UNIVERSITÀ DI GENOVA

SCUOLA POLITECNICA
DIME

Dipartimento di Ingegneria Meccanica, Energetica,
Gestionale e dei Trasporti

DOCTORATE IN:
ENGINEERING OF MACHINES AND SYSTEMS FOR
ENERGY, ENVIRONMENT AND TRANSPORT
Curriculum: Engineering of machines and systems for energy, environment and propulsion

CFD-Informed Methods for Condition Monitoring and Diagnostics of Turbomachinery

Supervisors:
Prof. Carlo Cravero
Dr. Michelangelo Mortello

Student:
Lorenzo Carrattieri

23 June 2026

Abstract

Turbomachinery plays a central role in industrial energy conversion systems, where reliable operation and early detection of performance degradation are essential for ensuring efficiency, safety and reduced operational costs. This thesis develops methods informed by Computational Fluid Dynamics (CFD) for condition monitoring and diagnostics of turbomachines operating under realistic industrial constraints. The research was conducted within an industrial PhD programme in collaboration with the Istituto Italiano della Saldatura (IIS) and Cranfield University.

The research follows a progressive methodological development combining CFD, signal analysis and data-driven methods. Initially, diagnostic techniques are developed for an industrial radial compressor using time-resolved CFD simulations. In this context, aerodynamic instabilities, such as surge, are considered from a maintenance viewpoint as fault mechanisms that can cause significant performance deterioration. Machine learning models are developed to reconstruct internal flow behaviour from limited pressure measurements and to classify compressor states approaching instability. Expanding on these results, a second approach utilises approximate entropy (ApEn) as a metric to identify the monitoring locations most sensitive to surge onset and provides a quantitative basis for sensor placement optimisation.

The expertise developed in this controlled numerical environment is subsequently transferred to wind turbines, a complementary class of turbomachinery for which experimental validation was possible. Digital Twin (DT) architectures are developed for a small-scale horizontal-axis wind turbine, integrating CFD-derived performance models with operational measurements. These DTs enable the detection of aerodynamic performance deviations, including yaw misalignment caused by sensor faults and blade surface degradation, while providing maintenance-oriented feedback to support operational decision-making.

Overall, the thesis demonstrates how CFD-informed diagnostic methodologies can evolve into DT frameworks capable of supporting condition monitoring and maintenance strategies for turbomachinery operating in complex industrial environments.

Publications

This thesis is based on the following peer-reviewed publications, a revised preprint manuscript, and a manuscript accepted for publication.

- L. Carrattieri, C. Cravero, D. Marsano, E. Valenti, V. Shishtla, C. Halbe. “The development of machine learning models for radial compressor monitoring with instability detection.“ *(2025) Journal of Turbomachinery, 147(5).*

- L. Carrattieri, C. Cravero, D. Marsano, E. Valenti. “The use of Approximate Entropy analysis for flow pattern identification in radial compressors to detect instable operating conditions.“ *(2024) Journal of Physics: Conference Series, 2893(1).*

- L. Carrattieri, C. Cravero, M. Lanza, M. Mortello, S. Tedeschi. “Enhancing yaw control resilience in wind turbines with CFD-informed digital twins.“ *(2025) Wind Energy Science Journal - Preprint.*

- L. Carrattieri, C. Cravero, M. Lanza, M. Mortello, S. Tedeschi. “Digital twin-based monitoring of wind turbine blade surface roughness using CFD-informed machine learning.“ *(2026) ASME Journal - In press.*

Copyright and permissions

Table of contents

1. Generalities on condition monitoring and diagnostics of turbomachines

1.1 Purpose and scope of the chapter

This chapter establishes the theoretical and methodological background for this thesis, delineating the methodologies, tools and challenges relevant to condition monitoring (CM) and diagnostics for turbomachinery. The objective is to establish a structured framework within which the established monitoring methodologies presented in the thesis can be understood, compared and critically evaluated.

The chapter proceeds by delineating the historical evolution of maintenance strategies for turbomachines, from corrective and preventive maintenance to condition-based and predictive solutions, while providing an overview of the principal families of diagnostic techniques. Classical monitoring methods, model-based and analytical diagnostics, signal processing and statistical approaches, together with machine learning (ML) and Digital Twin (DT) techniques, are discussed with emphasis on their respective assumptions, strengths and limitations.

By defining the scope and application of these methodologies, this chapter describes the conceptual boundaries of the thesis, concluding by emphasising the need for hybrid, physics-informed diagnostic paradigms. This foundation supports the methodological choices developed in the subsequent chapters.

1.2 Context and motivation for turbomachinery condition monitoring

Turbomachines, including compressors, turbines and pumps, are fundamental components for energy conversion and industrial processes. Their reliable and efficient operation directly affects plant safety and productivity. Unplanned shutdowns or undetected performance degradation can cause severe economic losses, safety hazards and greater impact to the environment [1–3]. As a result, the principal objective of turbomachine management becomes the continuous monitoring of their performance.

In the last few decades, the operational environment has changed. The request for maximum efficiency to satisfy global sustainability goals has increasingly aligned itself with the need to maximise asset reliability. Even minimal degradation of aerothermal performance because of faults such as fouling, erosion, misalignment and blade wear has become penal in energy production and emission considerations [4,5]. In addition, the rising age of industrially used machines has highlighted the need to maximise equipment's life without relying only on periodic overhauls.

The aforementioned challenges have progressively driven the development of structured maintenance philosophies. Among them, CM and predictive maintenance (PdM) represent a step forward from corrective and schedule-based maintenance philosophies toward data-driven decision-making. This paradigm aims at three primary industrial objectives: reliability (preventing unexpected failure), efficiency (preserving near-design performance) and sustainability (reducing waste and emissions) [6,7]. Turbomachinery poses challenges for CM due to high rotational speeds, complex flow patterns, severe mechanical and thermal loading. The variety of degradation mechanisms demands a multidisciplinary diagnostic framework combining mechanical, fluid dynamic and data-analytic perspectives [6,8] In today's industry, this will be accomplished through sensor systems and real-time computing aligned to define intelligent maintenance actions.

1.3 Historical evolution of maintenance and monitoring strategies

As introduced in section 1.2, over the years there has been a shift in the maintenance philosophy of the industry's equipment along with the rising levels of machine complexity, the widespread usage of sensors and advancements in data processing. The various maintenance strategies can be classed as follows: corrective maintenance (or run-to-failure maintenance), preventive maintenance and predictive maintenance [9]. A visual representation of how the different maintenance strategies work is provided in Figure 1.1.

In the run-to-failure approach, maintenance occurs only when a part of the asset has failed. This strategy is simple to manage but tends to result in high repair costs, unplanned downtime and often collateral damage to neighbouring subsystems.

The approach of preventive maintenance constituted the first shift from the purely corrective approach. Maintenance of the equipment takes place at predetermined regular time intervals (for example, x hours of usage of the equipment and/or y number of cycles). However, this approach implicitly assumes that degradation follows predictable, uniform trajectories, which may not reflect reality in complex turbomachinery.

These constraints have promoted the introduction of PdM (predictive maintenance), which takes a step ahead by leveraging real-time monitoring, data analytics and prognostics (such as estimating the Remaining Useful Life, RUL) to forecast failures and schedule maintenance just in time [10]. In the preventive approach, the maintenance interval is solely dependent on time/cycles, in contrast, in the case of PdM, the information derived from CM (condition monitoring) and fault detection/prognostics can identify when maintenance needs to be carried out at the best possible time.

In particular, CM refers to the systematic measurement, processing and interpretation of data that reflect the actual health state of a machine. It is not a maintenance strategy itself; rather, it supplies the diagnostic information required for both condition-based and predictive maintenance.

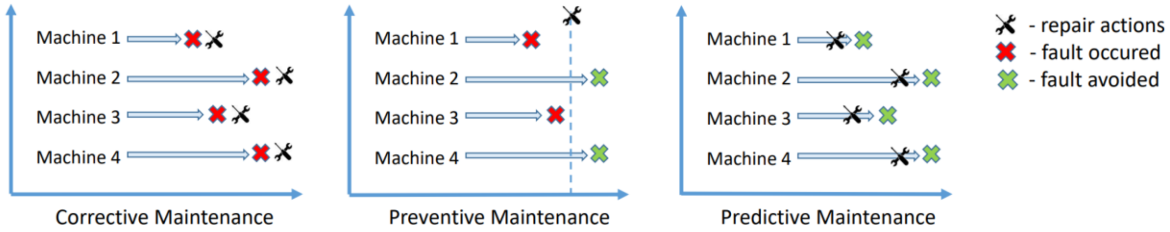

Figure 1.1. Different maintenance strategies with repair actions before and after potential failure (adapted from [9]).

A recent study emphasises this evolution, indicating that corrective and preventive strategies carry high-cost burdens and unplanned downtime, whereas PdM is framed as a key enabler of maximised asset availability and reliability under the Industry 4.0 paradigm [11].

In the domain of turbomachinery, the drivers for this shift are particularly compelling; the costs and risks associated with unplanned failures, such as rotor damage in a wind-turbine generator or blade loss in compressors, have increased substantially. Early detection therefore becomes highly valuable.

Degradations of machine performance (as might be caused through blade erosion, fouling, or bearing wear) can occur in a gradual fashion before being noticed until efficiency has already been substantially reduced. Preventive maintenance will not necessarily provide the best possible performance when compared to the measurement of the condition itself. Moreover, sustainability, regulatory demands and lifecycle-cost considerations compelled operators to achieve "more with less", meaning fewer downtime hours, less waste and lower emissions.

PdM supports these goals by reducing unnecessary interventions and avoiding catastrophic failures, with the associated waste and reparative burden. In addition to that, technological enablers have matured; with low-cost sensors, IoT connectivity, edge/cloud computing, real-time data analytics now making prognostics increasingly viable [12]. The benefits include improved reliability (fewer unexpected failures), improved operational efficiency (machine remains nearer to optimal performance for longer) and reduced environmental impact (less waste, fewer unnecessary interventions). For specific turbomachinery applications, these benefits are substantial and turn into an extended Mean Time Between Failures (MTBF), higher availability (reduced downtime), preserved performance (maintaining aerodynamic/hydraulic), early detection of degradation phenomena and thereby lifecycle extension of the equipment.

1.4 Classical monitoring techniques

In the past, the CM of turbomachinery was done by combining direct observation of their mechanical condition and various indicators related to their aerodynamic/thermodynamic performance. Even prior to the advent of machine learning or Digital Twin solutions in the field, the industrial process was already very sensor-heavy but simply less complex.

This section will highlight the classical monitoring methods from the areas of:

- vibration monitoring and spectral/order analysis,
- performance map-based health monitoring,
- pressure and temperature trending for surge and stall detection,
- lubrication (oil) analysis,
- and acoustic / acoustic-emission monitoring.

For each technique, typical implementations, advantages and limitations are highlighted.

1.4.1 Vibration-based monitoring

Vibration condition monitoring or VCM stands at the core of turbomachine CM. Modern plants employ accelerometer networks along with eddy-current proximity sensors positioned on bearings, casings and rotors to detect the vibrations in the turbomachine. The approach is well described in general reviews on vibration-based CM of rotating machinery and is standard in API 670-compliant systems [13–15]. Absolute casing accelerometers capture overall machine response, while relative shaft probes measure rotor motion in the bearing clearance.

Signals are typically sampled in the kHz range to capture synchronous frequencies (one or two times the shaft speed), blade passing frequencies, gear mesh frequencies and their sidebands. Industrial case studies for gas turbines and large compressors routinely use this setup for continuous protection and diagnostics [16]. Figure 1.2 presents an example of externally/internally mounted probes to measure the shaft vibrations, characterised by an unconstrained view of the shaft.

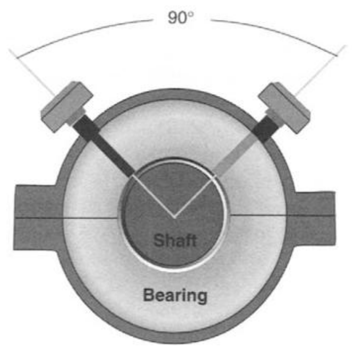

Figure 1.2. Example of externally/internally mounted Eddy probes (adapted from [17]).

The traditional analysis is based on the Fast Fourier Transform (FFT), used to obtain frequency spectra from vibrations in the time domain. Peaks at multiples of the shaft rotational frequency (orders) are associated with specific fault mechanisms. FFT theory is provided in Sect. 2.5.2 of Ch. 2. Industrial literature and textbooks (e.g. Scheffer and

Girdhar's *Practical Machinery Vibration Analysis and Predictive Maintenance* [17]) show how comparing current spectra with baseline signatures enables early detection of developing faults in gas turbines, compressors and pumps.

Among the advantages of this type of analysis there are a direct sensitivity to mechanical faults in rotating elements, large historical databases with well matured interpretation rules and easily automated alarm settings on band-limited spectral peaks. However, this type of monitoring suffers from a strong dependence on operating conditions (load, speed and process changes) which directly affect spectra. In addition, complex machines (multi-rotor gas turbines, multi-stage compressors) often produce overlapping spectral content that requires a well-established experience to perform an analysis. Finally, early-stage faults can be masked by process vibrations or poor sensor mounting.

Considering the aforementioned limitations of the FFT analysis in variable speed machines that involve many starts, stops and varying loads, the order tracking method is more preferable [18]. Simple FFT at fixed sampling frequency smears spectral peaks because the rotational frequency drifts over time. Consequently, order-tracking analysis has been used to correct simple FFT by resampling the vibration signal in the angular domain using a tachometer reference, so frequencies are expressed as multiples shaft speed rather than absolute Hz [19].

In turbomachinery, order tracking is used to follow the growth of specific orders during run-up/coast-down of gas turbines and turbogenerators, to separate shaft-related orders from blade-passing or gear-mesh components and to construct order maps (order vs. speed vs. amplitude) to identify resonance crossings and critical speeds.

This method is thus robust when applied to speed transients, essential for start-up/shutdown diagnostics and enhances detectability of speed-related faults, such as unbalance, misalignment, cracks. Despite that it requires reliable tachometer signals and more complex processing pipelines, together with a higher data volume and computational cost than simple steady-state FFT.

1.4.2 Performance map-based condition monitoring

Many original equipment manufacturers (OEMs) of gas turbines and industrial compressors use performance map-based monitoring to track gradual degradation phenomena, such as fouling, erosion and leakage. The idea is to compare the measured performance against the baseline component maps or thermodynamic models and infer health indicators [20,21].

In gas turbine engines, component map-based models are typically available for each unit (compressor, turbine and combustor) via non-dimensional maps of corrected mass flow, pressure ratio and isentropic efficiency. These maps are scaled to match the specific engine and operating conditions [6].

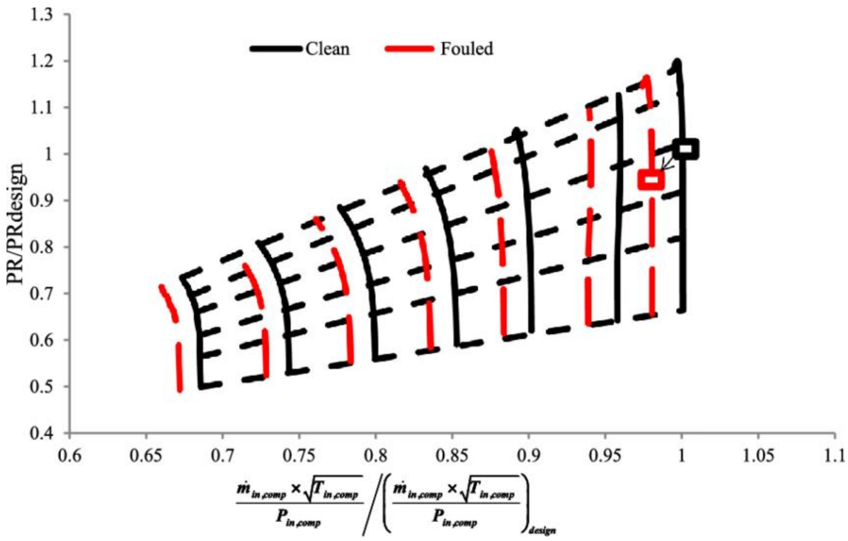

Figure 1.3. Compressor map under clean and fouled conditions (adapted from [22]).

CM of the engines proceeds by acquiring steady-state operating data (pressures, temperatures, shaft speeds, fuel flow), correcting them to reference conditions (inlet temperature/pressure, humidity), running a performance model to compute the "clean" expected outputs and finally comparing the measured vs. predicted performance to estimate degradation in compressor efficiency, turbine efficiency or flow capacity. Similarly, for industrial compressors and pumps, performance maps are used to verify that the measured head and flow remain within expected bands; persistent deviations might be symptoms of fouling, internal recirculation or mechanical damage [23,24]. Figure 1.3 presents an example of a compressor map under clean and fouled states, illustrating the principle of map-based CM.

Among the advantages for this type of monitoring approaches, there is a global picture of the thermodynamic health of the machine, not just local vibration or pressure anomalies. In addition, they are well suited for detecting slow, progressive degradation phenomena, necessitating relatively low additional instrumentation if main sensors are already installed.

Even so, there is the main requirement for high-quality models and accurate ambient corrections; in addition, degradation attribution is model-dependent so different faults can produce similar performance changes. Finally, these models typically operate at slower time scales (hours to days) rather than a real-time detection.

1.4.3 Pressure and temperature trending for surge and stall detection

Compressors and certain types of pumps involve the measurement of pressure and temperature to determine incipient surge in the process. These measurements are used by anti-surge control systems and typically include inlet and discharge static pressures, differential pressures across elements, upstream and downstream gas temperature and sometimes shaft power or driver torque [25–27].

For industrial centrifugal compressors, the operating point is continuously plotted on a pressure ratio vs mass flow map where the surge line is defined experimentally or analytically. Anti-surge controllers enforce a minimum distance to the surge line by opening recycle or blow-off valves [25]. Typical industrial implementations (e.g. in process plants or gas pipeline compressors) monitor: the distance to the surge line as a control variable, fast changes in discharge pressure or mass flow rate as indicators of surge cycles.

Map-based monitoring can be complemented by temperature and pressure trending; because surge or deep stall events usually cause rapid, cyclic recirculations and separations in the flow path, resulting in spikes and oscillations in pressure and discharge temperatures, together with changes in the driver load or current [28,29]. Practical thresholds are defined on the rate of change of discharge pressure, the rate of change of mass flow or differential pressure across the compressor, the rise in discharge temperatures and sometimes the rate of change in driver power [30,31].

These monitoring methods leverage sensors that are already present on the machine (pressures, temperatures, flow rates). Nevertheless, they are less sensitive to early precursors of instability and often activate when instability has already become significant, thus showing a strong dependence on accurate sensor calibration and process conditions. This makes it difficult to distinguish the root causes of instability (aerodynamic vs. instrumentation fault) without additional context.

1.4.4 Lubrication (oil) analysis

Oil analysis is another classic field in turbomachine CM. This particularly applies to the large gas turbines, steam turbines and industrial compressors. This analysis relies on measurements of quantities like oil viscosity and its temperature dependence, Total Acid Number (TAN), Total Base Number (TBN), water content, oxidation and contamination indicators, dissolved metals and wear particles. It is performed via spectrometric analysis, particle size distribution and morphology via ferrography [32].

Figure 1.4 illustrates a schematic example of a steam turbine oil system and corresponding sampling points [33]. This monitoring method shows a high sensitivity to early-stage wear and contamination (filter or seal failures, water ingress), providing direct information about lubrication health, essential for safe operation; however, the limitations of this analysis are rooted in a periodic oil sampling (weekly/monthly), which prevents this technique to be strictly real-time.

Moreover, online oil sensors exist but are less standardised. Ultimately, oil interpretation requires knowledge of oil formulation, additive packages and machine metallurgy, which makes this method less suited to detect fast-developing faults, whereas vibration or pressure signals analysis can be more immediate [34,35]. In practice, oil analysis is used as a complementary method, where vibration and process measurements provide rapid alarms and oil trends support longer-term maintenance planning and root-cause analysis.

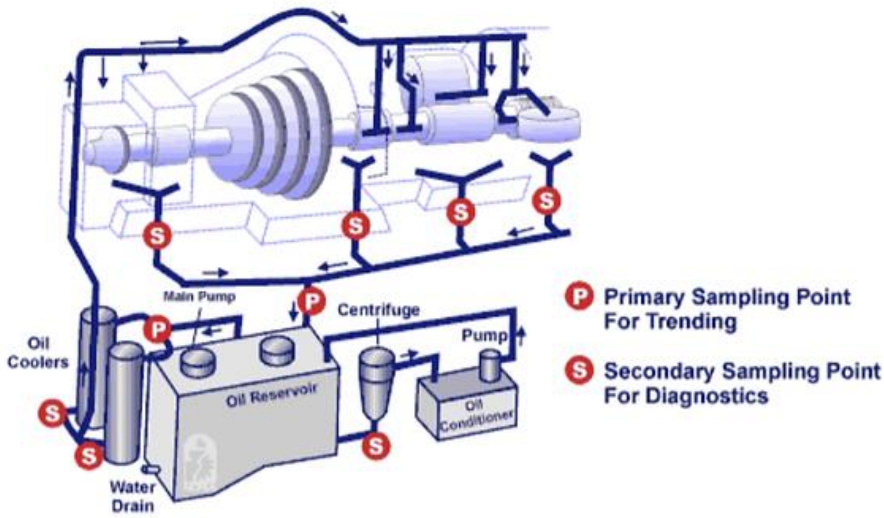

Figure 1.4. Example of a steam turbine oil system and corresponding sampling points (adapted from [33]).

1.4.5 Acoustic and acoustic-emission monitoring

In addition to conventional vibration accelerometers, some installations use microphones or acoustic emission (AE) sensors for detecting high-frequency phenomena like the progression of cracks, rubbing contacts, gas leaks, or combustion processes. Experimental gas turbine research, utilizing casing-mounted AE sensors, showed that they can differentiate various operating regimes and indicate unusual behaviour in gas turbine operations in combination with thermodynamic and vibration data [36].

This approach to CM is also marked by a sensitivity to very high frequency oscillations, ideal for the detection of faults prior to their observation in the classical range of vibrations, along with the use of non-contact sensors such as microphones.

However, the analysis procedure is also highly vulnerable to background noise levels compared to other methods. For industrial gas turbines, simpler methods of acoustic monitoring (such as sound pressure level measurement) are sometimes integrated into combustion tuning and flame instability detection [37,38], but vibration and pressure measurements remain the primary CM approaches.

1.5 Model-based and analytical diagnostics

Despite their widespread use, classical monitoring methods operate primarily on observed symptoms rather than on physical reasoning. Model-based diagnostics, discussed in the following section, address this gap by embedding system physics directly into the fault detection logic and forming the theoretical core of modern CM in turbomachines. Such techniques are different from data-driven methods because they employ explicit models to describe the expected system behaviour.

Faults are detected by evaluating the inconsistency, or residuals, between measured quantities and model predictions. The fundamental principle can be expressed as:

$$r(t) = y_m(t) - y_{model}(t) \tag{1.1}$$

where $r(t)$ is the residual at time t, $y_m(t)$ is the measured signal (e.g., pressure, temperature, speed) and $y_{model}(t)$ is the output predicted by the model under nominal conditions. Under healthy operation, residuals remain within a noise-dependent tolerance band, whereas a persistent deviation indicates a possible fault or degradation process.

This approach is also commonly known by the term analytical redundancy. Analytical redundancy is different from hardware redundancy because in hardware redundancy sensors or measurement instruments are duplicated. On the other hand, analytical redundancy relies on the relations between the system variables to reconstruct the expected values making the detection of faults in the system feasible from just one channel [39,40].

Finally, model-based diagnostics differs fundamentally from performance map-based monitoring introduced in Sect. 1.4.2, since they rely on static empirical baselines derived from OEM data rather than dynamic physics-based equations and state estimation to compute residuals.

1.5.1 Analytical redundancy and residual generation

In turbomachinery, analytical redundancy is typically constructed from thermodynamic and aerodynamic balance equations (mass, momentum and energy conservation) combined with component performance maps. For instance, in a gas turbine steady-state model, the model might calculate the pressure and temperature at the gas turbine exit given the gas inlet pressure, shaft speed and fuel flow rate [41]. Any discrepancy in the true gas turbine's exit temperature from the model's results shows a model residual. Such residuals may be developed within various formulations:

- Parity-space methods, which project measured signals onto subspaces defined by model equations to isolate inconsistencies.
- Observer-based approaches, where state estimators (e.g., Luenberger or Kalman observers) reconstruct internal states and compare them to measurements.
- Parameter estimation methods, which track changes in model parameters (e.g., compressor efficiency, flow capacity) as indicators of degradation [42].

For example, in a single-shaft gas turbine, the predicted compressor outlet temperature $T_{3,model}$ can be expressed as:

$$T_{3,model} = T_2\left(1 + \frac{1}{\eta_c}\left(\pi_c^{\frac{\gamma-1}{\gamma}} - 1\right)\right) \tag{1.2}$$

where T_2 is the inlet temperature, π_c is the compressor pressure ratio, η_c is the isentropic efficiency and γ is the specific heat ratio. A deviation beyond a calibrated threshold, between the measured and modelled outlet temperature $\Delta T_3 = T_{3,meas} - T_{3,model}$ suggests either a measurement bias or actual efficiency loss due to faults.

Analytical redundancy has been successfully used for identifying sensor drifts, air-path leaks and thermal losses in industrial gas turbines [43]. Its main advantages are interpretability and the ability to separate physical mechanisms from noise or unrelated process variations. However, it requires a validated physical model and accurate boundary conditions, with limitations that can become critical in complex multi-shaft or variable-geometry machines.

1.5.2 Kalman filtering and state estimation

For dynamic monitoring, analytical redundancy is extended through state observers such as the Kalman filter, which combines a process model with noisy measurements to produce optimal state estimates in the least-squares sense. In a linearized continuous-time form, the state-space equations can be written as:

$$\dot{x}(t) = Ax(t) + Bu(t) + w(t) \tag{1.3}$$

$$y(t) = Cx(t) + v(t) \tag{1.4}$$

where $x(t)$ represents the system state (pressures, temperatures, shaft speeds) at time t, $u(t)$ the inputs (fuel flow, inlet conditions), $y(t)$ the measured outputs, w, v process and measurement noise, respectively, with A, B, C representing the system, input, and observation matrices, respectively. The Kalman gain (K) optimally combines the model prediction with the measurement information, minimizing the expected estimation error covariance.

Kalman filters in turbomachines are especially suited for the detection of low rates of degradation (fouling/erosion) or faults like biased sensors. Unobserved parameters related to health measurements like the efficiency of the compressor, the flow rate of the turbine and the pressure ratio scale factors could also be monitored [44–46].

Advantages include robustness to measurement noise, adaptability to changing operating conditions when embedded in adaptive or extended formulations and real-time application. However, filter divergence may occur if the model or noise statistics are poorly tuned.

1.5.3 Parameter estimation and thermodynamic models

Thermodynamic models are at the base of most analytical redundancy systems. They describe steady or transient relationships between key quantities (e.g., mass flow, pressure ratio, temperature, efficiency) and can be formulated at various levels of detail.

Zero-dimensional (0D) and one-dimensional (1D) gas-path models provide a good balance between model fidelity and computational cost required for online applications. A 0D model treats each component as a lumped element through empirical maps [47]; a 1D model includes distributed parameters along flow paths, thus enabling the prediction of spatial gradients. Figure 1.5, illustrates an example of a 0D performance model of the NASA E3 High Pressure Turbine [47]. When assessing the health condition of the machine, measured quantities are compared against the model outputs to derive multiplicative health parameters θ (e.g., efficiency, flow capacity, pressure loss). For instance, a steady-state gas turbine model can be expressed as:

$$y_{meas} = f(u, \theta) + \varepsilon \tag{1.5}$$

where $f(\cdot)$ is the thermodynamic model, u the control inputs, y_{meas} the measured outputs and ε the noise. Estimation of health parameters corresponds to the inverse problem of minimizing residuals through least squares or recursive procedures [48,49].

These low-order models are sufficient in terms of speed for real-time implementation in CM. However, their drawback lies in their dependence on calibration. For complex machines, hybrid methods have been developed through the combination of 1D gas path analysis along with state estimation techniques, like Kalman filters.

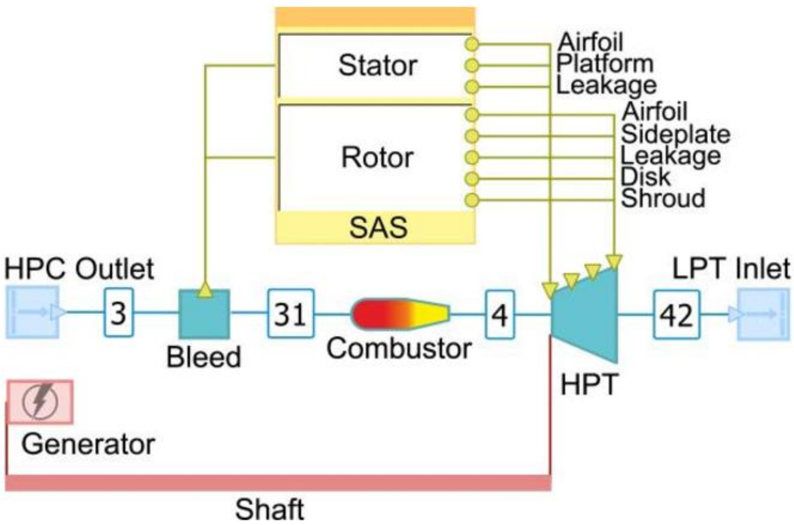

Figure 1.5. 0D performance model of the NASA E3 High Pressure Turbine (adapted from [47]).

Before the advent of machine learning and data-driven methods for diagnostics, analytical models were generally refined with the aid of CFD. High-fidelity CFD simulations provided insight into flow separation, leakage, or cooling effectiveness. The results could then be used for the generation of reduced-order or surrogate models for faster diagnostic use. Surrogate models such as polynomial regressions, Kriging surfaces and response surface methodologies, were used for this scope [50,51]. They enabled rapid estimation of changes in performance by monitoring parametric variations without the need to perform direct computationally expensive simulations. This approach bridged the gap between detailed physical understanding and real-time monitoring needs.

Compared with purely empirical or data-driven approaches, model-based diagnostics provide explicit physical rationale and interpretability. They prove advantageous when fault data are scarce or when new machine configurations lack historical datasets. On the other hand, thermodynamic models have the drawbacks of demanding accurate modelling, high-quality measurements and continuous calibration.

1.6 Signal processing and statistical approaches

The advent of signal processing and statistical analysis in turbomachine CM represented a paradigm shift from traditional parameter trending towards the current data-rich diagnostic approach. Although the traditional CM models mostly used steady-state parameters like pressure ratio, temperature-rise ratio, or efficiency values, engineers eventually realized that faults and flow instabilities usually occur in the form of dynamic signatures within the high-frequency data. Such dynamic signatures are not perceptible through averaged values but rather through modulations, wide-band noise, or occasional oscillations in the signals related to vibrations, pressures and flow rates. Signal processing was therefore introduced as a means of extracting informative features from raw measurements, transforming complex time histories into quantities that reveal the machine's evolving health state.

Early work in this area drew from the vibration analysis techniques for turbomachinery, where spectral decomposition provided a natural language for interpreting periodic mechanical and aerodynamic phenomena. Randall's foundational text [52], describes how Fourier-based spectral indicators became standard tools for diagnosing imbalance, misalignment, or blade-passing irregularities.

1.6.1 Evolution from spectral to time-frequency analysis

As presented in Sect. 1.4.1, the Fourier transform first enabled the identification of characteristic frequency components related to compressor surge, rotating stall and blade vibration. However, the assumption of stationarity (signals whose frequency content does not change over time) proved limiting. Phenomena like surge and stall are inherently transient, featuring bursts of oscillation that grow, decay, or shift in frequency as flow separation evolves.

To overcome this limitation, time-frequency techniques including the short-time Fourier transform (STFT) and continuous wavelet transform (CWT) were later introduced. These approaches are characterised by preserving temporal localization, thus allowing engineers to track how dominant frequencies evolve in time during the onset of instabilities. Silvestri et al. [53] for instance, used the joint wavelet and cepstral analysis of pressure signals from a turbocharger compressor to identify the region where incipient surge can manifest through an increase in energy along certain ridges in the time-frequency domain, prior to the manifestation of the macroscopic oscillations. Such cyclostationary analyses of pressure and vibration signals also appeared in subsequent works [54].

This shift represented a conceptual evolution: rather than viewing sensor outputs as static quantities, they became dynamic carriers of information. Monitoring was no longer the measurement of constant values but the observation of varying phenomena.

1.6.2 Statistical and entropy-based indicators

As the quantity of sensor data grew, the next step became quantifying signal irregularity, meaning how ordered or chaotic a time-series appeared. Classical variance and higher-order statistics could describe amplitude dispersion, but not the degree of unpredictability or pattern complexity. This led to the introduction of information-theoretic metrics, such as entropy-based measures. Here, entropy refers to information-theoretic entropy derived from a signal, not thermodynamic entropy, and quantifies the irregularity and unpredictability of the time series. In turbomachinery, it provided a new dimension of interpretation: a rise in entropy typically corresponded to a more disordered flow or intermittent oscillations, conditions preceding mechanical or aerodynamic instability.

In 2015, Liu and Wu [55] demonstrated the use of wavelet spectrum entropy to monitor non-stationary vibrations in scroll compressors, showing that entropy increases when flow transitions toward unsteady operation. Similarly, in the same year Zhang et al. [56] applied wavelet Rényi entropy to turbine vibration data and related changes in signal complexity to safety margins under fluctuating load conditions.

Another important metric is the approximate entropy (ApEn), formally introduced by Pincus in 1991 [57] and later popularized for machinery diagnostics by Yan and Gao (2007) [58], which provided an efficient way to evaluate the degree of predictability within a time series. High ApEn values are a symptom of low regularity, suggesting that the system is entering a dynamically unstable state. More details about ApEn theory are provided in Sect. 2.5.3 of Ch. 2.

1.6.3 Purpose and conceptual impact

The adoption of these statistical indicators was not motivated by mathematical novelty alone but by engineering necessity. Traditional monitoring methods offered limited visibility into the early stages of degradation or surge inception in compressors. Signal processing, by contrast, enabled:

- Feature extraction, isolating patterns or modulations hidden within broadband noise.
- Pattern recognition, describing evolving conditions through metrics (frequency shifts, energy concentration, entropy trends) that can be trended over time.
- Sensitivity to precursors, detecting instability or wear before amplitude-based alarms trigger.

Through these developments, engineers moved from descriptive monitoring, reporting whether a parameter exceeded its limit, to diagnostic monitoring, understanding why the signal changed. The turbomachine was no longer treated as a black box but as a dynamic system whose statistical signatures reflected its physical state.

1.6.4 Foundations for future data-driven diagnostics

By the end of the 2010s, the use of wavelet analysis, cyclostationary statistics and entropy-based metrics had created a rich set of monitoring frameworks capable of describing complex flows and vibration phenomena. This body of work laid the conceptual groundwork for modern machine-learning diagnostics, where the extracted indicators became model inputs for classification and prognostic algorithms.

However, in the pre-machine learning period, the emphasis was still interpretative. Every single feature (such as the energy in the frequency spectrum, the entropy level, the amplitude modulation rate) was directly physically meaningful. This made the initial models of signal processing fundamentally different from the ones developed during the data-science period.

1.7 Transition from rule-based and statistical diagnostics to machine-learning approaches

The evolution of turbomachinery diagnostics has been driven by a growing complexity in machine operation and the need to extract meaningful insights from large quantities of sensor data. As presented in the previous sections, historically CM relied on rule-based systems and statistical indicators derived from meaningful quantities like vibration, pressure and mass flow rate. These methods operated on the principle of fixed thresholds or limit rules, established empirically from baseline tests or accumulated operational experience. Given features, like root mean square values of vibrational signals, discharge pressures, or wavelet energy, were continuously compared to predefined reference bands. If the measured value exceeded the band, an alarm or protective action was triggered.

This rule-based logic offered simplicity and transparency. It formed the basis of early industrial monitoring systems, such as those following the API 670 standard [13], where amplitude thresholds, rates of change, or spectral peaks were directly linked to alarm levels. However, such systems lacked adaptability. They could not automatically account for varying ambient or load conditions, nor distinguish between benign transients and true degradation. The outcome resulted in a high rate of false alarms or missed detections, especially under complex operating regimes involving part-loads, transient startup sequences, or multi-stage coupling effects.

Before the adoption of modern ML techniques, a significant intermediate step was defined by methods like the multivariate statistical process control (MSPC) [59]. These methods extended the concept of analytical residuals into a purely data space by modelling the normal operating behaviour through historical datasets rather than physical equations. Instead of relying on fixed limits, engineers began defining alarm conditions based on deviations from historical averages or probabilistic models of normal behaviour. For example, multivariate statistical process control methods, such as the Hotelling's T^2 statistics or the principal component analysis (PCA), were applied to detect correlated shifts across multiple sensors [60–62].

These approaches provided a more flexible and noise-tolerant framework, effectively extending the concept of analytical residuals into a purely data space. Nevertheless, their capability remained limited by linear assumptions and the need for extensive calibration.

1.7.1 Emergence of machine learning in diagnostics

The next paradigm shift was the advent of ML, first as an extension of statistical analysis methods, followed by it being recognized as a paradigm. ML involved the ability to automatically discover the boundaries or relationships from data. This was very applicable in turbomachines since the operating ranges involved large numbers of parameters. Additionally, the failure signatures tended to overlap [63,64].

Supervised learning models, such as support vector machines (SVMs) and Random Forests (RFs), began to replace traditional rule-based classifiers. Trained on labelled datasets representing healthy and faulty states, these algorithms could infer nonlinear boundaries between different operating conditions. For example, Sodemann et al. [65] applied asymmetric SVMs to map surge boundaries in centrifugal air compressors, reducing false positives compared with static surge maps. Similarly, ensemble methods like RFs were used to classify CM data for fault detection, leveraging their inherent robustness to noisy inputs [66]. Unsupervised methods such as clustering and autoencoders further expanded diagnostic capabilities by identifying unseen or emerging fault modes. Instead of requiring training on pre-labelled data, these models defined a dynamic baseline for the normal machine operation and flagged deviations as potential anomalies. This was particularly relevant for gas turbines and process compressors, where complete fault datasets are rarely available. Yusoff et al. [67] employed a hybrid k-means-Gaussian Mixture Model framework to track slow degradation trends in compressor efficiency, automatically redefining the boundary between normal and incipient fault conditions.

Such adaptability reduces the need for manual recalibration after maintenance or seasonal environmental changes, improving monitoring reliability in long-term operation. Further details about ML theory and algorithms for CM are provided in Ch. 2.

1.7.2 Motivations and benefits of ML integration

The primary motivation for introducing ML in turbomachinery monitoring lies in its ability to handle complexity, adaptivity and nonlinear interactions among numerous physical variables; whereas rule-based systems operate on explicit conditions (e.g., if x > threshold, then alarm). ML models infer relationships implicitly, by identifying subtle correlations across a large number of process variables and sensor channels. This capability enables early fault recognition even when no single parameter exceeds its nominal limit, which is an essential advantage for multistage machines like compressors and turbines, where small local instabilities may propagate before being reflected in global measurements.

Another reason for the adoption of ML is the ability to scale and generalise. Data-driven models that are developed through ML are able to process multi-sensor data simultaneously for various operating conditions. Among them, Artificial Neural Networks (ANN), excel in approximating nonlinear mappings between input features and degradation states, capturing dependencies that analytical or low-order thermodynamic models would often linearize or neglect. This property makes ML particularly suitable for representing the soft nonlinearities inherent in aerothermal systems, e.g., tip-leakage dynamics, diffuser separations and thermal expansion effects.

Of equal importance is the adaptive learning process performed by ML models. Through the update of model weights or cluster boundaries, the ability to track the dynamics of system component health is achieved, beyond the capabilities of static thresholds methods.

However, the engineering community has also acknowledged the relevance of the concept of hybridization in ML being used in collaboration with physics models. For example, CFD analyses produce physically valid data that help in the learning process, thus enabling ML algorithms to make forecasts that are physically meaningful. In this context, ML can also serve as a surrogate or corrective layer, accelerating CFD predictions, interpolating sparse sensor data, or learning residual errors between simulation and experiment. Hammond et al. [68] and Cruz et al. [69] demonstrated the implementation of ANN or Gaussian process surrogates in CFD solvers for improving the analysis forecasts by maintaining the physically consistent behavior of the system. This hybrid CFD + ML philosophy leads to diagnostic systems that are both data-rich and physically constrained, bridging the traditional divide between numerical simulation and empirical monitoring.

Another key motivation is data fusion, meaning the capacity of ML frameworks to integrate heterogeneous information sources. Current CM for turbomachines goes beyond purely mechanical parameters because it also considers control system data, operational input data and maintenance history. ML models such as ensemble regression models and graph neural networks can combine different sources of input into one condition index, yielding more holistic diagnostics. Recent studies have shown that unsupervised knowledge-based hybrid models can quantify health degradation with high reliability while maintaining interpretability [70].

Recently, the explainability and transparency of newer ML implementations have begun to address the limitations about their "black-box" behaviour. Feature-importance measures, surrogate explainers and attention mechanisms now allow engineers to trace the model decisions back to physically meaningful indicators, such as pressure fluctuation intensity, temperature gradients, or modal energy content.

Considered together, these aspects highlight the importance of ML in turbomachine diagnostics. ML's ability to process different types of data, deal with non-stationarity in system behavior and retain physical interpretability, places it as the enabling intelligence that advanced conventional monitoring in turbomachinery.

The shift toward ML-based diagnostics naturally paved the way for DT technologies, which represent the next stage in turbomachinery monitoring. DTs integrate physics-based models, sensor data and potentially ML algorithms into a unified virtual representation of the machine, capable of continuous updating and predictive simulation. Hence, the evolution from fixed-threshold monitoring to adaptive, learning-based diagnostics signifies not merely a change in technique but a transformation in philosophy: from monitoring signals to understanding systems. In this trajectory, ML can serve as the enabling intelligence that connects traditional analytical reasoning with the predictive and integrative capacities of DTs. Further details are discussed in the next section.

1.8 Digital twins for condition monitoring

1.8.1 Digital Model, Digital Shadow and Digital Twin

The evolution process from simple digital rapresentations to bidirectionally interactive DTs has been formalized using the Digital Model, Digital Shadow and Digital Twin hierarchy. The hierarchy was first organized by Kritzinger et al. in 2018 [71] and it classifies digital technologies depending on the flow of data between the physical and virtual worlds.

- A Digital Model (DM) is the most elementary type, where the digital representation is static/semi-static and there is no automatic connection to the data. These include Computer-Aided Design (CAD) geometries, CFD models and finite-element models that don't depend on real-time operation.

- A Digital Shadow (DS) introduces one-way data flow from the physical system to the virtual one. Measurements update the digital representation, but the virtual object has no direct influence on the physical object. Most of the monitoring dashboards, CM tools and virtual sensors fall under this category.
- A Digital Twin (DT) is the highest-level representation, where there is an exchange of information in both directions. The digital counterpart receives live inputs and has the ability to send commands and/or optimizations in the opposite direction. This enables real-time closed-loop coordination. More details are provided in Sect. 1.8.2.

This conceptual hierarchy is illustrated in Figure 1.6, where the transition from model, to shadow, to twin introduces progressively stronger integration between sensing, simulation and control.

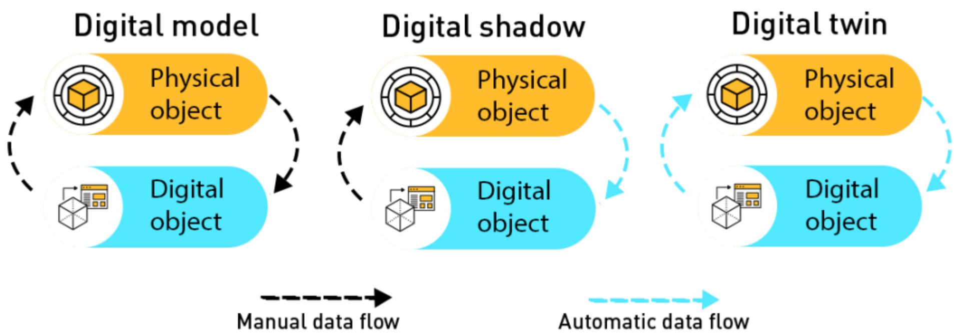

Figure 1.6. Digital Model vs. Digital Shadow vs. Digital Twin (adapted from [71]).

1.8.2 Concept and evolution of Digital Twins

The concept of DT was first conceptualized in the manufacturing and aerospace fields as a virtual representation of a physical asset that remains continuously synchronized with it through data exchange. The term was popularized by Grieves in 2014 [72], who envisioned DTs as the cornerstone of Product Lifecycle Management (PLM) and later refined by Tao et al. [73] as an “information mirroring model” capable of describing, predicting and optimizing physical systems in real time. Formally, a DT can be described as a dynamic mapping:

$$\hat{x}(t) = f\left(x_m(t), M_{\text{phys}}, M_{\text{data}}\right) \tag{1.6}$$

where $\hat{x}(t)$ represents the estimated state vector of the physical system at time t, f is the dynamic mapping function, $x_m(t)$ denotes sensor measurements, M_{phys} and M_{data} are the physics-based and data-driven model components, respectively. The synchronization between physical and virtual objects occurs through a bidirectional data flow, forming the basis of DT fidelity [74].

The development of DTs has been closely related to the rise of Industry 4.0, that integrates the Internet of Things (IoT), cloud computing and artificial intelligence into manufacturing and energy systems. Early implementations in the aerospace sector focused on component-level replicas for design optimization. As sensor networks and computational power grew with time, DTs evolved into operational tools capable of performing continuous health monitoring, anomaly detection and PdM. The modern DTs thus extend far beyond static modelling, they are a closed-loop digital ecosystem where sensing, simulation and learning coexist [75].

1.8.3 Digital twin architectures and frameworks

Most DT implementations present a three-layer architecture: (1) the *physical layer*, containing sensors and actuators; (2) the *virtual layer*, comprising computational models and simulators; and (3) the *data integration layer*, managing synchronization and analytics [76]. The fidelity of a DT depends on how well the loop between the layers is closed. That is how frequently data flows from the physical object to the virtual object and back to the control system.

Recent DT frameworks introduce distributed and multi-domain architectures to support real-time monitoring and predictive analytics. Abdullahi et al. [76] proposed a distributed DT framework within the Industrial Internet of Things (IIoT) using fog computing for wind turbines and demonstrated improved latency and data processing efficiency. Similarly, Liu and Shangguan [77] presented a multi-domain DT for marine steam turbines that integrated thermodynamic, mechanical and control subsystems within a Modelica-based architecture, achieving real-time symmetry between the digital and physical assets.

From a computational perspective, DTs can be classified into three categories [78]:

1. Physics-based twins, built from high-fidelity models (e.g., CFD, FEM) and capable of capturing system dynamics.
2. Data-driven twins, relying on machine learning and statistical inference models based on sensor data.
3. Hybrid twins, combining the above categories to balance accuracy and computational cost.

The hybrid approach presents a relevant potential for the turbomachinery sector, where physics-based flow models are commonly used and data-driven surrogates can learn residual dynamics. In 2024, Luan et al. [79] developed a surrogate-based DT for gas turbines using Markov projection approximation to track evolving system dynamics, achieving robustness against noise and real-time adaptability.

1.8.4 Digital Twin modelling and synchronisation

The essence of a DT lies in its ability to mirror the evolving state of a physical asset, not through a static replication, but through a continuous synchronisation between the physical and virtual object. This synchronisation process is what distinguishes a DT from traditional simulations or monitoring systems, because it enables right-time awareness and adaptive prediction.

In practice, DT development begins with the construction of a representative model that captures the key dynamics of the selected system. For turbomachinery, this model may combine thermodynamic performance equations, vibration dynamics or degradation trends, depending on the intended diagnostic scope. Modern approaches increasingly adopt hybrid modelling, due to the improvement in both predictive accuracy and interpretability, thus allowing the twin to remain robust even under varying operating or environmental conditions [79,80].

Once a model exists, the critical step is data synchronisation**,** that is, the continuous alignment between the physical object and its digital counterpart. This process relies on a seamless data pipeline that streams sensor data (e.g., pressures, temperatures, vibrations, control variables) to the virtual environment [76]. The frequency and reliability of this exchange determine the fidelity of the virtual twin, meaning its capacity to remain an accurate reflection of the physical object in both steady and transient regimes.

Beyond technical implementation, synchronisation also has a cognitive dimension. It enables the DT to act as a continuously learning entity, absorbing experience from operational data, updating its internal parameters and refining its understanding of the system's health state.

In summary, modelling and synchronisation transform the DT from a passive model into an active agent for system monitoring. The DT learns, adapts and responds in real time, forming the foundation for predictive diagnostics and autonomous maintenance.

1.8.5 Digital Twins for condition monitoring and maintenance

CM long dominated by empirical and model-based diagnostics, is being fundamentally reshaped by the DT paradigm. The condition of a component can be mathematically represented by a health state vector, $h(t)$, estimated as:

$$h(t) = g(x_m(t), \hat{x}(t), \theta) \qquad (1.7)$$

where $x_m(t)$ is the measured signal vector, $\hat{x}(t)$ is the DT-predicted state and θ represents degradation parameters or learned features. The residual $r(t) = x_m(t) - \hat{x}(t)$ thus forms the core of DT-based fault detection logic, deviations between measured and virtual outputs indicate abnormal operation. This conceptual formulation, common to most DT condition monitoring frameworks, has been adopted in several recent studies [80–82].

Beyond diagnostics, DTs can support PdM actions by integrating physics-informed degradation models with probabilistic forecasts. Abdullahi et al. [76] showed how distributed DTs, connected through IIoT architectures, can forecast failure progression and automatically trigger maintenance alerts. Mrzyk et al. [83] extended this concept to reusable IT architectures, reducing DT deployment time and improving scalability across different assets.

In modern implementations, the DT also integrates virtual sensing capabilities, commonly referred to as soft sensors, that infer quantities not directly measurable in real time. These models, typically built with algorithms like autoencoders, kernel regressors, or Bayesian networks, augment the physical sensor network by providing continuous virtual measurements and uncertainty bounds [84].

1.8.6 Digital Twins in turbomachinery applications

The use of DTs in turbomachinery has seen a quick development in the last decade, mostly because of improvements in sensor resolution, high-speed data acquisition and reduced-order modelling. A DT model of a gas turbine engine or compressor would essentially couple models related to the mechanics, controls and thermodynamics, and would be constantly updated through Supervisory Control and Data Acquisition (SCADA) and distributed sensors.

Early prototypes such as the AJAX DPS-180 DT developed by Prokhorenko et al. [85] replicated a reciprocating compressor to predict failure times and optimise service intervals based on IoT-enabled data collection. Similarly, the Stampede DT for process equipment, demonstrated by Rodriguez et al. [86], used dynamic simulation and clustering of process variables to estimate the RUL of gas compressors and coolers, detecting faults days before physical alarms were triggered. Luan et al. [87] showed that surrogate models in a DT of a gas turbine could track degradation modes such as compressor fouling or turbine erosion.

In the turbomachinery field, DTs have also evolved from offline performance models to real-time diagnostic systems capable of virtual sensing, anomaly detection and life prediction. Pezzini et al. [88] implemented a real-time DT environment for a micro gas turbine using distributed microservices, achieving < 1 % prediction error on unmeasurable parameters like the net power output. Likewise, Ma et al. [82] constructed a probabilistic performance DT for industrial gas turbines using uncertainty quantification and Wasserstein distance metrics to identify anomalies with an F1 score of 0.99. These results confirmed that DTs can perform model-based diagnostics under uncertainty, effectively becoming self-calibrating digital replicas of complex thermodynamic systems.

Recent hybrid solutions extended DT applications into fault prediction and life estimation. Zhu et al. [89] used a DT combining a generative adversarial network (AAKR-

ACGAN) to detect compressor blade faults up to one year before failure and demonstrating the potential of DT-based early warning. Similarly, Cruz-Manzo and Panov [90] embedded a real-time executable DT within a gas turbine controller to estimate compressor efficiency and RUL during fouling conditions.

The presented examples aim to reveal the multidimensional nature of DTs for turbomachinery: they act as virtual observers, predictive simulators and advisory systems, each role requiring different levels of fidelity and computational intensity.

1.8.7 Outlook and challenges for Digital Twins

Despite their rapid progress, DT in turbomachinery face several implementation challenges. One of the most persistent is the fidelity-latency trade-off. This is because achieving high diagnostic accuracy requires complex models and complex sensor networks, resulting in an increase in computational cost and limiting real-time responsiveness [91]. Therefore, current industrial applications increasingly rely on reduced-order models and data-fusion strategies, where simplified physical models and/or machine-learning surrogates replicate the dynamics of the system with lower latency. This represents the most practical route toward deployable DTs for CM and PdM, where streaming analytics and rapid decision-making are more critical than full physics resolution.

Another critical issue is uncertainty quantification (UQ). Because DTs rely on live sensor data, measurement noise, latency and model parameter drifts can affect the reliability of the DT’s output; so probabilistic DTs, incorporating Bayesian inference and ensemble learning, are emerging as a possible solution [82].

Likewise, standardization and interoperability remain open research topics, most industrial DTs are customized, feature that limits cross-platform integration. Efforts like the ISO 23247 framework [92] and modular IT architectures [83] offer possible solutions towards a standardized DT design.

The explainability of hybrid DTs is equally vital because engineers must be able to clearly interpret why the DT predicts faults, linking outcomes to tangible physical causes, such as pressure rise, temperature imbalance, or surge onset. Studies like Cruz-Manzo and Panov [90] highlighted how interpretable DTs can assist operators in making timely maintenance decisions without relying on unclear models.

Finally, cybersecurity and data integrity are becoming increasingly critical as well. As DTs rely on networked sensor data, vulnerabilities to cyber-attacks or communication failures could corrupt model outputs potentially leading to unsafe decisions. Future research is focusing on secure DTs integrating blockchain, federated learning and encrypted IoT channels [76].

1.9 Challenges and future perspectives for turbomachinery monitoring

Although the integration of ML, edge computing and DT technologies has improved and transformed turbomachinery CM, several technical and operational challenges limit a widespread industrial deployment. Among these, there are sensor reliability, data quality, UQ and real-time implementation, which are issues that directly influence the accuracy and trustworthiness of PdM systems.

Sensor reliability remains a fundamental bottleneck, due to the critical environments turbomachinery sensors are exposed to. Sensor drifts, calibration losses and intermittent failures are frequent, particularly for high-frequency pressure and vibration probes. Such degradation can propagate errors through the entire digital chain, from raw data acquisition to feature extraction and model inference. To mitigate these limitations, self-diagnostic sensors and adaptive calibration algorithms are being developed as emerging solutions, though standardization is still lacking.

Data quality and availability present parallel constraints as well. Industrial datasets can be often sparse, imbalanced, or contaminated by process disturbances. For supervised learning, the scarcity of fault-labelled data limits model generalization, while inconsistent data sampling disrupts temporal coherence. Future progress will depend on scalable data governance and automated data-curation frameworks, supported by edge-level preprocessing and robust anomaly rejection [76].

Real-time implementation is also a critical frontier. For example, the data output of high-frequency vibration and pressure systems often exceeds the processing capability of centralized analytics. For this reason, edge computing and fog architectures can be promising solutions to these limtations by moving computation closer to the machine, reducing latency and bandwidth consumption. Edge devices can execute lightweight inference models for rapid fault detection, while cloud infrastructures can handle deeper prognostic analysis and fleet-level learning [83].

Looking forward, the next generation of CM systems could increasingly rely on physics-informed machine learning (PIML) and hybrid DTs. PIML frameworks embed conservation laws, thermodynamic constraints and domain knowledge directly into the learning architectures of ML algorithms, combining the adaptability of data-driven models with the interpretability of physics. These models can infer physically consistent health parameters from limited data accelerating learning while maintaining transparency [80].

Similarly, hybrid DTs, integrating physical, statistical and artificial intelligence components, are evolving toward self-adaptive solutions capable of continuos learning and calibration. They will not merely mirror turbomachinery systems but also reason about them, autonomously identifying anomalies, estimating RUL and optimising operation under uncertainty. As computational resources become more distributed and standardised, hybrid

DTs are expected to form the core intelligence of future energy and propulsion systems, bridging the gap between monitoring, prediction and control.

In conclusion, the field of turbomachinery CM stands at the threshold of a new phase where trustworthy, interpretable and autonomous diagnostics will define a new industrial competitiveness. The fusion of physics-informed learning, secured IoT infrastructures and hybrid DTs will enable systems that not only detect faults but also understand their root causes, advancing the transition from corrective maintenance to predictive maintenance.

1.10 Structure of the thesis

This research followed a deliberate trajectory in which analytical, numerical and data-driven expertise was incrementally consolidated, ultimately leading to the design of DT-based diagnostic architectures capable of supporting industrial decision-making and maintenance actions. This aspect was particularly aligned with the industrial objectives of IIS. Throughout this trajectory, CFD-based analyses progressively interacted with experimental evidence, either directly or indirectly, in order to refine diagnostic hypotheses and assess their relevance under practical operating constraints.

The early chapters of the thesis were intentionally dedicated to the construction of a solid methodological and theoretical foundation, independent of any specific industrial asset. Ch. 0 establishes the conceptual background of CM and diagnostics for turbomachines, while Ch. 2 and 3 provide the necessary ML and CFD tools required to support data-driven and CFD-informed diagnostic approaches. The coupling between CFD and machine learning adopted here is CFD-informed rather than physics-informed in the strict sense: high-fidelity simulations are used both as a generator of labelled training data and as a forward model that is subsequently inverted to infer operating conditions, while the governing equations are not embedded as constraints within the learning process.

Building upon this foundation, Ch. 4 focuses on radial compressors for refrigerant applications and represents a first mature synthesis of the developed expertise. In this chapter, data-driven diagnostic methods are formulated and validated by means of high-fidelity, time-resolved CFD simulations, allowing the detection and characterisation of aerodynamic instabilities such as surge. From an industrial point of view, the natural complement of the present analysis would have consisted of a direct experimental validation of the proposed diagnostic strategies on an operational compressor.

However, such experimental validation was not possible within the timeframe of the doctoral programme due to logistical, economic and access-related constraints of typical industrial environments. This limitation did not limit the research scope but motivated instead a complementary and independent research activity on a different class of turbomachinery, namely wind turbines, for which experimental data and field validation were available.

Ch. 5, therefore, represents both a methodological evolution and a change of application domain. The focus shifts from enclosed turbomachinery operating under controlled boundary conditions to an unducted machine exposed to highly stochastic inflow, limited observability and strong coupling between aerodynamics, control and environmental effects. Within this context, the diagnostic expertise developed in the previous chapters is extended and adapted to the design and implementation of DT architectures for wind turbines.

In particular, the DTs presented in Ch. 5 are characterised by their ability not only to detect and diagnose aerodynamic performance deviations, such as yaw misalignment induced by sensor faults and blade surface degradation, but also to provide actionable indications to support maintenance decisions. This maintenance-oriented capability marks a key step towards industrial applicability and distinguishes the DTs developed in this thesis from purely monitoring-oriented approaches.

Finally, Ch. 6 summarises the main contributions of the thesis, examines methodological boundaries and outlines directions for future research for CM in turbomachinery.

2. Machine learning and data analysis tools for condition monitoring

2.1 Purpose and scope of the chapter

Machine learning (ML) has become one of the central analytical tools for modern CM. Ch. 0 introduced the reasons behind this shift, showing how classical techniques, which are based on either spectral indicators, thermodynamic models, or statistical thresholds, struggle to be consistent with the complexity and variability of contemporary turbomachinery operation. What is progressively more needed, and increasingly adopted, is a framework capable of learning patterns directly from data while retaining enough structure to remain interpretable and physically meaningful.

The scope of this chapter is to present the theoretical foundations of the ML methods used throughout the thesis. The focus of the present chapter is not on applications, which will appear later, but on the mathematical principles and algorithms that support them. The chapter, therefore, reviews the formulation of supervised and unsupervised learning problems, the construction of typical loss functions and the optimisation procedures used to train models. Particular attention is given to techniques for feature extraction and signal metrics, such as the Fast Fourier Transform (FFT) and the approximate entropy (ApEn), together with regression and classification models, including artificial neural networks (ANNs) and support vector machines (SVMs).

While the discussion aims to provide a general overview, the presentation is guided by the needs of turbomachinery diagnostics. The models considered here are those most frequently encountered when dealing with vibration, pressure or power-related signals, where the distinction between noise, normal variability and genuine degradation is subtle. For this reason, the mathematical formulations are introduced with care, together with the assumptions that make each method appropriate for certain classes of monitoring problems. Unless otherwise noted, all derivations of formulas for the fundamentals of ML, such as loss functions, optimisation techniques, classification and regression equations, as well as ANNs, rely on established standard texts in this field, as outlined by Bishop [93], Murphy [94] and Goodfellow, Ian, et al. [95].

In summary, Ch. 2 defines the analytical tools that will be used to build the diagnostic models and DT architectures later developed in the thesis. The intention is not to provide an exhaustive review of all ML algorithms, but to establish a coherent and rigorous basis on which the subsequent chapters can rely.

2.2 Fundamentals of machine learning

2.2.1 Learning problem formulation

ML provides a structured way to extract relationships or patterns from data and its formulation depends on the type of information available during the training phase. In the most general setting, a learning algorithm receives a collection of samples

$$\mathcal{X} = \{x^{(i)}\}_{i=1}^{m}, \tag{2.1}$$

where each $x^{(i)} \in \mathbb{R}^n$ represents a point in the input space $\mathcal{X}$. Depending on the learning paradigm (supervised or unsupervised, see Sect. 2.2.2), additional information may or may not be provided together with the inputs.

At the core of supervised learning lies a dataset Ɗ composed of m examples,

$$Ɗ = \{(x^{(i)}, y^{(i)})\}_{i=1}^{m}, \tag{2.2}$$

where each $x^{(i)} \in \mathbb{R}^n$ is a vector of input features and the vector $y^{(i)} \in \mathcal{Y}$ denotes the corresponding target and depends on the type of problem one is trying to solve. The aim is to identify a function

$$h_\theta: \mathbb{R}^n \to \mathcal{Y}, \tag{2.3}$$

parameterised by the parameters θ, which approximates the underlying mapping between x and y. The collection of all admissible functions h_θ defines the hypothesis space $\mathcal{H}$. Selecting a suitable hypothesis within $\mathcal{H}$ reduces the problem to choosing the parameters θ that best describes the data according to a chosen criterion. This means that the model will try to find θ so that for all training samples,

$$h_\theta(x^{(i)}) \approx y^{(i)} \tag{2.4}$$

In contrast, unsupervised learning operates only with the input set $\mathcal{X}$, without the associated targets. In this case, the objective is to uncover latent structure, cluster organisation or lower-dimensional representations of the data.

Once the learning paradigm is defined, the next step is to quantify model performance through a loss function and its cost function, as discussed in Sect. 2.2.3.

2.2.2 Supervised and unsupervised learning paradigms

ML includes several distinct learning paradigms; each one is characterised by the type of information available during the training phase and by the nature of the task to be solved. Although this thesis primarily relies on supervised learning, a brief overview of the broader ML landscape can be useful to frame the methods introduced in this chapter. The conceptual relationships among the main paradigms are depicted in Figure 2.1.

Supervised learning, as seen before, relies on a dataset consisting of input and output pairs $(x^{(i)}, y^{(i)})$. The objective is to learn a function that maps inputs to their corresponding outputs based on examples. This paradigm includes algorithms for:

- Regression, where the target variable is continuous.
- Classification, where the target consists of discrete labels.

Most diagnostic tasks in turbomachinery fall under this category, as they involve either predicting a physical quantity (e.g., pressure, power) or identifying a machine state (e.g., healthy, degraded, faulty). Supervised models learn by comparing their predictions to the known outputs and adjust their parameters to minimise a chosen loss function, as later discussed in Sect. 2.2.3.

Unsupervised learning works without labelled outputs. Its goal is to find latent structures within data by analysing the distribution, geometry, or statistical regularities of the input space. Typical tasks include:

- Clustering, which consists of identifying groups of similar samples, with algorithms such as K-Means, Gaussian Mixture Models.
- Dimensionality reduction, that is, projecting high-dimensional data into a lower-dimensional space, methods include the PCA.
- Density estimation, modelling the underlying probability distribution.

As previously introduced, in turbomachinery monitoring, unsupervised methods are often selected to characterise the nominal operating behaviour of the asset before any fault labels exist, to detect outliers in multivariate signals or to compress high-dimensional datasets into more interpretable representations.

Reinforcement learning (RL) differs fundamentally from both supervised and unsupervised learning paradigms. In RL, an agent interacts with an environment by taking actions, receiving feedback in the form of rewards or penalties. Over time, the agent learns a policy that maximises cumulative reward. Although RL is not used in this thesis, it is central to applications involving sequential decision-making or control, such as active flow control, adaptive control strategies or maintenance scheduling in energy systems.

Deep learning is another subfield of ML and refers to a family of models, primarily ANNs, that learn hierarchical representations of data by composing multiple layers of nonlinear transformations. DL can operate:

- in a supervised manner (e.g., deep classifiers or regressors),

- in an unsupervised manner (e.g., autoencoders, generative models),
- or as part of an RL agent (e.g., deep Q-networks).

Its strength lies in modelling complex, nonlinear relationships and temporal or spatial structures that are difficult to express using traditional methods.

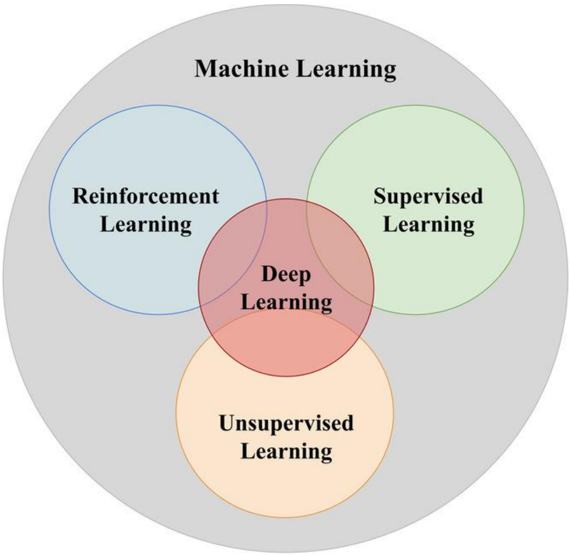

Figure 2.1. A conceptual representation of the relationships among supervised learning, unsupervised learning, reinforcement learning and deep learning (adapted from [96]).

2.2.3 Loss functions and cost function minimisation

This section introduces the metrics to quantify the model's performance. The theoretical content is presented through the lens of the supervised learning approach. The first step consists in introducing a loss function $L(h_\theta(x), y)$ that measures the discrepancy between the model prediction $h_\theta(x)$ and the true value y. The overall model performance on the dataset is described through the cost function:

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} L\left(h_\theta\left(x^{(i)}\right), y^{(i)}\right), \tag{2.5}$$

and learning consists of minimising $J(\theta)$ with respect to θ. In the case of regression problems, the most common choice for L is the mean squared error (MSE):

$$L_{\text{MSE}} = \frac{1}{2}(h_\theta(x) - y)^2, \tag{2.6}$$

which penalises large deviations and leads to a convex optimisation problem for linear models. In classification tasks, especially those involving multiple classes, the preferred choice is often the categorical cross-entropy loss:

$$L_{CCE} = -\sum_{k=1}^{C} y_k log(h_\theta(x)_k), \quad (2.7)$$

naturally arising from a probabilistic interpretation of the model output. Under this formulation, $h_\theta(x)$ approximates the conditional probability $P(y = 1 \mid x)$, so minimising the cross-entropy loss corresponds to maximising the likelihood of the observed labels.

A robust loss function used for regression tasks that combines the best properties of the MSE and mean absolute error (MAE) is the Huber loss function:

$$\begin{cases} L = \frac{1}{2}\left(y_k - h_\theta(x)\right)^2, & |y_k - h_\theta(x)| \leq \delta \\ L = \left(|y_k - h_\theta(x)| - \frac{1}{2}\delta\right)^2, & |y_k - h_\theta(x)| > \delta \end{cases} \quad (2.8)$$

with δ being a threshold parameter. The minimisation of the cost function is typically performed through numerical optimisation. Among the different optimisation algorithms, there is gradient descent, which iteratively updates the parameters along the direction of steepest decrease of the cost function (the negative gradient) until it reaches a minimum point representing the best model parameters:

$$\theta \leftarrow \theta - \alpha(\nabla_\theta J(\theta)), \quad (2.9)$$

with α denoting the learning rate. Figure 2.2 shows a graphical representation of how the optimisation algorithm works.

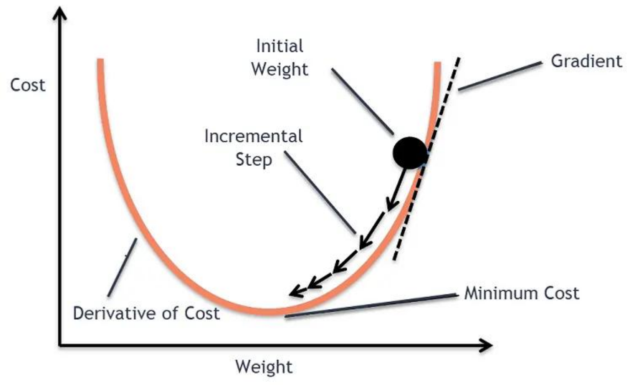

Figure 2.2. Representation of how the gradient descent algorithm works.

Variants include the stochastic gradient descent, where parameters are updated using a subset of the training samples; in this way additional stochasticity is introduced, often resulting in improved convergence for large datasets, see Figure 2.3.

Other algorithms, including momentum, RMSProp and Adam, further adapt the update direction based on the curvature or past gradients and are commonly employed in nonlinear models such as ANNs.

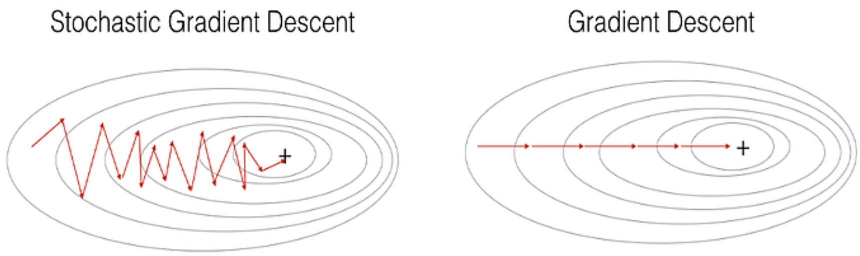

Figure 2.3. Comparison between the optimisation process of the stochastic gradient descent vs. the gradient descent.

2.2.4 Generalisation, overfitting and the bias-variance trade off

A central concern in ML is not simply the model's performance on the training data but its ability to generalise to new, unseen samples. A model that exactly fits the training examples may capture noise or transient fluctuations rather than the underlying physical behaviour. This phenomenon is known as overfitting and results in poor predictive capability outside the training set. Conversely, a model that is overly simplistic may fail to capture meaningful structure in the data, leading to what is known as underfitting (Figure 2.4).

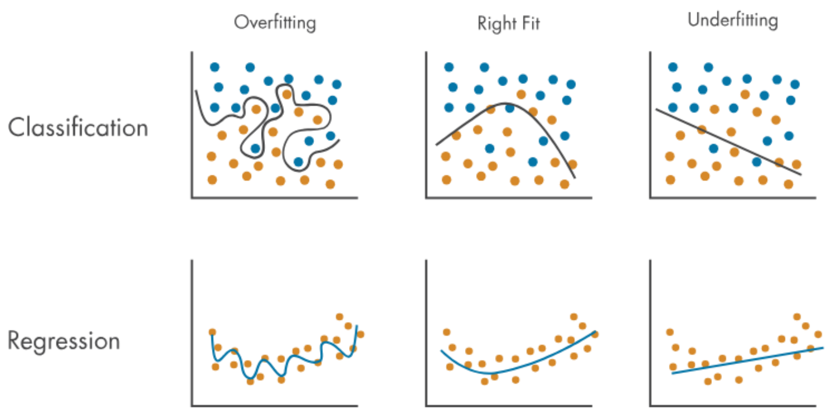

Figure 2.4. Comparison between overfitting, right fitting and underfitting (adapted from [97]).

These two extremes are often described through the bias-variance decomposition, where bias measures the error introduced by overly restrictive modelling assumptions and variance quantifies the sensitivity of the model to the randomness of the training data.

Achieving good generalisation requires an appropriate balance between these two contributions. Several strategies are used to mitigate overfitting: restricting the complexity of the hypothesis space, introducing regularisation terms that penalise large parameter values or employing early stopping in iterative optimisation schemes.

The data would normally be divided into training data, validation data and testing data. Despite this being dataset-dependent, normally 60-80 % would be dedicated to training data in particular, with the rest being divided relatively equally across the test and validation data (20-10 % each). This would provide an unbiased measure of the model's accuracy.

2.2.5 Role of feature extraction

Before model training, raw data must be transformed into data suitable for learning. This process is based on feature extraction and often includes statistical descriptors, spectral components obtained through Fourier analysis or complexity measures such as ApEn. Once features are extracted from data, they are normalised or rescaled (such as Min-Max scaling or Z-scaling) to ensure all inputs contribute equally to the learning process and to prevent features with larger numerical ranges from dominating the optimisation.

It is important to note that, the quality of those features makes all the difference in modelling or learning. For vibration or unsteady aerodynamic data in particular, features based on temporal patterns or frequency content may be significantly more informative compared to time-series samples.

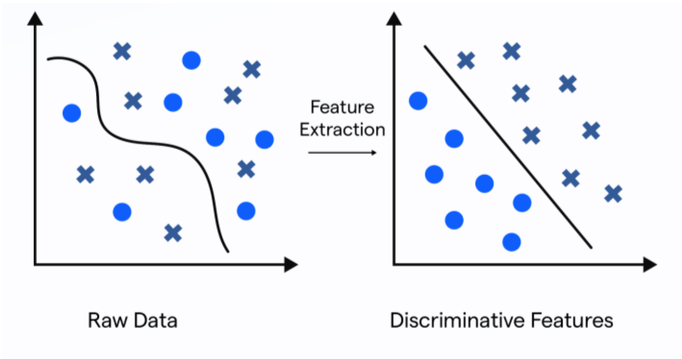

Figure 2.5. Graphical representation of the feature extraction process.

2.3 Supervised learning methods

Supervised learning represents the most commonly adopted paradigm in turbomachinery diagnostics, where the goal is to estimate a physical quantity or classify the health state of a component based on measured or derived features. Once the learning framework is formulated, as introduced in Sect. 2.2.1, the next step involves identifying the models that are able to handle the data inputs and predict the corresponding data outputs. This choice relies upon the complexity of the data relationship being learnt.

This section presents the main families of supervised models commonly used in engineering applications: linear and logistic regression, margin-based classifiers such as SVMs and ensemble methods. A brief overview of ANNs is also provided, with a dedicated section appearing later in the chapter due to their central role in modelling nonlinear phenomena (see 2.6).

2.3.1 Linear and logistic regression

Linear regression is one of the simplest supervised methods and provides a useful baseline for more complex models. The underlying assumption is that the target y varies approximately linearly with the input features x. The hypothesis function takes the form

$$h_\theta(x) = \theta^\top x, \tag{2.10}$$

and the parameters θ are determined by minimising the mean squared error introduced in Sect. 2.2.3. Even if linear models are appealing because of the computational efficiency and ease of interpretation, they are not effective in modelling scenarios in which there are high nonlinearities in the relationship between inputs and outputs that exist in vibration or aerothermal data.

A closely related model is logistic regression, which extends linear models to binary classification problems. Instead of predicting a continuous output, logistic regression estimates the probability that a sample belongs to a given class by passing the linear combination of features through a sigmoid function σ that maps any real number $h_\theta(x)$ (from -∞ to +∞) to a value in the range (0, 1):

$$\sigma(\theta^\top x) = p(\,y = 1 \mid x\,) = \frac{1}{1 + \exp(-\theta^\top x)}. \tag{2.11}$$

The decision boundary is therefore given by a hyperplane $\theta^\top x = 0$, while the probabilistic output is used to compute the cross-entropy loss (see Sect. 2.2.3). Logistic regression works well in situations where the boundary between the classes is more or less linear. However, in many diagnostic applications, there are nonlinear interactions between the variables requiring more flexible models.

2.3.2 Support vector machines

SVMs (support vector machines) address the limitations of linear classifiers by adopting a margin-based formulation. Instead of fitting the data directly, the idea is to find a separating hyperplane that maximises the distance between the decision boundary and the closest points from each class, respectively defined as support vectors. Only these points determine the margin, whereas all others play no role in the optimisation. From a diagnostic viewpoint, support vectors effectively represent the samples that define the boundary between normal and degraded states, their role becomes crucial when dealing with high-dimensional feature spaces or subtle differences across operating conditions.

The distance to the decision boundary is defined as geometric margin and its maximisation leads to a classifier that is capable of generalising well, even with relatively small datasets.

For a binary classification problem with labels $y^{(i)} \in \{-1, +1\}$, the linear SVM assumes a decision function of the form:

$$f(x) = w^{\top}x + b \tag{2.12}$$

where $w \in \mathbb{R}^n$ is the normal vector to the separating hyperplane and $b \in \mathbb{R}$ is the bias term that shifts the hyperplane. The decision boundary corresponds to the set of points for which $f(x) = 0$, see the following Figure 2.6.

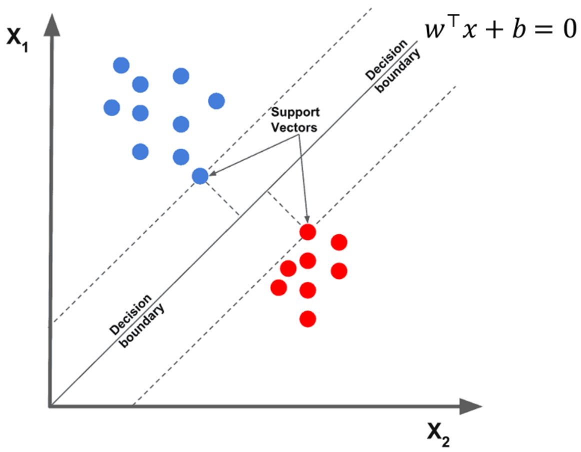

Figure 2.6. Decision boundary of an SVM.

The geometric margin of the classifier is formulated as:

$$\text{margin} = \frac{1}{\| w \|}. \tag{2.13}$$

The magnitude of w determines the orientation of the hyperplane, a smaller $\| w \|$ implies a larger margin and therefore a more robust classifier. Finding the maximum of this margin leads to a classifier that is robust to noise and generalises well. The optimisation problem can be formulated by adopting the hinge loss:

$$L_{\text{hinge}} = max\left(0, 1 - y^{(i)} f\left(x^{(i)}\right)\right), \tag{2.14}$$

together with a regularisation term that controls model complexity. The resulting cost function is

$$J(w,b) = \frac{1}{2} \parallel w \parallel^2 + C \sum_{i=1}^{m} L_{\text{hinge}}^{(i)}, \tag{2.15}$$

where the parameter $C > 0$ controls the trade-off between maximising the margin and penalising misclassifications. A large C encourages the model to fit the training data closely, potentially at the risk of overfitting. In contrast, a small C allows more misclassifications in favour of a wider margin and improved generalisation. The choice of C is therefore critical in diagnostic contexts, where training data may contain measurement noise or slight inconsistencies. Parameter optimisation is typically done through gradient-based or coordinate-ascent methods, building on the algorithms introduced in Sect. 2.2.3.

The linear SVM decision function can also be rewritten in terms of support vectors as:

$$f(x) = \sum_{i \in \mathcal{S}} \alpha_i y^{(i)} \left(x^{(i)}\right)^{\top} x + b, \tag{2.16}$$

where $\mathcal{S}$ is the set of support vectors, α_i (Lagrange multipliers) are non-negative dual coefficients obtained through the optimisation process, x is the new data point whose class is being predicted, $x^{(i)}$ is a support vector (a training data point) and $y^{(i)}$ is the true class label for the support vector $x^{(i)}$. This representation shows the inner product $\left(x^{(i)}\right)^{\top} x$, through which the classifier evaluates similarity between the input point and each support vector. The stronger the alignment between x and $x^{(i)}$, the larger its influence on the decision function.

This inner-product form naturally leads to the introduction of kernels, a key strength of SVMs. The kernel method is a strong tool that allows nonlinear classification by implicitly mapping the input data into a higher-dimensional space without explicitly computing the transformation. This is achieved by replacing every inner product $\left(x^{(i)}\right)^{\top} x$ with a kernel function:

$$K\left(x^{(i)}, x\right) = \phi\left(x^{(i)}\right)^{\top} \phi(x). \tag{2.17}$$

where ϕ is the function that maps the original data into the new, often higher-dimensional, feature space. Common kernels include:

- The Radial Basis Function (RBF) kernel:

$$K\big(x^{(i)}, x^{(j)}\big) = \exp\big(-\gamma \parallel x^{(i)} - x^{(j)} \parallel^2\big). \tag{2.18}$$

- The polynomial kernel:

$$\big(x^{(i)}, x^{(j)}\big) = ((x^{(i)})^\top x^{(j)} + c)^p. \tag{2.19}$$

The indices i and j refer to different training samples within the dataset, the parameter γ in the RBF kernel controls the radius of influence of each support vector, a large γ leads to highly flexible boundaries that might overfit; on the other hand, a small γ produces smoother and more global boundaries. In diagnostics, the (C, γ) parameter pair must be tuned carefully to avoid over classifying normal operational variability as a fault state, while still maintaining sensitivity to true degradation phenomena.

Because SVMs rely on support vectors rather than the full dataset, they remain effective in the moderate-data regime typical of turbomachinery monitoring systems. Their robustness to measurement noise, ability to handle nonlinear boundaries through kernels and strong generalisation properties make them suitable for fault discrimination tasks.

These advantages are applied later in Ch. 5 through an SVM classifier used to infer wind turbine blade roughness states through a DT-based model. This model demonstrates how aerodynamic features and experimental measurements can be combined within an SVM to detect subtle changes in turbine performance and surface condition.

2.3.3 Decision trees and ensemble methods

Decision trees provide another modelling technique called recursive partitioning. In each splitting step, the algorithm selects the variable and threshold value which separates observations best based on a criterion such as Gini impurity or information gain. The final model consists of a hierarchy of decision rules leading to a set of leaf nodes, each associated with a prediction.

Although trees are very intuitive and easy to understand, the individual tree tends to overfit the data. Ensemble methods mitigate this issue by combining the predictions of multiple, distinct base models (or weak learners) to produce a single, better predictive model. This approach is based on the diversity and collective strength of many models to reduce the overall variance and bias. Methods that work on the above concept are:

- Bagging, which builds numerous trees on each bootstrap sample and averages their predictions, reducing variance.
- Random Forests (RFs), which are an extension of bagging and choose subsets of features randomly at each split to increase robustness.

- Gradient Boosting Machines (e.g. XGBoost, LightGBM), which build trees sequentially, each to correct the errors made by the preceding ones and leading to highly accurate models.

These methods are common in CM due to their ability to handle heterogeneous features, mixed-type variables and complex nonlinear interactions. They also provide measures of feature importance, which can help interpret diagnostic outcomes.

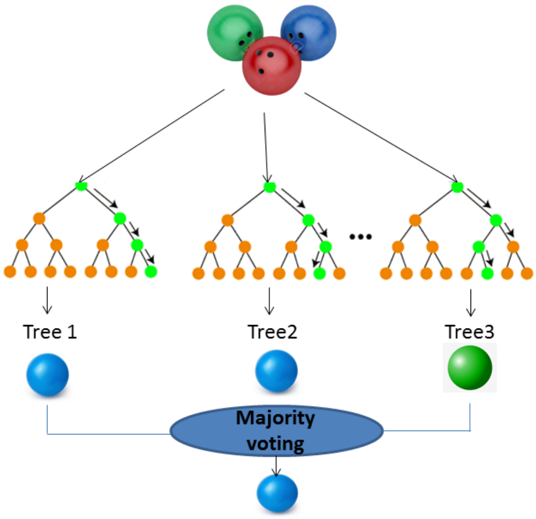

Figure 2.7. Schematic representation of how a RF works.

2.3.4 Artificial neural networks: a first overview

Included into the class of supervised learning models, there are also artificial neural networks (ANNs), or simply neural networks (NNs), a wide family of models able to learn highly nonlinear mappings through multilayer compositions of simple units. In the simplest case, a NN learns to predict by sequentially combining transformations and nonlinear activation functions on the input variables. The modelling power of NNs to approximate nonlinear functions can be proven by the universal approximation theorem [98].

NNs are key enablers for the diagnostic methods later employed in the studies of Ch. 4. Consequently, detailed information on NNs is further provided in Sect. 2.6; in this section, it is sufficient to note that NNs encompass a large variety of possible architectures, from fully connected feed-forward networks to convolutional and recurrent models and their training procedures utilize techniques of gradient-based optimization introduced in Sect. 2.2.

2.3.5 Model selection for turbomachinery diagnostics

Selecting the most appropriate supervised learning algorithm depends on what is to be achieved and what data is used. Linear and logistic regression are always applicable when dealing with simple correlations, besides serving as a reference. SVMs and ensemble methods excel when datasets are moderate in size but contain nonlinear patterns. NNs become advantageous when the problem demands high expressiveness or involves temporal structures, provided that enough training data are available.

Across these methods, the criteria which are utilised to select the most appropriate model include the ability to generalise, robustness to noisy data, interpretability and the associated computational cost of model deployment into a CM system.

2.4 Unsupervised learning and dimensionality reduction

Unsupervised learning refers to a family of methods designed to find a structure in data when no output labels are directly available. In the field of turbomachinery diagnostics, these techniques are often used to characterise normal operating behaviour, detect unusual or emerging patterns in multivariate sensor streams, or compress high-dimensional information into a reduced number of meaningful variables. Contrary to supervised models, which rely on an explicit input to output relationship, unsupervised methods operate only on the statistical properties of the input space provided.

This section follows the main unsupervised techniques relevant to the framework of this thesis, although no model was directly used for this research. Clustering methods, such as K-Means and Gaussian Mixture Models are introduced first, followed by a detailed derivation of Principal Component Analysis (PCA), which plays a central role in multivariate statistical process control (MSPC) and in the feature extraction pipeline introduced in Ch. 0. The section is concluded with a brief overview of autoencoders, bridging classical dimensionality-reduction methods and modern DL methods.

2.4.1 Clustering

Clustering groups of samples into coherent subsets based on similarity, without the need for explicit labels. Two of the most selected algorithms are K-Means clustering and Gaussian Mixture Models (GMMs), which differ in both their underlying assumptions and the resulting cluster shapes.

K-Means clustering. It is a method that partitions the dataset into k clusters by minimising the within-cluster variance. Given a set of input samples $\{x^{(i)}\}_{i=1}^{m}$, the algorithm seeks k centroids $\{\gamma_j\}_{j=1}^{k}$ to minimise the following cost function:

$$J = \sum_{i=1}^{m} \| x^{(i)} - \gamma_{c(i)} \|^2, \tag{2.20}$$

where $c(i)$ denotes the cluster assignment for the sample $x^{(i)}$. The algorithm alternates between:

- Assignment step:

$$c(i) = argmin_j \parallel x^{(i)} - \gamma_j \parallel^2. \tag{2.21}$$

- Update step:

$$\gamma_j = \frac{1}{\mid C_j \mid} \sum_{i \in C_j} x^{(i)}. \tag{2.22}$$

Despite its simplicity, K-Means is effective in identifying compact, spherical clusters in high-dimensional spaces (Figure 2.8). Nevertheless, this can also result in a limitation of the same algorithm; with non-spherical clusters (e.g., elongated, crescent-shaped or nested) the method will struggle to separate them into classes. It will attempt to cut the non-spherical clusters into spherical sections around their centroids, leading to poor clustering results.

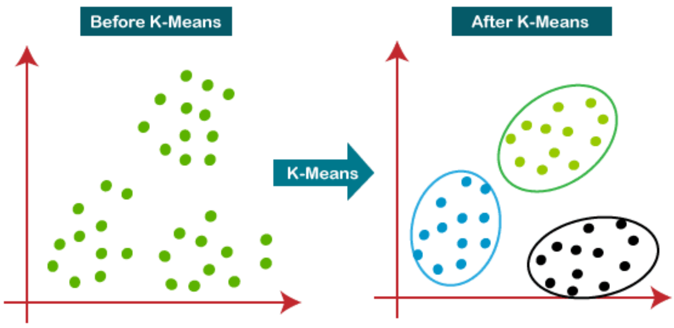

Figure 2.8. Graphical representation of how K-Means clustering works.

Gaussian Mixture Models. Given the previous limitations, GMMs extend clustering to situations where clusters are not spherical or equally sized. GMMs assume that data are generated by a weighted sum of k Gaussian components:

$$p(x) = \sum_{j=1}^{k} \pi_j \, \mathcal{N}(x;\, \mu_j, \Sigma_j), \tag{2.23}$$

where π_j are mixture weights and $\mathcal{N}(x; \mu_j, \Sigma_j)$ denotes the multivariate normal density evaluated at the point x, characterised by its own mean μ_j and covariance matrix Σ_j. Parameter estimation is typically performed via the Expectation-Maximisation (EM) algorithm, which alternates between:

1. E-step: computing posterior probabilities that a point belongs to a cluster j,
2. M-step: updating π_j, μ_j and Σ_j.

Because GMMs allow full covariance structures, they can capture elongated, anisotropic or overlapping clusters, situations frequently encountered in physical systems where operating conditions produce correlated patterns across sensors, (Figure 2.9).

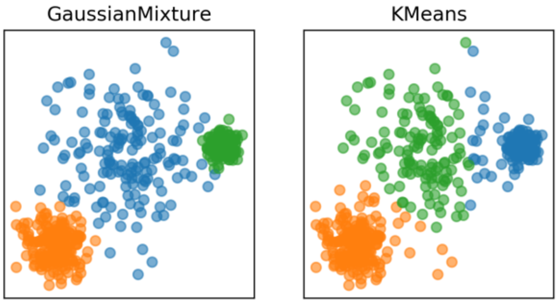

Figure 2.9. GMM vs. K-Means clustering.

2.4.2 Principal Component Analysis

PCA (Principal Component Analysis) is among the most powerful tools for dimensionality reduction and multivariate monitoring. The main characteristic of this method is finding a set of orthogonal directions, namely principal components, along which the variance of the dataset is maximised. This transformation enables to represent high-dimensional measurements with a reduced number of latent variables, while retaining most of the informative content.

Given a dataset $\{x^{(i)}\}_{i=1}^{m}$ with $x^{(i)} \in \mathbb{R}^n$. Before applying PCA, it is common practice to normalise the data, which typically involves subtracting the mean and scaling each variable so that all features contribute comparably. If this step is not followed, variables measured on larger numerical scales might dominate the principal components, leading to misleading results. Next, an empirical covariance matrix Σ can be defined with the following formula:

$$\Sigma = \frac{1}{m}\sum_{i=1}^{m} x^{(i)}(x^{(i)})^{\top}. \tag{2.24}$$

This matrix extracts the linear correlations among the variables, with its eigenstructure, it encodes the directions of maximal variance. At this step, PCA can be obtained by solving

$$\Sigma v_j = \lambda_j v_j, \tag{2.25}$$

where λ_j are eigenvalues and v_j the associated eigenvectors. By convention, the eigenvalues are ordered as follows:

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq 0, \tag{2.26}$$

so that the first principal component corresponds to the direction of maximum variance.

Let $V_k = [v_1, \ldots, v_k]$ be the matrix of the first k eigenvectors. The projection of each sample onto this reduced space is

$$z^{(i)} = V_k^\top x^{(i)}. \tag{2.27}$$

The vector $z^{(i)} \in \mathbb{R}^k$ contains the reduced set of latent variables containing the dominant patterns within the data. These components are often used in MSPC frameworks, as introduced in Ch. **1**, to monitor deviations from nominal operation via statistical metrics such as Hotelling's T^2 and the SPE index [60–62].

The contribution of the first k principal components to the total variance is quantified through the following explained-variance ratio (EV):

$$\mathrm{EV}(k) = \frac{\sum_{j=1}^{k} \lambda_j}{\sum_{j=1}^{n} \lambda_j}. \tag{2.28}$$

Selecting the smallest k that achieves a desired explained-variance ratio provides a balance between dimensionality reduction and information retention. Figure 2.10 presents, as an example, the first two components determined for a sample dataset.

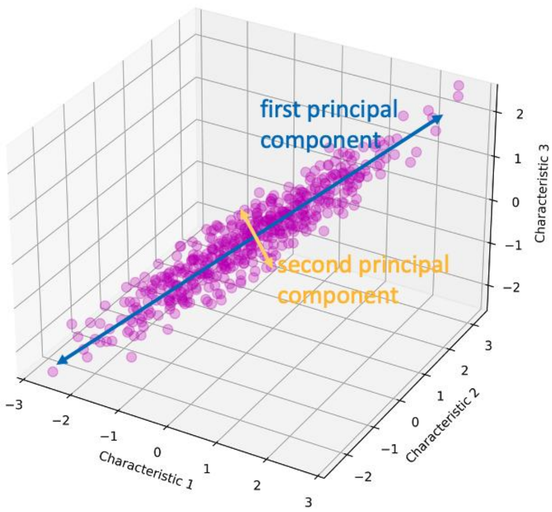

Figure 2.10. First two principal components of a sample dataset (adapted from [99]).

2.4.3 Autoencoders

Autoencoders represent another useful tool for dimensionality reduction. Using NNs, they extend the idea to a nonlinear regime to learn compressed representations (Figure 2.11). Trained in an unsupervised manner, an autoencoder generally consists of two parts: an encoder and a decoder. The encoder maps the input x to a low-dimensional latent vector z, on the other hand, the decoder reconstructs x from z. During training, the network minimises a reconstruction loss such as

$$L = \| x - \hat{x} \|^2. \tag{2.29}$$

This forces the latent representation to capture the essential structure of the given input. Unlike PCA, which has the limitation of performing to linear transformations, autoencoders can learn curved manifolds and even complex nonlinear patterns, making them useful for anomaly detection, denoising and feature extraction in high-dimensional sensor data.

In turbomachinery diagnostics, autoencoders can serve as nonlinear extensions of PCA, enabling the extraction of more expressive low-dimensional features, for instance, from multichannel pressure or vibration measurements, especially when the underlying relationships are hard to be well captured by linear models.

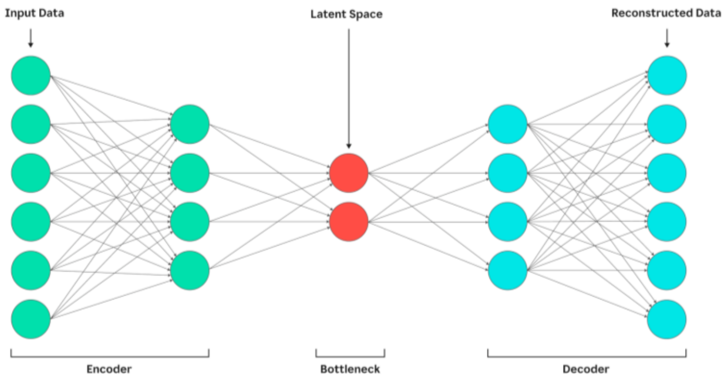

Figure 2.11. Example of an autoencoder architecture.

2.5 Feature extraction and signal metrics

Modern diagnostic approaches rarely operate directly on raw time-series measurements. Instead, informative quantities, typically known as features, are extracted from signals to derive relevant physical information into a reduced numerical representation that is more convenient for statistical modelling. In turbomachinery monitoring, where vibration, pressure, velocity or temperature signals may contain subtle indicators of degradation, the quality of the extracted features directly affects the overall success of the diagnostic model.

This section introduces the most commonly used features in the time and frequency domains, together with a detailed derivation of approximate entropy (ApEn), a non-linear complexity metric, relevant to the compressor studies described in Ch. 4. While feature selection relies on the nature of the signal and the associated fault mechanism, the general goal remains converting raw physical measurements into meaningful diagnostic fingerprints.

2.5.1 Time-domain features

Time-domain features quantify statistical characteristics of the time series as it evolves in time. These types of descriptors are characterised to be computationally inexpensive and are thus widely used in real-time monitoring systems.

Root Mean Square (RMS). RMS measures the average signal energy and it is sensitive to increases in oscillation amplitude. Given a discrete signal $\{x(t_i)\}_{i=1}^{N}$, the RMS is computed as:

$$\mathrm{RMS} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} x(t_i)^2}\,. \tag{2.30}$$

In turbomachinery, a rising RMS over time may be a symptom of faults such as increased vibration intensity due to imbalance, misalignment or aerodynamic instabilities.

Variance. Variance quantifies the dispersion of a signal relative to its mean $\bar{x}$:

$$\mathrm{Var}(x) = \frac{1}{N}\sum_{i=1}^{N} (x(t_i) - \bar{x})^2\,. \tag{2.31}$$

It helps detect changes in operating modes or fluctuations in flow and operating conditions.

Skewness. This type of feature measures waveform asymmetry based on the signal's mean and standard deviation σ:

$$\mathrm{Skew}(x) = \frac{1}{N}\sum_{i=1}^{N} \left(\frac{x(t_i) - \bar{x}}{\sigma}\right)^3\,. \tag{2.32}$$

The sign of skewness can serve as an indicator of possible abnormal behaviour. In particular, positive skewness indicates more positive outliers, while negative skewness indicates asymmetric behaviour such as intermittent stalls or localised shedding phenomena.

Cross-correlation. Cross-correlation is used to quantify the similarity between two signals $x(t)$ and $y(t)$:

$$R_{xy}(\tau) = \sum_{i=1}^{N} x(t_i)\, y(t_i + \tau)\,, \tag{2.33}$$

where τ denotes the time lag by which the signal $y(t)$ is shifted relative to $x(t)$. Positive values of τ correspond to a forward shift, while negative values indicate that y precedes x. It is used to detect time shifts, travelling flow structures and spatial coherence across sensors, particularly in compressor surge or rotating stall studies.

Residual errors. Residual features compare a measured signal $x(t)$ with a reference or model-generated signal $\hat{x}(t)$, among them compare the root mean square error (RMSE):

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x(t_i) - \hat{x}(t_i))^2} \tag{2.34}$$

and the coefficient of determination (R^2), ranging from 0 to 1:

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(x(t_i) - \hat{x}(t_i))^2}{\sum_{i=1}^{N}(x(t_i) - \bar{x})^2}. \tag{2.35}$$

Both quantify deviation from nominal behaviour and large deviations can indicate degradation (e.g., cooling failure or roughness-induced aerodynamic anomalies). The R^2 can serve as an easy quantity to check for model comparison, the closer its value is to one, the more similar the analysed signal is to the reference one.

2.5.2 Frequency-domain features

Time-domain signals in turbomachines include periodic phenomena, harmonic components or broadband variations related to fluid-mechanical processes, for example, rotor-stator interaction, blade passings, vortex shedding, rotating stalls and surge phenomena. The conversion of a signal to the frequency domain allows an investigation of the phenomena listed above to identify corresponding characteristic patterns in the flow processes.

For these reasons, frequency-domain features extracted through the Fast Fourier Transform (FFT) play a central role in turbomachinery diagnostics. Following the work of Cooley & Tukey [100], given a discrete signal $\{x(t_i)\}_{i=1}^{N}$ sampled at uniform intervals, the FFT efficiently computes the *Discrete Fourier Transform (DFT)*:

$$X(k) = \sum_{i=1}^{N} x(t_i)\, e^{-j2\pi(i-1)(k-1)/N}\,. \tag{2.36}$$

Where $i = 1, \dots, N$ indexes the time samples $k = 1, \dots, N$ indexes the frequency bins, j is the imaginary unit ($j^2 = -1$) and the exponential term represents the basis oscillations at discrete frequencies. The result $X(k)$ is a complex number whose magnitude and phase describe the amplitude and timing of the corresponding frequency component.

The index k cannot represent directly physical frequencies; instead, it must be mapped to the dimensional frequency axis via:

$$f_k = (k-1)\, f_s/N, \tag{2.37}$$

where f_s is the sampling frequency. From the magnitude spectrum $|\,X(k)\,|$, various features can be computed, such as dominant harmonic frequencies, spectral peaks and sideband energy. As an example, Fig. 2.12 illustrates an FFT spectrum obtained from a sensor located in the diffuser passage of a compressor operating under stalled conditions [101].

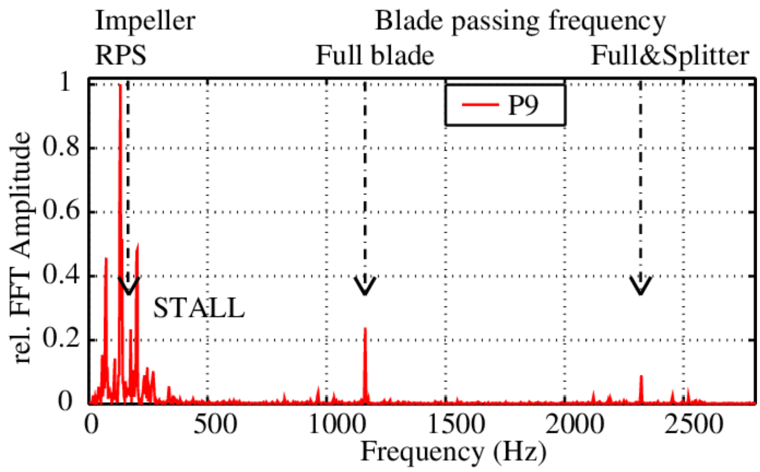

Figure 2.13. Example, FFT at stalled condition for a sensor in the diffuser passage of the large-scale rig (adapted from [101]).

Power Spectral Density (PSD). PSD quantifies how the signal's power is distributed across different frequencies f_k:

$$\mathrm{PSD}(f_k) = \frac{|\,X(k)\,|^2}{N\,\Delta t}. \tag{2.38}$$

Where $\Delta t = 1/f_s$ is the sampling interval. PSD is generally used to detect flow instabilities, rotating stall signatures or blade-passing frequencies.

Spectral Entropy. measures frequency-domain disorder by analysing how the signal's energy is distributed across the spectrum, analogously to approximate entropy in the time domain (see Sect. 2.5.3). By defining the normalised spectral power as:

$$P_k = \frac{\mid X(k) \mid^2}{\sum_{j=1}^{N} \mid X(j) \mid^2}, \tag{2.39}$$

the spectral entropy is computed as

$$H_s = -\sum_k P_k \log(P_k). \tag{2.40}$$

Higher spectral entropy corresponds to broad and flat spectra, typical of noisy conditions, whereas lower values indicate periodic or harmonically dominated behaviour.

Band Energy. Given a frequency band $\mathcal{B} \subseteq [0, f_s/2]$:

$$E_{\mathcal{B}} = \sum_{k:\, f_k \in \mathcal{B}} \mid X(k) \mid^2. \tag{2.41}$$

Band energy captures physical processes occurring in specific ranges (e.g., low-frequency stall, mid-frequency vibration, high-frequency turbulence).

2.5.3 Approximate entropy

Approximate entropy (ApEn) is a non-linear complexity measure that quantifies the predictability or regularity of a time series in the time domain, and it was formally introduced by Pincus in 1991 [57]. Despite the name, there is no direct connection to the thermodynamic entropy and unlike purely statistical descriptors (variance, skewness) or spectral indicators (PSD), ApEn analysis well captures structural irregularities, making it particularly useful for detecting early degradation or chaotic transitions in fluid mechanical systems.

ApEn operates by reconstructing the underlying dynamics of the signal through delay embedding, a concept originating from Takens' theorem [102]. The theorem provides a conceptual guideline: patterns must repeat at least twice within the analysed window. Given a signal $\{x(t_i)\}_{i=1}^{N}$ sampled at uniform intervals (Figure 2.14), ApEn is computed through three main steps: embedding, neighbourhood evaluation and entropy estimation. Embedding represents the first step; it involves the construction of a sequence of embedded vectors of embedding dimension m:

$$u^{(i)} = [x(t_i), x(t_i + t_d), \ldots, x(t_i + (m-1)t_d)]. \tag{2.42}$$

Where t_d is the time delay defined with the sampling frequency f_s and integer lag t', $i = 1, \ldots, M$ indexes the embedding vectors, with $M = N - (m-1)t_d$. These vectors capture short segments of the signal, preserving its local temporal structure.

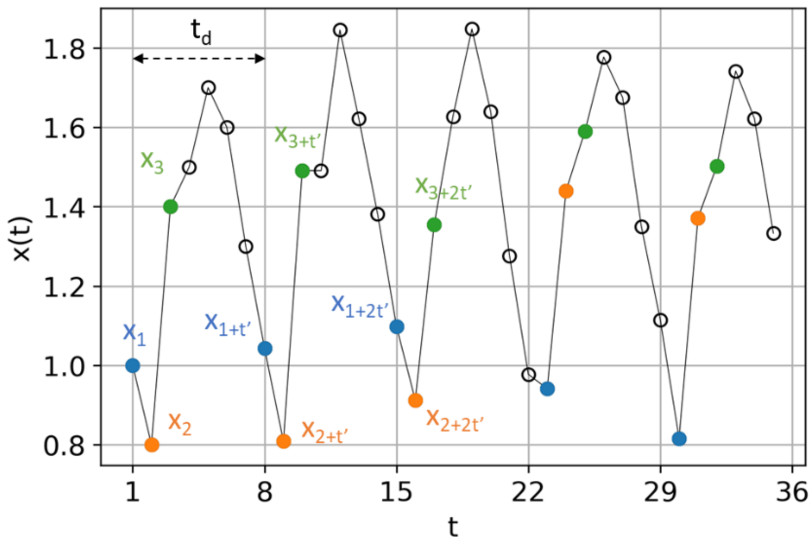

Figure 2.14. Example of signal x(t) with a time delay t_d (adapted from [103]).

Next, for each pair of embedded vectors $u^{(i)}$and $u^{(j)}$, their similarity can be quantified through the Chebyshev distance as:

$$d\left(u^{(i)}, u^{(j)}\right) = \max_{0 \le k < m} \mid x(t_{i+k}) - x\left(t_{j+k}\right) \mid. \tag{2.43}$$

Given a tolerance parameter r (radius of similarity), the fraction of vectors lying within this neighbourhood is:

$$C_i^{(m)}(r) = \frac{1}{M} \sum_{j=1}^{M} \Theta\left(r - d\left(u^{(i)}, u^{(j)}\right)\right). \tag{2.44}$$

Where the $\Theta(z)$ function returns:

- $\Theta(\mathrm{z}) = 0$, if its argument $\mathrm{z} < 0$;
- $\Theta(\mathrm{z}) = 1$, if its argument $\mathrm{z} \geq 0$.

$C_i^{(m)}(r)$ is the fraction of embedded vectors similar to $u^{(i)}$ within a neighbourhood of size r. The next step consists in defining an average logarithmic likelihood; this quantity measures the average regularity of patterns of length m:

$$\Phi^{(m)}(r) = \frac{1}{M}\sum_{i=1}^{M} ln\, C_i^{(m)}(r)\,, \tag{2.45}$$

Finally, ApEn is the difference in regularity when the embedding dimension increases from m to $m+1$:

$$ApEn(m, r, N) = \Phi^{(m)}(r) - \Phi^{(m+1)}(r). \tag{2.46}$$

Interpretation:

- Low ApEn: highly regular, repetitive signal.
- High ApEn: irregular, unpredictable, noisy signal.

Typical choices for the embedding dimension are m = 2 or 3, in fact m must be large enough to capture repeating patterns but also small enough to avoid noise amplification.

Regarding the radius of similarity r, 10 % to 25 % of the signal's standard deviation are sufficient to balance sensitivity and robustness. In addition, ApEn is sensitive to signal length and longer time series improve robustness.

In turbomachinery applications, rising ApEn often indicates transitions to unstable flow regimes, such as surge precursors or roughness-induced turbulence modulation. Conversely, a decreasing ApEn may indicate locking into periodic behaviour like synchronous vibration. Unlike traditional spectral features, ApEn can capture the nonlinear dynamical changes that precede measurable amplitude or frequency deviations.

This diagnostic capability is well-documented in literature. Yan and Gao [58] established ApEn as a diagnostic tool for machine health monitoring , showing that the ApEn of a vibration signal increases as structural defects initiate and grow, since deterioration enriches the signal's frequency content and reduces its regularity. However, the reliability of the metric depends on the choice of embedding dimension, radius of similarity and data length. To address this, Liu et al. [104] conducted a parametric study of approximate, sample and fuzzy entropy in axial compressors, finding that the fuzzy formulation reduces sensitivity to these parameters while preserving substantial stall-warning lead time.

This metric will be extensively employed in the compressor diagnostic studies developed in Ch. 4 to address the detection and characterisation of surge-related phenomena.

2.6 Artificial neural networks

As previously introduced in Sect. 2.3.4, ANNs are representative of a broad class of parametric models able to learn highly nonlinear relationships between inputs and outputs, both supervised and unsupervised. Their flexibility comes from the idea of composing simple computational units arranged into layers, thus allowing the approximation of complex functions even in cases where explicit analytical models are unavailable.

In turbomachinery diagnostics, NNs have been used to model multiple nonlinear phenomena, such as aeroelastic interactions, to detect instabilities, to reproduce high-dimensional surrogate models and extract temporal patterns from pressure or vibration signals. Although NNs can be trained within the same cost function minimisation framework introduced in Sect. 2.2.3, their layered structure comes with a specialised optimisation procedure, such as backpropagation, and requires careful regularisation strategies to prevent overfitting.

In a NN, hyperparameters are all those characteristics that determine the structure of the network, such as the number of neurons, hidden layers, input and output features and activation functions.

This section provides an overview of feedforward NNs, introduces convolutional and temporal architectures relevant to the first compressor study prsented in Ch. 4. Finally, it concludes with practical regularisation methods used to improve generalisation.

2.6.1 Feedforward neural networks

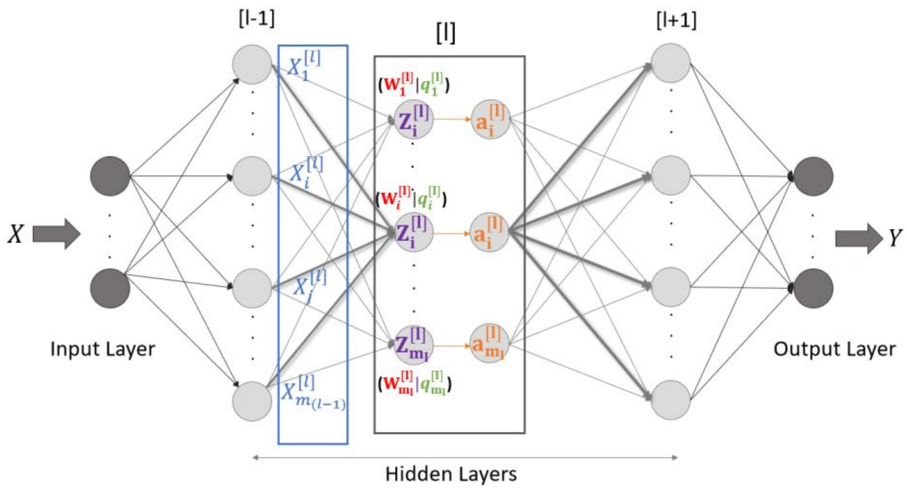

Figure 2.15. General architecture of a NN.

Architecture. A feedforward NN consists of an ordered sequence of layers, beginning with an input layer, followed by one or more hidden layers, and ending with an output layer (see Figure 2.15). Each hidden layer transforms its input through a mapping followed by a nonlinear activation function. Let $a^{[0]} = x \in \mathbb{R}^{m_{in}}$ denote the input vector; for each layer $[l]$, the transformation is

$$Z^{[l]} = W^{[l]}a^{[l-1]} + q^{[l]} \tag{2.47}$$

and

$$a^{[l]} = g^{[l]}(Z^{[l]}), \tag{2.48}$$

where:

- $Z^{[l]}$ is the pre-activation vector,
- $W^{[l]}$ is the weight matrix,
- $a^{[l-1]}$ is the previous layer's output,
- $q^{[l]}$ is the bias vector,
- $g^{[l]}$ is a nonlinear activation function.

The final output $a^{[L]}$ is passed to a loss function $L(a^{[L]}, y)$, as introduced in Sect. 2.2.3.

Activation functions. Nonlinear activations are the key to allow NNs to approximate arbitrary continuous functions. Common choices include (see Figure 2.16):

- Linear:

$$g(Z) = Z. \tag{2.49}$$

- Sigmoid:

$$g(Z) = \frac{1}{1 + \exp(-Z)}, \tag{2.50}$$

historically used in classification.

- Rectified Linear Unit (ReLU):

$$g(Z) = max(0, Z), \tag{2.51}$$

currently, the most widely used in deep learning due to its ability to mitigate vanishing gradients.

- Leaky ReLU:

$$g(Z) = max(Z, \gamma * Z), \tag{2.52}$$

where γ is a fixed scalar.

- Hyperbolic tangent:

$$g(Z) = \frac{exp(Z) - exp(-Z)}{exp(Z) + exp(-Z)}. \tag{2.53}$$

zero-centred and often easier to train.

- Swish:

$$g(Z) = \frac{Z}{1 + exp(-Z)}. \tag{2.54}$$

- Softmax:

$$g(Z_i) = \frac{\exp(Z_i)}{\Sigma_{i=1}^{m_{out}} exp(Z_i)}. \tag{2.55}$$

In multi-class classification problems, NNs typically use the softmax activation at the output layer. Through to the softmax function, the network converts the outputs into a probability distribution over the considered classes, giving a higher probability to the class that is most likely to have occurred. Training is usually performed with the cross-entropy cost (see 2.2.3, Eq. (2.7)), which measures how well the predicted probabilities match the true class labels.

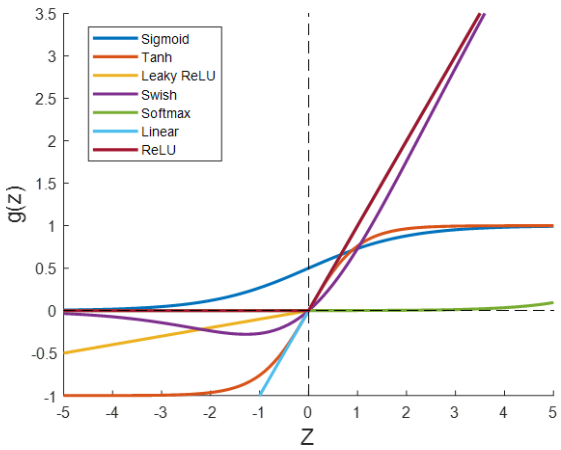

Figure 2.16. Plot of different activation functions.

The optimisation of a feedforward NN, that is, finding the optimal weight that minimises the selected loss function, is based on a propagation of information throughout the net that is bidirectional, namely forward and backward propagation (see Figure 2.17).

Forward propagation. The forward pass computes the output of each layer sequentially according to Eqs. (2.47) and (2.48). The computation cost scales approximately linearly with the number of layers and neurons, making deep architectures feasible for high-dimensional feature extraction.

Backward propagation. Learning consists of minimising the cost function by updating the parameters $W^{[l]}$ and $q^{[l]}$. Backpropagation computes the gradients efficiently by applying the chain rule in reverse.

Given a cost function $L(a^{[L]}, y)$, for each layer $[l]$, an error term can be defined as:

$$\delta^{[l]} = \frac{\partial L}{\partial Z^{[l]}}. \tag{2.56}$$

The gradients of the loss function with respect to the weights and biases are:

$$\frac{\partial L}{\partial W^{[l]}} = \delta^{[l]} (a^{[l-1]})^{\top}, \tag{2.57}$$

$$\frac{\partial L}{\partial q^{[l]}} = \delta^{[l]}. \tag{2.58}$$

The error term is propagated backwards:

$$\delta^{[l-1]} = \left(W^{[l]}\right)^{\top} \delta^{[l]} \odot g'^{[l-1]}\left(Z^{[l-1]}\right), \tag{2.59}$$

where $\odot$ denotes element-wise multiplication. Finally, the weights and biases are updated using an optimisation algorithm like gradient descent:

$$W_{i,j}^{[l]} = W_{i,j}^{[l]} - \alpha \cdot \frac{\partial L}{\partial W_{i,j}^{[l]}}, \tag{2.60}$$

$$q_i^{[l]} = q_i^{[l]} - \alpha \cdot \frac{\partial L}{\partial q_i^{[l]}}. \tag{2.61}$$

Here, α is the learning rate controlling the step size as introduced in Sect. 2.2.3, i denotes the neuron index in layer $[l]$, while j denotes the neuron index in layer $[l-1]$. This training loop constitutes the fundamental learning mechanism of all networks.

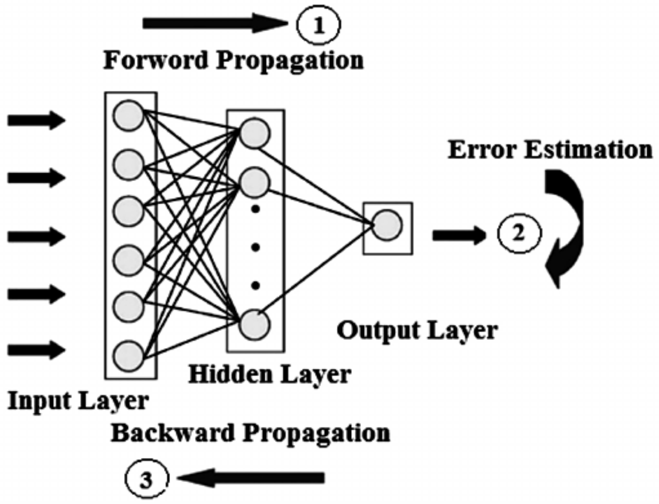

Figure 2.17. Schematic of forward and backward propagation through a NN.

Mini-batch training. In contrast to updating the weights for the entire network using all samples at once (full batch) or for one sample at a time (stochastic), mini batches are the norm for training neural networks via gradient descent.

The training data is divided into smaller batches, which give a gradient estimate. The algorithm processes the first batch and updates the model parameters, defining what is known as a step or iteration, then it proceeds to the next batch and updates the model parameters once again. When all the batches are processed, an epoch is concluded and the algorithm can move on with a new one. This method:

- helps stabilize training compared to purely stochastic update,
- is computationally efficient,
- it introduces a mild noise component that helps escape poor local minima,
- and it is naturally GPU-accelerated.

Mini-batch learning is currently the norm for DL and it is implicitly used for all the optimisation algorithms introduced in the later chapters.

Regularisation. Feedforward NNs possess large numbers of hyperparameters, making them highly expressive but also prone to overfitting. For this reason, regularisation techniques are used to prevent overfitting and improve the ability of the model to generalise. Among those there are:

- L2 regularisation:

$$J' = J + \lambda \sum_{l} \| W^{[l]} \|_2^2, \tag{2.62}$$

where $\lambda > 0$ is a regularisation coefficient. L2 regularisation penalises the magnitude of the weights by adding a quadratic term to the cost function, thus forcing the optimiser to prefer solutions characterised by smaller weights and consequently smoothing effectively the input-output mapping. This is because networks with small weights tend to produce less sensitive and more stable predictions, reducing the effect of measurement noise and operational variability. From the perspective of optimisation, L2 regularisation shrinks weights uniformly and prevents the growth of large coefficients that may cause overfitting.

- L1 regularisation:

$$J' = J + \lambda \sum_{l} \parallel W^{[l]} \parallel_1 \tag{2.63}$$

Unlike L2, L1 regularization encourages sparsity by setting many weights exactly to zero. Sparse networks are characterised by fewer effective connections and can act similarly to feature selectors, which is particularly advantageous when input features exhibit redundancy or strong correlations, conditions commonly encountered in high-dimensional diagnostic datasets.

- Dropout:

$$a_i^{[l]} \leftarrow \begin{cases} 0 & \text{with probability } p, \\ a_i^{[l]} & \text{with probability } 1 - p. \end{cases} \tag{2.64}$$

This is a stochastic regularisation method that randomly disables a fraction p of the neurons of the layer it was assigned to during each training iteration.

This causes the NN to learn meaningful representations that are not dependent upon any particular neuron or on a set of co-adapted neurons. Essentially, the dropout technique trains a very large ensemble of “thinned” networks, where all the networks share the same parameters. The last model will perform like the average of this ensemble and this will make it less prone to overfitting. During testing or inference, the full network is used and activations are scaled appropriately to account for the dropout applied during training. This ensures deterministic predictions and optimal use of all learned features.

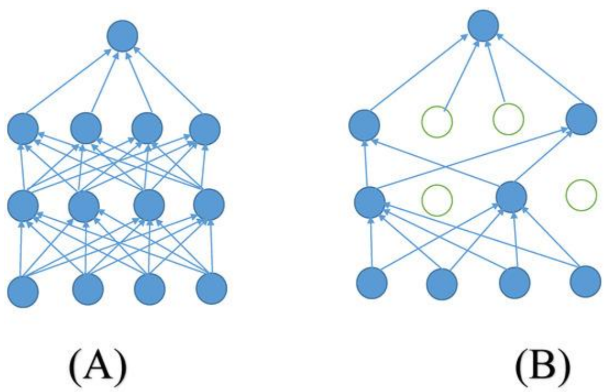

Figure 2.18. (A) NN without dropout. (B) NN with dropout (adapted from [105]).

- Early stopping.

This technique monitors the validation loss during the training process. Optimisation will stop when the loss starts increasing. This helps prevent the network from overfitting the data. However, it allows the model parameters for which the best generalisation is obtained to be retained. It is a very easy and efficient technique, which works best when the data sample is small.

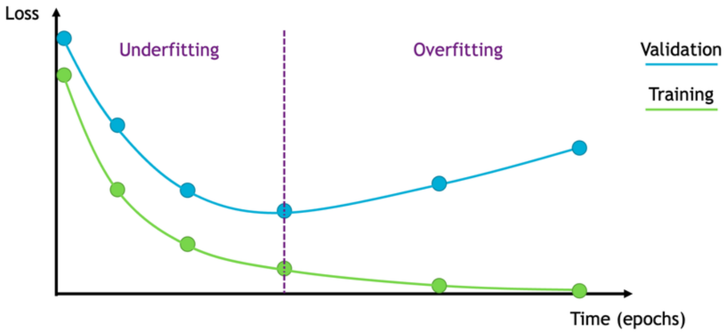

Figure 2.19. Application of early stopping (adapted from [106]),

- Data augmentation

When signal data are limited, augmentation strategies (e.g., noise injection, window slicing, random time shifts) increase training variability, reducing overfitting and improving robustness.

2.7 Convolutional and temporal neural networks

Feedforward NNs operate on fixed-length feature vectors, but many signals in turbomachinery, such as pressure, vibration, flow rate or acceleration measurements, exhibit local patterns, transients, harmonics, wave packets, or intermittency that are not easily captured by dense layers. Deep learning models, such as convolutional architectures, solve this limitation by leveraging local correlations in the input, enabling deep models to learn hierarchical representations of temporal dynamics. This section introduces the main ideas behind convolutional and temporal convolutional networks, as they appear later in the compressor instability study in Ch. 4.

2.7.1 Convolutional neural networks

The origin of Convolutional Neural Networks (CNNs) traces back to image processing, which aims to identify patterns such as edges, texture and shapes, regardless of their location. The characteristic operation of a CNN, known as convolution, involves a small window, known as a kernel or filter, which moves over the input, responding strongly when a meaningful pattern appears. Every filter in a CNN plays the role of a pattern detector, firing when it discerns its learned feature in an input. An architecture of multiple convolutional layers produces progressively more abstract representations, combining simple structures such as edges into textures and ultimately into semantic features (see Figure 2.20). The benefit of CNNs lies in two main properties:

- Local connectivity, the filter interacts only with a neighbourhood of pixels, giving sensitivity to local patterns.
- Weight sharing, the same filter is applied across the image, making the network invariant to translations and greatly reducing the number of parameters.

Although CNNs were first designed for two-dimensional images, the same principles can be applied to one-dimensional signals, for example, pressure or vibration traces used in turbomachinery diagnostics. When the input is a time series, the convolution becomes a sliding filter along the temporal axis, extracting local oscillatory patterns, bursts, modulations or transient disturbances without requiring a fully connected layer for each time step. This significantly enhances efficiency, enabling the identification of recurring patterns in time in a similar way that it would identify shapes in an image.

This shows that CNNs are a powerful tool for inferring a compact representation of pictures or raw measurements that holds physical significance, particularly when characteristic fault patterns are seen as localised events or waveforms that repeat in time.

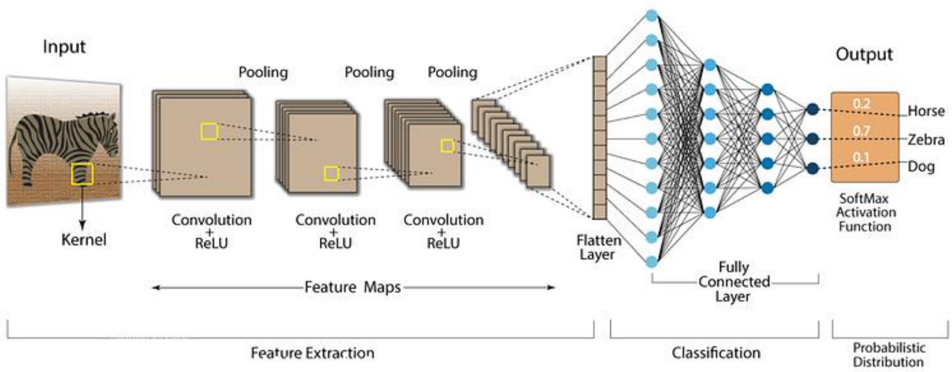

Figure 2.20. Example of a CNN architecture.

2.8 Temporal convolutional networks

Temporal Convolutional Networks (TCNs) further extend this concept to model long-range dependencies for time series data, as first systematically evaluated by Bai et al. (2018) [107]. A traditional CNN looks only at a small window in time, as defined by a filter size, whereas instabilities in turbomachinery, like those related to rotating stall, surge and roughness, might be many sample points long. To solve this, TCNs use dilated convolutions, where a filter skips samples at fixed intervals. This exponentially expands a receptive field without the need for a large number of parameters. An example of a TCN architecture is presented in Fig. 2.21.

This temporal representation has a hierarchical structure, which reflects the multi-resolution characteristics of flow phenomena, where shallow layers are able to capture high-frequency oscillations and deeper layers learn low-frequency, long-term evolutions. Furthermore, TCN architectures can be effectively adapted for time-series forecasting [108], this key characteristic is used in the compressor study of Ch. 4.

In conclusion, TCNs establish a very effective bridge between classical signal processing techniques and deep learning, providing a versatile yet robust platform for capturing the intricate temporal patterns that are so typical of turbomachine CM.

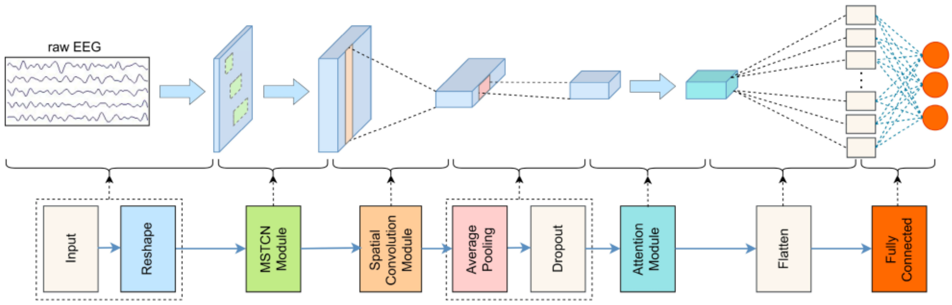

Figure 2.22. Example of a TCN architecture (adapted from [109]).

2.9 Chapter summary

This chapter has outlined the theoretical basis and computational techniques that are at the base of the data-driven diagnostic models developed in this thesis. Starting from a statistical formulation of a learning problem, this chapter has progressed from supervised and unsupervised approaches, through dimensionality-reducing techniques, feature extraction and nonlinear complexity metrics, to neural network models. These elements form a coherent analytical framework, in which physical measurements can be transformed into diagnostic features, mapped through regression or classification models and eventually embedded into DT environments for predictive decision-making.

These methods will be implemented and further specialised in the following chapters, where they are coupled with data from experiments and numerical analysis in order to identify aerodynamic degradation, anomalies and potential instabilities in turbomachines.

3. Computational Fluid Dynamics tools for turbomachinery analysis

3.1 Purpose and scope of the chapter

Computational Fluid Dynamics (CFD) simulations have increasingly proven to be an essential tool within the area of turbomachinery analysis and design, supporting activities that go from an initial aerodynamic assessment to a detailed examination of flow fields. Within the current thesis, CFD simulations have been conducted under different contexts and based on multiple numerical environments. The purpose of this chapter is to provide a coherent overview of the CFD tools adopted during the research activity, clarifying their role, capabilities and limitations in the overall methodological framework of the thesis.

CFD simulations were employed using both steady Reynolds-Averaged Navier-Stokes (RANS) and unsteady RANS (URANS) formulations, selected according to the physical phenomena of interest and the diagnostic objectives of each study. In this chapter, rather than displaying CFD solutions directly, as will be done in subsequent chapters, the objective is more about introducing the selected CFD environments and stating the reasoning for choosing them.

Three different CFD tools are presented; ANSYS CFX is introduced as an industrial-grade solver, with particular focus on its usage within a radial compressor simulation setup. The objective here will be to demonstrate common practices and solver functionalities associated with rotary machines, providing the high-fidelity unsteady datasets used for the diagnostic models in Ch. 4. ANSYS Fluent is presented as a highly flexible CFD solver and will be discussed within the scope of wind turbine simulations, where it served as the numerical engine for deriving the CFD performance maps in Ch. 5. Finally, the open-source code MULTALL is introduced as a dedicated turbomachinery solver within an academic context. Here, a comparative analysis against ANSYS CFX is performed on a three-stage axial compressor, originally a reference case for the MULTALL code, to evaluate solver performance and modeling assumptions.

Although MULTALL was initially selected for its computational efficiency in generating large datasets for NN training, it was ultimately discontinued for the primary scope of this thesis. The evaluation revealed that MULTALL's limitations in handling complex off-design conditions, combined with the unavailability of a real industrial axial compressor for validation, made it less suitable for the diagnostic objectives persued in the following chapters. Without experimental data from a physical machine, MULTALL's internal models could not be verified for degradation phenomena, meaning any extension to CM purposes would rely on unvalidated assumptions.

The discussion developed in this chapter is therefore not intended to establish performance predictions, nor to derive diagnostic indicators. It aims to present the process through which CFD tools were evaluated, understood and ultimately selected for subsequent developments. This approach ensured that the CFD tools used in the following chapters are grounded in a critical and informed use, with a clear awareness of the strengths and limitations of each numerical environment.

3.2 Role of CFD in turbomachinery design, analysis and diagnostics

CFD has made an integral contribution to the field of turbomachinery system development and understanding. It thus becomes essential to identify the various roles CFD has within turbomachinery engineering. All these roles affect modelling and solving.

In the context of aerodynamic design, CFD is primarily used as a predictive tool to evaluate flow behaviour within blade passages, stage diffusers and volutes under ideal or near-ideal conditions. In this stage, numerical analysis aim to guide geometric definitions, assess losses, identify separation regions and eventually optimize performance metrics such as pressure ratio or efficiency. CFD oriented to design often focuses on steady or quasi-steady solutions and may rely on simplified boundary conditions, symmetry or periodicity assumptions to reduce computational cost. The objective is not to perfectly replicate a specific operating history, but rather to investigate trends and sensitivities across a design space. Consequently, the emphasis is on robustness and repeatability instead of strict agreement with experimental data.

When CFD is employed for performance prediction, the focus shifts toward reproducing the behaviour of an existing machine under prescribed operating conditions. In this case, numerical simulations are used to be compared to experimental results, to extend performance maps beyond measured points or investigate flow phenomena that are difficult to capture through testing. The modelling fidelity typically increases compared to design applications, with more detailed representations of geometry, boundary conditions and loss mechanisms.

The use of CFD as a tool for CM and diagnostics introduces additional challenges. There is a requirement for diagnostic applications to be able to relate numerical predictions to measured signals and to interpret deviations in terms of physical degradation or faults. Contrary to design or performance studies, CFD-based diagnostics demands consistency across time, sensitivity to aerodynamic changes and compatibility with experimental data streams. In this context, CFD does not act as a standalone solution, but rather as a reference framework against which measured behaviour can be compared; leading to strict requirements on model validity, boundary condition definition and uncertainty management, since modelling inaccuracies could lead to misleading diagnostic conclusions.

In view of this, the choice between steady and unsteady simulations it is also important. Steady-state CFD remains widely used due to its reduced computational cost and relative robustness, making it suitable for parametric studies, baseline performance assessment and surrogate model generation. On the other hand, many turbomachinery phenomena closely related to diagnostics, such as rotating stall, surge inception, wake interaction and unsteady loading, are inherently time-dependent. Consequenlty, to capture these effects it is required to transition to unsteady simulations, which increase computational demand and data volume. As a result, unsteady CFD is often reserved for targeted investigations, while steady or averaged representations are preferred for integration within monitoring frameworks.

Moreover, access to experimental data plays a crucial role, when detailed measurements and validated geometries are available, CFD can be calibrated and employed more confidently as part of a diagnostic methodology. In the absence of experimental reference data, CFD results must be interpreted with caution and their role is often limited to qualitative analysis or methodological exploration.

Computational cost represents another critical factor itself. High-fidelity simulations can provide improved physical resolution but are rarely compatible with the time constraints offered by monitoring systems. For this reason, CFD-based diagnostic frameworks often rely on reduced-order models, for instance in the form of response surfaces or surrogate representations or also data-driven techniques. In this way, during operation, measured sensor data can be mapped or interpolated onto the CFD-based diagnostic models, allowing fast estimation of performance deviations without the need for direct real-time simulations.

Despite its strengths, CFD alone is not sufficient for diagnostics. Numerical simulations inevitably rely on assumptions regarding turbulence modelling, boundary conditions and geometric fidelity, all of which introduce a certain level of uncertainty. Furthermore, CFD does not naturally account for sensor noise or the operational variability of real machines. It follows that any diagnostic CFD model should be validated with experimental results under faulty conditions or at least be complemented by experimental measurements.

All these factors illustrate why no single CFD solution can be capable of serving as a comprehensive platform for addressing every aspect associated with turbomachinery analysis and monitoring. Different solvers and numerical environments offer complementary strengths and their combined use allows engineers to balance fidelity, flexibility and practicality.

3.3 CFD simulation workflow

Almost all CFD codes have a structured procedure, which can be broken down conceptually into three broad phases: pre-processing, solution and post-processing. Although these phases can be delineated in different ways, depending on the numerical platform, this subdivision represents an accepted framework that has been adopted by most CFD codes conventionally used for turbomachines. A brief description of these steps will be given below, as it will provide a general reference framework for the numerical methods described later within this chapter.

3.3.1 Pre-processing

The first step in a CFD simulation workflow is pre-processing. In this phase, the user selects the geometry of interest, defines the calculation domain for which the simulation will be done and discretises it thorugh a computational mesh into a series of computational nodes. The geometry can be constructed directly with the simulation software or imported from computer-aided design (CAD) drawings. The operations that can be performed in the pre-processing phase are listed here:

- Definition of the geometry.
- Generation of the mesh.
- Definition of the various parts of the mesh.
- Selection of the type of fluid.
- Selection of the physical phenomenon of interest.
- Definition of physical parameters, turbulence models, etc.
- Selection and definition of boundary conditions.
- Selection of calculation settings and parameters.

The domain solutions are obtained based on the predefined mesh nodes; therefore, a larger number of grid cells ensures greater accuracy in solving the problem, but at the same time, it comes with greater computational costs.

Generally, most of the industrial CFD codes, such as ANSYS, can automatically generate optimal grids for geometries that do not present significant variations in size. When the software mesher cannot apply its algorithm, the process is carried out manually, requiring a large amount of time and considerable experience.

3.3.2 Solver

The CFD solver has the function of solving and bringing to convergence the equations that describe the flow field for the geometry defined in the pre-processing phase. The following methods can be used:

- Finite differences.
- Finite elements.
- Finite volumes.

The approach most used by CFD codes is the finite volume method. The main stages of the scheme are:

- Application and integration of the partial differential equations of motion (continuity, momentum and energy) to the finite volumes that discretise the entire domain.

- Conversion of the differential equations into a system of algebraic equations with the introduction of approximations.
- Iterative solution of the algebraic equations.

The iterative approach is required by the non-linear nature of the equations, with the solution converging step by step towards a final one. The error between the final solution and the exact solution, which is often unknown, depends on numerous factors, such as the size and shape of the domain volumes and the final residuals. Complex physical phenomena such as combustion and turbulence are often modelled using empirical relationships, the approximation introduced by the models additionally contributes to the difference between the CFD solution and the exact solution.

3.3.3 Post-processing

Most modern CFD packages offer graphical interfaces for displaying the solution results. These are divided into:

- Geometry and mesh visualisation.
- Flow vectors visualisation.
- Streamlines visualisation.
- Graphical field visualisation of quantities, such as pressure and temperature.
- Image export.
- Animations.

The introduction of these packages allows users to visualise the results and facilitates their interpretation by quickly identifying the areas that require particular attention.

3.4 ANSYS CFX

The CFX commercial programme from the ANSYS suite was used to perform the calculations at the base of the radial compressor studies presented in Ch. 4. This powerful and intuitive code for fluid dynamics simulation is particularly appreciated in the field of turbomachinery. The programme consists of four main modules:

- TurboGrid is the programme for importing geometry and constructing the mesh. It was developed for the turbomachinery sector and is therefore very suitable for mesh generation. The mesh is implemented in a very guided manner, while still allowing the operator to intervene on the main parameters. It is also possible to construct the geometry with external CAD softwares, such as Solidworks, Solid Edge, AutoCAD 3D, PTC Creo and then import it into TurboGrid.
- CFX-Pre, where the mesh is loaded and prepared for simulation. Meshes can be imported not only from Turbogrid but also from other softwares such as ICEM. Within the program, the various fluid domains are defined and specific boundary conditions are specified for each one.

- CFX-Solver Manager is the programme that performs the calculations. During the calculation, there is the possibility to monitor certain variables defined by the user in the pre-processing stage and analyse the trends of the residuals. Depending on the type of problem being analysed, both steady-state or unsteady-state calculations can performed.
- CFX-Post is the programme that allows postprocessing the fluid domain and analyse the simulation results by displaying them with various graphical solutions such as diagrams, contours, vectors, streamlines, etc. In relation to unsteady simulations, it is also possible to create videos showing how the flow behaves in time.

Figure 3.1 summarises workflow scheme of ANSYS CFX in a logical diagram.

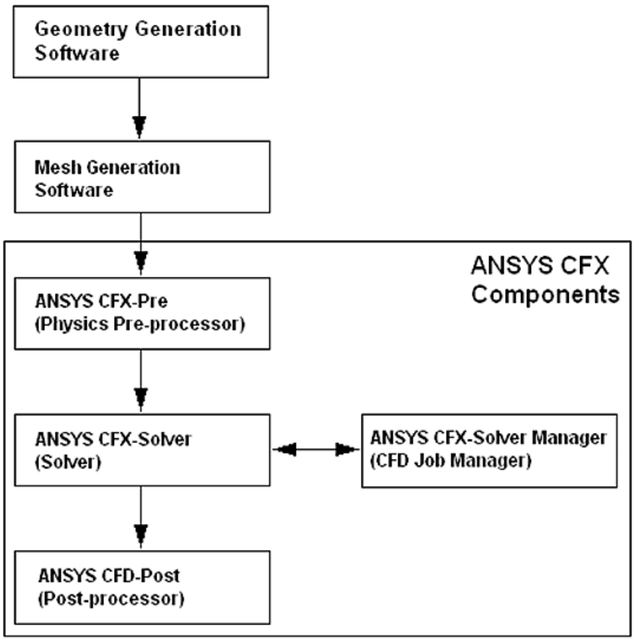

Figure 3.1. Workflow scheme of Ansys CFX.

3.4.1 Radial compressor CFD modelling as methodological reference

The radial compressor studies presented in the following Ch. 4 are mainly based on a database generated by a previous, extensive numerical investigation into the unsteady aerodynamic behaviour of a two-stage radial compressor for refrigerant applications. This foundational work, detailed by Marsano et al. in [110], employed ANSYS CFX to perform URANS simulations for three different operating points in the range from the best efficiency point (BEP) to the deep surge one (SU). The investigation was based on a consolidated modelling framework, which included validation against experimental results and ensured that the numerical outcomes were consistent with physical flow phenomena.

For the present thesis, the same CFX framework was used to run a new unsteady simulation for one additional operating point, referred to as "New Point (NP)" and located between the BEP and the near-surge one (NS).

It is mentioned that the primary focus of the following studies of Ch. 4 was on deriving diagnostic models critical for surge detection across the analysed operating range of the selected machine. These models were based on data obtained from the time-resolved pressure signals extracted from the four aforementioned operating points. The static pressure time series were monitored at key locations within the compressor flow path. This positioning highlighted how CFD can be effectively used to emulate virtual sensors and enabled easy access to unsteady flow quantities (pressure fluctuations associated with incipient surge and flow instability) that are difficult or impractical to measure experimentally.

Figure 3.2 illustrates the radial compressor geometry adopted in the reference CFD model [110], along with multiple reference sections. The monitors used for signal extraction were defined in the following sections:

- First stage: 12 at section 1 (Inducer) and 18 at section 3 (Diffuser).
- Second stage: 12 at section 6 (Inducer) and 22 at section 8 (Diffuser).

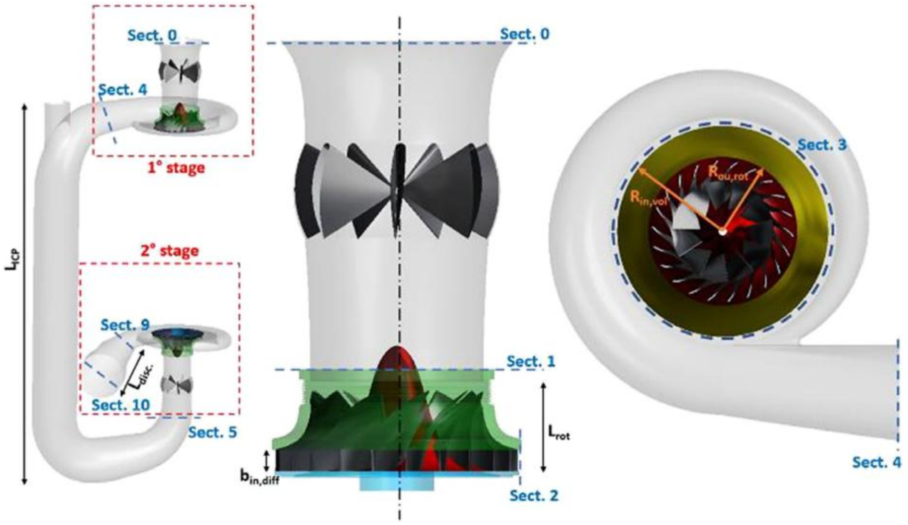

Figure 3.2. Schematic of the reference two-stage centrifugal compressor with reference control sections (adapted from [110]).

As previously introduced in the reference work [110], CFD simulations were carried out using two computational models, namely *Simple* and *Fully3D*.

In the *Simple* configuration, a reduced sector of the machine was initially modelled through steady-state RANS simulations, employing circumferential periodicity to provide a first solution of the aerodynamic behaviour of the compressor (see Figure 3.3). In particular, a single sector was used to model the flow path from the adduction duct to the impeller exit of the first stage. The impeller of the second stage was also modelled with one sector, while

the remaining elements were simulated in their full geometry. Structured meshes were generated for the bladed components using TurboGrid, whereas unstructured meshes were adopted for the other domains. Care was dedicated to near-wall grid refinement, ensuring dimensionless wall distances compatible with low-Reynolds turbulence modelling and to model the fluid as a real gas. A mesh sensitivity analysis was also performed to verify that predicted global performance variations remained within acceptable limits.
The *Fully3D* model represented the complete annular geometry of the two-stage compressor and was introduced to capture full circumferential flow interactions and time-dependent phenomena. Although secondary cavities were neglected in this configuration to limit the overall grid size, the model retained the full three-dimensional blade representation and allowed unsteady flow interactions between the different components to be resolved.

Compared to the previous configuration, this approach required significantly higher computational resources but enabled a physically consistent simulation of flow instabilities. The *Fully3D* configuration was first simulated in a steady form and then with an unsteady setup.

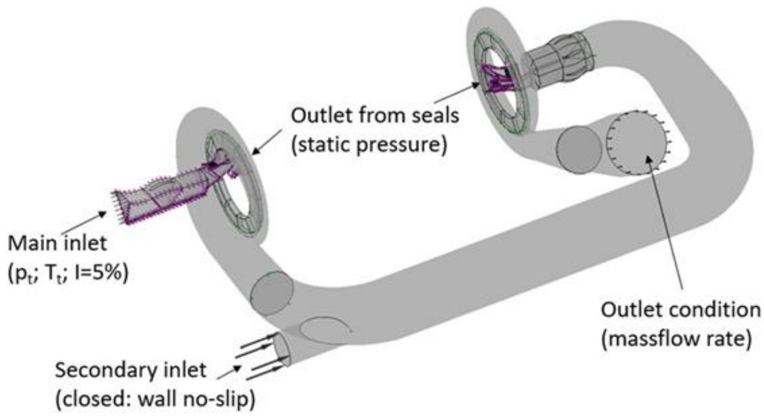

Figure 3.3. Geometry used in the Simple model of [110].

Figure 3.4 schematically compares the results from the different modelling approaches employed in [110], demonstrating their different levels of fidelity (the plotted results are averaged over one impeller rotation). The performance curves presented here show the computed total-to-static pressure ratio versus mass flow rate, defined as:

$$PR_{ts} = \frac{p_{out}}{p_{t,in}} \tag{3.1}$$

where p_{out} and $p_{t,in}$ are respectively the outlet static and inlet total pressure of the stage. It is possible to see that the *Simple* model (with implemented rough walls) already achieved a good agreement with the experimental data; however, the most accurate results were obtained with the *Fully3D* geometry in its unsteady configuration.

While simplified models are useful for preliminary assessments and parametric studies, 3D unsteady simulations are required to capture the temporal evolution of flow instabilities. For this reason, the unsteady Fully3D model was used to investigate the flow mechanisms that triggered instability.

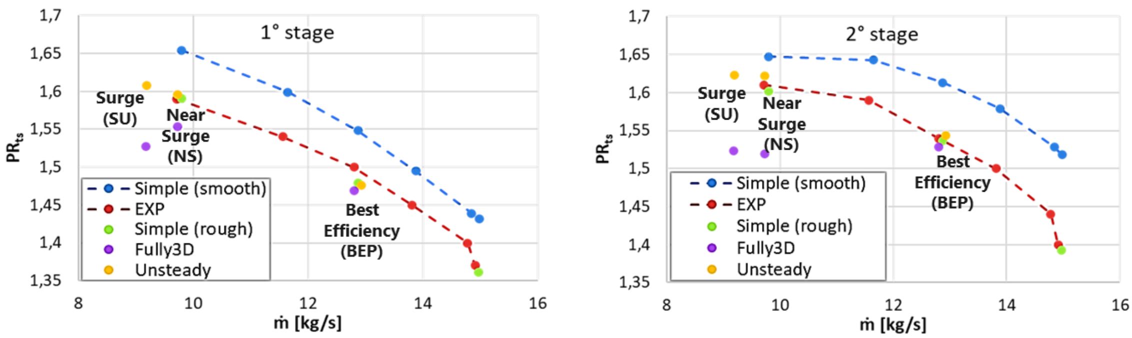

Figure 3.4. Performance curves of the different CFD models compared to the experimental data (adapted from [110]).

Finally, Figure 3.5 presents an example of the normalised static pressure signals, extracted from the *Fully3D* CFD simulations for the best efficiency and near surge conditions (BEP, NS) at 10 % of the diffuser exit of stage two. The monitored static pressure was normalised as:

$$CP = \frac{p - p_\infty}{p_{t,\infty} - p_\infty} \tag{3.2}$$

where p_∞ and $p_{t,\infty}$ are respectively the reference static and total pressure of the investigated component, that is, the inducer or the diffuser. Focusing on probe location, the figure also highlights the role of CFD in helping to understand which monitors are more sensitive to flow instabilities and therefore useful for developing diagnostic models. In other words, CFD does not act as a direct diagnostic tool, but rather as a means to inform the structure and interpretation of diagnostic datasets.

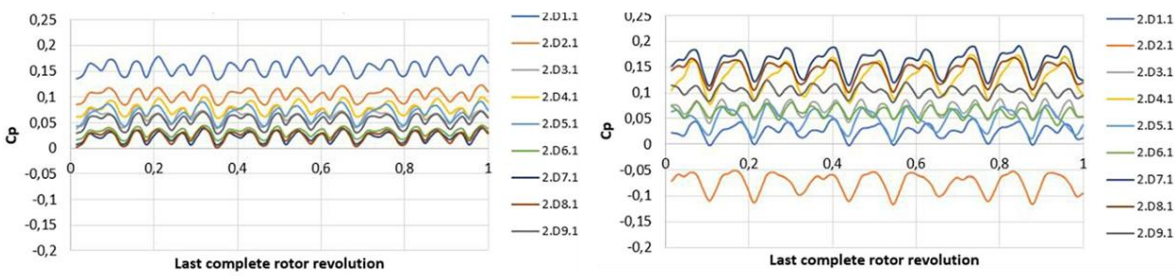

Figure 3.5. CP monitors for the BEP (right) and NS (left) conditions located at 10 % of the diffuser exit of stage two (adapted from [110]).

In summary, for the present thesis, the role of the ANSYS CFX modelling framework was not to provide new insights into fundamental radial compressor aerodynamic performance. Instead, its primary purpose was to represent a tool for establishing a validated numerical reference essential for the subsequent signal analysis. This reference, defined by the monitoring strategy illustrated with Figure 3.5, provided the pressure-signal database that will lead to the development of a diagnostic model for detecting and predicting incipient compressor instability (see Ch. 4).

3.5 ANSYS Fluent

ANSYS Fluent is a general-purpose CFD solver widely employed for the analysis of both internal and external flow problems. Unlike ANSYS CFX, which was specifically tailored to turbomachinery applications, Fluent was designed to accommodate a broader range of geometries, flow regimes and modelling requirements, which make it particularly suitable for applications involving complex external aerodynamics and large computational domains.

With relevance to wind turbine analysis, the aerodynamic problem is significantly different from classical compressor or gas turbine simulations. In this case, the flow field develops in an open domain, is strongly influenced by inflow conditions and is characterised by prominent three-dimensional effects and wake development. Fluent represents a great tool to study such configurations within a unified numerical framework, while maintaining compatibility with both structured and unstructured meshing strategies and also offering a wide variety of numerical formulations.

A further relevant aspect of Fluent is its easy integration within ANSYS Workbench. ANSYS Workbench enables a structured workflow from geometry preparation to mesh generation and solver execution. Such an environment is very helpful when designing a systematic study where geometric or operating parameters may be varied while preserving a consistent simulation setup.

In this context, ANSYS Fluent was adopted as the CFD environment for the wind turbine studies presented in Ch. 5. The present section focuses on the numerical workflow and software framework employed, while detailed modelling assumptions and simulation results are intentionally presented in the following chapter.

3.5.1 CFD workflow based on ANSYS Workbench

As introduced in Sect. 3.3, a CFD simulation can be conceptually divided into three main steps, namely pre-processing, solution and post-processing. In the present section, these phases were implemented and managed within the ANSYS Workbench environment, adopted as the integration platform for the wind turbine simulations carried out using ANSYS Fluent.

Rather than redefining the individual CFD phases, this section focuses on the role of ANSYS Workbench as a workflow-management layer, responsible for connecting geometry preparation, mesh generation, solver execution and post-processing within a unified and consistent numerical framework.

One of the main advantages of using ANSYS Workbench lies in its ability to enforce consistency across the various steps of the numerical procedure. Changes introduced at the geometry level can be automatically propagated to downstream components, such as the meshing and solver stages, reducing the risk of inconsistencies and improving reproducibility. This aspect is particularly relevant when dealing with complex geometries or when multiple simulation cases must be generated following the same modelling logic.

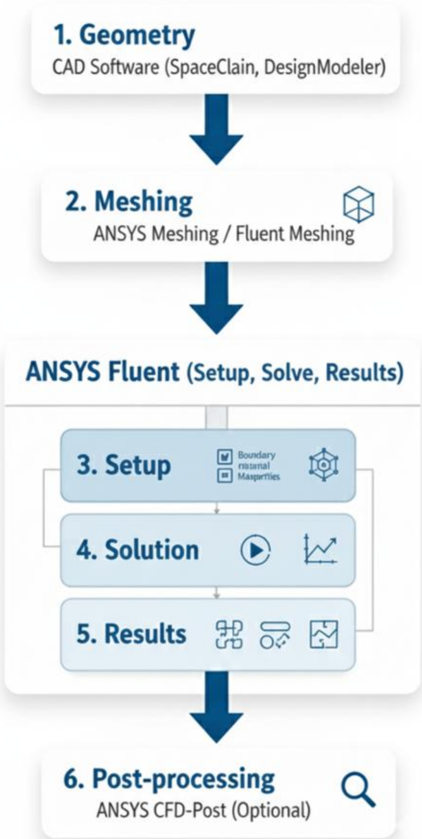

Figure 3.6. Workflow of ANSYS Workbench for a CFD simulation with ANSYS Fluent.

Figure 3.6 presents the ANSYS Workbench framework, where the CFD workflow is typically organised into a sequence of interconnected modules. Geometry definition and modification are handled in dedicated preprocessing tools, after which the computational domain is discretised through the meshing module. The generated mesh is then passed to the Fluent solver, where the governing equations are solved and the resulting flow field is made available for visualisation and data extraction during post-processing.

3.5.2 Geometry preparation in ANSYS SpaceClaim

The geometry preparation phase represents an important step in the CFD workflow for wind turbine simulations, as it directly influences mesh quality, numerical stability and the overall robustness of the solution. For the wind turbine studies presented in Ch. 5, geometry handling and preprocessing were performed using ANSYS SpaceClaim, which provides a flexible and efficient environment for preparing CAD models for CFD applications within the ANSYS Workbench framework.

The starting point of the preprocessing stage consists of importing the wind turbine geometry into SpaceClaim and verifying its consistency for numerical analysis. Next, a key objective of the geometry preparation stage is to define the fluid domain surrounding the wind turbine. For external aerodynamic simulations, this typically consists of defining an open computational domain that extends sufficiently upstream, downstream and laterally from the rotor to minimise boundary effects on the flow solution. Further details are provided in Sects. 5.3.6 and 5.5.2 of Ch. 5.

Within SpaceClaim, this was achieved by creating a surrounding volume around the turbine geometry and performing a Boolean subtraction with the turbine CAD model to obtain a clean fluid region suitable for meshing. Figure 3.7 shows, as an example, a detail of the subtraction process around the selected wind turbine.

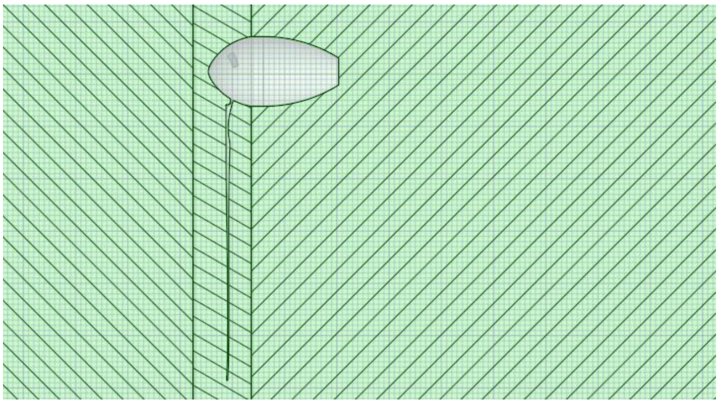

Figure 3.7. SpaceClaim section of the fluid domain showing the Boolean subtraction result.

Next, depending on the type of analysed aerodynamic fault, care was used to define proper inlet and outlet surfaces, lateral boundaries and interfaces associated with the rotating components (further details are provided in Ch. 5). To support a structured and repeatable workflow for simulating multiple operating points, these regions were organised using named selections, which allow boundary conditions and solver settings to be consistently assigned in subsequent stages without ambiguity.

Another important aspect of the preprocessing stage concerns the definition of a proper reference system. For wind turbine simulations, the correct alignment of the rotor axis and the inflow direction is essential to ensure consistency across different operating scenarios and to define a reference for the following studies.

Overall, SpaceClaim preparation led to a robust geometry with defined boundaries, ensuring consistent CFD simulations. The following meshing strategy is discussed in the next section.

3.5.3 Meshing strategy in ANSYS Workbench and polyhedral conversion

The meshing stage represents a critical phase of the CFD workflow. In this section, the meshing strategy is discussed from a numerical and methodological perspective, before fine tuning and case-specific refinements are addressed in the following chapters. Starting from the fluid domain prepared in ANSYS SpaceClaim, an initial surface and volume mesh was generated using unstructured elements in the mesher of ANSYS Workbench. In this way, complex external geometries such as the wind turbine rotor, nacelle and the surrounding domains can be efficiently discretised. The surface mesh resolution was defined to adequately represent blade geometry and domain boundaries; the volume mesh was generated to grant smooth cell-size transitions, with an acceptable element quality throughout the computational domain.

In the regions close to the rotor and in the wake area, a higher mesh density was assigned due to the expected presence of stronger flow gradients and interaction effects. On the other hand, progressively coarser mesh resolutions were employed farther away from the turbine where flow variations are less pronounced. This strategy improves the computational cost and allows capturing the main aerodynamic features without an unnecessary increase in the total number of cells.

Boundary layer representation was managed through the introduction of near-wall prism layers along the blade and nacelle surfaces. The mesh was constructed to offer a consistent near-wall discretisation scheme suited for boundary layer modelling in Fluent. The detailed wall treatment and turbulence modelling assumptions are discussed in the following Ch. 5.

After the generation of the first unstructured mesh, a polyhedral mesh conversion was applied. In the Workbench workflow, this conversion is possible only in the ANSYS Fluent software. Moving from a tetrahedral to a polyhedral mesh presents several numerical benefits; polyhedral meshes typically reduce the overall number of cells required to discretise a given domain, while increasing the number of faces per cell. This results in improved numerical stability and more accurate gradient reconstruction, particularly in the complex flow regions developing around the wind turbine. In addition, polyhedral meshes are known to enhance solver robustness and convergence behaviour, which is especially beneficial when conducting multiple simulations within a systematic numerical campaign.

Figure 3.8 illustrates a comparison between the initial tetrahedral mesh and the corresponding polyhedral mesh obtained after conversion. A systematic change can be noted in the cell topology and overall mesh structure. The adoption of polyhedral meshes represents a methodological choice which aims to improve numerical efficiency and stability while maintaining fidelity for wind turbine CFD simulations.

In essence, the meshing strategy introduced in this section laid a numerically robust foundation for the subsequent solver step. The combination of the advantages of unstructured meshing with those of the polyhedral discretisation method makes the generated meshes appropriate for the CFD investigation of wind turbine designs.

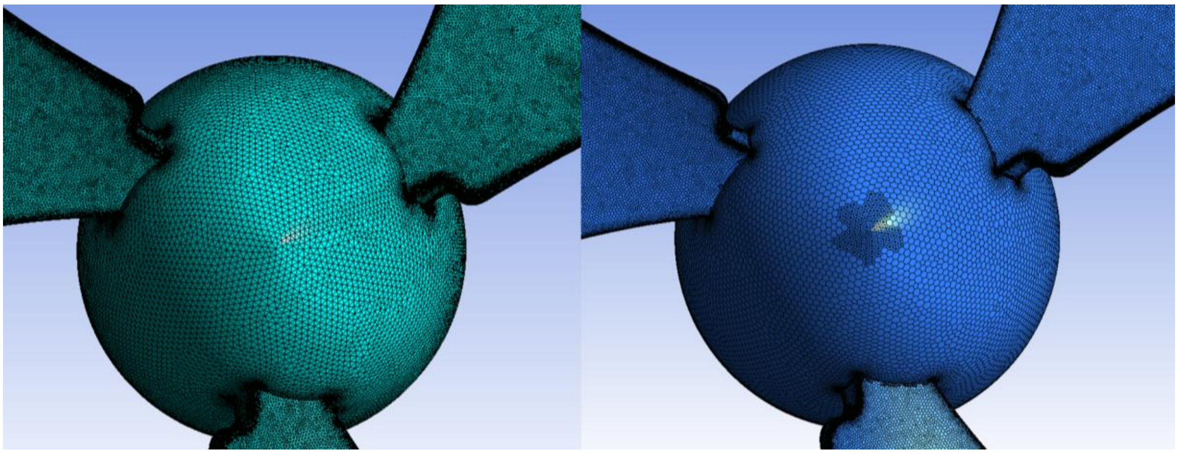

Figure 3.8. Conversion from tetrahedral mesh (right) to polyhedral mesh (left).

3.5.4 ANSYS Fluent solver environment and solution process

ANSYS Fluent provides a comprehensive solver environment for the numerical solution of the governing equations of fluid flow, structured around a modular solution process. Once the computational mesh is imported into the solver, the simulation setup is organised through a sequence of steps that include turbulence model selection, solution control, convergence monitoring and data storage for post-processing. This structure allows the user to manage complex simulations in a clear and replicable manner.

A meaningful possibility of the Fluent solver in the context of wind turbine simulations is its ability to easily handle rotating regions within external flow configurations. One of the key methods used in the modelling of the rotating machinery in a steady configuration is that of the Multiple Reference Frame (MRF) method [111]. In the MRF method, the set of governing equations is solved for multiple reference frames for the rotating and stationary parts of the region of interest. This is achieved by introducing additional source terms in the momentum equations that account for Coriolis and centrifugal effects in the rotating frame, while the stationary frame maintains the usual inertial form.

During the solution process, Fluent allows continuous monitoring of convergence through both residual-based criteria and user-defined integral quantities. This monitoring capability helps to assess numerical stability and solution consistency throughout the simulation. In addition to convergence indicators, the solver also enables the definition of monitors associated with flow variables of interest, which can be recorded during the solution process and later analysed during post-processing.

Another important aspect of the Fluent solver environment is the accessibility of simulation data for post-processing and further analysis. Flow field variables, surface quantities and integral outputs can be readily exported in standard formats, supporting their use in external processing tools or other numerical workflows.

In summary, ANSYS Fluent represents a tool that offers a structured and robust numerical solution process for external aerodynamic simulations of wind turbines. Detailed solver settings and modelling assumptions, intentionally omitted here, are presented in the following Ch. 5, where the wind turbine simulations are discussed in detail.

3.5.5 Concluding remarks on Fluent as a CFD tool

Within this section, the software tool ANSYS Fluent has been presented as a general purpose tool for CFD simulations, which can also be applied for the numerical simulation of the aerodynamic behaviour of wind turbines. Its flexibility in handling complex external flow domains, together with rotating components makes it well suited for applications that extend beyond the assumptions of classical turbomachinery solvers.

The inclusion of Fluent in the environment of ANSYS Workbench allows for a seamless process from geometry preparation, mesh generation, through executing solvers and finally post-processing. This helps in creating a structured framework for setting up CFD simulations that are consistent and reliable, especially in the case of complex geometries.

3.6 MULTALL: design-oriented CFD for turbomachines

3.6.1 Overview of the MULTALL system and design philosophy

MULTALL is a CFD system specifically developed for the preliminary aerodynamic design and steady-state analysis of turbomachinery by Prof. J. Denton [112]. Unlike other general-purpose or industrial RANS solvers, MULTALL was developed as an environment that allowed rapid design exploration of multi-stage compressors and turbines, rather than providing a high-fidelity validation of a final industrial configuration.

The MULTALL system was developed to allow a direct transition from preliminary design concepts to three-dimensional flow analysis. The core idea of this approach was that for many design-oriented studies, a fast and reasonably accurate three-dimensional solver is often preferred over a highly detailed but computationally expensive simulation. Consequently, MULTALL was designed for iterative analysis, where several design alternatives could be analysed for limited computing time, thus allowing designers to examine a large portion of the design space for possible improvements.

This key aspect is achieved through numerical models and assumptions selected to balance physical realism and computational efficiency. In addition, thanks to the open structure and flexibility of MULTALL, the user can load a predefined machine geometry and move directly to three-dimensional flow simulations. By offering fast, physically informed assessments of multi-stage aerodynamics, MULTALL can serve as an effective tool for generating large turbomachinery databases through parallel simulations, leveraging both its high calculation speed and low memory requirements.

3.6.2 MULTALL workflow: MEANGEN - STAGEN - MULTALL

The MULTALL system is structured around a workflow composed of three main modules: MEANGEN, STAGEN and MULTALL. Each module focuses on a specific level of the turbomachinery design process, enabling a progressive transition from preliminary design parameters to three-dimensional multi-stage flow simulations within a single and linked framework.

MEANGEN. This represents the first stage of the workflow, responsible for the definition of the mean-line design of the turbomachine. In this module, the overall thermodynamic and aerodynamic parameters of the machine stages are specified and a preliminary description of the blade-row characteristics is generated. In particular, MEANGEN outputs the initial estimates of the flow angles, velocity triangles and key geometric quantities that define the base operating characteristics of the machine. This information represents the starting point for the subsequent stages of the design process.

STAGEN. The output file of MEANGEN is then passed to the second module STAGEN, which performs the stacking of individual blade rows and the definition of the coordinates of the meridional flow channel. The axial and radial distributions of the stages are also established and the overall 3D geometry of the machine is constructed in a simplified and consistent manner. In addition, this module also allows the user to upload a known geometry, instead of relying on one preliminarily generated one with MEANGEN. Therefore, STAGEN acts as an intermediate link between the mean-line design and a full three-dimensional analysis.

MULTALL. The third module, which performs a three-dimensional RANS analysis of the complete multi-stage turbomachine, using the geometry and flow information generated by the output file of STAGEN. MULTALL solves the governing equations of the flow over the entire machine, allowing to capture the interaction between consecutive blade rows in a steady-state framework.

One of the main advantages of the workflow of the MEANGEN-STAGEN-MULTALL workflow is based on the automatic transfer of data from one module to another. Keeping a consistent data structure throughout the design flow reduces the need for extensive manual intervention, thus shortening the design iteration cycle and allowing multiple design variants to be analysed efficiently and consistently.

Figure 3.9 schematically shows the MULTALL workflow, highlighting the information flow from MEANGEN to STAGEN and finally to the MULTALL solver.

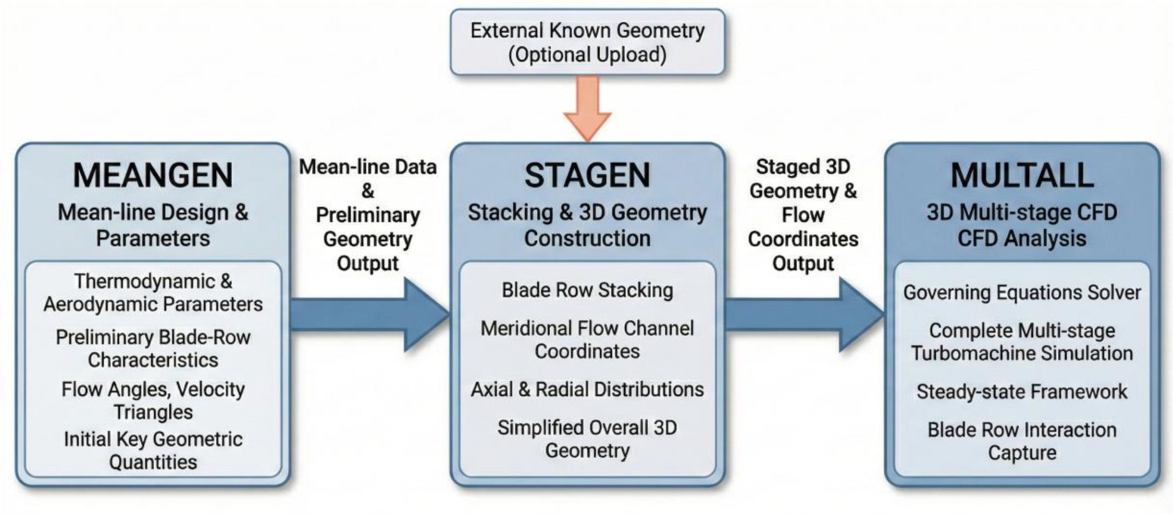

Figure 3.9. Schematic of the MULTALL workflow (based on [112]).

3.6.3 Grid topology and interface modelling

The spatial discretisation in MULTALL is based on structured, multi-block H-type grids specifically developed for turbomachinery applications. This grid topology allows a consistent representation of blade-to-blade, spanwise and streamwise directions while maintaining compatibility with the mixing plane treatment between blade rows.

Each passage is discretised using a structured grid with grid lines lying in the direction of the main flow. While discretisation in the blade-to-blade direction captures the variation of flow in each passage, discretisation in the spanwise direction is necessary due to radial gradients caused by secondary flow and end wall effects. Discretisation in the streamwise direction lies along the meridional channel from inlet to outlet and local grid refinement is commonly employed near leading edges.

A key aspect of the MULTALL methodology is the multigrid approach, which allows individual blade rows to be discretised independently and then assembled into a complete multi-stage configuration. This modular approach makes grid generation easier and enables efficient modification of individual stages without requiring a full regeneration of the entire computational mesh.

The interaction between adjacent blade rows in relative motion is modelled using mixing plane interfaces. At each mixing plane, flow quantities are circumferentially averaged before being passed to the downstream blade row. This approach represents a common strategy to model steady-state interaction between rotating and stationary components while filtering out unsteady blade-passing effects. The mixing plane model conserves mass, momentum and energy in a pitchwise-averaged sense and also introduces an entropy increase to take into account the physical mixing process.

The use of mixing planes significantly reduces the computational cost of the simulation compared to fully unsteady simulations. While this approach does not capture time-resolved rotor-stator interactions, it is well suited for design iterations where the primary interest lies in defining the mean flow behaviour of the machine together.

For these reasons, the structured H-grid topology combined with mixing-plane interfaces adopted in MULTALL represents a robust and efficient numerical strategy for multi-stage turbomachinery simulations, particularly in the context of rapid design exploration and parametric studies. Figure 3.10 presents, as an example, a blade-to-blade view of the grid on a streamwise surface of an axial turbine.

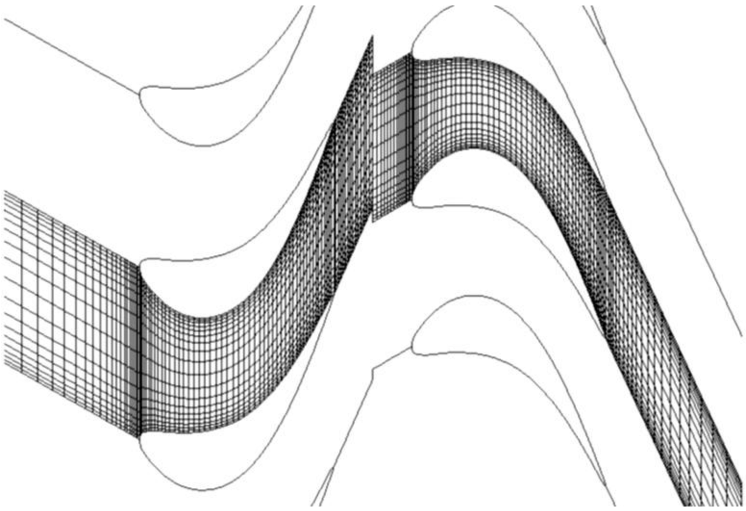

Figure 3.10. Example of a blade-to-blade view of a grid on a streamwise surface of an axial turbine (based on [3]).

3.6.4 Physical modelling and numerical assumptions in MULTALL

The MULTALL solver is based on a modelling strategy that specifically targets the requirements of turbomachinery design studies, where the main objective is the rapid and physically consistent estimation of performance trends, instead of a detailed resolution of all turbulent flow structures. For this reason, the numerical formulation of MULTALL is calibrated to represent the dominant flow mechanisms governing losses and stage interaction and it deliberately simplifies aspects that would otherwise lead to high computational cost.

Viscous effects within the machine are described through turbulence closure models that provide the eddy viscosity field required by the 3D Navier-Stokes solution. The dominant viscous loss sources captured by this approach include boundary layer dissipation and wake mixing losses. These models provide algebraic or differential closures for the turbulent stresses, allowing the viscous losses to emerge naturally from the Navier-Stokes solution. The user can select between the mixing length model, available in two versions (with the second one being an updated and improved version of the former), and the Spalart-Allmaras turbulence model. The latter, as implemented within MULTALL, has been observed in practice to be less robust for steady multistage simulations compared to the most recent version of the mixing length model, which was therefore often preferred. Both solutions are mainly focused on loss prediction rather than on the detailed reconstruction of turbulent flow structures, consequently leading to loss underestimation when dealing with highly separated flows.

Another important aspect of the physical modelling implemented in MULTALL concerns the treatment of tip clearance and leakage flows. In design-oriented simulations, explicitly meshing clearance gaps would lead to a significant increase in grid complexity and computational cost. Therefore, MULTALL represents clearance effects through simplified modelling strategies, which account for the associated mass flow leakage and energy losses without resolving the detailed flow within the clearance region. In particular, tip leakage for plain tip clearances is calculated using the pinched tip model, in which the blade is thinned towards the tip and across the tip gap the blade thickness is set to zero (see Figure 3.11). Although with a certain degree of simplification, this method allows to include the influence of clearance on the stage performance and on the overall machine behaviour in a consistent and computationally efficient manner.

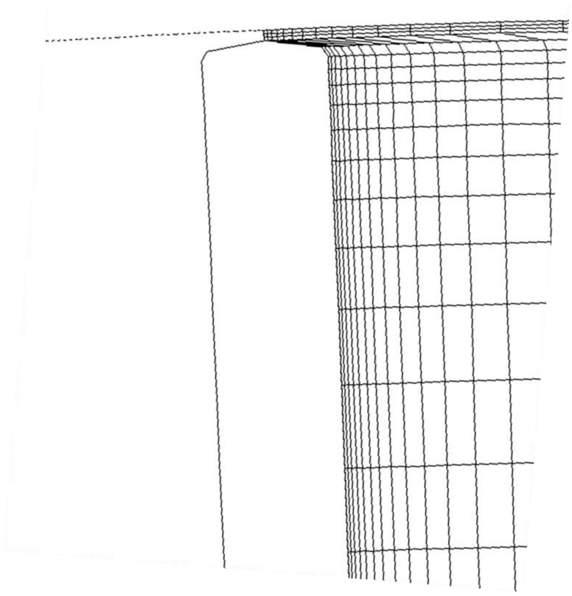

Figure 3.11. Example of a tip clearance mesh for an axial compressor blade on MULTALL.

A further key element covered in MULTALL is the trailing-edge modelling. When discretising the mesh, problems may arise for blades with blunt trailing edges. With a very fine grid around a blunt trailing edge, a local area of low pressure may be found on the pressure side of the blade, because the flow does not separate early enough and consequently follows a high curved surface. This effect, in return, provides an unrealistic load on the blade reaching negative values at the trailing edge.

To avoid this problem, MULTALL offers the possibility to use the cusp fitted at the trailing edge. This option sets a triangular area that extends the trailing edge of the blade, bypassing the excessive curvature of the streamlines (see Figure 3.12Figure 3.12). The cusp is modelled to carry no tangential force and so does not contribute to the lift on the blade or the work done by it. However, it does exert a meridional force on the blade, which is exactly balanced by the force it exerts on the flow. Finally, when dealing with blades with very thick trailing edges, which make it difficult to implement a cusp, the code also offers to apply a body force in the region immediately downstream of the trailing edge as another possible solution.

In summary, the physical modelling strategy adopted in MULTALL represents a deliberate compromise between accuracy and efficiency. By prioritising the representation of dominant loss mechanisms and stage interactions over detailed turbulence resolution, the solver enables fast and robust simulations of multi-stage turbomachinery configurations. This modelling strategy is fully consistent with the design-oriented objectives for which MULTALL was developed and explains its widespread use in academic research and preliminary design studies.

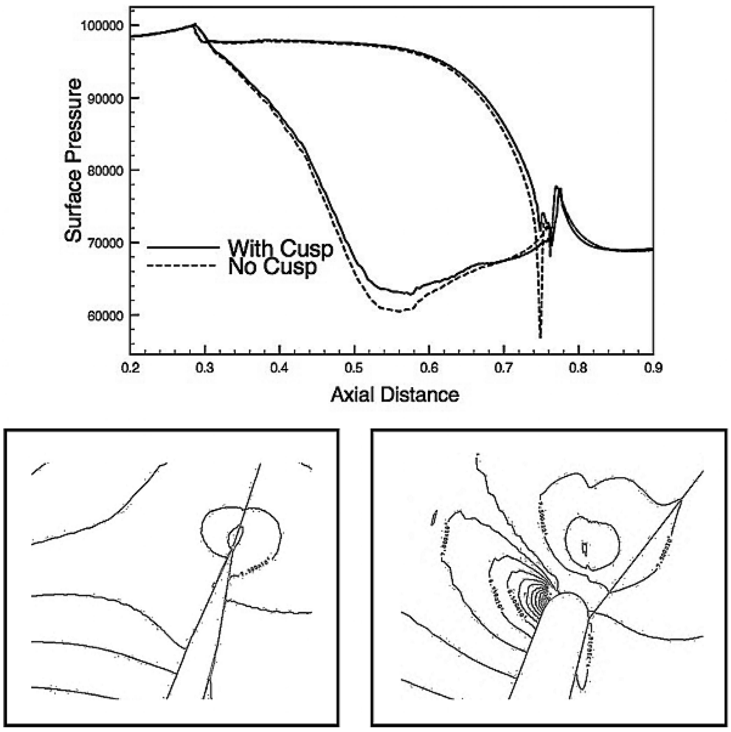

Figure 3.12. Use of a trailing edge cusp on MULTALL (adapted from [112]).

3.7 Comparative assessment of CFD solvers based on a three-stage axial compressor

As previously introduced in the chapter's introduction, a comparative analysis was conducted to evaluate the predictive trends and consistency of the MULTALL solver. The primary objective was to determine if the code could reliably generate large datasets required for training diagnostic NNs for axial compressors.

For this reason, a three-stage axial compressor originally proposed by Prof. Denton was selected as a reference test case [112]; the compressor consists of an academic design example, publicly available in the MULTALL program as a MEANGEN input file. It is highlighted that the selected machine does not correspond to a real industrial compressor for which experimental data is available. Conversely, it served as a numerical benchmark for CFD solvers comparison, aiming to investigate whether MULTALL was able to reproduce consistent aerodynamic trends when compared with a higher-fidelity RANS solver and to establish its modelling limits.

In this context, ANSYS CFX was employed as the reference high-fidelity solver. The Denton compressor geometry, originally generated within the MEANGEN-STAGEN-MULTALL workflow, was extracted and adapted for use in CFX. This approach enabled a systematic evaluation of solver consistency and isolated the impact of modelling assumptions, turbulence treatment and geometry resolution.

3.7.1 Reference compressor case

The availability of the original MEANGEN input file allowed the complete reconstruction of the selected machine geometry using the original MULTALL workflow. This aspect made the Denton compressor suitable as a benchmark for numerical comparison and avoided ambiguities associated with geometry reconstruction or undocumented modelling assumptions.

The compressor consisted of three consecutive axial stages with a mean radius of 0.5 m. It was designed to process an air mass flow rate of 50 kg s^{-1} at ambient conditions, while operating at a design speed of 5000 min^{-1}. The design is characterised by a reaction of 0.7, a flow coefficient (ϕ) of 0.6 and a load coefficient (ψ) of 0.4. Figure 3.13 presents its associated MULTALL mesh.

The resulting geometry and flow information were first employed on MULTALL to perform steady-state three-dimensional simulations of the machine.

The comparison between MULTALL and CFX was focused on the following elements:

- Global performance characteristics, including total pressure ratio, efficiency and mass flow rate, evaluated across the entire operating range.
- Circumferentially averaged spanwise distributions of selected quantities, such as axial velocity and swirl angle, analysed at the design operating condition.

- Qualitative and quantitative trends when moving away from the design point, particularly towards the surge point.

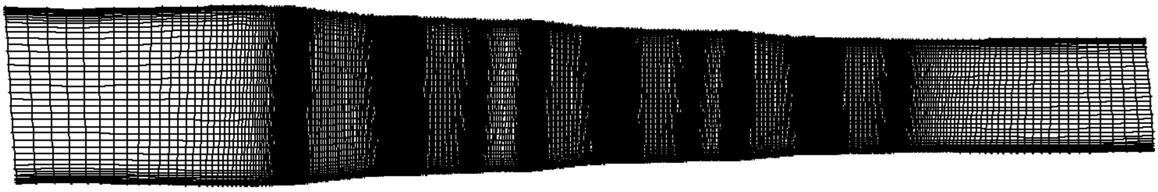

Figure 3.13. Mesh generated for the Denton three-stage compressor

A quantitative relative difference within 5 % between the two solvers was considered acceptable, in line with common practice in compressor CFD comparison studies.

It is important to note that, due to the absence of experimental data associated with the Denton compressor, the present analysis did not claim validation of either solver against experimental measurements. The use of the Denton case, as stated, is solely for comparison purposes to enable an evaluation of the capabilities and limitations of the MULTALL code compared to a higher-fidelity RANS solver.

3.7.2 Geometry extraction and preparation for high-fidelity CFD

The reference geometry used for the present comparison was generated using the native MEANGEN-STAGEN-MULTALL workflow. Starting from the original MEANGEN input file, the meanline design parameters were first used to define the velocity triangles and overall stage characteristics. The output data from MEANGEN was subsequently passed to STAGEN, where twelve blade sections per row were prescribed. From the resulting file, later used as input in MULTALL, the generated geometry was extracted and converted into a format suitable for high-fidelity RANS simulations.

The input for MULTALL provides direct access to the mesh node coordinates in axial, radial and circumferential directions for each blade row. Based on this information, five blade sections were extracted at consistent spanwise indices across all rotors and stators, together with the same corresponding hub and shroud contours. These data were exported as discrete coordinate sets and subsequently reorganised into a compatible format for the CFX preprocessing environment.

During the initial geometry transfer to CFX, it was noticed that the discretisation adopted within MULTALL, while adequate for its design-oriented modelling approach, was not sufficient for use in a high-fidelity RANS solver like CFX. Specifically, the number of points directly extracted from individual blade sections was limited, especially in regions of high curvature, such as the leading edge (see Figure 3.14).

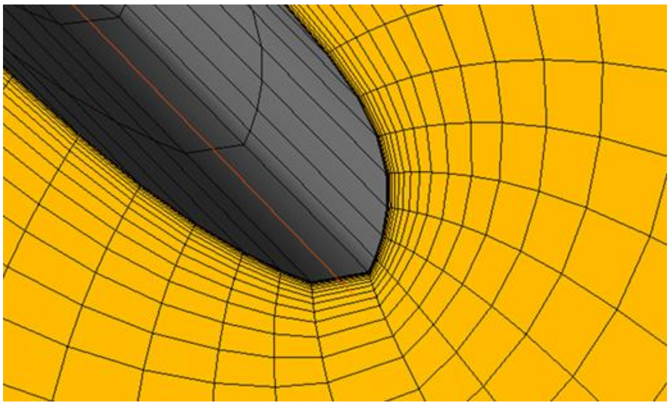

Figure 3.14. MULTALL geometry, directly imported into ANSYS CFX.

For this reason, the blade geometry was further refined using the B-spline curve fitting tools of the ICEM CFD software. Next, by resampling the profile with a higher point density at the leading edge, a smoother curvature was achieved, thus allowing the CFD solver to accurately capture the stagnation point location. Without this refinement, the polygonal approximation of the leading edge resulted in a distorted pressure distribution and eventually forced the stagnation region.

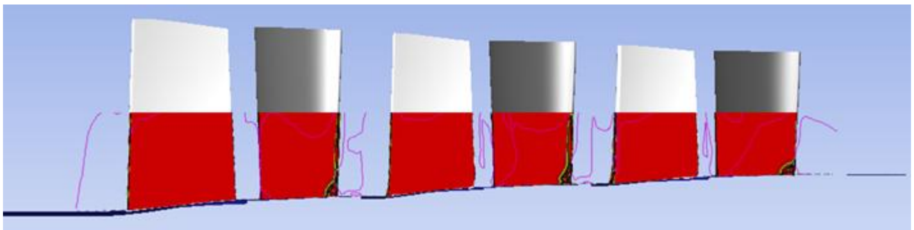

Figure 3.15. Final geometry comparison for the Denton compressor: red (MULTALL) and white-grey (CFX).

The mesh refinement was primarily driven by the modelling hypotheses that CFX employs. Unlike MULTALL, designed to predict mean aerodynamic characteristics with a relatively low geometric level, in CFX insufficient geometric resolution at the blade leading edge can significantly influence the boundary layer development and the associated loss mechanisms. Therefore, the purpose of the refinement was to make sure that the geometry representation was consistent without any effects on the original aerodynamic design of the compressor, meaning that any difference found between the results obtained with MULTALL and CFX could be assigned to the solver assumptions and not the geometry. A meridional view of the two final geometries is presented in Figure 3.15.

3.7.3 Numerical setup and solver alignment

For a proper comparison between MULTALL and ANSYS CFX, the numerical setup was defined with the main objective of granting solver-to-solver consistency. Both solvers were analysed under steady-state conditions, avoiding the introduction of time-dependent effects that could not be resolved in MULTALL.

In addition, because in MULTALL the interaction between consecutive blade rows is modelled through mixing planes, an equivalent treatment was adopted in CFX, where mixing plane interfaces were set at the same axial locations to preserve consistency in the modelling of rotor-stator interactions. Whereas frozen rotors were used for stator-rotor interfaces.

Boundary conditions and operating points were aligned across the two numerical codes as well. The same design rotational speed was imposed for all simulations. The inlet boundary condition was defined in terms of uniform total pressure and total temperature, 101325 Pa and 300 K, respectively. In MULTALL, the outlet is typically specified through a static pressure condition (radial equilibrium); the same equivalent boundary condition was applied in CFX. By progressively varying the outlet static pressure, the full compressor performance map was obtained with both solvers, enabling a direct comparison of operating trends from the near choke state towards the surge condition.

Regarding turbulence modelling, within MULTALL, simulations were carried out using the most recent version of the mixing length turbulence model. This model, primarily calibrated for the prediction of mean losses, relies on wall-function treatments for the evaluation of near-wall shear stresses, rather than on an explicit resolution of the viscous sublayer.

In CFX, the standard k-ε turbulence model was selected. While the SST k-ω model is typically preferred for resolving boundary layers in turbomachinery; this study employed the standard k-ε model to maintain consistency between solvers. By adopting the k-ε formulation, CFX utilised wall functions (in their scalable form) that mirror the near-wall treatment and loss-modelling framework of MULTALL. This alignment is crucial for a direct and comparative analysis of the aerodynamic losses predicted by both codes.

In other words, the turbulence modelling choice on CFX was driven by the need to establish a fair and controlled comparison between solvers characterised by different modelling philosophies, ensuring that observed differences in the results could be attributed primarily to the specific characteristics and limitations of the solvers, rather than to mismatched modelling assumptions.

Finally, a mesh sensitivity analysis was performed, leading to about 1.4 million elements on MULTALL (Figure 3.13) and about 8.1 million elements on CFX, with an average y^+ value of 30.

3.7.4 Results of the solver comparison

The comparison between MULTALL and ANSYS CFX was first conducted using global performance parameters across the analysed operating range and then complemented by a local flow field assessment at the design operating condition.

Figure 3.16 presents the total pressure ratio (right) and the total isentropic efficiency (left) as a function of mass flow rate predicted by the two solvers. Regarding the pressure ratio, both codes reproduced a consistent overall trend, exhibiting a monotonic decrease in pressure ratio with increasing mass flow rate. Table 3.1 presents the CFX results in terms of mass flow rate (MFR), total pressure ratio (PR), total isentropic efficiency and the corresponding MULTALL relative percentage errors with respect to CFX. It was observed a very good agreement between the pressure ratios obtained by the two codes, indeed deriving from the same type of outlet boundary condition. All percentage errors were below 1 % and the best result was obtained for a mass flow rate of 49 kg s^{-1}.

Of great interest is also the percentage difference in terms of mass flow rate. When moving towards the surge condition, a clear divergence was observed between the two software behaviours, over 5 % for MFRs values below 47.90 kg s^{-1}.

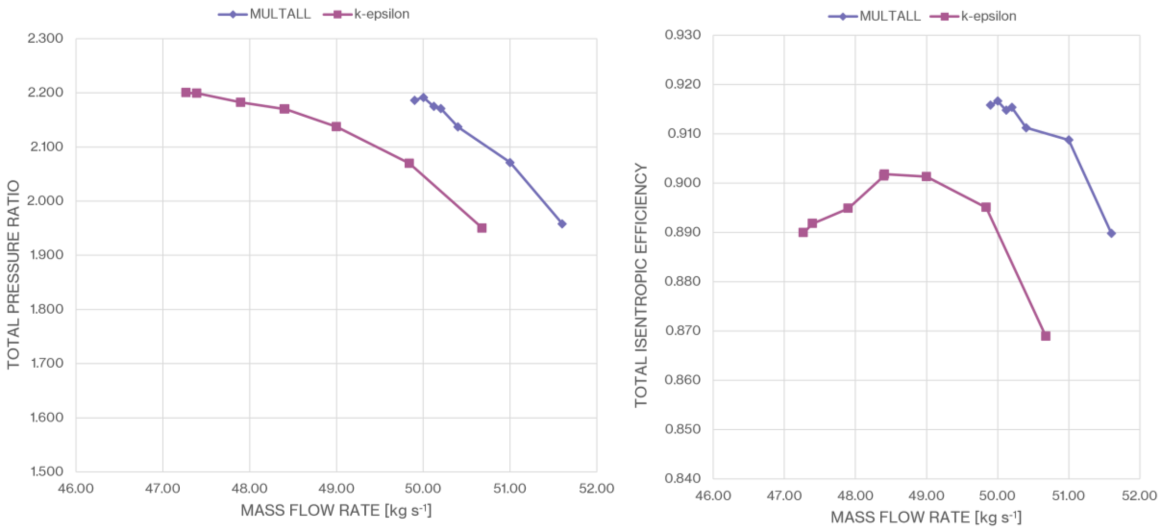

Figure 3.16. Characteristic curves of the Denton compressor obtained with the MULTALL code and ANSYS CFXIn particular, the data was split into 70 %.

Regarding efficiency, the two solvers exhibited different characteristic trends. MULTALL predicted a relatively flat efficiency plateau at lower MFRs, with a peak efficiency of 0.917 at the design flow of 50 kg s^{-1}. On the other hand, CFX reproduced a more typical parabolic curve, reaching a maximum efficiency of 0.902 at a slightly lower MFR of 48.40 kg s^{-1}. Despite this shift in behaviour, the discrepancy between the two solvers remained limited, with relative efficiency differences staying below 3%.

Table 3.1. ANSYS CFX results and MULTALL relative percentage error with respect to CFX.

MFR [kg s^{-1}]	Δ% MFR	PR	Δ% PR	Efficiency	Δ% Eff.
50.68	1.82%	1.95	0.42%	0.869	2.41%
49.84	2.33%	2.07	0.08%	0.895	1.53%

49.00	2.86%	2.14	0.02%	0.901	1.10%
48.40	3.72%	2.17	0.06%	0.902	1.55%
47.90	4.63%	2.18	0.33%	0.895	2.23%
47.40	5.49%	2.19	0.34%	0.891	2.79%
47.27	5.56%	2.20	0.62%	0.890	2.91%

The agreement was particularly close in the vicinity of the design operating condition, with larger deviations observed when moving away from it.

To complement the global performance comparison, circumferentially averaged spanwise distributions were sampled at a station downstream of the last stator (axial node 656 out of 704), for the design point condition. Figure 3.17 presents the swirl angle (a) and axial velocity (b) profiles, respectively. The spanwise distributions of swirl angle exhibited a generally consistent behaviour between the two solvers at the design condition. Both solvers predicted comparable swirl levels over a large portion of the span, while noticeable discrepancies are present in the endwall regions. In particular, CFX predicted larger deviations near the hub; on the other hand, MULTALL generated a smoother spanwise variation.

The axial velocity distributions predicted by MULTALL and CFX show similar trends over most of the blade span, with the highest level of agreement observed in the mid-span region. More pronounced differences were found again near the hub, where CFX predicted stronger velocity gradients.

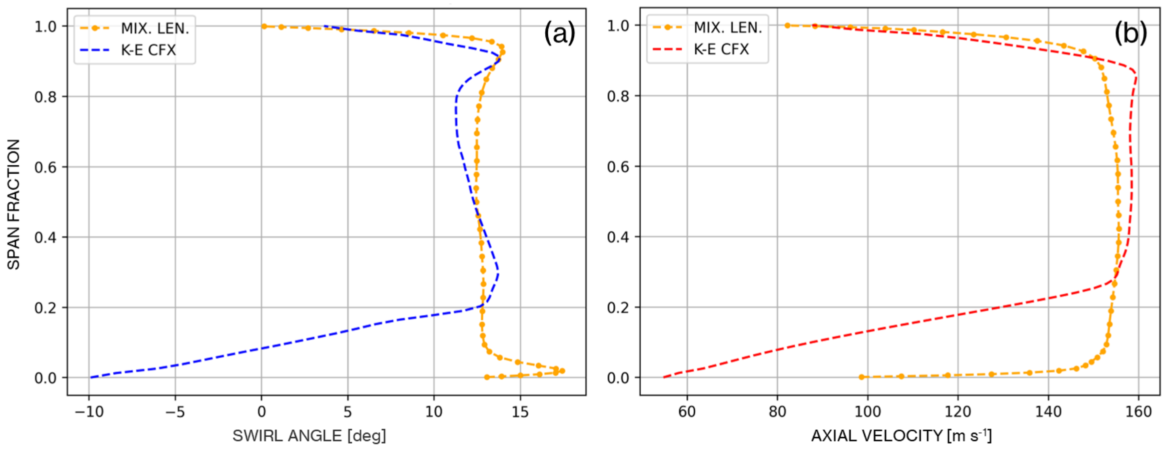

Figure 3.17. Spanwise distributions of swirl angle (a) and axial velocity (b) at the design point.

In conclusion, it was found that MULTALL is well suited for design-oriented studies, trend analyses and also preliminary performance assessments of multistage machines, particularly close to the nominal operating condition. At the same time, this comparison clearly indicated the limitations of the code when applied to off-design conditions where high separations, secondary flow effects and viscous effects are of major importance. Within the controlled numerical environment considered in this thesis, these differences can be attributed to the distinct modelling philosophies of the two solvers rather than to inconsistencies in geometry or boundary conditions.

3.7.5 Methodological scope and limitations of MULTALL

Beyond standard performance evaluation, the main focus of this thesis is on machine operation under faulty conditions, where the flow field is far from design conditions. Degradation mechanisms such as fouling, erosion or surface roughness growth cause local flow distortions, alter boundary layer behaviour and generate non-uniform losses, which cannot be entirely compatible with the solver approach adopted in MULTALL. As shown in the previous section, when operating outside the machine's intended design range, the solver's underlying assumptions progressively lose validity.

This limitation is further supported by another constraint of the present research: the unavailability of a real industrial compressor with experimental data suitable for validation under faulty operating conditions. Without experimental data, it would be impossible to recalibrate or verify MULTALL's internal models in the presence of degradation phenomena. Thus, any extension of its use to CM purposes would rely on unvalidated assumptions.

For these reasons, the use of MULTALL was discontinued in the next phases of this work. This choice is not the result of the solver's numerical inadequacy within its original scope, but a discrepancy between the solver's modelling approach and the validation needs set by the further studies regarding faults and degradation of turbomachines.

3.8 Chapter summary

This chapter presented the CFD tools employed within the thesis for the future development of diagnostic models for turbomachines. It focused on the modelling assumptions, the implications of the workflow and the technical application of each tool for repeatable aerodynamic analyses. As introduced, rather than presenting CFD results directly, the purpose of this chapter was to address the methodological context in which the different numerical solvers were evaluated.

In addition, a quantitative comparison of a three-stage axial compressor using the design-oriented turbomachinery code MULTALL and the higher-fidelity code ANSYS CFX was performed to verify solver consistency and to establish the boundaries of applicability of MULTALL in off-design applications and in the lack of experimental data. The results highlighted that, while MULTALL remains suitable for trend-based analyses, its modelling philosophy and validation constraints are not compatible with investigations of faulty and degraded machines. For these reasons, MULTALL was discontinued in favour of CFD approaches better aligned with the objectives of the subsequent chapters.

4. Condition monitoring and instability diagnostics in radial compressors

4.1 Purpose and scope of the chapter

The current chapter addresses CM and diagnostic methods for radial compressors, with a specific focus on the detection, reconstruction and spatial localisation of unsteady flow phenomena associated with the onset of aerodynamic instability. From a maintenance and operational reliability perspective, the onset of instability in a compressor, and in particular surge, is a failure mechanism that can cause severe degradation in performance, reduce component life and lead to potential structural damage.

Instability that typically manifests in the form of low-frequency oscillations in the internal flow field. From a monitoring perspective, the early identification of such unsteady phenomena is therefore essential, specifically in applications where the compressor operates close to its stability limits or is tightly coupled with downstream systems.

This chapter integrates and extends the CFD modelling framework introduced in Ch. 3 and the ML theory of Ch. 2, presenting results from two peer-reviewed studies by the author into a unified framework for compressor CM [103,113]. The studies are based on a numerical analysis of a two-stage radial compressor for refrigerant applications, previously introduced in Sect. 3.4.1 of Ch. 3, originally based on the work of Marsano et al. [110]. The CFD simulations generated an URANS database, which provided time-resolved static pressure signals at multiple locations within the machine.

Two distinct yet complementary monitoring approaches were developed. The first approach employed a ML-based architecture to reconstruct diffuser time series from the corresponding set of inducer monitors, fewer in number and more practical to install. This reconstruction allows for the classification of the stage operating state regarding instability, employing tailored ML models for both instantaneous and short-term predictive assessments. This method is grounded in the physical observation that instability mechanisms primarily originate and develop within the diffusers.

The second approach employs entropy-based signal processing, approximate entropy (ApEn), to quantify the complexity of pressure signals in both the inducer and diffuser. This method identifies the monitoring locations most sensitive to the onset of surge and provides a quantitative basis for sensor placement optimisation that effectively complements the predictive ML framework.

Although the approaches presented here were developed and assessed using numerically generated time series, they were formulated with the explicit objective of applicability to measured industrial signals and can be compatible with typical compressor instrumentation.

4.2 Compressor configuration and URANS dataset

The first stage of the compressor under examination was equipped with 11 impeller blades plus 11 splitter blades, while the second stage was defined by a 9-bladed impeller with 9 splitter blades. Both stages featured vaneless diffusers and inlet guide vanes (IGVs).

As mentioned before, this machine flow behaviour was analysed through URANS simulations at 100 % of its design rotational speed at four operating points, highlighted with orange markers in the characteristic curves presented in Figure 4.1

As detailed in Sect. 3.4.1 of Ch. 3, these are the Best Efficiency Point (BEP), the Near Surge point (NS), the Surge point (SU) and a final simulated point referred to as New Point (NP).

Within each compressor stage, two distinct elements are important, namely the inducer, located upstream of the impeller, and the diffuser, where the flow undergoes deceleration and pressure recovery. In the previous study [110], monitoring points were strategically positioned within each stage inducer and diffuser to mimic a realistic experimental sampling system. In the inducer, 12 static pressure monitors were subdivided into 4 circumferential positions (RA, RB, RC, RD) at 3 radial positions (10 %, 50 % and 90 % of span). Figure 4.2 (a) presents the set of inducer monitors for the first stage of the radial compressor.

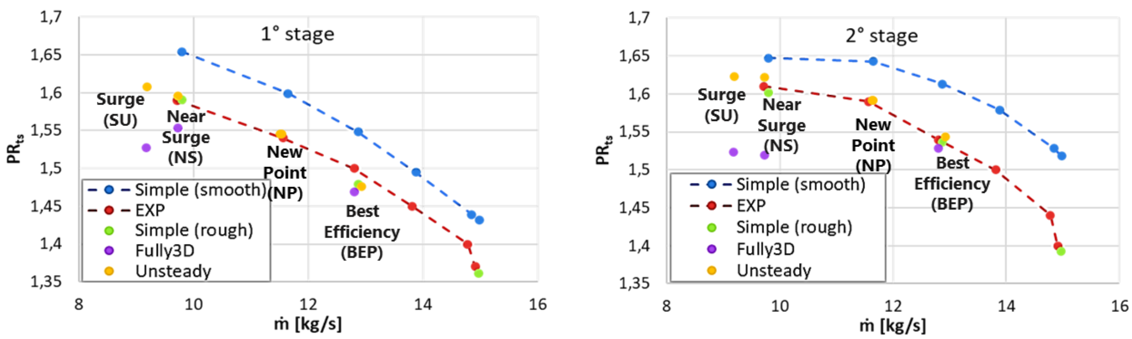

Figure 4.1. Characteristic curves of the two stages of the analysed radial compressor (adapted from [113]).

In each diffuser shroud, pressure monitors were positioned at two key radial locations (10 % and 90 % of the diffuser exit). For each stage, the number of diffuser monitors matched the corresponding number of impeller blades, resulting in a total of 22 probes for the first diffuser and 18 for the second one.

As depicted in Figure 4.2 (b), numbers were assigned in ascending order following the circumference of the volute from the tongue to its outlet. Specifically, 1 to 11 for stage 1 and numbers from 1 to 9 for stage 2. The second digit represented the percentage of the diffuser exit (10 % or 90 %).

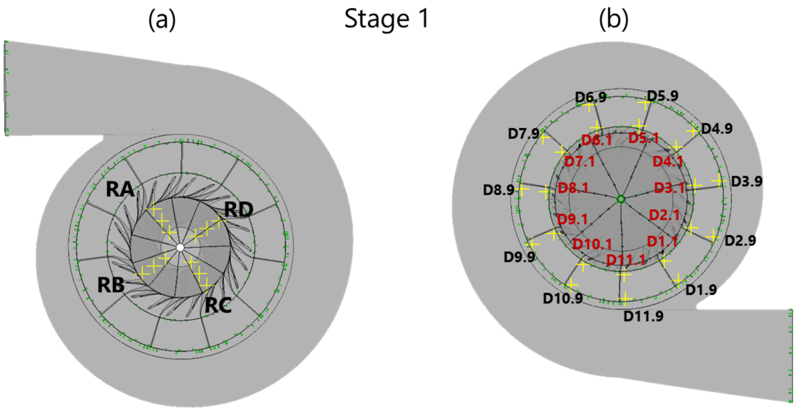

Figure 4.2. Pressure monitors for stage 1 (adapted from [113]).

To ensure accuracy, time series were sampled only after the numerical simulations had reached a statistically steady regime. In particular, for each of the four simulated flow rates, only the last 1800-time steps were selected, equivalent to about 27 impeller rotations, with a simulation time step equal to 8.26×10^{-5} s.

The consolidated URANS database comprised an inducer dataset of 4x12x1800 pressure records for both stage 1 and stage 2, plus 4x22x1800 and 4x18x1800 samples for the stage 1 and stage 2 diffusers, respectively. For every operating point, each monitored signal was subsequently normalised by calculating a pressure coefficient (CP) time series as defined in Eq. (3.2) of Ch. 3.

4.3 Monitoring strategy and physical origin of instability

The monitoring approach followed in the present chapter is directly informed by the physical mechanisms governing compressor instability; with focus on the spatial localisation of unsteady phenomena within the stage.

From the physical point of view, the diffuser is the most important region where the development of instability can take place. This is consistent with the classical synthesis of stall and surge by Day [114], who, while focusing primarily on axial aero-engine compressors, explicitly distinguishes centrifugal machines as a separate class in which instability frequently originates in the exit diffuser. During off-design conditions, the adverse pressure gradient formed in the diffuser enhances flow separation, rotating stall structures, and low-frequency pressure oscillations, which develop progressively as the compressor approaches surge.

Although the instability mechanisms originate and evolve essentially within the diffuser, their effects are not restricted to this latter region. The associated unsteady pressure fluctuations propagate upstream through the impeller, imprinting indirect signatures on the flow field at the inducer and inlet sections. For this reason, the measurements taken upstream may have some diagnostically relevant information, despite being spatially detached from the very origin of instability.

On that basis, a distributed approach to monitoring is developed here, integrating measurements in both inducer and diffuser regions. Diffuser monitors provide direct access to the unsteady dynamics associated with instability inception and growth, on the other hand, inducer monitors act as indirect indicators of the evolving stage behaviour. This distinction underpins the diagnostic logic elaborated in the next sections and represents an implicit trade-off between physical relevance and practical issues related to sensor deployment.

The use of multiple spatially distributed monitors is further motivated by the inherently non-uniform nature of instability phenomena within radial compressors. Flow separation and rotating stall cells influence the diffuser non-uniformly, thus the signature of instability may appear strong or, at least, more pronounced at specific circumferential or radial locations. Consequently, relying on a single monitoring point might be insufficient to characterise the onset and progression of instability with adequate robustness.

This framework represents the conceptual basis for reconstruction, classification and localization methodologies developed in the subsequent sections.

4.4 ML-based diagnostic architecture

Building on the monitoring strategy introduced in Sect. 4.3, a multi-model ML architecture was employed to assess the aerodynamic stability of the investigated radial compressor using static pressure time series extracted from the URANS simulations.

The diagnostic strategy was based on the decomposition of the problem into complementary submodels, each addressing a specific aspect of the instability detection and prediction task. The first model, SIGMA (Signal Input-Output Generalized Mapping Algorithm), was capable of regression and was designed for predicting the flow structure in the compressor stage diffuser, based on the set of inducer's signals. The second model, OPSA (OPerational Stability Assessment), was a classification model that was developed to identify signal patterns in the predictions provided by the SIGMA model. The OPSA model helped to determine whether the compressor was close to surge or not by classifying the compressor operational state into three possible classes: stable, transitional and unstable.

In addition to the present-state assessment, obtained by the combination of the SIGMA and OPSA models, a forecasting capability was also introduced through a third model, TCAN (Temporal Convolutional Attention Neural Networks for Time Series Forecasting). The TCAN architecture employed here is that of Lin, Koprinska & Rana (2021) [108], an attention-augmented extension of the temporal convolutional network (TCN) by Bai et al. [107], and should not be confused with other architectures sharing the same name. Utilized to predict the future evolution of the stage inducer pressure signals, TCAN's forecasted outputs are coupled with the SIGMA-OPSA architecture. This integrated approach ultimately enables the short-term prediction of compressor operating conditions.

A schematic representation of the combined ML architecture is provided in Figure 4.3, illustrating the flow of information between the individual submodels and their role in present and future state assessment.

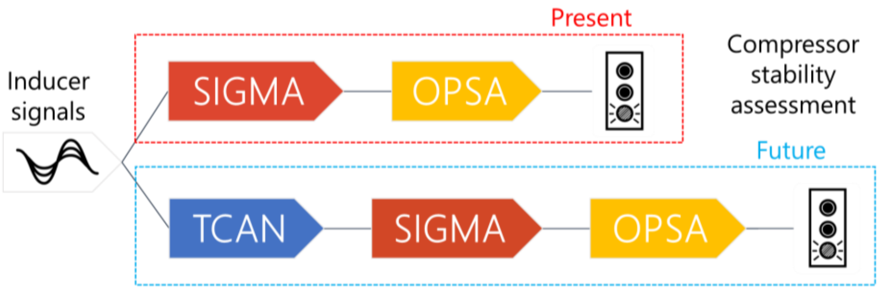

Figure 4.3. Schematic of the ML models combination (adapted from [113]).

4.4.1 Signal preprocessing

In order to train the ML architecture for proper compressor satbility assessment, the signals from each monitor and operating point were normalised through a Z-score normalization procedure. Each signal was standardized using the following formula:

$$Zx = \frac{X - \mu}{\sigma} \tag{4.1}$$

where Zx represents the standardized signal, X is the raw score, μ refers to the mean of the dataset and σ indicates the standard deviation of the dataset.

For each operating point, coherent noise was also added to each signal to improve the robustness of the ML models, to reduce the possibility of overfitting and to facilitate a better representation of the variations and uncertainties present in real-world data. Indeed, industrial signals can be influenced by various sources of noise and uncertainty and the addition of noise can help make simulation results more realistic.

As outlined in Sect. 4.2, the dataset associated with each compressor stage consisted of time series of monitored static pressure signals for the four operating points: BEP, NS, SU and NP, lasting 1800 time steps, respectively.

4.4.2 SIGMA and OPSA data structuring

To train and validate the SIGMA and OPSA models the selected dataset included signals related to the BEP, SU and NP. For each operating condition, only the first 1700-time steps were considered. The time series associated with the selected states were stacked to define an inducer matrix of dimensions 12x5100. In addition, diffuser matrices of dimensions 22x5100 for stage 1 and 18x5100 for stage 2 were defined.

To improve the robustness and accuracy of the model and to reduce the possibility of bias, training, and validation datasets were shuffled randomly. Furthermore, shuffling the rows ensured that the model did not base its predictions on a particular ordering of the data, thereby promoting greater objectivity in the evaluations.

In particular, following standard practice for datasets of this size [93,94], the data was split into 70 % for model training, while the remaining 30 % was dedicated to model validation. Consequently, the training inducer matrices for both stages had dimensions of 12x3570; and the corresponding validation matrix resulted to be 12x1530. On the other hand, the training matrix for the stage 1 diffuser had dimensions of 22x3570, while its validation counterpart was 22x1530. Similarly, for the stage 2 diffuser, the training matrix resulted to be of 18x3570, while its validation counterpart had dimensions of 18x1530. With regard to the dataset used for the TCAN model, further details will be provided in the next section.

For testing the ML architecture in its present-state configuration, two tests were defined. For the first test (Test 1), the final 100 data points of the BEP, NP and SU signals, not used during the training phase, were selected. For each inducer, the records were stacked together resulting in an input matrix of 12x300, while for the two diffusers, the input matrices were 22x300 and 18x300, respectively. In the second test (Test 2), the SIGMA-OPSA architecture was tested with the full near NS dataset, which is 1800-time steps long.

This testing approach exposed SIGMA and OPSA to input signals associated with a compressor-processed operating range not encountered during the training and validation phase of each compressor stage. This methodology allowed to assess the model’s ability to generalize and perform on data outside of its training phase. The fourth operating point was solely deployed for testing and served as a crucial indicator of the model's robustness and generalized performance. If the model performed well under this unseen condition, it implies that the learning process had captured critical features and trends applicable to different operational settings.

4.4.3 TCAN data structuring

Moving to the third model utilised in the future-state ML architecture, TCAN was fine-tuned to process inducer signals. This resulted in a smaller training dataset, quicker to train and not dependent on the number of diffuser monitors in the two stages.

The model was trained using a dataset moving from the NP to the NS operating state. These conditions were selected to ensure the TCAN framework activated only upon detecting a transitional state during real-time monitoring, a critical requirement for industrial deployment. Furthermore, the analysis focused exclusively on the mean forecasted signals, the additional quantile outputs produced by the algorithm were omitted from this specific analysis.

To train the TCAN, an artificial transient operation of the compressor was created. For the NP condition, the last 1200 time steps of the 1800-step long signals were used, while for the NS condition, only the first 800 time steps were condidered. The number of selected time steps for the NP was greater than those of the NS to ensure that the TCAN could also efficiently forecast signals trending towards the transitional condition.

In addition, to complete the creation of artificial signals and assemble NP and NS time series, two transition functions were defined for each pressure monitor, one towards the NP condition and another one between the NP and NS conditions, using dedicated exponential functions.

As a result, 12 inducer transient signals for each stage were generated, starting from the first transition and ending at the NS condition. The first transition lasted for 400 time steps, while the second one lasted for 800 time steps, An example of a training transient signal is given in Figure 4.4 for probe 12 of stage 1. The model was trained on the analytically composed time series and validated using the predicted 132 data points, equivalent to two rotational periods, subsequent to the training NS signal.

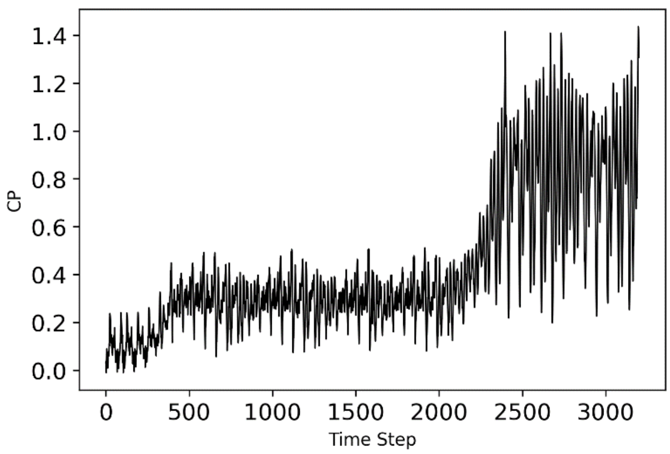

Figure 4.4. TCAN Training signal for probe RD (90%) of stage 1 (adapted from [113]).

As part of the TCAN testing phase, two transient operations were generated (Upward test and Downward test). These operations moved from the NP to the NS and from the NS to the NP, respectively.

To validate the model, only data points excluded from the training phase were utilized. Specifically, both test cases used the initial 300 time steps of the NP state. For the Upward test a 300-step window (from step 1368 to 1668) of the NS condition was selected to predict the final 132 time steps across the 12 inducer time series. Conversely, the Downward test utilized the final 300 steps of the NS state to predict the subsequent 132 steps of NP operation.

Each test incorporated a 300-step transition between operating states. To assess the model's sensitivity to transitional dynamics, this phase was modeled using an exponential function distinct from the one employed during training.

4.4.4 Specifics of SIGMA model

The primary objective of SIGMA was to establish a correlation and a map between the input signals received from the stage inducer and the corresponding diffuser output signals. To accomplish this, SIGMA had to be able to return the recorded values by the output probes, instant by instant, provided the related input signals. So that, in principle, only the input signals from the inlet sensors would be then sufficient to have a clear understanding of the flow behavior in the stage diffuser.

SIGMA consisted of a shallow NN, specifically a multi-layer perceptron (MLP), initially designed for stage 1 and later adapted for stage 2. It comprised an input layer with 12 neurons, processing samples from the 12 inducer monitors, followed by two hidden layers with 156 neurons each. The NN terminated with an output layer containing 22 neurons for stage 1 and 18 for stage 2, corresponding to the values recorded by the output probes located in the diffuser.

The model was therefore capable of mapping the output pressure distribution at the diffuser shroud based solely on the inducer signals. With regard to the activation functions, the linear function was selected for the input layer, while the ReLU and Leaky ReLU (γ = 0.2) functions were chosen for the two hidden layers, respectively. The Swish activation function was chosen for the output layer. Dropout layers were also incorporated to prevent overfitting, with a dropout rate of 0.2 and no need for L2 regularization.

To optimize the model, the Huber loss function was selected, with its parameter δ set equal to 0.8 . As introduced in Sect. 2.2.3 of Ch. 2, this type of function achieves balance between the robustness of quadratic errors and the linearity of absolute errors, making the NN more resilient to outliers. To train the model, the ADAM optimization algorithm was implemented, featuring a learning rate of 0.0002, a batch size of and trained for 350 epochs.

Through hyperparameter tuning, the model's parameters were adjusted by analyzing both the training and validation loss values. The hyperparameters were refined iteratively, balancing the model complexity with its generalization performance.

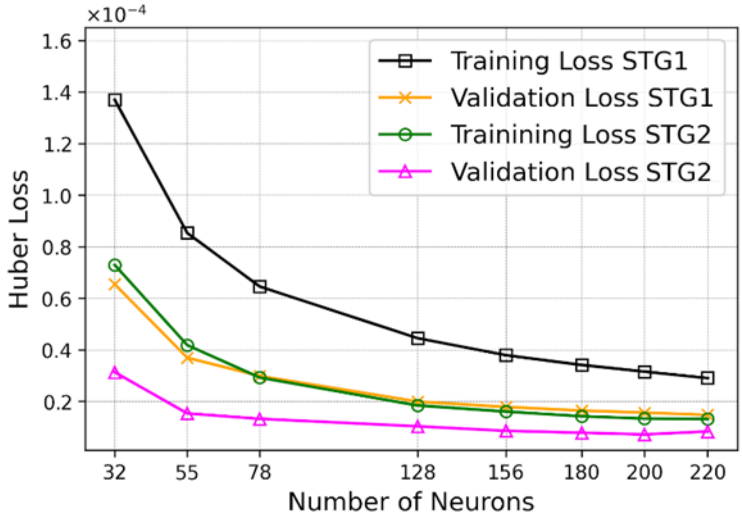

Figure 4.5. Huber loss for several SIGMA architectures (adapted from [113]).

In Figure 4.5, presents the number of neurons selected for the hidden layers against the training and validation losses for both compressor stages.

A configuration of 156 neurons per layer was selected for both stages, as the validation loss plateaued beyond this threshold. This choice balanced model complexity with computational efficiency, ensuring the architecture remained easy to train while mitigating the risk of overfitting.

4.4.5 Specifics of OPSA model

The OPSA classification model was designed as a valuable ML too to categorize flow patterns in the stage diffuser. By analyzing CP signals in the diffuser, the model could identify compressor stability conditions. As previously stated, the diffuser is the key area where flow instabilities establish, thus where the stability state is best classified.

A visual representation of the CP trends in the diffuser, for the four operating points simulated with CFD, is displayed in Figure 4.6. The plots display the averaged values for each probe over one impeller revolution, spanning 66 timesteps. The 10 % and 90 % distributions are presented with polar plots, as a function of number of probes and then translated into a Cartesian plot, to visualize clearly the flow behavior at the different operating points.

Clear and specific flow patterns are made visible by both 10, 90 % curves for each operational state. The curves have an approximate slope that progressively reverses when moving from BEP to the SU point. As for the NP, a transitional behavior can be seen in the curve trend where the general slope is almost zero. This indicates that it is in a condition somewhere between stability and instability.

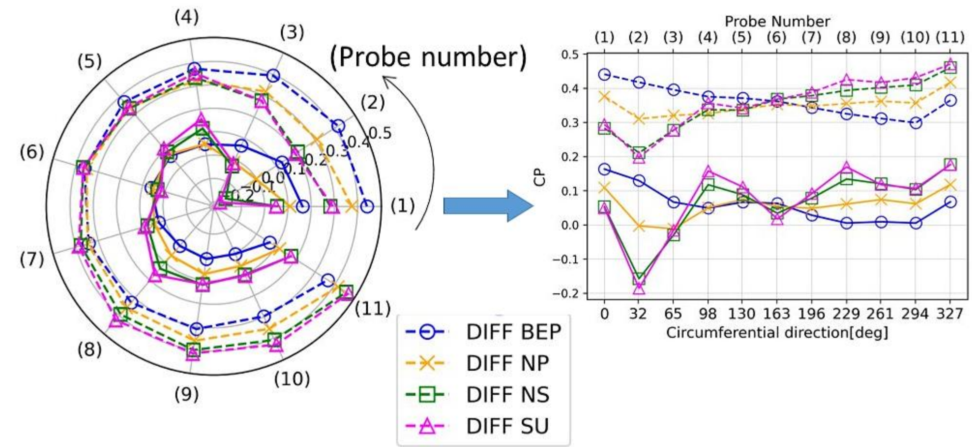

Figure 4.6. Averaged CP distributions for each compressor operating state (stage 1) (adapted from [113]).

As a result of these findings, a ML model for classification purposes was developed that could distinguish flow patterns in the stage diffuser into three categories: stable (class 0), transitional (class 1) and unstable (class 2). BEP falls into class 0, NP falls into class 1 and NS and SU points fall into class 2.

For the future application and combination of the SIGMA and OPSA models, the classification algorithm would receive the diffuser CP signals reproduced by the regression model SIGMA.

After an analysis of the revolution-averaged diffuser curves, some important elements can be identified to define the input features for OPSA. These elements include:

a) The average value of the CP (at both 10 % and 90 %) is different under different conditions.
b) The CP value decreases between probes 1 and 2 (at 10 % and 90 %) and the effect becomes more relevant when moving towards unstable conditions.
c) Between probes 2 and 3 (at 10 % and 90 %), there is a strong pressure rise under unstable conditions, which is not present at the BEP and is only hinted at 90 % at the NP. This is also a consequence of the previous point.
d) The point-to-point variation in CP between 90 % and 10 % curves, specifically for probes 2 and 3, tends to increase towards surge.

Based on these observations, eight input features were extracted from the revolution-averaged CP diffuser signals. These features formed the input layer of a classification-based MLP, which consisted of 8 neurons followed by two hidden layers of 66 neurons each, utilizing ReLU activation functions throughout. Dropout regularization with a rate of 0.2 was included in both hidden layers. The model was trained using the ADAM optimization algorithm, with a batch size of 64, a learning rate of 0.005 for 300 epochs. The implemented cost function was the logistic regression.

It is also worth noting that the selected activation function for the last layer was the Softmax. This enabled the model structure to return the probability distribution of the three classes of operation: stable (0), transitional (1) and unstable (2), respectively.

The OPSA training utilized CFD data from stable (BEP), transitional (NP), and unstable (SU) operating points, matching the same diffuser dataset of the SIGMA training dataset. The same tests defined for SIGMA were analogously defined OSPA. In addition, they were performed twice: first on CFD data and then on data generated by the SIGMA model.

This was done to evaluate the strength of the coupling between the two algorithms. In addition, to ensure network's accuracy, the instantaneous predicted values were averaged every 66-time steps (one impeller rotation), covering a duration of approximately 27 periods. This averaging choice was motivated by the fact that practical applications require the machine operational stability to be classified over at least one rotational period, rather than instantaneously.

Finally, the hyperparameters for the OPSA model were calibrated based on the data from stage 1. Subsequently, it was observed that the same hyperparameters yielded accurate results for stage 2, as well.

4.4.6 Specifics of TCAN model

To meet the research goals for short-term stability prediction, the TCAN model was selected as an algorithm specifically designed for probabilistic forecasting tasks, with a specific focus on time series forecasting [108].

Unlike other TCNs, which may require numerous convolutional layers for long input sequences and lack interpretability, TCAN introduced an innovative solution. By integrating an attention mechanism with temporal convolutions, TCAN leveraged the hierarchical convolutional structure of TCNs to capture temporal dependencies. Moreover, the model employed sparse attention to selectively focus on crucial timesteps, facilitating an extended receptive field without the need for a deeper architecture. This sparse attention layer not only enhanced forecasting accuracy but also enabled the interpretability of the results. Notably, TCAN achieved higher accuracy with a reduced number of convolutional layers, making it faster to train.

The evaluation conducted by the authors on three large solar power datasets underscored TCAN as a compelling alternative, offering a balance of accuracy, efficiency and interpretability in the realm of deep learning-based forecasting models.

Hyperparameter tuning for the TCAN model was done with a lookback period of 132 time steps (two impeller rotations) and a forecast period of 132 time steps, as well. The model was trained with three dilation rates [1, 2, 4], 32 filters, a kernel size of 3 and a batch size of 32. The dropout rate was set to zero and the regularization parameter to 1.25. The selected activation function was the Softmax, plus a learning rate scheduler. The model was trained for 250 epochs, with the same hyperparameter configuration used for both compressor stages.

4.4.7 Reconstruction and classification results

The results of the three ML models are presented sequentially, starting with the regression model SIGMA, followed by the OPSA classifier and finally the TCAN forecasting model.

SIGMA results. This section introduces the results of the first ML model, SIGMA. The model was evaluated using the data from Test 1 and Test 2, previously described in Sect. 4.4.2. Figure 4.7. Regression results of Test 1 and Test 2 for stage 1 (a) and stage 2 (b) (adapted from [113]).(a) and (b) present the instantaneous predicted values for each monitoring probe within their respective diffuser. The values were compared against their reference values for all four operating points included in Test 1 and Test 2.

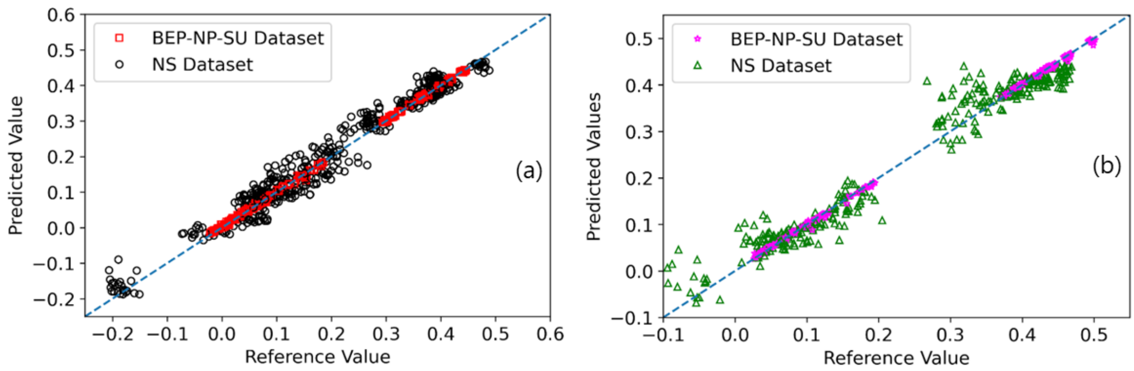

Figure 4.7. Regression results of Test 1 and Test 2 for stage 1 (a) and stage 2 (b) (adapted from [113]).

Both stages showed a good match with the corresponding reference values. However, the NS point exhibited greater oscillation due to the intrinsic nature of Test 2.

An example of the predicited monitors of Test 1, following the BEP-NP-SU order, is shown in Figure 4.8 (a). These two time series corresponded to the pressure probes closest to the volute's tongue (pressure probes 1) of stage 1, situated at 10 % and 90 % of the diffuser exit, respectively. The predicted results indicated a satisfactory coherence with the reference values. Similarly, Figure 4.8 (b) presents the time series associated with the NS conidition of Test 2, showing the same signals probes selected for the previous figure. It is evident that the NS time series had a good average agreement with the reference ones, but with greater fluctuations.

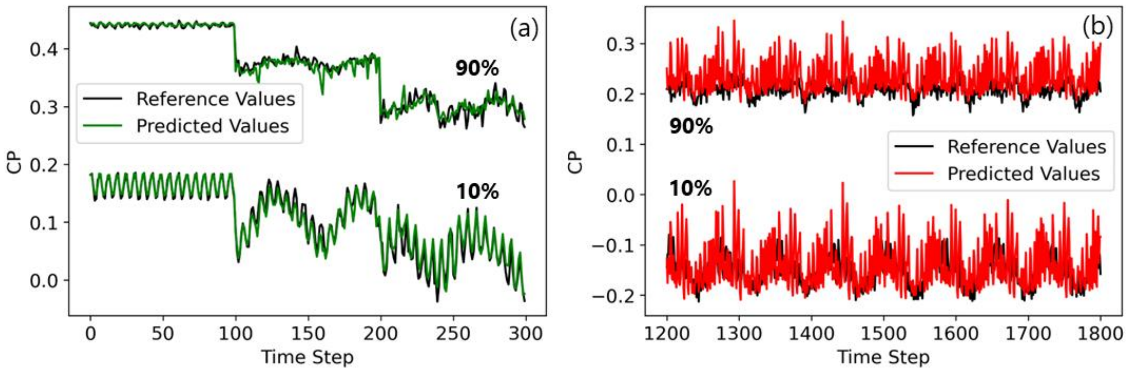

Figure 4.8. SIGMA results of Test 1(a) and Test 2 (b) for stage 1 (adapted from [113]).

Figure 4.9 presents the CP contours for diffuser 1, generated in MATLAB for a generic instant. To determine the 10 % and 90 % CP distributions, a cubic spline fit was employed to ensure consistency with the post-processed CFD contours. The resulting plots compared the reconstructed BEP contours against the reference state from Test 1; the NS point for Test 2 was processed analogously. Even though the SIGMA model was not trained on the NS dataset, a good agreement can be seen for both tests. However, the accuracy of the results is lower than that of Test 1 due to this same reason. Nevertheless, the flow regime is well predicted.

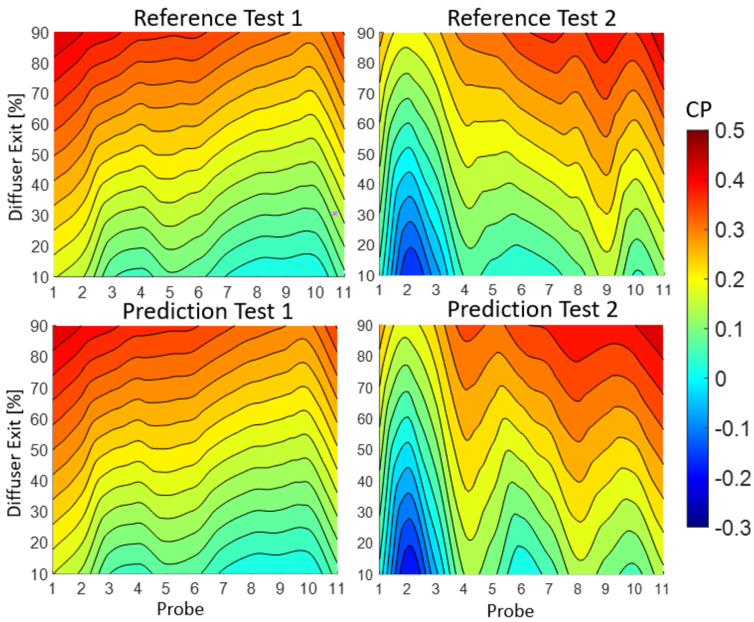

Figure 4.9. CP contours for BEP (left) and NS (right) operations (adapted from [113]).

Moving on to the averaged data, the signals produced instantly at each stage diffuser were averaged over each period of impeller rotation. For the NS case, this resulted in 27 averaged CP curves. As a result, averaged CP distributions at 10 % and 90 % of the diffuser exit were obtained and could be compared against the reference ones along the circumferential coordinate for both Tests 1 and 2.

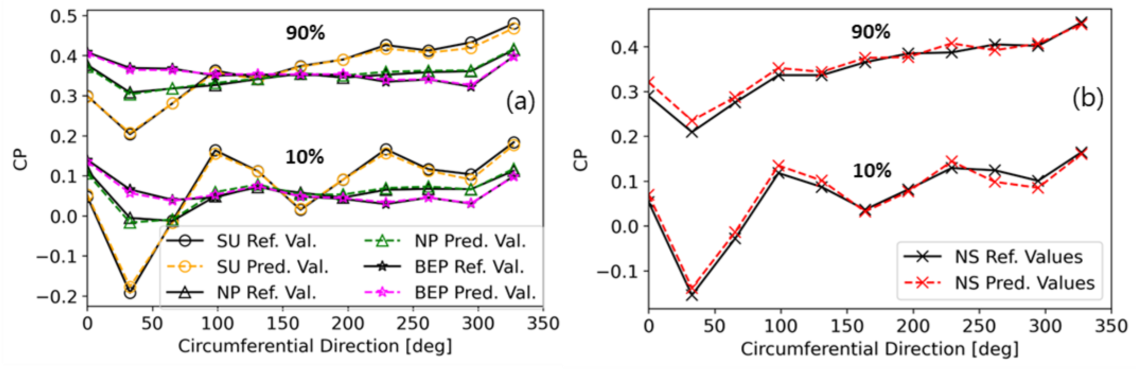

Figure 4.10. Averaged diffuser distributions of Test 1 (a) and Test 2 (b) for stage 1 (adapted from [113]).

Figure 4.10 (a) and (b) shows, an example, the last averaged CP distributions for all four compressor operating points for Tests 1 and 2 of stage 1. The averaged values demonstrated very good conformity, even for the NS condition, as the fluctuations were smoothed out. These results were critical for the forthcoming OPSA model.

Once the accuracy of the SIGMA model was verified based on the 12 signals from the input probes to the stage, a sensitivity study on the accuracy of the model was performed by re-training SIGMA with a reduced number of inducer probes and testing it again on the four compressor operating points of Tests 1 and 2. The result analysis was performed on the averaged CP values.

The first monitors to be excluded were the ones at 10 % of the inducer span, thus reducing the number of input probes to 8; finally, the remaining probes at 50 % were aslo eliminated, leaving only the four probes located at the inducer's shroud. The purpose of the test was to determine whether the network could still identify the operating pattern despite a reduced setup of inducer monitors.

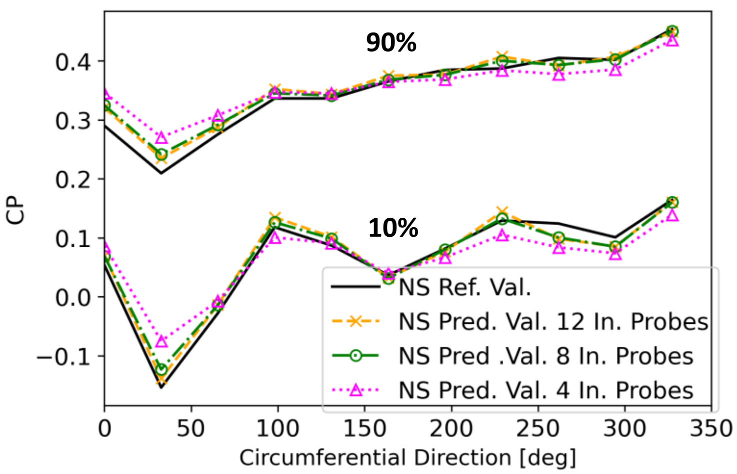

Figure 4.11. Averaged diffuser distributions for different numbers of input probes for stage 1 (NS) (adapted from [113]).

In Figure 4.11, the results of Test 2 (NS condition, last rotational period) of stage 1 are shown as a representative example. Even with only four inducer probes, the model could produce an accurate CP trend, with no need to change the subsequent OPSA model. The RMSE values for Tests 1 and 2 are presented in

Table 4.1. These results account for both stages and include comparisons for both instantaneous CP distributions and their averaged counterparts.

Table 4.1. RMSES of the SIGMA model predictions (adapted from [113])..

RMSE	Stage 1		Stage 2	
	12 Probes		**12 Probes**	
	Test 1	**Test 2**	**Test 1**	**Test 2**
Instantaneous	0.012	0.036	0.008	0.021
Averaged	0.005	0.015	0.003	0.013
	8 Probes		**8 Probes**	
	Test 1	**Test 2**	**Test 1**	**Test 2**
Instantaneous	0.018	0.036	0.012	0.018
Averaged	0.008	0.015	0.004	0.009
	4 Probes		**4 Probes**	
	Test 1	**Test 2**	**Test 1**	**Test 2**
Instantaneous	0.033	0.048	0.026	0.027
Averaged	0.017	0.029	0.014	0.009

OPSA results. As previously introduced in Sect. 4.4.2, the OPSA classification model was trained using the CFD diffuser signals as input. In the test phase, both CFD and SIGMA signals were used, taking into account both instantaneous and averaged CP distributions (one impeller rotation). In addition, the signals produced with SIGMA were considered for also for the reduced set of inducer probes (12, 8, 4).

The predictions of the classification model are reported, for both stages, in Table 4.2 for Test 1 and 2 in terms of accuracy and defined as:

$$Accuracy = \frac{\#\, True\ Prediction}{\#\, Total\ Prediction}\ [\%] \tag{4.2}$$

It is highlighted that, regarding the predictions of the classification tests of Test 1, the three operating conditions (BEP, NP, SU) were categorized separately. This was equivalent to classify almost two rotational periods for each operating point, being the time series length 100 time-steps.

Table 4.2. Accuracy of the OPSA test predictions (adapted from [113]).

Accuracy	Stage 1		Stage 2	
	From CFD signals			
	Test 1	**Test 2**	**Test 1**	**Test 2**
Instantaneous	100 %	99.9 %	100 %	99.2 %
Averaged	100 %	100 %	100 %	100 %
	From SIGMA signals			
	12 Probes		**12 Probes**	
	Test 1	**Test 2**	**Test 1**	**Test 2**
Instantaneous	99.0 %	84.3 %	100 %	72.5 %
Averaged	100 %	100 %	100 %	84.0 %
	8 Probes		**8 Probes**	
	Test 1	**Test 2**	**Test 1**	**Test 2**
Instantaneous	95.3 %	79.6 %	97.7 %	86.3 %
Averaged	100 %	100 %	100 %	100 %
	4 Probes		**4 Probes**	
	Test 1	**Test 2**	**Test 1**	**Test 2**
Instantaneous	76.3 %	48.8 %	78.3 %	80.6 %
Averaged	100 %	76.0 %	100 %	100 %

Regarding the averaged NS distributions for Test 2, Stage 1 accuracy remained at 100 % when input probes were reduced to eight, but declined when only four were used. Conversely, Stage 2 exhibited an inverse sensitivity, with classification accuracy increasing as the number of input features was reduced.

It can be concluded that a reduction in the number of input probes resulted in a reduction of the spatial resolution of the input flow. If this affected the accuracy of stage 1, it might not have had the same effect on stage 2. In fact, stage 2 was exposed to a non-uniform inlet flow coming from stage 1 and passing through the interconnecting channel of the compressor. Consequently, fewer input signals meant sampling this nonuniformity less effectively, leading to an increase in overall accuracy. Therefore, reducing the number of probes in this case can lead to benefits in pattern recognition operation.

TCAN results. This section presents the results of the Upward and Downward tests of the TCAN model. The first test was designed to drive the compressor toward an unstable operating state (NS) from a transitional one (NP); while the second test aimed to move the compressor from the NS state to the NP one. As previously described in Sect. 4.4.3, to predict the future trends of the input time series, TCAN forecasted the next 132 time-steps, equal to two rotational periods, for all the 12 inducer monitors.

In Figure 4.12, the results obtained for the first stage of the upward test of the inducer probe RB (10 %) are reported as an example. A good match with the future reference values was obtained, demostrating the strong capabilities of the TCAN model.

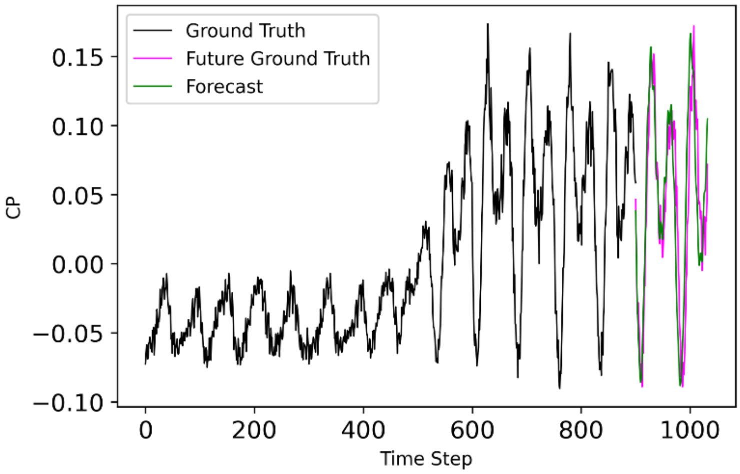

Figure 4.12. Transient signal of Upward test (adapted from [113]).

Figure 4.13 presents the results for the second Downward test. In particular, the time series associated with the inducer probe RC (10 %) is presented. The TCAN model proved to be effective in predicting the NP condition, even during a different type of transient operation of the compressor. It is important to note that the model was not originally designed to forecast NP operation data, rather it learned from the history of NS condition, which includes the NP data.

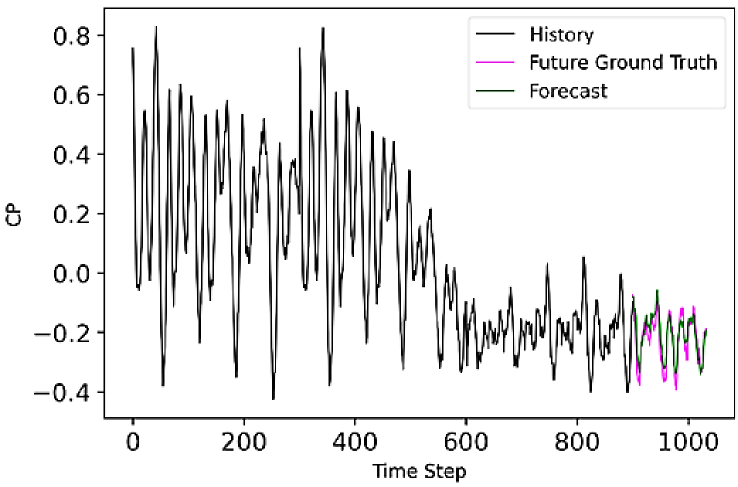

Figure 4.13. Transient signal of Downward test (adapted from [113]).

To assess accuracy, the RMSE was calculated between the forecasted time steps and the test references. As shown in the bar plot of Figure 4.14, the largest errors occurred in the Upward test for stage 2, specifically at the 90 % span. This confirmed that the signals entering the second inducer are significantly more distorted under unstable conditions, complicating the prediction process.

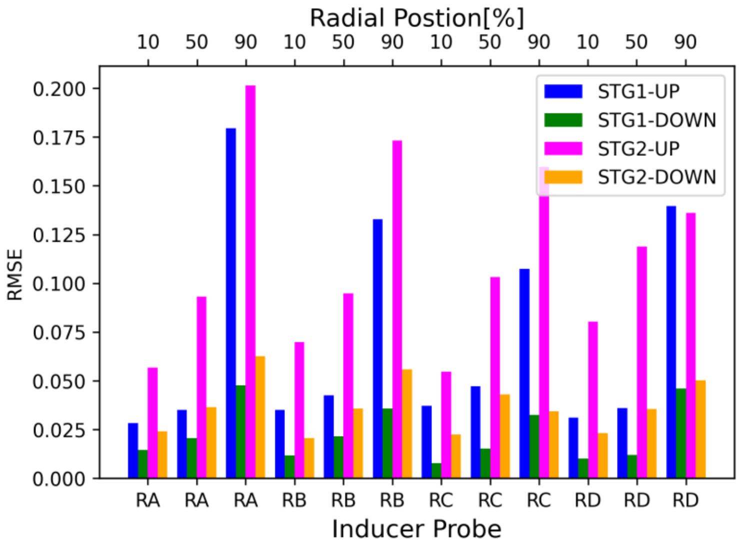

Figure 4.14. RMSEs of TCAN forecasted signals for the Upward and Downward tests (adapted from [113]).

The classification results for the ML architecture in its future-state assessment were obtained by combining the two ML models SIGMA-OPSA with the forcasting model TCAN as detailed in Figure 4.3 of Sect. 4.4. The predictive chain begins with the TCAN model forecasting the inducer behavior, followed by SIGMA mapping these results to diffuser time series. The process concludes with the OPSA model, which classifies the stability of the stage. This operation was performed on both stages and applied to all test configurations, evaluating both instantaneous and averaged signals over the two impeller rotations predicted by TCAN.

Accuracy results are presented in Table 4.3, illustrating a good match for both tests. From the istantaneous predictions, it was found that the NP condition was better classified than the NS one of the Upward test. This was probably due to the majority of the training dataset of the TCAN being from the NP condition. In addition, the averaged signals were perfectly classified and the future operational condition correctly assessed.

Table 4.3. Accuracy results for the OPSA classification given the TCAN forecast (adapted from [113]).

Accuracy	**Stage 1**		**Stage 2**	
	12 Probes		**12 probes**	
	Downward test	**Upward test**	**Downward test**	**Upward test**
Instantaneous	90.9 %	86.4 %	95.5 %	88.6 %
Averaged	100 %	100 %	100 %	100 %

4.5 Discussion of ML-based reconstruction and motivation for localisation

The results presented in the previous sections demonstrate that the integration of ML models with CFD-derived time series enables reliable detection and short-term prediction of compressor instability. By combining regression, classification and forecasting submodels, the proposed architecture reconstructed diffuser behaviour from upstream inducer measurements and classified the compressor operating state with high accuracy, even when the number of available inducer probes was reduced.

From a practical CM perspective, this represented a relevant outcome, suggesting that stability assessment can be achieved using a limited of sensors. The ability to operate with reduced upstream instrumentation supports the applicability of the approach to real compressor installations, where dense measurement layouts are often impractical.

At the same time, the adopted ML-based framework exhibited intrinsic limitations. While the reconstruction and classification steps successfully captured diffuser-dominated instability dynamics, the ML approach treated the diffuser monitoring locations as an undifferentiated set. Consequently, this lack of spatial resolution prevented a quantitative assessment of how sensitive individual measurement locations were to developing instabilities.

This limitation becomes particularly relevant when considering sensor placement and optimisation in practical monitoring systems. Identifying which diffuser locations carry the earliest and most informative signatures of instability is essential to further reduce instrumentation requirements and improve robustness.

For this reason, a complementary analysis based on ApEn is introduced in the following section.

4.6 Entropy-based localisation of diffuser instability signatures

To address the limitations identified in the previous section and to enable spatial localisation of the most sensitive probes to instability, an entropy-based signal analysis approach is introduced, using ApEn as a complexity metric, whose theoretical background is presented in Sect. 2.5.3 of Ch. 2.

In the context of compressor monitoring, ApEn provides a measure of the predictability or regularity of a time series and is therefore well-suited to capture changes in flow dynamics associated with the onset of aerodynamic instabilities. As compressor operation approaches surge, the pressure field in the compressor stages becomes increasingly irregular; a behaviour that was directly reflected in an increase in signal complexity across the monitored locations.

The application of approximate entropy to compressor instability builds on a growing body of work. Lou and Key [115] first used ApEn for compressor stall warning. Zhao et al. [116] subsequently extended the analysis from a small set of operating points to an entire speed line in both centrifugal and axial machines, observing that ApEn reaches a minimum near the optimal-incidence mid-flow region, with a characteristic decrease followed by a sharp increase immediately before surge onset. The study builds directly on the radial-compressor ApEn analysis developed earlier in this work [103].

Most closely related to the present sensor-placement objective, Zahid et al. [117] applied ApEn to spatially distributed pressure measurements in a low-speed centrifugal blower and introduced a differential form, DApEn, which responded sharply to the onset of inlet recirculation and surge while remaining insensitive to blade-passing periodicity. Their work demonstrated how entropy-based measures vary across sensor locations and how DApEn can improve temporal localization of instability onset. The entropy-based sensor-ranking approach proposed here is complementary, using spatial variations in entropy response to identify the diffuser locations most sensitive to instability development.

The second study applied the ApEn analysis on the same dataset of URANS-derived time series (see Sect. 4.2). Although the primary focus of the entropy analysis was on diffuser probes, ApEn was also computed for inducer pressure signals in order to verify its monotonic increase towards surge at these locations.

This method aimed to complement the ML-based diagnostic framework by providing a quantitative basis for sensor placement optimisation.

4.6.1 Analysis of ApEn behaviour on inducer signals

As a first exploratory step, the ApEn function was computed for each complete inducer pressure signal in order to investigate whether signal complexity exhibited a consistent trend with compressor operating condition. For this initial analysis, the embedding dimension (m) was set to 8 and the radius of similarity (r) to 20 % of the signal standard deviation.

The results indicated that ApEn values were able to provide a clear indication of the current operating point and could help construct a pseudo-performance map of the machine. Figure 4.15 presents a bar plot of the ApEn values for the four inducer signals of stage 1 at 90 % span, shown as a function of each simulated mass flow rate. For each operating condition, the mean ApEn value across the inducer probes was calculated and a linear regression was performed on the mean ApEn values.

The figure highlights a clear monotonic trend of ApEn with a decrease in mass flow, already observable at an inducer level. This analysis was intended to verify whether ApEn could reproduce a pseudo-performance behaviour of the compressor prior to its application as a spatial localisation tool.

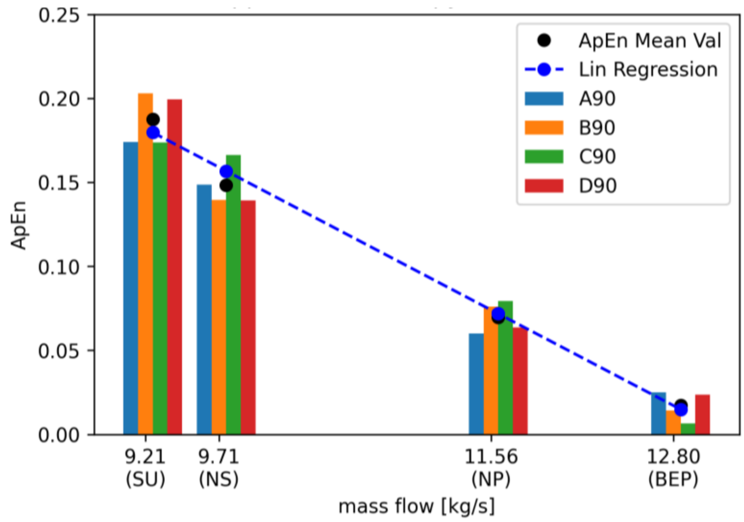

Figure 4.15. Individual ApEn values of inducer 1 pressure monitors located at 90 % span, together with a linear fit of the mean values associated with each operating condition (adapted from [103]).

4.6.2 ApEn implementation on inducer signals

As highlighted in the previous section, when moving towards surge conditions, ApEn emerged as a relevant tool for identifying the growing disorder within inducer signals. To achieve accurate performance for this specific application, the ApEn function parameters need proper calibration.

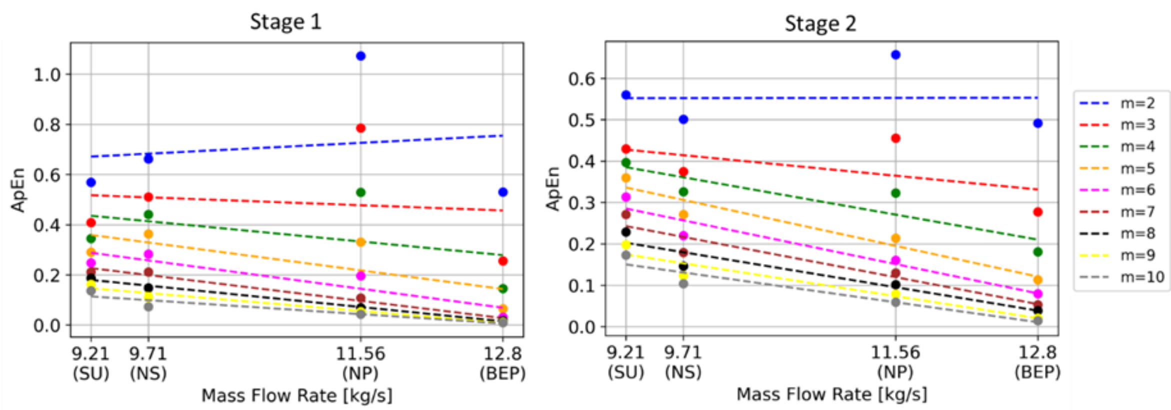

Figure 4.16. Mean inducer ApEn values of probes at 90 % of span, for all four operating points, and with different values of embedding dimension m (adapted from [103]).

First, the calibration process involved keeping the number of time steps (N) of the URANS-derived time series introduced in Sect. 4.2 fixed, together with r set to 20 % of the signal standard deviation. The factor that required crucial evaluation was the embedding dimension m.

Both stages were tested for different m values to determine which led to an ApEn performance curve that best represented the sampled signals. Figure 4.16 presents the mean

ApEn values calculated for the four signals at 90 % of the inducer span and for the four compressor operating points, along with the corresponding linear fit of each selected m.

It can be observed that higher values of embedding dimension deliver a higher negative slope of the linear fit curve, leading to a function that was more capable of differentiating among the different operating regimes of the machine. The slope of the fitting curves reached a plateau around m equal 7.

The reason for this stood in the higher dimensionality of the embedding vectors, which contained more information about the dynamic system, thus increasing the discriminatory capability of the ApEn function.

Table 4.4. Monotonic decreasing trend check and R^2 of the linear fit for different m *values, for stages 1 and 2 (adapted from [103]).*

	m	2	3	4	5	6	7	8	9	10
Stage 1	**Trend**	-	-	-	-	-	✓	✓	✓	✓
	$\mathbf{R^2}$	0.02	0.02	0.20	0.55	0.83	0.98	0.99	0.96	0.86
Stage 2	**Trend**	-	-	-	✓	✓	✓	✓	✓	✓
	$\mathbf{R^2}$	0.00	0.32	0.80	0.93	0.92	0.91	0.90	0.91	0.91

Table 4.4 summarises whether each case, presented in Figure 4.16, showed a monotonic decreasing trend of ApEn, measured at the mean data points, when transitioning from the SU condition to the BEP one at the inducer of both stages. The table also includes the R^2 parameter, which proves how well the data fits the linear regression model; the closer R^2 is to 1, the better the fit. R^2 is defined as:

$$R^2 = 1 - \frac{\sum_{q=1}^{Q}(y_q - \hat{y}_q)}{\sum_{q=1}^{Q}(y_q - \bar{y})}, \tag{4.3}$$

where Q is the number of predictions, y_q are the true values, $\bar{y}$ is the mean of the true values, while $\hat{y}_q$ are the predicted values from the linear fit model.

When calibrating m, it was searched for the values that granted a monotonic decreasing trend in mean inducer distribution, plus the ones giving the highest values of R^2. According to

Table 4.4, for stage 1, a satisfying trend was observed from m equal to 7, while for stage 2, this was verified from m equal to 5. To meet both requests on the best embedding dimension, a value of 8 was chosen for the first stage and, analogously, an embedding dimension of 5 was determined to be the optimal solution for the second stage.

It is important to note that the value of m depends heavily on N, thus time series of varying lengths may require a different embedding dimension to maintain the same window size.

After calibrating the appropriate embedding dimensions for each stage, a sensitivity analysis on r was conducted to identify the optimal percentage of the signal standard deviation. The results are presented in Table 4.5, showing whether the monotonic trend in ApEn is satisfied and the corresponding R^2 parameter values for each stage.

Consequently, for stage 1, r was chosen to be 20 % of the signal standard deviation, as this value was the only one to fulfil the trend condition and had the maximum R^2 of the linear fit. For stage 2, the procedure was repeated and it was found that r equal to 10 % resulted to be the optimal percentage.

Table 4.5. Monotonic decreasing trend check and R^2 of the linear fit for different r values, for stages 1 and 2 (adapted from [103]).

	r	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Stage 1	**Trend**	-	✓	-	-	-	-	-	-	-	-
	$\mathbf{R^2}$	0.39	0.99	0.77	0.39	0.12	0.00	0.07	0.32	0.61	0.77
Stage 2	**Trend**	✓	-	-	-	-	-	-	-	-	-
	$\mathbf{R^2}$	0.97	0.93	0.72	0.04	0.426	0.71	0.81	0.88	0.91	0.87

Figure 4.17 summarises the final linear fits for both stages, illustrating the calibrated increase in ApEn as the inducer moves toward the surge condition of the compressor.

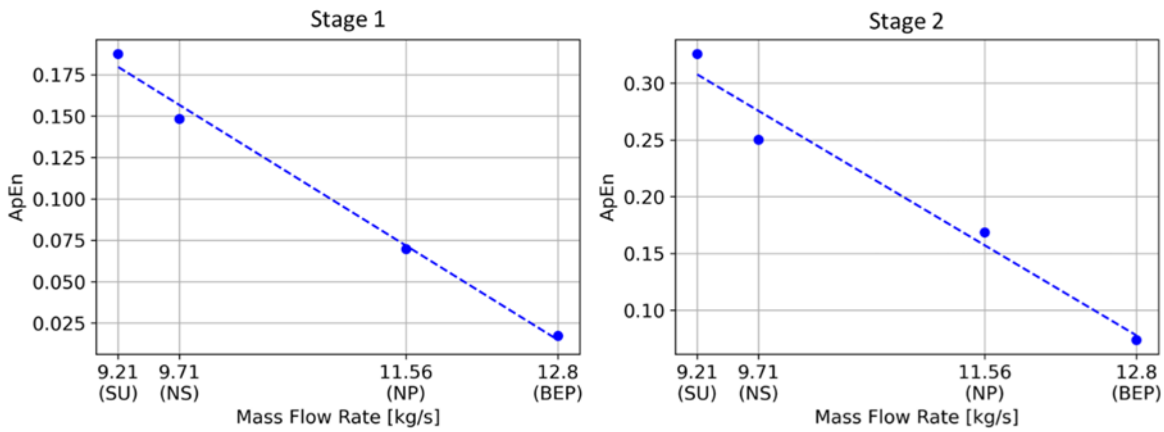

Figure 4.17. Final linear fit with m=8, r=20 % for stage 1 and m=5, r=10 % for stage 2 (adapted from [103]).

4.6.3 ApEn implementation on diffuser signals

The findings of Sect. 4.6.2 suggest that computing ApEn for diffuser signals could help to identify which diffuser locations carry the earliest and most informative signatures of instability. Previous frequency analysis [110], identified that the near surge (NS) and surge

(SU) operational states are characterised by two low frequencies that arise in the diffuser monitors due to the onset of aerodynamic instability.

The two frequencies are linked to rotating stall cells with high thermodynamic entropy that develop in the stage volute [110]. While the first 50 Hz signal was related to the cells themselves, the second 200 Hz frequency captured their interaction with the volute tongue. Based on these findings, a proper calibration of the embedding dimension and of the radius of similarity was critical to properly identify the most sensitive monitors to compressor surge.

According to Takens' theorem [102], the analysed phenomenon must repeat itself at least twice in the time series window defined for ApEn calculation. For instance, if the studied mechanism has frequency $f_{phenomenon}$ [Hz], N time steps and a sampling interval Δt [s], the right embedding dimension is computed by:

$$m = \frac{\Delta t * N}{\frac{1}{f_{phenomenon}} * 2} \tag{4.4}$$

Consequently, for a frequency of 50 Hz, given the diffuser time series introduced in Sec. 4.4.1, the right m results to be 3. If the focus goes on the mechanisms of 200 Hz, then the right embedding dimension is 11.

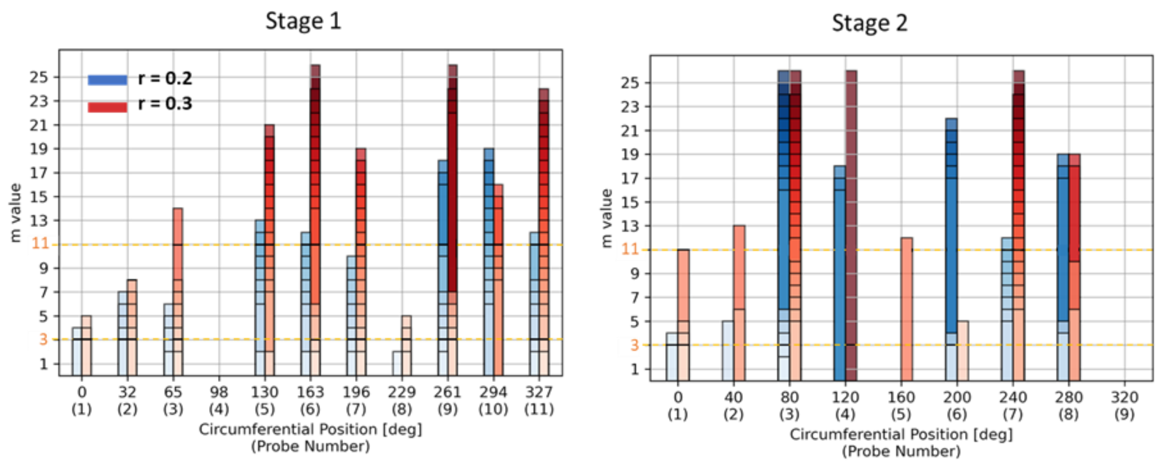

Figure 4.18. Stages 1 and 2 diffuser monitors (90 %) that exhibit a monotonic increasing trend in ApEn with m (adapted from [103]).

Figure 4.18 shows bar plots illustrating m values for two different selections of r and for the probes at 90 % of the diffuser exit of stages 1 and 2, respectively. Based on the findings of Sect. 4.4.7, only the 90 % time series were considered because they exhibited clearly defined instability patterns. The bars specifically mark if the selected probe presents a monotonic decreasing trend in ApEn, moving from SU to BEP. With regard to an appropriate selection of r, the goal was to find the probes that covered both low and high-frequency phenomena, that is, both low and high values of m.

For stage 1, an r value of 0.3 was preferred, as it allowed for higher values of m. Next, upon closer inspection of the first target value (m = 3), monitors 1, 2, 3, 6, 7, 8, 9 and 11 consistently exhibited a monotonic trend. These results were in line with [110], indicating that these locations were already sensitive to the rotating cell pass. Moreover, for the second target value (m = 11) corresponding to a frequency of 200 Hz, probes 3, 5, 6, 7, 10 and 11 presented a positive trend. Based on these findings, probes 7 and 11, positioned 131° apart, were chosen for their ability to capture the selected instability frequencies and provide consistent monitoring throughout the spectrum.

For stage 2, a value of 0.3 was again selected for r. Within this similarity radius, probes 1 and 7 independently identified the frequencies of the analysed instability phenomena. Although these probes are 120° apart and probe 9 is physically closer to the volute tongue, probe 7 appears to be the nearest sensor capable of detecting the 200 Hz phenomenon.

This discrepancy is likely attributable to two factors: the reduced probe count in diffuser 2 compared to diffuser 1, which limits spatial resolution, or the fact that the inflow entering stage 2 was already perturbed by the upstream influence of stage 1.

The preceding analysis suggests that ApEn may serve as a useful tool for informing probe placement in experimental monitoring. Based on these findings, positioning one diffuser probe near the volute tongue, together with a second one approximately 120°-130° apart, could potentially improve the early detection of aerodynamic instabilities. Such a configuration could enable timely intervention by providing an accurate diagnosis of developing instabilities.

4.7 Chapter summary

This chapter addressed the problem of CM and early detection of aerodynamic instability in radial compressors**,** with the objective of supporting maintenance strategies based on unsteady pressure measurements. Compressor surge and related instabilities represent critical failure modes from both an operational and maintenance perspective, thus motivating the development of monitoring approaches capable of detecting degradation and instability onset before the occurrence of severe performance loss or mechanical damage.

Within this context, a unified dataset of URANS-derived time series was employed to develop complementary diagnostic methodologies under controlled and repeatable conditions. A ML-based approach was first developed to reconstruct diffuser pressure dynamics from upstream inducer measurements and to classify the compressor operating state with respect to instability. Through the integration of regression, classification and forecasting into a single architecture, the analysis demonstrated that upstream signals alone are reliable for present-state assessment and short-term forecasting. From a CM perspective, this represents a key advantage, as inducer measurements are generally more accessible and practical to implement than downstream instrumentation, particularly when monitoring locations are constrained.

Building on the ML-based approach, an entropy-based analysis was subsequently introduced to further resolve the spatial sensitivity within the diffuser. ApEn was selected as a diagnostic metric to quantify signal complexity and to localise the diffuser regions most sensitive to instability-related dynamics. An initial analysis confirmed the monotonic evolution of entropy with the compressor operating condition at the inducer level, while the subsequent localisation study identified the most sensitive monitoring locations as those associated with the dominant instability mechanisms.

Overall, the results of this chapter demonstrate how combining data-driven techniques with entropy-based localization can lead to smarter monitoring layouts for radial compressors. Through these approaches, monitoring systems can be designed based on the actual flow behaviour of the machine, providing a foundation for efficient, reduced-order tools that maintain high diagnostic accuracy with fewer sensors.

While the compressor studies provide a controlled numerical environment to develop and validate monitoring methodologies based on CFD and ML, the following Ch. 5 extends these concepts to wind turbines operating under real atmospheric conditions. In this context, DT architectures are introduced to integrate CFD-informed models with experimental data for aerodynamic fault detection.

5. Digital Twin-based monitoring of wind turbine aerodynamic performance

5.1 Purpose and scope of the chapter

Wind turbines represent a particularly challenging class of assets for CM and diagnostics. Unlike enclosed turbomachinery, which operates under comparatively controlled boundary conditions, wind turbines are exposed to stochastic inflow, atmospheric turbulence and rapid variations in the operating point. Despite the availability of multiple sensors, aerodynamic states and degradation mechanisms are typically indirectly observable, as monitoring relies mainly on control-oriented signals and global performance indicators. These characteristics restrict the direct application of conventional CM techniques and make the interpretation of performance deviations challenging.

Within this context, Digital Twins (DTs) have gained increasing attention for integrating physical modelling, operational data and diagnostic logic into a unified architecture. For wind turbines, effective DT implementations must be resilient to measurement uncertainty, inflow variability and incomplete observability, while still providing actionable information for diagnostics and maintenance decision-making.

The purpose of this chapter is to present the development of a DT-based monitoring methodology for wind turbines, consolidating the results of two works by the author into a coherent and fault-driven diagnostic framework [118,119], one currently available as a revised preprint and the other accepted for publication following peer review. The chapter addresses two distinct aerodynamic fault mechanisms of wind turbines: yaw misalignment induced by faulty wind-direction sensing and blade surface degradation associated with leading-edge erosion (LEE).

Compared to the diagnostic models developed in Ch. 4, the DTs presented here are built on an existing asset operating in real atmospheric conditions. While both approaches are grounded in CFD-informed modelling, the architectures developed in this chapter explicitly integrate high-fidelity aerodynamic simulations with operational measurements to enable diagnostic inference in a real-world environment.

An initial DT implementation is first introduced to address yaw misalignment, focusing on short-term performance deviations and corrective diagnostic actions. This initial DT was designed to operate under conditions of limited observability, inferring persistent static yaw misalignment through quasi-steady inversion of CFD-derived performance surfaces based on time-averaged operational measurements.

The methodological experience gained from the first DT subsequently motivated the refinement and formalisation of a second architecture, tailored for monitoring and diagnosing early aerodynamic degradation associated with LEE. In wind turbines, blade surface degradation manifests over longer time scales and requires a monitoring approach capable of supporting maintenance-oriented assessment rather than immediate corrective action.

Accordingly, the DTs presented in this chapter evolve from a first implementation based on direct CFD-informed inference for yaw misalignment detection to a second hybrid formulation combining CFD references and data-driven models for early LEE monitoring.

Despite differences in complexity and internal organisation, both share a common architecture rooted in performance-based monitoring and CFD-informed interpretation. This progression from corrective control-state inference to degradation-oriented health assessment illustrates the scalability of the proposed DT architecture across fault classes with different temporal and operational characteristics.

In this way, the chapter establishes a continuous methodological development in which DTs are employed as practical instruments for performance-based CM rather than as isolated numerical constructs. By addressing both corrective diagnostics and degradation monitoring within a coherent CFD-informed framework, the chapter highlights the role of DTs as enabling tools for interpreting aerodynamic deviations, providing a methodological basis for supporting maintenance-oriented decision processes in wind energy systems.

5.2 Digital Twin architecture for aerodynamic fault monitoring

The DTs presented in this chapter are built upon a common architectural foundation, developed to enable aerodynamic fault monitoring on a small-scale horizontal-axis wind turbine (HAWT) operating under real atmospheric conditions. Although the subsequent sections address distinct fault classes, static yaw misalignment and LEE, both models are instantiations of the same CFD-informed monitoring architecture.

The following sections present the proposed DT architecture, structured into four interacting layers, as proposed in recent DT literature [120,121]: virtual, physical, data, and intelligent. This layered formulation provides a consistent division that separates sensing, modelling and inference functions while preserving their dynamic coupling.

5.2.1 Virtual layer

The virtual layer constituted the physical base of the CFD-informed DTs. It consisted of precomputed steady-state CFD simulations generating aerodynamic performance surfaces parameterised by the relevant fault variable.

In the yaw misalignment case, the performance space included the yaw misalignment angle (γ) as an additional dimension. In the LEE case, the parameterisation was expressed through equivalent sand-grain roughness height (k_s). Despite these differences, both implementations shared the same modelling principle:

1. Generate a structured CFD performance dataset across a discretised aerodynamic domain, focusing on the aerodynamic power coefficient C_p^{CFD} defined as follows:

$$C_p^{CFD} = \frac{P_{aero}}{0.5\rho A U_\infty^3}, \tag{5.1}$$

where P_{aero} is the aerodynamic power obtained through CFD simulations at the prescribed uniform inlet velocity U_∞, ρ is the air density and A is the area of the rotor disc.

2. Construct interpolable response surfaces relating operating conditions to aerodynamic performance.
3. Use these surfaces as reference maps for aerodynamic state inference.

The virtual model did not simulate the turbine continuously. Instead, it provided a calibrated performance manifold against which measured behaviour could be compared. The DT thus functioned as a performance-consistency evaluator, where deviations between measured and CFD-informed predictions formed the basis of diagnostic inference.

5.2.2 Physical layer

The physical layer consisted of the wind turbine asset and its associated sensor network. The selected machine was a 1.3 m diameter HAWT (Pikasola 400W-12V) instrumented to measure electrical, mechanical and environmental quantities. Further details are provided in Sect. 5.3.2. The turbine was operated under outdoor conditions characterised by variable wind speed, direction and turbulence intensity. This environmental variability was not treated as noise but as an intrinsic part of the monitored system.

5.2.3 Data layer

The data layer handled acquisition, storage, filtering and transformation of sensor signals. Measurements were sampled at 1 Hz and streamed to a structured database. From the raw signals, derived performance parameters were computed, namely the measured power coefficient:

$$C_p^{meas} = \frac{P_{\mathrm{el}}}{0.5\rho A U_\infty^3}, \tag{5.2}$$

together with the tip-speed ratio:

$$TSR = \frac{\omega R}{U_\infty}, \tag{5.3}$$

where P_{el} is the electrical power extracted by the generator, U_∞ is the measured wind speed by the anemomether, ω is the registered angular speed of the rotor and R its radius.

The C_p^{meas} quantified turbine performance derived from field measurements of electrical power output, thereby reflecting the combined aerodynamic and electromechanical behaviour of the system, while serving as the experimental counterpart to the virtual layer's predicted aerodynamic power coefficient C_p^{CFD}.

By design, the DT adopted a quasi-steady aerodynamic interpretation. Individual 1 Hz measurements were treated as instantaneous operating points within the CFD response space, while diagnostic outcomes were derived from time-aggregated datasets rather than single samples. This temporal aggregation was critical for filtering inflow fluctuations and rotor dynamic effects, effectively smoothing the high-frequency stochastic noise characteristic of real atmospheric conditions.

5.2.4 Intelligent layer

The intelligent layer performed fault inference by interpreting discrepancies between the measured aerodynamic state and the virtual performance baseline. Although its internal logic was fault-dependent, its architectural role remained consistent across both case studies: to transform CFD-inferred information into actionable feedback in order to close the information loop of the DT.

For control-state faults such as static yaw misalignment, the intelligent layer performed inverse interpolation within the CFD-derived performance space to estimate the misalignment angle that best explained the observed operating condition. The inferred deviation was then used to generate a corrective feedback action, restoring the turbine to an aerodynamically consistent yaw alignment.

For degradation faults such as LEE, the intelligent layer extracted residual-based features and applied a supervised classifier to distinguish between clean and degraded states. From probabilistic outputs of the classifier, a scalar health indicator was derived. This indicator did not directly modify control variables; instead, it provided maintenance-oriented feedback, enabling condition-based decision logic and supporting timely intervention before significant performance losses occurred.

In both cases, the inference mechanism closed the information loop between physics and measurements. However, the nature of the feedback was fault-adaptive, moving from immediate control correction in the yaw case to a health-state quantification with maintenance guidance in the LEE case. This distinction highlights the adaptability of the proposed DT architecture to different classes of aerodynamic faults while preserving a unified inferential structure.

5.3 First DT: yaw misalignment detection and correction

The DT introduced in the following sections addresses static yaw misalignment induced by faulty wind-direction readings through a performance-based inference supported by extensive CFD analysis. This DT was formulated as a functional model, wherein the virtual aerodynamic behaviour served to interpret measured performance deviations and diagnose the presence of misalignment. The primary objective was short-term diagnostic assessment and immediate corrective action, distinguishing it from long-term degradation tracking or maintenance scheduling.

While the layered architecture was not formally articulated at the time of the first study, it emerged implicitly through the integration of experimental measurements, CFD-derived performance maps and inverse interpolation logic. The subsequent LEE study builds upon this methodological foundation, formalising the architecture and extending the same logic toward degradation-oriented fault detection and probabilistic health assessment.

5.3.1 Yaw misalignment: physical background and fault description

Yaw misalignment represents a common and impactful fault affecting HAWT operation. As shown in Figure 5.1, it occurs when the rotor axis is not aligned with the incoming wind direction, leading to a reduction in the effective aerodynamic loading of the rotor and, consequently, to a power production loss. Even relatively small yaw errors can result in measurable efficiency penalties, making yaw alignment a critical aspect of wind turbine performance monitoring [122–124].

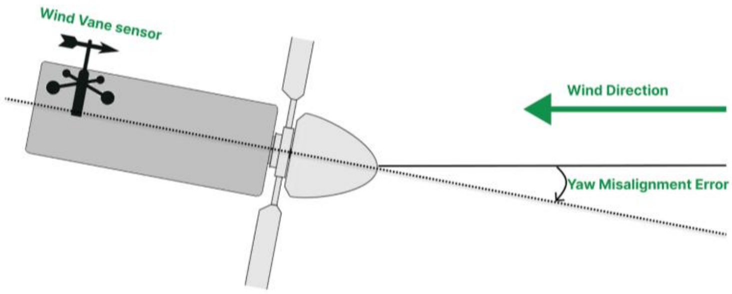

Figure 5.1. Schematic representation of yaw misalignment in an HAWT.

Yaw misalignment is generally categorised into two types, namely static and dynamic. Static yaw misalignment refers to a constant offset between the nacelle position and the actual wind direction, typically arising from problems associated with the wind vane sensor, such as exposure to harsh environmental conditions, miscalibration or software faults.

Dynamic yaw misalignment is primarily caused by the delayed response of the yaw control system to rapidly changing wind directions, resulting in transient misalignments that induce unsteady aerodynamic loads, power fluctuations and additional fatigue stresses on the turbine structure. While static and dynamic misalignments often occur together in real-world operations, this study specifically focused on the impact of static yaw misalignment.

From a CM perspective, yaw misalignment primarily affects short-term aerodynamic performance, manifesting through deviations in power production and remaining, in principle, reversible through corrective control actions. These fault characteristics make yaw misalignment an ideal candidate for the initial development of a DT-based monitoring and corrective approach, aimed at assessing feasibility under conditions of limited observability and high inflow variability.

5.3.2 Wind turbine system

The Pikasola 400W-12V Wind Turbine Generator, a fixed-pitch small-scale HAWT, served as the experimental system for developing and validating the proposed DT-based condition monitoring approach (Figure 5.2). The turbine is originally passively yawed by the wind thanks to its tail, while rotor speed is purely wind-driven. To ensure structural safety, an electromagnetic braking system within the turbine controller provides overspeed protection, triggering automatically upon reaching predefined electrical limits (15 V or 30 A).

It features a rotor of diameter 1.3 m, and it is characterised by a rated power of 400 W generated at 12 V DC. The rated rotational speed of 800 min^{-1} is reached for an incoming wind speed of 13 m s^{-1}, whereas power starts to be delivered for a cut-in speed of 2.5 m s^{-1}. For this first DT application, due to site constraints and practical portability, the generator was installed on a 2.0 m long pole, which was equipped with a dedicated adjustable basement frame.

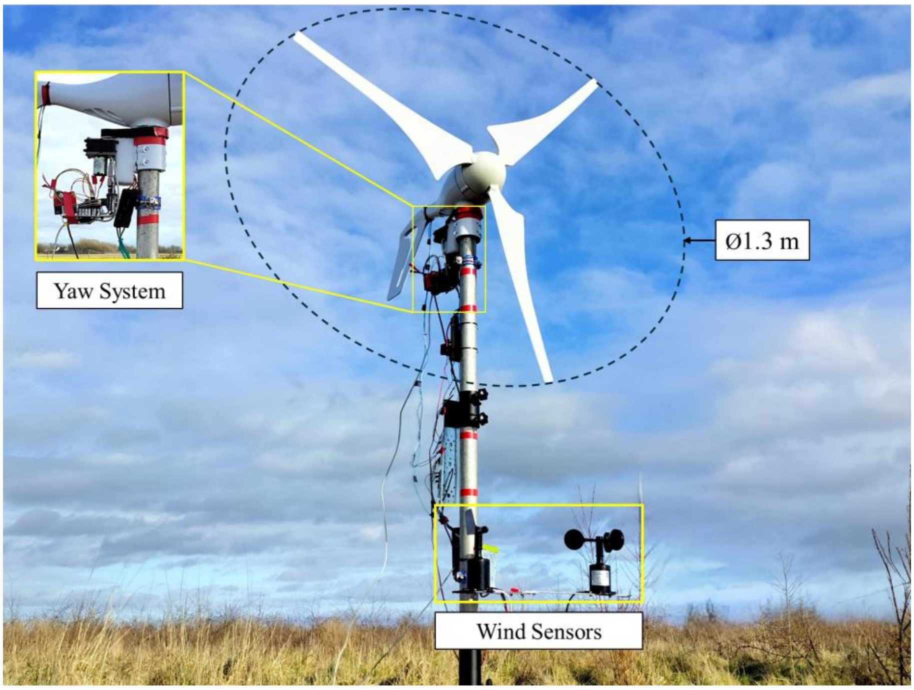

Figure 5.2. Small-scale wind turbine (Pikasola 400W-12V, 1.3 m rotor diameter) on a 2.0 m pole with dedicated base during outdoor testing at Cranfield University airfield, UK (adapted from [118]).

The turbine was tested outdoors at the airfield of Cranfield University, Bedfordshire, United Kingdom, located at a latitude of 52° 03' 59.5" N and longitude of 0° 37' 34.0" W. The tests were conducted during January and February 2025. Wind statistics were evaluated over 10-minute averaging windows. During the campaign, 10-minute mean wind speeds ranged between 5.37 and 7.40 m s^{-1} (overall average 6.24 m s^{-1}), with gusts reaching 17 m s^{-1}. The corresponding turbulence intensity ranged between 31 % and 47 %, with an average of 38 %. A Weibull fit to the full wind speed dataset yielded a shape parameter k = 2.60 and scale parameter c = 6.91 m s^{-1}.

The proposed DT-based method was assessed via a three-trial outdoor campaign, conducted under natural wind fluctuations. Each trial was tailored to a distinct development objective: Trial I focused on real-time data acquisition and inflow characterisation; Trial II tested the turbine in a correctly yawed state to perform numerical-experimental power matching with CFD; and Trial III validated the DT's capacity for yaw misalignment detection and fault correction.

5.3.3 Sensor network and data flow

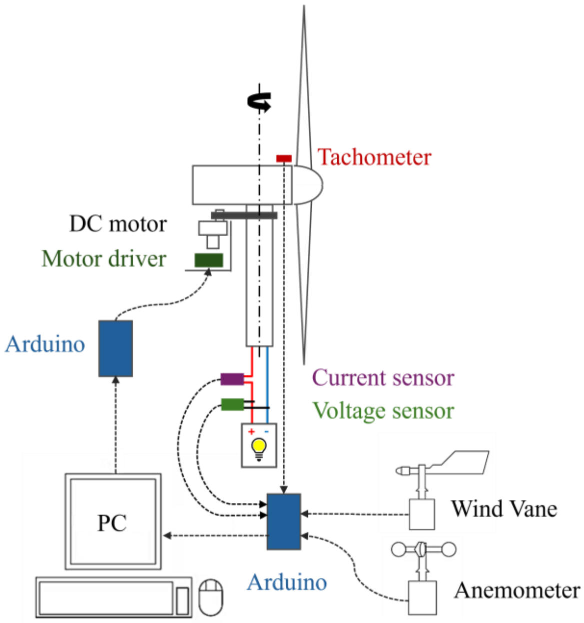

Figure 5.3. Schematic of the sensor network and data acquisition system for the instrumented HAWT (adapted from [118]).

The experimental setup included a custom-built sensor network installed on the asset to enable real-time monitoring. The entire network was designed to be compact, allowing for outdoor deployment in varying locations without the need for fixed infrastructure. This flexibility enabled opportune repositioning of the turbine in open field conditions, facilitating experimental tests under naturally fluctuating wind conditions. The following sensors were used to sample data through an Arduino Yun microcontroller:

- Wind vane: RS-FXJT-V05-360 model.
- Anemometer: RS-FSJT-NPN model.
- Tachometer: A3144 from Allegro.
- Current sensor: ACS712ELCTR-30A.

Data from all sensors was collected at a frequency of 1 Hz and stored in a cloud MongoDB database [125] via a custom Python script. The adopted sampling frequency was chosen according to the recommendations from the IEC 61400-12 standard for wind measurement systems, which suggests 1 Hz for proper accuracy [126]. An overview of the selected sensors and the resulting data flow is presented in the following schematic of Figure 5.3.

As introduced in Figure 5.2, the anemometer and wind vane were located at the same elevation (0.65 m above ground) to ensure they sampled a consistent local inflow; both were placed 0.3 m ahead of the rotor plane to minimise aerodynamic interference from the blades. A 0.4 m separation between the sensors ensured negligible mutual interference, with the configuration's suitability validated through CFD postprocessing.

While the sensor arrangement was nonstandard relative to IEC-specified meteorological tower spacing [126] and the vertical shear between 0.65 m and the 2.0 m hub height was not explicitly measured, the setup was sufficient for local flow characterisation. It enabled necessary yaw corrections within the specific constraints of small-scale, portable wind turbine testing, effectively capturing the flow disturbance experienced by the DT rather than providing a formal IEC-compliant hub-height measurement. Any systematic bias in absolute wind speed magnitude arising from vertical shear was absorbed through the numerical-experimental power matching introduced in Sect. 5.3.8.

The A3144 Hall effect sensor was used as a tachometer to measure rotor speed through three magnets attached behind each blade's root; the sensor was positioned right after the rotor to easily detect a full revolution. A custom Arduino routine processed the pulses to calculate the rotational speed. To measure the wind generator voltage, a voltage divider was implemented to scale the maximum possible voltage (12 V) to 5 V, compatible with the Arduino microcontroller.

Electrical loading was provided by a 12 V (55 W) halogen lamp wired downstream of the rectification stage, establishing an approximately resistive load profile. This arrangement did not impose active rotor speed regulation; instead, the rotational speed was wind-driven and governed by the balance between aerodynamic torque, generator electromagnetic resistance and mechanical losses. Consequently, fluctuations in rotational velocity and power generation were purely a function of the inflow conditions and the turbine's aerodynamic response. DC power was calculated by multiplying the current measured by the ACS712 current sensor and the voltage read through the voltage divider.

5.3.4 Yaw control system

A custom yaw control system was developed to adjust and correct the turbine's yaw angle. Yaw actuation was built using a Crouzet 82840 (Gear Ratio = 750) brush 24 V DC motor connected to the nacelle via a belt drive and controlled using an Arduino Uno microcontroller together with the BTS7960 motor driver module (see Figure 5.2 Figure 5.3). The system was operated via a Python interface, which allowed for careful actuation by inputting the requested angular displacement and the direction of movement.

The motor shaft was fitted with a 35 mm-diameter pulley that, via a 5 mm-pitch belt, drove a 75 mm-diameter pulley mounted on the nacelle shaft. The pulley axes were spaced 80 mm apart. This design ensured sufficient robustness to overcome gyroscopic forces from rotor rotation and system friction. The system was manually calibrated to ensure accurate yaw actuations with a yaw speed of 1° s^{-1}. Calibration was performed by tracing visible markers around the nacelle shaft, allowing angle adjustments to be verified against known reference positions.

5.3.5 Definition of angular quantities and sign conventions

To ensure consistency across experimental, CFD, and DT frameworks, all angular quantities used a shared ground-based reference. The wind direction (θ_w) was measured as a clockwise horizontal inflow angle in the interval [0°, 360°), while the nacelle yaw (θ_y) defined the absolute rotor orientation. During testing, yaw motion was constrained to ±50° about the 180° reference. The yaw misalignment angle γ was defined as the signed difference:

$$\gamma = \theta_w - \theta_y. \tag{5.4}$$

A positive yaw misalignment ($\gamma > 0°$) indicates the wind is clockwise relative to the rotor axis (approaching from the right when looking downstream), while ($\gamma < 0°$) denotes a counterclockwise deviation. Under aligned conditions, $\gamma = 0°$.

Crucially, γ was a derived signed difference rather than an independent variable, as both wind direction and nacelle yaw were defined as absolute orientations within a shared reference frame. This convention was maintained across all CFD simulations to ensure consistency between numerical and experimental results.

Finally, corrective yaw actuation was always applied with opposite sign to the inferred γ to restore alignment ($\gamma \rightarrow 0°$).

5.3.6 CFD modelling

Geometry and domain. The machine geometry was developed as a computer-aided design (CAD) model. In particular, the blade geometry was extracted from a previous model already developed by Utah Valley University [127], since previous research had already been conducted on the same specific turbine; the nose and the nacelle of the wind turbine generator were designed by reverse engineering from the physical asset. Simplifications included not modelling the wind turbine's pole and tail, but only the rotor and its nacelle.

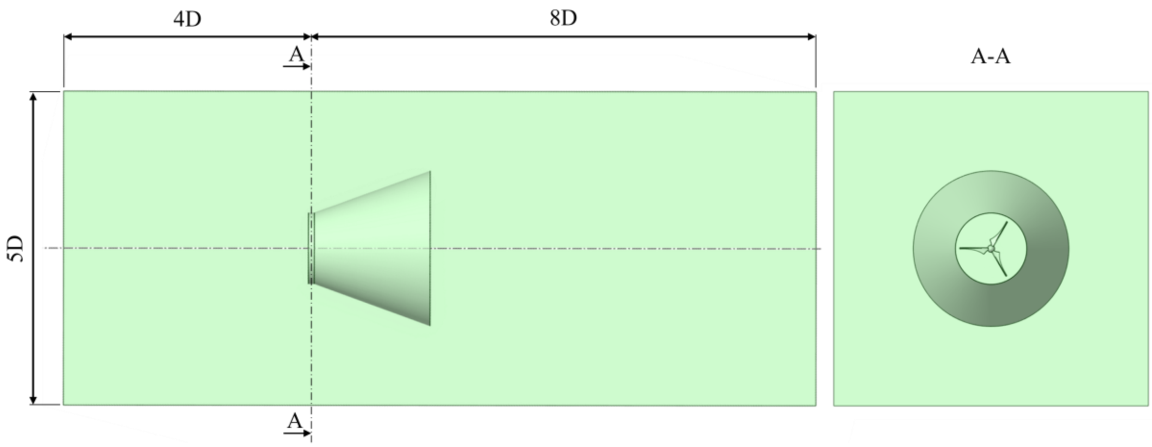

Figure 5.4. Overview of the CFD domain around the wind turbine rotor (adapted from [118]).

As presented in Figure 5.4, the fluid domain consisted of a box space around the full rotor of size 5Dx5Dx12D, where D is the turbine diameter. This setup results in a blockage ratio of approximately 3.1 % (A_{rotor} / A_{cross} = $\pi D^2/4$ / $25D^2$ = $\pi/100$), which is sufficiently low to limit confinement effects in external rotor simulations. The rotor was positioned within a disk subdomain 4 diameters downstream of the inlet. This was followed by a conical subdomain with a locally denser mesh to capture the near-wake flow structures and the onset of wake formation. The resulting computational mesh was designed to be polyhedral and unstructured, consisting of approximately 31 million elements, with a blade first layer thickness of 1.22e-5 m that granted an averaged y^+ value of about 1 for the analysed operations. Although a systematic mesh independence study was not conducted, the mesh resolution was chosen based on prior CFD experience from similar applications [128,129], ensuring precise near-wall treatment and stable convergence. This domain sizing was aligned with established literature [130,131], to accurately capture the flow field around the rotor while minimising computational timing and cost.

Simulation settings. Simulations were performed using the software Ansys Fluent 2024R1 on the Delta2 High-Performance Computing (HPC) cluster of Cranfield University. The Shear Stress Transport (SST) *k-ω* turbulence model was employed to compute the flow behaviour around the turbine under steady-state conditions. The choice of this model was due to its capacity to predict power output and aerodynamic loads more accurately than other RANS turbulence models, as commonly emphasised in literature [132–134].

The turbine operated within a low-Reynolds regime ($Re \approx 10^5$), calculated based on rated conditions (800 min^{-1}, 13 m s^{-1}) and chord dimensions ranging from 0.027 m to 0.135 m. Because utility-scale HAWTs typically work at much higher Reynolds numbers ($Re \approx 10^6$ -

10^7) [135,136]. While the underlying DT logic remains valid across scales, a direct transfer of yaw-response data to larger rotors is not feasible without re-deriving the CFD database to ensure accuracy at different Reynolds regimes.

The rotation of the rotor was modelled through the multiple reference frame method (MRF) [111], which was applied to both the disk and conical subdomains. These rotating regions were coupled to the surrounding static fluid domain via a frozen rotor interface, allowing for a steady-state representation of the rotor's influence on the flow field at a fixed angular position.

Boundary conditions included a uniform inlet velocity with a turbulence intensity of 5 %. The simulated wind speeds were U_∞ = 3, 5, 8, and 13 m s^{-1}, in order to cover an incoming inflow that ranged from the cut-in speed to the rated speed of the selected wind turbine. Static yaw misalignment was reproduced by modifying the inlet flow angle relative to the turbine's rotor axis by a constant yaw misalignment angle, γ (see Eq. (5.4) of Sect. 5.3.5). When simulating yaw misalignment states, two faces of the box-shaped domain were set as inlets, whereas the opposing two downstream faces were set as pressure outlets. The top and bottom walls were configured as free-slip walls.

No vertical wind-speed measurements from ground to blade tip were available for CFD calibration. Given the significant spatial and temporal variability of the inflow conditions during the experimental campaigns, a uniform inlet velocity profile was adopted as a simplified and consistent boundary condition. The CFD results were then calibrated a posteriori through comparison with experimental measurements.

A common theoretical assumption is that yaw misalignment results in a symmetric power loss for positive and negative yaw misalignment angles [137]. After verifying this hypothesis on power production, only positive static yaw misalignment angles were tested, specifically at γ = 5°, 10°, and 15°.

Design of experiments. CFD simulations were performed across the selected range of yaw misalignment angles and wind speeds. For each combination, the rotor's rotational velocity was adjusted to achieve the target TSR, ranging from 2 to 8 [131,137]. To isolate the aerodynamic effects of flow misalignment, for a given combination of wind speed U_∞ and TSR, the rotational velocity was held constant across all analysed yaw angles. The aerodynamic power of the rotor was computed as the product of the converged rotor torque and the imposed rotational speed corresponding to the analysed operating point, and used to compute the aerodynamic power coefficient C_p^{CFD}, see Eq. (5.1). A summary of the CFD design of experiments adopted is reported in Table 5.1.

It is acknowledged that the outdoor experimental campaign was affected by atmospheric phenomena such as wind shear and gusts, which are not explicitly resolved within the present steady RANS framework. In the CFD simulations, turbulence is modelled rather than resolved, with the inlet turbulence intensity prescribed via turbulent kinetic energy, resulting in an idealised inflow condition.

Table 5.1. Summary of the design of experiments for the CFD simulations (adapted from [118]).

Parameter	**Value**
Yaw misalignment angle (positive), γ [°]	0, 5, 10, 15
Inlet wind speed, U_∞ [m s^{-1}]	3, 5, 8, 13
Tip-speed ratio range, TSR [-]	2 - 8
Simulations per yaw misalignment case	28
Total CFD simulations	112

This choice was adopted intentionally to isolate the aerodynamic effects of yaw misalignment under controlled and repeatable conditions. The connection between the idealised CFD framework and real outdoor operation is established through the experimental power matching procedure described in Sect. 5.3.8, where CFD-derived performance surfaces are anchored to measured operating data.

5.3.7 Estimation of yaw misalignment from CFD-derived performance data

The CFD results were used to populate a three-dimensional performance space, mapping the C_p^{CFD} against TSR and yaw misalignment angle γ. In addition, for inlet wind speeds of 8 m s^{-1} and 13 m s^{-1}, the (C_p^{CFD}, TSR) distributions demonstrated nearly identical behaviour across all simulated values of γ, with a relative percentage error below 5 % due to aerodynamic similarity. Therefore, it was considered acceptable to select a single averaged C_p^{CFD} dataset to characterise the turbine's performance for the 8 to 13 m s^{-1} range.

Due to computational limitations related to the extensive time required for each high-fidelity simulation, simulating the full range of yaw and TSR combinations across additional wind speeds were prohibitively expensive in terms of both CPU hours and storage. Hence, linear interpolation among the selected wind speeds was considered sufficient to generate (C_p^{CFD}, TSR, γ) data at intermediate wind speeds (U_∞ = 4, 6, 7 m s^{-1}). This approach was justified by the monotonic and smooth evolution of the non-dimensional aerodynamic coefficients within the analysed Reynolds number regime and by the fact that the CFD database was intended to represent a structured performance space capturing global aerodynamic behaviour rather than detailed local flow features.

The CFD-derived performance space served as the foundation for estimating and inferring the yaw misalignment angle $\gamma = f(C_p^{CFD}, TSR, U_\infty)$ as part of the virtual object of the DT model. Specifically, for each reference inflow condition U_∞, the corresponding (C_p^{CFD}, TSR) input space was isolated. Six Clough-Tocher piecewise cubic interpolants were then constructed: five associated with the discrete wind speeds U_∞ = 3, 4, 5, 6, 7 m s^{-1}, and one representing the averaged dataset for the 8-13 m s^{-1} range. The generally monotonic and smooth evolution of the interpolants, particularly within the operational TSR range, ensured a well-posed inversion, yielding a unique solution for γ.

Yaw inference accuracy was assessed through a local perturbation analysis. Stochastic perturbations, with standard deviations $\sigma_{C_p^{CFD}}$ = 0.005 and σ_{TSR} = 0.05, were applied to CFD reference points to represent the residual variability of 30 s aggregated measurements, and the inverse interpolants were queried accordingly. The resulting median absolute yaw inference error remained below 2° across all defined wind speeds.

During experimental testing, the magnitude of γ and its sign were determined through direct measurements of C_p^{meas} and TSR, following a numerical-experimental power matching process. This ensured consistency between measured data and CFD-derived performance metrics, as detailed in the following Sect. 5.3.8. This matching of measured and CFD-derived performance to pin down the fault parameter follows the same principle as the combined CFD-SCADA framework of Castellani et al. [138], who treated each turbine of a wind farm as a virtual meteorological mast and isolated a systematic anemometer bias from the mismatch between RANS-simulated and measured wind, identifying the inclusion of rotor orientation as a natural extension of their approach.

Finally, for wind direction changes exceeding the maximum yaw offset simulated in CFD, the inference output was set to γ = 15°.

5.3.8 Numerical-experimental power matching

To ensure consistency between the power output of the wind generator and the aerodynamic predictions derived from CFD simulations, a three-step correction technique was employed. Consistency between experimental tests and numerical simulations was necessary to grant accurate real-time monitoring within the DT feedback logic for proper CFD interpolation.

First, a Clogh-Tocher piecewise cubic interpolant was defined using the correctly aligned (γ = 0°) CFD dataset. Mapping the C_p^{CFD} values across the $(TSR\ , U_\infty)$ input space allowed the model to predict the aerodynamic power coefficient for any measured operating point, analogously to the approach detailed in Sect. 5.3.7.

Second, instantaneous experimental records of TSR and U_∞ from Trial II, which established the performance reference for the aligned turbine (Sect. 5.3.12), were fed into the interpolator to determine the predicted C_p^{CFD} for γ = 0° via CFD. Before interpolation, outlier filtering was applied to each experimental time series using the interquartile range (IQR) method with a threshold equal to 2.5. This value was selected as a more conservative criterion than the standard 1.5 in order to avoid excessive removal of valid transient fluctuations observed in the experimental signals.

This step was formulated as a quasi-steady correlation rather than a time-resolved matching of flow dynamics; at each acquisition instant, the measured TSR and U_∞ queried the CFD-derived interpolator to evaluate the numerical response predicted for those specific conditions. Consequently, the steady RANS simulations served as a physics-based reference for the experimental measurements without attempting to reproduce instantaneous, stochastic unsteady phenomena.

In a third step, a correction factor (CF) dataset was determined as the pointwise residual difference between interpolation-predicted and experimentally obtained C_p^{meas} values under the same measured TSR and U_∞:

$$CF(TSR, U_\infty) = C_p^{CFD}(TSR\ , U_\infty) - C_p^{meas}(TSR\ , U_\infty) \tag{5.5}$$

Next, a cubic spline model was fitted to the pointwise CF values, with the smoothness parameter s equal to 1, to create a correction surface mapping the (CF, TSR, U_∞) space.

This method was introduced to compensate for discrepancies arising from sensor noise, unmodelled aerodynamic effects and mechanical-electrical conversion inefficiencies. The interpolation constructed using correctly aligned data was subsequently adopted as a first-order approximation for yawed operating conditions, under the assumption that yaw misalignment does not introduce significant changes in conversion losses beyond the aerodynamic reduction in C_p^{meas} reduction itself [137,139].

As a result, during the third trial used to validate the DT's yaw detection capability (see Sect. 5.3.14), the CF was evaluated for each record of TSR and U_∞ inside the decision interval. The factor, obtained by interpolating the cubic spline model derived from aligned conditions, was applied to the measured power coefficient to compensate for systematic non-aerodynamic offsets prior to DT inference. The corrected power coefficient supplied to the DT model was thus defined as:

$$C_{p,new}^{meas}(TSR, U_\infty) = CF(TSR\ , U_\infty) + C_p^{meas}(TSR\ , U_\infty) \tag{5.6}$$

5.3.9 DT logic for yaw fault detection and correction

The DT yaw fault detection logic was designed as a closed-loop, condition-based procedure operating on performance indicators averaged over discrete time intervals. The algorithm's objective was to detect misalignment originating from faulty wind direction measurements, infer the magnitude and sign of the yaw misalignment angle using CFD interpolation, and apply the necessary feedback for correction via the yaw system (Sect. 5.3.4).

The detection procedure operated over fixed temporal windows, aggregating sensor data from the continuous cloud stream to mitigate the influence of short-term inflow variability and measurement noise. In the present implementation, data were fetched over a 30 s sampling window, resulting in an effective loop duration of approximately 32-33 s. This timing was influenced by the characteristics of the data acquisition infrastructure but did not affect the generality of the proposed detection logic.

At each decision step, the DT model processed three dimensionless inputs: the corrected power coefficient $C_{p,new}^{meas}$, the TSR, and the wind speed U_{∞}. The yaw misalignment angle γ was then inferred by querying CFD-derived performance interpolators. The algorithm operated on the assumption that, once non-aerodynamic biases were compensated through the CF, short-term variations in power output were driven primarily by yaw misalignment.

Based on these inputs and assumptions, the DT yaw detection procedure was structured into four sequential steps:

1. Data conditioning: experimental time series of C_p^{meas}, TSR, and U_{∞} were filtered to remove outliers using the IQR method (threshold equal to 2.5).
2. Performance correction: the power coefficient C_p^{meas} was corrected through a corresponding correction factor CF, yielding a corrected power coefficient $C_{p,new}^{meas}$ consistent with the CFD performance results.
3. Misalignment inference: the corrected $C_{p,new}^{meas}$, together with TSR, and U_{∞}, were averaged over the current time interval and used to infer a yaw misalignment angle γ by interpolating CFD-derived performance data. Before interpolation, the averaged wind speed was rounded to select the closest corresponding response surface.
4. Fault decision and correction: yaw fault detection was based on the joint evaluation of the inferred angle γ and the temporal evolution of the corrected power coefficient $C_{p,new}^{meas}$ across consecutive intervals, enabling the identification of persistent performance degradation associated with yaw misalignment.

For each interval, the percentage change in $C_{p,new}^{meas}$ was defined as:

$$\Delta C_{p,new}^{meas} = \frac{C_{p,new}^{meas}(i) - C_{p,new}^{meas}(i-1)}{C_{p,new}^{meas}(i-1)} * 100[\%], \tag{5.7}$$

where i denotes the current decision interval. A yaw fault was flagged only if two conditions were simultaneously satisfied: (1) the inferred yaw misalignment angle crossed a predefined threshold of 5° for the first time, and (2) the corrected power coefficient $C_{p,new}^{meas}$ exhibited a consistent decrease ($\Delta C_{p,new}^{meas} < 0$) over two consecutive intervals relative to their respective preceding values. This combined criterion prevented false positives due to transient inflow fluctuations.

Within the yaw fault detection logic, the DT logic applied a feedback-based sign inference strategy to resolve the sign ambiguity of the misalignment. The algorithm initially assumed a positive yaw misalignment angle and applied a corresponding corrective action. If, despite this correction, the inferred misalignment remained above the threshold and the power coefficient decreased again in the subsequent interval, the initial hypothesis was rejected and a double corrective action in the opposite direction was applied. This feedback-based procedure enabled the DT to infer the sign of the yaw misalignment angle based on performance response rather than relying on wind direction measurements.

In the absence of inferred faults, the wind vane sensor data were considered to be correct, and the control logic operated to adjust the yaw angle of the turbine to be aligned with the measured wind direction. To ensure a consistent response, a control flag was implemented in the algorithm to disable further actuations linked to the healthy sensor state once a faulty sensor state was detected. As a result, the turbine's yaw angle adjustments relied only on CFD interpolation, maintaining alignment with the wind direction even in the presence of unreliable sensor data. In this configuration, the CFD interpolation itself acted as a performance-based virtual sensor, providing yaw measurements otherwise provided by the faulty device.

The control loop was executed at the defined sampling interval of 30 s, after which a new set of measurements was acquired and the detection and correction logic was re-evaluated. During experimental testing, the DT model could be manually interrupted, in which case the yaw system was driven to a neutral 180° reference position to ensure a safe and repeatable shutdown.

5.3.10 CFD results and integration within the DT

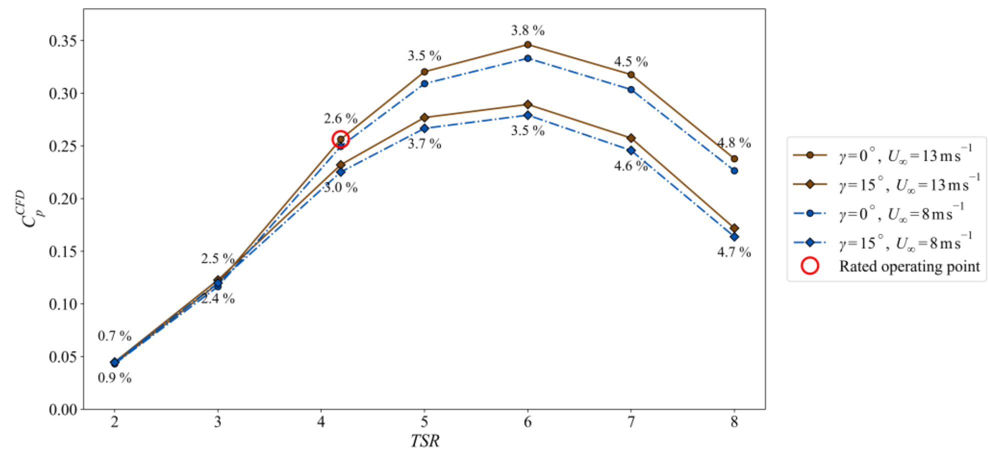

Figure 5.5. CFD-derived C_p^{CFD}-TSR curves for γ = 0° and γ = 15°, at U_∞ = 8 m s^{-1} and 13 m s^{-1}. Relative deviations from the 13 m s^{-1} baseline are illustrated, confirming aerodynamic similarity across this range (adapted from [118]).

The following sections present the validation results of the first DT implementation developed for yaw misalignment detection and correction. The validation is structured in two complementary parts. The first one covers the numerical foundation of the DT, examined through the analysis of the CFD-derived performance surfaces and the corresponding power loss trend induced by yaw. The second part assesses the complete architecture under real operating conditions through the three outdoor experimental trials previously introduced in Sect. 5.3.2, , evaluating detection capability, corrective behaviour, and overall inference robustness.

Figure 5.5 illustrates the aerodynamic power performance obtained from CFD simulations, expressed as C_p^{CFD} against TSR, for the extreme yaw misalignment angles simulated in this study (γ = 0° and 15°) and across wind speeds of 8 m s^{-1} and 13 m s^{-1}. Relative deviations from the reference case at 13 m s^{-1} are marked on each curve. Additionally, a red circle highlights the rated operating point of the selected small-scale HAWT (TSR = 4.19).

The results confirmed that in the TSR range from 2 to 8, the aerodynamic performance of the rotor was almost identical in terms of wind speeds. In any yaw misalignment angle, the relative difference in C_p^{CFD} between 8 m s^{-1} and 13 m s^{-1} was less than 5 %, highlighting the aerodynamic similarity between the numerical results. Consequently, by modelling the wind turbine's aerodynamic performance across an 8-13 m s^{-1} inflow range with a single averaged curve per yaw direction, the DT's inference dimensionality was effectively reduced without sacrificing physical accuracy.

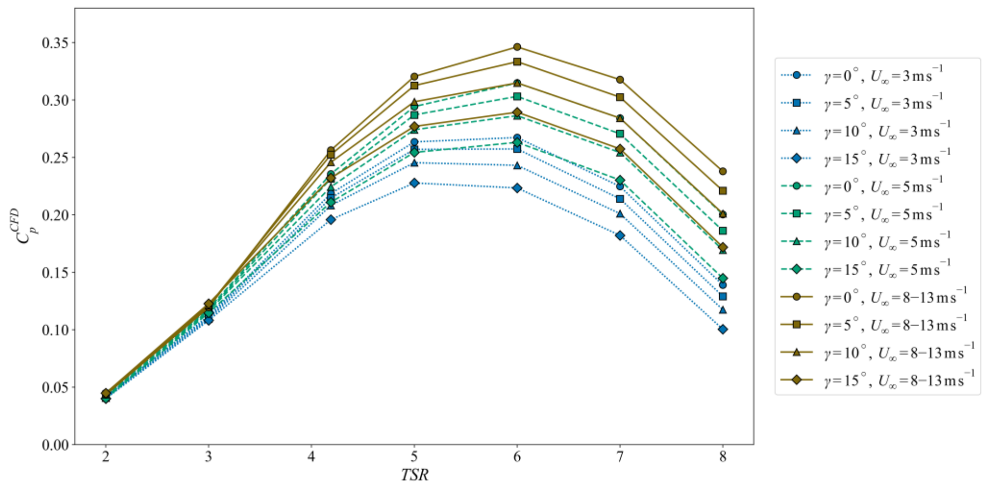

Figure 5.6. CFD-derived C_p^{CFD}-TSR curves for γ = 0° and γ = 15°, at U_∞ =3 m s^{-1}, 5 m s^{-1}, and for the averaged 8-13 m s^{-1} inflow range (adapted from [118]).

Figure 5.6 extends the analysis by presenting the complete set of CFD-derived C_p^{CFD}-TSR, curves for all simulated yaw misalignment angles (γ = 0°, 5°, 10°, 15°) and for the three directly investigated inflow regimes (U_∞ = 3 m s^{-1}, 5 m s^{-1} and the averaged 8-13 m s^{-1} range).

Across all wind speeds, increasing yaw misalignment produced a systematic reduction in aerodynamic performance, evidenced by a progressive downward shift of the C_p^{CFD}-TSR curves and a decrease in peak power coefficient. This monotonic degradation with increasing γ was observed for each U_∞ and was consistent with trends reported in the literature for yaw-misaligned HAWTs [124,137]. The figure also highlights the averaged performance curve for the 8-13 m s^{-1} range, where aerodynamic similarity was observed (Figure 5.5). Peak power extraction for this range occurred at TSR = 6 under aligned conditions (γ = 0°).

Overall, the CFD results indicated that the aerodynamic response of the rotor to yaw misalignment was well-structured across the investigated operating range, with performance losses that scaled consistently with increasing yaw misalignment while preserving the characteristic C_p^{CFD}-TSR behaviour. This response defined a robust aerodynamic mapping, which constituted the numerical foundation for the DT to infer γ from measured performance parameters.

The CFD results for the rated wind speed of 13 m s^{-1} showed that the yaw-induced power degradation followed a cosine-based loss pattern with a TSR-dependent exponent. To quantify the yaw-induced power loss trend, the power output at various yaw misalignment angles was fitted to the classical cosine-exponent model [137], expressed as:

$$P(\gamma) = P_0 \, cos^n(\gamma) \tag{5.8}$$

Where P_0 is the power extracted for the turbine properly aligned with the wind and n the exponent governing the power drop. A linear regression of $\log(P(\gamma)/P_{0)})$ versus $\log\,(cos(\gamma))$ was used to estimate the best-fit exponent n, following the approach in yaw sensitivity modelling. As shown in

Table 5.2, the exponent increased with TSR , ranging from -0.94 at TSR = 2 to 8.93 at TSR = 8, with the rated operating point of the small-scale HAWT being represented by TSR = 4.19. This trend confirmed the nonlinearity of yaw sensitivity with operating conditions and further supports the fidelity of the aerodynamic model.

Table 5.2 Cosine law exponent characterising the yaw-induced power loss at each TSR for the rated wind speed (adapted from [118]).

TSR	Exponent n
2.00	-0.94
3.00	-0.65
4.19	**2.80**
5.00	4.10
6.00	4.97
7.00	5.77
8.00	8.93

5.3.11 Trial I - Data acquisition and streaming verification

As previously introduced in Sect. 5.3.2, the DT-based method was assessed through a three-trial outdoor campaign conducted under natural wind fluctuations. Each trial addressed a distinct validation objective. Trial I established real-time data acquisition and inflow characterisation; Trial II evaluated the turbine in a correctly yawed configuration and enabled numerical-experimental power matching with CFD, providing the CF model required for DT inference; and Trial III demonstrated the DT's capability for yaw misalignment detection and autonomous fault correction.

The first trial aimed to verify real-time acquisition and cloud storage of sensor data. Sensor values (voltage, current, wind speed, direction and rotational speed) were streamed at 1 Hz and successfully stored in the cloud database. Wind speed and direction data were used for inflow characterisation, rotational speed for tip speed ratio calculation, current and voltage for electrical power estimation. This data laid the foundation for deriving C_p^{meas} and TSR in later trials. Wind data logged in this trial indicated outliers due to natural turbulence, which encouraged implementing IQR filtering.

5.3.12 Trial II - Electromechanical power characterisation

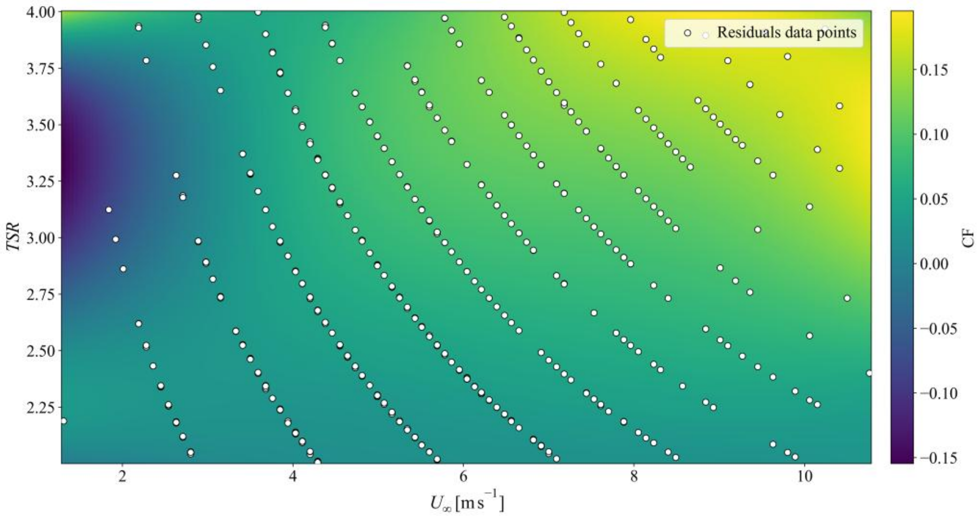

Figure 5.7. Correction factor (CF) values derived from the pointwise difference (residuals) between experimental and CFD-based C_p^{CFD} results (adapted from [118]).

A second experimental session took place at the airfield of Cranfield University to measure the electrical power output under actual loading conditions for the correctly yawed turbine (γ = 0°). During Trial II, the turbine was left to be passively yawed by the wind. From the current and voltage records, the electromechanical C_p^{meas} could be calculated and later compared against the aerodynamic C_p^{CFD} derived from CFD simulations. A notable deviation was observed between these datasets due to conversion losses and electrical load limitations. As Trial II represented the sole experimental reference used to establish the numerical-experimental power matching required by the DT, all corrections employed in subsequent trials were derived exclusively from this dataset.

To bridge this gap, Figure 5.7 presents the pointwise residual CF datapoints, defined as the pointwise residual difference between the instantaneous measured power coefficient C_p^{meas} values and the corresponding values interpolated from the CFD surface for aligned conditions ($\gamma = 0°$), as detailed in Sect. 5.3.8 and Eq. (5.5). These residual samples were used to construct a smooth cubic spline correction surface, shown as the coloured field in Figure 5.7, while the superimposed markers indicate the original residual datapoints.

The correction data exhibited a banded distribution, corresponding to discrete operating conditions reached during testing. After IQR-based outlier removal, the dataset contained measured wind speeds between 2.5 m s^{-1} and 11 m s^{-1}, and TSR values between 2 and 4.

The fitted spline exhibited a definite positive gradient in CF with increasing alongside both U_∞ and TSR. This trend revealed larger discrepancies between experimental and CFD-based values at higher operating points, likely stemming from increased rotational speeds and unaccounted electrical losses. By integrating this correction model, the DT was allowed to effectively balance field measurements with CFD predictions and to ensure consistent performance inference together with robust fault detection.

The effectiveness of the CF correction was quantitatively assessed under aligned operating conditions ($\gamma = 0°$) using 5-fold cross-validation on the experimental dataset from Trial II. The agreement between experimental and CFD-derived power coefficients was evaluated in terms of the mean absolute error (MAE), root mean square error (RMSE), and mean bias, defined here as the average signed difference between the measured and CFD-predicted power coefficients.

Without correction, the measured power coefficient exhibited a strong systematic underestimation relative to CFD predictions (MAE = 0.109, RMSE = 0.128, bias = -0.107). After applying the CFD-informed correction surface, the discrepancy was significantly reduced, with the MAE decreasing to 0.018 and the RMSE to 0.027, corresponding to average reductions of 83 % and 79 %, respectively, while the mean bias was effectively eliminated.

These results confirmed that the model effectively compensated for systematic experimental offsets rather than overfitting the calibration data. By validating the CF exclusively under aligned conditions (γ = 0°), non-aerodynamic biases were successfully isolated to avoid contaminating the yaw-dependent power loss data.

Consistent with the assumptions in Sect. 5.3.8, the power matching was applied to all yawed conditions as a first-order compensation for non-aerodynamic losses. By applying this correction to individual measurements prior to temporal averaging, the aerodynamic variability required for the DT-based yaw misalignment inference was preserved, ensuring that yaw-induced variations were handled exclusively through CFD interpolation.

5.3.13 CFD-based yaw misalignment inference within the DT

Figure 5.8 illustrates the CFD-based inference of yaw misalignment embedded within the DT, shown for the performance surface at U_∞= 4 m s^{-1} as a representative example. The surface represents the forward aerodynamic response $C_p^{CFD}(TSR\,, U_\infty)$ obtained through piecewise cubic interpolation of CFD results.

Within the DT, this surface was queried in inverse mode. During each decision interval, experimental measurements of C_p^{meas}, TSR, and U_∞ were first filtered to remove outliers and each retained C_p^{meas} sample was individually corrected through the numerical-experimental power matching procedure. The corrected power coefficient $C_{p,new}^{meas}$, together with TSR and wind speed U_∞, was then averaged over the current time window. This defined a well-posed operating point within the CFD-derived performance space, effectively representing the turbine's state. The generally monotonic relationship between C_p^{CFD} and γ enabled unique yaw inference via numerical inversion of the interpolated surface.

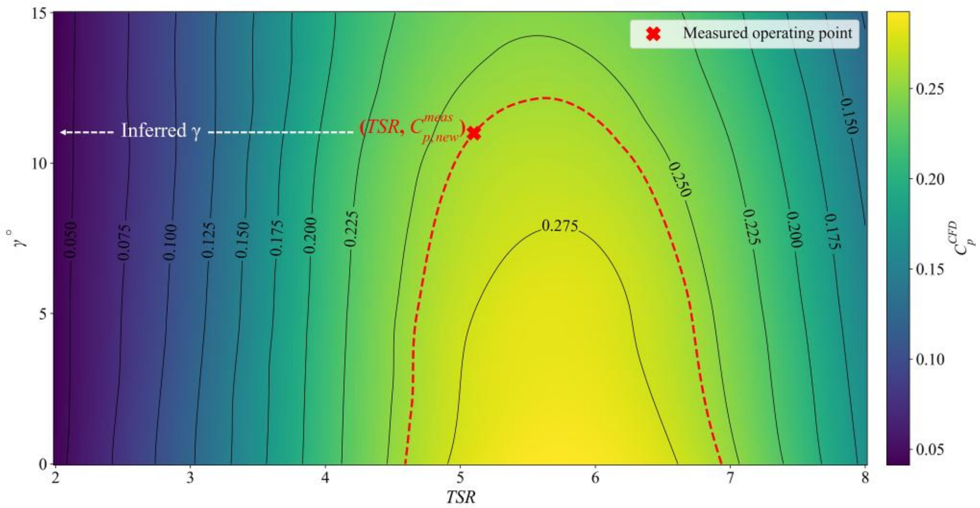

Figure 5.8. Example of the CFD-derived performance surface at U_∞= 4 m s^{-1}, showing C_p^{CFD} as a function of TSR and γ, used by the DT for yaw misalignment inference (adapted from [118]).

For each decision step, the DT selected the CFD response surface associated with the integer-rounded wind-speed class, and operating points exceeding the simulated yaw range were clipped to $\gamma = 15°$. The corrected and time-averaged operating point was then projected onto the (TSR,C_p^{CFD}) plane of the selected surface, and the corresponding yaw misalignment angle γ was inferred through piecewise cubic interpolation as described in Sect. 5.3.7.

In this configuration, the CFD interpolation acted as a performance-based virtual sensor, replacing direct wind-direction measurements when these are unreliable and providing the DT with a robust basis for yaw fault detection.

5.3.14 Trial III - DT validation and fault detection

The final trial tested the ability of the proposed method to detect and correct yaw misalignment. During this phase, in addition to the standard signals (wind speed, wind direction, power, rotational speed), the CFD-inferred yaw misalignment angle γ, the measured yaw position, and the averaged $C_{p,new}^{meas}$ trend were computed over each time interval. These parameters were used by the DT to detect misalignments and decide whether to override faulty wind direction readings. The trial consisted of four sub-tests, designed to progressively validate the yaw actuation, fault detection capability, correction logic, and recovery behaviour of the DT under challenging inflow conditions:

- Test 1, verification of yaw system actuation: A test of the customised yaw system of the turbine (Sect. 5.3.4) that validated its ability to align the rotor with the incoming wind direction at each sampling interval.
- Test 2, positive yaw misalignment: A static yaw misalignment of +10° was intentionally induced by subtracting 10° from the wind vane measurements before running the test, thereby introducing a systematic bias in the wind direction signal. As a result, the nacelle was oriented 10° away from the real wind direction, creating a controlled physical misalignment angle caused by a simulated sensor fault. The DT model correctly sensed the misalignment through decreased performance and successfully aligned the turbine with the real wind direction, circumventing the faulty sensor after the deviation had been confirmed. At the end of the second test, the DT was also capable of detecting and following an average change in the wind direction, acting as a performance-based virtual yaw sensor in addition to diagnosing and correcting the faulty yaw measurement.
- Test 3, positive yaw misalignment: A repeat test with a +10° offset was performed, similarly to Test 2. However, the DT failed to confirm the misalignment in this instance. This was likely due to increased turbulence and high-frequency gusts, which temporarily obscured the performance degradation required for successful detection.
- Test 4, negative yaw misalignment: A static yaw misalignment of -15° was intentionally induced by adding +15° to the wind vane measurements before running the test, thereby introducing a systematic positive bias in the wind direction signal. As a result, the nacelle was oriented 15° away from the real wind direction in the opposite direction, creating a controlled physical misalignment angle caused by a simulated sensor fault. The DT model correctly sensed the misalignment from the interpolated CFD performance space and computed the appropriate yaw correction command.

Although these results demonstrated the promise of the DT in independent fault detection and correction, the system also proved sensitive to wind turbulence and flow inhomogeneities typical of near-ground operation. Such phenomena are inherent to the lower atmospheric boundary layer and can occasionally induce spurious corrections. In utility-scale wind turbines operating at higher hub heights, where the inflow is generally more uniform and less affected by surface roughness, the influence of such disturbances is expected to be reduced, although further validation would be necessary to fully assess this behaviour.

Test 1 - Yaw actuation verification. Figure 5.9 presents the outcome of Test 1, which was designed primarily to verify the real-time responsiveness of the modified yaw actuation system under dynamic inflow conditions. Variable wind conditions were registered during the test. As observed in the first panel, the turbine yaw angle was first initialised at 180°. In the absence of detected sensor faults, the DT control logic allowed the yaw system to align the turbine with the measured wind direction at each control interval, while the CFD-based yaw inference operated in parallel as a monitoring layer. Rapid changes in the wind direction occurred during the test, with values up to -18°. The DT model captured these changes through the resulting variations in $C_{p,new}^{meas}$ and TSR, intermittently inferring yaw misalignment angles of 15° and 10° as shown in the second panel.

Although the CFD-based inference intermittently indicated large apparent yaw deviations under gusty conditions, the prescribed misalignment criteria were not met systematically; therefore, no sensor fault state was triggered, and the yaw actuation remained driven by the wind vane measurements, consistent with the scope of Test 1.

This observation suggests that DT could potentially monitor fast changes in wind direction and $C_{p,new}^{meas}$. Furthermore, the analysis demonstrated that misalignment detection is more likely to occur during conditions of high wind strength, which led to greater TSRs and increased the responsiveness of the aerodynamic model.

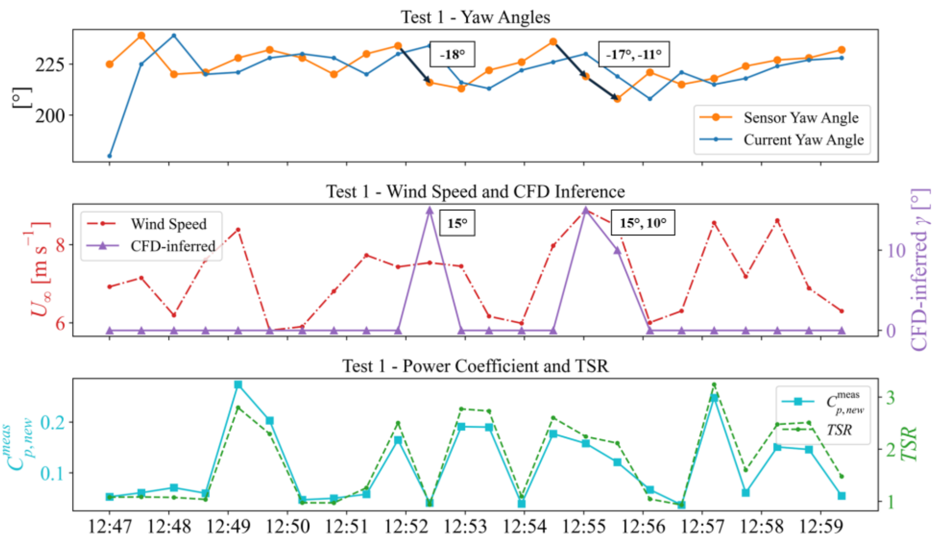

Figure 5.9. Time series from Test 1: verification of the yaw actuation system and initial DT fault inference under gusty conditions (adapted from [118]).

Test 2 - Detection and correction of +10° misalignment. Test 2 results are shown in Figure 5.10. The test started with a -10° artificial bias in the wind vane readings, inducing a positive static yaw misalignment of +10°. A 13° yaw misalignment was detected by the DT model at time 13:14:09 after about 3 minutes and corrective feedback was sent at the end of the loop, marked with a vertical dashed grey line. Due to the system's mechanical inertia, a second 9° error was detected in the subsequent loop, but no actuation was performed because of a $C_{p,new}^{meas}$ increase of 112 % following the first actuation.

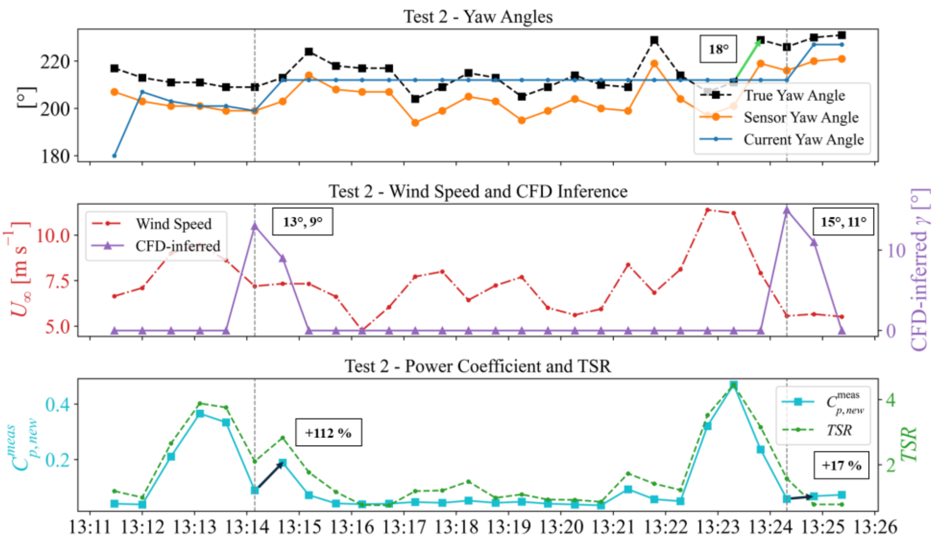

Figure 5.10. Time series from Test 2: detection and correction of a +10° yaw misalignment (adapted from [118]).

In addition, the first panel shows the real wind direction (marked in black), corresponding to the original wind vane measurement prior to the induced bias, towards which the system adjusted after the corrective feedback provided by the DT. Additionally, a strong wind direction change (+18°) was inferred close to the end of the test, which was inferred by the DT as a +15° deviation in the subsequent interval, triggering a corresponding feedback action. A partial recovery of $C_{p,new}^{meas}$ (about +17 %) ensued and prevented a second feedback command. This reaction further illustrates the system's capability to adapt to changing inflow conditions and the role of the CFD inference as a performance-based virtual sensor; the DT did not issue any unnecessary actuation commands, demonstrating robustness against false positives under gusty conditions.

Test 3 - Unsuccessful correction of +10° misalignment. This test is included to document a representative operational condition under which yaw correction was not achieved, highlighting the influence of inflow variability on DT-based control actions rather than a malfunction of the inference logic. Figure 5.11 presents the results for Test 3. As for Test 2, a static positive yaw misalignment of +10° was intentionally induced in the wind vane readings from the beginning of the test.

At time 13:33:36, the DT, through CFD interpolation, inferred a misalignment of 12°. However, due to strong fluctuations in the wind inflow, which directly affected the power production, the conditions required by the DT decision logic were not consistently satisfied. As in Test 1, the inferred misalignment was not sustained over consecutive intervals with a stable and coherent power deficit, and the DT therefore remained in a monitoring state without issuing corrective commands. This test was characterised by the lowest maximum wind speeds registered during the trial.

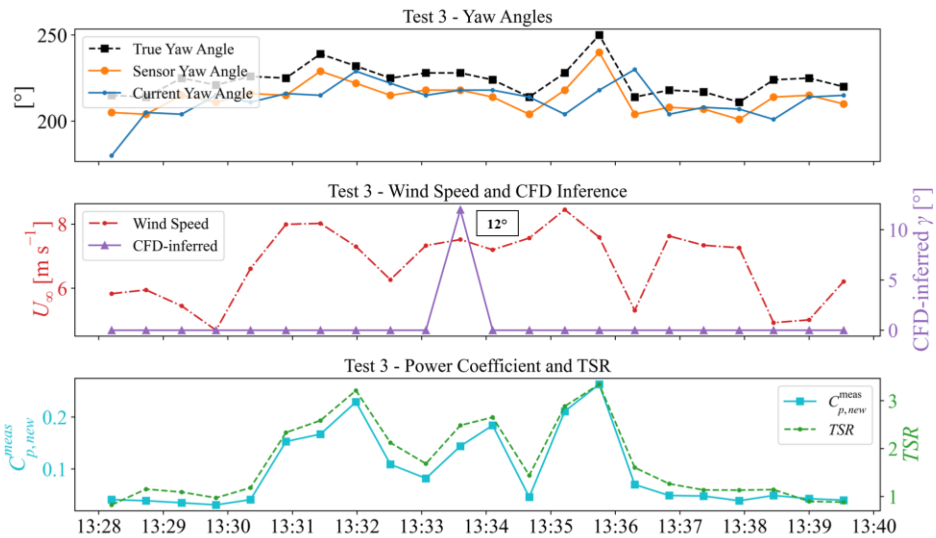

Figure 5.11. Time series from Test 3: fault detection under +10° misalignment suppressed by unstable inflow (adapted from [118]).

Test 4 - Detection and correction of -15° misalignment. Figure 5.12 presents the results from Test 4 that concluded the Trial III. In this scenario, a static negative yaw misalignment of -15° was intentionally simulated. At the beginning of the test, the DT signalled the presence of a misalignment, initially inferring a yaw misalignment angle of 14°. However, only at time 13:49:36, after about 7 minutes, could the algorithm confirm a sustained misalignment (15°) and issue a corrective feedback command, which was initially classified as positive according to inference logic (Sect. 5.3.9).

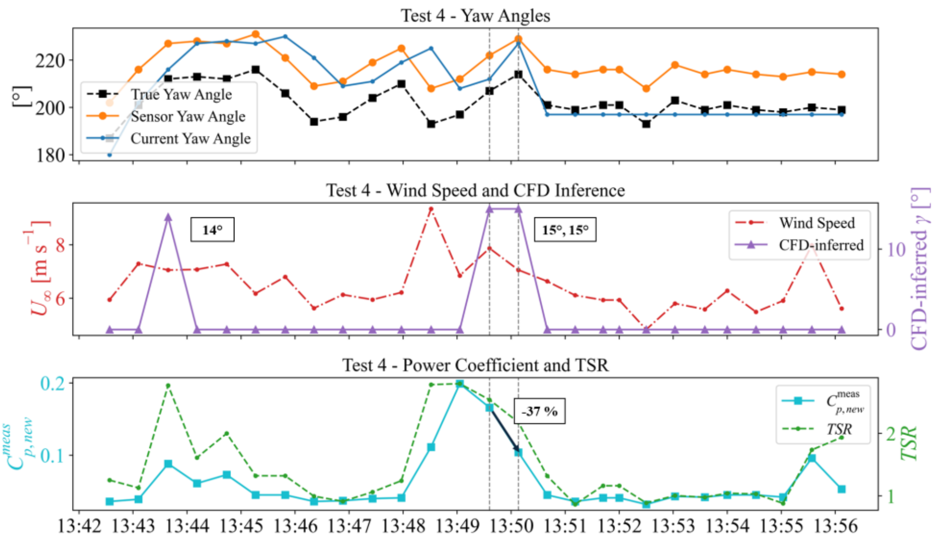

Figure 5.12. Time series from Test 4: detection of -15° fault and recovery sequence. (adapted from [118]).

Following the first feedback action, the registered $C_{p,new}^{\text{meas}}$ showed a -37 % decrease. This performance drop indicated that the applied actuation was in the incorrect direction (positive clockwise rotation). In the subsequent loop, the DT again detected a 15° misalignment and, based on the post-actuation power deficit, issued a second corrective command of -30°. This second actuation resolved the negative yaw fault and successfully realigned the turbine with the wind inflow. The test therefore highlighted the DT's capability to self-correct directional misclassification through performance-based feedback.

In this case, recovery of $C_{p,new}^{meas}$ in the time series was not clearly distinguishable due to a rapid decrease in wind speed, which drove the turbine to TSR values close to 1. Although a modest increase in power is visible after the final correction, it is difficult to isolate this recovery from the effects of reduced inflow velocity. These trial outcomes directly informed the accuracy evaluation summarised in Sect. 5.3.15.

5.3.15 Accuracy evaluation metrics

Table 5.3. Evaluation of yaw detection and correction performance using the DT during the third experimental trial (adapted from [118]).

Metric	Definition	Evaluation based on tests
Detection success rate	Percentage of faulted tests in which the CFD interpolation detected yaw misalignment ($\gamma \geq 5°$).	3 out of 3 faulted tests: **100 %**
Yaw correction accuracy	$\lvert\gamma_{inferred} - \gamma_{true}\rvert$	Test 2 (first feedback): **3°** Test 2 (second feedback): **3°** Test 4: **0°**

$C_{p,new}^{meas}$ *recovery (%)*	Percent increase in $C_{p,new}^{meas}$ following yaw correction (when applicable).	Test 2 (first feedback): +**112 %** Test 2 (second feedback): **+17 %** Test 4: **-37 %** after mis-actuation, followed by a recovery attempt, but with an unclear signal due to wind drop.

Table 5.3 summarises the performance of the DT model during Trial III, where three out of four tests were subjected to induced static yaw faults in the wind vane readings. The DT successfully inferred misalignment conditions in all faulted cases, with a 100 % detection success rate. Regarding Tests 2 and 4, yaw correction accuracy, defined as the absolute difference between the estimated and true yaw misalignment angles, ranged from 0° to 3°, indicating good agreement between the DT inference and the imposed yaw faults.

Large recoveries in $C_{p,new}^{meas}$ were observed after corrective action, such as a +112 % recovery in Test 2. In Test 4, an incorrect initial actuation resulted in a -37 % drop, necessitating a second corrective action, though further recovery was masked by falling wind speed.

5.4 Limitations and architectural lessons from the first DT

The first DT implementation demonstrated that CFD inference can operate as a performance-based virtual sensor under real atmospheric conditions, enabling detection and correction of static yaw misalignment without prior exposure to faulty datasets. However, the outdoor validation also highlighted structural and methodological constraints that contributed to the subsequent evolution of the architecture.

A primary limitation emerged from the turbine operating in close proximity to the ground, due to site constraints during the first experimental campaign, exposing the rotor to elevated shear and turbulence levels. Although 30 s aggregation mitigated short-term variability in most cases, elevated turbulence occasionally obscured the aerodynamic signature required for reliable corrective action. This observation emphasised the importance of environmental observability and filtering strategies when applying performance-based inference in stochastic conditions.

Second, the yaw DT relied on explicit parametric embedding of the fault variable (γ) within the CFD database, enabling direct inversion in a structured performance space. This formulation resulted to be appropriate for deterministic control-state faults such as static yaw misalignment. In contrast, degradation phenomena such as LEE (leading-edge erosion) produce distributed aerodynamic effects whose statistical signatures may overlap under real inflow variability.

These observations motivated two refinements in the subsequent study. Experimentally, the monitoring setup was improved to reduce ground-induced interference and to separate environmental and performance acquisition subsystems, enhancing measurement clarity. Architecturally, the DT was extended toward a hybrid formulation, retaining CFD-derived performance maps as the physical reference while introducing data-driven classification to improve discrimination between statistically overlapping degradation states.

Additionally, in contrast to the first implementation, the following study explicitly formalises the four-layer DT architecture, reflecting the methodological consolidation derived from the yaw demonstrator. This refined DT is adapted for LEE monitoring, illustrating how the layered architecture addresses progressive aerodynamic degradation and supports maintenance-oriented decision-making.

5.5 Second DT: monitoring of early-stage LEE using CFD-informed ML

The second DT implementation builds directly upon the architectural and experimental insights gained from the first yaw demonstrator. While the first DT addressed a deterministic and reversible control-state deviation, the present study targets early-stage LEE as a progressive aerodynamic degradation mechanism affecting blade surface integrity and performance.

Particularly in the low-Reynolds regime characteristic of small-scale wind turbines, the increased surface roughness caused by early LEE triggers an earlier laminar-to-turbulent transition. This alters boundary-layer development and gradually reduces power production, often preserving the general shape of the performance curve while inducing a monotonic but subtle power deficit. Under real atmospheric conditions characterised by stochastic turbulence and inflow variability, performance data associated with different roughness levels may exhibit partial statistical overlap, complicating direct inversion within a purely physics-based framework.

For this reason, in the second DT a hybrid formulation was adopted. CFD-derived performance maps remained the physical reference within the virtual layer, ensuring interpretability and consistency with aerodynamic principles. However, residual-based features extracted from measured-virtual discrepancies were processed through a supervised classification model to enhance discrimination between clean and degraded states. Compared to the corrective logic of the yaw DT, the objective here was early degradation detection and health-state quantification to support maintenance-oriented decisions rather than immediate control actuation. A direct precedent is the work of Castorrini et al. [140], who trained ANN metamodels on CFD databases to predict eroded-airfoil force coefficients for energy-loss estimation; the present formulation instead uses CFD performance maps as a reference against which measured residuals are classified for early detection.

5.5.1 LEE: physical background and fault description

LEE has emerged as a critical reliability issue across the entire wind energy sector, contributing to reduced aerodynamic efficiency and long-term power losses, where rain, hail, salt and airborne particulates progressively erode the blade leading edge [141–144]. As the rotor size and tip velocity have increased with the goal of enhanced power production, the effects of airborne droplets and particles on the structure of the rotor surface have increased rapidly and led to an accelerated degradation of the blades [142,145,146].

Early-stage LEE corresponds to equivalent roughness heights from a few tens up to 100 µm [144,147–150], already producing measurable lift loss and drag rise. While these initial surface modifications may appear negligible, they fundamentally alter the aerodynamic performance of the airfoil. The mechanisms underlying this sensitivity have been reviewed in the broader turbomachinery context by Bons [151], who attributes the aerodynamic penalty of surface roughness to three coupled effects: the promotion of earlier laminar-to-turbulent transition, an increase in turbulent boundary-layer momentum thickness and, under adverse pressure gradients, the onset of flow separation. The magnitude of these effects is governed primarily by the Reynolds number and the roughness size, which is consistent with the strong influence that even sub-100 µm roughness exerts on the low-Reynolds blades considered in this study.

Consequently, even incipient degradation can lead to significant cumulative energy production losses, necessitating a monitoring framework capable of distinguishing these subtle performance shifts from environmental noise.

5.5.2 CFD modelling - virtual layer development

Geometry and domain. The aerodynamic response surfaces at the base of the virtual layer of the proposed DT were obtained from RANS simulations performed in ANSYS Fluent 2024R1, on the Delta2 HPC cluster of Cranfield University. The same validated geometry and computational setup previously detailed in Sect. 5.3 were used for this study. The CFD model reproduced the selected small-scale HAWT (Pikasola 400W-12V, 1.3 m rotor diameter) with its rotor and nacelle, without modelling the pole.

As introduced in Sect. 5.3.6, the fluid domain consisted of a box domain extending 4D upstream of the rotor plane and 8D downstream of it, where D is the turbine's diameter. To model the rotor's rotation, a Multiple Reference Frame (MRF) method [111] was employed for two consecutive rotating volumes: a disk domain enclosing the blades and a downstream cone-shaped domain. Both rotating domains utilised a locally densified mesh and were linked to the surrounding static domain via a frozen rotor interface.

Simulation settings. Consistent with common research practices [133,148,149], the Transition SST (γ-Re_θ) turbulence model was selected [152,153] for this CFD analysis. The polyhedral mesh resulted in approximately 30.4 million cells. Specifically, the first cell height was chosen to achieve an average $y^+ \approx 1$ with local maxima below 8, consistent with best-practice guidelines for near-wall resolution in transitional flows [111].

The inlet was prescribed as a uniform velocity boundary normal to the inlet plane, with turbulence intensity TI = 30 % and turbulent viscosity ratio equal to 200. A pressure outlet with zero-gauge pressure was set downstream, while all other lateral walls were treated as free slip walls.

To represent LEE in its early or incipient stages, surface degradation was modelled as distributed equivalent roughness applied to selected blade regions, without altering the blade geometry. At these early stages, erosion primarily modifies boundary-layer behaviour before inducing macroscopic geometric discontinuities. Under such conditions, equivalent sand-grain roughness provides a physically consistent aerodynamic surrogate for capturing roughness-induced performance alteration, as commonly adopted in the literature [144,147–150]. In addition, the roughness-correlation option was enabled to couple surface roughness to transition onset. In accordance with standard interpretations of uniform sand-grain roughness in near-wall modelling [154,155], the geometric roughness height k was set equal to the equivalent sand-grain roughness height k_s,

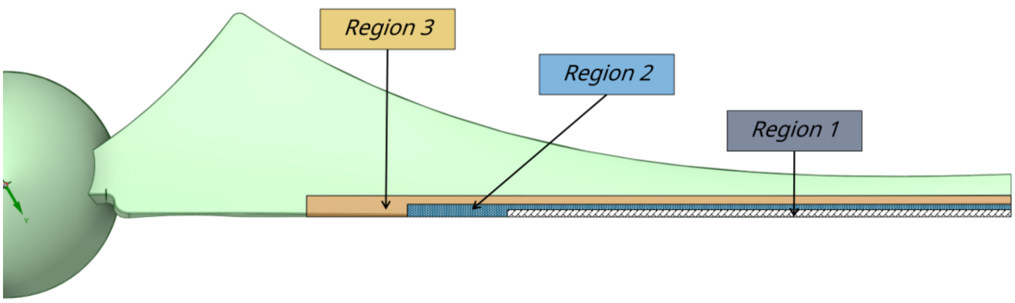

Figure 5.13. Roughness regions, defined for the CFD study, simulating progressive early-stage LEE (adapted from [119]).

Four surface conditions were investigated: the baseline clean condition (k_s = 0 μm) and three progressively larger rough regions, defined on each blade without altering their geometry (see Figure 5.13). The three rough regions had distinct roughness heights of 20, 40 and 60 μm; each patch equally extended on both suction and pressure sides and differed by spanwise coverage and chordwise extent:

- *Region 1*: 15 % of the tip chord, r/R = 50 % - 100 % (k_s = 20 μm).
- *Region 2*: 30 % of the tip chord, r/R = 40 % - 100 % (k_s = 40 μm).
- *Region 3*: 50 % of the tip chord, r/R = 30 % - 100 % (k_s = 60 μm).

Where r is the radial coordinate and R is the turbine radius.

Design of experiments. Simulations were performed for five inlet wind speeds (U_∞ = 3, 5, 8, 10 and 13 m s^{-1}), fixing TSR in the range from 2 to 8, and resulting in a total of 140 simulations (35 per surface condition). A summary of the CFD design of experiments adopted for the DT virtual layer is reported in Table 5.4.

Table 5.4. Summary of the CFD design of experiments for the DT virtual layer (adapted from [119]).

Parameter	**Value**
Equivalent sand-grain roughness height, k_s [μm]	0, 20, 40, 60
Inlet wind speed, U_∞ [m s^{-1}]	3, 5, 8, 10, 13
Tip speed ratio range, TSR [-]	2 - 8
Simulations per roughness case	35
Total CFD simulations	140

The aerodynamic power on the rotor was computed as the product of the converged rotor torque and the imposed rotational speed corresponding to the analysed operating point and used to compute the aerodynamic power coefficient C_p^{CFD} see Eq. (5.1).

The resulting four aerodynamic performance maps, constructed from discrete CFD data points (C_p^{CFD}, TSR, U_∞) for each roughness case, defined the numerical core of the DT's virtual layer. Each map was exported as a tabulated dataset and implemented in Python using a Clough-Tocher piecewise cubic 2D interpolation scheme [156], providing expected aerodynamic C_p^{CFD} values for all roughness conditions under measured inflow parameters (TSR-U_∞).

The interpolation accuracy was assessed using a leave-one-out cross-validation (LOOCV) on the CFD database for each roughness condition. Across all cases, the normalised root mean square error (NRMSE) remained below 3.4 %, with maximum absolute errors below 0.02 in C_p^{CFD} These results confirmed that the Clough-Tocher interpolation provided a smooth and sufficiently accurate physics-based reference for feature extraction within the sampled operating envelope. Consequently, the four interpolators were later queried in the intelligent layer to compare real-time measurements with physics-based predictions, forming the interface between the virtual and intelligent layers.

5.5.3 Experimental implementation - physical and data layer development

The experimental campaign supporting the physical layer of the DT was carried out on the rooftop of the Istituto Italiano della Saldatura (IIS) in Genoa, Italy (44° 25′ 40″ N, 8° 57′ 20″ E), approximately 58 m above sea level. The selected HAWT was the same Pikasola 400W-12V model previously introduced in Sect. 5.3.2.

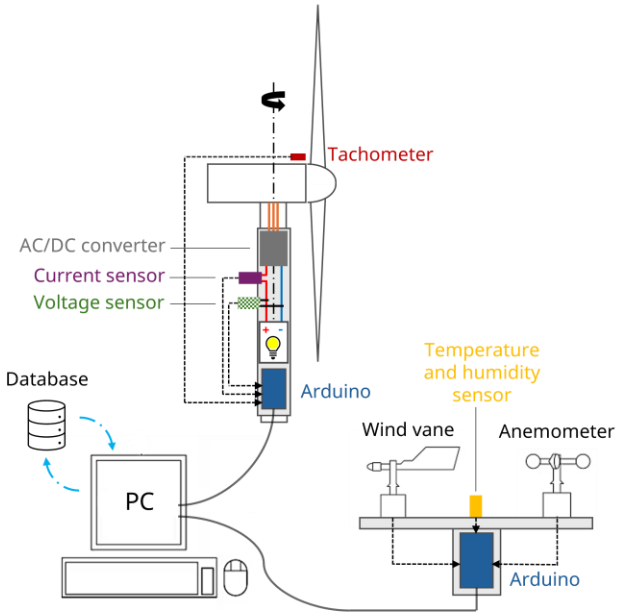

Figure 5.14. Schematic of the multiple elements forming the physical layer of the DT (adapted from [119]).

For this test campaign, conducted between July and September 2025, the turbine was installed on a steel telescopic pole secured to one of the rooftop's perimeter walls. This installation elevated the turbine an additional 4 meters above the roof level, resulting in a total hub height of about 62 meters above sea level. Mean wind speeds during tests ranged between 4 and 5 m s^{-1}, with gusts occasionally exceeding 10 m s^{-1}.

Compared to the previous study's setup [118], the DHT 22/2302 temperature-humidity sensor was added to the sensor network, and the DC sensor was replaced with the ACS712ELC-20A module for its higher sensitivity. A schematic of the elements forming the physical layer of the DT is provided in Figure 5.14.

Two Arduino Uno boards handled data acquisition for two distinct subsystems: one collected electrical and rotational quantities (voltage, current and rotor speed), while the second recorded environmental parameters (wind speed and direction, air temperature and humidity). Onboard computations also provided instantaneous values of DC power and air density (from temperature and humidity).

The two subsystems, consisting of the two microcontrollers and the associated sensors, were housed on two separate metal bases that were magnetically attached to the steel pole of the turbine. The first base, dedicated to the performance sensors, was secured vertically along the pole's height. This base also integrated the turbine's AC-DC rectifier, which was directly wired to the three-phase generator. The second base, housing the environmental Arduino board and the weather-station sensors, was attached ahead of the rotor (Figure 5.15).

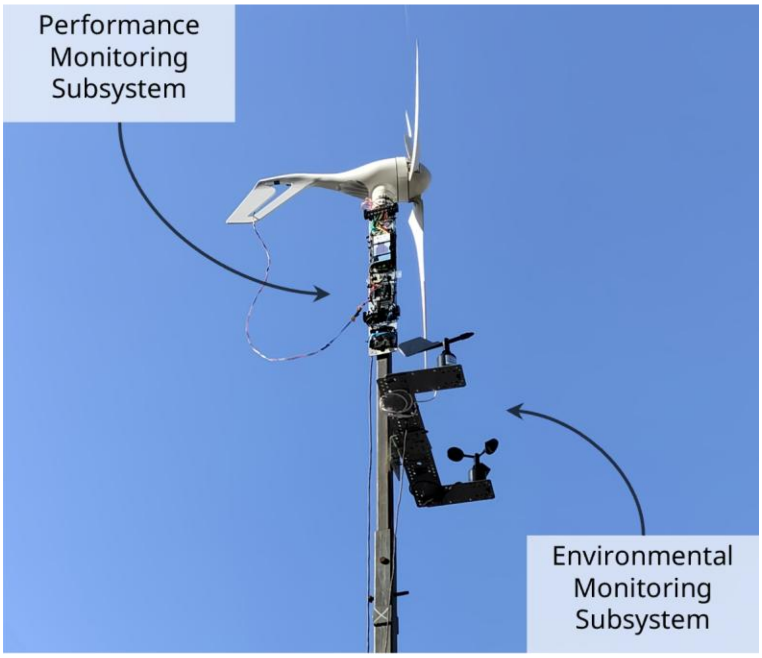

Figure 5.15. Photograph of the experimental setup for the selected small-scale HAWT (adapted from [119]).

Specifically, the sensors were positioned 1.2 m below the turbine axis and 0.25 m upstream of the rotor plane to minimise aerodynamic interference (a configuration validated through CFD postprocessing). Finally, the Hall-effect tachometer was fixed right after the rotor to detect the passage of the three blade-root magnets, enabling accurate measurement of the rotational speed and streaming data via wire to its reference microcontroller (Figure 5.16).

All channels were sampled at 1 Hz, in accordance with the IEC 61400-12 recommendations for wind-performance measurements [126]. The raw data streams from both microcontrollers were transmitted via USB serial connection to a host PC and uploaded in real time to two dedicated MongoDB Atlas collections using dedicated Python scripts [125]. Each acquisition script inserted a unique record per second into its MongoDB collection, time-stamped both by device and wall-clock time. This guaranteed initial synchronisation at the database level, while finer alignment was later performed when data was fetched back from the collections.

Before every outdoor test, all analogue channels were offset-calibrated and verified in-field under steady reference conditions, following the same calibration approach adopted in [118]. Residual measurement uncertainty was not modelled explicitly; instead, robustness was addressed at the system level through temporal aggregation and feature-based inference, such that diagnostic decisions were driven by aerodynamic trends rather than by individual sensor readings (see Sect. 5.5.4).

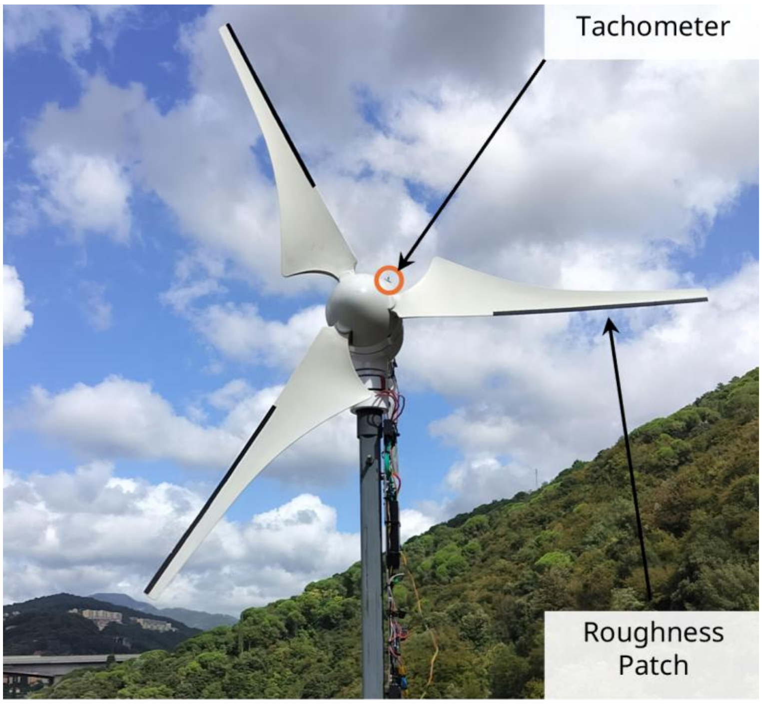

Figure 5.16. Example of the LEE setup for the P320 case. The tachometer used to measure rotor speed is also visible (adapted from [119]).

The turbine operated under a fixed electrical load consisting of a 12 V (55 W) halogen bulb connected downstream of the rectifier, providing an approximately resistive load condition. This arrangement did not impose active rotor speed regulation; instead, the rotational speed was wind-driven and governed by the balance between aerodynamic torque, generator electromagnetic resistance and mechanical losses. As a result, variations in rotor speed and power output primarily reflected changes in wind conditions and aerodynamic performance rather than control-induced effects.

The entire sensor network was designed to be compact, allowing rapid deployment and retrieval between test sessions. The setup modularity and rooftop accessibility facilitated repeated outdoor testing in variable wind conditions, ensuring that the measured performance data faithfully represented the turbine's dynamic behaviour for the validation of the DT. To enhance the observability of the physical twin, a real-time monitoring dashboard was implemented using the PyQtGraph Python library [157]. It consisted of two interfaces developed to visualise live data streams from the turbine’s power and weather subsystems (see Figure 5.17).

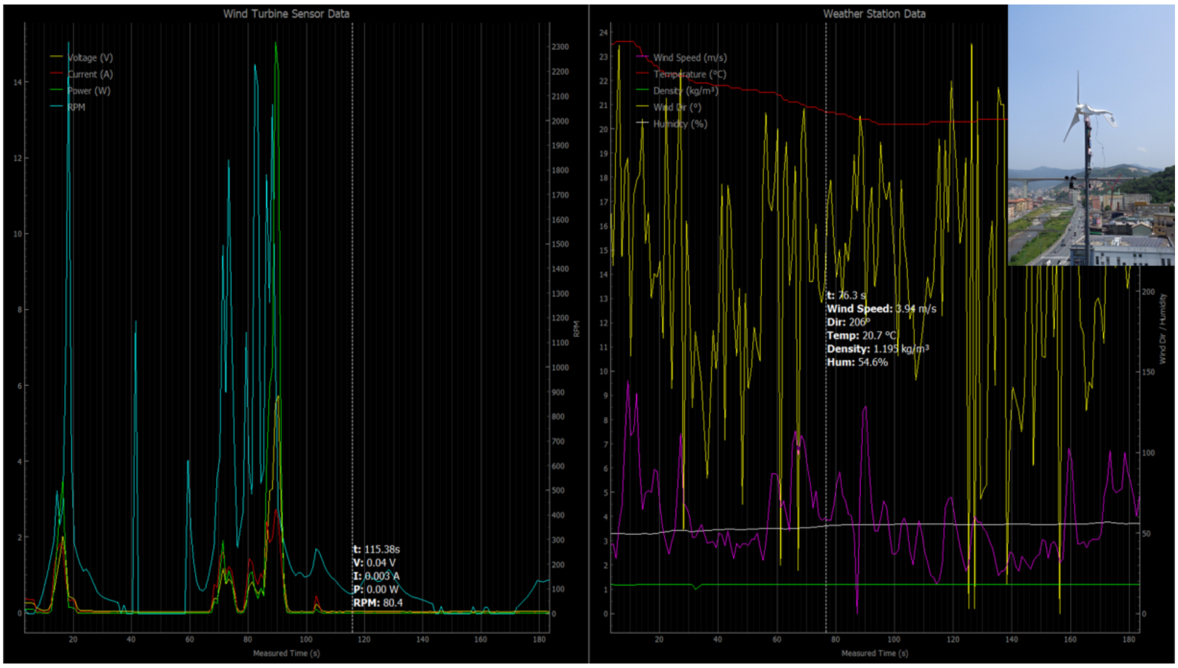

Figure 5.17. Example of two interfaces developed to visualise live data streams from the turbine's power and weather subsystems (based on [119]).

To emulate the roughness configurations analysed in the CFD study, controlled surface modifications were applied to the physical turbine blades. This was achieved using sandpaper patches with calibrated grit sizes, where the median grit diameter (D50%) was correlated to the target k_s values adopted in the numerical simulations [111,158]. This mapping was intended as an approximate aerodynamic equivalence rather than a strict geometric correspondence, consistent with the early-stage LEE focus of the study. Sandpaper patches were attached on both suction and pressure sides following the same spanwise and chordwise extents as the CFD regions described in Sect. 5.5.2.

A total of seven configurations were tested. The clean baseline condition was acquired twice (*clean*$_1$ and *clean*$_2$), serving as both a reference and a later test case for the DT roughness evaluation logic. This was followed by five progressively roughened states defined using the FEPA P-grit nomenclature [158]:

- *P1200* (D50% = 15.3 µm), matching the *Region 1* area.
- *P800* (D50% = 21.8 µm), matching the *Region 1* area.
- *P320* (D50% = 46.2 µm), matching the *Region 2* area.
- *P240* (D50% = 58.5 µm), matching the *Region 3* area.
- and *P100* (D50% = 156 µm), matching the *Region 3* area.

Each patch was cut to match the local blade curvature and applied using a removable adhesive, ensuring repeatable placement and minimal added mass (Figure 5.). Although patch boundaries and finite step heights may introduce local tripping effects in addition to distributed roughness, these effects were not isolated explicitly. The focus remained on reproducing the integrated aerodynamic response associated with early-stage LEE rather than resolving detailed transition mechanisms.

For each configuration, the turbine was tested under similar environmental conditions to isolate the aerodynamic impact of surface roughness. The specific roles of each roughness configuration within model training and validation are detailed in the following Sect. 5.5.4.

Upon completion of each test-day run (typically spanning about 3 hours of active turbine operation), the synchronised turbine and weather readings were retrieved from the MongoDB collections. Subsequently, C_p^{meas} and TSR values were automatically computed for each measured record. To ensure consistency between simulated and measured operating conditions, the records were filtered to retain only those with $C_p^{meas} > 0$ and bounded within the CFD domain ($2 \leq TSR \leq 8$; $3 \leq U_\infty \leq 13$ m s^{-1}). Although the C_p^{meas} reflected the overall electromechanical efficiency of the turbine, its variations were primarily attributed to aerodynamic effects, as mechanical and electrical losses could be considered approximately constant during operating periods [159,160].

For each roughness configuration dataset, every measured operating point (TSR-U_∞) was mapped onto the CFD-derived performance space using the four CFD interpolators introduced in Sect. 5.5.2. The corresponding C_p^{CFD} value was extracted for the same (TSR-U_∞) pair and four residual terms were computed as follows:

$$\text{res}_x = C_p^{CFD}(x = k_s) - C_p^{meas}, \tag{5.9}$$

The residuals represented the deviation of the measured (electromechanical) power coefficient from the CFD-predicted aerodynamic one at the equivalent sand-grain roughness height (k_s = 0, 20, 40, 60 μm). Subsequently, the residuals were divided by the measured TSR to account for changes in operating condition and by combining them with the measured aerodynamic variables ultimately formed the input feature space used by the DT to infer the blade surface condition.

The inference step relied on a supervised machine learning model trained to capture the relationship between aerodynamic behaviour and surface degradation. The selected model was an SVM classifier, whose input vector consisted of seven final hybrid features: $\{ U_\infty, TSR, C_p^{meas}, \frac{\text{res}_0}{TSR}, \frac{\text{res}_{20}}{TSR}, \frac{\text{res}_{40}}{TSR}, \frac{\text{res}_{60}}{TSR} \}$. This set of seven hybrid features was selected adapted from model accuracy and physical interpretability, ensuring that the SVM was trained on variables representative of both aerodynamic sensitivity and operational behaviour of the physical asset.

The final experimental datasets presented the following dimensions: 2465x7 (*clean$_1$*), 2189x7 (*clean$_2$*), 1997x7 (*P1200*), 2371x7 (*P800*), 2269x7 (*P320*), 1970x7 (*P240*) and 2013x7 (*P100*), where the second dimension corresponds to the number of hybrid features per record.

5.5.4 Intelligent layer development

The intelligent layer incorporated a decision engine adapted from the SVM classifier, which was trained to identify aerodynamic degradation states adapted from the experimental configurations introduced in Sect. 5.5.3. The classifier development pipeline followed standard machine learning practices adapted to a physics-bounded dataset [161].

Several classifiers were preliminarily evaluated, including decision trees, gradient-boosted trees (XGBoost), and multilayer perceptrons (MLPs). While comparable performance could be achieved after tuning, the SVM showed the most stable generalisation across unseen roughness configurations and limited sensitivity to hyperparameter choices. Given the low-dimensional, CFD-informed feature space, as well as the size of the dataset, the SVM was selected for its favourable trade-off between accuracy, robustness and probabilistic calibration.

The SVM was trained using a radial-basis-function (RBF) kernel and on the two datasets corresponding to the clean (*clean$_I$*) and rough (*P240*) conditions, representing the extreme aerodynamic states analysed with CFD (k_s = 0, 60 µm). The classifier was trained on the *clean$_I$* and *P240* configurations to maximise robustness and feature discrimination, due to the substantial overlap in the experimental feature space exhibited by intermediate roughness levels, limiting reliable multi-class separability. An 80/20 train-test split with a three-fold stratified cross-validation was applied during hyperparameter tuning. The model pipeline included median imputation, quantile transformation to a normal distribution and standard scaling. Class imbalance between the two datasets was handled exclusively through cost-sensitive learning by means of balanced class weights in the SVM objective function. A two-stage grid search optimised the regularisation parameter and the kernel coefficient to maximise the macro-averaged F1-score.

To retain sensitivity to intermediate degradation states despite the binary formulation, the calibrated posterior probability of the degraded *P240* class was used to define a continuous Roughness Index (RI) as a monotonic descriptor of surface degradation severity:

$$RI = 100 \times P(\mathrm{P240})\ [\%], \tag{5.10}$$

The RI ranged from 0 % (high confidence in a clean surface) to 100 % (high confidence in a fully roughened surface), providing a monotonic quantitative measure of the turbine's aerodynamic health rather than a discrete class label.

Two RI thresholds were subsequently introduced and employed by the DT to support maintenance-oriented decision logic. Rather arbitrary tuning, the thresholds were defined using the *P800* and *P320* configurations, introduced as intermediate roughness states representing progressive early-stage LEE between the clean and fully degraded states.

The first threshold, $RI_{\text{minimal→moderate}}$ = 31.30 ± 0.37 %, was obtained by averaging the median RI values associated with the *clean₁* and *P800* datasets, representing the transition from minimal to moderately surface roughness. Similarly, the second threshold, $RI_{\text{moderate→critical}}$ = 60.75 ± 0.26 %, was defined using the *P320* and *P240* configurations, marking the transition from moderate to severe surface roughness.

The uncertainties are one bootstrap standard error, obtained by resampling the contributing RI distributions (2000 resamples) and recomputing each threshold as the mean of the two resampled medians; they are small relative to the separation between the two thresholds and they reflect within-sample variability only, as each roughness state is represented by a single test campaign.

Although the specific values for the thresholds were calibrated for the current experimental scenario, their primary purpose was to illustrate the method's applicability rather than represent universal limits.

The remaining configurations *$clean_2$*, *P1200* and *P100* were excluded from model training and instead employed for an independent testing of the DT's inference capability. These cases served to evaluate the model's response to both near-clean conditions (*$clean_2$* and *P1200*) and to extreme surface degradation (*P100*), beyond the calibrated roughness range.

After model calibration, the trained SVM was embedded within the intelligent-layer algorithm for field operation, where it performed classification of surface roughness states from the incoming sensor data. As mentioned before, the present study adopted an offline batch-processing strategy, where the DT operated on complete test-day datasets collected during the experimental campaigns. Each record included voltage, current, electrical power, rotor speed, wind speed, direction, air temperature, humidity and density. From these, the DT computed instantaneous C_p^{meas} and TSR values according to Eqs. (5.2) and (5.3).

The data were then processed through the same CFD-bounded preprocessing pipeline defined for SVM model training. The algorithm discarded samples outside the CFD parameter space and used the pre-stored CFD interpolators from the virtual twin to evaluate $C_p^{CFD}(k_s)$ for the clean and three degraded conditions.

For each record, residuals $(C_p^{CFD}(k_s) - C_p^{meas})$ were calculated and divided by the measured TSR to reconstruct the seven hybrid features in the same order used during classifier training, ensuring feature-space consistency between the SVM model and the DT inference.

The SVM classifier was used to evaluate the test-day dataset, producing class probabilities and the corresponding RI distribution. Next, the intelligent layer computed the median RI across the entire batch, classified the turbine roughness state using the previously defined thresholds and provided the corresponding feedback:

- If $RI \leq RI_{\text{minimal→moderate}}$: *Clean turbine (minimal surface roughness)* - Max r/R coverage < 50 % - No corrective action needed.
- If $RI_{\text{minimal→moderate}} \leq RI \leq RI_{\text{moderate→critical}}$: *Turbine in progressive state (moderate surface roughness)* - Max r/R coverage < 70 % - In this state, the DT identifies a level of aerodynamic deterioration that, in typical operating environments, warrants a short-term maintenance schedule to limit further performance loss.
- If $RI \geq RI_{\text{moderate→critical}}$: *Turbine in alarm state (severe surface roughness)* - Max r/R coverage > 70 % - Under these conditions, the DT flags the need for urgent inspection, as the associated aerodynamic penalties become significant.

Adapted from the predicted roughness state, the layer also inferred the corresponding spanwise extent of surface degradation (e.g., < 50 %, < 70 % or > 70 %). This inference was possible because each experimental configuration was associated with a specific r/R coverage range, established in the numerical and experimental analyses, that served as an indicative range rather than a precise geometric reconstruction.

The suggested maintenance intervals (e.g., "short-term maintenance schedule") followed the convention adopted in this demonstrator and should be interpreted in the context of site-dependent LLE progression rates. Previous studies have shown that LEE evolves at distinctly different timescales depending on the blade material and local environmental conditions, including precipitation characteristics, particulate load; consequently, the DT recommendations expressed relative urgency adapted from aerodynamic performance deterioration, rather than prescribing absolute maintenance windows [162–164].

While this version of the DT operated on daily datasets, the same architecture inherently supports continuous operation. In such a configuration, the same inference workflow would be executed in real time, automatically aggregating measurements over fixed time windows to deliver daily diagnostics equivalent to those obtained through the current batch-processing approach.

5.5.5 Virtual layer results: CFD response surfaces

Figure 5.18 shows the interpolated aerodynamic maps of the power coefficient $C_p^{CFD}(k_s)$ as a function of the TSR and the inflow velocity U_∞ for the four roughness cases (k_s = 0, 20, 40, 60 μm). The interpolation was performed using the Clough-Tocher interpolant introduced in Sect. 2.2. For all cases, the rotor exhibited the typical behaviour of small HAWTs, with a single pronounced C_p^{CFD} maximum around $TSR \approx$ 6-6.5 and a monotonic decay toward both lower and higher TSRs. As roughness increases, the overall shape of the response surfaces was preserved, with the peak slightly shifting toward higher TSRs, and the low-TSR region remaining weakly affected.

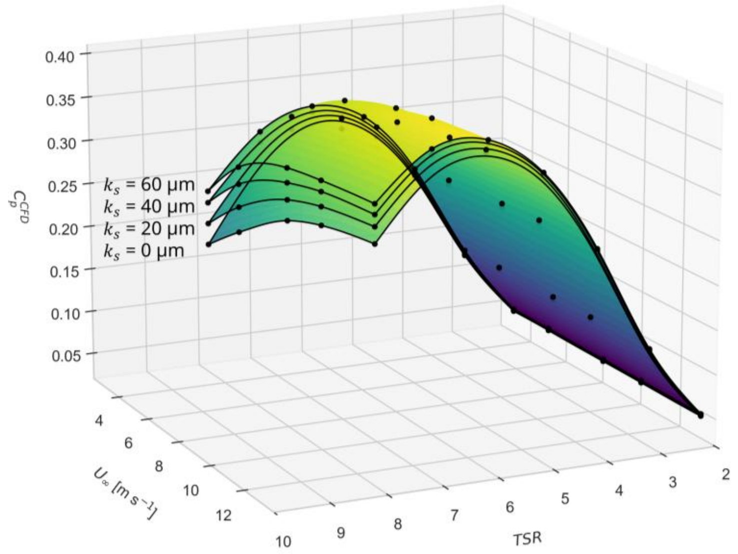

Figure 5.18. CFD response surfaces for the four analysed roughness configurations (adapted from [119]).

As previously introduced in Sect. 5.3.6 for the first DT, at rated operation (800 min^{-1} and 13 m s^{-1}), considering the minimum and maximum chord lengths of 0.027-0.135 m and the relative velocities typical of the operating TSR range yielded chord-based Reynolds numbers of order $Re_c \approx 10^5$, placing the turbine in a low-Reynolds regime. This contrasts with utility-scale HAWTs, which typically operate at $Re_c \approx 10^6$ - 10^7 [135,136]. Therefore, direct transfer of the roughness-response maps across scales is not expected without re-deriving the CFD reference database, although the DT architecture and residual-based inference logic remain applicable.

Table 5.5 summarises the global maxima extracted from the interpolated CFD surfaces. The clean condition (k_s = 0 µm) presented a peak C_p^{CFD} = 0.3639 at TSR = 6.22 and U_∞ = 13 m s^{-1}. Across the analysed roughness cases, the extracted global maxima exhibited moderate variations, with C_p^{CFD} increasing up to 0.3850 for the k_s = 60 µm surface (corresponding to +5.8 % relative to the clean case). The optimum TSR drifted marginally from 6.22 to 6.46 as roughness grew, while the optimal inflow velocity remained at 13 m s^{-1}.

Table 5.5. Global C_p^{CFD} maxima for each roughness surface (interpolated), (adapted from [119]).

k_s [µm]	$C_{p,max}^{CFD}$	$TSR_{C_{p,\max}}$	U_∞ [m s^{-1}]	$\Delta C_{p,max}^{CFD}$ vs clean [%]
0	0.3639	6.221	13.0	0.00
20	0.3727	6.312	13.0	+2.41
40	0.3801	6.402	13.0	+4.45
60	0.3850	6.462	13.0	+5.80

The nondimensional roughness height, k_s^+, was evaluated at rated operation as:

$$k_s^+ = \frac{u_\tau k_s}{\nu} \tag{5.11}$$

following the standard formulations established in [165,166], where u_τ is the friction velocity and ν the kinematic viscosity. Using patch-averaged values, the resulting k_s^+ was about 3.6 for the 20 μm case and 23.7 for k_s = 60 μm. These values indicated that the lowest roughness case lay close to the onset of roughness-induced transition ($k_s^+ \approx$ 2-5), while the highest roughness case fell within the transitionally rough regime, well below fully rough conditions. These findings were consistent with classical rough-wall boundary-layer theory and the roughness treatment adopted in the γ-Re_θ model [165,166].

Although surface roughness is generally associated with drag penalties and reduced power under realistic atmospheric inflow and mature erosion conditions, the present CFD simulations were conducted under steady, idealised inflow assumptions to generate smooth physics-based reference surfaces for DT integration.

Within this controlled numerical framework, the application of leading-edge roughness may act as an effective transition trigger at the low-to-moderate Reynolds numbers of the present rotor, potentially altering laminar separation behaviour in a narrow region around the optimal operating point. However, given the known sensitivity of transitional RANS predictions to inflow turbulence and transition modelling assumptions, the observed increase in the extracted C_p^{CFD} maxima was conservatively interpreted as model-dependent rather than as a general performance benefit of roughness. Importantly, the magnitude of the variation remained moderate, ensuring smooth behaviour across the entire surface family, essential for DT integration [167–169].

Inside the DT architecture, these CFD-derived maps were not meant to reproduce the exact outdoor power curve but to provide a physically informed reference space $C_p^{CFD}(TSR\,, U_\infty, k_s)$ for each roughness state. Accordingly, this behaviour did not compromise the surrogate models or the DT workflow, since inference relied on residual features computed against these reference surfaces, while the experimental datasets retained the expected degradation trends under atmospheric inflow (see Sect. 5.5.6).

Figure 5.19 schematically illustrates the interpolation process, using the clean case as an example. To obtain the virtual C_p^{CFD}, each measured operating point (TSR, U_∞) was projected onto the corresponding CFD surface using the Clough-Tocher interpolant, an operation performed for every roughness configuration analysed numerically.

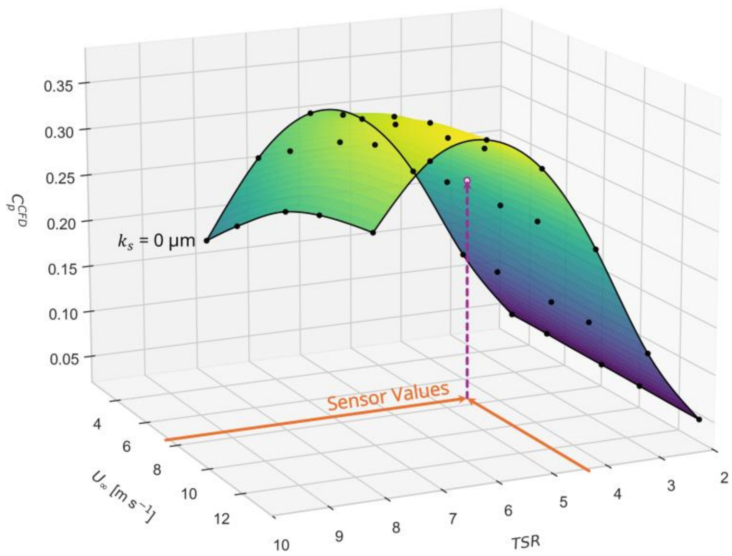

Figure 5.19. Graphical representation of the CFD interpolation process (clean case), (adapted from [119]).

The difference between this reference value and the measured C_p^{meas} formed residual-based features used by the DT to quantify deviations from the nominal aerodynamic behaviour. In this way, the CFD model provided a consistent virtual reference for field data interpretation, while the experimental datasets provided the natural variability observed during outdoor operation.

5.5.6 Experimental performance analysis: outdoor roughness campaign

Figure 5.20 shows the distribution of the measured power coefficient C_p^{meas} as a function of TSR for the *clean₁* test. The raw 1 Hz data were first filtered, retaining only samples with $C_p^{meas} > 0$ and later bounded within the CFD domain ($2 \leq TSR \leq 8$; $3 \leq U_\infty \leq 13$ m s^{-1}). The resulting cloud, coloured by instantaneous wind speed, exhibited the expected bell-shaped trend: for increasing TSR, C_p^{meas} rose from approximately zero to a broad maximum around $TSR \approx$ 5-6 and then decreased again. The colour distribution confirms that higher wind speeds populated the upper part of the cloud, while low-speed points clustered near $C_p^{meas} \approx 0$. This behaviour was consistent with the typical power-curve shape of small HAWTs and indicated that the experimental setup, sensors and processing chain provided physically coherent operating points.

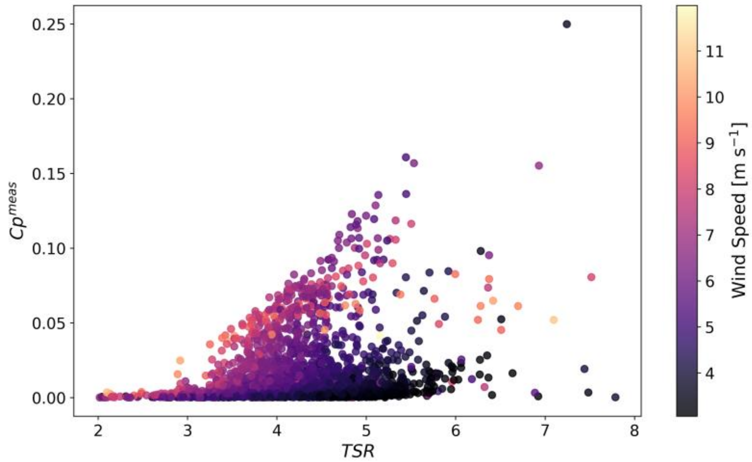

Figure 5.20. CFD-bounded dataset fir the clean₁ test case (adapted from [119]).

A more detailed comparison between the *clean₁* and $P240$ configurations is provided in Figure 5.21. Here, the data were grouped into 1 m s^{-1} wind speed bins. Within each bin, TSR was further discretised with a width of ΔTSR = 0.25, and for every populated TSR bin (more than three samples) the median TSR and median C_p^{meas} were computed together with the corresponding standard deviation. The shaded bands indicate ± one standard deviation around each median.

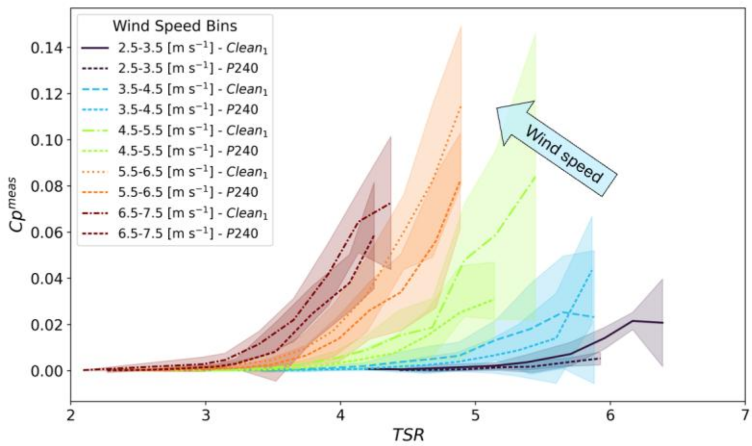

Figure 5.21. Binned power curves for clean₁ and P240 bounded test cases (adapted from [119]).

The clean curves systematically lay above the rough curves across almost the entire operating range, indicating a consistent reduction in C_p^{meas} due to the $P240$ leading-edge roughness. This same behaviour was observed for all other analysed roughness configurations. As wind speed increased, the curve slope became steeper, providing higher power for the same TSR. However, due to the limited duration of the experiments and the intermittent nature of the wind, it was not possible to obtain statistically meaningful median curves at the turbine's rated wind speed. Similarly, points associated with wind velocities greater than about 5 m s^{-1} could not reach TSR values greater than about 5, restricting the observable range of the performance map and justifying the use of CFD to model the expected behaviour of the turbine. The bands for clean and rough data partially overlapped within a given wind speed bin, especially at low C_p^{meas} and in off-design regions.

This overlap highlighted the combined influence of inflow variability and measurement scatter, showing that distinguishing the surface condition directly from C_p^{meas}-TSR median curves is nontrivial. These limitations further justified the integration of machine learning methods within the DT architecture, informed by CFD simulations.

5.5.7 Decision engine development: SVM classification and calibration

The SVM binary classifier was trained to discriminate between clean (*clean$_I$*) and severely rough (*P240*) configurations, which represented the extreme aerodynamic states analysed in CFD (k_s = 0, 60 µm). Two models were initially trained using an automated hyperparameter optimisation, one employing only experimental features: $\{U_\infty, TSR, C_p^{meas}\}$ and another one using hybrid features, which combined the experimental quantities with four residual-based descriptors derived from the CFD reference surfaces, embedding direct information on the deviation of measured performance from the virtual C_p^{CFD} maps: $\{U_\infty, TSR, C_p^{meas}, \frac{\mathrm{res}_0}{TSR}, \frac{\mathrm{res}_{20}}{TSR}, \frac{\mathrm{res}_{40}}{TSR}, \frac{\mathrm{res}_{60}}{TSR}\}$.

Table 5.6. Comparison of SVM performance for different feature sets (clean$_I$, P240), (adapted from [119]).

Feature set	Inputs	Accuracy	F1 (P240)	Unknown fraction (P320)	Unknown Fraction (P800)
Experimental	3	0.784	0.761	13.4 %	8.0 %
Hybrid	7	0.797	0.780	8.2 %	5.5 %

Table 5.6 presents the comparison results of the two trained models. Although the overall accuracy improvement of the hybrid model over the purely experimental one was modest (0.797 vs 0.784), the hybrid classifier achieved markedly better F1-score for the degraded *P240* configuration (0.780 vs 0.761) and a higher recall (0.812 vs 0.772). In practice, this meant that the DT missed 18.8 % of rough cases instead of 22.8 %. This reduction in false negatives is critical from a maintenance standpoint, as it translates to fewer undetected surface-degradation events in operation that would otherwise delay maintenance intervention.

For the *clean$_I$* configuration, the hybrid model maintained similar precision (0.839 vs 0.813) and recall (0.783 vs 0.795), confirming that the added CFD residuals improved roughness detection without sacrificing normal-state reliability.

Consequently, the classifier was evaluated on the intermediate roughness datasets *P320* and *P800*, which were excluded from the training process. For these configurations, the proportion of samples flagged as *unknown fraction*, defined as records whose maximum posterior class probability fell below a confidence threshold, decreased from 13.4 % to 8.2 % for the *P320* state and from 8.0 % to 5.5 % for *P800*, indicating a clear gain in classification confidence and generalisation capability. The confidence threshold was conservatively set as the 5th percentile of the maximum posterior probability distribution computed on correctly classified training samples.

Figure 5.22 shows the confusion matrix of the hybrid SVM evaluated on the held-out test set for the selected classes ($clean_1$, *P240*). The error distribution indicated a reduced number of false negatives for the rough condition compared to false positives for the clean case, reflecting the desired conservative bias of the classifier toward detecting aerodynamic degradation while preserving reliable identification of healthy operation.

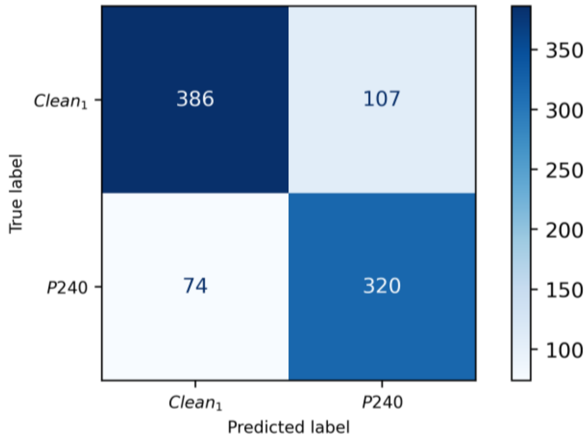

Figure 5.22. Confusion matrix for the hybrid svm classifier (clean1, P240), (adapted from [119]).

Figure 5.23 reports the receiver operating characteristic (ROC) curve for the hybrid SVM on the held-out test set for the selected classes ($clean_1$, *P240*). The curve lay well above the random-classifier diagonal, yielding an area under the curve (AUC) of 0.858, thus indicating good class separability across a wide range of probability thresholds. This validated that the classifier retained discriminative power beyond a single operating point and supported the use of calibrated posterior probabilities for continuous degradation assessment, as required for the implementation of the RI.

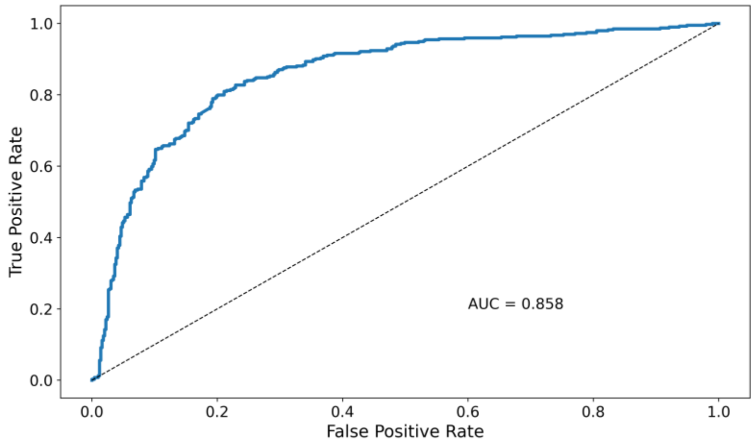

Figure 5.23. ROC curve illustrating classifier separability across probability thresholds (adapted from [119]).

Figure 5.24 shows the reliability diagram for the calibrated hybrid SVM, representing the model's confidence in identifying the rough condition. This plot compares the predicted probability of the *P240* class with the observed fraction of true rough cases in each probability bin. A perfectly calibrated model lies on the diagonal (dashed line), meaning that its predicted probabilities correspond to the real frequency of each event (e.g., a 0.7 prediction means the event occurred in 70 % of such cases). The reliability curve indicated mild under-prediction for clean states and slight over-prediction for rough ones, reflecting a conservative yet well-calibrated behaviour suitable for condition monitoring.

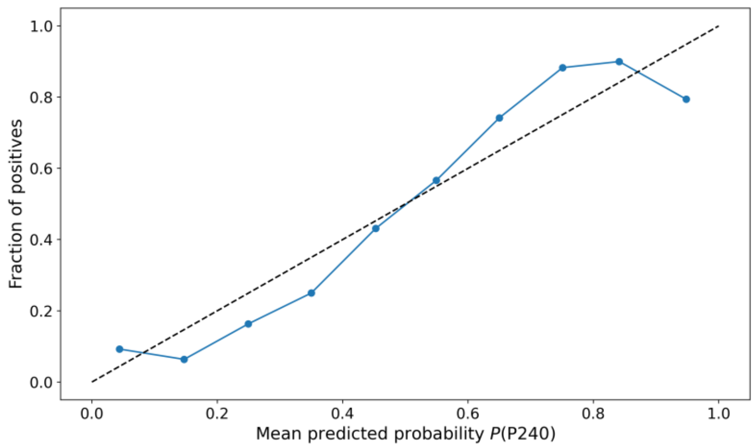

Figure 5.24. Hybrid SVM reliability diagram (adapted from [119]).

The calibration quality was assessed using the Brier score and the Expected Calibration Error (ECE), which quantified the mean squared difference between predicted probabilities and true labels (ranging from 0 to 1) and the average deviation between predicted confidence and true event frequency, respectively. The final hybrid model achieved a Brier score of 0.155 and an ECE of 0.0739, while the experimental model obtained 0.161 and 0.0670, respectively.

In summary, together these results justified the use of the CFD-informed model, with predicted probabilities well aligned with the true frequencies of each class, essential for a robust condition-monitoring DT.

5.5.8 Intelligent layer inference: roughness index and decision logic

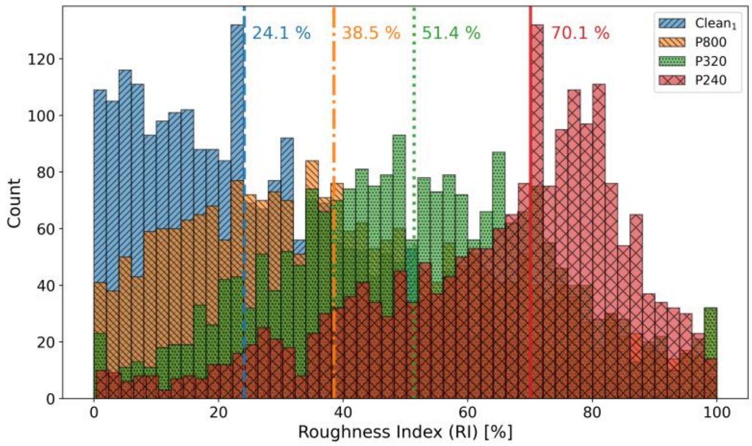

Figure 5.25. RI distributions and medians for the test cases clean$_1$, P800, P240 (adapted from [119]).

Figure 5.25 shows the RI distributions computed for the *clean$_1$, P800, P320* and *P240* experimental configurations using the calibrated hybrid SVM probabilities of the *P240* class.

The RI represents the likelihood that a given operating state corresponds to the severely rough configuration. Its values increased monotonically from clean to progressively rough conditions, confirming that this index can be successfully utilised in the DT architecture to capture the aerodynamic impact of surface deterioration.

The median RI values extracted from each distribution were approximately 24.1 %, 38.5 %, 51.4 % and 70.1 % for the *clean$_1$*, *P800*, *P320* and *P240* datasets, respectively. Despite partial overlap between adjacent states, as expected given the natural experimental variability, the separation between medians remained consistent, demonstrating that the RI can distinguish incremental levels of roughness with statistical robustness.

As introduced in Sect. 5.5.4, from these distributions, two operational decision thresholds were derived by averaging the medians of consecutive states: $RI_{\text{minimal→moderate}}$ = 31.30 % and $RI_{\text{moderate→critical}}$ = 60.75 %. These thresholds corresponded to the three previously defined feedback levels: *minimal*, *moderate* and *critical surface* roughness, later provided by the DT adapted from the evaluated median RI value of the dataset.

Figure 5.26 presents similar information as box plots, visually linking the three feedback regions (green-yellow-red) with the statistical spread of the RI per roughness configuration. The monotonic increase of both median and interquartile range with roughness confirms that the RI acts as a continuous degradation metric, rather than a discrete classifier output.

This representation enabled the DT to move from categorical fault detection to condition-based maintenance, providing interpretable, probabilistic guidance on the aerodynamic health of the turbine.

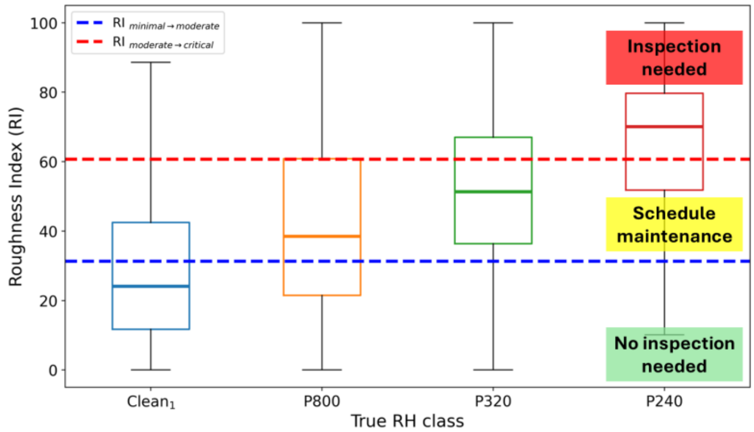

Figure 5.26. Box plots for the test cases clean₁, P800, P320, P240 and the defined thresholds (adapted from [119]).

5.5.9 Digital Twin diagnostic performance and operational perspective

The final validation stage involved three dedicated test-day tests: *clean₂*, *P1200* and *P100*, representing a progressive accumulation of surface roughness. Each dataset contained approximately 2000 valid 1 Hz samples across the seven input features used by the SVM. The corresponding *RI* distributions are shown in Figure 5.27. The median *RI* values were 26.6 % for *clean₂*, 29.1 % for *P1200* and 61.8 % for *P100*, with narrow 95 % confidence intervals (± 1 %). Both *clean₂* and *P1200* fell below the first decision threshold ($RI_{\text{minimal→moderate}}$ = 31.30 %), correctly classified as *minimally rough*, without the need for an inspection. At the same time, *P100* exceeded the alarm limit ($RI_{\text{moderate→critical}}$ = 60.75 %), triggering an *inspection-needed* status, while remaining well below full saturation.

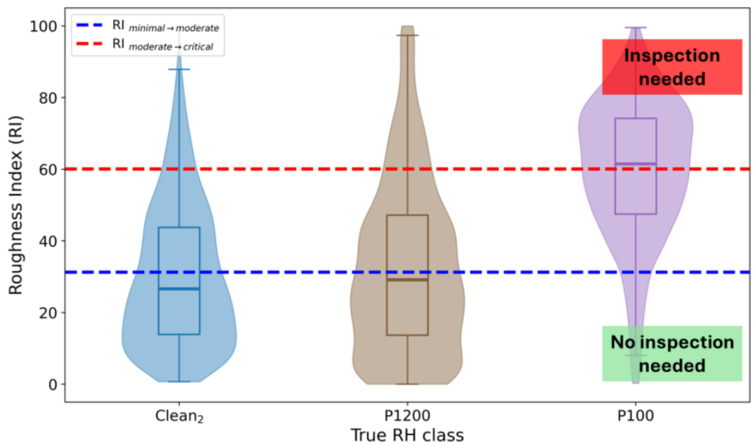

Figure 5.27. Violin-box plots of the three test cases (clean₂, P1200, P100), (adapted from [119]).

Despite $P100$ corresponding to a sand-grain diameter (D50% = 156 μm) well beyond the CFD training range ($k_s \leq 60$ μm), the model exhibited stable and conservative inference behaviour. Once the aerodynamic deviation became comparable to that of the severely degraded reference state ($P240$), further increases in physical roughness may not have induced a proportional change in the extracted residual features, leading to a stabilisation of the median RI. Within this regime, the RI therefore acted as a bounded indicator of critical aerodynamic degradation rather than as a continuously calibrated severity scale.

From a CM standpoint, these tests demonstrated the DT's capacity to autonomously assess aerodynamic health at the end of each operating day, integrating data-driven inference with physics-based performance maps. Importantly, the $P100$ case clarified the generalisation limits of the proposed approach. Generalisation is therefore expected within the aerodynamic response space spanned by the CFD-informed hybrid features, rather than across arbitrary erosion morphologies or absolute roughness magnitudes.

The DT was intentionally designed to detect and act on early-stage LEE, where surface degradation can be reasonably approximated by distributed equivalent roughness. Beyond this envelope, as illustrated by the $P100$ case, the model did not extrapolate degradation severity linearly with increasing roughness amplitude. Instead, it transitioned into a conservative regime that reliably flagged a critical condition without over-interpreting severity.

Finally, although the present work validated the DT in steady monitoring mode, the same RI trend could be tracked over time to forecast future degradation, thereby easily extending the framework toward predictive maintenance capability.

5.6 Chapter summary

This chapter presented the development of DT-based monitoring approaches for aerodynamic fault detection in wind turbines operating under real atmospheric conditions. The proposed DTs were experimentally validated on a small-scale HAWT, while the methodology targeted aerodynamic performance behaviour rather than turbine size.

Building upon the diagnostic principles introduced in Ch. 4, the proposed models were directly deployed on an existing turbomachine and combined CFD-derived aerodynamic references with operational measurements to enable CFD-informed fault inference.

A common DT architecture was first introduced, structured into four interacting layers: virtual, physical, data and intelligent. Within this framework, steady-state CFD simulations provided the aerodynamic reference space, while operational measurements were translated into dimensionless performance indicators enabling comparison between measured and predicted turbine behaviour.

The first DT implementation addressed static yaw misalignment induced by faulty wind direction measurements. By embedding the yaw angle directly as a parameter within the

CFD performance database, the DT was able to infer static misalignment through inverse interpolation of aerodynamic response surfaces. Outdoor validation experiments demonstrated that this approach could detect injected yaw errors up to 15° and restore aligned conditions with correction errors typically within 0-3°. The results confirmed that CFD-informed performance inversion can act as a performance-based virtual sensor for yaw alignment, enabling corrective control actions without requiring prior training on faulty datasets.

The experimental analysis revealed two primary limitations. First, ground-level atmospheric turbulence, driven by site constraints that required the turbine to operate close to the ground, exposed the rotor to elevated shear and increased signal variability. Second, the yaw DT's explicit parametric embedding of γ, while effective for deterministic faults, was less directly applicable to progressive degradation mechanisms such as LEE, which produce distributed aerodynamic effects whose performance signatures may partially overlap under real inflow variability.

These observations motivated the development of a second DT implementation targeting early-stage LEE. To address this challenge, the second DT adopted a hybrid formulation combining CFD-derived aerodynamic references with a data-driven classification model. Residual features extracted from discrepancies between measured and CFD-predicted performance were processed through a supervised classifier to distinguish between clean and degraded states. This architecture enabled early-stage erosion monitoring and the derivation of a scalar health indicator supporting maintenance-oriented decision-making rather than direct corrective control.

Despite the differences in fault characteristics and inference strategies, both DT implementations were developed within the same architectural framework. The progression from deterministic performance inversion to hybrid physics-data inference illustrated the adaptability of the proposed DT architecture across fault classes with distinct temporal and operational characteristics.

Overall, the results demonstrated that CFD-informed DTs can support both corrective diagnostics and degradation monitoring in wind turbines operating under realistic atmospheric conditions. By integrating aerodynamic knowledge directly within the monitoring framework, the proposed approach achieved interpretable fault detection while remaining resilient to measurement uncertainty and inflow variability.

Ultimately, the methodological evolution presented in this chapter illustrated how DT architectures can progressively integrate physics-based modelling and data-driven inference to address increasingly complex aerodynamic faults, contributing to more robust condition monitoring strategies for wind energy systems.

6. Conclusions

6.1 Summary of the research work

This thesis addressed the development of methods for condition monitoring and diagnostics of turbomachinery operating under industrial constraints. The research was conducted within an industrial PhD programme in collaboration with the Istituto Italiano della Saldatura (IIS) and Cranfield University.

The work followed a progressive methodological trajectory combining Computational Fluid Dynamics (CFD), signal analysis and data-driven techniques, ultimately leading to the development of Digital Twin (DT) diagnostic architectures capable of supporting monitoring and maintenance decision-making.

The early part of the research focused on the development of diagnostic expertise within a controlled numerical environment. Radial compressors were selected as the first application domain, where time-resolved CFD simulations enabled the analysis of static pressure signals associated with aerodynamic instabilities. In this context, surge was interpreted from a maintenance perspective as a fault mechanism capable of producing severe performance degradation and potentially damaging operating conditions.

Machine-learning models were developed to reconstruct internal flow behaviour from limited inducer pressure measurements. This reconstruction allowed for the classification of the stage operating state regarding instability, for both instantaneous and short-term predictive assessments. To enhance this framework, an approximate entropy (ApEn) approach was integrated to identify the monitoring locations most sensitive to the onset of surge within the stage diffusers, providing a quantitative foundation for sensor placement optimization.

The methodological knowledge obtained from this stage provided the foundation for the second part of the thesis, where the research focus shifted toward wind turbines. Unlike the compressor study, this application enabled experimental validation under real operating conditions. Two DT architectures were therefore developed for a small-scale horizontal-axis wind turbine, integrating CFD-derived aerodynamic performance models with real-time operational measurements.

The DTs demonstrated the capability to detect and diagnose aerodynamic performance deviations, including yaw misalignment caused by sensor faults and blade surface degradation due to early-stage leading-edge erosion. Importantly, the DTs were implemented not only for monitoring purposes but also to provide maintenance-oriented feedback, supporting operational decision-making.

Through this progression, the thesis demonstrates how CFD-informed monitoring methodologies can evolve from diagnostic indicators developed in numerical environments to DT systems capable of operating on real machines. It is worth emphasising that, across both application domains, this coupling was CFD-informed rather than physics-informed in the strict sense: CFD served as the source of training data and as the forward model inverted to infer operating conditions, rather than entering the learning process as an embedded physical constraint.

6.2 Main scientific contributions

The research presented in this thesis contributes to the field of turbomachinery condition monitoring in several ways.

a) The thesis develops and validates signal-based methods, informed by CFD, for aerodynamic instability detection in radial compressors. The results demonstrate that signal data-driven models and complexity metrics can successfully detect instability signatures within pressure signals, providing a basis for non-intrusive monitoring strategies.
b) The research implements two CFD-informed DTs for wind turbine diagnostics, where aerodynamic performance maps generated from numerical simulations are integrated with real-time sensor measurements. By mapping measured operating points onto performance surfaces, this approach enables the inference of performance deviations without extensive historical fault data, offering a physically interpretable alternative to purely data-driven monitoring models.
c) The proposed DT architectures are experimentally demonstrated under real outdoor operating conditions. Specifically, the first DT successfully identifies yaw misalignment caused by faulty wind direction measurements and enables correction of the inferred misalignment with errors within 0-3°, restoring aerodynamic performance. This demonstrates the capability of CFD-informed DTs to diagnose sensor-induced faults that would otherwise compromise turbine operation.
d) On this basis, the thesis develops a second DT for detecting aerodynamic degradation associated with blade surface roughness representative of early-stage leading-edge erosion. By combining CFD-derived performance models with machine-learning classification, the DT enables indirect inference of surface degradation from operational measurements. The resulting Roughness Index provides a continuous and interpretable indicator of aerodynamic health, enabling actionable guidance for maintenance decision-making under realistic operational uncertainty.

6.3 Limitations of the present research

Despite the promising results obtained, several limitations should be acknowledged. The compressor diagnostics developed in Ch. 4 were validated primarily using CFD-generated data rather than experimental measurements. Although the numerical simulations allowed controlled analysis of instability phenomena, direct experimental validation on industrial compressors would further strengthen the applicability of the proposed diagnostic indicators.

In the wind turbine studies of Ch. 5, the DT architectures were demonstrated on a small-scale turbine under outdoor conditions. While the methodological principles are scalable, the quantitative transfer of the results to utility-scale turbines would require the generation of aerodynamic performance maps at appropriate Reynolds numbers and operating regimes.

Additionally, the DTs developed in this thesis operate under a quasi-steady framework, relying on time-averaged measurements rather than fully transient aerodynamic modelling. This approach is suitable for detecting persistent performance deviations but may require further development for highly dynamic operating conditions.

6.4 Future research directions

The results obtained in this thesis suggest several promising directions for future work. One important development concerns the experimental validation of compressor diagnostic indicators using real measurement data. Such studies would allow the assessment of signal-based monitoring techniques under realistic noise levels and sensor limitations.

A second research direction involves the extension of the DT architectures toward multi-fault diagnostic frameworks, capable of simultaneously identifying multiple degradation mechanisms affecting turbomachinery performance.

Furthermore, the adaptation and testing of the proposed DT architectures for utility-scale wind turbines would be a significant step toward validating the scalability of these models within the global energy infrastructure.

Finally, future developments may explore the implementation of real-time DT monitoring systems**,** integrating edge computing architectures and distributed sensor networks to enable continuous diagnostics and predictive maintenance in industrial turbomachinery installations.

Acknowledgements

Bana gerçek aşkı tanıtan kadına

Bibliography

[1] Olesen, J. F., and Shaker, H. R., 2020, "Predictive Maintenance for Pump Systems and Thermal Power Plants: State-of-the-Art Review, Trends and Challenges," Sensors (Switzerland), **20**(8). https://doi.org/10.3390/S20082425.

[2] Achouch, M., Dimitrova, M., Dhouib, R., Ibrahim, H., Adda, M., Sattarpanah Karganroudi, S., Ziane, K., and Aminzadeh, A., 2023, "Predictive Maintenance and Fault Monitoring Enabled by Machine Learning: Experimental Analysis of a TA-48 Multistage Centrifugal Plant Compressor," Applied Sciences (Switzerland), **13**(3). https://doi.org/10.3390/APP13031790.

[3] Orrù, P. F., Zoccheddu, A., Sassu, L., Mattia, C., Cozza, R., and Arena, S., 2020, "Machine Learning Approach Using MLP and SVM Algorithms for the Fault Prediction of a Centrifugal Pump in the Oil and Gas Industry," Sustainability (Switzerland), **12**(11). https://doi.org/10.3390/SU12114776.

[4] Cappugi, L., Castorrini, A., Bonfiglioli, A., Minisci, E., and Campobasso, M. S., 2021, "Machine Learning-Enabled Prediction of Wind Turbine Energy Yield Losses Due to General Blade Leading Edge Erosion," Energy Convers. Manag., **245**. https://doi.org/10.1016/J.ENCONMAN.2021.114567.

[5] Alqallaf, J., and Teixeira, J. A., 2021, "Blade Roughness Effects on Compressor and Engine Performance—a Cfd and Thermodynamic Study," Aerospace, **8**(11). https://doi.org/10.3390/AEROSPACE8110330.

[6] Tahan, M., Tsoutsanis, E., Muhammad, M., and Abdul Karim, Z. A., 2017, "Performance-Based Health Monitoring, Diagnostics and Prognostics for Condition-Based Maintenance of Gas Turbines: A Review," Appl. Energy, **198**, pp. 122–144. https://doi.org/10.1016/J.APENERGY.2017.04.048.

[7] Sepe, M., Graziano, A., Badora, M., Di Stazio, A., Bellani, L., Compare, M., and Zio, E., 2021, "A Physics-Informed Machine Learning Framework for Predictive Maintenance Applied to Turbomachinery Assets," Journal of the Global Power and Propulsion Society, **2021**(Special Issue). https://doi.org/10.33737/JGPPS/134845.

[8] Abdelrhman, A. M., Hee, L. M., Leong, M. S., and Al-Obaidi, S., 2014, "Condition Monitoring of Blade in Turbomachinery: A Review," Advances in Mechanical Engineering, **6**, p. 210717. https://doi.org/10.1155/2014/210717.

[9] Molęda, M., Małysiak-Mrozek, B., Ding, W., Sunderam, V., and Mrozek, D., 2023, "From Corrective to Predictive Maintenance—A Review of Maintenance Approaches for the Power Industry," Sensors (Basel), **23**(13), p. 5970. https://doi.org/10.3390/S23135970.

[10] Zhu, T., Ran, Y., Zhou, X., and Wen, Y., 2024, "A Survey of Predictive Maintenance: Systems, Purposes and Approaches." https://doi.org/10.1109/COMST.2025.3567802.

[11] Ahmed Murtaza, A., Saher, A., Hamza Zafar, M., Kumayl Raza Moosavi, S., Faisal Aftab, M., and Sanfilippo, F., 2024, "Paradigm Shift for Predictive Maintenance and Condition Monitoring from Industry 4.0 to Industry 5.0: A Systematic Review, Challenges and Case Study," Results in Engineering, **24**, p. 102935. https://doi.org/10.1016/J.RINENG.2024.102935.

[12] Garcia, J., Rios-Colque, L., Peña, A., and Rojas, L., 2025, "Condition Monitoring and Predictive Maintenance in Industrial Equipment: An NLP-Assisted Review of Signal Processing, Hybrid Models, and Implementation Challenges," Applied Sciences (Switzerland), **15**(10), p. 5465. https://doi.org/10.3390/APP15105465/S1.

[13] American Petrolium Intitute, 2025, *API STD 670: Machinery Protection Systems*. [Online]. Available: https://www.apiwebstore.org/standards/670. [Accessed: 11-Nov-2025].

[14] Tiboni, M., Remino, C., Bussola, R., and Amici, C., 2022, "A Review on Vibration-Based Condition Monitoring of Rotating Machinery," Applied Sciences 2022, Vol. 12, Page 972, **12**(3), p. 972. https://doi.org/10.3390/APP12030972.

[15] Mohanta, R. K., Chelliah, T. R., Allamsetty, S., Akula, A., and Ghosh, R., 2017, "Sources of Vibration and Their Treatment in Hydro Power Stations-A Review," Engineering Science and Technology, an International Journal, **20**(2), pp. 637–648. https://doi.org/10.1016/j.jestch.2016.11.004.

[16] Zargar, O. A., 2014, "Vibration Analysis of Gas Turbine SIEMENS 162MW- V94.2 Related to Iran Power Plant Industry in Fars Province," Journal of Mechanical Design and Vibration, **2**(1), pp. 1–10. https://doi.org/10.12691/JMDV-2-1-1.

[17] Girdhar, P., and Scheffer, C., 2004, *Practical Machinery Vibration Analysis and Predictive Maintenance*, Elsevier. https://doi.org/10.1016/B978-0-7506-6275-8.X5000-0.

[18] Wang, K., Wang, Y., Luo, D., and He, Q., 2015, "Application of Order Tracking Rationale to Non-Rotating Cantilever Beam Analysis," 2014 International Conference on Prognostics and Health Management, PHM 2014. https://doi.org/10.1109/ICPHM.2014.7036365.

[19] Fyfe, K. R., and Munck, E. D. S., 1997, "ANALYSIS OF COMPUTED ORDER TRACKING," Mech. Syst. Signal Process., **11**(2), pp. 187–205. https://doi.org/10.1006/MSSP.1996.0056.

[20] Tsoutsanis, E., Meskin, N., Benammar, M., and Khorasani, K., 2014, "A Component Map Tuning Method for Performance Prediction and Diagnostics of Gas Turbine Compressors," Appl. Energy, **135**, pp. 572–585. https://doi.org/10.1016/J.APENERGY.2014.08.115.

[21] Rompokos, P., Aretakis, N., Roumeliotis, I., and Mathioudakis, K., 2020, "Application of an Advanced Adaptation Methodology for Gas Turbine Performance Monitoring." https://doi.org/10.33737/GPPS20-TC-92.

[22] Sanaye, S., and Hosseini, S., 2020, "Prediction of Blade Life Cycle for an Industrial Gas Turbine at Off-Design Conditions by Applying Thermodynamics, Turbo-Machinery and Artificial Neural Network Models," Energy Reports, **6**, pp. 1268–1285. https://doi.org/10.1016/J.EGYR.2020.05.008.

[23] Liang, X., Duan, F., Bennett, I., and Mba, D., 2021, "A Comprehensive Health Indicator Integrated by the Dynamic Risk Profile from Condition Monitoring Data and the Function of Financial Losses," Energies (Basel)., **14**(1). https://doi.org/10.3390/EN14010028.

[24] Saubanov, O., Valeev, A., and Akimov, V., 2020, "Developing a Complex Approach to the Definition of a Technical Condition of Pump-and-Compressor Equipment," IOP Conf. Ser. Earth Environ. Sci., **459**(2). https://doi.org/10.1088/1755-1315/459/2/022082.

[25] Alsuwian, T., Amin, A. A., Iqbal, M. S., and Maqsood, M. T., 2023, "A Review of Anti-Surge Control Systems of Compressors and Advanced Fault-Tolerant Control Techniques for Integration Perspective," Heliyon, **9**(9), p. e19557. https://doi.org/10.1016/J.HELIYON.2023.E19557.

[26] Mlkvik, M., Olšiak, R., and Knížat, B., 2023, "A Method of Stall Recognition Using Nonlinear Feature Extraction from the Compressor Outlet Pressure," Heliyon, **9**(10). https://doi.org/10.1016/J.HELIYON.2023.E20909.

[27] Amin, A. A., Maqsood, M. T., and Mahmood-ul-Hasan, K., 2021, "Surge Protection of Centrifugal Compressors Using Advanced Anti-Surge Control System," Measurement and Control (United Kingdom), **54**(5–6), pp. 967–982. https://doi.org/10.1177/0020294020983372.

[28] Silvestri, P., Marelli, S., and Usai, V., 2023, "Experimental Investigation on the Vibrational and Fluid Dynamics Behaviour of a Turbocharger Compressor in the Transition to Surge Operation," E3S Web of Conferences, **414**. https://doi.org/10.1051/E3SCONF/202341402016.

[29] Liu, A. X., and Zheng, X. Q., 2013, "Methods of Surge Point Judgment for Compressor Experiments," Exp. Therm. Fluid Sci., **51**, pp. 204–213. https://doi.org/10.1016/J.EXPTHERMFLUSCI.2013.07.015.

[30] Towler, M. D., Allison, T., Krueger, P., and Wygant, K., 2018, "A Novel Approach to Surge Control: High-Frequency Pressure Variance As an Indicator of Impending Surge in Centrifugal Compressors," Volume 9: Oil and Gas Applications; Supercritical CO2 Power Cycles; Wind Energy, **9**. https://doi.org/10.1115/GT2018-77222.

[31] Orkisz, M., and Lipnicki, P., 2014, "Rotating Compressor Surge Detection Using Variable Speed Drive Signals," 2014 International Symposium on Power Electronics, Electrical Drives, Automation and Motion, pp. 455–460. https://doi.org/10.1109/SPEEDAM.2014.6871978.

[32] SOLIMAN, M. H. AHMED., 2020, "MACHINERY OIL ANALYSIS & CONDITION MONITORING : A Practical Guide to Sampling and Analyzing Oil... to Improve Equipment Reliability." [Online]. Available: https://www.researchgate.net/publication/346965794_Machinery_Oil_Analysis_Condition_Monitoring_A_Practical_Guide_to_Sampling_and_Analyzing_Oil_to_Improve_Equipment_Reliability. [Accessed: 12-Nov-2025].

[33] "Steam-Turbine-Oil-Analysis-Practices." [Online]. Available: https://www.minimacsystems.com/steam-turbine-oil-analysis-practices?srsltid=AfmBOoqtX6OprV3demq6DWaO_oVJGIpgM6pjYUiD6JmsgkuKWTeWb9Jy. [Accessed: 27-Nov-2025].

[34] Kuoppala, R. J., Jantunen, E. O., and Enwald, P. A., 2018, "Condition Monitoring Methods for Rotating Machinery," Operational Reliability and Systematic Maintenance, pp. 175–198. https://doi.org/10.1201/9781351075183-10.

[35] Sun, J., Wang, L., Li, J., Li, F., Li, J., and Lu, H., 2021, "Online Oil Debris Monitoring of Rotating Machinery: A Detailed Review of More than Three Decades," Mech. Syst. Signal Process., **149**. https://doi.org/10.1016/J.YMSSP.2020.107341.

[36] Giuliani, F., Andracher, L., Moosbrugger, V., Paulitsch, N., and Hofer, A., 2020, "Combined Optic-Acoustic Monitoring of Combustion in a Gas Turbine," International Journal of Turbomachinery, Propulsion and Power 2020, Vol. 5, Page 15, **5**(3), p. 15. https://doi.org/10.3390/IJTPP5030015.

[37] Momeni, S., Koduru, J. P., Gonzalez, M., Zarate, B., and Godinez, V., 2013, "Online Acoustic Emission Monitoring of Combustion Turbines for Compressor Stator Vane Crack Detection," Industrial and Commercial Applications of Smart Structures Technologies 2013, **8690**, p. 86900B. https://doi.org/10.1117/12.2013870.

[38] Shahkar, S., and Khorasani, K., 2019, "Gas Turbine Condition Monitoring Using Acoustic Emission Signals," J. Nondestruct. Eval. Diagn. Progn. Eng. Syst., **2**(3). https://doi.org/10.1115/1.4044232.

[39] Basseville, M., and Nikiforov, I. V, 1993, *Detection of Abrupt Change Theory and Application*. https://doi.org/10.1016/0967-0661(94)90196-1.

[40] Isermann, R., 2006, "Fault-Diagnosis Systems: An Introduction from Fault Detection to Fault Tolerance," Fault-Diagnosis Systems: An Introduction from Fault Detection to Fault Tolerance, pp. 1–475. https://doi.org/10.1007/3-540-30368-5/COVER.

[41] Vili Panov, 2013, "MODEL-BASED CONTROL AND DIAGNOSTIC TECHNIQUES FOR OPERATIONAL IMPROVEMENTS OF GAS TURBINE

ENGINES." [Online]. Available: https://www.researchgate.net/publication/279574637_MODEL-BASED_CONTROL_AND_DIAGNOSTIC_TECHNIQUES_FOR_OPERATIONAL_IMPROVEMENTS_OF_GAS_TURBINE_ENGINES. [Accessed: 12-Nov-2025].

[42] Chen, J., and Patton, R. J., 1999, *Robust Model-Based Fault Diagnosis for Dynamic Systems*, Springer US, Boston, MA. https://doi.org/10.1007/978-1-4615-5149-2.

[43] Tsoutsanis, E., Meskin, N., Benammar, M., and Khorasani, K., 2014, "A Component Map Tuning Method for Performance Prediction and Diagnostics of Gas Turbine Compressors," Appl. Energy, **135**, pp. 572–585. https://doi.org/10.1016/J.APENERGY.2014.08.115.

[44] Simon, D., 2006, *Optimal State Estimation: Kalman, H∞, and Nonlinear Approaches*, Wiley Blackwell. https://doi.org/10.1002/0470045345.

[45] Volponi, A. J., DePold, H., Ganguli, R., and Daguang, C., 2003, "The Use of Kalman Filter and Neural Network Methodologies in Gas Turbine Performance Diagnostics: A Comparative Study," J. Eng. Gas Turbine. Power, **125**(4), pp. 917–924. https://doi.org/10.1115/1.1419016.

[46] Kobayashi, T., and Simon, D. L., 2009, "Application of a Bank of Kalman Filters for Aircraft Engine Fault Diagnostics," American Society of Mechanical Engineers, International Gas Turbine Institute, Turbo Expo (Publication) IGTI, **1**, pp. 461–470. https://doi.org/10.1115/GT2003-38550.

[47] Wehrel, P., and Carvalho, F., 2024, "Cooling Model Calibration in a Collaborative Turbine Preliminary Design Process Using the NASA Energy Efficient Engine Part I: 0D Performance Modeling," International Journal of Gas Turbine, Propulsion and Power Systems, **15**(3), pp. 18–30. https://doi.org/10.38036/JGPP.15.3_18.

[48] Saravanamuttoo, H., Roger, G. F. C., Cohen, H., and Straznicky P.V., 2009, *Gas Turbine Theory*, Pearson Education, London. [Online]. Available: https://www.scirp.org/reference/referencespapers?referenceid=2118693. [Accessed: 12-Nov-2025].

[49] Stamatis, A., Mathioudakis, K., and Papailiou, K., 1992, "Optimal Measurement and Health Index Selection for Gas Turbine Performance Status and Fault Diagnosis," J. Eng. Gas Turbine. Power, **114**(2), pp. 209–216. https://doi.org/10.1115/1.2906574.

[50] Wu, X., Liu, B., Ricks, N., and Ghorbaniasl, G., 2019, "Surrogate Models for Performance Prediction of Axial Compressors Using Through-Flow Approach," Energies 2020, Vol. 13, Page 169, **13**(1), p. 169. https://doi.org/10.3390/EN13010169.

[51] Aversano, G., Bellemans, A., Li, Z., Coussement, A., Gicquel, O., and Parente, A., 2019, "Application of Reduced-Order Models Based on PCA & Kriging for the

Development of Digital Twins of Reacting Flow Applications," Comput. Chem. Eng., **121**, pp. 422–441. https://doi.org/10.1016/J.COMPCHEMENG.2018.09.022.

[52] Randall, R. B., 2010, *Vibration-Based Condition Monitoring: Industrial, Aerospace and Automotive Applications*, wiley. https://doi.org/10.1002/9780470977668.

[53] Silvestri, P., Niccolini Marmont Du Haut Champ, C. A., Reggio, F., Ferrari, M. L., and Massardo, A. F., 2023, "Vibro-Acoustic Responses and Pressure Signal Analysis for Early Surge Detection in a Turbocharger Compressor," Proceedings of the ASME Turbo Expo, **5**. https://doi.org/10.1115/GT2023-101699.

[54] Stajuda, M., Cava, D. G., and Liśkiewicz, G., 2021, "Cyclostationary Approach for Instabilities Detection and Condition Monitoring of Centrifugal Compressor," Proceedings of the ASME Turbo Expo, **2E-2020**. https://doi.org/10.1115/GT2020-15699.

[55] Liu, T., and Wu, Z., 2015, "A Vibration Analysis Based on Wavelet Entropy Method of a Scroll Compressor," Entropy 2015, Vol. 17, Pages 7076-7086, **17**(10), pp. 7076–7086. https://doi.org/10.3390/E17107076.

[56] Zhang, X., Wang, B., and Chen, X., 2015, "Operational Safety Assessment of Turbo Generators with Wavelet Rényi Entropy from Sensor-Dependent Vibration Signals," Sensors 2015, Vol. 15, Pages 8898-8918, **15**(4), pp. 8898–8918. https://doi.org/10.3390/S150408898.

[57] Pincus, S. M., 1991, "Approximate Entropy as a Measure of System Complexity," Proc. Natl. Acad. Sci. U. S. A., **88**(6), pp. 2297–2301. https://doi.org/10.1073/PNAS.88.6.2297.

[58] Yan, R., and Gao, R. X., 2007, "Approximate Entropy as a Diagnostic Tool for Machine Health Monitoring," Mech. Syst. Signal Process., **21**(2), pp. 824–839. https://doi.org/10.1016/J.YMSSP.2006.02.009.

[59] Martin, E. B., Morris, A. J., and Zhang, J., 1996, "Process Performance Monitoring Using Multivariate Statistical Process Control," IEE Proceedings: Control Theory and Applications, **143**(2), pp. 132–144. https://doi.org/10.1049/IP-CTA:19960321.

[60] Bounoua, W., Benkara, A. B., Kouadri, A., and Bakdi, A., 2020, "Online Monitoring Scheme Using Principal Component Analysis through Kullback-Leibler Divergence Analysis Technique for Fault Detection," Transactions of the Institute of Measurement and Control, **42**(6), pp. 1225–1238. https://doi.org/10.1177/0142331219888370.

[61] Imran, M., Dai, H. L., Zaidi, F. S., Tran, K. P., Abbas, Z., and Nazir, H. Z., 2023, "Incorporating Principal Component Analysis into Hotelling T2 Control Chart for Compositional Data Monitoring," Comput. Ind. Eng., **186**. https://doi.org/10.1016/J.CIE.2023.109755.

[62] MacGregor, J. F., and Kourti, T., 1995, "Statistical Process Control of Multivariate Processes," Control Eng. Pract., **3**(3), pp. 403–414. https://doi.org/10.1016/0967-0661(95)00014-L.

[63] Capelli, L., Massaccesi, G., Molano, J. C. C., Campo, F., Borghi, D., Rubini, R., and Cocconcelli, M., 2022, "A Structured Approach to Machine Learning Condition Monitoring," Applied Condition Monitoring, **19**, pp. 33–54. https://doi.org/10.1007/978-3-030-79519-1_3.

[64] Marfo, W., Tosh, D. K., and Moore, S. V., 2022, "Condition Monitoring and Anomaly Detection in Cyber-Physical Systems," 2022 17th Annual System of Systems Engineering Conference (SOSE), pp. 106–111. https://doi.org/10.1109/SOSE55472.2022.9812638.

[65] Sodemann, A., Li, Y., Lee, J., Lancaster, R., Rucker, M., Carnagie, J., and Inman, A., 2006, "Data-Driven Surge Map Modeling for Centrifugal Air Compressors," American Society of Mechanical Engineers, Dynamic Systems and Control Division (Publication) DSC. https://doi.org/10.1115/IMECE2006-13103.

[66] Akbar, G., and Dhini, A., 2023, "Fault Detection Model of Rotating Machinery Using Machine Learning: Case Study of Oil and Gas Company," 2023 15th International Conference on Software, Knowledge, Information Management and Applications (SKIMA), pp. 29–33. https://doi.org/10.1109/SKIMA59232.2023.10387361.

[67] Yusoff, Mohd. D., Ooi, C. S., Lim, M. H., and Leong, Mohd. S., 2019, "A Hybrid K-Means-GMM Machine Learning Technique for Turbomachinery Condition Monitoring," MATEC Web of Conferences, **255**, p. 06008. https://doi.org/10.1051/MATECCONF/201925506008.

[68] Hammond, J., Pepper, N., Montomoli, F., and Michelassi, V., 2022, "Machine Learning Methods in CFD for Turbomachinery: A Review," International Journal of Turbomachinery, Propulsion and Power, **7**(2). https://doi.org/10.3390/IJTPP7020016.

[69] Cruz, G. G., Ottavy, X., and Fontaneto, F., 2024, "Towards Improved Turbomachinery Measurements: A Comprehensive Analysis of Gaussian Process Modeling for a Data-Driven Bayesian Hybrid Measurement Technique," International Journal of Turbomachinery, Propulsion and Power, **9**(3). https://doi.org/10.3390/IJTPP9030028.

[70] Bajarunas, K., Baptista, M., Goebel, K., and Chao, M. A., "Health Index Estimation Through Integration of General Knowledge with Unsupervised Learning," Reliab. Eng. Syst. Saf., **251**. https://doi.org/10.48550/ARXIV.2405.04990.

[71] Kritzinger, W., Karner, M., Traar, G., Henjes, J., and Sihn, W., 2018, "Digital Twin in Manufacturing: A Categorical Literature Review and Classification," IFAC-PapersOnLine, **51**(11), pp. 1016–1022. https://doi.org/10.1016/J.IFACOL.2018.08.474.

[72] Grieves, M., 2014, “Digital Twin: Manufacturing Excellence through Virtual Factory Replication.” [Online]. Available: https://orkg.org/resources/R154580. [Accessed: 17-Nov-2025].

[73] Tao, F., Zhang, M., and Nee, A. Y. C., 2019, *Digital Twin Driven Smart Manufactoring*, Elsevier. https://doi.org/10.1016/B978-0-12-817630-6.00005-9.

[74] Boschert, S., and Rosen, R., 2016, *Digital Twin-the Simulation Aspect*, Springer International Publishing. https://doi.org/10.1007/978-3-319-32156-1_5.

[75] Rasheed, A., San, O., and Kvamsdal, T., 2020, “Digital Twin: Values, Challenges and Enablers from a Modeling Perspective,” IEEE Access, **8**, pp. 21980–22012. https://doi.org/10.1109/ACCESS.2020.2970143.

[76] Abdullahi, I., Longo, S., and Samie, M., 2024, “Towards a Distributed Digital Twin Framework for Predictive Maintenance in Industrial Internet of Things (IIoT),” Sensors (Basel), **24**(8). https://doi.org/10.3390/S24082663.

[77] Liu, Y., Shangguan, D., Chen, L., Liu, X., Yin, G., and Li, G., 2025, “Multi-Domain Digital Twin and Real-Time Performance Optimization for Marine Steam Turbines,” Symmetry (Basel)., **17**(5). https://doi.org/10.3390/SYM17050689.

[78] Liu, H., Xia, M., Williams, D., Sun, J., and Yan, H., 2022, “Digital Twin-Driven Machine Condition Monitoring: A Literature Review,” J. Sensors, **2022**, pp. 1–13. https://doi.org/10.1155/2022/6129995.

[79] Luan, J., Li, S., Cao, Y., and Gu, C., 2025, “Application of a Surrogate Model for Condition Monitoring of a Digital Twin Gas Turbine,” Appl. Math. Model., **137**. https://doi.org/10.1016/J.APM.2024.115683.

[80] Rasheed, A., San, O., and Kvamsdal, T., 2020, “Digital Twin: Values, Challenges and Enablers from a Modeling Perspective,” IEEE Access, **8**, pp. 21980–22012. https://doi.org/10.1109/ACCESS.2020.2970143.

[81] Liu, H., Xia, M., Williams, D., Sun, J., and Yan, H., 2022, “Digital Twin-Driven Machine Condition Monitoring: A Literature Review,” J. Sens., **2022**(1), p. 6129995. https://doi.org/10.1155/2022/6129995.

[82] Ma, Y., Zhu, X., Lu, J., Yang, P., and Sun, J., 2023, “Construction of Data-Driven Performance Digital Twin for a Real-World Gas Turbine Anomaly Detection Considering Uncertainty,” Sensors (Basel), **23**(15). https://doi.org/10.3390/S23156660.

[83] Mrzyk, P., Kubacki, J., Luttmer, J., Pluhnau, R., and Nagarajah, A., 2023, “Digital Twins for Predictive Maintenance: A Case Study for a Flexible IT-Architecture,” Procedia CIRP, **119**, pp. 152–157. https://doi.org/10.1016/J.PROCIR.2023.03.087.

[84] de Castro-Cros, M., Rosso, S., Bahilo, E., Velasco, M., and Angulo, C., 2021, "Condition Assessment of Industrial Gas Turbine Compressor Using a Drift Soft Sensor Based in Autoencoder," Sensors (Basel), **21**(8). https://doi.org/10.3390/S21082708.

[85] Prokhorenko, A., Kravchenko, S., and Solodkii, E., 2021, "DIGITAL TWIN OF GAS RECIPROCATING COMPRESSOR UNIT: CONCEPT, ARCHITECTURE & PILOT IMPLEMENTATION," Internal Combustion Engines, (2), pp. 68–72. https://doi.org/10.20998/0419-8719.2021.2.09.

[86] Rodriguez, D., Clare, P., Srikonda, R., and Suvarna, M., 2022, "Stampede Digital Twin: An Advanced Solution for Process Equipment Condition Monitoring," Day 3 Wed, October 05, 2022, **2022-October**. https://doi.org/10.2118/210106-MS.

[87] Luan, J., Li, S., Cao, Y., and Gu, C., 2025, "Application of a Surrogate Model for Condition Monitoring of a Digital Twin Gas Turbine," Appl. Math. Model., **137**. https://doi.org/10.1016/J.APM.2024.115683.

[88] Pezzini, P., Bonilla, H., Johnson, G., Reinhart, Z., and Bryden, K. M., 2021, "A Digital Twin Environment Designed for the Implementation of Real Time Monitoring Tool," ASME 2021 Power Conference, **2021-July**. https://doi.org/10.1115/POWER2021-65384.

[89] Zhu, Y., Zhang, X., and Luo, M., 2025, "A Fault Early Warning Method Based on Auto-Associative Kernel Regression and Auxiliary Classifier Generative Adversarial Network (AAKR-ACGAN) of Gas Turbine Compressor Blades," Energies (Basel)., **18**(3). https://doi.org/10.3390/EN18030461.

[90] Cruz-Manzo, S., Panov, V., and Bingham, C., 2024, "Compressor Fouling Diagnostics and Prognostics in a Gas Turbine," International Journal of Gas Turbine, Propulsion and Power Systems, **15**(1), pp. 9–14. https://doi.org/10.38036/JGPP.15.1_9.

[91] De Kooning, J. D. M., Stockman, K., De Maeyer, J., Jarquin-Laguna, A., and Vandevelde, L., 2021, "Digital Twins for Wind Energy Conversion Systems: A Literature Review of Potential Modelling Techniques Focused on Model Fidelity and Computational Load," Processes 2021, Vol. 9, Page 2224, **9**(12), p. 2224. https://doi.org/10.3390/PR9122224.

[92] *ISO 23247-1:2021 - Automation Systems and Integration — Digital Twin Framework for Manufacturing — Part 1: Overview and General Principles*. [Online]. Available: https://www.iso.org/standard/75066.html. [Accessed: 17-Nov-2025].

[93] Bishop, C. M., 2006, *Pattern Recognition and Machine Learning* . [Online]. Available: https://link.springer.com/book/9780387310732. [Accessed: 11-Dec-2025].

[94] Murphy, K. P. ., 2012, *Machine Learning : A Probabilistic Perspective*, MIT Press.

[95] Ian Goodfellow, Aaron Courville, and Yoshua Bengio, 2016, *Deep Learning*, The MIT Press. [Online]. Available: http://www.deeplearningbook.org. [Accessed: 11-Dec-2025].

[96] Hu, M., Zhang, J., Matkovic, L., Liu, T., and Yang, X., 2023, "Reinforcement Learning in Medical Image Analysis: Concepts, Applications, Challenges, and Future Directions," J. Appl. Clin. Med. Phys., **24**(2), p. e13898. https://doi.org/10.1002/ACM2.13898;WGROUP:STRING:PUBLICATION.

[97] "Overfitting - MATLAB & Simulink." [Online]. Available: https://it.mathworks.com/discovery/overfitting.html. [Accessed: 03-Dec-2025].

[98] Augustine, M. T., 2024, "A Survey on Universal Approximation Theorems." [Online]. Available: http://arxiv.org/abs/2407.12895. [Accessed: 09-Mar-2026].

[99] Follette, K. B., 2023, "An Introduction to High Contrast Differential Imaging of Exoplanets and Disks." [Online]. Available: http://arxiv.org/abs/2308.01354. [Accessed: 11-Dec-2025].

[100] Cooley, J. W., and Tukey, J. W., 1965, "An Algorithm for the Machine Calculation of Complex Fourier Series," Math. Comput., **19**(90), p. 297. https://doi.org/10.2307/2003354.

[101] Schleer, M., and Abhari, R. S., 2005, "Influence of Geometric Scaling on the Stability and Range of a Turbocharger Centrifugal Compressor," Proceedings of the ASME Turbo Expo, **6 PART B**, pp. 859–869. https://doi.org/10.1115/GT2005-68713.

[102] Takens, F., 1981, "Detecting Strange Attractors in Turbulence," pp. 366–381. https://doi.org/10.1007/BFB0091924.

[103] Carrattieri, L., Cravero, C., Marsano, D., and Valenti, E., 2024, "The Use of Approximate Entropy Analysis for Flow Pattern Identification in Radial Compressors to Detect Instable Operating Conditions," J. Phys. Conf. Ser., **2893**(1), p. 012127. https://doi.org/10.1088/1742-6596/2893/1/012127.

[104] Liu, Y., Du, J., Li, J., Xu, Y., Zhu, J., and Nie, C., 2023, "A Stall Diagnosis Method Based on Entropy Feature Identification in Axial Compressors," International Journal of Mechanical System Dynamics, **3**(1), pp. 73–84. https://doi.org/10.1002/MSD2.12064;CTYPE:STRING:JOURNAL.

[105] Salehin, I., and Kang, D. K., 2023, "A Review on Dropout Regularization Approaches for Deep Neural Networks within the Scholarly Domain," Electronics 2023, Vol. 12, Page 3106, **12**(14), p. 3106. https://doi.org/10.3390/ELECTRONICS12143106.

[106] *Deep Learning: Basics and Convolutional Neural Networks (CNN)*. [Online]. Available: https://www.researchgate.net/publication/371207468_Deep_learning_basics_and_convolutional_neural_networks_CNN. [Accessed: 10-Dec-2025].

[107] Bai, S., Kolter, J. Z., and Koltun, V., 2018, "An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling." [Online]. Available: https://arxiv.org/pdf/1803.01271. [Accessed: 02-Jun-2026].

[108] Lin, Y., Koprinska, I., and Rana, M., 2021, "Temporal Convolutional Attention Neural Networks for Time Series Forecasting," Proceedings of the International Joint Conference on Neural Networks, **2021-July**. https://doi.org/10.1109/IJCNN52387.2021.9534351.

[109] Sheng, J., Xu, J., Li, H., Liu, Z., Zhou, H., You, Y., Song, T., and Zuo, G., 2023, "A Multi-Scale Temporal Convolutional Network with Attention Mechanism for Force Level Classification during Motor Imagery of Unilateral Upper-Limb Movements," Entropy 2023, Vol. 25, Page 464, **25**(3), p. 464. https://doi.org/10.3390/E25030464.

[110] Cravero, C., Marsano, D., Sishtla, V., Halbe, C., and Cousins, W. T., 2024, "Numerical Investigations of Near Surge Operating Conditions in a Two-Stage Radial Compressor With Refrigerant Gas," J. Eng. Gas Turbine. Power, **146**(2). https://doi.org/10.1115/1.4063577/1168931.

[111] "ANSYS FLUENT 12.0 Theory Guide - 2.3.1 The Multiple Reference Frame Model." [Online]. Available: https://www.afs.enea.it/project/neptunius/docs/fluent/html/th/node33.htm#mrf-formulation.

[112] Denton, J. D., 2017, "Multall-an Open Source,Computational Fluid Dynamics Based, Turbomachinery Design System," J. Turbomach., **139**(12). https://doi.org/10.1115/1.4037819/378803.

[113] Carrattieri, L., Cravero, C., Marsano, D., Valenti, E., Sishtla, V., and Halbe, C., 2025, "The Development of Machine Learning Models for Radial Compressor Monitoring with Instability Detection," J. Turbomach., **147**(5). https://doi.org/10.1115/1.4066918/1207362.

[114] Day, I. J., 2016, "Stall, Surge, and 75 Years of Research," J. Turbomach., **138**(1). https://doi.org/10.1115/1.4031473/378558.

[115] Lou, F., and Key, N. L., 2020, "Compressor Stall Warning Using Nonlinear Feature Extraction Algorithms," J. Eng. Gas Turbine. Power, **142**(12). https://doi.org/10.1115/1.4048990/1089752.

[116] Zhao, B., Li, B., Cui, W., Huang, C., and Zhou, T., 2025, "Approximate Entropy Identification of Flow Unsteadiness and Surge Precursor in Both Subsonic Centrifugal and Axial Compressors," Physics of Fluids, **37**(2). https://doi.org/10.1063/5.0251040/3335571.

[117] Zahid, M., Liskiewicz, G., and Korbel, P., 2026, "The Analysis of Compression System Instabilities Using Approximate Entropy for Centrifugal Blower," Mech.

Syst. Signal Process., **242**, p. 113628. https://doi.org/10.1016/J.YMSSP.2025.113628.

[118] Carrattieri, L., Cravero, C., Lanza, M. M., Mortello, M., and Tedeschi, S., 2025, “Enhancing Yaw Control Resilience in Wind Turbines with CFD-Informed Digital Twins,” Wind Energy Science. https://doi.org/10.5194/wes-2025-216.

[119] Carrattieri, L., Cravero, C., Lanza, M., Mortello, M., and Tedeschi, S., 2026, “Digital Twin-Based Monitoring of Wind Turbine Blade Surface Roughness Using CFD-Informed Machine Learning,” ASME Journal (In press).

[120] El Bazi, N., Mabrouki, M., Laayati, O., Ouhabi, N., El Hadraoui, H., Hammouch, F. E., and Chebak, A., 2023, “Generic Multi-Layered Digital-Twin-Framework-Enabled Asset Lifecycle Management for the Sustainable Mining Industry,” Sustainability 2023, Vol. 15, Page 3470, **15**(4), p. 3470. https://doi.org/10.3390/SU15043470.

[121] Jeong, D. Y., Baek, M. S., Lim, T. B., Kim, Y. W., Kim, S. H., Lee, Y. T., Jung, W. S., and Lee, I. B., 2022, “Digital Twin: Technology Evolution Stages and Implementation Layers with Technology Elements,” IEEE Access, **10**, pp. 52609–52620. https://doi.org/10.1109/ACCESS.2022.3174220.

[122] Gao, L., and Hong, J., 2021, “Data-Driven Yaw Misalignment Correction for Utility-Scale Wind Turbines,” Journal of Renewable and Sustainable Energy, **13**(6). https://doi.org/10.1063/5.0056671.

[123] Astolfi, D., Castellani, F., Becchetti, M., Lombardi, A., and Terzi, L., 2020, “Wind Turbine Systematic Yaw Error: Operation Data Analysis Techniques for Detecting It and Assessing Its Performance Impact,” Energies 2020, Vol. 13, Page 2351, **13**(9), p. 2351. https://doi.org/10.3390/EN13092351.

[124] Liew, J., Urbán, A. M., and Andersen, S. J., 2020, “Analytical Model for the Power-Yaw Sensitivity of Wind Turbines Operating in Full Wake,” Wind Energ. Sci, **5**, pp. 427–437. https://doi.org/10.5194/wes-5-427-2020.

[125] “MongoDB Atlas | The Modern, Multi-Cloud Database | MongoDB.” [Online]. Available: https://www.mongodb.com/products/platform. [Accessed: 13-Oct-2025].

[126] INTERNATIONAL ELECTROTECHNICAL COMMISSION, 2022, “IEC 61400-12-1:2022 International Standard Norm.” [Online]. Available: https://www.boutique.afnor.org/fr-fr/norme/iec-614001212022/systemes-de-generation-denergie-eolienne-partie-121-mesures-de-performance-/xs142003/332674. [Accessed: 13-Oct-2025].

[127] Shakya, P., Thomas, M., Seibi, A. C., Shekaramiz, M., and Masoum, M. A. S., 2024, “Fluid-Structure Interaction and Life Prediction of Small-Scale Damaged Horizontal Axis Wind Turbine Blades,” Results in Engineering, **23**, p. 102388. https://doi.org/10.1016/J.RINENG.2024.102388.

[128] Carrattieri, L., Cravero, C., Marsano, D., Valenti, E., Sishtla, V., and Halbe, C., 2025, "The Development of Machine Learning Models for Radial Compressor Monitoring With Instability Detection," J. Turbomach., **147**(5). https://doi.org/10.1115/1.4066918.

[129] Cravero, C., Leutcha, P. J., and Marsano, D., 2025, "Development of an Analytical Model to Evaluate the Effect of the Ported Shroud on Centrifugal Compressors," Green Energy and Intelligent Transportation, **4**(2), p. 100249. https://doi.org/10.1016/J.GEITS.2024.100249.

[130] Siddiqui, M. S., Khalid, M. H., Badar, A. W., Saeed, M., and Asim, T., 2022, "Parametric Analysis Using CFD to Study the Impact of Geometric and Numerical Modeling on the Performance of a Small Scale Horizontal Axis Wind Turbine," Energies 2022, Vol. 15, Page 505, **15**(2), p. 505. https://doi.org/10.3390/EN15020505.

[131] Siddiqui, M. S., Rasheed, A., Tabib, M., and Kvamsdal, T., 2019, "Numerical Investigation of Modeling Frameworks and Geometric Approximations on NREL 5 MW Wind Turbine," Renew. Energy, **132**, pp. 1058–1075. https://doi.org/10.1016/J.RENENE.2018.07.062.

[132] Al-Ttowi, A., Mohammed, A. N., Al-Alimi, S., Zhou, W., Saif, Y., and Ismail, I. F., 2024, "Computational Fluid Dynamics (CFD) Investigation of NREL Phase VI Wind Turbine Performance Using Various Turbulence Models," Processes 2024, Vol. 12, Page 1994, **12**(9), p. 1994. https://doi.org/10.3390/PR12091994.

[133] Jha, S., Jayakumar, G., Abdulgafoor, C. P., Vadlamani, N. R., and Ghosh, S., 2024, "CFD Predictions of the Aerodynamic Characteristics of S1223 Airfoil at Low Reynolds' Number," AIAA Aviation Forum and ASCEND, 2024. https://doi.org/10.2514/6.2024-4431.

[134] Muiruri, P. I., Motsamai, O. S., and Ndeda, R., 2019, "A Comparative Study of RANS-Based Turbulence Models for an Upscale Wind Turbine Blade," SN Appl. Sci., **1**(3), pp. 1–15. https://doi.org/10.1007/S42452-019-0254-5/FIGURES/15.

[135] Lahoz, M., Nabhani, A., Saemian, M., and Bergada, J. M., 2024, "Wind Turbine Enhancement via Active Flow Control Implementation," Applied Sciences (Switzerland), **14**(23). https://doi.org/10.3390/APP142311404.

[136] Michna, J., Rogowski, K., Bangga, G., and Hansen, M. O. L., 2021, "Accuracy of the Gamma Re-Theta Transition Model for Simulating the DU-91-W2-250 Airfoil at High Reynolds Numbers," Energies (Basel)., **14**(24). https://doi.org/10.3390/EN14248224.

[137] Lu, J., Li, C., Li, X., Liu, H., Zhang, G., Liu, N., and Liu, L., 2023, "Analytical Model for the Power Production of a Yaw-Misaligned Wind Turbine," Physics of Fluids, **35**(12). https://doi.org/10.1063/5.0174267/2929442.

[138] Castellani, F., Pandit, R., Natili, F., Belcastro, F., and Astolfi, D., 2023, "Advanced Methods for Wind Turbine Performance Analysis Based on SCADA Data and CFD Simulations," Energies 2023, Vol. 16, Page 1081, **16**(3), p. 1081. https://doi.org/10.3390/EN16031081.

[139] Astolfi, D., Gao, L., Pandit, R., and Hong, J., 2023, "Experimental Analysis of the Effect of Static Yaw Error on Wind Turbine Nacelle Anemometer Measurements," Proceedings - 2023 IEEE International Conference on Environment and Electrical Engineering and 2023 IEEE Industrial and Commercial Power Systems Europe, EEEIC / I and CPS Europe 2023. https://doi.org/10.1109/EEEIC/ICPSEUROPE57605.2023.10194644.

[140] Soriano, M. A., Siegel, H. G., Johnson, N. P., Castorrini, A., Ortolani, A., Minisci, E., and Campobasso, M. S., 2024, "Opensource Machine Learning Metamodels for Assessing Blade Performance Impairment Due to General Leading Edge Degradation," J. Phys. Conf. Ser., **2767**(5), p. 052055. https://doi.org/10.1088/1742-6596/2767/5/052055.

[141] Pryor, S. C., Barthelmie, R. J., Cadence, J., Dellwik, E., Hasager, C. B., Kral, S. T., Reuder, J., Rodgers, M., and Veraart, M., 2022, "Atmospheric Drivers of Wind Turbine Blade Leading Edge Erosion: Review and Recommendations for Future Research," Energies 2022, Vol. 15, Page 8553, **15**(22), p. 8553. https://doi.org/10.3390/EN15228553.

[142] Herring, R., Dyer, K., Martin, F., and Ward, C., 2019, "The Increasing Importance of Leading Edge Erosion and a Review of Existing Protection Solutions," Renewable & Sustainable Energy Reviews, **115**. https://doi.org/10.1016/J.RSER.2019.109382.

[143] Pryor, S. C., Coburn, J. J., and Barthelmie, R. J., 2025, "Spatiotemporal Variability in Wind Turbine Blade Leading Edge Erosion," Energies 2025, Vol. 18, Page 425, **18**(2), p. 425. https://doi.org/10.3390/EN18020425.

[144] Maniaci, D. C., Olsen, A. S., Bak, C., and Forsting, A. M., 2025, *Accuracy of LEE Performance Loss Model Based on Field Observations*, International Energy Agency. [Online]. Available: https://orbit.dtu.dk/en/publications/accuracy-of-lee-performance-loss-model-based-on-field-observation/. [Accessed: 23-Oct-2025].

[145] Duthé, G., Abdallah, I., Barber, S., and Chatzi, E., 2021, "Modeling and Monitoring Erosion of the Leading Edge of Wind Turbine Blades," Energies 2021, Vol. 14, Page 7262, **14**(21), p. 7262. https://doi.org/10.3390/EN14217262.

[146] Mishnaevsky, L., Hasager, C. B., Bak, C., Tilg, A. M., Bech, J. I., Doagou Rad, S., and Fæster, S., 2021, "Leading Edge Erosion of Wind Turbine Blades: Understanding, Prevention and Protection," Renew. Energy, **169**, pp. 953–969. https://doi.org/10.1016/J.RENENE.2021.01.044.

[147] Malik, T. H., and Bak, C., 2024, "Full-Scale Wind Turbine Performance Assessment Using the Turbine Performance Integral (TPI) Method: A Study of Aerodynamic

Degradation and Operational Influences," Wind Energy Science, **9**(10), pp. 2017–2037. https://doi.org/10.5194/WES-9-2017-2024.

[148] Vimalakanthan, K., Van Der Mijle Meijer, H., Bakhmet, I., and Schepers, G., 2023, "Computational Fluid Dynamics (CFD) Modeling of Actual Eroded Wind Turbine Blades," Wind Energy Science, **8**(1), pp. 41–69. https://doi.org/10.5194/WES-8-41-2023.

[149] Castorrini, A., Ortolani, A., and Campobasso, M. S., 2023, "Assessing the Progression of Wind Turbine Energy Yield Losses Due to Blade Erosion by Resolving Damage Geometries from Lab Tests and Field Observations," SSRN Electronic Journal. https://doi.org/10.2139/SSRN.4385156.

[150] Liew, M., Xiao, M., Jones, B. M., and Gaudern, N., 2014, "A Practical Study of the Aerodynamic Impact of Wind Turbine Blade Leading Edge Erosion," J. Phys. Conf. Ser., **524**(1), p. 012031. https://doi.org/10.1088/1742-6596/524/1/012031.

[151] Bons, J. P., 2010, "A Review of Surface Roughness Effects in Gas Turbines," J. Turbomach., **132**(2). https://doi.org/10.1115/1.3066315/464607.

[152] Menter, F. R., 1994, "Two-Equation Eddy-Viscosity Turbulence Models for Engineering Applications," AIAA Journal, **32**(8), pp. 1598–1605. https://doi.org/10.2514/3.12149.

[153] Langtry, R. B., Menter, F. R., Likki, S. R., Suzen, Y. B., Huang, P. G., and Völker, S., 2006, "A Correlation-Based Transition Model Using Local Variables - Part II: Test Cases and Industrial Applications," J. Turbomach., **128**(3), pp. 423–434. https://doi.org/10.1115/1.2184353/476613/A-CORRELATION-BASED-TRANSITION-MODEL-USING-LOCAL.

[154] Forooghi, P., Stroh, A., Magagnato, F., Jakirlić, S., and Frohnapfel, B., 2017, "Toward a Universal Roughness Correlation," Journal of Fluids Engineering, Transactions of the ASME, **139**(12). https://doi.org/10.1115/1.4037280/371945.

[155] Bhatt, C. P., and McClain, S. T., 2009, "Assessment of Uncertainty in Equivalent Sand-Grain Roughness Methods," ASME International Mechanical Engineering Congress and Exposition, Proceedings, **8 PART A**, pp. 719–728. https://doi.org/10.1115/IMECE2007-42105.

[156] "Scipy.Interpolate.CloughTocher2DInterpolator — SciPy v1.9.2 Manual." [Online]. Available: https://docs.scipy.org/doc//scipy-1.9.2/reference/generated/scipy.interpolate.CloughTocher2DInterpolator.html. [Accessed: 29-Oct-2025].

[157] "PyQtGraph - Scientific Graphics and GUI Library for Python." [Online]. Available: https://www.pyqtgraph.org/. [Accessed: 04-Nov-2025].

[158] "Technical Library-Henan Silk Road Abrasives & Tools Co., Ltd." [Online]. Available: https://www.silkroadabrasives.com/Technical-Library. [Accessed: 30-Oct-2025].

[159] Han, W., Kim, J., and Kim, B., 2018, "Effects of Contamination and Erosion at the Leading Edge of Blade Tip Airfoils on the Annual Energy Production of Wind Turbines," Renew. Energy, **115**, pp. 817–823. https://doi.org/10.1016/J.RENENE.2017.09.002.

[160] Malik, T. H., and Bak, C., 2025, "Challenges in Detecting Wind Turbine Power Loss: The Effects of Blade Erosion, Turbulence, and Time Averaging," Wind Energy Science, **10**(1), pp. 227–243. https://doi.org/10.5194/WES-10-227-2025.

[161] Carrattieri, L., Cravero, C., Marsano, D., Valenti, E., Sishtla, V., and Halbe, C., 2025, "The Development of Machine Learning Models for Radial Compressor Monitoring with Instability Detection," J. Turbomach., **147**(5). https://doi.org/10.1115/1.4066918/1207362.

[162] Keegan, M. H., Nash, D. H., and Stack, M. M., 2013, "On Erosion Issues Associated with the Leading Edge of Wind Turbine Blades," J. Phys. D Appl. Phys., **46**(38), p. 383001. https://doi.org/10.1088/0022-3727/46/38/383001.

[163] Sareen, A., Sapre, C. A., and Selig, M. S., 2014, "Effects of Leading Edge Erosion on Wind Turbine Blade Performance," Wind Energy, **17**(10), pp. 1531–1542. https://doi.org/10.1002/WE.1649.

[164] Doagou-Rad, S., and Mishnaevsky, L., 2020, "Rain Erosion of Wind Turbine Blades: Computational Analysis of Parameters Controlling the Surface Degradation," Meccanica, **55**(4), pp. 725–743. https://doi.org/10.1007/S11012-019-01089-X.

[165] "4.18.5. Wall Roughness Effects in Turbulent Wall-Bounded Flows." [Online]. Available: https://ansyshelp.ansys.com/public/account/secured?returnurl=/Views/Secured/corp/v242/en/flu_th/x1-5720008.164.html. [Accessed: 22-Jan-2026].

[166] Schlichting, H., and Gersten, K., 2016, *Boundary-Layer Theory*, Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-662-52919-5/COVER.

[167] Burdett, T. A., Gregg, J. A., and Van Treuren, K. W., 2012, "A Numerical Study on Improving Airfoil Performance at Low Reynolds Numbers for Small-Scale Wind Turbines Using Intentional Roughness," 50th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition. https://doi.org/10.2514/6.2012-136.

[168] Zhang, Y., Igarashi, T., and Hu, H., 2011, "Experimental Investigations on the Performance Degradation of a Low-Reynolds-Number Airfoil with Distributed Leading Edge Roughness." https://doi.org/10.2514/6.2011-1102.

[169] Ehrmann, R., Wilcox, B., White, E., and Maniaci, D., 2017, “Effect of Surface Roughness on Wind Turbine Performance.” https://doi.org/10.2172/1596202.