10.37965/jdmd.2026.1442

A Domain-Knowledge-Guided Modular Neural Network Framework for Gas Turbine Performance and Emission Prediction

Shazaib Ahsan[1], Tamiru Alemu Lemma[2], Jie Zhang[3] and Xihui Liang[1,*]

[1] Department of Mechanical Engineering, University of Manitoba, Winnipeg R3T 2N2, Manitoba, Canada.

[2] Department of Mechanical Engineering, Universiti Teknologi PETRONAS, Seri Iskandar 32610, Malaysia.

[3]School of Mechatronics Engineering, Southwest Petroleum University, Chengdu, China

*Corresponding author: xihui.liang@umanitoba.ca

Received 24 May 2026; Revised 05 August 2026; Accepted 25 August 2026; Published online 01 September 2026

Gas turbine performance and emission prediction are essential for condition monitoring, fault detection, predictive maintenance, and regulatory compliance, but conventional black-box models provide limited interpretability and poor subsystem-level traceability. This study proposes a domain-knowledge-guided modular neural network framework that decomposes gas turbine behavior into physically meaningful modules trained individually and integrated into a system-level model. Domain knowledge is embedded through modular decomposition, structured information flow, and physically meaningful inter-module connections, enabling measurement-level interpretation, module-level traceability, and modular updating. The framework is evaluated using two case studies. The framework is general at the modelling-procedure level, but the specific modular decomposition is case-specific and must be redefined according to the engine architecture, available measurements, measurement resolution, and target outputs. A high-bypass turbofan dataset, representative of a Pratt & Whitney PW-4056-like configuration, assesses detailed component-level modelling under a controlled single-realization measurement-noise perturbation. Under the controlled perturbation, the framework achieved a maximum mean absolute percentage error of approximately 2.62%, indicating stable prediction for the tested sensor-noise case. A real industrial predictive emission monitoring dataset evaluates adaptability using a reduced section-level structure based on available ambient, process, and emission measurements. Although initial integration showed cascading error, sequential fine-tuning reduced downstream emission mean absolute error by approximately 10.6% for CO and 39.2% for NOx. The fine-tuned framework matched monolithic Artificial Neural Network accuracy while providing intermediate predictions and subsystem-level error traceability, supporting interpretable gas turbine digital twins.

Keywords: Performance prediction; Emission prediction; Modular neural network; Domain knowledge integration; Data-driven modelling.

1. Introduction

Gas turbines are widely used in aerospace propulsion and industrial power-generation applications because of their high-power density, operational flexibility, and reliability. During operation, gas turbines experience continuously varying working conditions due to changes in operating settings, ambient temperature, pressure, humidity, load demand, and mission profile. These variations affect gas-path parameters, component efficiencies, fuel consumption, emissions, and overall system performance. In addition, as gas turbines accumulate operating hours, their performance and health condition inevitably change due to degradation, fouling, erosion, component faults, and maintenance actions [1]. Recent reliability studies have also shown that varying environmental conditions and operational settings can affect gas turbine degradation and failure behavior, highlighting the need for measurement-driven models that account for changing operating contexts [2]. Accurate interpretation of gas turbine measurements is therefore essential for safe operation, efficient energy conversion, condition monitoring, predictive maintenance, and regulation-compliant operation.

This requirement has motivated the development of measurement-driven gas turbine models capable of predicting performance-related quantities under variable operating conditions. Such measurement-based performance analysis has long been central to gas turbine diagnostics and health monitoring [3]. In engine health management, such models support continuous assessment of system behavior and help reduce reliance on offline or post-operation analysis alone [4,5]. However, gas turbines are multi-component systems in which changes or prediction errors in one section can propagate through downstream components and affect the overall system response [6]. Therefore, an effective modelling framework should provide not only accurate predictions but also subsystem-level traceability, measurement-level interpretation, and the ability to analyze how errors evolve across interconnected components.

The concept of digital twins has further increased the need for accurate, traceable, and updateable gas turbine models for performance assessment, degradation monitoring, and fault diagnosis [7–10]. Traditional gas turbine performance modelling has relied on thermodynamic relationships and component characteristic maps. Previous studies developed map-based and modular performance models for part-load prediction, aero-derivative gas turbines, and detailed component analysis [11–13]. Although these approaches are physically meaningful, their practical implementation can be limited by the availability of accurate component maps, which are often proprietary to manufacturers and unavailable to end users [14,15]. In addition, interpolation over limited map data can be unreliable when the engine operates over wide environmental ranges, transient conditions, or degraded states [16].

With the increasing availability of operational measurements, data-driven modelling has emerged as an effective approach for gas turbine performance prediction, emission estimation, fault detection, and condition monitoring. ANN and related machine-learning models have demonstrated strong capability in capturing nonlinear relationships between operating conditions and performance outputs using industrial, experimental, and simulated data [17–21]. Similar methods have been developed for gas-path fault detection and isolation, including neural-fuzzy networks, ANN-based diagnostic architectures, hybrid SVM-ANN frameworks, and dynamic prediction models for anomaly detection [22–27]. Although these approaches can provide accurate predictions, many of them treat the gas turbine as a monolithic input–output system or use domain models mainly to generate training data. This limits their ability to represent interactions among individual gas turbine components and to trace how prediction errors propagate through the system. This limitation is important because gas turbines contain interacting sections connected through gas-path flow, shaft-speed coupling, fuel addition, bypass/core flow interaction, and exhaust mixing. Therefore, although monolithic models may provide useful final-output accuracy, they often lack the structural transparency needed for measurement-level interpretation, subsystem-level error tracing, and modular model updating. This motivates modular modelling approaches that explicitly represent component-level relationships while incorporating gas turbine domain knowledge and maintaining consistency across interconnected modules.

In addition to performance prediction and fault diagnosis, predictive emission monitoring has become an important application of data-driven gas turbine modelling. Predictive emission monitoring systems estimate emission-related outputs such as carbon monoxide (CO) and nitrogen oxides (NOx) from available ambient and process measurements, providing an alternative to direct emission measurement when continuous monitoring is difficult or costly. The industrial predictive emission monitoring system (PEMS) dataset introduced by Kaya et al. [28] is particularly relevant because it reflects real operating variability, measurement uncertainty, partial sensor availability, and nonlinear emission behavior. However, many predictive emission models are still formulated as direct input–output mappings from measured variables to final emissions, which limits their ability to explain intermediate gas-path behavior or support subsystem-level traceability.

Recent studies have also shown the growing use of data-driven, data mining, and surrogate-modelling approaches in gas turbine engineering beyond direct performance prediction. For example, Yang et al. [29] developed a data-mining-based optimization framework for a multi-stage axial flow gas turbine, while Zeng et al. [30] applied a surrogate-model-based optimization method to the design of a heavy-duty gas turbine intake air volute. These studies demonstrate the increasing role of data-driven approximations in gas turbine component-level analysis and performance improvement.

In parallel, other studies have attempted to improve the engineering relevance of data-driven gas turbine models by incorporating domain knowledge into model development. Rather than relying only on direct black-box mappings between measured inputs and final outputs, these approaches use prior knowledge of gas turbine operation to guide feature selection, model structure, or the interpretation of predicted quantities. Liu and Huang [31] developed a data-driven gas turbine modelling approach that considered physical relationships in the learning process, while Lin et al. [32] proposed an aircraft engine thrust prediction model by combining domain knowledge with neural networks. Wang et al. [33] developed a knowledge-and-data jointly driven aeroengine gas-path performance assessment method that combines Fingerprint-based mechanistic knowledge with gas-path parameter deviation values to assess multi-component performance degradation. Bielski et al. [34] introduced a knowledge-guided deep-learning approach for gas turbine temporal dynamics by incorporating prior knowledge about permissible system states into the learning process. Chen et al. [35] proposed a data-knowledge hybrid gas-path diagnosis method by integrating physical model-based gas path analysis with ontology-based fault knowledge reasoning. Mo et al. [36] further proposed a physical-embedded data-driven method in which aeroengine domain knowledge is used to design the network structure and regulate internal information flow. He et al. [37] also integrated data-driven modelling with mechanistic knowledge for aeroengine module performance evaluation, showing the importance of module-level interpretation in engine health assessment. These studies indicate that data-driven gas turbine models can benefit from incorporating system-level or module-level knowledge rather than treating the engine as an entirely unstructured input-output system.

However, several gaps remain. First, existing hybrid, knowledge-guided, or physical-embedded studies often use domain knowledge to guide temporal response learning, support knowledge-based diagnosis, fuse physical and data-driven information, or evaluate component/module performance degradation. Fewer studies explicitly preserve and evaluate intermediate gas-path or section-level predictions throughout an integrated modular neural-network structure. Second, when separately trained component- or section-level models are connected, downstream modules receive predicted upstream quantities rather than the measured quantities used during independent training. This distributional difference can cause prediction errors to propagate and accumulate across the modular system, yet cascading error is rarely quantified or explicitly reduced. Third, limited attention has been given to whether the same modular modelling strategy can be adapted across different levels of measurement availability, ranging from a fully observable simulated turbofan to a real industrial PEMS application with substantially fewer section-level measurements.

To address these limitations, this study proposes a domain-knowledge-guided modular neural network framework for gas turbine performance and emission prediction. The proposed framework models physically meaningful modules separately and then integrates them according to known gas-path and inter-component couplings. In this work, domain knowledge is incorporated through modular decomposition, structured information flow, physically meaningful input-output selection, shaft-speed-related coupling, fuel-addition logic, and bypass/core flow representation where applicable. Artificial Neural Networks (ANN) are used as differentiable surrogate models because they allow system-level fine-tuning after module integration. This fine-tuning process is used to reduce cascading error while preserving intermediate module-level predictions and system interpretability.

The proposed framework is evaluated using two complementary gas turbine case studies. The first case study uses a simulated high-bypass turbofan engine dataset representative of a Pratt & Whitney PW-4056-like configuration [38]. This case provides detailed component-level modular representation and is used to evaluate the full modular structure, system integration, cascading-error behavior, and robustness under controlled measurement-noise perturbation. The second case study uses a real industrial PEMS dataset for CO and NOx prediction [28]. Because this dataset provides only a reduced set of section-level measurements, the modular structure is redefined into cold-section, hot-section, and emission modules using the available ambient, process, and emission measurements. The real-data case is also used to compare the proposed framework with monolithic ANN and optimized machine-learning and deep-learning baselines, allowing the trade-off between final-output accuracy, interpretability, and modular adaptability to be examined. Thus, the proposed framework should be understood as a general domain-knowledge-guided modular modelling procedure, not as a single fixed module topology that can be transferred unchanged across all gas turbine layouts or sensor configurations.

The main contributions of this work can be summarized as follows:

- A domain-knowledge-guided modular neural network framework is developed for gas turbine performance and emission prediction, enabling structured modelling beyond monolithic black-box prediction.
- A component integration and sequential fine-tuning strategy is introduced to reduce cascading prediction error in modular gas turbine models while preserving intermediate-state prediction and subsystem-level traceability.
- The proposed framework is evaluated using both a full simulated turbofan dataset and a real industrial predictive emission monitoring dataset, demonstrating applicability across different measurement configurations and modelling resolutions.
- A comparative evaluation is conducted against monolithic ANN and optimized machine-learning baselines to quantify the trade-off between final-output accuracy, modular interpretability, and adaptability under different modelling resolutions.

The remainder of this paper is organized as follows. Section 2 presents the proposed domain-knowledge-guided modular modelling framework, including component modelling, system integration, and sequential fine-tuning. Section 3 describes case studies and experimental setups, preprocessing procedure, performance metrics, and benchmark models. Section 4 presents the results for the simulated turbofan case, the controlled measurement-noise perturbation test, real PEMS evaluation, and baseline comparisons. Section 4 also discusses the implications, limitations, and potential extensions of the proposed framework, followed by conclusions in Section 5.

2. Proposed Domain-Knowledge-Guided Modular Modelling Framework

This section presents the proposed domain-knowledge-guided modular neural network framework for gas turbine performance and emission prediction. The framework is designed to retain the gas-turbine system structure while using ANN to learn nonlinear relationships from measured operating data. Instead of developing a single monolithic model that directly maps all available measurements to final outputs, the proposed framework decomposes the gas turbine into physically meaningful modules, trains each module individually, integrates the trained modules according to known gas-path connections, and then fine-tunes the integrated system model to reduce cumulative prediction error.

The overall framework is illustrated in Fig. 1. Gas turbine domain knowledge and available measurements are first used to define the modular model structure. This step determines the modelling resolution, the input-output variables of each module, and the connections between modules. Each module is then developed independently using an ANN model, allowing the local behavior of each gas turbine section to be learned before system-level integration. The independently trained ANN modules are subsequently connected to form the integrated modular system.

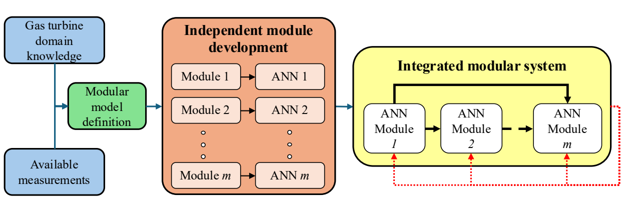


Fig. 1. Overall framework of the proposed domain-knowledge-guided modular modelling approach.

In the integrated modular system shown in Fig.1, solid arrows between adjacent ANN modules represent the sequential flow, where the output of an upstream module is passed to a downstream module when physically appropriate. The upper solid black connection represents non-adjacent cross-component coupling, such as shaft-speed or shared-variable relationships between modules that are not directly adjacent in the gas-path sequence. The red dotted arrows represent sequential fine-tuning updates driven by the combined module-level and system-level loss. These fine-tuning updates allow the connected ANN modules to adjust after integration, reducing the mismatch that can occur when downstream modules receive predicted upstream quantities instead of measured upstream variables.

It should be noted that the proposed framework is not intended to impose one fixed module structure on all gas turbine systems. The generality of the proposed framework lies in the modular decomposition, integration, and fine-tuning procedure, whereas the specific module boundaries, input-output definitions, and modelling resolution are case-dependent. Rather, it provides a general modelling procedure in which the module resolution is defined according to the engine configuration, available measurements, and target outputs. When detailed station-level gas-path measurements are available, as in the simulated high-bypass turbofan case, the framework can be implemented using component-level modules such as the fan, compressors, combustor, turbines, bypass section, and final system-output module. When only reduced process measurements are available, as in the industrial PEMS case, the same framework is applied at a coarser section level using cold-section, hot-section, and emission-output modules. Therefore, the two case studies are not intended to represent identical module layouts, but to demonstrate how the same domain-knowledge-guided modelling strategy can be adapted to different gas turbine configurations and measurement resolutions. The specific module definitions used for the simulated turbofan and industrial PEMS case studies are described in Section 3.

2.1. Modular component model formulation

The proposed framework begins by decomposing the gas turbine system into physically meaningful modules. Each module represents a component or system section whose behavior can be described using available measurements. The purpose of this decomposition is to avoid treating the gas turbine as a single unstructured input-output system and instead preserve the expected relationships between different sections of the engine.

In this study, the module boundaries are not selected only by dividing the engine into arbitrary data groups. They are defined using a criteria-based approach that combines gas turbine domain knowledge with the measurements available for training. The purpose is not to obtain a mathematically unique or globally optimal module split, because such a split depends on the engine layout, sensor availability, and target outputs. Instead, the objective is to define modules that are physically meaningful, trainable from available data, and useful for preserving intermediate system information.

The module-boundary decision criteria used in this work are as follows. First, a module boundary is placed where the dominant physical process changes, such as compression, combustion, expansion, bypass-flow passage, or emission formation. Second, each module must have measurable or inferable input and output variables so that it can be trained and evaluated independently. Third, the direction of information transfer between modules must follow the expected gas-path or system-level causality, so that upstream outputs are passed to downstream modules only when they represent physically meaningful inputs. Fourth, known inter-component couplings, such as shaft-speed-related coupling or bypass/core flow interaction, are retained when the corresponding variables are available. Fifth, the selected module outputs should provide intermediate information that is useful for subsystem-level traceability, rather than only serving as hidden numerical features.

The proposed framework is also different from a general modular neural network and from conventional physics-informed machine learning. In a general modular neural network, the division into subnetworks may be based mainly on computational convenience, feature grouping, or empirical architecture design. In the present framework, the modules are defined using gas turbine domain knowledge, including physical process boundaries, gas-path direction, known inter-component couplings, and available measurement locations. Therefore, the module outputs are intended to retain physically meaningful intermediate gas-path or section-level information, rather than serving only as hidden numerical features. This is related to recent physical-embedded aeroengine modelling studies, where domain knowledge is used to guide network structure and information flow [36], but the present work further focuses on independently trained modules, system-level integration, cascading-error analysis, and sequential fine-tuning. The framework is also different from physics-informed neural networks or physics-informed machine learning approaches that usually introduce governing equations, residual constraints, or physical laws directly into the loss function. In this work, domain knowledge is used primarily to guide modular decomposition, input-output selection, information flow, and sequential fine-tuning. Therefore, the proposed approach can be applied even when complete governing equations or proprietary component maps are unavailable, while still preserving a physically interpretable modular structure.

Based on these criteria, the simulated high-bypass turbofan case is represented using detailed component-level modules because station-level gas-path quantities are available for the fan, compressors, combustor, turbines, bypass section, and final system-output stage. In contrast, the industrial PEMS case is represented using reduced section-level modules because the dataset contains ambient, process, power-related, and emission measurements, but not complete station-level component measurements. Therefore, the reduced cold-section, hot-section, and emission-output modules are guided by the same criteria, but implemented at a coarser measurement resolution. This explains why the two case studies use different modular structures while following the same domain-knowledge-guided modelling procedure.

Each module is represented using a feedforward ANN. For a module m, the model can be generally expressed as:

$$\hat{\boldsymbol{y}}_{\boldsymbol{m}} = f_m(\boldsymbol{x}_m; \theta_m) \tag{1}$$

where $\boldsymbol{x}_m$ is the selected input vector for module m, $\hat{\boldsymbol{y}}_{\boldsymbol{m}}$ is the predicted output vector, f_m is the ANN mapping, and θ_m represents the trainable parameters of the model. This formulation allows each module to learn its own specific nonlinear relationship while preserving the modular structure of the gas turbine system.

Each module is first trained independently using the corresponding module-level input-output data. To improve generalization and reduce overfitting, L2 regularization is included in the training objective. The component-level loss function is defined as:

$$\mathcal{L}_m = \frac{1}{N}\sum_{i=1}^{N}\left|\boldsymbol{y}_m^{(i)} - \hat{\boldsymbol{y}}_m^{(i)}\right| + \lambda_r \|\boldsymbol{\theta}_m\|_2^2 \tag{2}$$

where $N$is the number of training samples, $y_m^{(i)}$is the measured output of module m, $\hat{\boldsymbol{y}}_m^{(i)}$ is the corresponding predicted output, θ_m represents the trainable parameters of the module, and λ_r controls the strength of L2 regularization.

Training the modules independently provides two benefits. First, each module learns the local behavior of its corresponding gas turbine section before being affected by errors from other modules. Second, the prediction performance of each module can be evaluated separately, which provides intermediate-level interpretability that is not available in a monolithic input-output model.

The ANN architecture and training hyperparameters were selected through a validation-based hyperparameter optimization process. The tuned parameters included the number of hidden layers, number of neurons per hidden layer, batch size, and learning rate. Rectified Linear Unit (ReLU) activation was used in the hidden layers to maintain a consistent ANN structure across modules. The output-layer activation was selected according to the output scaling and model type; the exact settings are reported in Table 4 and Appendix A. The best-performing model for each module was selected based on validation loss and then saved for system-level integration. The detailed search ranges, selected architectures, and final training settings for each case study are reported in Section 3.

2.2. System-level integration and sequential fine-tuning

After the component-level ANN models are trained, they are integrated to form a system-level modular model. The integration follows the expected gas-path sequence and inter-component relationships defined during the modular formulation stage. In this process, the predicted output of an upstream module is passed to the downstream module when such a connection is physically meaningful. Therefore, the integrated model does not treat the gas turbine as a single direct input-output mapping, but as a connected sequence of interacting module-level models.

For a gas turbine system consisting of M modules, the integrated modular prediction can be generally expressed as:

$$\hat{\boldsymbol{y}}_m = f_m\left(\boldsymbol{x}_m, \hat{\boldsymbol{y}}_{up,m}; \boldsymbol{\theta}_m\right), \quad m = 1,2,\dots,M \tag{3}$$

where $\hat{\mathbf{y}}_m$ is the predicted output of module m, $\mathbf{x}_m$ is the external or directly available input vector for that module, $\hat{\boldsymbol{y}}_{up,m}$ represents the predicted output received from upstream module or modules, and $\boldsymbol{\theta}_m$ represents the trained parameters of the module. For the first module in a sequence, $\hat{\boldsymbol{y}}_{up,m}$ is not used because no upstream module exists. Equation (3) describes the general integration rule. The case-specific connected forms for the simulated turbofan and industrial PEMS datasets are provided in Sections 3.1 and 3.2, respectively, to show exactly which predicted variables are passed between modules in each implementation.

Although the independently trained modules can provide accurate component-level predictions, direct integration does not always guarantee the best system-level performance. During independent component training, a downstream module may be trained using measured upstream variables. However, during system-level integration, those measured upstream variables are replaced by predicted outputs from preceding modules. As a result, small upstream prediction errors can influence the inputs of downstream modules and may accumulate through the connected structure. Therefore, the initially integrated modular model is treated as an intermediate model rather than the final optimized system model.

The layer-by-layer propagation of module error can be understood from the difference between independent module training and integrated prediction. During independent training, module i learns a mapping from measured or reference input x_i to measured or reference output y_i. After system integration, however, the same module receives an input $\hat{x}_i$ that contains predicted upstream quantities. Therefore, the error of module i can be viewed conceptually as: $e_i = f_i(\hat{x}_i) - y_i = [f_i(\hat{x}_i) - f_i(x_i)] + [f_i(x_i) - y_i]$ where the first term represents the error caused by using predicted upstream inputs and the second term represents the local approximation error of the module itself. If a downstream module is sensitive to the upstream variable, a small upstream error can change the downstream input state and increase the downstream output error. This downstream error can then become part of the input uncertainty for the next module, causing error accumulation across the connected modular structure. Thus, the notation $\hat{y}_i + e_i$ in Fig. 2 is used only as a schematic representation of predicted output with associated prediction error, rather than as a separate algebraic input added to the model.

The cascading-error mechanism and the sequential fine-tuning workflow are illustrated schematically in Fig. 2. As shown in Fig. 2(a), the module models are first trained independently using their corresponding input-output pairs. After integration, downstream modules receive predicted upstream outputs rather than the measured upstream quantities used during independent training, as illustrated in Fig. 2(b). Consequently, upstream prediction errors may propagate through the connected structure and affect downstream module predictions. Sequential fine-tuning is therefore applied to the integrated model, as shown in Fig. 2(c). In each stage, one module is set as trainable while the remaining modules are frozen; Module 2 is shown as an example of the active module.

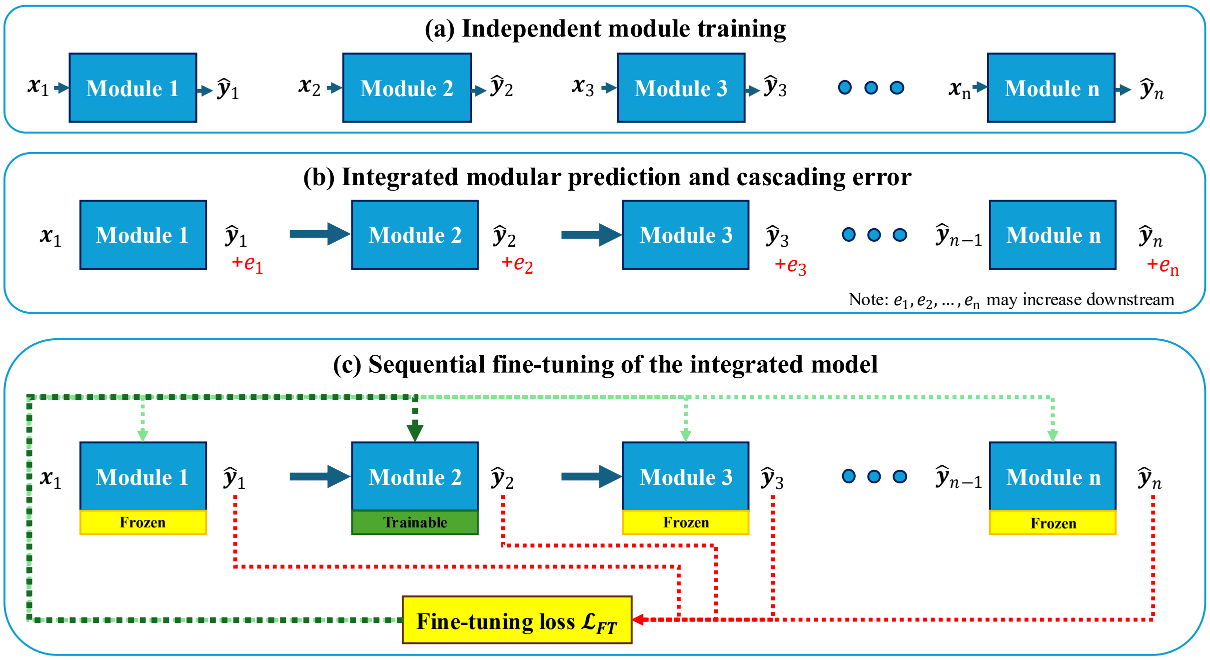


Fig. 2. Schematic illustration of independent module training, integrated prediction with cascading error, and sequential fine-tuning in the proposed modular ANN framework. In panel (c), Module 2 is shown as the active trainable module for illustration.

To address this issue, the proposed framework applies sequential fine-tuning after system integration. The purpose of fine-tuning is to adjust the connected modules so that they operate more consistently within the integrated system. Instead of optimizing only the final system output, the fine-tuning process considers both intermediate module-level predictions and final output predictions. This helps reduce accumulated error while preserving the interpretability of the intermediate module outputs. The system-level fine-tuning objective is defined as:

$$\mathcal{L}_{FT} = \sum_{m=1}^{M} \alpha_m \left[\frac{1}{N}\sum_{i=1}^{N} \left|\boldsymbol{y}_m^{(i)} - \hat{\boldsymbol{y}}_m^{(i)}\right|\right] + \alpha_{sys}\left[\frac{1}{N}\sum_{i=1}^{N}\left|\boldsymbol{y}_{sys}^{(i)} - \hat{\boldsymbol{y}}_{sys}^{(i)}\right|\right] + \lambda_{FT}\|\boldsymbol{\theta}_T\|_2^2 \tag{4}$$

where $\mathcal{L}_{FT}$ is the total fine-tuning loss, $\mathbf{y}_m^{(i)}$ and $\hat{\mathbf{y}}_m^{(i)}$ are the measured and predicted outputs of module m, $\mathbf{y}_{sys}^{(i)}$ and $\hat{\mathbf{y}}_{sys}^{(i)}$ are the measured and predicted final system-level outputs, α_m and α_{sys} are weighting coefficients for module-level and system-level losses, respectively, $\boldsymbol{\theta}_T$ represents the trainable parameters during fine-tuning, and λ_{FT} controls the L2 regularization strength during fine-tuning.

The loss terms are calculated using normalized variables so that outputs with larger numerical magnitudes do not dominate the optimization. This is important because gas turbine variables such as pressure, temperature, thrust, fuel consumption, and emissions can have different physical units and numerical ranges. In this study, the same weighting strategy was used for all sequential fine-tuning stages. Unit weighting coefficients were used, with $\alpha_m = 1$ for the module-level loss terms and $\alpha_{sys} = 1$ for the system-level loss term. This setting was selected as a neutral weighting strategy to avoid additional manual tuning for individual modules. The purpose was to allow each fine-tuning stage to reduce the accumulated system-level error while keeping the intermediate module outputs close to their corresponding physical targets.

The fine-tuning is performed sequentially. In each stage, one module is set to trainable and updated while the remaining modules are kept fixed. During this process, the selected architecture of each module is not changed; the number of hidden layers, number of neurons, activation functions, and module input-output definitions remain fixed. Only the trainable parameters of the active module, such as the Dense-layer weights and biases, are updated. This allows the integrated model to correct local sources of accumulated error without unnecessarily disturbing all previously trained modules at once. After the sequential module-wise updates, a final fine-tuning step is applied to the complete integrated model to improve overall system consistency. The best fine-tuned model is selected using validation loss. The case-specific fine-tuning order, trainable/frozen module settings, optimizer settings, and stopping criterion used in the two case studies are described in Section 3.3.

It should be noted that sequential fine-tuning is not a second architecture search or a re-design of the modular structure. Instead, it adapts the connected modules to the conditions encountered during system-level prediction, where downstream modules receive predicted upstream quantities rather than measured upstream quantities. As a result, the error of some individual module outputs may slightly increase if this adjustment helps reduce accumulated downstream error and improve the final system-level outputs. Therefore, the fine-tuned model is evaluated using both module-level errors and final system-level errors, with key system-level outputs used as the primary indicators of integrated model performance.

This two-stage development process, consisting of independent module training followed by system-level fine-tuning, allows the framework to preserve component-level interpretability while improving integrated prediction performance. The independent training stage ensures that each module learns the behavior of its corresponding gas turbine section, while the fine-tuning stage improves the coordinated behavior of the connected system.

2.3. Overall workflow of the proposed framework

Overall, the proposed framework follows a two-stage modelling strategy. In the first stage, physically meaningful modules are defined and trained independently using validation-based hyperparameter optimization. In the second stage, the trained modules are connected as per the gas-path and inter-component relationships and refined through sequential fine-tuning. The final model is evaluated at both the module-output level and the final system-output level, allowing the framework to combine prediction accuracy with modular interpretability.

3. Case studies and experimental setup

This section describes the datasets, modular model implementation, preprocessing procedure, training setup, benchmark models, and evaluation metrics used to assess the proposed framework. Two gas turbine case studies are considered to evaluate the framework under different configurations and modelling resolutions. Because the available measurements and target outputs differ between the two cases, the modular decomposition is redefined separately for each dataset rather than transferred unchanged from one case to the other. The first case study uses a simulated high-bypass turbofan dataset representative of a Pratt & Whitney PW-4056-like engine [38]. This dataset is used to implement a detailed modular structure. The second case study uses a real industrial PEMS dataset for CO and NOx prediction [28]. The modular structure is implemented at a coarser resolution using cold-section, hot-section, and emission modules.

Both case studies follow the general procedure described in Section 2. The specific module definitions, dataset characteristics, preprocessing procedure, training settings, and evaluation metrics are described below. The two case studies used in this work are given in Table 1.

Table 1. Summary of two case studies used for framework evaluation.

Case	Modular	Main	Main

10.37965/jdmd.2026.1442

study	resolution	prediction outputs	purpose
Simulated turbofan engine	Detailed component-level modules	Thrust, thrust-specific fuel consumption (TSFC), Overall pressure ratio (OPR)	Evaluate full modular framework, integration, fine-tuning, and noise robustness
Industrial PEMS dataset	Reduced section-level modules	CO and NOx emissions	Evaluate applicability under real operating data and baselines comparison

3.1. Case Study 1: Simulated high-bypass turbofan dataset

The simulated dataset was generated using the Gas Turbine Simulation Program (GSP), which is a gas turbine performance simulation environment used to model engine operating behaviour under controlled operating and environmental conditions. The GSP model used in this study was developed and calibrated in our previous work [38] for a high-bypass turbofan configuration inspired by the Pratt & Whitney PW-4056. The simulation dataset was used because detailed station-wise measurements from real aero-engines are difficult to obtain and are often proprietary. In contrast, the GSP model provides access to internal gas-path variables while preserving thermodynamic consistency across the engine stations. Additional details can be found in our previous study [38].

The healthy operating dataset used in this study was generated by varying the main environmental and operating parameters over a predefined envelope. The simulated conditions included altitude from 0 to 3 km, Mach number from 0 to 0.6, ambient temperature from -15^{o}C to 45^{o}C, and relative humidity from 0% to 100%. For each steady-state operating point, station-wise gas-path variables were recorded, including component temperatures, pressures, efficiencies, shaft speeds, fuel flow, thrust, TSFC, and OPR. These variables allowed the simulated turbofan case to be used for detailed component-level modular modelling, system integration, and error-propagation analysis. Although the simulated dataset also contains different fault-condition data, only healthy steady-state operating points were used in the present study because the objective was to evaluate the proposed modular framework for nominal performance prediction and system-level integration. The complete healthy dataset used in this study contained 5,122 steady-state samples, which were divided into training, validation, and testing subsets using the 70/15/15 split described in Section 3.3.

The simulated engine structure follows the expected gas-path sequence of a high-bypass turbofan. The inlet air first passes through the fan, after which the flow is divided into bypass and core streams. The bypass stream is modelled through the bypass section, while the core stream passes through the low-pressure compressor (LPC), high-pressure compressor (HPC), combustor, high-pressure turbine (HPT), and low-pressure turbine (LPT). Fuel flow is introduced at the combustor module, where the compressed air is heated before entering the turbine section. The HPT is associated with the high-pressure spool and is coupled with the HPC, while the LPT is associated with the low-pressure spool and is coupled with the fan and LPC. The outputs from the bypass and core streams are finally used by the system-output module to predict overall engine performance. A schematic of a high bypass turbofan engine is provided in Fig. 3.

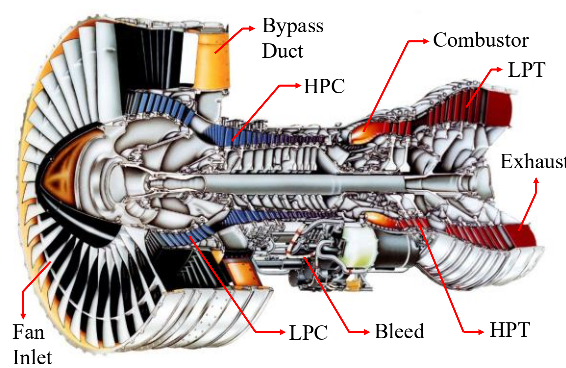


Fig. 3. Schematic of a high bypass turbofan engine configuration. Reproduced with permission taken from RTX Corporation, Pratt & Whitney division.

The modular structure used for this case study is shown in Fig. 4. The figure shows the main gas-path sequence through the fan, LPC, HPC, combustor, HPT, and LPT, together with the parallel bypass branch. The blue and yellow module colors indicate low-pressure-shaft-related and high-pressure-shaft-related sections, respectively. The tapered compressor and turbine shapes represent compression and expansion processes. The system-output module combines the downstream core-stream and bypass-stream information to predict thrust, TSFC, and OPR. The operating and ambient-condition variables are used as the initial inputs to the fan module. The fan output is then passed to the bypass section and the core-flow modules according to the physical flow division. The LPC and HPC modules represent the compression process in the core stream, while the combustor module represents fuel addition and the corresponding rise in gas temperature. The HPT and LPT modules represent the expansion process and mechanical coupling through the high-pressure and low-pressure shafts. The system-output module combines the relevant downstream core and bypass information to predict thrust, TSFC, and OPR.

The input and output variables of each module were selected according to the physical function of the corresponding gas turbine section and the available simulated measurements. This allows the proposed framework to preserve intermediate gas-path quantities rather than directly mapping the initial operating conditions to final engine outputs. Table 2 summarizes the module-level input-output structure used for the simulated turbofan case.

To make the integration procedure explicit, the connected modular prediction process for the simulated turbofan case can be written as follows. The entry input vector is defined as $x_0 = [Z, Mach, T_{amb}, P_{amb}, RH]$, where Z is altitude, $Mach$ is Mach number, T_{amb} is ambient temperature, P_{amb} is ambient pressure, and RH is relative humidity. The fan and bypass-section modules are evaluated from the inlet/ambient-condition variables.

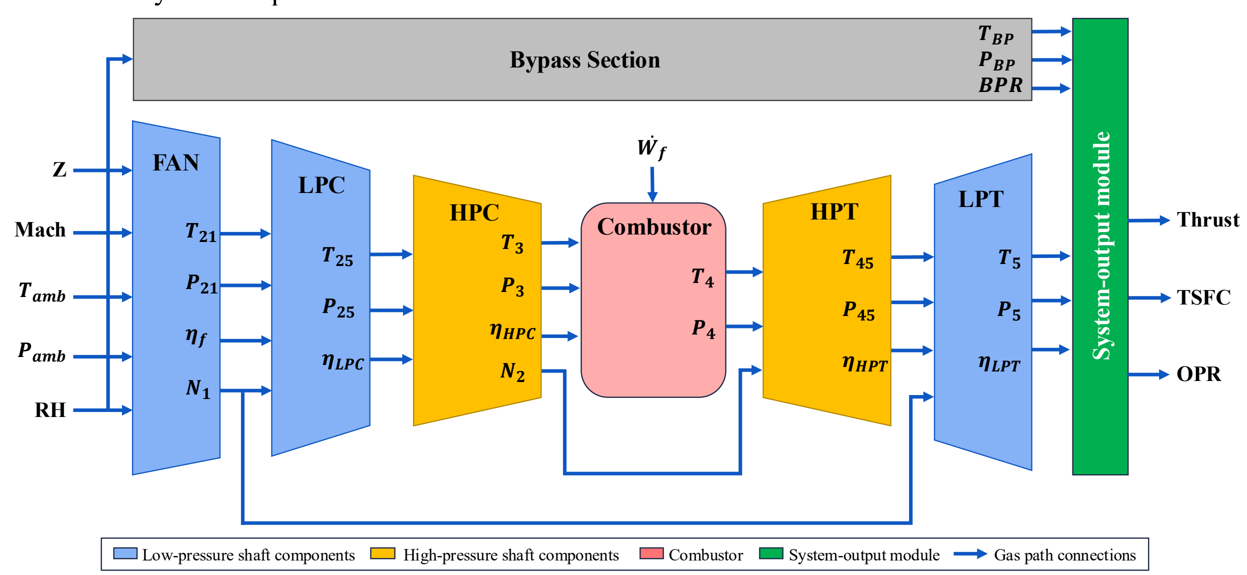


Fig. 4. Proposed modular system structure for the simulated high-bypass turbofan case.

Table 2. Input and output variables of the simulated turbofan modular structure.

Component	Inputs	Outputs
Fan	Z: Altitude $Mach$: Mach number T_{amb}: Ambient Temperature	T_{21}: Fan outlet temperature P_{21}: Fan outlet pressure η_f: Fan efficiency

	P_{amb}: Ambient Pressure RH: Relative Humidity	N_1: Low-pressure shaft rpm
LPC	T_{21}: Fan outlet temperature P_{21}: Fan outlet pressure η_f: Fan efficiency N_1: Low-pressure shaft rpm	T_{25}: LPC outlet temperature P_{25}: LPC outlet pressure η_{LPC}: LPC efficiency
HPC	T_{25}: LPC outlet temperature P_{25}: LPC outlet pressure η_{LPC}: LPC efficiency	T_3: HPC outlet temperature P_3: HPC outlet pressure η_{HPC}: HPC efficiency N_2: High-pressure shaft rpm
Combustor	T_3: HPC outlet temperature P_3: HPC outlet pressure η_{HPC}: HPC efficiency $\dot{W}_f$: Fuel flow rate	T_4: Combustor outlet temperature P_4: Combustor outlet pressure
HPT	T_4: Combustor outlet temperature P_4: Combustor outlet pressure N_2: High-pressure shaft rpm	T_{45}: HPT outlet temperature P_{45}: HPT outlet pressure η_{HPT}: HPT efficiency
LPT	T_{45}: HPT outlet temperature P_{45}: HPT outlet pressure η_{HPT}: HPT efficiency N_1: Low-pressure shaft rpm	T_5: LPT outlet temperature P_5: LPT outlet pressure η_{LPT}: LPT efficiency
Bypass section	Z: Altitude $Mach$: Mach number T_{amb}: Ambient Temperature P_{amb}: Ambient Pressure RH: Relative Humidity	T_{BP}: Bypass section temperature P_{BP}: Bypass section pressure BPR: Bypass ratio
System-output module	T_{BP}: Bypass temperature P_{BP}: Bypass pressure BPR: Bypass ratio T_5: LPT outlet temperature P_5: LPT outlet pressure η_{LPT}: LPT efficiency	$Thrust$ $TSFC$: Thrust-specific fuel consumption OPR: Overall pressure ratio

$$
\begin{gathered}
\hat{\boldsymbol{y}}_f = f_f(\boldsymbol{x}_0), \\
\hat{\boldsymbol{y}}_{LPC} = f_{LPC}(\hat{\boldsymbol{y}}_f), \\
\hat{\boldsymbol{y}}_{HPC} = f_{HPC}(\hat{\boldsymbol{y}}_{LPC}), \\
\hat{\boldsymbol{y}}_{comb} = f_{comb}(\hat{\boldsymbol{y}}^*{}_{HPC}, \dot{W}_f), \\
\hat{\boldsymbol{y}}_{HPT} = f_{HPT}(\hat{\boldsymbol{y}}_{comb}, \hat{N}_2), \\
\hat{\boldsymbol{y}}_{LPT} = f_{LPT}(\hat{\boldsymbol{y}}_{HPT}, \hat{N}_1), \\
\hat{\boldsymbol{y}}_{BP} = f_{BP}(\boldsymbol{x}_0), \\
\hat{\boldsymbol{y}}_{sys} = f_{sys}(\hat{\boldsymbol{y}}_{BP}, \hat{\boldsymbol{y}}_{LPT})
\end{gathered}
\tag{5}
$$

where f_f, f_{LPC}, f_{HPC}, f_{comb}, f_{HPT}, f_{LPT}, f_{BP}, and f_{sys} represent the fan, low-pressure compressor, high-pressure compressor, combustor, high-pressure turbine, low-pressure turbine, bypass-section, and system-output modules, respectively. All module outputs are represented as vectors. The superscript * in $\hat{\boldsymbol{y}}^*{}_{HPC}$ is used only in the integrated model formulation to denote the selected HPC output sub-vector passed to the combustor module. It does not represent an additional training target, a new model output, or a special operation during independent module training. Specifically, $\hat{\boldsymbol{y}}^*{}_{HPC}$ contains the predicted HPC gas-path outputs used as combustor inputs, while the predicted high-pressure spool speed $\hat{N}_2$ is excluded because it is used later as an auxiliary coupling variable for the turbine modules rather than as a combustor gas-path input. The final output vector $\hat{\boldsymbol{y}}_{sys}$ contains the predicted thrust, thrust-specific fuel consumption (TSFC), and overall pressure ratio (OPR). The bypass branch is not passed through the core-flow modules; instead, it is predicted in parallel from the inlet/ambient variables and then combined with the core-flow output in the system-output module. The low-pressure spool speed $\hat{N}_1$ and high-pressure spool speed $\hat{N}_2$ are used as auxiliary coupling variables for the turbine modules rather than as gas-path flow variables.

10.37965/jdmd.2026.1442

The system-output module is not treated as a physical gas turbine component. Instead, it is used as a final output module that combines the relevant information from the bypass and core streams to estimate overall engine performance. This distinction is important because the purpose of the modular framework is not only to predict final outputs, but also to preserve intermediate module-level predictions that can support subsystem-level analysis.

The simulated dataset was also used to examine the response of the proposed framework under controlled measurement uncertainty. For this purpose, Gaussian noise was added to selected gas-path parameters according to literature-reported sensor uncertainty levels [24,39–41]. The same modular development, system integration, and sequential fine-tuning procedure was then applied to the noisy dataset. The noise levels and the corresponding controlled perturbation results are presented in Sections 3.3 and 4.1.4, respectively.

3.2. Case Study 2: Industrial PEMS dataset

The second case study uses a real industrial gas turbine PEMS dataset for CO and NOx prediction [28]. Unlike the simulated turbofan case, which allows a detailed component-level modular structure, the PEMS dataset represents industrial operating data with a different measurement configuration. Therefore, the proposed framework is implemented using a reduced section-level modular structure rather than a full component-level gas-path decomposition. This allows the same modelling philosophy to be evaluated under real operating conditions using the available process and ambient measurements.

The PEMS dataset contains hourly averaged measurements collected from an industrial gas turbine operating under practical plant conditions. The available variables include ambient measurements, process-related gas turbine parameters, power-generation-related quantities, and emission outputs. The target outputs for this case study are CO and NOx. Because these data are collected from real operation, they include practical effects such as measurement variability, operating-condition changes, nonlinear emission behaviour, and possible outlier effects. This makes the dataset suitable for evaluating whether the proposed modular framework can be applied beyond simulated gas turbine data.

In conventional predictive emission monitoring, the available measurements are commonly used in a direct input-output manner, where ambient and process variables are mapped directly to system outputs. In the present study, the PEMS dataset is reorganized into a reduced modular structure to preserve a degree of subsystem-level traceability. The modular structure consists of three sections-level modules: a cold-section module, a hot-section module, and an emission module. This structure does not attempt to reproduce every physical component of the industrial gas turbine. Instead, it groups the available variables according to their physical meaning and expected information flow within the system.

Table 3. Input and output variables of each module for the industrial PEMS case study.

Module	Inputs	Outputs
Cold-section module	AT: Ambient Temperature AP: Ambient Pressure AH: Ambient Humidity $AFDP$: Air Filter Difference Pressure	TIT: Turbine Inlet Temperature CDP: Compressor Discharge Pressure
Hot-section module	TIT: Turbine Inlet Temperature CDP: Compressor Discharge Pressure	TAT: Turbine Afterburner Temperature $GTEP$: Gas Turbine Exhaust Pressure TEY: Turbine Energy Yield
Emission module	TIT: Turbine Inlet Temperature CDP: Compressor Discharge Pressure TAT: Turbine Afterburner Temperature $GTEP$: Gas Turbine	CO: Carbon Monoxide NOx: Nitrogen Oxides

10.37965/jdmd.2026.1442

Exhaust Pressure
AH: Ambient Humidity

The reduced modular structure used for the PEMS case is shown in Fig. 5. The cold-section module uses ambient and inlet-related measurements to predict compressor and inlet-temperature-related quantities. The outputs of the cold-section module are then used as inputs to the hot-section module, which predicts downstream turbine and power-generation-related quantities. Finally, the emission module uses selected cold-section and hot-section quantities to predict CO and NOx emissions. In this way, the PEMS case follows the same framework introduced in Section 2, but with a lower modular resolution defined by the available industrial measurements. The input and output variables of the reduced PEMS modular structure are summarized in Table 3.

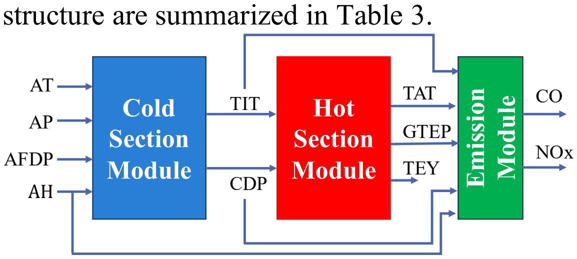


Fig. 5. Reduced modular structure of the industrial PEMS case study.

For the industrial PEMS case, the same integration procedure is implemented at a reduced section-level resolution because complete station-level gas-path measurements are not available. The connected PEMS model is expressed as:

$$\begin{gathered}\hat{\boldsymbol{y}}_{cold} = f_{cold}(AT, AP, AFDP, AH),\\ \hat{\boldsymbol{y}}_{hot} = f_{hot}(\hat{\boldsymbol{y}}_{cold}),\\ \hat{\boldsymbol{y}}_{em} = f_{em}\left(\hat{\boldsymbol{y}}^{*}{}_{hot}, \hat{\boldsymbol{y}}_{cold}, AH\right)\end{gathered} \tag{6}$$

Here, $\hat{\boldsymbol{y}}_{cold} = [\widehat{TIT}, \widehat{CDP}]$, $\hat{\boldsymbol{y}}_{hot} = [\widehat{TAT}, \widehat{GTEP}, \widehat{TEY}]$, and $\hat{\boldsymbol{y}}_{em} = [\widehat{CO}, \widehat{NOx}]$. The superscript * in $\hat{\boldsymbol{y}}^{*}{}_{hot}$ is used only in the integrated model formulation to denote the selected hot-section output sub-vector passed to the emission module. Specifically, $\hat{\boldsymbol{y}}^{*}_{hot} = [\widehat{TAT}, \widehat{GTEP}]$, where $\widehat{TEY}$ is excluded because turbine energy yield is not directly relevant to emission formation and does not provide additional predictive value for the emission outputs. The superscript * does not represent an additional training target, a new model output, or a special operation during independent module training. During independent training, the emission module was trained using the corresponding measured upstream quantities, whereas after integration it receives predicted upstream quantities from the upstream modules. This replacement can shift the downstream module inputs away from the distribution observed during independent training, causing cascading prediction error in the initially integrated model and motivating the sequential fine-tuning step.

The emission module is treated as the final prediction module of the reduced PEMS framework. Similar to the system-output module in the simulated turbofan case, it is not considered a separate physical gas turbine component. Instead, it combines relevant upstream section-level predictions and directly available measurements to estimate the final target outputs. This distinction allows the proposed framework to maintain a modular structure while adapting to the measurement configuration of the industrial dataset.

The use of the PEMS dataset serves two purposes in this study. First, it evaluates the applicability of the proposed framework using real industrial gas turbine data. Second, it allows the proposed modular ANN framework to be compared with monolithic ANN and optimized black-box machine-learning baseline models for emission prediction. This comparison is important because the proposed framework is not intended to only maximize final-output accuracy, but also to preserve intermediate section-level predictions and provide subsystem-level traceability.

3.3. Data preprocessing and model training

The two case studies include variables with different physical units, numerical magnitudes, and operating ranges. Direct use of these variables during ANN training can lead to unstable optimization because variables with larger numerical values may dominate the loss function and gradient updates. Therefore, all input and output variables were normalized before model training. In this study, min-max normalization was applied to scale each variable into the range [0,1], as follows:

$$x_{norm} = \frac{x - min(x)}{max(x) - min(x)} \quad (7)$$

where x is the original variable, and x_{min} and x_{max} are the minimum and maximum values of that variable. To avoid data leakage, the normalization parameters were calculated from the training set and then applied to validation and test sets.

For both case studies, the datasets were divided into training, validation, and testing subsets. Specifically, 70% of the available samples were used for model training, 15% were used for validation during hyperparameter optimization and model selection, and the remaining 15% were reserved for final testing.

The final selected architectures and main training settings of the ANN models are reported in Appendix A. During sequential fine-tuning, the final selected architectures were kept fixed. A module-wise fine-tuning stage refers to one update step in which a single selected module is set as trainable while all other modules in the integrated model are frozen. For the simulated turbofan case, the modules were updated in the upstream-to-downstream sequence of the integrated gas-path mode: fan → LPC → HPC → combustor → HPT → LPT → bypass section → system-output module. After these module-wise updates, a final integrated fine-tuning step was applied with the complete model connected. For the industrial PEMS case, the modules were updated in the sequence cold section → hot section → emission module, followed by a final integrated fine-tuning step of the complete model. The same fine-tuning loss formulation defined in Eq. (4) was used, and the optimizer and training settings are given in Table 4 and Appendix A. Each fine-tuning stage was terminated when validation-based early-stopping was triggered or when the maximum number of epochs reported in Table 4 was reached. Here, early-stopping means that training was stopped when the validation loss did not improve for a specified number of consecutive epochs. The patience value was set to 25 epochs for independent module training and 150 epochs for sequential fine-tuning to allow sufficient validation-loss improvement during the integrated-model adjustment process. In all cases, the model checkpoint with the lowest validation loss was retained.

All modular ANN models were implemented using feedforward ANNs. ReLU activation was used in the hidden layers, and batch-normalization layers were included after the Dense hidden layers to improve training stability. Because the modular ANN outputs were min-max normalized, sigmoid output activation was used for the modular output layers, and the scaled predictions were re-scaled to the original physical units before evaluation. For the optimized monolithic ANN baselines, the output activation was set according to the corresponding implementation and is reported in Appendix A. The Adam optimizer was used for model training. For each module, the ANN architecture and training settings were selected using validation-based hyperparameter optimization with Keras Tuner Random Search. The tuned hyperparameters included the number of hidden layers, number of neurons in each hidden layer, learning rate, and batch size. The complete hyperparameter search space and fixed training settings are summarized in Table 4.

Table 4. Hyperparameter search space and training settings used for ANN module development.

Parameter	Search range / setting
Number of hidden layers	2-6
Neurons per hidden layer	10-256
Hidden-layer activation function	ReLU
Output-layer activation function	Sigmoid for min-max-normalized modular ANN outputs; model-specific setting for benchmark ANN models
Optimizer	Adam
Learning rate	Default Adam learning

	rate 0.001 for modular ANN models; 10^{-4}–10^{-2} logarithmic search for optimized monolithic ANN models
Batch size	32, 64, 128, 256, 512, 1024
Maximum epochs	500
Early stopping	Based on validation loss
Model checkpointing	Best validation-loss model saved
Training loss	MAE + L2 regularization
Model selection criterion	Lowest validation loss
Data split	70% training, 15% validation, 15% testing
L2 regularization coefficient	1×10^{-4}

After completion of the hyperparameter search, the best-performing configuration for each module was selected based on lowest validation loss obtained using the Mean Absolute Error (MAE) based loss function defined in Equation (2). The selected module models were then saved and used for system-level integration and sequential fine-tuning according to the procedure described in Section 2. The general feedforward ANN structure used for module-level model development is shown in Fig. 6. Each module receives min-max-scaled input variables, passes them through Dense hidden layers with ReLU activation and batch-normalization layers, and produces scaled output predictions that are re-scaled to the original physical units. The number of hidden layers and neurons is module-specific, and the final selected architectures are reported in Appendix A. Hyperparameter tuning and model training were conducted on a Lenovo laptop equipped with an Intel Core i5-12500H processor, 16 logical processors, and 16 GB RAM. The implementation was carried out using Python 3.11, TensorFlow, and Keras Tuner.

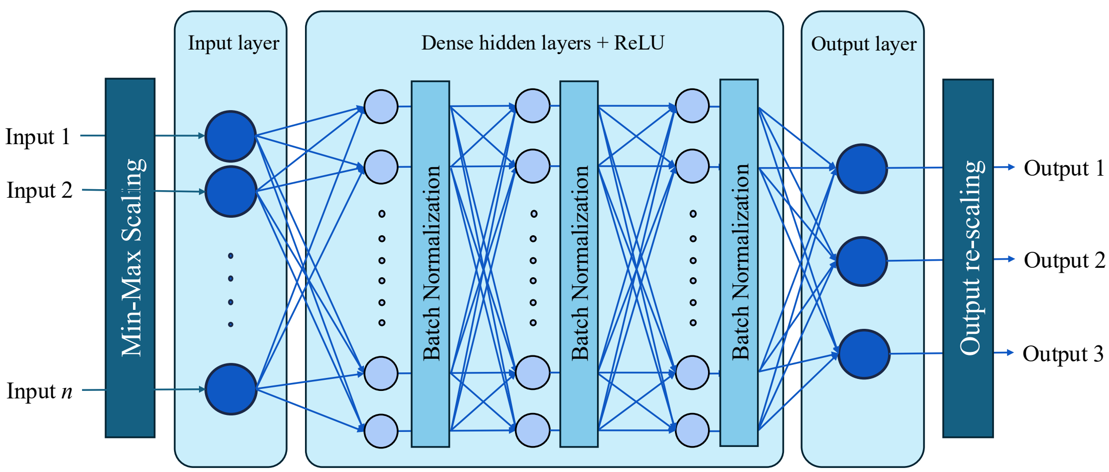


Fig. 6. General feedforward ANN architecture used for module-level model development.

Table 5. Sensor-specific Gaussian noise standard deviations used for the controlled measurement-noise perturbation test of the simulated turbofan case.

Description	Unit	Symbol	Gaussian noise standard deviation (σ)
LPC exit temperature	K	T_{25}	0.4
LPC exit pressure	bar	P_{25}	0.2
HPC exit temperature	K	T_3	0.4
HPC exit pressure	bar	P_3	0.2
Fuel flow rate	kg/s	$\dot{W}_f$	0.5
Combustor exit temperature	K	T_4	0.4
Combustor exit pressure	bar	P_4	0.2
HPT exit temperature	K	T_{45}	0.4
HPT exit pressure	bar	P_{45}	0.2
LPT exit temperature	K	T_5	0.4
LPT exit pressure	bar	P_5	0.2

For the simulated turbofan case, an additional controlled measurement-noise perturbation test was conducted to assess the robustness of the proposed framework. Gaussian noise was added to selected gas-path parameters according to literature-reported sensor uncertainty levels [24,39–41]. The same preprocessing, module training, integration, and fine-tuning procedure were then repeated using the noisy dataset. Table 5 summarizes the noise levels used in this study.

For the measurement-noise evaluation, zero-mean Gaussian noise was added to the selected gas-path variables listed in Table 5 before normalization. For each selected variable x_j, the noisy value was generated as $x_{j,noisy} = x_j + \epsilon_j$, where $\epsilon_j \sim \mathcal{N}(0, \sigma_j^2)$ and σ_j is the sensor-specific standard deviation reported in Table 5. The noise was sampled independently for each selected variable and each sample. One fixed noisy dataset was generated using the same number of samples as the corresponding clean healthy dataset. Therefore, the results reported in Section 4.1.4 should be interpreted as a controlled single-realization sensor-noise perturbation test. They were not averaged over repeated Gaussian noise realizations, and confidence intervals were not calculated. Multi-realization Monte Carlo testing, multiple noise-intensity levels, and confidence-interval reporting are left for future work.

For the industrial PEMS case, no artificial noise was added because the dataset already represents real operating measurements collected under practical plant conditions. Therefore, the PEMS case was used to evaluate the proposed framework under naturally occurring measurement variability, operating-condition changes, and real-data uncertainty.

3.4. Benchmark models and evaluation metrics

For the simulated turbofan case, the main objective was to evaluate the detailed modular implementation of the proposed framework, including component-level prediction, system-level integration, cumulative error after integration, sequential fine-tuning, and response under a controlled measurement-noise perturbation. In addition, an optimized monolithic ANN baseline was developed to provide a direct final-output comparison with the proposed modular framework. This baseline was included to assess whether the modular structure improves or maintains final-output prediction accuracy while providing intermediate module-level predictions. To ensure a fair comparison, the monolithic ANN used the same training, validation, and testing partitions, min-max normalization procedure, optimizer, MAE-based loss function, early-stopping criterion, and hyperparameter search space used for the modular models. The input vector consisted of the external operating and control variables used by the modular framework, $[Z, Mach, T_{amb}, P_{amb}, RH, \dot{W}_f]$, and the output vector consisted of the final system-level quantities, namely thrust, TSFC, and OPR. Intermediate gas-path variables were not used as monolithic ANN inputs, so that the baseline represented a direct system-level input-output model rather than a model supplied with component-level information.

For the industrial PEMS case, the benchmark comparison was conducted as a real-data tabular emission-monitoring problem. This comparison is appropriate because predictive emission monitoring models are commonly developed as direct input-output models that map available ambient and process measurements to final emission outputs. Therefore, comparing the proposed modular ANN framework with optimized black-box machine-learning models allows the trade-off between final-output accuracy and modular interpretability to be examined.

For the industrial PEMS case, the proposed modular ANN framework was compared with several direct input-output benchmark models, including a monolithic ANN, a multi-output monolithic ANN, an optimized Random Forest regression model, an optimized Extra Trees regression model, an optimized Support Vector Regression model with a radial basis function kernel, and an optimized Extreme Gradient Boosting (XGBoost) regression model. These models were selected because they are commonly used for nonlinear regression problems involving tabular industrial data and provide suitable black-box baselines for predictive emission monitoring. The initially integrated modular ANN before fine-tuning was also reported as an intermediate stage of the proposed framework to quantify the effect of sequential fine-tuning on the final CO and NOx prediction accuracy. Two additional deep-learning baselines, a Deep Multilayer Perceptron (MLP) and a Tabular Transformer-style neural network, were also implemented for the PEMS case using the same train/validation/test split, normalization procedure, loss function, and early-stopping criterion.

Different evaluation metrics were used according to the objective of each case study. For the simulated turbofan case, mean absolute percentage error (MAPE) [42] was used as the main evaluation metric because it provides a scale-independent percentage measure of prediction error across gas turbine outputs with different units and magnitudes, such as temperatures, pressures, thrust, TSFC, and OPR. The MAPE is defined as:

$$MAPE = \frac{100}{N}\sum_{i=1}^{N}\left|\frac{y^{(i)} - \hat{y}^{(i)}}{y^{(i)}}\right| \tag{8}$$

where $y^{(i)}$ and $\hat{y}^{(i)}$ are the measured and predicted values, respectively, and N is the number of test samples.

Although MAE was used as the loss function during ANN training, MAPE was used as an evaluation metric for the simulated turbofan case. This distinction is intentional. MAE provides a stable optimization objective during training and avoids the numerical instability that can occur when percentage-based losses are used directly in gradient-based optimization. MAPE, on the other hand, is useful for reporting the final prediction accuracy because the simulated gas turbine outputs are positive and non-zero, allowing the prediction error to be interpreted as a percentage of the measured value.

For the PEMS case, MAE was used as the primary evaluation metric. This choice is consistent with the original study of Kaya et al. [28], allowing direct comparison with previously reported CO and NOx prediction results on the same dataset. In addition, MAE reports the prediction error in the original emission units, which is more practical for emission-monitoring applications. MAPE was not used as metric for the PEMS case because CO values can be small in some operating regions (near zero), causing percentage errors to become artificially large and less representative of practical prediction quality. The MAE is defined as:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|y^{(i)} - \hat{y}^{(i)}\right| \tag{9}$$

4. Results and discussion

The simulated turbofan case is first used to evaluate the detailed modular implementation of the proposed framework, including component-level prediction, system-level integration, sequential fine-tuning, and response under a controlled measurement-noise perturbation. The industrial PEMS case is then used to evaluate the applicability of the framework to real operating data and to compare the modular ANN framework with direct input-output benchmark models.

4.1. Results for simulated high-bypass turbofan dataset

The simulated turbofan case study was used to evaluate the proposed framework under a detailed component-level modular structure. The analysis was performed in three stages. First, each module was trained and evaluated independently to assess whether the ANN models could capture the local behavior of the corresponding gas turbine sections. Second, the trained modules were integrated according to the gas-path and inter-component relationships to evaluate the effect of system-level coupling. Third, sequential fine-tuning was applied to reduce accumulated prediction errors after integration.

10.37965/jdmd.2026.1442

4.1.1. Component-level prediction results

The first stage of model development involved training independent ANN models for the fan, bypass section, LPC, HPC, combustor, HPT, LPT, and system-output module. Each module was trained using the input-output variables defined in Table 2 and the hyperparameter optimization procedure described in Section 3.3. The purpose of this stage was to ensure that each module could accurately learn the local nonlinear relationship between its selected inputs and outputs before being integrated into the system-level modular model.

Fig. 7 shows the representative predictions for the system-output module. The predicted values closely follow the actual values for the system output parameters, indicating that the independently trained ANN model can capture the local system-output behavior under variable operating and ambient conditions. Similar training and evaluation procedures were applied to all other modules.

Component level rows in Table 6 summarize the MAPE values obtained for all independently trained component models. The results show that the individual ANN modules achieved low prediction errors across the major gas-path sections. This confirms that the selected input-output variables and module-level ANN models were able to represent the local behavior of the corresponding gas turbine sections. The system-output module was also evaluated at this stage; however, it should be noted that this module does not represent a physical component. Instead, it acts as the final prediction module used to estimate thrust, TSFC, and OPR from the relevant upstream core and bypass information.

These component-level results are important because they establish the baseline accuracy of each module before system-level integration. However, strong independent performance does not necessarily guarantee the same level of accuracy after integration because downstream modules receive predicted upstream outputs rather than measured upstream variables. This issue is evaluated in the next subsection.

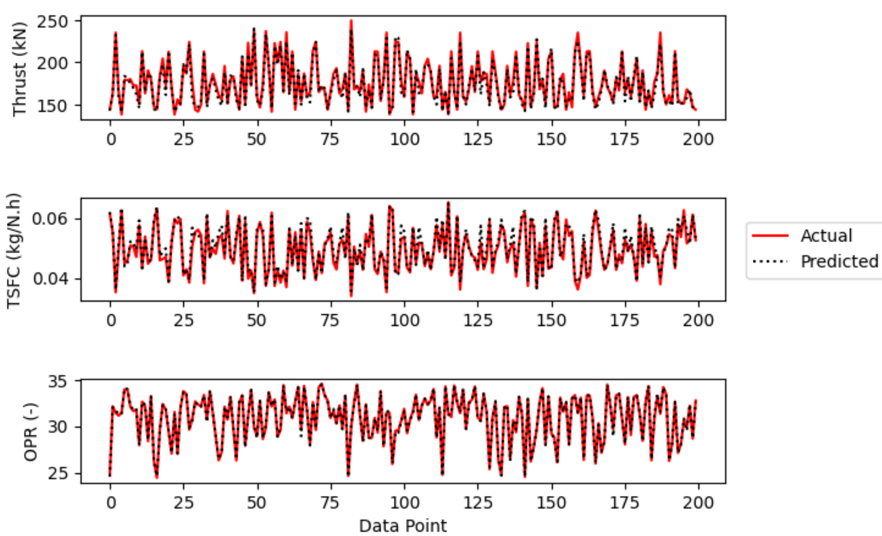


Fig. 7. Representative prediction performance of the system-output module for the simulated turbofan.

Table 6. MAPE comparison of the simulated turbofan modular model at different modelling stages.

Component model	Stage	MAPE values of the output parameters (%)			
		Parameter 1	Parameter 2	Parameter 3	Parameter 4
Fan	Component level	T_{21}	P_{21}	η_{fan}	N_1
		0.4861	0.3373	0.0511	0.2806

10.37965/jdmd.2026.1442

	Before fine tune	0.4861	0.3373	0.0511	0.2806
	After fine tune	0.3915	0.3314	0.0597	0.2729
Bypass section		T_{BP}	P_{BP}	BPR	-
	Component level	0.2590	0.4452	0.1511	
	Before fine tune	0.5626	0.4324	0.1498	
	After fine tune	0.3754	0.3723	0.1476	
LPC		T_{25}	P_{25}	η_{LPC}	-
	Component level	0.3511	0.2796	0.2418	
	Before fine tune	0.5258	0.4408	0.2542	
	After fine tune	0.3765	0.2767	0.1335	
HPC		T_3	P_3	η_{HPC}	N_2
	Component level	0.3339	0.2801	0.0106	0.2300
	Before fine tune	0.6491	0.9735	0.0161	0.3479
	After fine tune	0.3480	0.7556	0.0106	0.2499
Combustor		T_4	P_4	-	-
	Component level	0.2263	0.1814		
	Before fine tune	0.7606	0.6359		
	After fine tune	0.3771	0.3373		
HPT		T_{45}	P_{45}	η_{HPT}	-
	Component level	0.1877	0.1167	0.0650	
	Before fine tune	0.7949	0.6035	0.1023	
	After fine tune	0.4065	0.1875	0.0725	
LPT		T_5	P_5	η_{LPT}	-
	Component level	0.1908	0.1604	0.0262	
	Before fine tune	0.9295	0.5841	0.0493	
	After fine tune	0.2821	0.2424	0.0229	
System-output module		$Thrust$	$TSFC$	OPR	-
	Component level	2.2584	2.4510	0.3218	
	Before fine tune	6.1518	6.1437	0.8105	
	After fine tune	2.4909	2.3694	0.4966	

4.1.2. System-level integration and sequential fine-tuning

After the component-level models were trained, the best-performing ANN model for each module was integrated according to the modular structure shown in Fig. 4 and as per the connected prediction sequence defined in Section 3.1. In the initially integrated model, the predicted outputs of upstream modules were passed to downstream modules according to the defined gas-path, bypass/core and shaft-coupling relationships. This stage evaluates whether independently trained modules can maintain their prediction accuracy when connected as a complete modular gas turbine system.

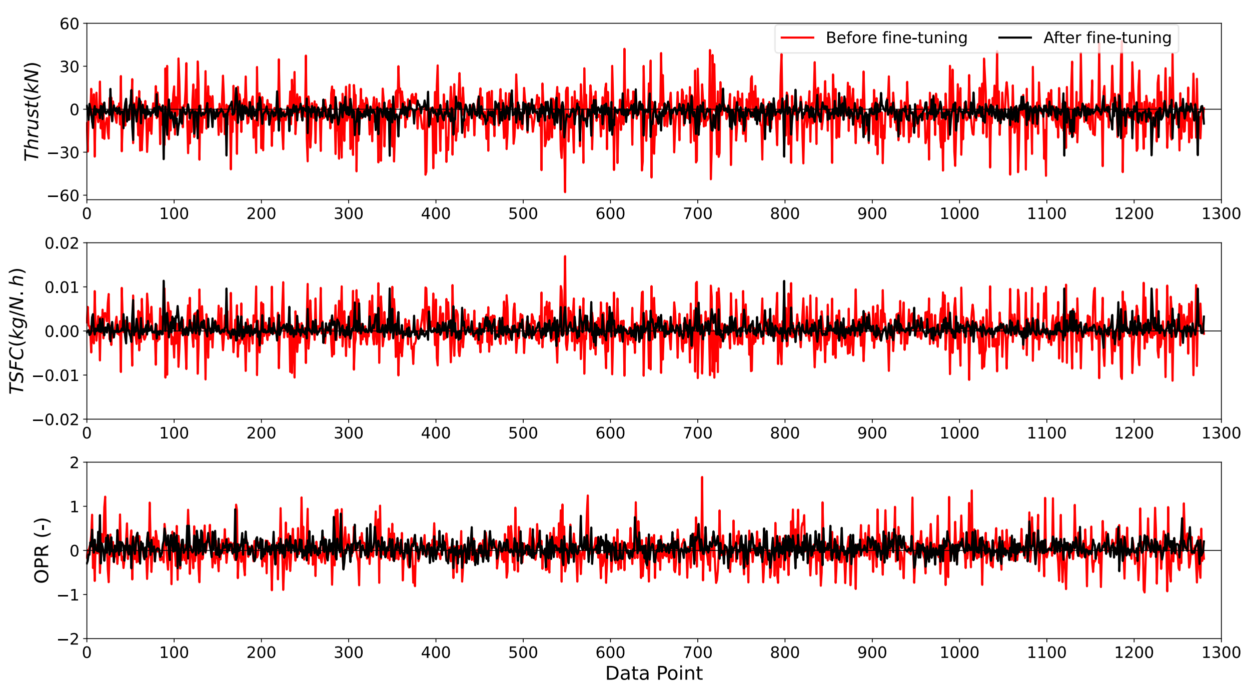
Before fine tune rows in Table 6 presents the MAPE values obtained after direct system-level integration. Compared with the independently trained component results, the prediction errors increase for downstream modules. This occurs because downstream modules no longer receive the measured upstream variables used during independent training; instead, they receive predicted upstream outputs from preceding ANN modules. Therefore, small upstream prediction deviations can influence downstream predictions and accumulate through the connected modular structure.

The effect of this error accumulation is most visible in the final system-output module. Before fine-tuning, the MAPE values for thrust, TSFC, and OPR were 6.1518%, 6.1437%, and 0.8105%, respectively. These results indicate that direct integration of independently trained modules is not sufficient to obtain the best system-level performance. Therefore, sequential fine-tuning was applied to improve the behavior of the connected modular system.

After fine tune rows in Table 6 shows the MAPE values after sequential fine-tuning. The fine-tuning results also show that the error of every intermediate output does not necessarily decrease after fine-tuning. For example, the MAPE of η_{fan} increased from 0.0511% before fine-tuning to 0.0597% after fine-tuning. However, during the same fine-tuning process, the final system-level errors decreased from 6.1518% to 2.4909% for thrust, from 6.1437% to 2.3694% for TSFC, and from 0.8105% to 0.4966% for OPR. This behavior indicates a trade-off between local module accuracy and global integrated-model accuracy. The purpose of fine-tuning is therefore not to make every intermediate output more accurate in isolation, but to calibrate the connected model so that the final integrated outputs improve while intermediate predictions remain physically traceable.

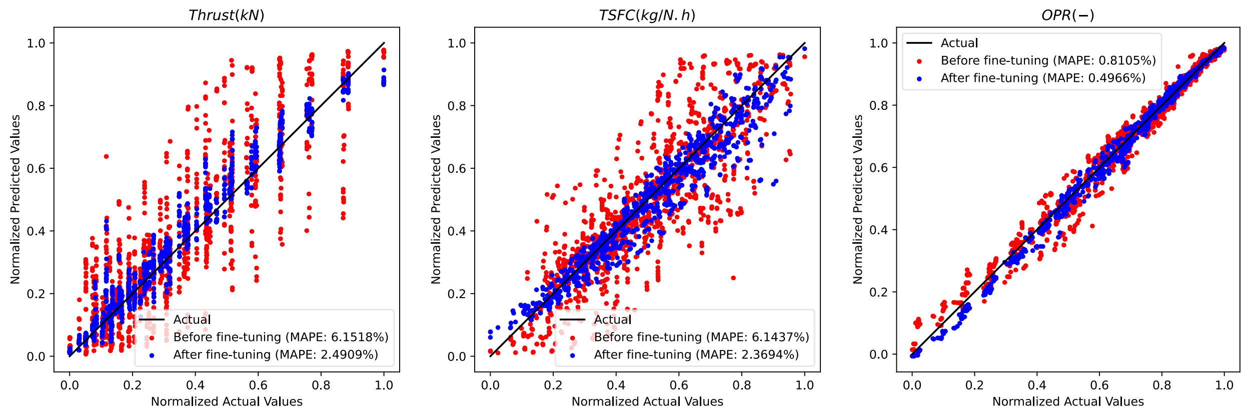
Fig. 8 further illustrates the effect of sequential fine-tuning on the final system outputs. Fig. 8(a) compares the prediction errors of the final system-output module before and after fine-tuning, while Fig. 8(b) shows the agreement between actual and predicted normalized values. The improvement after fine-tuning confirms that the proposed framework does not simply connect independently trained modules but also adjusts the integrated modular system to reduce the mismatch caused by replacing measured upstream variables with predicted upstream outputs.

This result supports the role of sequential fine-tuning in the proposed framework. The independent module-training stage allows each module to learn local gas turbine behavior, while the fine-tuning stage improves the coordinated behavior of the connected system. Therefore, the final integrated model preserves intermediate module-level predictions while achieving improved final-output accuracy.



(a)

10.37965/jdmd.2026.1442



(b)

Fig. 8. Effect of sequential fine-tuning on the final system-output module predictions for the simulated turbofan case: (a) MAPE comparison before and after fine-tuning and (b) representative actual and predicted normalized output trends.

To further evaluate the final-output accuracy of the proposed modular framework, the fine-tuned modular ANN was compared with an optimized monolithic ANN baseline trained end-to-end on the same clean simulated turbofan dataset. The monolithic ANN used the same external operating and control inputs as the modular framework and directly predicted thrust, TSFC, and OPR. The results are summarized in Table 7. The monolithic ANN achieved MAPE values of 2.2580%, 2.5089%, and 0.9443% for thrust, TSFC, and OPR, respectively. The fine-tuned modular ANN achieved corresponding MAPE values of 2.4909%, 2.3694%, and 0.4966%. These results show that the modular framework does not uniformly outperform the monolithic ANN for every final output; the monolithic ANN produced a slightly lower thrust error, whereas the modular ANN produced lower TSFC and OPR errors. When averaged across the three final outputs, the modular ANN achieved a slightly lower MAPE than the monolithic ANN, with average values of 1.7856% and 1.9037%, respectively. Therefore, the comparison does not indicate a general loss of final-output accuracy due to modularization. Instead, it shows that the proposed framework maintains competitive end-to-end prediction accuracy while additionally preserving intermediate gas-path predictions. This added structure supports subsystem-level traceability, cascading-error analysis, and modular model updating, which are not available from the monolithic ANN baseline.

Table 7. Final-output MAPE comparison between the optimized monolithic ANN and the proposed modular ANN for the simulated turbofan case.

Model	Thrust MAPE (%)	TSFC MAPE (%)	OPR MAPE (%)
Optimized monolithic ANN	2.2580	2.5089	0.9443
Proposed modular ANN after fine-tuning	2.4909	2.3694	0.4966

4.1.3. Physical consistency verification

To further assess whether the predicted intermediate quantities remain physically reasonable, a post-training physical consistency verification was performed for the simulated turbofan case. This verification was not used as an additional training constraint; rather, it was used to check whether the trained modular model preserved the basic gas-path trends expected from the engine configuration. The checked conditions included temperature and pressure increases across the fan and compressor modules, temperature increase across the combustor, temperature and pressure decreases across the turbine modules, efficiency values within the physically admissible range, and positive final system-level outputs.

The results are summarized in Table 8. No violations were observed for the fan-compressor temperature-rise and pressure-rise checks, combustor temperature-rise check, turbine temperature-drop and pressure-drop checks, efficiency-range check, or final-output positivity check. For the combustor, a pressure loss is normally expected because of flow resistance and mixing through the combustor. Therefore, the strict condition $\hat{P}_4 \le \hat{P}_3$ was also checked. This condition was violated in 17.96% of the test samples. However, the average relative pressure excess in the violating samples was only 0.47% of $\hat{P}_3$, and only 0.72% of all test samples had a relative pressure excess greater than 1%. The 1% value is not used to redefine the physical constraint; it is only reported to indicate the practical magnitude of the strict pressure-loss violations. These results show that the model preserves the main physical trends within the tested operating envelope, while also confirming that detailed thermodynamic constraints are not strictly enforced during training. Explicit enforcement of such constraints is left for future physics-informed extensions.

Table 8. Physical consistency verification of the fine-tuned modular model for the simulated turbofan case.

Constraint group	Checked condition	Violation rate (%)
Fan-compressor temperature rise	$\hat{T}_1 \le \hat{T}_{2.1} \le \hat{T}_{2.5} \le \hat{T}_3$	0.00
Fan-compressor pressure rise	$\hat{P}_1 \le \hat{P}_{2.1} \le \hat{P}_{2.5} \le \hat{P}_3$	0.00
Combustor temperature rise	$\hat{T}_4 \ge \hat{T}_3$	0.00
Strict combustor pressure loss	$\hat{P}_4 \le \hat{P}_3$	17.96
Combustor pressure excess above 1%	$\frac{\hat{P}_4 - \hat{P}_3}{\hat{P}_3} > 1\%$	0.72
Turbine temperature drop	$\hat{T}_5 \le \hat{T}_{4.5} \le \hat{T}_4$	0.00
Turbine pressure drop	$\hat{P}_5 \le \hat{P}_{4.5} \le \hat{P}_4$	0.00
Efficiency range	$0 \le \hat{\eta}_i \le 1;\ i \in \{fan, LPC, HPC, HPT, LPT\}$	0.00
Final-output positivity	$\widehat{Thrust}, \widehat{TSFC}, \widehat{OPR} > 0$	0.00

4.1.4. Controlled measurement-noise perturbation test

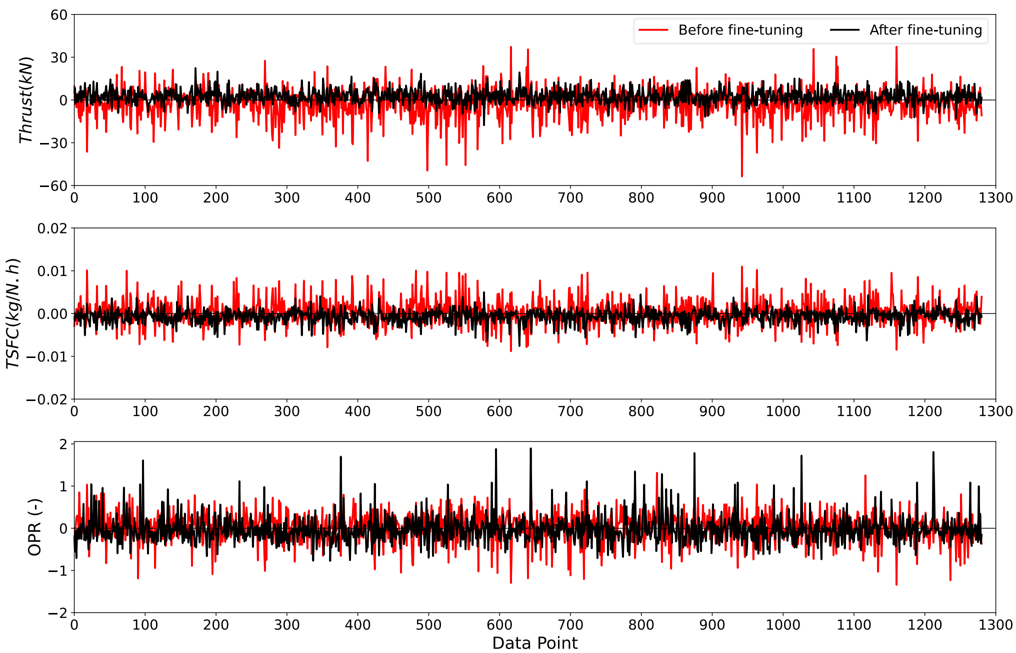
To evaluate the response of the proposed framework under controlled measurement uncertainty, Gaussian noise was added to selected gas-path parameters using the noise levels described in Table 5. The same modular training, integration, and sequential fine-tuning procedure was then repeated using the noisy dataset. This analysis was designed as a controlled sensor-noise perturbation test, not as a full Monte Carlo uncertainty-quantification study, and was used to examine whether the proposed framework could maintain acceptable prediction accuracy when sensor-level uncertainty is introduced.

Table 9 presents the component-level prediction results obtained under noisy measurement conditions. The results show that the proposed framework maintains stable prediction performance despite the presence of simulated sensor noise. Although the prediction errors increase slightly compared with the clean-data case, the overall accuracy remains acceptable across the gas-path modules.

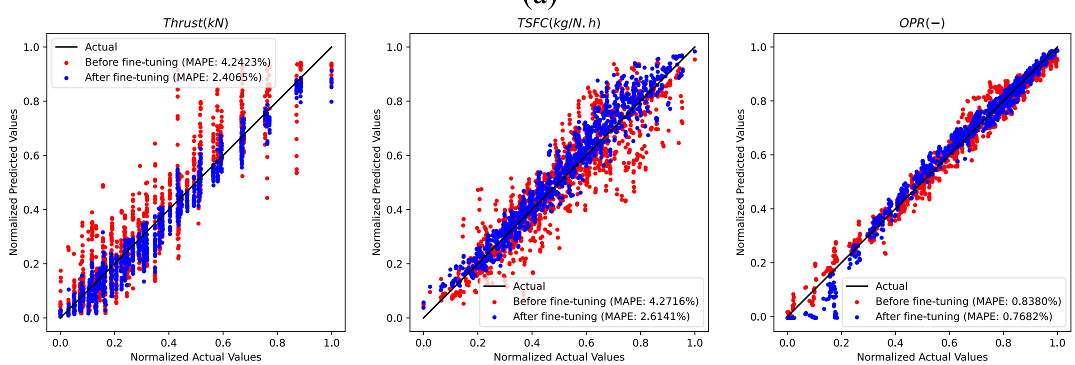
Table 9. Component-level MAPE under measurement noise before and after sequential fine-tuning, including average module error and percentage change.

Component model	Stage	MAPE values of the output parameters (%)				Average MAPE (Percentage change)
		Parameter 1	Parameter 2	Parameter 3	Parameter 4	
Fan	Before fine tune	T_{21}	P_{21}	η_{fan}	N_1	

		0.5202	0.4122	0.0489	0.3943	0.3439
	After fine tune	0.5503	0.4245	0.0481	0.3052	0.3320 (−3.45%)
Bypass section		T_{BP}	P_{BP}	BPR	-	
	Before fine tune	0.5735	0.4102	0.1490		0.3776
	After fine tune	0.4714	0.4768	0.1468		0.3650 (−3.33%)
LPC		T_{25}	P_{25}	η_{LPC}	-	
	Before fine tune	0.5131	2.0329	0.1959		0.9140
	After fine tune	0.3066	1.9831	0.1049		0.7982 (−12.66%)
HPC		T_3	P_3	η_{HPC}	N_2	
	Before fine tune	0.5660	1.3051	0.0184	0.4485	0.5845
	After fine tune	0.3279	1.2052	0.0158	0.3467	0.4739 (−18.92%)
Combustor		T_4	P_4	-	-	
	Before fine tune	0.5814	1.3377			0.9596
	After fine tune	0.2939	0.7759			0.5349 (−44.25%)
HPT		T_{45}	P_{45}	η_{HPT}	-	
	Before fine tune	0.5999	1.2625	0.0882		0.6502
	After fine tune	0.3445	0.5914	0.0458		0.3272 (−49.67%)
LPT		T_5	P_5	η_{LPT}	-	
	Before fine tune	0.6834	2.5688	0.0353		1.0958
	After fine tune	0.3036	2.1783	0.0250		0.8356 (−23.74%)
System-output module		$Thrust$	$TSFC$	OPR	-	
	Before fine tune	4.2423	4.2716	0.8380		3.1173
	After fine tune	2.4065	2.6141	0.7682		1.9296 (−38.10%)



(a)



(b)

Fig. 9. Engine model performance before and after the fine-tuning process for data with measurement noise (a) actual vs predicted values (b) normalized values.

To provide a compact comparison, the output-wise MAPE values were averaged for each module, and the resulting module-wise average MAPE values are also provided in Table 9 which compares the prediction errors under noisy conditions before and after sequential fine-tuning. Similar to the clean-data case, the initially integrated model shows increased downstream error due to propagation of upstream prediction deviations. After sequential fine-tuning, the final system-level errors are reduced, demonstrating that the fine-tuning strategy remains effective even when measurement noise is present.

The reductions in MAPE values, highlighted in Fig. 9, demonstrate the effectiveness of fine-tuning for system integration under the tested noise case. The controlled perturbation results show that the proposed framework can tolerate the applied measurement-noise realization while preserving the modular prediction structure. This is important for practical gas turbine monitoring because sensor measurements are affected by environmental conditions, calibration error, signal interference, and operational variability. The results suggest that the modular ANN framework, combined with sequential fine-tuning, can provide stable system-level prediction even when the input data contain measurement noise.

It should be noted that this measurement-noise evaluation was based on one fixed Gaussian noise realization. Therefore, the results demonstrate the behavior of the proposed framework under a representative controlled sensor-noise perturbation, rather than statistical robustness over repeated noise realizations. Because Monte Carlo trials were not performed, confidence intervals are not reported. Repeated noise realizations, multiple noise-intensity levels, and confidence-interval reporting will be considered in future work to further quantify statistical robustness.

4.2. Results for Industrial PEMS dataset

The industrial PEMS case study was used to evaluate the proposed framework using real gas turbine operating data. Unlike the simulated turbofan case, where detailed component-level gas-path variables were available, the PEMS dataset was implemented using a reduced modular structure consisting of cold-section, hot-section, and emission modules. The purpose of this case study was to assess whether the proposed modular modelling and sequential fine-tuning strategy can be applied to real operational data while preserving intermediate section-level predictions.

4.2.1. Effect of sequential fine-tuning on the modular PEMS model

The reduced modular ANN model was first developed by training the cold-section, hot-section, and emission modules independently. The trained modules were then integrated according to the structure shown in Fig. 5 and as per the connected prediction sequence defined in Section 3.2. In the initially integrated model, downstream modules received predicted upstream quantities instead of measured upstream values. Sequential fine-tuning was then applied to reduce the mismatch introduced by this integration process. The prediction performance before and after fine-tuning is summarized in Table 10.

Table 10. Module-level MAE of the industrial PEMS modular ANN before and after sequential fine-tuning.

Module	Output	Before fine tune	After fine tune	MAE reduction (%)
Cold section module	TIT	5.6644	5.6644	-
	CDP	0.3498	0.3498	-
Hot section module	TAT	2.1432	2.0631	3.7388
	GTEP	1.5187	1.5135	0.3388
	TEY	5.2873	5.2697	0.3319
Emission module	CO	0.9784	0.8746	10.6076
	NOx	8.5516	5.1971	39.2270

Table 10 shows that the effect of sequential fine-tuning is more pronounced in the downstream modules than in the upstream module. The cold-section module shows identical MAE values before and after fine-tuning because it is the entry module of the reduced PEMS structure and receives only directly measured input variables. Therefore, it is not affected by the replacement of upstream measured quantities with upstream predicted outputs. In the sequential fine-tuning process, the cold-section module was retained as the upstream boundary module, while the downstream hot-section and emission modules were adjusted to improve integrated system behaviour.

For the hot-section module, sequential fine-tuning provides modest reductions in MAE. The MAE of TAT decreases from 2.1432 to 2.0631, corresponding to a reduction of 3.7388%. Smaller improvements are observed for GTEP and TEY, with MAE reductions of 0.3388% and 0.3319%, respectively. These modest reductions indicate that the hot-section module was already relatively stable after initial integration, but fine-tuning still slightly improved the consistency of the downstream predictions.

The improvement is more significant for the final emission module. The MAE of CO decreases from 0.9784 before fine-tuning to 0.8746 after fine-tuning, giving a reduction of 10.6076%. For NOx, the MAE decreases from 8.5516 to 5.1971, corresponding to a reduction of 39.2270%. This improvement confirms that sequential fine-tuning is particularly useful for the final output module, where prediction errors from upstream section-level modules can influence the final emission estimates.

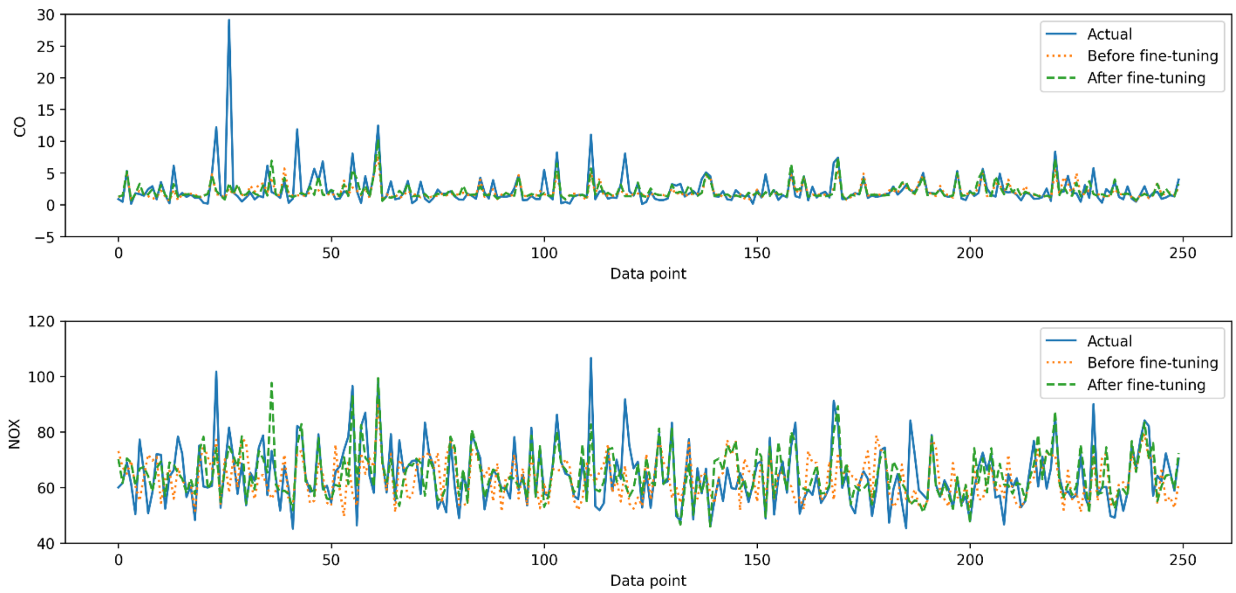


Fig. 10. Actual and predicted CO and NOx values of the industrial PEMS emission module before and after sequential fine-tuning.

Fig. 10 compares the actual and predicted CO and NOx values before and after sequential fine-tuning for representative test samples. For CO, both models capture the general low-emission trend, although sharp emission spikes remain difficult to predict because they occur only over limited operating regions. After fine-tuning, the predicted CO trend becomes closer to the actual values in the normal operating range, which is consistent with the reduction in MAE reported in Table 10. For NOx, the improvement after fine-tuning is more visible, with the fine-tuned model following the actual fluctuations more closely than the initially integrated model.

Overall, the PEMS results demonstrate that the proposed framework can be adapted to real industrial gas turbine data using a reduced modular structure. The before-and-after comparison shows that sequential fine-tuning improves the behaviour of the connected modules, especially for the final emission outputs, while preserving intermediate section-level predictions.

4.2.2. Benchmark comparison for the industrial PEMS case

To further evaluate the prediction performance of the proposed modular ANN framework on the industrial PEMS dataset, its final fine-tuned results were compared with several direct input-output benchmark models. This comparison was performed only for the PEMS case because predictive emission monitoring is commonly formulated as a tabular regression problem, where available ambient and process measurements are directly mapped to final emission outputs. Therefore, comparison with optimized black-box regression models provides a suitable reference for assessing the trade-off between final-output prediction accuracy and modular interpretability.

The benchmark models included a monolithic ANN, a multi-output monolithic ANN, an optimized Random Forest regression model, an optimized Extra Trees regression model, an optimized Support Vector Regression model with a radial basis function kernel, and an optimized XGBoost regression model. To further strengthen the comparison with deep-learning baselines, two additional neural-network models were implemented using the same input variables, output variables, train/validation/test split, normalization procedure, loss function, and early-stopping criterion. These two models were selected because the PEMS case is a tabular steady-state regression problem: the Deep MLP provides a standard fully connected deep-learning baseline for tabular regression, while the Tabular Transformer-style model provides an attention-based tabular neural-network baseline for capturing feature interactions among continuous input variables. The first model was a deep multilayer perceptron with four dense hidden layers, batch normalization, ReLU activation, and a two-output linear regression layer. The second was an attention-based tabular neural network, referred to here as a Tabular Transformer-style model, in which each continuous input variable was tokenized into a latent representation and processed through Transformer-style self-attention blocks before the final regression layer. In addition, the best MAE values reported by Kaya et al. [28] for the same PEMS dataset are included as a literature reference. This comparison is used to contextualize the prediction accuracy of the proposed framework; however, it should be interpreted with caution because differences in data partitioning, preprocessing, and model development settings may affect direct comparability with previously reported literature results.

Table 11. MAE comparison of the proposed modular ANN framework with benchmark models for the industrial PEMS dataset.

Model	CO MAE	NOx MAE
Kaya et al. [28], best reported model	0.9300	7.9100
Monolithic ANN	0.8826	5.1636
Multi-output monolithic ANN	0.8878	5.2353
Deep MLP	0.8845	5.1140
Tabular Transformer-style neural network	0.8895	5.3380
Random Forest regression	0.8701	4.9324
Extra Trees regression	0.8384	4.7731
Support Vector Regression with radial basis function kernel	0.9588	5.9434
XGBoost regression	0.9288	5.2855
Proposed modular ANN after sequential fine-tuning	0.8746	5.1971

Table 11 shows that the proposed modular ANN after sequential fine-tuning achieved MAE values of 0.8746 for CO and 5.1971 for NOx. These values are lower than the best MAE values reported by Kaya et al. [28], which were 0.93 for CO and 7.91 for NOx. Compared with the additional deep-learning baselines, the proposed modular ANN achieved lower CO MAE than both the Deep MLP (0.8845) and the Tabular Transformer-style model (0.8895). For NOx, the Deep MLP achieved a slightly lower MAE (5.1140) than the proposed modular ANN (5.1971), while the Tabular Transformer-style model produced a higher MAE (5.3380). These results show that the proposed modular framework provides competitive prediction accuracy on the PEMS dataset while also preserving intermediate section-level predictions through the cold-section and hot-section modules.

Among the benchmark models developed in this study, the optimized Extra Trees regression model achieved the lowest MAE for both CO and NOx, with values of 0.8384 and 4.7731, respectively. The optimized Random Forest regression model also achieved slightly lower MAE values than the proposed modular ANN, and the Deep MLP achieved slightly lower NOx MAE. However, these benchmark models are direct black-box input-output predictors and do not provide intermediate section-level predictions. Therefore, although they achieve strong final-output accuracy, they do not support subsystem-level traceability or analysis of how prediction errors evolve through the modular structure.

The practical value of the modular structure can be illustrated using the PEMS fine-tuning results in Table 10. In a direct black-box model such as Extra Trees, only the final CO and NOx errors are available, so it is difficult to determine whether a large emission prediction error originates from the upstream operating-condition representation or from the final emission-mapping stage. In contrast, the proposed modular ANN provides intermediate errors for the cold-section, hot-section, and emission-output modules. In this case, the cold-section errors remained unchanged after integration because this module receives directly measured inputs, while the hot-section errors changed only slightly after fine-tuning. The main improvement occurred in the emission-output module, where the CO MAE decreased from 0.9784 to 0.8746 and the NOx MAE decreased from 8.5516 to 5.1971. This indicates that the dominant source of the final emission prediction error was not the upstream cold-section representation, but the downstream emission-output mapping under integrated predicted-input conditions. This example demonstrates how the modular framework can localize the section in which prediction error is amplified, whereas a direct final-output model can only report the final prediction error.

The proposed modular ANN achieved prediction accuracy close to the best-performing black-box models while retaining the structural advantages of the modular framework. Compared with the initially integrated modular ANN before fine-tuning, the proposed fine-tuned model reduced the CO MAE from 0.9784 to 0.8746 and the NOx MAE from 8.5516 to 5.1971. This confirms that sequential fine-tuning substantially improves the integrated modular model and makes final-output accuracy comparable with direct input-output benchmark models.

Overall, these results indicate that the proposed modular ANN provides competitive final-output accuracy while preserving intermediate section-level predictions that are not available in direct input-output benchmark models. The comparison with additional deep-learning baselines further shows that increasing the complexity of a direct neural-network model does not automatically provide better performance for both emission outputs. Therefore, the proposed framework should be interpreted as an accuracy–traceability trade-off rather than only as a final-output black-box predictor. The broader trade-off between black-box prediction accuracy and modular interpretability is further discussed in Section 4.3.

4.3. Discussion

The results demonstrate that the proposed domain-knowledge-guided modular ANN framework provides a structured approach for gas turbine performance and emission prediction. In the simulated turbofan case, the use of separate modules for the fan, bypass section, compressors, combustor, turbines, and system-output module allowed the local behaviour of individual gas turbine sections to be learned before system-level integration. This supports the main objective of the framework: preserving gas turbine structure and intermediate module-level information rather than treating the system as a single unstructured input-output mapping.

A key finding from the simulated turbofan case is that accurate independent module performance does not automatically guarantee accurate system-level prediction after integration. When the trained modules are connected, downstream modules receive predicted upstream quantities rather than measured upstream variables. This can introduce accumulated error through the connected modular structure. Sequential fine-tuning addressed this issue by adjusting the integrated model after module connection. The reduction of final-output errors for thrust, TSFC, and OPR after fine-tuning confirms that this step is not simply an additional training stage, but a necessary system-level refinement process for improving the consistency of the connected modules.

The measurement-noise analysis further shows that the framework maintains stable prediction performance under controlled sensor uncertainty. This is important because gas turbine measurements are affected by sensor noise, calibration differences, environmental variation, and operational disturbances. The controlled measurement-noise perturbation results indicate that the proposed modular ANN framework maintained stable prediction accuracy under the tested noisy input condition while preserving the modular prediction structure. This supports its potential use in measurement-oriented gas turbine monitoring applications, although broader statistical robustness should be assessed through repeated noise realizations in future work.

The industrial PEMS case extends the evaluation beyond the simulated turbofan dataset and demonstrates that the same modular modelling philosophy can be applied to real gas turbine operating data. Since the PEMS dataset does not provide the same detailed station-level measurements as the simulated turbofan case, the framework was implemented using a reduced section-level structure consisting of cold-section, hot-section, and emission/output modules. The improvement in CO and NOx prediction after sequential fine-tuning shows that the proposed integration and refinement strategy remains useful even when the modular resolution is coarser than a full component-level gas turbine model.

The benchmark comparison for the PEMS dataset highlights an important trade-off between final-output accuracy and modular interpretability. The optimized Extra Trees regression model achieved the lowest MAE among the benchmark models, while the Deep MLP also achieved slightly lower NOx MAE than the proposed modular ANN. This shows that direct black-box regression models, including both ensemble methods and deep neural networks, can be highly effective for tabular emission prediction. However, these models operate as direct input-output predictors and do not provide intermediate section-level predictions. In contrast, the proposed modular ANN achieved competitive emission prediction accuracy while preserving cold-section and hot-section outputs before final CO and NOx prediction. Therefore, the value of the proposed framework is not only in final-output accuracy, but also in subsystem-level traceability, error-propagation analysis, and modular updating capability.

Therefore, the modular framework should be interpreted as an accuracy–traceability trade-off: although the optimized Extra Trees model achieved the lowest final-output MAE, the proposed modular ANN provides intermediate section-level outputs that can be used to identify whether prediction errors are mainly introduced in the upstream operating-condition representation, the hot-section mapping, or the final emission-output mapping.

The modular structure also offers practical advantages for long-term model use. Because each module is developed separately before integration, individual modules can be retrained or refined when new measurements, updated operating conditions, or modified component information become available. This is relevant for industrial gas turbines, where operating regimes, maintenance actions, component aging, and sensor configurations may change over time. A modular formulation can therefore support incremental model improvement without requiring the complete system model to be redesigned from the beginning.

Despite these advantages, some limitations remain. The simulated turbofan case provides detailed component-level modelling under controlled operating conditions, but simulation data cannot fully reproduce all uncertainties present in real engine operation. The PEMS case addresses this limitation by introducing real industrial measurements; however, the available variables support only a reduced section-level modular structure rather than a detailed component-level gas-path model. In addition, the present study focuses on performance and emission prediction rather than explicit fault diagnosis or degradation prognosis. Future work should evaluate the framework using real gas turbine datasets with richer station-level measurements, degradation conditions, and fault-related operating scenarios. Further extensions could also incorporate physics-informed loss terms when reliable thermodynamic relationships, component-map information, or conservation constraints are available. For example, recent component-level gas turbine compressor Physics Informed Neural Network (PINN) modelling has shown how thermodynamic differential constraints can be embedded in the neural-network loss function to improve physical consistency, and such component-level PINN models could be integrated into a modular system-level architecture in future work [43]. Uncertainty quantification and transfer-learning strategies should also be investigated to improve generalization across different gas turbine configurations.

5. Conclusions

This study proposed a domain-knowledge-guided modular ANN framework for gas turbine performance and emission prediction. The framework decomposes the gas turbine system into physically meaningful modules, trains each module independently, integrates the trained modules according to gas-path and inter-component relationships, and applies sequential fine-tuning to improve integrated system performance.

The simulated high-bypass turbofan case demonstrated the effectiveness of the framework for detailed component-level modular modelling. The results showed that direct integration of independently trained modules can introduce accumulated prediction error, while sequential fine-tuning substantially improves the final system-output accuracy. After fine-tuning, the MAPE values for thrust, TSFC, and OPR were reduced to 2.4909%, 2.3694%, and 0.4966%, respectively. The framework also maintained stable performance under a controlled single-realization sensor-noise perturbation. This result should be interpreted as a controlled perturbation test rather than a full Monte Carlo robustness analysis; multi-realization testing with confidence-interval reporting is left for future work.

The industrial PEMS case further demonstrated that the same modular modelling philosophy can be adapted to real gas turbine operating data using a reduced section-level structure. Sequential fine-tuning improved the final emission predictions, reducing the MAE by approximately 10.6% for CO and 39.2% for NOx. The benchmark comparison showed that the proposed modular ANN achieved competitive accuracy compared with direct input-output machine-learning models and deep-learning benchmark models, while additionally preserving intermediate section-level predictions.

Overall, the proposed framework provides a flexible and interpretable modelling approach for gas turbine systems with different configurations and modelling resolutions. Accordingly, the proposed approach should not be interpreted as a universal fixed module layout; the modular decomposition must be redefined according to the specific gas turbine architecture, available sensor set, measurement resolution, and prediction objective. Its main value lies in combining prediction accuracy with modular traceability, error-propagation analysis, and the possibility of updating individual modules as new data become available. These characteristics make the framework suitable for measurement-oriented gas turbine performance analysis, predictive emission monitoring, and modular digital twin development.

Future work will focus on evaluating the framework using real gas turbine datasets with richer station-level measurements, explicit degradation conditions, and fault-related operating scenarios. Adaptive or automatically learned fine-tuning weights may also be investigated to further balance component-level accuracy and final system-level prediction accuracy under different operating objectives. Further extensions may also include physics-informed loss terms, uncertainty quantification, and transfer-learning strategies to improve generalization across different gas turbine configurations.

Acknowledgments

The authors thank the University of Manitoba, Canada, and Universiti Teknologi PETRONAS, Malaysia, for the facilities provided for this research.

Appendix: Final selected ANN architectures

Table A1. Final selected architecture and training settings of the ANN models for simulated turbofan dataset.

Module	Number	Neurons in	Batch

	of hidden layers	hidden layers	size
Fan	3	15/50/25	512
Bypass section	4	40/30/50/10	512
LPC	2	25/20	512
HPC	4	30/40/45/25	512
Combustor	2	15/50	512
HPT	4	45/20/10/50	512
LPT	3	15/20/50	1024
System-output module	5	40/40/55/45/35	512
Optimized monolithic ANN	3	224/32/128	256

Note: The number of hidden layers refers to Dense hidden layers only and excludes the output layer. ReLU activation was used in all hidden layers. The modular ANN models used sigmoid output activation because the outputs were min-max normalized, whereas the optimized monolithic ANN used a linear output layer. For the modular ANN models, Adam optimizer with the default learning rate of 0.001 was used. For the optimized monolithic ANN, Adam optimizer with a selected learning rate of $3.70\text{x}10^{-4}$ was used. Early-stopping based on validation loss was used during training.

Table A2. Final selected architecture and training settings of the ANN models for PEMS dataset.

Module	Number of hidden layers	Neurons in hidden layers	Batch size
Cold section	4	70/90/50/90	256
Hot section	5	50/20/40/80/40	256
Emission module	4	80/60/70/40	512
Optimized monolithic ANN	5	90/40/50/100/20	256

Note: The number of hidden layers refers to Dense hidden layers only and excludes the output layer. ReLU activation was used in all hidden layers. The PEMS modular ANN models used sigmoid output activation because the outputs were min-max normalized. The Adam optimizer with the default learning rate of 0.001 was used for the PEMS modular ANN models. Early-stopping based on validation loss was used during training.

References

1. *Marinai, L., Probert, D. & Singh, R. Prospects for aero gas-turbine diagnostics: a review. Appl. Energy **79**, 109–126 (2004).*

2. *Cui, B., Ahsan, S., Zhang, J. & Liang, X. An Improved Cox Proportional Hazards Model for Reliability Analysis of Aviation Gas Turbines Considering Varying Environmental Conditions and Operational Settings. Reliab. Eng. Syst. Saf. 111451 (2025) doi:https://doi.org/10.1016/j.ress.2025.111451.*

3. *Li, Y. G. Performance-analysis-based gas turbine diagnostics: A review. Proceedings of the Institution of Mechanical Engineers, Part A: Journal of Power and Energy **216**, 363–377 (2002).*

4. *Kobayashi, T. & Simon, D. L. Integration of on-line and off-line diagnostic algorithms for aircraft engine health management. (2007).*

5. *Simon, D. L. An integrated architecture for on-board aircraft engine performance trend monitoring and gas path fault diagnostics. in 57th JANNAF Joint Propulsion Meeting (2010).*

6. *Kim, S., Kim, N. H. & Choi, J.-H. A study toward appropriate architecture of system-level prognostics: physics-based and data-driven approaches. IEEE Access **9**, 157960–157972 (2021).*

7. *Zhang, Y. et al. Digital twin-driven partial domain adaptation network for intelligent fault diagnosis of rolling bearing. Reliab. Eng. Syst. Saf. **234**, 109186 (2023).*

8. *Xia, M. et al. Intelligent fault diagnosis of machinery using digital twin-assisted deep transfer learning. Reliab. Eng. Syst. Saf. **215**, 107938 (2021).*

9. *Sun, J., Yan, Z., Han, Y., Zhu, X. & Yang, C. Deep learning framework for gas turbine performance digital twin and degradation prognostics from airline*

operator perspective. Reliab. Eng. Syst. Saf. ***238****, 109404 (2023).*

10. *Chen, G., Chen, J., Zi, Y. & Miao, H. Hyper-parameter optimization based nonlinear multistate deterioration modeling for deterioration level assessment and remaining useful life prognostics. Reliab. Eng. Syst. Saf.* ***167****, 517–526 (2017).*

11. *Lee, J. J. & Kim, T. S. Development of a gas turbine performance analysis program and its application. Energy* ***36****, 5274–5285 (2011).*

12. *Zhu, P. & Saravanamuttoo, H. I. H. Simulation of an advanced twin-spool industrial gas turbine. in Turbo Expo: Power for Land, Sea, and Air vol. 79009 V003T07A001 (Citeseer, 1991).*

13. *Haglind, F. & Elmegaard, B. Methodologies for predicting the part-load performance of aero-derivative gas turbines. Energy* ***34****, 1484–1492 (2009).*

14. *Liu, Z. & Karimi, I. A. Gas turbine performance prediction via machine learning. Energy* ***192****, 116627 (2020).*

15. *Salilew, W. M., Karim, Z. A. A. & Lemma, T. A. Investigation of fault detection and isolation accuracy of different Machine learning techniques with different data processing methods for gas turbine. Alexandria Engineering Journal* ***61****, 12635–12651 (2022).*

16. *Xiao, D., Lin, Z., Yu, A., Tang, K. & Xiao, H. Data-driven method embedded physical knowledge for entire lifecycle degradation monitoring in aircraft engines. Reliab. Eng. Syst. Saf.* ***247****, 110100 (2024).*

17. *Tüfekci, P. Prediction of full load electrical power output of a base load operated combined cycle power plant using machine learning methods. International Journal of Electrical Power and Energy Systems* ***60****, 126–140 (2014).*

18. *Fast, M., Assadi, M. & De, S. Development and multi-utility of an ANN model for an industrial gas turbine. Appl. Energy* ***86****, 9–17 (2009).*

19. *Goyal, V., Xu, M., Kapat, J. & Vesely, L. Prediction enhancement of machine learning using time series modeling in gas turbines. J. Eng. Gas Turbine. Power* ***145****, 121005 (2023).*

20. *Nikpey, H., Assadi, M., Breuhaus, P. & Mørkved, P. T. Experimental evaluation and ANN modeling of a recuperative micro gas turbine burning mixtures of natural gas and biogas. Appl. Energy* ***117****, 30–41 (2014).*

21. *Rossi, F., Velázquez, D., Monedero, I. & Biscarri, F. Artificial neural networks and physical modeling for determination of baseline consumption of CHP plants. Expert Syst. Appl.* ***41****, 4658–4669 (2014).*

22. *Amiri, M. H., Hashjin, N. M., Najafabadi, M. K., Beheshti, A. & Khodadadi, N. An innovative data-driven AI approach for detecting and isolating faults in gas turbines at power plants. Expert Syst. Appl.* ***263****, 125497 (2025).*

23. *Hashmi, M. B., Mansouri, M., Fentaye, A. D., Ahsan, S. & Kyprianidis, K. An artificial neural network-based fault diagnostics approach for hydrogen-fueled micro gas turbines. Energies (Basel).* ***17****, 719 (2024).*

24. *Amare, D. F., Aklilu, T. B. & Gilani, S. I. Gas path fault diagnostics using a hybrid intelligent method for industrial gas turbine engines. Journal of the Brazilian Society of Mechanical Sciences and Engineering* ***40****, 1–17 (2018).*

25. *Fentaye, A. D., Ul-Haq Gilani, S. I., Baheta, A. T. & Li, Y.-G. Performance-*

*based fault diagnosis of a gas turbine engine using an integrated support vector machine and artificial neural network method. Proceedings of the Institution of Mechanical Engineers, Part A: Journal of Power and Energy **233**, 786–802 (2019).*

26. *Bai, M., Liu, J., Chai, J., Zhao, X. & Yu, D. Anomaly detection of gas turbines based on normal pattern extraction. Appl. Therm. Eng. **166**, 1–18 (2020).*

27. *Talebi, S. S., Madadi, A., Tousi, A. M. & Kiaee, M. Micro Gas Turbine fault detection and isolation with a combination of Artificial Neural Network and off-design performance analysis. Eng. Appl. Artif. Intell. **113**, 104900 (2022).*

28. *Kaya, H., Tüfekci, P. & Uzun, E. Predicting CO and NOx emissions from gas turbines: Novel data and a benchmark PEMS. Turkish Journal of Electrical Engineering and Computer Sciences **27**, 4783–4796 (2019).*

29. *Yang, C., Yin, X., Wang, H., Li, Z. & Yang, J. Efficient design and optimization of multi-stage axial flow gas turbine using data mining technology. Proceedings of the Institution of Mechanical Engineers, Part A: Journal of Power and Energy 09576509261415972 (2026).*

30. *Zeng, L. et al. Optimization of the design of a heavy-duty gas turbine intake air volute using the surrogate model method. Proceedings of the Institution of Mechanical Engineers, Part A: Journal of Power and Energy **240**, 3–23 (2026).*

31. *Liu, R. & Huang, C. Data-driven modeling approach of heavy-duty gas turbine with physical constraint by MTGNN and Transformer. Control Eng. Pract. **151**, 106014 (2024).*

32. *Lin, Z., Xiao, H., Zhang, X. & Wang, Z. Thrust Prediction of Aircraft Engine Enabled by Fusing Domain Knowledge and Neural Network Model. Aerospace **10**, 493 (2023).*

33. *Zhe, W., Xuyun, F. U., Zhengfeng, B. A. I. & Xiangzhao, X. I. A. Knowledge and data jointly driven aeroengine gas path performance assessment method. Chinese Journal of Aeronautics **37**, 533–557 (2024).*

34. *Bielski, P. et al. Knowledge-Guided Learning of Temporal Dynamics and its Application to Gas Turbines. in Proceedings of the 15th ACM International Conference on Future and Sustainable Energy Systems 279–290 (2024).*

35. *Chen, J. et al. A data-knowledge hybrid driven method for gas turbine gas path diagnosis. Applied Sciences **12**, 5961 (2022).*

36. *Mo, T., Dai, S., Fu, A., Zhu, X. & Li, S. Aeroengine performance prediction using a physical-embedded data-driven method. IEEE Trans. Aerosp. Electron. Syst. (2025).*

37. *He, W., Lin, L., Fu, S. & Liu, D. Performance evaluation method for modules based on organic fusion of data-driven methods and mechanistic knowledge. Advanced Engineering Informatics **69**, 104056 (2026).*

38. *Ahsan, S., Lemma, T. A., Hashmi, M. B. & Liang, X. Investigation of operational settings, environmental conditions, and faults on the gas turbine performance. Meas. Sci. Technol. **35**, 125902 (2024).*

39. *Fentaye, A. D., Baheta, A. T. & Gilani, S. I. U.-H. Gas turbine gas-path fault identification using nested artificial neural networks. Aircraft Engineering*

and Aerospace Technology ***90****, 992–999 (2018).*

40. *Ogaji, S. O. T. & Singh, R. Advanced engine diagnostics using artificial neural networks. Appl. Soft Comput.* ***3****, 259–271 (2003).*

41. *Mohammadi, E. & Montazeri-Gh, M. A fuzzy-based gas turbine fault detection and identification system for full and part-load performance deterioration. Aerosp. Sci. Technol.* ***46****, 82–93 (2015).*

42. *Saxena, A. et al. Metrics for evaluating performance of prognostic techniques. in 2008 international conference on prognostics and health management 1–17 (IEEE, 2008).*

43. *Ahsan, S., Lemma, T. A., Zhang, J. & Liang, X. Physics-Informed Modeling of Gas Turbine Compressors for PHM Applications. in 2026 IEEE International Conference on Prognostics and Health Management (ICPHM) 175–182 (IEEE, 2026).*