**
CAE-7	2-64-128	1.00	2.1 (1.6)	14 (10)	0.59	3.2 (2.7)	14 (10)	0.73	4.7 (3.2)	18 (13)
CAE-8	2-64-128-256	1.00	2.4 (1.6)	20 (9)	0.62	5.1 (3.0)	23 (9)	0.75	6.3 (3.2)	29 (11)
CAE-9	2-64-128-256-512	1.00	3.4 (1.7)	32 (10)	0.43	9.3 (3.8)	59 (11)	0.75	9.9 (3.3)	50 (12)
CAE-10	2-128-256	1.00	2.5 (1.6)	19 (11)	0.60	5.9 (2.6)	19 (11)	0.77	6.4 (3.3)	27 (15)
CAE-11	2-128-256-512	1.00	3.8 (1.7)	36 (12)	**0.65**	**12.9 (3.2)**	**69 (10)**	0.73	10.0 (3.2)	46 (14)
CAE-12	2-128-256-512-1024	1.00	6.5 (1.9)	56 (14)	0.46	28.4 (4.1)	319 (13)	0.77	16.7 (3.8)	86 (14)

CPU: Intel Core i7-9700K CPU.
GPU: NVIDIA GeForce RTX 2080 Ti.
The computational times shown outside the parentheses are evaluated on the CPU, and the ones shown inside the parentheses are evaluated on the GPU.
The unit of training time is in minutes, and the unit of testing time is in microseconds (i.e., 10^{-6} sec).
The bolded results represent the ones obtained by the optimal architecture.

Table 6 VAE architectures

CAE model	Number of neurons in each layer (encoder)	Nominal state			Micrometeorite impact			Fire		
		F1 score	Train time	Test time	F1 score	Train time	Test time	F1 score	Train time	Test time
VAE-1	800-64-3	**1.00**	**6.1 (6.9)**	**9 (9)**	0.58	6.1 (7.0)	8 (9)	0.45	6.3 (7.3)	9 (10)
VAE-2	800-128-3	1.00	6.1 (6.8)	8 (9)	0.57	6.2 (7.3)	8 (10)	0.45	6.6 (7.3)	9 (11)
VAE-3	800-256-3	1.00	6.4 (6.8)	9 (9)	0.59	6.3 (6.9)	8 (10)	0.50	7.6 (7.2)	10 (12)
VAE-4	800-512-3	1.00	7.4 (6.8)	9 (9)	0.54	7.6 (7.2)	9 (11)	0.45	9.0 (7.2)	10 (9)
VAE-5	800-512-256-3	Too many trainable parameters cause model overfitting			0.61	8.3 (7.2)	9 (11)	0.50	9.9 (7.3)	10 (10)
VAE-6	800-512-256-128-3				**0.64**	**9.0 (7.5)**	**9 (9)**	0.48	10.5 (7.5)	10 (12)
VAE-7	800-512-256-128-8				0.60	9.0 (7.6)	9 (11)	0.46	10.5 (7.5)	10 (13)
VAE-8	800-512-256-128-16				0.59	9.0 (7.5)	9 (12)	0.47	10.5 (7.5)	10 (13)
VAE-9	800-512-256-128-64-3				0.61	9.3 (7.5)	9 (9)	0.47	10.6 (7.5)	10 (10)
VAE-10	800-512-256-128-64-32-3				0.62	9.3 (7.7)	9 (9)	**0.79**	**10.9 (7.8)**	**11 (10)**
VAE-11	800-1024-3				0.57	9.1 (7.0)	9 (11)	0.52	12.2 (7.2)	11 (12)

CPU: Intel Core i7-9700K CPU.
GPU: NVIDIA GeForce RTX 2080 Ti.
The computational times shown outside the parentheses are evaluated on the CPU, and the ones shown inside the parentheses are evaluated on the GPU.
The unit of training time is in minutes, and the unit of testing time is in microseconds (i.e., 10^{-6} sec).
The bolded results represent the ones obtained by the optimal architecture.

Table 7 Results obtained by optimal AANNs, CAEs, and VAEs

	Data-driven model	True positive	True negative	False positive	False negative	Precision	Recall	F1 score
Nominal state	CAE-1	441	4839	0	0	1.00	1.00	1.00
	AANN-3	441	4836	3	0	0.99	1.00	1.00
	VAE-1	441	4839	0	0	1.00	1.00	1.00
Micrometeorite impact	CAE-11	263	4730	105	182	0.71	0.59	0.65
	AANN-4	259	4673	162	186	0.62	0.58	0.60
	VAE-6	270	4706	129	175	0.68	0.61	0.64
Fire	CAE-6	359	4739	91	91	0.80	0.80	0.80
	AANN-1	381	4771	59	69	0.87	0.85	0.86
	VAE-10	389	4688	142	61	0.73	0.86	0.79

shown in Fig. 9, and the corresponding detection results are visualized in Fig. 10. T1 to T4 denote four temperature sensors, and P1 to P4 denote four pressure sensors. The yellow regions represent the detected faults, while the red boxes represent the actual faults. If the yellow region coincides with the red box, the approach detects the sensor fault correctly. The temperature and pressure reconstructed by AANNs and CAEs are compared with the actual ones in Fig. 11. According to Figs. 10a and 10b, it is shown that both AANN and CAE can detect the sensor faults accurately in the nominal state, as all the yellow regions (i.e., detected sensor faults) coincide with the red boxes (i.e., actual sensor faults). This can be explained by the good data reconstruction of both AANN and CAE in the nominal state, as presented in Figs. 11a and 11b.

However, in micrometeorite impact Scenario #5, AANN produces false alarms when detecting the faults in temperature sensors, as shown in Fig. 10d. According to Figs. 11c and 11d, AANN overfits the training data and produces a better data reconstruction in fault-free cases. However, CAE has a better generalization capability since it can reconstruct the signal better than AANN (i.e., the reconstructed signal is closer to the actual one) when there is a sensor fault. In other words, the data reconstruction of AANN is more vulnerable to sensor faults. For example, in micrometeorite impact Scenario #5, the fixed bias in temperature sensor T1 causes a larger reconstruction error in AANN. When using AANN, the data reconstruction for other temperature sensors, including T2, T3, and T4, is also affected by the sensor fault in T1, which leads to the false alarms in these temperature sensors, as shown in Fig. 10d.

On the other hand, in fire scenarios, AANN produces better detection performance than CAE. For instance, in fire Scenario #4, CAE can barely detect the sensor fault in T2, while AANN has a higher confidence in detecting the fault, as demonstrated in Figs. 10e and 10f. According to the reconstructed data shown in Figs. 11e and 11f, AANN has better data reconstruction than CAE even if there is a drift in sensor T2. It can be observed that both temperature and pressure just monotonously increase after the fire happens, which contributes to a simpler data variation than that in micrometeorite impact scenarios. Accordingly, the generalization capability of the trained AANN is higher in fire

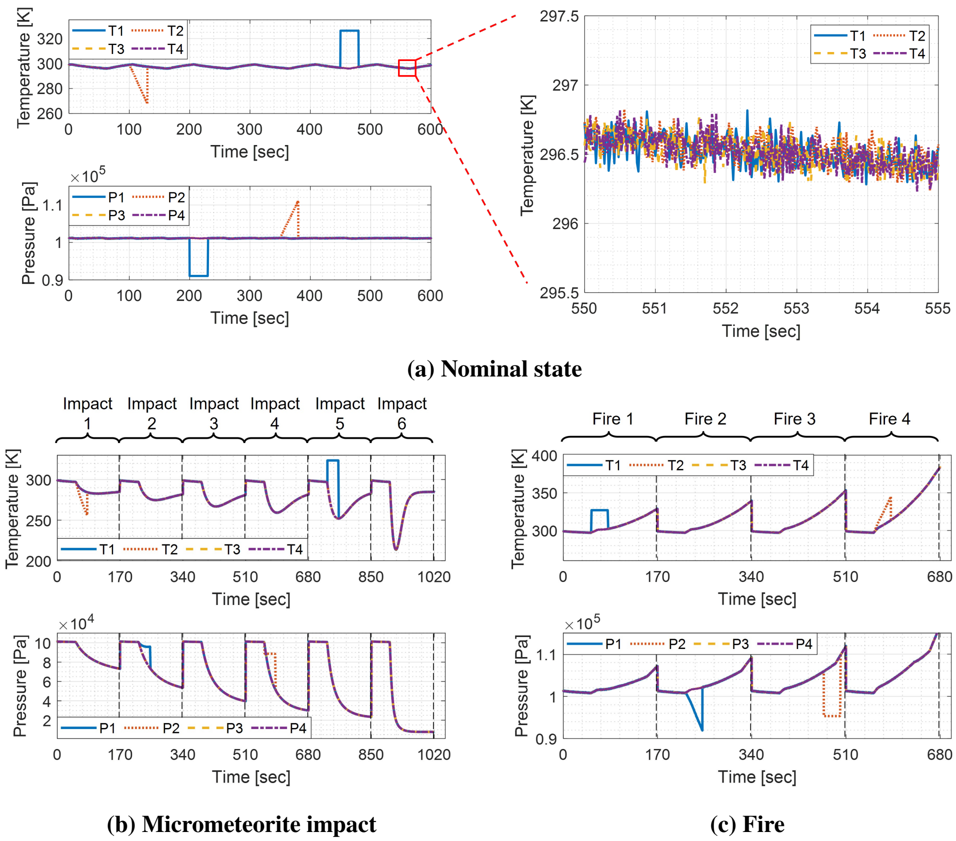


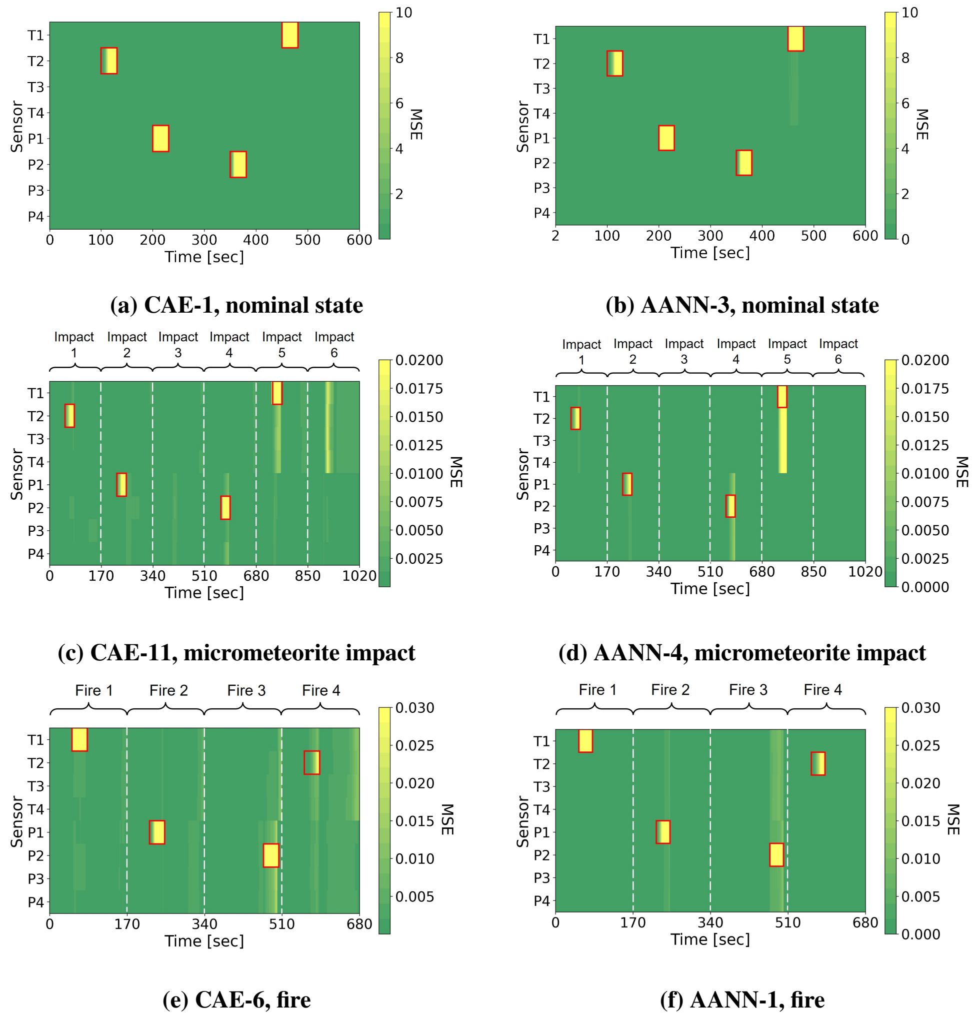
Fig. 9 Sample testing cases. The sensor noise with a signal-to-noise ratio of 70 is added. Sensor faults are randomly injected into the testing data.

scenarios than in micrometeorite impact scenarios, which leads to a better detection performance of AANN in fire scenarios.

B. Temperature-pressure Coupling

Temperature and pressure interact with each other in the interior environment, which can be simulated in HabSim. For example, in the nominal state, both temperature and pressure remain in a small region, as shown in Fig. 5. Moreover, both temperature and pressure decrease after the micrometeorite impact happens, as shown in Fig. 6. The drop speeds of temperature and pressure are correlated, and they are both proportional to the size of the hole radius caused by the micrometeorite strike. Additionally, both temperature and pressure increase after the fire starts, as shown in Fig. 7. The rise speeds of temperature and pressure are also correlated, and they are both proportional to fire ILs and spread rates.

To investigate the effect of the coupling between temperature and pressure on the detection performance of CAE, the CAE models with the same architecture are trained and tested using the temperature-only data, the pressure-only data, and the combined temperature and pressure data, respectively. According to Table 7, the optimal CAE-1, CAE-11, and CAE-6 in the nominal state, micrometeorite impact, and fire scenarios are selected, respectively. When training the



(a) CAE-1, nominal state

(b) AANN-3, nominal state

(c) CAE-11, micrometeorite impact

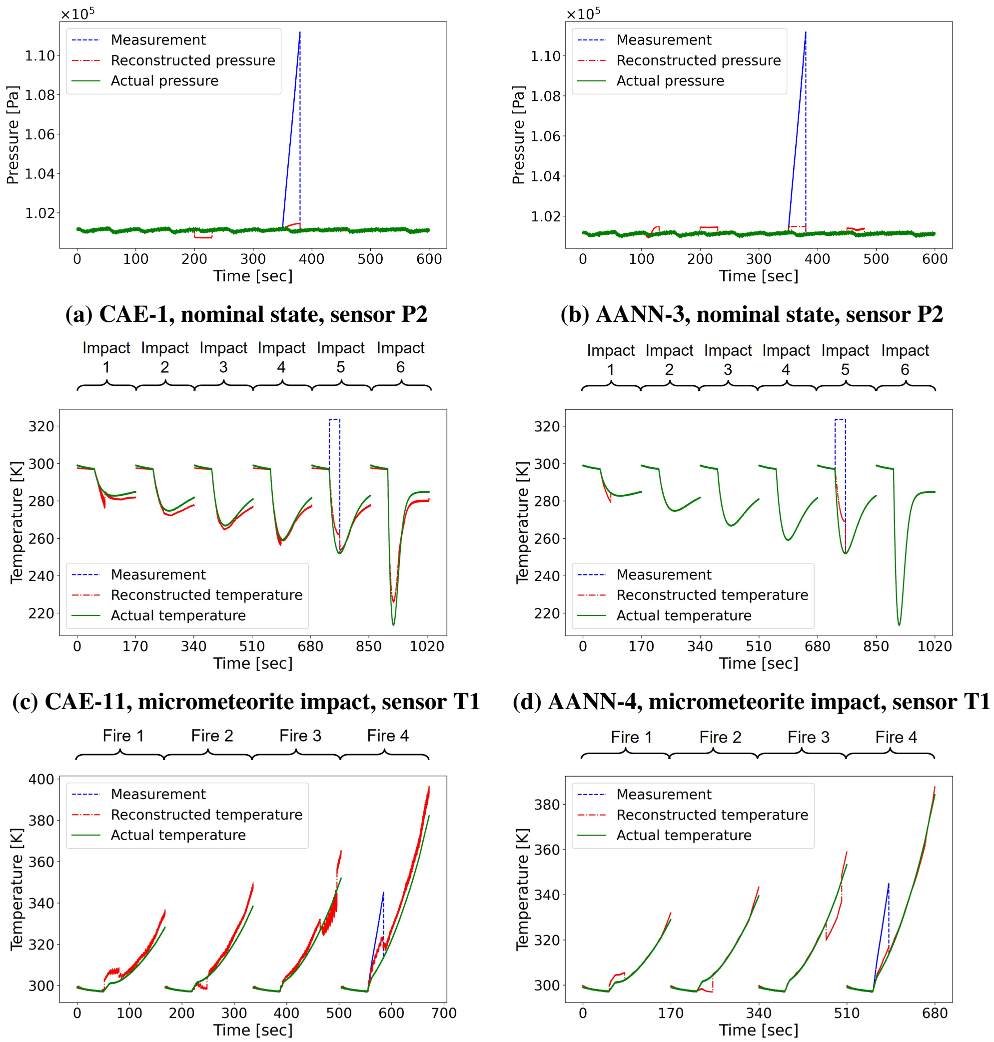
(d) AANN-4, micrometeorite impact

(e) CAE-6, fire

(f) AANN-1, fire

Fig. 10 Visualization of detection results.

optimal CAE models without the coupling, the size of the input and output is modified to enable them to have a single channel such that they are compatible with temperature-only or pressure-only data. The results obtained by the CAEs with and without the temperature-pressure coupling in different habitat scenarios are summarized in Table 8. In the nominal state, CAE can achieve an F1 score of 1.00 either with or without the coupling. However, in micrometeorite impact and fire scenarios, the CAE with the coupling can produce higher F1 scores, as the coupling enhances the redundant information among sensors. It should be noted that even with the coupling, CAE performs better on the temperature data in micrometeorite impact scenarios, while it performs better on the pressure data in fire scenarios. This is because, as can be observed in Figs. 6a and 7b, temperature and pressure have smaller decrease and increase rates in micrometeorite impact and fire scenarios, respectively. By modifying the number of neurons in the input and output layers, the AANN models with the same architecture can be trained and tested using the temperature-only data, the pressure-only data, and the combined temperature and pressure data, respectively. The results obtained by the optimal



(a) CAE-1, nominal state, sensor P2

(b) AANN-3, nominal state, sensor P2

(c) CAE-11, micrometeorite impact, sensor T1

(d) AANN-4, micrometeorite impact, sensor T1

(e) CAE-6, fire, sensor T2

(f) AANN-1, fire, sensor T2

Fig. 11 Comparison between reconstructed and actual signals.

AANNs (i.e., AANN-3, AANN-4, and AANN-1) with and without the temperature-pressure coupling are summarized in Table 9. It is indicated that leveraging the coupling between temperature and pressure can also enhance the detection performance of AANN. Additionally, similar to CAE, AANN performs better on the temperature data in micrometeorite impact scenarios, while it performs better on the pressure data in fire scenarios.

Utilizing the sample testing cases shown in Fig. 9, the detection results obtained by the CAEs with and without the temperature-pressure coupling are visualized in Figs. 10 and 12, respectively. It can be observed that, in the nominal state, the CAEs with and without the coupling have comparable performances, and all the sensor faults can be correctly detected. However, in micrometeorite impact Scenario #6, the CAE without the coupling produces false alarms in temperature sensors T1 to T4, as shown in Fig. 12c. Additionally, in fire Scenario #4, the fault in temperature sensor T2 is completely missed when using the CAE without the coupling, as presented in Fig. 12e. Moreover, in fire Scenario #3,

Table 8 Results obtained by the CAEs with and without the temperature-pressure coupling

			True positive	True negative	False positive	False negative	Precision	Recall	F1 score
Nominal state CAE-1	With coupling	Temperature	219	2421	0	0	1.00	1.00	1.00
		Pressure	222	2418	0	0	1.00	1.00	1.00
	Without coupling	Temperature	219	2421	0	0	1.00	1.00	1.00
		Pressure	222	2418	0	0	1.00	1.00	1.00
Meteorite impact CAE-11	With coupling	Temperature	160	2416	21	43	0.88	0.79	0.83
		Pressure	103	2314	84	139	0.55	0.43	0.48
	Without coupling	Temperature	126	2412	25	77	0.83	0.62	0.71
		Pressure	92	2315	83	150	0.53	0.38	0.44
Fire CAE-6	With coupling	Temperature	146	2354	71	69	0.67	0.68	0.68
		Pressure	213	2385	20	22	0.91	0.91	0.91
	Without coupling	Temperature	139	2345	80	76	0.63	0.65	0.64
		Pressure	179	2351	54	56	0.77	0.76	0.76

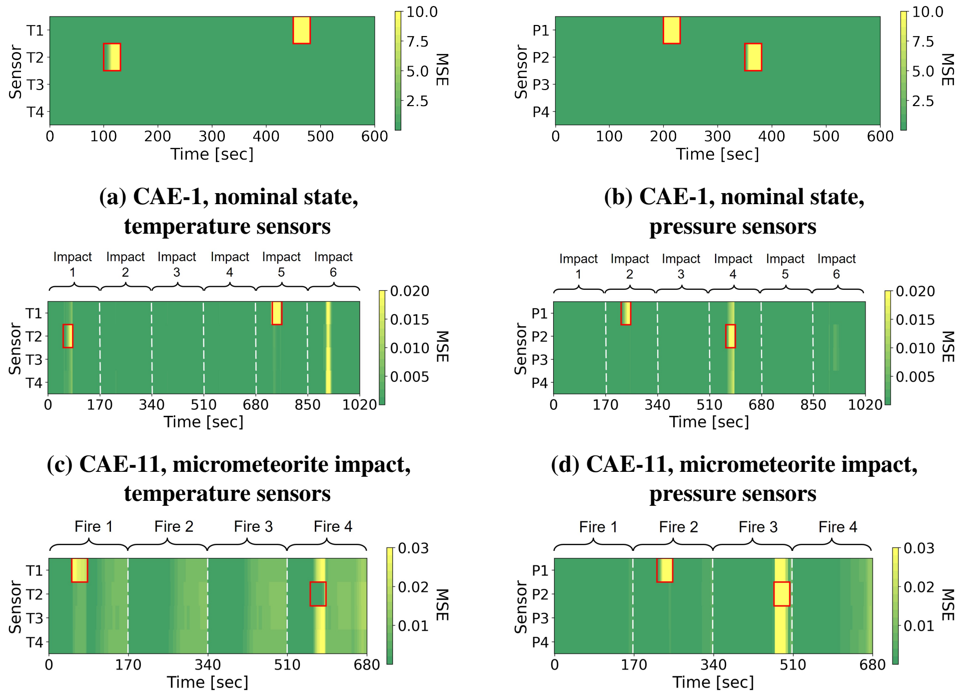
Table 9 Results obtained by the AANNs with and without the temperature-pressure coupling

			True positive	True negative	False positive	False negative	Precision	Recall	F1 score
Nominal state AANN-3	With coupling	Temperature	219	2418	3	0	0.99	1.00	0.99
		Pressure	222	2418	0	0	1.00	1.00	1.00
	Without coupling	Temperature	219	2419	2	0	0.99	1.00	1.00
		Pressure	222	2418	0	0	1.00	1.00	1.00
Meteorite impact AANN-4	With coupling	Temperature	152	2383	54	51	0.74	0.75	0.74
		Pressure	107	2290	108	135	0.50	0.44	0.47
	Without coupling	Temperature	150	2380	57	53	0.72	0.74	0.73
		Pressure	116	2257	141	126	0.45	0.48	0.46
Fire AANN-1	With coupling	Temperature	167	2389	36	48	0.82	0.78	0.80
		Pressure	214	2382	23	21	0.90	0.91	0.91
	Without coupling	Temperature	167	2379	46	48	0.78	0.78	0.78
		Pressure	205	2332	73	30	0.74	0.87	0.80

the CAE without the coupling generates false alarms in pressure sensors P1, P3, and P4, while detecting the fault in the pressure sensor P2, as demonstrated in Fig. 12f. Therefore, the CAE with the temperature-pressure coupling can achieve a better detection performance as a result of the higher redundant information among sensors provided by the correlation between temperature and pressure in the interior environment.

C. Multiple Sensor Failures

In the habitat simulator, there are four temperature sensors and four pressure sensors in the interior environment. Multiple sensors measuring the same physical property provide redundant information, and this sensor redundancy is critical for CAE and AANN to reconstruct the signal of faulty sensors accurately. The illustrative examples presented in



(a) **CAE-1, nominal state, temperature sensors**

(b) **CAE-1, nominal state, pressure sensors**

(c) **CAE-11, micrometeorite impact, temperature sensors**

(d) **CAE-11, micrometeorite impact, pressure sensors**

(e) **CAE-6, fire, temperature sensors**

(f) **CAE-6, fire, pressure sensors**

Fig. 12 Visualization of the detection results obtained by the CAEs without the temperature-pressures coupling.

Sections IV.A and IV.B only consider a single faulty sensor occurring at a time. However, multiple sensors could be damaged simultaneously, which is likely to happen in micrometeorite impact and fire scenarios. It should be noted that when multiple sensors fail at the same time, it leads to less redundant information among the sensors, which can deteriorate the detection performance of CAE. In this regard, the detection performance of CAE is also evaluated in cases where two, three, and four sensors fail simultaneously. To this end, in each habitat scenario, three new testing datasets with 5280 testing cases in each dataset are generated by randomly selecting two, three, and four faulty sensors from a total of eight sensors. The fixed bias or drift faults are randomly inserted into the selected faulty sensors. The detection results under multiple sensor failures using the optimal CAEs (i.e., CAE-1, CAE-11, and CAE-6) and optimal AANNs (i.e., AANN-3, AANN-4, AANN-1) are summarized in Tables 10 and 11, respectively.

In the nominal state, CAE can maintain a good performance with an F1 score of 1.00 when one, two, three, or four sensors fail at the same time. This ability can be explained by the simpler data variation of temperature and pressure in the nominal state, which enables CAE to retain a higher sensor redundancy even with multiple faulty sensors. On the other hand, in micrometeorite impact and fire scenarios, the number of false positives increases and the F1 score decreases with the increase in the number of faulty sensors. This is because the complex data variation of temperature and pressure in micrometeorite impact and fire scenarios makes the data reconstruction of CAE vulnerable to anomalous data from faulty sensors. In this case, increasing faulty sensors can significantly reduce the redundant information

Table 10 Results using CAEs under multiple sensor failures

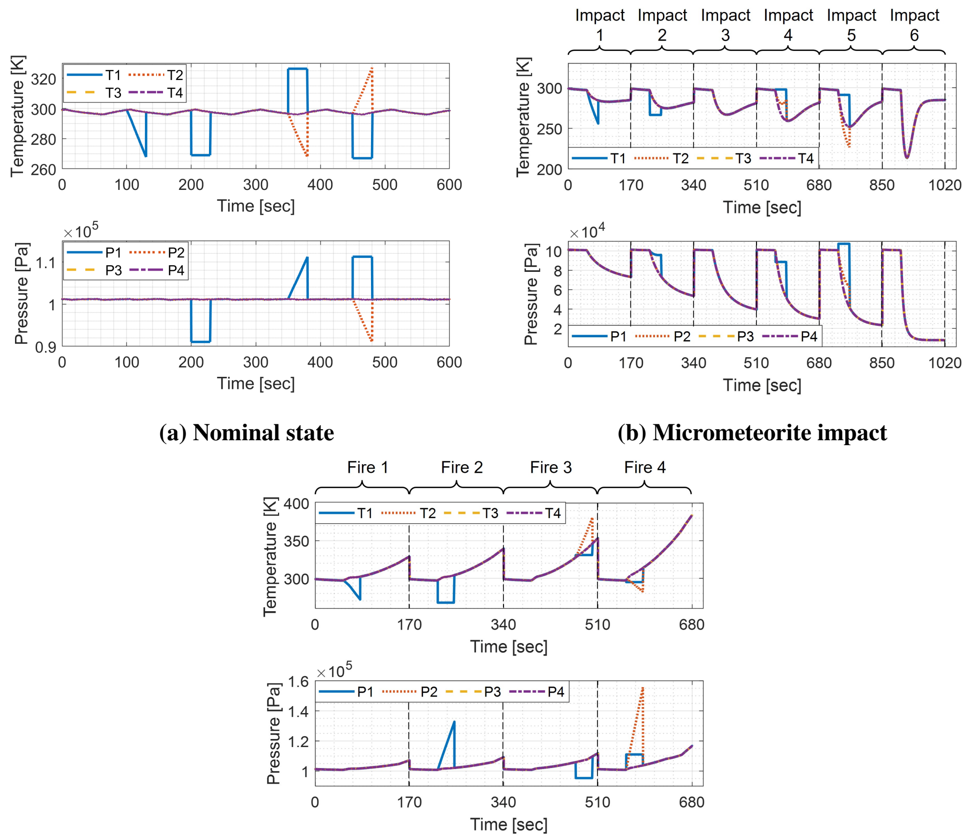
	Number of faulty sensors	True positive	True negative	False positive	False negative	Precision	Recall	F1 score
Nominal state CAE-1	1	441	4839	0	0	1.00	1.00	1.00
	2	626	4654	0	0	1.00	1.00	1.00
	3	918	4362	0	0	1.00	1.00	1.00
	4	1300	3980	0	0	1.00	1.00	1.00
Micrometeorite impact CAE-11	1	263	4730	105	182	0.71	0.59	0.65
	2	217	4510	426	127	0.34	0.63	0.44
	3	216	4303	650	111	0.25	0.66	0.36
	4	231	4053	893	103	0.21	0.69	0.32
Fire CAE-6	1	359	4739	91	91	0.80	0.80	0.80
	2	261	4551	406	62	0.39	0.81	0.53
	3	269	4217	737	57	0.27	0.83	0.40
	4	284	3929	1026	41	0.22	0.87	0.35

Table 11 Results using AANNs under multiple sensor failures

	Number of faulty sensors	True positive	True negative	False positive	False negative	Precision	Recall	F1 score
Nominal state AANN-3	1	441	4836	3	0	0.99	1.00	1.00
	2	626	4645	9	0	0.99	1.00	0.99
	3	918	4358	4	0	1.00	1.00	1.00
	4	1300	3975	5	0	1.00	1.00	1.00
Micrometeorite impact AANN-4	1	259	4673	162	186	0.62	0.58	0.60
	2	228	4456	480	116	0.32	0.66	0.43
	3	218	4254	699	109	0.24	0.67	0.35
	4	218	3959	987	116	0.18	0.65	0.28
Fire AANN-1	1	381	4771	59	69	0.87	0.85	0.86
	2	284	4630	327	39	0.46	0.88	0.61
	3	282	4233	721	44	0.28	0.87	0.42
	4	294	3944	1011	31	0.23	0.90	0.36

among the sensors. For example, the F1 score under four sensor failures is almost reduced to half of that when only one sensor fails. Moreover, it can be observed that CAE can achieve better detection performance (i.e., higher F1 scores) in fire scenarios than in micrometeorite impact scenarios. This outcome is due to the relatively simpler data variation in fire scenarios (i.e., monotonous increase in both temperature and pressure after a fire happens). According to Tables 10 and 11, it is shown that CAEs and AANNs have comparable performance under multiple sensor failures.

To better understand how CAEs perform under multiple sensor failures, CAEs are tested using sample testing cases shown in Fig. 13. Specifically, the number of faulty sensors happening simultaneously ranges from one to four. The corresponding detection results obtained by CAEs are visualized in Fig. 14. It is indicated that CAE can successfully detect all the sensor faults in the nominal state when the number of faulty sensors occurring concurrently is one, two, three, or four, as shown in Fig. 14a. However, in micrometeorite impact and fire scenarios, even though CAE is able to correctly detect the faults when there are one or two faulty sensors, it may miss some faults in temperature sensors when the number of faulty sensors is three or four, as shown in Figs. 14b and 14c. Therefore, it is recommended to deploy more redundant sensors in micrometeorite impact and fire scenarios than in the nominal state to ensure an adequate detection performance of CAE.



(a) Nominal state

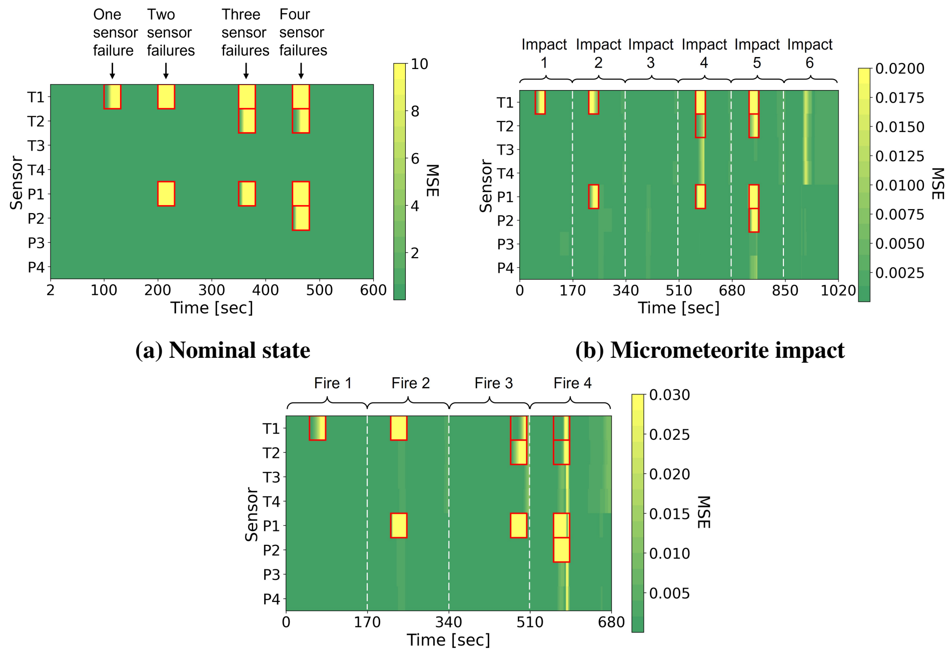
(b) Micrometeorite impact

(c) Fire

Fig. 13 Sample testing cases under multiple sensor failures. The number of faulty sensors happening simultaneously ranges from one to four.

D. Variable Number of Sensors

At the early stage of the commissioning of the habitat, the CAE models are trained using a default sensor configuration. For example, there are four temperature sensors and four pressure sensors in the interior environment, which is the sensor configuration utilized in Sections IV.A, IV.B, and IV.C. But in the future, additional sensors may be added to the habitat, or damaged sensors may be removed from the habitat. In these cases, the pre-trained data-driven models may not work under new sensor configurations. For instance, AANN utilizes fully connected layers in which every



(a) Nominal state

(b) Micrometeorite impact

(c) Fire

Fig. 14 Visualization of the detection results obtained by CAEs under multiple sensor failures.

neuron in one layer is connected with every neuron in its adjacent layers using independent weights, which makes the trained AANN not compatible with different sizes of input. However, since CAE employs convolution operation by sliding a kernel with shared weights across the input, it can be adaptable to different sizes of the input. Therefore, unlike AANN, the CAE trained using the default sensor configuration can also be applied to new sensor configurations without re-designing the network architecture and re-training the neural network. In other words, CAE can continuously work regardless of removing or adding sensors, which is conducive to the resilient operation of space habitats.

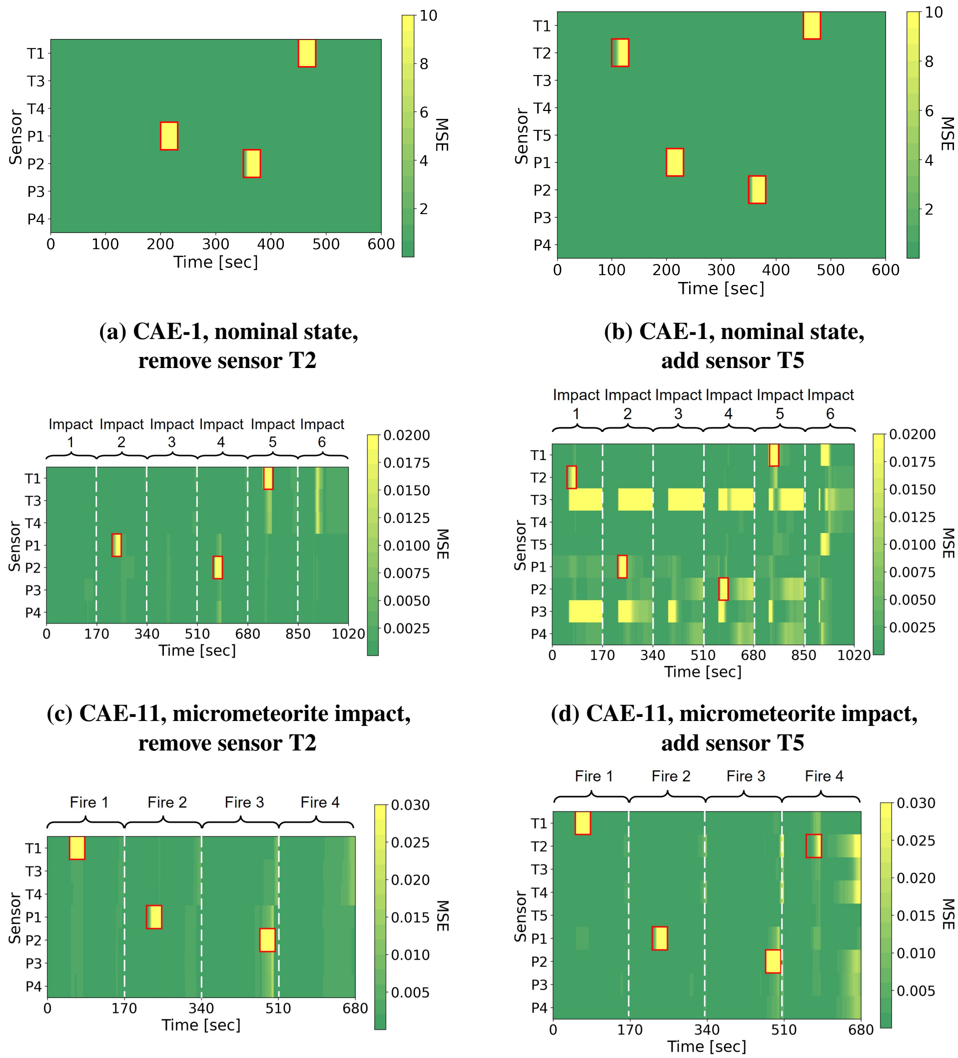
It should be noted that the implementation of the temperature-pressure coupling in CAE, as discussed in Section IV.B, requires the same size of the channels for temperature and pressure. To satisfy this requirement, the strategy using virtual replica sensors is proposed. To be specific, when extra sensors are added, the input size of CAE is increased accordingly. If the number of temperature and pressure sensors is not equal after adding sensors, virtual replica sensors that copy the data from the adjacent physical sensors are introduced to make sure the size of the channels for temperature and pressure is the same. The output size of CNN can be automatically adjusted using convolution operation to match the input size. When faulty sensors are removed, the input size of CAE remains unchanged, and the removed sensors are replaced with virtual replica sensors.

To validate the capability of CAE to be adaptable to a variable number of sensors, the CAEs trained using the default sensor configuration in different habitat scenarios (i.e., CAE-1, CAE-11, and CAE-6) are tested using two new sensor configurations where one temperature sensor is added and removed, respectively. Single sensor failure cases are randomly added to the testing datasets using the strategy discussed in Section III.C. In the end, 4620 and 5940 testing cases are generated under two new sensor configurations where sensor T2 is removed and sensor T5 is added, respectively. The detection results under different sensor configurations are summarized in Table 12. It can be observed that in the nominal state, the trained CAEs can be perfectly applied to new sensor configurations with an F1 score of 1.0. However, in micrometeorite impact scenarios, the performance of the CAE when applied to new sensor configurations is inferior to that when applied to the default sensor configuration. Moreover, the CAE trained with fire scenarios has better performance than the one trained with micrometeorite impact scenarios when applied to either default or new sensor configurations. This can be explained by the simpler data variation (i.e., monotonous increase) of the temperature and pressure in fire scenarios. Based on the sample testing cases shown in Fig. 9, the detection results of trained CAEs under new sensor configurations are visualized in Fig. 15. It is shown that the CAEs after removing sensor T2 (Figs. 15a, 15c, and 15e) have comparable performance to the CAEs using the default sensor configuration (Figs. 10a, 10c, and 10e). However, the CAEs after adding sensor T5 produce false positives in micrometeorite impact scenarios (Figs. 15d). This is mainly due to the more complicated data variation of the temperature and pressure in micrometeorite impact scenarios. Furthermore, adding sensor T5 increases the input size of CAE, which may change the receptive field of CAE.

Table 12 Results under different sensor configurations

	Sensor configuration	True positive	True negative	False positive	False negative	Precision	Recall	F1 score
Nominal state CAE-1	Default	441	4839	0	0	1.00	1.00	1.00
	Remove T2	384	4236	0	0	1.00	1.00	1.00
	Add T5	401	5539	0	0	1.00	1.00	1.00
Micrometeorite impact CAE-11	Default	263	4730	105	182	0.71	0.59	0.65
	Remove T2	220	4086	133	181	0.62	0.55	0.58
	Add T5	182	5397	154	207	0.54	0.47	0.50
Fire CAE-6	Default	359	4739	91	91	0.80	0.80	0.80
	Remove T2	332	4137	78	73	0.81	0.82	0.81
	Add T5	303	5469	71	97	0.81	0.76	0.78

If the data-driven model is trained using the temperature-only or pressure-only data without the coupling, CAE has the capability of completely removing the sensor without replacing it with the virtual replica sensor, while AANN needs to be re-trained if the sensor is removed without replacement. In this regard, CAE can reduce the resources for data transmission, data storage, and data processing. The AANNs and CAEs trained with temperature-only data under the default sensor configuration are tested under the new sensor configuration where sensor T2 is removed. The performance of the CAEs with and without the replacement of sensor T2, as well as the AANNs without replacement, is summarized in Table 13. It is shown that the CAEs without replacement have comparable performance to the CAEs and AANNs with replacement. Therefore, CAE can maintain a similar performance with AANN but reduce the data and



(a) CAE-1, nominal state, remove sensor T2

(b) CAE-1, nominal state, add sensor T5

(c) CAE-11, micrometeorite impact, remove sensor T2

(d) CAE-11, micrometeorite impact, add sensor T5

(e) CAE-6, fire, remove sensor T2

(f) CAE-6, fire, add sensor T5

Fig. 15 Visualization of the detection results obtained by CAEs using new sensor configurations.

computation resources in deep-space habitats.

V. Conclusions

In this paper, an unsupervised learning approach based on CAEs is developed to detect sensor faults in smart extraterrestrial habitats. Several illustrative examples are demonstrated in different habitat scenarios simulated by HabSim, including the nominal state of the habitat, micrometeorite impact, and fire scenarios. The performance of AANN, CAE, and VAE is systematically compared. Based on the illustrative example results, the following conclusions can be drawn:

(1) It is shown that CAE and VAE have faster testing times than AANN, while AANN has a faster training time than

Table 13 Results for scenarios in which sensor T2 is removed with and without replacement.

	Data-driven model	Replacement of sensor T2	True positive	True negative	False positive	False negative	Precision	Recall	F1 score
Nominal state	CAE-1	No	162	1818	0	0	1.00	1.00	1.00
	CAE-1	Yes	162	1818	0	0	1.00	1.00	1.00
	AANN-3	Yes	162	1818	0	0	1.00	1.00	1.00
Micrometeorite impact	CAE-11	No	97	1783	38	62	0.72	0.61	0.66
	CAE-11	Yes	91	1787	34	68	0.73	0.57	0.64
	AANN-4	Yes	105	1765	56	54	0.65	0.66	0.66
Fire	CAE-6	No	104	1770	40	66	0.72	0.61	0.66
	CAE-6	Yes	101	1770	40	69	0.72	0.59	0.65
	AANN-1	Yes	125	1764	46	45	0.73	0.74	0.73

CAE and VAE. In the nominal state, CAE, VAE, and AANN have comparable performance, and all can achieve an F1 score of 1.0. However, in micrometeorite impact scenarios, CAE performs better than VAE and AANN, while in fire scenarios, AANN performs better than CAE and VAE. This is because AANN overfits the training data, while CAE has better generalization capability in micrometeorite impact scenarios. Moreover, CAE, VAE, and AANN produce higher F1 scores in fire scenarios than in micrometeorite impact scenarios. This can be explained by the simpler data variation, i.e., monotonous increase, of the temperature and pressure in fire scenarios.

(2) Utilizing the temperature-pressure coupling can improve the detection performance of CAE and AANN, as the coupling enhances the redundant information among sensors. However, even with the coupling, CAE and AANN perform better on the temperature data in micrometeorite impact scenarios, while they perform better on the pressure data in fire scenarios. This is because temperature and pressure have smaller decrease and increase rates in micrometeorite impact and fire scenarios, respectively.

(3) CAE and AANN have comparable performance when detecting multiple sensor failures. In the nominal state, both CAE and AANN can achieve an F1 score of 1.0 even with multiple faulty sensors. However, in micrometeorite impact and fire scenarios, with the increase in faulty sensors, the redundant information among sensors is reduced, which causes a decrease in the F1 scores of CAE and AANN.

(4) CAE has the capability of being applied to new sensor configurations without re-designing the network architecture and re-training the neural network. In the nominal state, CAE can achieve an F1 score of 1.0 when either removing sensors or adding sensors. CAE performs better in fire scenarios than in micrometeorite scenarios when applied to new sensor configurations. Additionally, CAE can completely remove the faulty sensor without replacing it with the virtual replica sensor, which reduces the data and computation resources in extraterrestrial habitats.

Acknowledgments

This research is funded by the National Aeronautics and Space Administration under award number 80NSSC19K1076. The authors sincerely thank the HabSim modelers at Resilient ExtraTerrestrial Habitats Institute (RETHi), including Adnan Shahriar, Leila Chebbo, Seungho Rhee, Murali Krishnan, and Kairui Hao.

References

[1] Jiang, H., van der Veek, B., Kirk, D., and Gutierrez, H., "Real-Time Estimation of Time-Varying Bending Modes Using Fiber Bragg Grating Sensor Arrays," *AIAA Journal*, Vol. 51, No. 1, 2013, pp. 178–185. https://doi.org/10.2514/1.J051702.

[2] Shelke, A., Uddin, A., and Yang, J., "Impact Identification in Sandwich Structures Using Solitary Wave-Supporting Granular Crystal Sensors," *AIAA Journal*, Vol. 52, No. 10, 2014, pp. 2283–2290. https://doi.org/10.2514/1.J052868.

[3] Topac, O. T., Sara Ha, S. Y., Chen, X., Gamble, L., Inman, D., and Chang, F.-K., "Hybrid Models for Situational Awareness of an Aerial Vehicle from Multimodal Sensing," *AIAA Journal*, Vol. 61, No. 1, 2023, pp. 305–314. https://doi.org/10.2514/1.J061926.

[4] Spirkovska, L., Iverson, D., Hall, D., Taylor, W., Patterson-Hine, A., Brown, B., Ferrell, B., and Waterman, R., "Anomaly Detection for Next-Generation Space Launch Ground Operations," *SpaceOps 2010 Conference*, AIAA, 2010, pp. 1–10. https://doi.org/10.2514/6.2010-2182.

[5] Du, Z., Jin, X., and Yang, Y., "Fault diagnosis for temperature, flow rate and pressure sensors in VAV systems using wavelet neural network," *Applied Energy*, Vol. 86, No. 9, 2009, pp. 1624–1631. https://doi.org/10.1016/j.apenergy.2009.01.015.

[6] Shang, L., and Liu, G., "Sensor and Actuator Fault Detection and Isolation for a High Performance Aircraft Engine Bleed Air Temperature Control System," *IEEE Transactions on Control Systems Technology*, Vol. 19, No. 5, 2011, pp. 1260–1268. https://doi.org/10.1109/TCST.2010.2076353.

[7] Yang, X.-B., Jin, X.-Q., Du, Z.-M., and Zhu, Y.-H., "A novel model-based fault detection method for temperature sensor using fractal correlation dimension," *Building and Environment*, Vol. 46, No. 4, 2011, pp. 970–979. https://doi.org/10.1016/j.buildenv.2010.10.030.

[8] Dey, S., Mohon, S., Pisu, P., and Ayalew, B., "Sensor Fault Detection, Isolation, and Estimation in Lithium-Ion Batteries," *IEEE Transactions on Control Systems Technology*, Vol. 24, No. 6, 2016, pp. 2141–2149. https://doi.org/10.1109/TCST.2016.2538200.

[9] Brunton, S. L., Nathan Kutz, J., Manohar, K., Aravkin, A. Y., Morgansen, K., Klemisch, J., Goebel, N., Buttrick, J., Poskin, J., Blom-Schieber, A. W., Hogan, T., and McDonald, D., "Data-Driven Aerospace Engineering: Reframing the Industry with Machine Learning," *AIAA Journal*, Vol. 59, No. 8, 2021, pp. 2820–2847. https://doi.org/10.2514/1.J060131.

[10] de Silva, B. M., Callaham, J., Jonker, J., Goebel, N., Klemisch, J., McDonald, D., Hicks, N., Nathan Kutz, J., Brunton, S. L., and Aravkin, A. Y., "Hybrid Learning Approach to Sensor Fault Detection with Flight Test Data," *AIAA Journal*, Vol. 59, No. 9, 2021, pp. 3490–3503. https://doi.org/10.2514/1.J059943.

[11] Dutta, A., McKay, M. E., Kopsaftopoulos, F., and Gandhi, F., "Multicopter Fault Detection and Identification via Data-Driven Statistical Learning Methods," *AIAA Journal*, Vol. 60, No. 1, 2022, pp. 160–175. https://doi.org/10.2514/1.J060353.

[12] Dutta, A., Niemiec, R., Kopsaftopoulos, F., and Gandhi, F., "Machine-Learning Based Rotor Fault Diagnosis in a Multicopter with Strain Data," *AIAA Journal*, Vol. 61, No. 9, 2023, pp. 4182–4194. https://doi.org/10.2514/1.J062399.

[13] Février, S., Nachar, S., Mathelin, L., Giordano, F., and Podvin, B., "Data-Driven Model to Predict Aircraft Vibration Environment," *AIAA Journal*, Vol. 61, No. 10, 2023, pp. 4610–4622. https://doi.org/10.2514/1.J062735.

[14] Jian, X., Zhong, H., Xia, Y., and Sun, L., "Faulty data detection and classification for bridge structural health monitoring via statistical and deep-learning approach," *Structural Control and Health Monitoring*, Vol. 28, No. 11, 2021, p. e2824. https://doi.org/10.1002/stc.2824.

[15] Tang, Z., Chen, Z., Bao, Y., and Li, H., "Convolutional neural network-based data anomaly detection method using multiple information for structural health monitoring," *Structural Control and Health Monitoring*, Vol. 26, No. 1, 2019, p. e2296. https://doi.org/10.1002/stc.2296.

[16] Zhang, H., Lin, J., Hua, J., Gao, F., and Tong, T., "Data Anomaly Detection for Bridge SHM Based on CNN Combined with Statistic Features," *Journal of Nondestructive Evaluation*, Vol. 41, No. 1, 2022, p. 28. https://doi.org/10.1007/s10921-022-00857-2.

[17] Zhang, Y., and Lei, Y., "Data Anomaly Detection of Bridge Structures Using Convolutional Neural Network Based on Structural Vibration Signals," *Symmetry*, Vol. 13, No. 7, 2021, p. 1186. https://doi.org/10.3390/sym13071186.

[18] Jana, D., Patil, J., Herkal, S., Nagarajaiah, S., and Duenas-Osorio, L., "CNN and Convolutional Autoencoder (CAE) based real-time sensor fault detection, localization, and correction," *Mechanical Systems and Signal Processing*, Vol. 169, 2022, p. 108723. https://doi.org/10.1016/j.ymssp.2021.108723.

[19] Bao, Y., Tang, Z., Li, H., and Zhang, Y., "Computer vision and deep learning–based data anomaly detection method for structural health monitoring," *Structural Health Monitoring*, Vol. 18, No. 2, 2019, pp. 401–421. https://doi.org/10.1177/1475921718757405.

[20] Mallak, A., and Fathi, M., "Sensor and Component Fault Detection and Diagnosis for Hydraulic Machinery Integrating LSTM Autoencoder Detector and Diagnostic Classifiers," *Sensors*, Vol. 21, No. 2, 2021. https://doi.org/10.3390/s21020433.

[21] Xu, J., Dang, D., Ma, Q., Liu, X., and Han, Q., "A novel and robust data anomaly detection framework using LAL-AdaBoost for structural health monitoring," *Journal of Civil Structural Health Monitoring*, Vol. 12, No. 2, 2022, pp. 305–321. https://doi.org/10.1007/s13349-021-00544-2.

[22] Zhang, Y., Tang, Z., and Yang, R., "Data anomaly detection for structural health monitoring by multi-view representation based on local binary patterns," *Measurement*, Vol. 202, 2022, p. 111804. https://doi.org/10.1016/j.measurement.2022.111804.

[23] Palmé, T., Fast, M., and Thern, M., "Gas turbine sensor validation through classification with artificial neural networks," *Applied Energy*, Vol. 88, No. 11, 2011, pp. 3898–3904. https://doi.org/10.1016/j.apenergy.2011.03.047.

[24] Pereira, M., and Glisic, B., "Detection and quantification of temperature sensor drift using probabilistic neural networks," *Expert Systems with Applications*, Vol. 213, 2023, p. 118884. https://doi.org/10.1016/j.eswa.2022.118884.

[25] Elnour, M., Meskin, N., and Al-Naemi, M., "Sensor data validation and fault diagnosis using Auto-Associative Neural Network for HVAC systems," *Journal of Building Engineering*, Vol. 27, 2020, p. 100935. https://doi.org/10.1016/j.jobe.2019.100935.

[26] Mao, J., Wang, H., and Billie F Spencer, J., "Toward data anomaly detection for automated structural health monitoring: Exploiting generative adversarial nets and autoencoders," *Structural Health Monitoring*, Vol. 20, No. 4, 2021, pp. 1609–1626. https://doi.org/10.1177/1475921720924601.

[27] Ba-Alawi, A. H., Vilela, P., Loy-Benitez, J., Heo, S., and Yoo, C., "Intelligent sensor validation for sustainable influent quality monitoring in wastewater treatment plants using stacked denoising autoencoders," *Journal of Water Process Engineering*, Vol. 43, 2021, p. 102206. https://doi.org/10.1016/j.jwpe.2021.102206.

[28] Loy-Benitez, J., Li, Q., Nam, K., and Yoo, C., "Sustainable subway indoor air quality monitoring and fault-tolerant ventilation control using a sparse autoencoder-driven sensor self-validation," *Sustainable Cities and Society*, Vol. 52, 2020, p. 101847. https://doi.org/10.1016/j.scs.2019.101847.

[29] Loy-Benitez, J., Heo, S., and Yoo, C., "Soft sensor validation for monitoring and resilient control of sequential subway indoor air quality through memory-gated recurrent neural networks-based autoencoders," *Control Engineering Practice*, Vol. 97, 2020, p. 104330. https://doi.org/10.1016/j.conengprac.2020.104330.

[30] Sun, J., Wang, X., Xiong, N., and Shao, J., "Learning Sparse Representation With Variational Auto-Encoder for Anomaly Detection," *IEEE Access*, Vol. 6, 2018, pp. 33353–33361. https://doi.org/10.1109/ACCESS.2018.2848210.

[31] Xu, H., Chen, W., Zhao, N., Li, Z., Bu, J., Li, Z., Liu, Y., Zhao, Y., Pei, D., Feng, Y., Chen, J., Wang, Z., and Qiao, H., "Unsupervised Anomaly Detection via Variational Auto-Encoder for Seasonal KPIs in Web Applications," *Proceedings of the 2018 World Wide Web Conference*, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 2018, p. 187–196. https://doi.org/10.1145/3178876.3185996.

[32] Shu, X., Bao, T., Zhou, Y., Xu, R., Li, Y., and Zhang, K., "Unsupervised dam anomaly detection with spatial–temporal variational autoencoder," *Structural Health Monitoring*, Vol. 22, No. 1, 2023, pp. 39–55. https://doi.org/10.1177/14759217211073301.

[33] Zhang, S., Ye, F., Wang, B., and Habetler, T. G., "Semi-Supervised Bearing Fault Diagnosis and Classification Using Variational Autoencoder-Based Deep Generative Models," *IEEE Sensors Journal*, Vol. 21, No. 5, 2021, pp. 6476–6486. https://doi.org/10.1109/JSEN.2020.3040696.

[34] He, A., and Jin, X., "Deep Variational Autoencoder Classifier for Intelligent Fault Diagnosis Adaptive to Unseen Fault Categories," *IEEE Transactions on Reliability*, Vol. 70, No. 4, 2021, pp. 1581–1595. https://doi.org/10.1109/TR.2021.3090310.

[35] Han, P., Ellefsen, A. L., Li, G., Holmeset, F. T., and Zhang, H., "Fault Detection With LSTM-Based Variational Autoencoder for Maritime Components," *IEEE Sensors Journal*, Vol. 21, No. 19, 2021, pp. 21903–21912. https://doi.org/10.1109/JSEN.2021.3105226.

[36] Zhang, M., Guo, J., Li, X., and Jin, R., "Data-Driven Anomaly Detection Approach for Time-Series Streaming Data," *Sensors*, Vol. 20, No. 19, 2020. https://doi.org/10.3390/s20195646.

[37] Du, Z., and Jin, X., "Detection and diagnosis for sensor fault in HVAC systems," *Energy Conversion and Management*, Vol. 48, No. 3, 2007, pp. 693–702. https://doi.org/10.1016/j.enconman.2006.09.023.

[38] Du, Z., Jin, X., and Wu, L., "Fault detection and diagnosis based on improved PCA with JAA method in VAV systems," *Building and Environment*, Vol. 42, No. 9, 2007, pp. 3221–3232. https://doi.org/10.1016/j.buildenv.2006.08.011.

[39] Chen, Y., and Lan, L., "Fault detection, diagnosis and data recovery for a real building heating/cooling billing system," *Energy Conversion and Management*, Vol. 51, No. 5, 2010, pp. 1015–1024. https://doi.org/10.1016/j.enconman.2009.12.004.

[40] Thiyagarajan, K., Kodagoda, S., Van Nguyen, L., and Ranasinghe, R., "Sensor Failure Detection and Faulty Data Accommodation Approach for Instrumented Wastewater Infrastructures," *IEEE Access*, Vol. 6, 2018, pp. 56562–56574. https://doi.org/10.1109/ACCESS.2018.2872506.

[41] Zhang, Y.-M., Wang, H., Wan, H.-P., Mao, J.-X., and Xu, Y.-C., "Anomaly detection of structural health monitoring data using the maximum likelihood estimation-based Bayesian dynamic linear model," *Structural Health Monitoring*, Vol. 20, No. 6, 2021, pp. 2936–2952. https://doi.org/10.1177/1475921720977020.

[42] Xu, X., Forde, M. C., Ren, Y., Huang, Q., and Liu, B., "Multi-index probabilistic anomaly detection for large span bridges using Bayesian estimation and evidential reasoning," *Structural Health Monitoring*, Vol. 0, No. 0, 2022, p. 14759217221092786. https://doi.org/10.1177/14759217221092786.

[43] Thiyagarajan, K., Kodagoda, S., Ulapane, N., and Prasad, M., "A Temporal Forecasting Driven Approach Using Facebook's Prophet Method for Anomaly Detection in Sewer Air Temperature Sensor System," *2020 15th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, 2020, pp. 25–30. https://doi.org/10.1109/ICIEA48937.2020.9248142.

[44] Schwabacher, M., Oza, N., and Matthews, B., "Unsupervised Anomaly Detection for Liquid-Fueled Rocket Propulsion Health Monitoring," *Journal of Aerospace Computing, Information, and Communication*, Vol. 6, No. 7, 2009, pp. 464–482. https://doi.org/10.2514/1.42783.

[45] Merrill, W. C., "Sensor failure detection for jet engines using analytical redundancy," *Journal of Guidance, Control, and Dynamics*, Vol. 8, No. 6, 1985, pp. 673–682. https://doi.org/10.2514/3.20041.

[46] Merrill, W. C., DeLaat, J. C., and Bruton, W. M., "Advanced detection, isolation, and accommodation of sensor failures - Real-time evaluation," *Journal of Guidance, Control, and Dynamics*, Vol. 11, No. 6, 1988, pp. 517–526. https://doi.org/10.2514/3.20348.

[47] Kobayashi, T., and Simon, D. L., "Evaluation of an Enhanced Bank of Kalman Filters for In-Flight Aircraft Engine Sensor Fault Diagnostics ," *Journal of Engineering for Gas Turbines and Power*, Vol. 127, No. 3, 2005, pp. 497–504. https://doi.org/10.1115/1.1850505.

[48] Richardson, N. O., Wong, E., and Melcher, K., "Analytical Redundancy Using Kalman Filters for Rocket Engine Sensor Validation," NASA technical report, NASA, Glenn Research Center, 10 2020. URL https://ntrs.nasa.gov/api/citations/20205000446/downloads/TM-20205000446.pdf.

[49] Mattern, D. L., Jaw, L. C., Guo, T.-H., Graham, R., and McCoy, W., "Using Neural Networks for Sensor Validation," NASA technical report, NASA, Lewis Research Center, 7 1998. URL https://ntrs.nasa.gov/api/citations/19980209658/downloads/19980209658.pdf.

[50] Moller, J. C., Litt, J. S., and Guo, T.-H., "Neural Network-Based Sensor Validation for Turboshaft Engines," NASA technical report, NASA, Lewis Research Center, 11 1998. URL https://ntrs.nasa.gov/api/citations/19990008960/downloads/19990008960.pdf.

[51] Maul, W. A., Melcher, K. J., Chicatelli, A. K., and Sowers, T. S., "Sensor Data Qualification for Autonomous Operation of Space Systems," NASA technical report, NASA, Glenn Research Center, 11 2006. URL https://ntrs.nasa.gov/api/citations/20070001482/downloads/20070001482.pdf.

[52] Csank, J. T., and Simon, D. L., "Sensor Data Qualification Technique Applied to Gas Turbine Engines," NASA technical report, NASA, Glenn Research Center, 12 2013. URL https://ntrs.nasa.gov/api/citations/20140011476/downloads/20140011476.pdf.

[53] Isermann, R., *Fault-tolerant components and control*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, Chap. 19, pp. 355–365. https://doi.org/10.1007/3-540-30368-5_19.

[54] Kramer, M. A., "Nonlinear principal component analysis using autoassociative neural networks," *AIChE Journal*, Vol. 37, 1991, pp. 233–243. https://doi.org/10.1002/aic.690370209.

[55] Kramer, M., "Autoassociative neural networks," *Computers and Chemical Engineering*, Vol. 16, No. 4, 1992, pp. 313–328. https://doi.org/10.1016/0098-1354(92)80051-A.

[56] Guo, T.-H., Saus, J., Lin, C.-F., and Ge, J.-H., "Sensor validation for turbofan engines using an autoassociative neural network," *Guidance, Navigation, and Control Conference*, AIAA, San Diego, CA, USA, 1996, pp. 1–8. https://doi.org/10.2514/6.1996-3926.

[57] Lu, P.-J., and Hsu, T.-C., "Application of Autoassociative Neural Network on Gas-Path Sensor Data Validation," *Journal of Propulsion and Power*, Vol. 18, No. 4, 2002, pp. 879–888. https://doi.org/10.2514/2.6013.

[58] Ogaji, S., Singh, R., and Probert, S., "Multiple-sensor fault-diagnoses for a 2-shaft stationary gas-turbine," *Applied Energy*, Vol. 71, No. 4, 2002, pp. 321–339. https://doi.org/10.1016/S0306-2619(02)00015-6.

[59] Shah, B., Sarvajith, M., Sankar, B., and Thennavarajan, S., "Multi-Auto Associative Neural Network based sensor validation and estimation for aero-engine," *2013 IEEE AUTOTESTCON*, Schaumburg, IL, USA, 2013, pp. 1–7. https://doi.org/10.1109/AUTEST.2013.6645076.

[60] Sadough Vanini, Z. N., Meskin, N., and Khorasani, K., "Multiple-Model Sensor and Components Fault Diagnosis in Gas Turbine Engines Using Autoassociative Neural Networks," *Journal of Engineering for Gas Turbines and Power*, Vol. 136, No. 9, 2014. https://doi.org/10.1115/1.4027215.

[61] Hines, J. W., Uhrig, R. E., and Wrest, D. J., "Use of Autoassociative Neural Networks for Signal Validation," *Journal of Intelligent and Robotic Systems*, Vol. 21, No. 2, 1998, pp. 143–154. https://doi.org/10.1023/A:1007981322574.

[62] Marseguerra, M., and Zoia, A., "The AutoAssociative Neural Network in signal analysis: II. Application to on-line monitoring of a simulated BWR component," *Annals of Nuclear Energy*, Vol. 32, No. 11, 2005, pp. 1207–1223. https://doi.org/10.1016/j.anucene.2005.03.005.

[63] Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, G., Cai, J., and Chen, T., "Recent advances in convolutional neural networks," *Pattern Recognition*, Vol. 77, 2018, pp. 354–377. https://doi.org/10.1016/j.patcog.2017.10.013.

[64] Yamashita, R., Nishio, M., Do, R. K. G., and Togashi, K., "Convolutional neural networks: an overview and application in radiology," *Insights into Imaging*, Vol. 9, No. 4, 2018, pp. 611–629. https://doi.org/10.1007/s13244-018-0639-9.

[65] Chen, S., Yu, J., and Wang, S., "One-dimensional convolutional auto-encoder-based feature learning for fault diagnosis of multivariate processes," *Journal of Process Control*, Vol. 87, 2020, pp. 54–67. https://doi.org/10.1016/j.jprocont.2020.01.004.

[66] Rai, K., Hojatpanah, F., Badrkhani Ajaei, F., and Grolinger, K., "Deep Learning for High-Impedance Fault Detection: Convolutional Autoencoders," *Energies*, Vol. 14, No. 12, 2021. https://doi.org/10.3390/en14123623.

[67] Yoo, S., Jung, J. H., Lee, J.-K., Shin, S. W., and Jang, D. S., "A Convolutional Autoencoder Based Fault Diagnosis Method for a Hydraulic Solenoid Valve Considering Unknown Faults," *Sensors*, Vol. 23, No. 16, 2023. https://doi.org/10.3390/s23167249.

[68] Wang, S., Wang, X., Zhang, L., and Zhong, Y., "Auto-AD: Autonomous Hyperspectral Anomaly Detection Network Based on Fully Convolutional Autoencoder," *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 60, 2022, pp. 1–14. https://doi.org/10.1109/TGRS.2021.3057721.

[69] Saeedi, J., and Giusti, A., "Semi-supervised visual anomaly detection based on convolutional autoencoder and transfer learning," *Machine Learning with Applications*, Vol. 11, 2023, p. 100451. https://doi.org/10.1016/j.mlwa.2023.100451.

[70] Thill, M., Konen, W., Wang, H., and Bäck, T., "Temporal convolutional autoencoder for unsupervised anomaly detection in time series," *Applied Soft Computing*, Vol. 112, 2021, p. 107751. https://doi.org/10.1016/j.asoc.2021.107751.

[71] Zhang, Y., Chen, Y., Wang, J., and Pan, Z., "Unsupervised Deep Anomaly Detection for Multi-Sensor Time-Series Signals," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 35, No. 2, 2023, pp. 2118–2132. https://doi.org/10.1109/TKDE.2021.3102110.

[72] Yin, C., Zhang, S., Wang, J., and Xiong, N. N., "Anomaly Detection Based on Convolutional Recurrent Autoencoder for IoT Time Series," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, Vol. 52, No. 1, 2022, pp. 112–122. https://doi.org/10.1109/TSMC.2020.2968516.

[73] Wang, B., and Yang, C., "Video Anomaly Detection Based on Convolutional Recurrent AutoEncoder," *Sensors*, Vol. 22, No. 12, 2022. https://doi.org/10.3390/s22124647.

[74] Goodfellow, I., Bengio, Y., and Courville, A., *Deep Learning*, MIT Press, 2016. URL http://www.deeplearningbook.org.

[75] Bank, D., Koenigstein, N., and Giryes, R., "Autoencoders," *arXiv e-prints*, 2020, arXiv:2003.05991. https://doi.org/10.48550/arXiv.2003.05991.

[76] LeCun, Y., Bengio, Y., and Hinton, G., "Deep learning," *Nature*, Vol. 521, No. 7553, 2015, pp. 436–444. https://doi.org/10.1038/nature14539.

[77] Ioffe, S., and Szegedy, C., "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," *arXiv e-prints*, 2015, arXiv:1502.03167. https://doi.org/10.48550/arXiv.1502.03167.

[78] Azimi, M., Lund, A., Fu, Y., Montoya, H., Luca Vaccino, M. K., Rhee, S., Chebbo, L., Shahriar, A., Wang, Z., Maghareh, A., and Dyke, S., "HabSim: A Modular Coupled Virtual Testbed for Simulating ExtraTerrestrial Habitat Systems," *AIAA Journal*, 2024, In review.

[79] Ripple, D. C., "NIST ITS-90 Thermocouple Database," , 1999. URL https://srdata.nist.gov/its90/main/.

[80] Balaban, E., Saxena, A., Bansal, P., Goebel, K. F., and Curran, S., "Modeling, Detection, and Disambiguation of Sensor Faults for Aerospace Applications," *IEEE Sensors Journal*, Vol. 9, No. 12, 2009, pp. 1907–1917. https://doi.org/10.1109/JSEN.2009.2030284.

[81] Li, D., Cai, Z., Qin, B., and Deng, L., "Signal frequency domain analysis and sensor fault diagnosis based on artificial intelligence," *Computer Communications*, Vol. 160, 2020, pp. 71–80. https://doi.org/10.1016/j.comcom.2020.05.034.

[82] Machin, J., Tucker, D., and Pearce, J., "A Comprehensive Survey of Reported Thermocouple Drift Rates Since 1972," *International Journal of Thermophysics*, Vol. 42, No. 10, 2021, p. 139. https://doi.org/10.1007/s10765-021-02892-z.

[83] Verhelst, J., Ham, G. V., Saelens, D., and Helsen, L., "Economic impact of persistent sensor and actuator faults in concrete core activated office buildings," *Energy and Buildings*, Vol. 142, 2017, pp. 111–127. https://doi.org/10.1016/j.enbuild.2017.02.052.