Aircraft Engines Performances Estimation from Multi-Point and Multi-Time Operational Data via Neural Networks

Dong Quan Vu
Digital Sciences & Technologies
Safran Tech
Rue des Jeunes Bois, Châteaufort
78118 Magny-Les-Hameaux, France
dong-quan.vu@safrangroup.com

Sebastien Razakarivony
Digital Sciences & Technologies
Safran Tech
Rue des Jeunes Bois, Châteaufort
78118 Magny-Les-Hameaux, France
sebastien.razakarivony@safrangroup.com

Solene Thepaut
Digital Sciences & Technologies
Safran Tech
Rue des Jeunes Bois, Châteaufort
78118 Magny-Les-Hameaux, France
solene.thepaut@safrangroup.com

Guillaume Doquet
Digital Sciences & Technologies
Safran Tech
Rue des Jeunes Bois, Châteaufort
78118 Magny-Les-Hameaux, France
guillaume.doquet@safrangroup.com

Yosra Marnissi
Digital Sciences & Technologies
Safran Tech
Rue des Jeunes Bois, Châteaufort
78118 Magny-Les-Hameaux, France
marnissi.yosra@gmail.com

Michel Nocture
Performance/ Operability
Safran Aircraft Engines,
Moissy-Cramayel, France
michel.nocture@safrangroup.com

***Abstract*—Turbine engine monitoring is an essential enabler for predictive maintenance strategies. Among popular monitoring techniques, estimating performance indicators is a long-standing subject from which a variety of methods have been proposed—including recent applications of machine learning. Traditionally, indicators estimation is viewed under the single-point perspective: at one fixed operating condition, mapping observed measurement data to indicator values. In this perspective, while state-of-the-art methods have promising results in some favorable testing scenarios, none of them have completely resolved *under-determined cases* in which the number of indicators to-be-estimated is larger than the number of observed measurements.**

In this work, we study a use case in monitoring aircraft engines that falls into an under-determined setting. In order to mitigate this, we adapt a multi-point perspective that leverages measurements at multiple pre-determined steady-state operating conditions. It is noteworthy that we relax the condition-independence hypotheses made by previous multi-point models to better align with real scenarios. Unlike previous works under the multi-point perspective, we focus on machine learning methods as resolutions; in particular, we conduct a series of experiments with artificial neural networks in the considered use case. *Our first contribution* is to demonstrate that the estimation precision provided by tested models is significantly improved in the multi-point perspective in comparison with the single-point case. *Secondly*, we explore a scenario involving temporal data (i.e., data from consecutive flight missions) and show that neural networks with appropriate architectures can well exploit such "multi-time" data structure.

***Index Terms*—Turbine Engine Monitoring, Indication Construction, Artificial Neural Networks.**

I. Introduction

Monitoring and analyzing performance of turbine engines has a long history, as long as the engine itself. Performance monitoring is essential in assuring the engines' operation and their safety as well as in optimizing their designs. Moreover, performance monitoring plays a key role in recent developments of predictive maintenance and prognostic activities, especially in the aeronautic industry [1], [2].

There exists a variety of engine performance monitoring methods with a wide range of approaches and principles. For example, elements of engines can be monitored by vibration analysis or by tracking down informative measurable quantities (such as exhaust gas temperature or fuel consumption rate). Among the most popular performance monitoring technique, *constructing and estimating performance indicators from operational data* receives a lot of attention as it does not require operation interruptions and it is capable of giving information at modular levels (hence, allowing a better fault localization and diagnosis). The history of this approach can be tracked back to as early as the 60s and 70s with gas path analysis methods [3], [4]; then in the last several decades, it was further developed with applications of filtering-based algorithms [5], [6] and especially machine-learning oriented methods [7], [8].

In this work, we particularly focus on modular *efficiencies* and modular *corrected air mass flow rates*—these are two most well-used types of performance indicators that are considered in the literature. Typically, they are defined as combinations of physical quantities, such as temperature and pressure, measured at inlet and outlet positions of each engine module with respect to an operational regime [9], [10]. While these indicators are richly informative for maintenance activities, it is costly (and sometimes even impossible) in practice to measure or calculate them directly. Instead, the consensus of the literature is to leverage the sensors' measurements

979-8-3503-5409-6/24/$31.00 ©2024 IEEE
DOI 10.1109/CAI59869.2024.00036

during flight missions to help *indirectly* derive or estimate efficiencies and flow rates. This is known as the turbine engine performance's inverse problem, described as follows.

a) Turbine engine performance's inverse problem: The dominant perspective of the literature is to first establish a forward model S_u (typically, a thermodynamic simulator) at a known operating condition u (e.g., altitude, speed) to map each state x of performance indicators (i.e., values of efficiencies and flow rates) to some selected measurements y, i.e., $y = S_u(x)$; then, given a set of real sensors data y_{real}, one estimates the values of performance indicators corresponding to y_{real} as an *inverse problem* of the S_u model. Several possible approaches are summarized in Section I-B. A practical example is given in Table II.

Table I
A PERFORMANCE INVERSE PROBLEM OF A TURBOFAN ENGINE (FROM LEFT TO RIGHT) AND ITS ASSOCIATED FORWARD MODEL (FROM RIGHT TO LEFT). NOTATION: η = EFFICIENCY, Γ = AIR MASS FLOW RATE, TURB = TURBINE.

Measurable quantities		Performance state
Core rotation speed (N2)		η Fan
Compressor inlet temperature (T25)		Γ Fan
Compressor outlet temperature (T3)	$\rightarrow$	η Booster
Exhaust gas temperature (EGT)	$\leftarrow$	Γ Booster
Compressor outlet pressure (PS3)		η Compressor
Combustion chamber fuel flow (WF)		Γ Compressor
		η Low-pressure Turb
		Γ Low-pressure Turb
		η High-pressure Turb
		Γ High-pressure Turb

b) A case study and main challenges: In practice, the number of sensors available on aircraft engines is often very limited due to physical constraints (temperature, space, and weight) and so is the number of corresponding measurable quantities. As a consequence, the inverse problem defined above often falls into an *under-determination* setting as the number of indicators to-be-estimated is larger than the dimension of data. In this work, we consider the following real-life under-determined scenario (see Section II-B for more details):

How to better estimate efficiencies and flow rates of each of the 5 modules of an aircraft engine (i.e., 10 indicators to-be-estimated) from data coming from 6 sensors measured at a handful of operating conditions?

While state-of-the-art methods, including machine-learning-based approaches, have promising results in some favorable testing scenarios, their performances are subpar in such an under-determination case (see e.g., turbofan example in [7]). This is one of the reasons why there is a contradiction between the diversity of methods proposed and a limited number of the algorithms applied in real systems.

c) Multi-point approach: A workaround solution that somewhat mitigates the difficulties of under-determination is the so-called multi-point perspective proposed by Stamatis [11] and further developed in [12]–[14]. The key principle is to leverage sensors' measurements at different operating conditions. While this approach increases the dimension of measurements data, it relies upon hypotheses about certain types of independence of efficiencies and flow rates with respect to operating conditions. For example, the differences between values of these indicators and a nominal reference are assumed to remain constant during a flight, regardless of operating conditions [11]. While there exist scenarios where such assumptions might hold (e.g., deformation of indicators w.r.t. engines' physical wear), they are theoretically questionable as definitions of efficiencies depend on quantities of operating conditions. We will further discuss this issue in Section II-A.

A. Contributions

In this work, we investigate the uses of several standard machine learning techniques in resolving the engine performance's inverse problem under a multi-point perspective, with the aircraft engine use case described above as a demonstration example.

Our first contribution is to conduct a series of numerical experiments to test the performance of several artificial neural network (ANN) models in our considered use case. In particular, we show a steady improvement in performance of tested ANN models with multi-point data than the case with classic single-point measurements. We also show that different combinations of measurements have different impacts on algorithmic performances. We select a neural network model trained with data at 4 operating conditions (takeoff, cruise and 2 climbs conditions) as a show case and analyze the respective results.

Secondly, we consider a scenario where temporal data is available; in particular, when we have access to measurements from two different flight missions of the same engine. We name this the *multi-time* scenario. Instead of using previous machine models to separately estimate the values of performance indicators at these missions, we provide an alternative framework in estimating the variation between the engine's states in these missions. We analyze the advantages of this multi-time framework and investigate some potential approaches in exploiting more measurements data.

B. Literature Review

In this section, we briefly present the main resolution methods existing in the literature of the turbine engine performance estimation problem.

a) Gas Path Analysis (GPA): GPA refers to the group of strategies searching for a best match (in an optimization framework) between measurements changes and associated indicators changes. Linear GPA (LGPA) [3], and then its non Linear versions [15], [16] relying on a linearization of the system, perform well in a number of favorable scenarios [17]–[19]. However, GPA underperforms in under-determined systems, and is subject to the "smearing effect" [1] [13], [15] and it unavoidably depends on the initialized point. Despite this, NLGPA remains the basic framework for several other methods, including multi-point techniques (discussed below).

[1]GPA tends to smear the defaults of one module over the others.

b) Bayesian Filtering: When time-series data are available, a well-known approach is look at the estimation as a filtering problem, notably, using Bayesian Filtering (BF). It requires an a priori degradation model and an observation model and balances between these models' predictions with the real observations in a probabilistic framework. The most well-used examples are the Kalman Filters (KF), with many years of publications on the subject [5], [20]–[25]. and particle filter (PF) [26]–[29]. BF methods strongly rely on their a priori models. KF is optimal in linear cases but falls short in systems with a high level of non-linearity (even with non-linear versions of KF such as extended KF or unscented KF) and PF has a high computational cost due to treating multiple particles at each time step.

c) Neural Networks: The popularity of neural networks (NN) methods have grown exponentially in many fields including in the engine performance's inverse problem [30]–[33]. This approach relies on leveraging physical models to build a training dataset. Once this is obtained, the performance indicators estimation task is posed as a regression problem, where measurement data are the inputs and indicators estimations are the outputs. Recent works using NN show promising results in several tested cases. In particular, De Giorgi et al. [30], [31] use a simple one-hidden-layer neural network to estimate discretized values of performance indicators. On the other hand, estimations made by NN are shown to outperform a polynomial regression in [8] and [33]. Similarly, Menga et al. [32] use Extreme Machine Learning to obatin a relative error around 5% on efficiencies.

In general, once an NN model is trained, it gives advantages in execution time and does not require knowledge of physical models or additional tuning. Importantly, it is observed that NN's precision declines in such scenarios but the relation between measurements and observability is not addressed [33].

d) Multi-point perspective: Multi-point technique also has a long history, traced back to [11], [34]. These works assume an independence of performance indicators on operating conditions. This hypothesis is considered to be questionable by later works [13], [14]. To mitigate this, [13], [14] and its more recent development [35]–[38] proposed the artificial multi operating point analysis (AMOPA) variant based on the so-called artificial operating conditions. This approach discards the independence hypothesis; however, it still has limitations as not all artificial points are meaningful [14].

Pinelli et al. [12] propose another relaxation of multi-point perspective that aims at estimating deformation coefficients [39] (also known as scalar scale factors) of efficiencies and flow rates. In this context, a component map, which consists of a series of curves representing the performances of a specific reference engine, serves as benchmarks and the scalar coefficients reflect how this performance map is deformed in the event of degradation. Unlike other classic multi-point approaches, Pinelli et al. only impose the independence hypothesis on scalar coefficients. Therefore, efficiencies and flow rates are permitted to vary across different operating conditions, but the amount of changes in each operating condition remains constant. It is noteworthy that all state-of-the-art works that leverage the multi-point perspective focus solely on gas-path analysis methods.

Despite considerable advantages, neural networks underperform in under-determined cases [8], [33] and have not been applied to multi-point and/or multi-time scenarios.

II. Estimation of Performance Indicators with Operational Data

A. Performance Indicators and Scalar Coefficients

a) Classic definition of efficiencies and flow rates: In the literature of engine performance's inverse problem, two most well-used types of indicators are modular efficiencies and modular corrected air mass flow rates. Hereinafter, for the sake of brevity, we refer to them simply as *efficiencies* and *flow rates*. Intuitively, efficiency is defined as the ratio between work done by a module and the energy it receives and flow rate is the mass of substances passing a module per unit of time (normalized to ambient conditions). Their specific definitions vary between different modules and across different types of engines (see [9], [10] for concrete examples).

In this work, we will not delve into explicit formulations of efficiencies and flow rates, and only keep a general form for conceptualization purpose. In particular, let θ_M denote an ambiguous degradation state of a module M (e.g., fan, booster, compressor, turbines) and $\theta := (\theta_M)_{\forall M}$ is the state of the whole engine. At each state θ, we denote

- u a (multi-dimensional) vector reflecting measurable exogenous physical quantities (e.g., ambient pressure, flight speed), often called *operating conditions*.
- y_u^θ a (multi-dimensional) variable representing measurable endogenous physical quantities (e.g., inlet and outlet temperature of a module)—often called *measurements*,

Denote by $\mathrm{H}_{u,M}^\theta$ a performance indicator (either efficiency or flow rate) at a module M of an engine being in a state θ and operating in a condition u, in general, it can be defined by the following relation:

$$\mathrm{H}_{u,M}^\theta = g_{u,M}\left(y_u^\theta, \omega\right), \tag{1}$$

where $g_{u,M}$ denotes a function representing certain thermodynamic relations and ω denotes other related quantities that are not measured in the system (i.e., not included in y or u), which can be intuitively understood as factors generating uncertainties of the considered estimation problem.

In *classic engine performance's inverse problem* [33], [40], the purpose is to estimate values of indicators in a set $\{\mathrm{H}_{u,M}^\theta, \forall M\}$ (for an unknown θ) given a data point $\left(y_u^\theta, u\right)$.

b) Deformation coefficients of component map: In reality, it is impossible to track $\mathrm{H}_{u,M}^\theta$ at all possible conditions and modules' states. Instead, a common practice is to first establish *a reference component map*, representing relations between efficiencies, flow rates and other relevant quantities, at a reference state θ_{ref} (e.g., a new and clean engine) and some pre-determined conditions u_{ref}; then, apply a model on

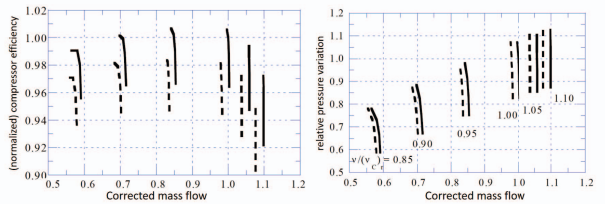

Figure 1. A compressor map (adapted from [12]): Efficiency and corrected mass flow in "new and clean" condition (solid line) and in a degradation state with deformation coefficient 0.02 (dash line).

how such component map is deformed with variations in degradation states.

In particular, the dominant perspective of the literature is to hypothesize that when θ changes, all representation curves in the component map are *shifted uniformly* with respect to that of θ_{ref}. These shifts are characterized by a scalar, namely the deformation coefficient. Mathematically, let us denote η_M^θ the deformation coefficient (associated with an indicator H_M^θ), we have the generic relation:

$$\mathrm{H}_{u_{\text{ref}},M}^{\theta} = p_{u_{\text{ref}},M}^{\text{shift}}\left(\eta_M^\theta, \mathrm{H}_{u_{\text{ref}},M}^{\theta_{\text{ref}}}\right), \tag{2}$$

Here, $p_{u_{\text{ref}},M}^{\text{shift}}$ is an empirical model on the shifts of component maps. An example is given in Figure 1. As a consequence, *given values of the scalar deformation coefficients* η_M^θ, *it is possible to retrieve the value* $\mathrm{H}_{u,M}^{\theta}$ *of the classic indicator* (i.e., efficiencies or flow rates) as long as the corresponding component map of the reference state is available with operating condition u.

B. Problem Statement: An Aircraft Engine Use Case

In this work, we investigate a use case of estimating performance indicators of a turbofan engine from operational data. In particular, we have access to a dataset $\mathcal{D}^\theta = \left\{\left(y_1^\theta, u_1\right), \ldots, \left(y_N^\theta, u_N\right)\right\}$ recording the measurements and operating conditions of the engine at an unknown degradation state θ. In particular, each data point containing 6 measurements of 6 sensors (i.e., $y_i^\theta \in \mathbb{R}^6, \forall i$) and each operating condition is a pre-determined steady state in one of flying phases (such as takeoff, climbs, and cruise); the details are given in Table II. Note that by this formulation of $\mathcal{D}^\theta$, we implicitly assume that there exists a time period that the considered engine is operated in all conditions $u_1, \ldots, u_N$ without a change in engine's degradation states (e.g., remaining as constant θ during a flight).

We assume that there exist deformation coefficients η_M^θ, corresponding component maps and known functions $p_{u_i,M}^{\text{shift}}$ for each $i \in [N]$ such that one can retrieve efficiencies and flow rates of the engine corresponding to D^θ by following Equations 2 (and replacing u_{ref} by u_i).

From the data set D^θ, *we aim to estimate* η^θ*—a vector representing 10 deformation coefficients* associated with efficiencies and flow rates of five modules: fan, booster, compressor, low-pressure turbine and high-pressure turbine.

Table II
MEASUREMENTS IN THE CONSIDERED USE CASE.

Notation	Measurable quantities	Unit
N2	core rotation speed	rpm/s
T25	compressor inlet temperature	°K
T3	compressor outlet temperature	°K
EGT	exhaust gas temperature	°K
PS3	compressor outlet pressure	bar
WF	combustion chamber fuel flow	kg/s

Finally, we assume the availability of a *thermodynamic simulator*, denoted S, of the considered engine. In particular, given pre-determined values of η and a steady operation condition u, S gives simulated values y. In the sequel, we will use S to construct a dataset for training and validating the performance of several machine learning models.

Remark 1 (Under-determination)**.** The classic single-point perspective aims at estimating efficiencies and flow rates which, by definition, vary according to operation conditions' changes. As a consequence, if we consider the problem described above in this perspective, for *each point* (y_i^θ, u_i) in the dataset—which is a 6-dimensional vector, it requires to estimate 10 to-be-estimated variables (of the form $\mathrm{H}_{u_i,M}^{\theta}$). It is clear that the problem described above falls into an under-determined setting.

Remark 2 (Multi-point perspective and relaxed hypotheses)**.** The model described above follows a multi-point perspective, i.e., we will leverage simultaneously all the points in the D^θ dataset. In other words, for each dataset D^θ—which is a $(6 \times N)$-dimensional vector, we estimate 10 variables which are the deformation coefficients.
It is also noteworthy that previous works in multi-point perspective and deformation coefficients [12] assume a shifting model uniformly across all considered operating conditions (i.e., in Equation 2, $p_{u_{\text{ref}},M}^{\text{shift}}$ is fixed regardless of u_{ref}). In this work, we do not use this assumption as we observe in our empirical data that shifts at various operating conditions can differ from one another.

Remark 3 (Comparison with state-of-the-art machine learning approaches)**.** As discussed in Section I-B, previous works that use NN as a resolution for the engine performance's inverse problem only follows the single-point perspective [8], [30]–[33]. As a consequence, applying directly such approaches in under-determined cases, such as our use case, will encounter difficulties. To highlight this, in Section III, we will demonstrate the under-performance of NN in the case where $N = 1$ (i.e., there is one data point in D^θ) which is equivalent to the single-point perspective.
Moreover, previous works often aim at establishing a surrogate model for a thermodynamic simulator; hence, the tested NN model is trained with data at many different operating conditions. This is questionable because most of available simulators are only stable at steady (designed) states and in practice, data are rarely abundant as they are recorded by on-board

equipment. In particular, in our use case, measurements data are only available at a handful of pre-determined operation conditions (steady states). Note that in this work, we do not focus on the problem of selecting operating conditions which should be considered as a pre-processing task; see e.g., [13] for a discussion.

III. Experimental Results: Machine Learning Methods in Estimating Performance Indicators

We first conduct a series of experiments to investigate the performance of neural networks in our considered use case of engine performance's inverse problems. Note that in all the experiments reported below (and in Section IV), we let the tested neural networks converge with a common stopping criterion on validation losses (in mean absolute errors).

a) Data generation: We generate several datasets by using a thermodynamic simulator—designed and operated by Safran (based on Janus language)—of a turbofan engine. As previously stated, this simulator, served as a physical model, can be considered as a forward model of the inverse problem at hand; in other words, we first choose a set of values of deformation coefficients and a set of operating conditions then input these quantities into the simulator to retrieve the simulated data of measurements.

In particular, in this work, we focus on a set, denoted by $\mathcal{U}$, containing 4 specific operating conditions namely Takeoff, Cruise, Climb1 and Climb2. In particular, each operating condition carries the information of quantities such as ambient temperature, ambient pressure, flight speed (mach), pilot's control, power extraction and valves' position.

In the proposed experiments, we will construct different scenarios by selecting specific subsets of $\mathcal{U}$ when generating data. Additionally, for each $U \subset \mathcal{U}$, we create a training dataset of size $S_{\text{train}} = 10e4$ and a test set of size $S_{\text{test}} = 2 \times 10e3$ by sampling values of deformation coefficients via a Latin hypercube sampling scheme [41] in the range $[0.95, 1.00]^{10}$. It is noteworthy that by this sampling scheme, we ensure that there is no intersection between training and test sets. We summarize the data generation step in Figure 2.

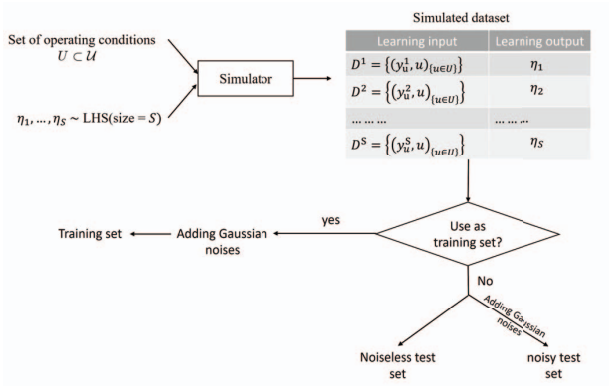

Figure 2. Data generation via a thermodynamic simulator including the processing of constructing training and test sets.

In practice, measurements data are subjected to noises (due to the uncertainties of sensors and unknown factors surrounding the system). To better simulate real-life situations, we incorporate such noise into our dataset. Specifically, for all training sets, we introduce independent zero-mean Gaussian noise to each measurement element. The variances of these noises are determined based on empirical data from related sensors. Additionally, we create two types of test sets: one without any added noise and the other with zero-mean Gaussian noises similar to those introduced in the training set. Note that most state-of-the-art works following neural networks approach in estimating engine performance indicators [31], [33] are only tested with noiseless simulated data.

Finally, note that in this work, we only tested the concerned methods on a handful of instances of possible datasets (and report below results corresponding to only one of them); in future works, we will conduct more complete tests with the variations of involved elements and report more complete results in these cases.

b) Choices of neural network architectures: In all subsequent experiments, unless indicated otherwise, we report the results from a simple neural network, having 3 hidden layers of size 100, integrated with RELU activation functions, Adam optimizer and L1-loss, with dropout (probability=0.2) on the last layer. Note that in all these experiments, we also tested other neural networks by varying other hyperparameters such as the size (100, 200, 300 nodes per layer) and the depth (up to 10 hidden layers), as well as the drop-out probability, activation functions and types of optimizer (e.g., with a step-size scheduler); we also alternatively trained these networks with the mean squared error loss instead of L1-loss. As these results are quite similar to that of the network mentioned above and due to space constraint, we opt to only present results of this simple network. At the moment, we do not see compelling evidences for the needs in testing with more complicated and advanced architectures (given the trade-off in their increasing complexities) and leave this as a future research direction.

c) Experiment 1: double-point data: In the first experiment, for each subset of $\mathcal{U}$ that contains *two operating conditions*, we construct a training dataset and test sets; each having a 12-dimensional input (corresponding to measurements at the two operating conditions) and 10-dimensional output. We call this the *double-point* scenario. In Figure 3, we report the mean absolute error (MAE) of estimations given by our chosen neural network. From this figure, we observe that different combinations of data associated with different combinations of operating conditions have different impacts on the estimation errors of the tested neural network. In particular, among all double-point data, a combination of measurements at Takeoff and Climb2 conditions helps our neural networks have the best precision.

d) Experiment 2: variations in sizes of multi-point data: In contrast to the previous experiment where the number of operating conditions (N) remained fixed, this experiment compares the performance of neural networks trained with a varying N. Figure 4 illustrates the estimation errors of tested

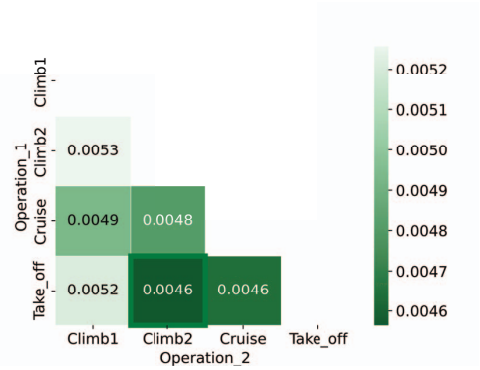

Figure 3. Estimation errors of neural networks trained with double-point (noisy) data and validate with noiseless test data.

neural networks trained with single-point (data from only one operating condition, therefore being under-determined), double-point, triple-point (data from a set of 3 operating conditions), and quadruple-point data (data from all 4 operating conditions in $\mathcal{U}$). The results demonstrate that as N increases, the performances of the associated neural networks improve. This aligns with the intuitive expectation that training with more information about the system reduces the challenges posed by the under-determined setting.

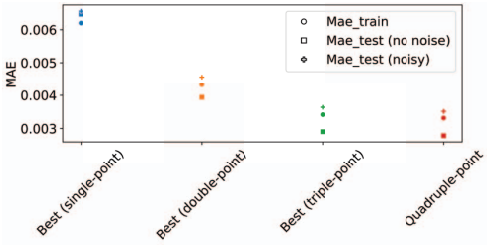

Figure 4. Estimation errors of neural networks as the number of operating condition N varies. We only report the results associated with the combination that gives the best precision among all scenarios in *each* type: single-point, double-point and triple-point data.

e) Experiment 3: Further analysis in quadruple-point scenario: Finally, we select the most favorable scenario in our previously tested case, i.e., when we have access to measurements at all 4 operating conditions in $\mathcal{U}$. In order to highlight the improvements given by this particular multi-point scenario, in Table III, we report the estimation errors corresponding to each considered deformation coefficient. From these results, we observe that our tested neural network can obtain acceptable precision in estimating most of the required indicators (and are significantly improved in comparison to single-point perspective). The two cases with the highest errors are the coefficients corresponding to efficiencies of fan and flow rates of low-pressure turbine. This reflects the fact that measurements in our dataset (even enriched via the multi-point perspective) do not give tremendous information about these modules.

Table III
ESTIMATION ERRORS OF NEURAL NETWORKS TRAINED WITH QUADRUPLE-POINT DATA: MAE ON TRAINING SET, NOISELESS TEST SET AND NOISY TEST SET. IN PARENTHESES: IMPROVEMENTS IN COMPARISON TO NETWORKS (OF THE SAME ARCHITECTURE) TRAINED WITH SINGLE-POINT DATA.

Elements of η	MAE train	MAE test (no noise)	MAE test (noisy)
η^E Fan	0.00786 (-0.0034)	0.00806 (-0.0043)	0.00917 (-0.0032)
η^F Fan	0.00028 (-0.0044)	0.00021 (-0.005)	0.0003 (-0.0049)
η^E Booster	0.00283 (-0.0040)	0.00178 (-0.0062)	0.00309 (-0.0049)
η^F Booster	0.00136 (-0.0017)	0.00087 (-0.0027)	0.00148 (-0.0021)
η^E Compr	0.00127 (-0.0017)	0.00081 (-0.0026)	0.00137 (-0.0021)
η^F Compr	0.00218 (-0.0029)	0.00133 (-0.0045)	0.00235 (-0.0035)
η^E HPTurb	0.00385 (-0.0015)	0.00385 (-0.0022)	0.00445 (-0.0016)
η^F HPTurb	0.00054 (-0.0004)	0.00015 (-0.0001)	0.00055 (-0.0004)
η^E LPTurb	0.00458 (-0.0061)	0.00335 (-0.0085)	0.00494 (-0.0069)
η^F LPTurb	0.00642 (-0.0027)	0.00656 (-0.0038)	0.00747 (-0.0029)

Notation: η^E = deformation coefficient of efficiency, η^F = deformation coefficient of flow rate, Compr = compressor, HPTurb = high-pressure turbine, LPTurb = low-pressure turbine.

As a conclusion, in this initial set of experiments, we showed that data derived from various combinations of operating conditions significantly influences the precision of the tested models. Particularly, dataset encompassing measurements from a higher number of operating conditions offers an improved performance (notably, training in multi-point data is significantly better than using only single-point data).

IV. LEVERAGING MULTI-TIME DATA

In Section III, we explored a scenario wherein data remains static: each data point represents a single degradation state of the engine. For brevity, we will refer to this as *"single-time"* data in subsequent sections.

On the contrary, in real-life scenario, we often have access to time-series operational data, i.e., data from different flight missions of the same engine (that correspond to different degradation states). A straightforward approach involves treating these data points as independent and using neural networks trained on single-time data to estimate them separately (as discussed in the previous section). However, this method disregards temporal information entirely. As a consequence, a natural question arises: *"Can we enhance the performance of neural networks by treating measurements at different times of an engine as a whole data point for learning?"*

To distinguish with single-time scenario, we refer to this way of exploiting time-series data as the *"multi-time" perspective*. In this section, we investigate some approaches in using neural networks with multi-time data. As a demonstration, we will focus on comparing single-time quadruple-point and double-time quadruple-point scenarios (with the set $\mathcal{U}$). It is noteworthy that while such temporal information is exploited by Bayesian Filtering methods (see Section I), existing neural network methods have yet to address this aspect.

a) Multi-time data generation: We construct a new dataset corresponding to *pairs* of degradation states: the first state being at a specific instant in the life of the engine, and the second one being a degradation of this engine after 100 flights. Mathematically, we first sample a value, called

η_1, of deformation coefficients (similarly to previous experiments, we also leverage Latin hypercube sampling). Then, we generate the degradation directions, i.e., the 10-dimensional vectors indicating how performance indicators in η_1 decrease after 100 flights (for this reason, all these directions belong to the $\mathbb{R}^{10}_{\leq 0}$ space). In particular, we select ten degradation directions Δ at random such that $\Delta \sim \mathcal{N}(\lambda, \text{var}_\lambda)$ where $\lambda \in \mathbb{R}^{10}_{\leq 0}$ represents the associated mean and var_λ represents the associated variance (according to our empirical data). For each pair of η_1 and Δ, we compute

$$\eta_2 = \eta_1 + \Delta. \tag{3}$$

Finally, we input η_1 and η_2, together with the operating set $\mathcal{U}$ to the simulator S to achieve a $(6 \times N \times 2)$-dimensional measurement data. We call such this the *double-time* dataset.

b) Experiment 4: Comparisons of estimations made from single-time and double-time data: In this experiment, we construct a new neural network model, having 3 hidden layers of size 200, integrated with RELU activation functions, Adam optimizer and L1-loss, with dropout (probability=0.2) on the last layer to train with double-time data. Note that this architecture was also used in the single-time setting (but with a slightly bigger size since the input is twice the size).

In Figure 5, we report the estimation errors given by this new neural network in estimating either η_1 or η_2 in comparison with that of the neural network trained on single-time data described in Section III. We see clearly that the performance is improved by using multi-time data. This aligns with the intuition that multi-time data is more informative than single-time data. We can also see that η_2 is slightly more difficult to estimate than η_1, which is expected as η_2 has more variations than η_1 due to our choice of generating data.

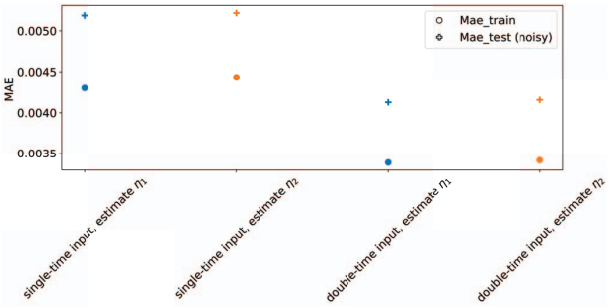

Figure 5. Estimation error in using single-time and double-time input to learn single-time output.

c) Experiment 5: Learning to give double-time estimations: In this experiment, we explore several new neural networks employing a similar architecture to the previous ones but designed to generate outputs of different sizes. In particular, we construct a network learning to estimate a 20-dimensional vector (η_1, η_2) and another network learning to estimate another 20-dimensional vector of the form (η_1, Δ) (Note that all neural networks described before this point have a 10-dimensional output). In words, the former aims at simultaneously give estimations of both values of deformation coefficients in the pair of degradation states and the latter aims at estimating the first state and the degradation speed (that will be used to derive the second state).

In Figure 5, we report the estimation errors given by these two neural networks. We saw that adding explicit information about the Δ in output does not improve performances.

As a conclusion regarding multi-time data, we observe that letting a network learning point-by-point (with double-time input or single-time input) is better than learning simultaneously both points of η_1 and η_2. Our hypothesis is that to have the same degree of accuracy, the considered networks should be bigger as the size of output is larger.

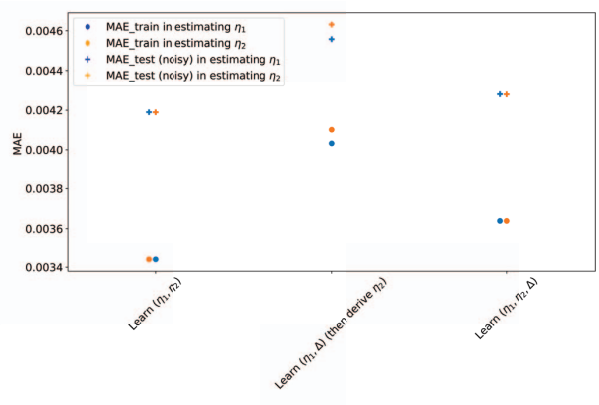

Figure 6. Estimation error in using double-time input to learn double-time output.

V. Conclusion

In this work, we study a use case of engine performance's inverse problem. While previous works have employed neural networks to tackle similar issues, they primarily focused on the single-point perspective resulting in an under-determined setting and leading to subpar performances. In order to mitigate this, we adapt a multi-point perspective to train neural networks with more enriched data. We demonstrate, by conducting a series of experiments, that the estimation precision provided by tested models is significantly improved in the multi-point perspective in comparison with the single-point case. Finally, we also investigate the usage of several possible approaches in leveraging multi-time data, which also resulting in improved performance of certain tested neural networks.

We note that in this study, our experiments were confined to operating conditions within a pre-determined set representing the operating conditions observed in the real-life scenario that we investigated. We speculate that the performance of neural networks could be augmented by training them with data from a wider range of operating conditions. We leave this for future investigations. Similarly, to better exploit time-series data, a possible approach for future study is to use Bayesian filters (such as Kalman filter or particle filter)—that are classic in

the application context of engine monitoring—to post-process outputs of neural networks in order to take the best-of-both-world advantages.

REFERENCES

[1] H. Hanachi, C. Mechefske, J. Liu, A. Banerjee, and Y. Chen, "Performance-based gas turbine health monitoring, diagnostics, and prognostics: A survey," *IEEE Transactions on Reliability*, vol. 67, no. 3, pp. 1340–1363, 2018.

[2] A. D. Fentaye, A. T. Baheta, S. I. Gilani, and K. G. Kyprianidis, "A review on gas turbine gas-path diagnostics: State-of-the-art methods, challenges and opportunities," *Aerospace*, vol. 6, no. 7, p. 83, 2019.

[3] L. A. Urban, *Gas turbine engine parameter interrelationships*. Hamilton Standard Division of United Aircraft Corporation, 1969.

[4] ——, "Gas path analysis applied to turbine engine condition monitoring," *Journal of Aircraft*, vol. 10, no. 7, pp. 400–406, 1973.

[5] A. J. Volponi, H. DePold, R. Ganguli, and C. Daguang, "The use of kalman filter and neural network methodologies in gas turbine performance diagnostics: a comparative study," *J. Eng. Gas Turbines Power*, vol. 125, no. 4, pp. 917–924, 2003.

[6] D. Simon, "A comparison of filtering approaches for aircraft engine health estimation," *Aerospace Science and technology*, vol. 12, no. 4, pp. 276–284, 2008.

[7] I. Loboda and Y. Feldshteyn, "Polynomials and neural networks for gas turbine monitoring: a comparative study," 2011.

[8] I. G. Castillo, I. Loboda, and J. L. Pérez Ruiz, "Data-driven models for gas turbine online diagnosis," *Machines*, vol. 9, no. 12, p. 372, 2021.

[9] J. R. Howell and R. O. Buckius, *Fundamentals of engineering thermodynamics*. McGraw-Hill Companies, 1987, vol. 2.

[10] J. Holman, *Thermodynamics*. McGraw-Hill, 1988.

[11] A. Stamatis, K. Mathioudakis, G. Berios, and K. Papailiou, "Jet engine fault detection with discrete operating points gas path analysis," *Journal of Propulsion and Power*, vol. 7, no. 6, pp. 1043–1048, 1991.

[12] M. Pinelli, P. R. Spina, and M. Venturini, "Optimized operating point selection for gas turbine health state analysis by using a multi-point technique," in *Turbo Expo: Power for Land, Sea, and Air*, vol. 36878, 2003, pp. 43–51.

[13] A. Stamatis, "Optimum use of existing sensor information for gas turbine diagnostics," in *Turbo Expo: Power for Land, Sea, and Air*, vol. 43123, 2008, pp. 47–54.

[14] ——, "Evaluation of gas path analysis methods for gas turbine diagnosis," *Journal of Mechanical Science and Technology*, vol. 25, pp. 469–477, 2011.

[15] D. L. Doel, "An assessment of weighted-least-squares-based gas path analysis," 1994.

[16] P. Escher, "Pythia: An object-orientated gas path analysis computer program for general applications," 1995.

[17] H. Hanachi, J. Liu, A. Banerjee, and Y. Chen, "A framework with nonlinear system model and nonparametric noise for gas turbine degradation state estimation," *Measurement Science and Technology*, vol. 26, no. 6, p. 065604, 2015.

[18] Y. Li and P. Nilkitsaranont, "Gas turbine performance prognostic for condition-based maintenance," *Applied energy*, vol. 86, no. 10, pp. 2152–2161, 2009.

[19] S. Ogaji, S. Sampath, R. Singh, and S. Probert, "Parameter selection for diagnosing a gas-turbine's performance-deterioration," *Applied energy*, vol. 73, no. 1, pp. 25–46, 2002.

[20] D. Simon and D. L. Simon, "Kalman filter constraint switching for turbofan engine health estimation," *European journal of control*, vol. 12, no. 3, pp. 331–343, 2006.

[21] ——, "Constrained kalman filtering via density function truncation for turbofan engine health estimation," *International Journal of Systems Science*, vol. 41, no. 2, pp. 159–171, 2010.

[22] J. S. Litt, "An optimal orthogonal decomposition method for kalman filter-based turbofan engine thrust estimation," 2008.

[23] T. Kobayashi and D. L. Simon, "Hybrid kalman filter: a new approach for aircraft engine in-flight diagnostics," Tech. Rep., 2006.

[24] F. Lu, J.-q. Huang, C.-s. Ji, D.-d. Zhang, and H.-b. Jiao, "Gas path online fault diagnostics using a nonlinear integrated model for gas turbine engines," *International Journal of Turbo & Jet-Engines*, vol. 31, no. 3, pp. 261–275, 2014.

[25] F. Lu, T. Gao, J. Huang, and X. Qiu, "Nonlinear kalman filters for aircraft engine gas path health estimation with measurement uncertainty," *Aerospace Science and Technology*, vol. 76, pp. 126–140, 2018.

[26] F. Lu, J. Huang, and Y. Lv, "Gas path health monitoring for a turbofan engine based on a nonlinear filtering approach," *Energies*, vol. 6, no. 1, pp. 492–513, 2013.

[27] N. Daroogheh, N. Meskin, and K. Khorasani, "Particle filtering for state and parameter estimation in gas turbine engine fault diagnostics," in *2013 American Control Conference*. IEEE, 2013, pp. 4343–4349.

[28] Q. Wang, J. Huang, and F. Lu, "An improved particle filtering algorithm for aircraft engine gas-path fault diagnosis," *Advances in Mechanical Engineering*, vol. 8, no. 7, p. 1687814016659602, 2016.

[29] D. Q. Vu, Y. Marnissi, S. Razakarivony, and M. Nocture, "A constrained langevin-adapted particle filter for aircraft engines' health monitoring," in *2023 IEEE Conference on Artificial Intelligence (CAI)*. IEEE, 2023, pp. 183–184.

[30] M. G. De Giorgi, S. Campilongo, and A. Ficarella, "A diagnostics tool for aero-engines health monitoring using machine learning technique," *Energy Procedia*, vol. 148, pp. 860–867, 2018.

[31] M. G. De Giorgi, L. Strafella, N. Menga, and A. Ficarella, "Intelligent combined neural network and kernel principal component analysis tool for engine health monitoring purposes," *Aerospace*, vol. 9, no. 3, p. 118, 2022.

[32] N. Menga, A. Mothakani, M. G. De Giorgi, R. Przysowa, and A. Ficarella, "Extreme learning machine-based diagnostics for component degradation in a microturbine," *Energies*, vol. 15, no. 19, p. 7304, 2022.

[33] I. Loboda, I. González Castillo, S. Yepifanov, and R. Zelenskyi, "Nonlinear surrogate models for gas turbine diagnosis," in *Turbo Expo: Power for Land, Sea, and Air*, vol. 85987. American Society of Mechanical Engineers, 2022, p. V002T05A024.

[34] A. Stamatis and K. Papailiou, "Discrete operating conditions gas path analysis," in *AGARD conference proceedings*, no. 448, 1988, pp. 33–1.

[35] R. Salamat, "Gas path diagnostics for compressors," 2012.

[36] T. Rootliep, "Turbofan condition monitoring using evolutionary algorithm based gas path analysis: at klm engine services," 2020.

[37] T. Rootliep, W. Visser, and M. Nollet, "Evolutionary algorithm for enhanced gas path analysis in turbofan engines," in *Turbo Expo: Power for Land, Sea, and Air*, vol. 84898. American Society of Mechanical Engineers, 2021, p. V001T01A011.

[38] H. Qin, J. Zhao, L. Ren, B. Li, and Z. Li, "Performance degradation evaluation of low bypass ratio turbofan engine based on flight data," *Sustainability*, vol. 14, no. 13, p. 8052, 2022.

[39] R. Bettocchi and P. Spina, *Diagnosis of gas turbine operating conditions by means of the inverse cycle calculation*. American Society of Mechanical Engineers, 1999, vol. 78590.

[40] A. Volponi, "Foundation of gas path analysis (part i and ii)," *Gas Turbine Condition Monitoring and Fault Diagnosis*, vol. 1, 2003.

[41] B. Tang, "Orthogonal array-based latin hypercubes," *Journal of the American statistical association*, vol. 88, no. 424, pp. 1392–1397, 1993.