Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Robust Multi-Task Learning and Entropy-Aware Multi-Objective Optimization for Operational Decision Support in Data-Driven Wind Turbine Digital Twins

Ehsan Saber Germi

ehsan.saber@iau.ir

Islamic Azad University

Sajjad Jahanbakhsh Gudakahriz

Islamic Azad University

Research Article

Keywords: Robust multi-task learning, Multi-objective optimization, Metaheuristic algorithms, Uncertainty-aware decision support, Wind turbine, SCADA data

Posted Date: November 23rd, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11379994/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Robust Multi-Task Learning and Entropy-Aware Multi-Objective Optimization for Operational Decision Support in Data-Driven Wind Turbine Digital Twins

Ehsan Saber Germi[1], Sajjad Jahanbakhsh Gudakahriz[1]

[1] Department of Computer Engineering, Germi.C., Islamic Azad University, Ardabil, Iran

Corresponding Author

Ehsan Saber Germi
Institutional Email: ehsan.saber@iau.ir
Personal Email: ehsansaber2001@gmail.com

Co-author

Sajjad Jahanbakhsh Gudakahriz
Institutional Email: Sa.jahanbakhsh@iau.ac.ir

Abstract

Reliable operational decision support for wind turbines requires more than accurate prediction: predictive outputs must also be translated into operational alternatives, while accounting for the limitations of data-driven models. This study develops an integrated, data-driven decision-support framework that combines perturbation-based robust multi-task learning with entropy-aware multi-objective optimization. Using industrial supervisory control and data acquisition (SCADA) data from an Aventa AV-7 wind turbine, the framework jointly predicts power output and classifies operational status into 11 classes. Robust training incorporates Gaussian input noise and random feature dropout, and predictive performance is evaluated alongside gradient-boosting baselines. Under nominal training, LightGBM achieves an RMSE of 0.0403 and an R^2 of 0.9996, while the robustly trained models retain R^2 values ranging from 0.9957 to 0.9991. The trained deep multi-task learning model is then used as a surrogate for multi-objective optimization with six metaheuristic algorithms. The optimization objectives are predicted power output and an entropy-based index calculated from the predicted operational-status probabilities. The resulting Pareto solutions are summarized through three decision-oriented policies—Safe, Knee-Point, and Max-Power—that represent different trade-offs between predicted performance and model-derived

operational-status uncertainty. To assess the consistency of the optimization surrogate's power predictions, selected optimization-derived solutions are evaluated using a separately trained LightGBM regressor, yielding Pearson correlation coefficients of 0.9561–0.9926 across the six optimization algorithms. The principal contribution is the integration of joint performance and status prediction, perturbation-based robustness, status-probability entropy, Pareto search, and independent model-based assessment within a single monitoring-to-decision pipeline. The framework provides a computational basis for exploring candidate operating policies; the entropy index is not a calibrated failure probability, and the physical effectiveness of the resulting policies requires validation against measured operating responses.

Keywords: Robust multi-task learning; Multi-objective optimization; Metaheuristic algorithms; Uncertainty-aware decision support; Wind turbine; SCADA data

1. Introduction

The increasing availability of industrial sensing, supervisory control and data acquisition (SCADA) systems, and data-driven analytics has created new opportunities for developing intelligent monitoring and decision-support systems for complex industrial assets. In wind-energy systems, SCADA infrastructure provides measurements of operational, mechanical, thermal, environmental, and electrical variables that can support condition monitoring, performance assessment, anomaly detection, and fault diagnosis without requiring additional sensing infrastructure. The literature on SCADA-based wind-turbine monitoring documents the potential of machine-learning methods to extract operational information from routinely collected measurements (Maldonado-Correa et al., 2020; Pandit et al., 2023). More recent reviews also indicate that SCADA-based monitoring remains an active research direction, with data quality, generalization, and the translation of predictive results into actionable operational support continuing to pose important challenges.

A key challenge is that predictive monitoring and operational decision-making are often treated as separate stages. A monitoring model may accurately estimate a continuous performance variable or identify an abnormal operating condition, but these outputs do not by themselves determine how an operating condition should be selected. This distinction matters for complex industrial assets whose behavior depends on interacting variables and cannot be adequately characterized by a single output. For a wind turbine, predicted power output describes quantitative performance, whereas operational status provides complementary information about the state associated with that performance. A decision-support framework can therefore benefit from representing both outputs and explicitly connecting them to a subsequent decision process.

Digital Twin (DT) technology provides a conceptual framework for connecting an industrial asset with a computational representation that supports monitoring, prediction, diagnosis, and decision-making. In wind-energy research, DTs have been investigated for condition monitoring, virtual sensing, fault diagnosis, predictive maintenance, performance assessment, and operational

analysis (Kandemir et al., 2024; Li & Shen, 2022; Olatunji et al., 2021). Recent reviews further describe the expanding role of DT-enabled asset management and decision support in energy systems (Amin et al., 2025). Nevertheless, a data-driven predictive model should not automatically be equated with a complete physical or cyber-physical DT. The extent to which a DT supports operational decisions depends on the role of its virtual representation and on how information produced by that representation is used in subsequent tasks.

This distinction has motivated work extending DT applications beyond monitoring toward prediction and optimization. Recent research has demonstrated explicit coupling between DT-based operating-state prediction and optimization, in which predicted information is supplied to an optimization procedure and the resulting decisions are connected to a virtual representation (Meng et al., 2026). Similar developments have emerged in energy and engineering applications, where data-driven and model-driven DTs are being investigated as computational environments for prediction, optimization, and asset management. These developments indicate that simply combining a predictive model with a DT label is not, by itself, a sufficient research contribution. A more specific challenge is to define and evaluate the information flow through which measurements are transformed into predictive representations and then into decision-oriented outputs.

Multi-task learning (MTL) provides a mechanism for jointly representing complementary aspects of asset behavior. MTL learns related tasks through a shared representation, allowing information relevant to one task to contribute to the learning of another (Caruana, 1997). This approach is useful when multiple outputs describe different but related characteristics of the same asset. In the present study, continuous PowerOutput prediction and discrete StatusAnlage classification are formulated as related tasks within a shared predictive architecture. The regression branch represents quantitative turbine performance, while the classification branch estimates the operational-status distribution. Their joint representation supplies two complementary outputs for the subsequent decision process. Whether this joint representation offers practical benefits, however, must be assessed through predictive evaluation and its role in the downstream optimization framework rather than assumed from the use of MTL alone.

The classification branch also provides information beyond the single status class selected by an argmax decision. Specifically, it produces a probability distribution over 11 operational-status classes. A concentrated distribution indicates that the model assigns most of its probability to one class, whereas a more dispersed distribution indicates that probability is distributed across several classes. This information can be summarized using Shannon entropy and incorporated into an optimization objective. In this study, entropy is used as a model-derived index of operational-status prediction uncertainty. It is not interpreted as a calibrated probability of failure, a direct measure of equipment reliability, or a physical safety metric. Its role is narrower: to represent the dispersion of the model's predicted status probabilities and make that property part of the decision formulation.

The reliability of a monitoring-to-decision pipeline is also affected by the quality and distribution of industrial data. SCADA datasets may contain measurement noise, temporary signal corruption, missing or degraded observations, and imbalanced operating-state distributions. Class imbalance can cause aggregate performance measures to be dominated by frequent classes, while imperfect inputs can degrade learned representations and predictive performance (He & Garcia, 2009). These issues are especially relevant when model outputs subsequently become optimization objectives, because errors or unstable status estimates can influence the operating solutions favored by the optimizer. Research on wind-turbine condition monitoring under poor-quality data has likewise emphasized the importance of robustness when SCADA measurements are imperfect (Ding et al., 2024). Accordingly, robustness should be considered not only as a property of an isolated predictive model but also in relation to the downstream decisions that consume its outputs.

A further challenge is the presence of competing operational objectives. Maximizing power production does not necessarily correspond to selecting an operating condition with the lowest uncertainty in the predicted operational status. The relevant operating variables interact nonlinearly, and the range of candidate solutions is restricted by the bounds and operating relationships represented in the formulation. Multi-objective optimization is suitable for examining trade-offs between competing objectives without reducing them to a single scalar objective using a predetermined weighting scheme. In wind-energy research, optimization methods have been applied to power optimization, fault prediction, control, and related operational tasks. For example, Daaji et al. (2025) formulated wind-turbine fault prediction as a multi-objective genetic-programming problem, while Lei et al. (2025) developed an optimization-based framework for wind-turbine power control using an improved whale optimization algorithm. These studies illustrate the relevance of optimization to wind-energy applications, although their objectives and their positions within the overall monitoring-to-decision process differ from those considered here.

The present study treats optimization as a downstream decision layer of a data-driven monitoring framework. The multi-task model supplies two outputs to this layer: predicted power and the probability distribution over operational-status classes. The latter is summarized by an entropy-based index, while predicted power represents the performance objective. The resulting formulation explores Pareto trade-offs between these quantities under the operating-variable bounds and relationships implemented in the optimization model. This approach is intended to characterize alternative model-based candidate solutions rather than to establish, by itself, that every solution is physically realizable or optimal in an operating turbine.

The assessment of optimization-derived solutions raises a related methodological issue. If the same predictive model both defines the optimization landscape and evaluates the resulting solutions, the assessment remains dependent on that model's assumptions and errors. A separately trained predictive model can provide complementary evidence about the consistency of predicted performance. Accordingly, this study uses a separate LightGBM regression model to assess the power predictions associated with selected optimization-derived solutions. LightGBM does not

generate the operational-status probability distribution and does not contribute to the entropy objective. Instead, it provides a separate regression-based comparison of power performance. Agreement between the two models offers evidence of cross-model predictive consistency, but it does not establish the actual physical response of the turbine to the proposed operating conditions. This distinction is important when interpreting the results as decision support rather than as experimentally validated control recommendations.

Against this background, this study develops a robust, data-driven framework for wind-turbine monitoring and entropy-aware multi-objective operational decision support. The framework is evaluated using SCADA measurements from a commercial Aventa AV-7 wind turbine. A multi-task learning model jointly predicts continuous PowerOutput and the 11-class StatusAnlage operational state. Robustness is investigated through controlled input perturbations, including Gaussian noise and feature dropout, to examine the behavior of the learned representation under deliberately imperfect inputs. The operational-status probability distribution is transformed into an entropy-based index, which is optimized jointly with predicted power. The optimization procedure is evaluated using six metaheuristic algorithms: NSGA-II, MOPSO, MOEA/D, hybrid MOGWO, enhanced MOWOA, and enhanced MOGOA. Representative solutions from the resulting Pareto sets are interpreted through three operating-policy categories, and a separate LightGBM regression model is used to assess the consistency of their predicted power performance.

The research gap addressed here is best characterized as a specific integration and evaluation gap, rather than the absence of any individual methodological component. SCADA-based monitoring, MTL, DTs, robust learning, entropy, and metaheuristic optimization have all been studied in their respective contexts. The focus of this work is to combine these elements in a pipeline that explicitly connects joint performance and operational-status prediction to an entropy-aware multi-objective formulation, examines robustness to controlled input perturbations, compares multiple optimization algorithms, and evaluates selected optimization-derived power predictions using a separate regression model. The study therefore investigates the utility and limitations of this combined formulation in a wind-turbine application; it does not claim that any one of these established methods is newly introduced.

The main contributions of this study are as follows:

1. **An integrated monitoring-to-decision framework:** A data-driven framework is developed to connect SCADA-based prediction with uncertainty-aware multi-objective optimization, making the information flow from joint prediction to candidate operating-policy extraction explicit.
2. **Joint representation of performance and operational status:** A multi-task learning architecture jointly models continuous PowerOutput and 11-class StatusAnlage classification through a shared representation, supplying complementary performance and status-probability outputs to the decision layer.

3. **Perturbation-based robustness assessment:** Gaussian input noise and random feature dropout are incorporated into training to examine how the learned multi-task representation behaves under controlled input perturbations. The assessment is limited to the perturbations investigated and is not taken as evidence of robustness to all real-world sensor faults.
4. **Entropy-aware multi-objective formulation:** Shannon entropy calculated from the predicted operational-status probabilities is incorporated as a model-derived operational uncertainty index and optimized jointly with predicted power, allowing performance–uncertainty trade-offs to be examined without treating entropy as a calibrated failure probability.
5. **Comparative Pareto search and policy interpretation:** Six multi-objective metaheuristic algorithms—NSGA-II, MOPSO, MOEA/D, hybrid MOGWO, enhanced MOWOA, and enhanced MOGOA—are used to explore the formulated objective space. Representative solutions are organized into Safe, Knee-Point, and Max-Power policy categories. These categories describe model-based trade-off preferences and should not be interpreted as physically validated operating recommendations.
6. **Independent model-based assessment of predicted power:** A separately trained LightGBM regressor is used to assess the consistency of power predictions associated with selected optimization-derived solutions. This provides a complementary cross-model check without using LightGBM to generate the status probabilities or entropy objective. It does not replace validation against measured physical responses.

Overall, the contribution lies in the implementation and evaluation of a unified data-driven pipeline that connects joint performance and operational-status prediction, perturbation-based robustness, entropy-based uncertainty characterization, multi-objective search, and separate model-based assessment. The study does not claim to introduce DTs, MTL, robust learning, entropy, or metaheuristic optimization individually. Instead, it investigates how these established components can be organized into a monitoring-to-decision workflow and what can—and cannot—be concluded from its predictive and optimization results **(Figure 1.1)**.

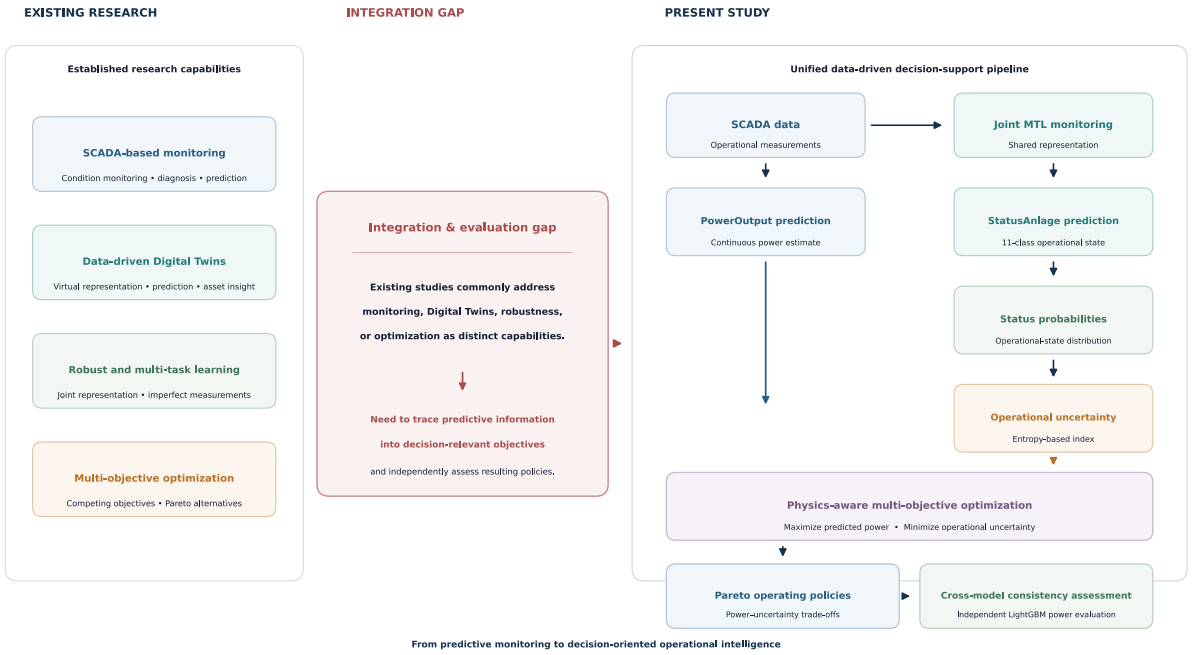


Figure 1.1 Conceptual positioning of the proposed data-driven decision-support framework, linking SCADA-based monitoring with joint performance and operational-state prediction, uncertainty characterization, and physics-aware multi-objective optimization.

2. Related Work

2.1. Industrial Intelligent Monitoring of Wind Turbines

SCADA-based intelligent monitoring is an established approach for extracting operational information from wind turbines. Conventional SCADA systems provide measurements of electrical, mechanical, thermal, and environmental variables at relatively high temporal resolution, enabling data-driven analysis of turbine behavior without requiring additional sensing infrastructure. Machine-learning methods have consequently been investigated for condition monitoring, anomaly detection, power-curve modeling, and fault diagnosis using operational measurements (Maldonado-Correa et al., 2020; Pandit et al., 2023).

A major direction in this literature is the development of predictive models for identifying abnormal operating conditions. Such approaches commonly formulate monitoring as classification, regression, anomaly detection, or combinations of these tasks. Although they can provide useful information about individual aspects of turbine behavior, a model focused on a single monitoring target does not necessarily provide the joint information needed for subsequent operational decisions. Power production and operational status are complementary: power describes quantitative performance, whereas status characterizes the operating state associated with that performance. Jointly representing these outputs is therefore a relevant modeling choice when both are intended to inform a downstream decision process.

MTL provides a general approach for learning related outputs within a shared representation. Its underlying principle is that joint learning can exploit information shared across prediction tasks (Caruana, 1997). In industrial monitoring, this allows a model to estimate continuous variables and discrete operational states within one architecture. However, the value of such a representation for decision support depends on more than its predictive scores. Its robustness to imperfect inputs and the manner in which its outputs are used downstream also require examination.

Class imbalance is another important consideration in operational-status classification. Wind-turbine SCADA datasets may contain many observations of normal or frequently occurring conditions, while other states are comparatively rare. Overall accuracy can consequently provide an incomplete account of classification performance, making class-sensitive metrics important for assessing the representation of less frequent states (He & Garcia, 2009). This issue also matters when class probabilities are subsequently used to construct an uncertainty index, because the interpretation of that index depends on the behavior of the classification model.

Recent work has investigated more specialized learning strategies for wind-turbine fault diagnosis. Qiao et al. (2025), for example, studied model-agnostic meta-learning for few-shot fault diagnosis of wind-turbine generators, addressing learning when labeled fault examples are limited. Daaji et al. (2025), in contrast, formulated wind-turbine fault prediction using multi-objective genetic programming and considered competing fault-related objectives. These studies demonstrate the use of advanced learning and optimization methods in wind-turbine monitoring. Their primary focus, however, is fault diagnosis or prediction, rather than the particular combination of joint power and operational-status prediction with a downstream power–entropy optimization procedure examined in this study.

2.2. Data-Driven Digital Twins for Wind-Energy Systems

Digital Twins extend conventional predictive modeling by providing a computational representation of a physical asset that can be connected to measurements from that asset. Depending on their implementation, DTs can support monitoring, diagnosis, prediction, maintenance, simulation, and decision-making. Within wind energy, DT research has addressed component-level monitoring, fault diagnosis, predictive maintenance, virtual sensing, and performance analysis (Hu et al., 2025; Kandemir et al., 2024; Li & Shen, 2022; Olatunji et al., 2021).

The availability of SCADA data has encouraged data-driven DT approaches in which machine-learning models approximate relationships between operating conditions and asset responses. These approaches are attractive when detailed physical models are difficult to construct or when historical operating data can be used to learn nonlinear relationships. Recent reviews indicate that DT applications in wind-energy systems increasingly encompass monitoring, predictive maintenance, optimization, and energy-management applications (Leon-Medina et al., 2025).

Nevertheless, a data-driven predictive model should not automatically be considered a complete operational DT. A decision-oriented framework requires a defined mechanism through which its virtual representation supplies information for subsequent operational tasks. Amin et al. (2025) reviewed DT technology for smart asset management in the energy industry and discussed the broader development of DT-enabled asset-management capabilities. Meng et al. (2026) developed a DT-driven framework coupling dynamic operating-state prediction with process optimization in a flexible production line. Their study illustrates how prediction and optimization can be linked explicitly rather than treated as independent stages.

Other work has investigated DTs for engineering optimization. Daareyni et al. (2026), for example, combined data-driven and model-driven approaches within a DT framework for engineering-design optimization. Such studies show that DTs can provide computational environments for optimization in addition to monitoring. However, the application domain, decision variables, uncertainty representation, and optimization objectives vary substantially across studies.

In wind-energy applications, previous research has established the relevance of DTs for data-driven monitoring and prediction, while more recent work has extended their use toward optimization and asset management. The present study builds on this direction by investigating a specific computational connection between SCADA-based joint prediction and entropy-aware Pareto search. Its scope is a data-driven decision-support framework; it does not claim real-time bidirectional synchronization or experimental validation of a complete cyber-physical DT.

2.3. Robust and Multi-Task Learning for Industrial Monitoring

Industrial machine-learning systems operate under conditions that may differ from those represented in their training data. Sensor noise, temporary signal corruption, missing observations, measurement errors, and sensor faults can modify input distributions and degrade predictive performance. Robust learning approaches seek to preserve useful predictive behavior when inputs are perturbed or partially corrupted.

In wind-turbine applications, data quality is particularly important because many monitoring methods rely directly on SCADA measurements. Research on fault diagnosis and condition monitoring under imperfect or degraded measurements has emphasized the importance of considering robustness when developing intelligent monitoring systems (Ding et al., 2024). More broadly, the reliability of wind-power prediction can be affected by the quality and representativeness of the available data.

MTL offers a complementary modeling strategy by learning related outputs through a shared representation. Instead of training separate power-regression and status-classification models, a shared architecture can jointly encode information relevant to turbine performance and operational condition. In the present framework, this approach is combined with explicit perturbation mechanisms: Gaussian input noise and random feature dropout are introduced during training. These mechanisms are intended to examine whether the learned representation remains useful

under the specified perturbations. They do not, by themselves, establish robustness to every form of sensor failure, missing data, distribution shift, or real-world degradation.

It is also important to distinguish predictive robustness from decision robustness. A model may retain acceptable regression or classification performance under input perturbations while the optimization procedure selects different candidate solutions. Conversely, a change in a pointwise predictive metric does not necessarily translate directly into a corresponding change in decision quality. These possibilities motivate considering robustness in relation to the complete monitoring-to-decision workflow, while recognizing that a systematic demonstration of decision robustness requires evaluation of how perturbations affect the resulting optimization outputs and policies.

2.4. Multi-Objective Optimization in Wind-Energy Applications

Wind-turbine operation involves multiple objectives and constraints. Increasing power production may conflict with conservative operating preferences, component-loading considerations, temperature limitations, or uncertainty in the estimated operating state. A single-objective formulation may therefore fail to represent the range of trade-offs relevant to operational decision-making.

Metaheuristic optimization has been widely applied to nonlinear engineering problems because population-based methods can explore complex decision spaces without requiring analytical gradients. In wind-energy applications, optimization has been used for power maximization, control, parameter tuning, fault prediction, and energy management. Ala et al. (2023), for example, compared multiple optimization algorithms for wind-energy optimization within a hybrid decision-making framework, illustrating the use of different metaheuristics to investigate optimization performance.

More recent work has combined machine learning with optimization. Lei et al. (2025) developed a wind-turbine power-optimization control framework using an improved whale optimization algorithm and a kernel extreme learning machine. Their approach combines data-driven wind-speed estimation with optimization and adaptive control to improve power-generation performance.

Daaji et al. (2025) used multi-objective genetic programming for wind-turbine fault prediction, demonstrating the application of competing objectives to a fault-prediction task. Their formulation differs from the present study in that it focuses on fault-prediction objectives rather than jointly optimizing predicted power and an entropy index derived from operational-status probabilities.

A state-of-the-art review in *Expert Systems with Applications* also documents the expanding use of machine learning, deep learning, reinforcement learning, and hybrid intelligence in wind-energy forecasting, control, maintenance, diagnostics, and optimization (Dubey et al., 2026).

These studies establish that intelligent optimization is already an active area of wind-energy research. Accordingly, the contribution of the present study is not the application of metaheuristics

to wind turbines in isolation. Rather, it investigates their use as a downstream decision layer in which the objectives are derived from a robust multi-task monitoring model. The comparative use of six algorithms provides an examination of alternative search methods within this particular formulation; it does not, by itself, establish that one algorithm is universally superior or that the resulting solutions are physically validated.

2.5. Uncertainty-Aware Decision Support and Digital-Twin Optimization

Uncertainty is relevant to intelligent decision support because predictive models may produce different levels of ambiguity across operating conditions. In classification, the complete probability distribution over possible states contains more information than the single class selected by an argmax decision. A concentrated distribution indicates that the model assigns most of its probability to one state, whereas a more dispersed distribution indicates that probability is distributed across multiple states.

This property provides a way to connect classification outputs with optimization. Instead of using the predicted status only as a categorical label, the probability distribution can be summarized by a scalar index. Shannon entropy quantifies the uncertainty associated with a discrete probability distribution (Shannon, 1948). In the present framework, the entropy of the 11-class StatusAnlage probability distribution is used as a model-derived operational-status uncertainty index and optimized jointly with predicted power.

The interpretation of this index must remain specific. Entropy reflects the dispersion of the predicted class probabilities; it does not necessarily measure calibrated predictive uncertainty, distinguish epistemic from aleatoric uncertainty, or quantify the physical probability of failure. In particular, a lower entropy value should not be interpreted automatically as lower physical risk or greater equipment reliability. Its role in this study is to provide a consistent, decision-oriented quantity for comparing model predictions across candidate operating conditions.

Recent DT research has increasingly investigated uncertainty and optimization together. Studies have examined probabilistic representations, predictive maintenance, and optimization under uncertain operating conditions, while work in *Expert Systems with Applications* has demonstrated explicit coupling between DT-based dynamic-state prediction and process optimization (Meng et al., 2026). These developments support the broader direction of DT-enabled decision support. The present study investigates a narrower question within wind-turbine operation: how the probability distribution produced by an operational-status classifier can be summarized by entropy and incorporated alongside predicted power in a multi-objective search.

To consolidate the studies discussed in Sections 2.1–2.5, Table 2.1 summarizes representative contributions in SCADA-based monitoring, multi-task and robust learning, wind-energy DTs, and optimization-based decision support. The comparison distinguishes established capabilities in prior work from the particular integration and evaluation strategy investigated here.

Table 2.1 Comparison of Related Studies and Positioning of the Present Work

Study	Primary focus and method	Contribution relevant to this study	Scope relative to the present work
Maldonado-Correa et al. (2020); Pandit et al. (2023)	Reviews of SCADA-based condition and performance monitoring for wind turbines.	Summarize data-driven monitoring tasks, methods, and practical challenges associated with SCADA data.	Provide a broad monitoring perspective; the present study evaluates a specific monitoring-to-optimization pipeline.
Ding et al. (2024)	Multi-task learning for wind-turbine condition monitoring using poor-quality data.	Directly addresses multi-task monitoring under degraded data quality.	The present study connects joint prediction to status-probability entropy and downstream multi-objective optimization.
Zhang et al. (2024); Qiao et al. (2025)	Multi-task normal-behavior modeling for generator fault diagnosis; few-shot diagnosis using model-agnostic meta-learning.	Demonstrate advanced learning strategies for wind-turbine generator diagnosis under different data conditions.	Focus on fault diagnosis rather than joint optimization of predicted power and status-derived entropy.
Kandemir et al. (2024); Leon-Medina et al. (2025)	Reviews of predictive DTs for wind-energy systems and DT technologies for turbine components.	Summarize DT architectures, applications, and challenges in wind-energy systems.	Provide broad reviews rather than an experimental evaluation of the particular monitoring and optimization pipeline proposed here.
Ala et al. (2023); Lei et al. (2025)	Comparative evaluation of wind-energy optimization algorithms; power-optimization control using an improved whale optimization algorithm and a kernel extreme learning machine.	Demonstrate the application of metaheuristic and learning-based methods to wind-energy optimization and control.	Emphasize optimization or control of energy production rather than the combination of status-probability entropy and the Pareto formulation investigated here.
Daaji et al. (2025)	Multi-objective genetic programming for wind-turbine fault prediction.	Applies a multi-objective formulation to a wind-turbine prediction task involving competing fault-related objectives.	Optimizes fault-prediction objectives, whereas the present study jointly considers predicted power and status-prediction entropy.
Amin et al. (2025)	Review of DTs for smart asset management in the energy industry.	Places DT technology in the broader context of intelligent energy-asset management.	Provides an energy-industry perspective rather than the specific turbine-level learning and optimization implementation examined here.
Meng et al. (2026); Daareyni et al. (2026)	DT-driven operating-state prediction and process optimization; data-driven and model-driven DTs for engineering-design optimization.	Demonstrate prediction–optimization integration in industrial and engineering applications.	Address manufacturing-process or engineering-design contexts rather than SCADA-based wind-turbine decision support using joint power/status prediction and entropy-based optimization.
Present study	A data-driven framework integrating deep MTL, perturbation-based	Jointly predicts power output and 11-class operational status; incorporates noise perturbation and feature	Connects SCADA-based monitoring to Pareto-based decision support, derives Safe, Knee-Point, and Max-Power policy categories, and

	robustness, entropy-aware multi-objective optimization, and operational decision support.	dropout during training; uses entropy of predicted status probabilities as an operational-status uncertainty index; and compares six metaheuristic optimizers within the implemented operating constraints.	separately assesses optimization-derived power predictions using a pretrained LightGBM regressor. The framework is surrogate-based; entropy is not a calibrated failure probability, and real-time physical synchronization and physical validation of the proposed policies are not claimed.

2.6. Research Gap and Positioning of the Present Study

The literature reviewed above shows that the main components relevant to intelligent wind-turbine decision support are established research areas. SCADA-based machine learning has been widely investigated for monitoring and diagnosis; DTs have been developed for prediction, maintenance, and asset management; robust learning has been explored for imperfect industrial data; and metaheuristic algorithms have been applied to wind-energy optimization. Previous studies have also demonstrated explicit coupling between DTs, prediction, and optimization. Therefore, neither the use of these methods individually nor their general combination can, by itself, be considered the research gap.

The more specific gap addressed by this study concerns the formulation and evaluation of a connected monitoring-to-decision workflow for data-driven wind-turbine operation. In this workflow, power performance and operational status are jointly predicted; the predicted status distribution is converted into an entropy-based objective; multiple optimization algorithms are used to explore power–entropy trade-offs; and selected optimization-derived power predictions are assessed using a separate regression model. The contribution is consequently an application-specific integration and evaluation of established methods, rather than a claim to introduce a new learning paradigm, uncertainty measure, or metaheuristic algorithm.

The present study addresses this gap through the following elements:

1. **Joint representation of performance and operational status:** Continuous PowerOutput regression and 11-class StatusAnlage classification are learned jointly within a multi-task architecture.
2. **Perturbation-based robustness assessment:** Gaussian input perturbation and random feature dropout are incorporated during training to examine the resilience of the learned representation under the specified imperfect-input conditions.
3. **Decision-oriented uncertainty characterization:** The probability distribution produced by the operational-status classifier is transformed into an entropy-based index and included as an optimization objective. This index represents the dispersion of predicted status probabilities, not a calibrated failure probability.

4. **Multi-objective optimization under implemented operating constraints:** Predicted power and the entropy-based index are optimized jointly using six metaheuristic algorithms, allowing alternative model-based trade-offs to be examined under the bounds and operating relationships encoded in the formulation.
5. **Policy-oriented interpretation:** Selected Pareto solutions are organized into Safe, Knee-Point, and Max-Power categories to communicate different preferences concerning predicted performance and status-probability entropy. These labels represent model-based decision categories, not experimentally verified levels of physical safety or operational feasibility.
6. **Independent regression-based assessment:** A separate LightGBM regressor is used to examine the consistency of predicted power for selected optimization-derived solutions. It does not generate the operational-status probabilities or the entropy objective, and its predictions provide complementary model-based evidence rather than direct physical validation.

Accordingly, the contribution of this study lies in integrating these established capabilities into a unified data-driven decision-support framework and explicitly tracing the information flow from SCADA observations through joint performance and operational-status prediction, uncertainty characterization, multi-objective optimization, policy interpretation, and separate regression-based assessment. The resulting framework provides a structured computational approach for investigating the connection between predictive monitoring and operational optimization, while keeping the limits of model-derived uncertainty and surrogate-based evaluation explicit.

3. Proposed Intelligent System

3.1. System Overview

The proposed framework is designed as a data-driven intelligent decision-support system for condition monitoring and risk-aware performance optimization of wind turbines. As illustrated in **Figure 3.1**, the framework establishes a sequential pipeline from industrial SCADA measurements to data preparation, robust multi-task predictive modeling, operational uncertainty estimation, constrained multi-objective optimization, and interpretable operating policies. Rather than treating monitoring and optimization as separate tasks, the proposed architecture integrates them through a common predictive surrogate that provides both continuous performance estimation and discrete operational-state information.

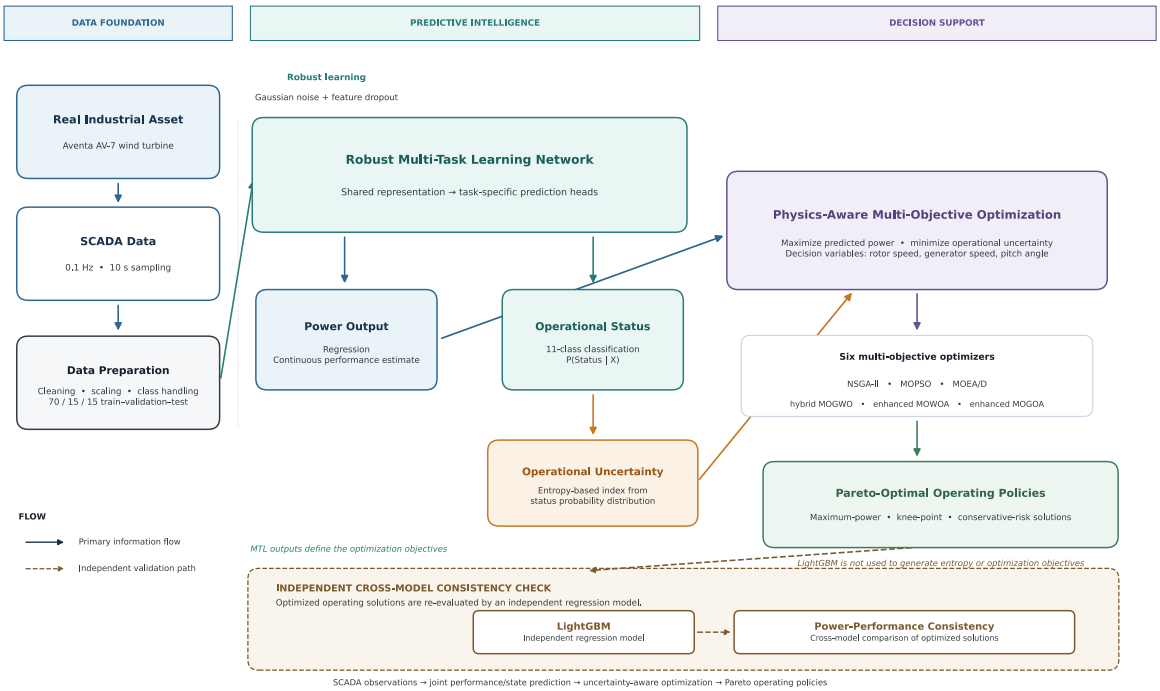


Figure 3.1 Overall framework of the proposed data-driven decision-support system, from SCADA data preparation and robust multi-task prediction to uncertainty-aware optimization and independent cross-model consistency checking.

Following this overall structure, the subsequent subsections describe the individual components of the framework, including data preparation, predictive modeling, robust training, uncertainty characterization, and the constrained multi-objective optimization procedure.

The framework receives eight SCADA-derived variables as input, including RotorSpeed, GeneratorSpeed, GeneratorTemperature, WindSpeed, SpeiseSpannung, MaxWindHeute, offsetWindDirection, and PitchDeg. These measurements are first preprocessed using the same scaling procedure applied during model training and are then provided to a multi-task learning (MTL) model. The MTL architecture contains a shared nonlinear representation followed by two task-specific prediction heads. The first head estimates the continuous PowerOutput of the turbine, whereas the second predicts the probability distribution over 11 operational-status classes represented by the StatusAnlage variable.

The predicted operational-status probabilities are subsequently transformed into an entropy-based operational uncertainty index. This index quantifies the dispersion of the model's probability distribution and is used as a decision-oriented measure of operational uncertainty. It is not interpreted as a calibrated probability of failure; rather, it provides a continuous uncertainty-related quantity that can be incorporated into the optimization process. Consequently, each candidate operating condition can be evaluated simultaneously in terms of predicted power performance and entropy-based operational uncertainty.

To improve the reliability of the predictive surrogate under imperfect measurements, the learning stage incorporates a robust-training strategy based on controlled Gaussian input perturbations and feature dropout. The resulting robust MTL model is retained as the optimization surrogate because it provides both the continuous power prediction and the operational-status probability distribution required to construct the uncertainty objective. The optimization layer then searches the feasible operating space using six multi-objective metaheuristic algorithms: NSGA-II, MOPSO, MOEA/D, Hybrid MOGWO, Enhanced MOWOA, and Enhanced MOGOA. The first three represent established multi-objective optimization frameworks, whereas the latter three are customized implementations developed in this study by extending the corresponding single-objective metaheuristic search mechanisms with Pareto archiving, multi-leader guidance, and additional exploration operators (Coello Coello & Lechuga, 2002; Deb et al., 2002; Mirjalili & Lewis, 2016; Mirjalili et al., 2014; Saremi et al., 2017; Zhang & Li, 2007).

The optimization problem is formulated to maximize predicted power while minimizing entropy-based operational uncertainty, subject to operating bounds and turbine-specific physical constraints. Candidate solutions are processed through a physics-aware repair and penalty mechanism that enforces consistency with operating limits, rotor-speed–wind relationships, pitch-angle constraints, generator–rotor coupling, and generator-temperature considerations. The resulting nondominated solutions form a Pareto front from which three decision-oriented policies—Max-Power, Safe, and Knee-Point—are extracted to represent different trade-offs between performance and operational uncertainty.

Finally, representative operating solutions obtained from the Pareto-based policies are independently evaluated using a separate LightGBM regression model. This external validation is restricted to assessing the consistency of the predicted power associated with the optimization-derived operating points; the operational-status probabilities and entropy-based uncertainty used during optimization remain derived from the MTL model. This separation enables the optimization process and its power predictions to be examined from an independent regression perspective without changing the defined uncertainty mechanism of the proposed framework.

3.2. Data-Driven Digital Twin

In this study, the Digital Twin is implemented as a data-driven predictive representation of the operational behavior of the Aventa AV-7 wind turbine. Rather than relying on a detailed first-principles aerodynamic or electromechanical model, the proposed twin learns the relationship between measured SCADA variables and turbine operating states directly from historical industrial data. This formulation provides a computationally efficient representation that can be repeatedly evaluated within the subsequent optimization process.

Let the input vector at a given operating condition be defined as

$$x = [x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8]^T \qquad (3.1)$$

where the elements correspond to RotorSpeed, GeneratorSpeed, GeneratorTemperature, WindSpeed, SpeiseSpannung, MaxWindHeute, offsetWindDirection, and PitchDeg, respectively.

The proposed Digital Twin maps the input vector to two complementary outputs:

$$\mathcal{T}(x) = \left[\hat{P}, (x)\ p(x)\right] \qquad (3.2)$$

where $\hat{P}(x)$denotes the predicted PowerOutput and $p(x)$represents the predicted probability distribution over the 11 operational-status classes of the StatusAnlage variable.

The probability vector is defined as

$$p(x) = [p_1, (x), p_2, (x) \ldots p_{11}(x)] \qquad (3.3)$$

with

$$\sum_{k=1}^{11} p_k\,(x) = 1,\ p_k(x) \geq 0. \qquad (3.4)$$

This dual-output formulation enables the Digital Twin to represent two complementary aspects of turbine operation. The regression branch provides a continuous estimate of expected power performance, whereas the classification branch characterizes the operating condition through a probability distribution over discrete operational states. This probability distribution is subsequently used to construct the entropy-based operational uncertainty measure described in Section 3.5.

The Digital Twin is trained using historical SCADA observations and subsequently serves as the predictive surrogate within the optimization layer. Therefore, the optimization algorithms evaluate candidate operating conditions through the learned data-driven representation rather than through repeated interaction with the physical turbine or a computationally expensive first-principles simulation.

The proposed Digital Twin is consequently defined as a **data-driven predictive twin**, rather than a complete physics-based digital replica of the turbine. Physical knowledge is incorporated separately at the optimization stage through operating constraints, repair rules, and penalty terms, while the predictive relationships between SCADA variables, power output, and operational status are learned from data. This separation allows the framework to combine data-driven prediction with physics-aware decision making without characterizing the learning model itself as physics-informed.

3.3. Multi-Task Learning

The proposed Digital Twin employs a multi-task learning (MTL) architecture to jointly model continuous turbine performance and discrete operational status from the same SCADA input vector. The main motivation for this design is to learn a shared representation of turbine operating conditions while simultaneously preserving task-specific outputs for power prediction and

operational-state classification. The underlying principle follows the multi-task learning paradigm, in which related tasks are learned jointly through shared representations rather than being modeled completely independently (Caruana, 1997).

Let $x$denote the standardized input vector defined in Eq. (3.1). The shared representation is obtained through three fully connected layers:

$$h_1 = \text{Dropout}(\text{ReLU}(\text{BN}(W_1 x + b_1))) \quad (3.5)$$
$$h_2 = \text{Dropout}(\text{ReLU}(\text{BN}(W_2 h_1 + b_2))) \quad (3.6)$$
$$h_3 = \text{ReLU}(W_3 h_2 + b_3) \quad (3.7)$$

where the dimensions of the shared layers are 128, 64, and 32 neurons, respectively. Batch normalization and a dropout rate of 0.2 are applied to the first two hidden layers.

The resulting shared representation $h_3$is passed to two task-specific prediction heads.

For the regression task, the power-prediction head is defined as

$$h_P = \text{ReLU}(W_P h_3 + b_P) \quad (3.8)$$
$$\hat{P} = W_{P,o} h_P + b_{P,o} \quad (3.9)$$

where the intermediate regression layer contains 16 neurons and the final layer produces a single continuous PowerOutput prediction.

For the operational-status classification task, the classification head is defined as

$$h_S = \text{ReLU}(W_S h_3 + b_S) \quad (3.10)$$
$$z = W_{S,o} h_S + b_{S,o} \quad (3.11)$$

where the intermediate layer contains 16 neurons and the final layer produces 11 logits corresponding to the valid StatusAnlage classes.

The logits are converted into class probabilities using the softmax function:

$$p_k = \frac{\exp(z_k)}{\sum_{j=1}^{11} \exp(z_j)}, \; k = 1, \ldots, 11. \quad (3.12)$$

Thus, the classification branch provides a complete probability distribution rather than only a single predicted class. This property is essential because the probability distribution is subsequently used to calculate the entropy-based operational uncertainty described in Section 3.5.

The two tasks are trained jointly. The total training objective can be expressed generally as a weighted combination of the regression and classification losses:

$$\mathcal{L}_{MTL} = \lambda_P \mathcal{L}_P + \lambda_S \mathcal{L}_S \quad (3.13)$$

where $\mathcal{L}_P$and $\mathcal{L}_S$denote the regression and classification losses, respectively, and $\lambda_P$and $\lambda_S$control their relative contributions.

The model is optimized using the Adam optimizer. The remaining training hyperparameters, including the learning rate, weight decay, early-stopping criterion, and batch-size configuration, are reported in the experimental setup to avoid duplication between the methodological and experimental sections.

The resulting MTL model serves as the central predictive component of the proposed Digital Twin. Its regression branch provides the power estimate required for performance optimization, while its classification branch provides the operational-state probability distribution required for entropy-based uncertainty estimation. This joint representation therefore enables the same predictive surrogate to support both performance-oriented and uncertainty-aware decision making.

3.4. Robust Learning

Industrial SCADA measurements may be affected by sensor noise, transient disturbances, missing information, or temporary degradation of individual measurements. To improve the robustness of the proposed predictive model under such imperfect inputs, a perturbation-based training strategy is incorporated into the MTL architecture. The objective is to expose the model to controlled deviations from the nominal training data and thereby reduce its sensitivity to moderate input perturbations.

Let $x$denote a standardized input vector. During robust training, a perturbed input is generated by adding zero-mean Gaussian noise:

$$\tilde{x} = x + \epsilon,\ \epsilon \sim \mathcal{N}(0, \sigma^2 I) \qquad (3.14)$$

where $\sigma$is the standard deviation of the applied Gaussian perturbation and Idenotes the identity matrix.

In addition to additive noise, a feature-dropout mechanism is employed to simulate the temporary loss or degradation of selected input measurements. A binary mask $m$is sampled independently for the input features according to

$$m_j \sim \text{Bernoulli}(1 - p_d),\ j = 1, \dots, 8 \qquad (3.15)$$

where $p_d$denotes the feature-dropout probability.

The perturbed input used by the model can therefore be represented as

$$x_{robust} = (x + \epsilon) \odot m \qquad (3.16)$$

where $\odot$ denotes element-wise multiplication. In the implemented robust-training procedure, the feature-dropout probability is set to 0.05.

The resulting training samples expose the MTL model to both continuous measurement perturbations and partial feature degradation. The same multi-task prediction structure is then retained, with the model jointly learning PowerOutput regression and 11-class StatusAnlage classification from the perturbed inputs.

The robust model is evaluated using the same held-out test data used for the baseline models, while the perturbation and feature-masking mechanisms are applied during training rather than being interpreted as modifications to the test set. This enables the effect of robust training to be assessed under a consistent evaluation protocol.

The resulting robust MTL model is subsequently used as the predictive surrogate in the optimization stage. This selection is motivated by its dual-output structure: the regression branch provides predicted turbine power, while the classification branch provides the class-probability distribution required to calculate the entropy-based operational uncertainty. Therefore, robustness is incorporated before the optimization stage, allowing the subsequent decision-support process to operate on a predictive model trained to tolerate controlled input perturbations.

3.5. Entropy-Based Operational Uncertainty

The classification branch of the MTL model produces a probability distribution over the 11 operational-status classes of the turbine. Instead of using only the most probable class, the proposed framework retains the complete probability distribution to quantify the uncertainty associated with the predicted operating condition. This enables the operational-state information to be incorporated directly into the subsequent multi-objective optimization process.

For a given operating condition x, let the predicted probability distribution be

$$p(x) = [p_1, (x), \dots p_{11}(x)]. \qquad (3.17)$$

The operational uncertainty is quantified using Shannon entropy (Shannon, 1948):

$$H(x) = -\sum_{k=1}^{11} p_k\,(x)\ln(p_k(x) + \varepsilon) \qquad (3.18)$$

where $H(x)$denotes the entropy-based operational uncertainty index and $\varepsilon > 0$is a small numerical stabilizer introduced to prevent numerical issues when a predicted class probability approaches zero.

The entropy increases when the predicted probability mass is distributed across multiple operational-status classes and decreases when the model assigns most of the probability mass to a single class. Therefore, the index provides a continuous representation of the concentration or dispersion of the model's operational-state prediction (Shannon, 1948).

Importantly, $H(x)$is not interpreted as a calibrated probability of turbine failure, damage, or unsafe operation. It is an entropy-based uncertainty indicator derived from the classification probability distribution. Its role in the proposed framework is to provide a decision-oriented measure that can be jointly considered with predicted power during multi-objective optimization.

The uncertainty objective is therefore formulated together with the power objective. For an operating condition x, the optimization seeks to reduce $H(x)$while increasing the predicted power

output $\hat{P}(x)$. This formulation allows the resulting Pareto front to represent alternative operating conditions with different trade-offs between expected power performance and operational uncertainty.

Because the entropy objective is explicitly derived from the class probabilities generated by the MTL classification branch, the MTL model is retained as the predictive surrogate during optimization. The resulting uncertainty measure is therefore internally consistent with the probabilistic representation learned by the proposed Digital Twin.

3.6. Physics-Aware Multi-Objective Optimization

Following the development of the data-driven Digital Twin and the entropy-based operational uncertainty measure described in Sections 3.2–3.5, a physics-aware multi-objective optimization layer was developed to identify feasible operating conditions that balance turbine power performance against uncertainty in the predicted operational state. Multi-objective optimization is appropriate for such settings because competing objectives generally lead to a set of nondominated solutions rather than a single universally optimal operating point (Deb et al., 2002; Zhang and Li, 2007).

The optimization was formulated as a scenario-based problem in which wind speed was treated as an exogenous operating condition, while the remaining seven variables of the Digital Twin input vector defined the optimization search space.

For a given wind-speed scenario, the seven-dimensional decision vector was defined as

$$u = \left[\omega_r, , \omega_g, , T_g, , V_s\, W_{max}\, \Delta\theta_w\, \beta\right]^T \qquad (3.19)$$

where $\omega_r, \omega_g, T_g, V_s, W_{max}, \Delta\theta_w$, and β denote rotor speed, generator speed, generator temperature, supply voltage, maximum recorded wind-speed indicator, wind-direction offset, and blade-pitch angle, respectively.

Wind speed was not optimized directly; instead, it was fixed for each operating scenario and inserted into the Digital Twin input vector as the exogenous variable V_w.

For each candidate operating point, the corresponding eight-dimensional Digital Twin input was therefore constructed as

$$x = \left[\omega_r, , \omega_g, , T_g, , V_w, V_s\, W_{max}\, \Delta\theta_w\, \beta\right]^T. \qquad (3.20)$$

The predicted power output and the operational-status probability distribution were then obtained from the Digital Twin as described in Sections 3.2–3.5. The uncertainty component used in the optimization was the Shannon-entropy-based operational uncertainty defined in Section 3.5. Thus, the optimization did not introduce a separate uncertainty measure or independently estimate a failure probability.

The optimization considered two competing objectives: minimizing the uncertainty associated with the predicted operational-state distribution and maximizing predicted power. In minimization form, the conceptual optimization problem can be expressed as

$$\min_{u} F(u) = [F_1, (u)F_2(u)] \qquad (3.21)$$

where

$$F_1(u) = H(x) + \lambda_R C_{phys}(u) \qquad (3.22)$$

and

$$F_2(u) = -\hat{P}(x). \qquad (3.23)$$

Here, $H(x)$denotes the entropy-based operational uncertainty, $\hat{P}(x)$denotes the predicted power output, $C_{phys}(u)$represents the physics-consistency penalty, and $\lambda_R$controls its contribution to the uncertainty-related objective.

The formulation in Eqs. (3.21)–(3.23) represents the common conceptual structure of the optimization layer. The exact penalty terms, scaling procedures, and implementation-specific coefficients are reported in the experimental setup because they may differ among the optimization implementations.

Importantly, $H(x)$is not a calibrated probability of failure. It is an uncertainty index derived from the Shannon entropy of the predicted probability distribution over the 11 operational-status classes, as defined in Section 3.5. Lower values therefore indicate a more concentrated and less uncertain predicted operational-state distribution.

Physics-Aware Feasibility Mechanism

A repair mechanism was applied to candidate solutions before evaluation by the Digital Twin. The purpose of this mechanism was to prevent the search algorithms from exploring physically implausible combinations of turbine operating variables and to incorporate the principal operating relationships of the turbine.

The operating range was divided into below-rated and above-rated regions according to the turbine operating regime. In the below-rated region, rotor speed was constrained according to the prescribed wind-speed-dependent operating relationship:

$$\omega_r \in \left[\omega_{r,\min}, (V_w)\omega_{r,\max}(V_w)\right]. \qquad (3.24)$$

The corresponding pitch-angle range was also restricted according to the applicable below-rated operating regime:

$$\beta \in [\beta_{\min}, (V_w)\beta_{\max}(V_w)]. \qquad (3.25)$$

For above-rated operation, rotor speed was restricted to the applicable approximately constant-speed operating range, while pitch angle was adjusted according to wind-speed-dependent operating limits. Because the exact above-rated repair ranges were not identical in all six implementations, these implementation-specific limits were retained in the corresponding optimization implementations rather than being represented as a single universal analytical constraint.

The generator-to-rotor speed relationship was enforced through the feasibility condition

$$\omega_g = N_g \omega_r \qquad (3.26)$$

where $N_g$denotes the gearbox ratio. For the Aventa AV-7 system considered in this study, the reference gearbox ratio was fixed at

$$N_g = 11.5. \qquad (3.27)$$

In addition to this repair constraint, deviation of the generator-to-rotor speed ratio from the target ratio was incorporated into the physics-consistency penalty:

$$C_{ratio} = \left| \frac{\omega_g}{\omega_r + \varepsilon} - N_g \right|. \qquad (3.28)$$

where $\varepsilon > 0$is a small numerical stabilizer used to avoid division by zero.

Additional penalty components were used in the corresponding implementations to discourage deviations from intended operating characteristics, including generator-temperature deviation from the reference operating value and rotor-speed deviation from prescribed operating targets. A general form of the rotor-speed deviation penalty can be written as

$$C_\omega = \left| \omega_r - \omega_r^{ref}(V_w) \right|. \qquad (3.29)$$

Similarly, the temperature-related penalty can be represented as

$$C_T = \left| T_g - T_g^{ref} \right|. \qquad (3.30)$$

The complete physics-consistency penalty can therefore be expressed generally as a combination of the applicable penalty components:

$$C_{phys} = \lambda_{ratio} C_{ratio} + \lambda_\omega C_\omega + \lambda_T C_T + \cdots \qquad (3.31)$$

where the coefficients and applicable terms depend on the corresponding implementation.

These penalty components were incorporated into the uncertainty-related objective according to the implementation-specific formulations. Therefore, the repair mechanism primarily enforced feasibility, whereas the penalty terms discouraged solutions that remained feasible but deviated from the intended operating characteristics.

The complete repair mechanism was applied before Digital Twin evaluation. Consequently, the seven decision variables were not treated as an unconstrained numerical search space; instead, the optimization algorithms explored a physics-aware operational state space conditioned on the specified wind-speed scenario.

Multi-Objective Search Algorithms

Six population-based multi-objective optimization algorithms were employed: NSGA-II, MOPSO, MOEA/D, Hybrid MOGWO, Enhanced MOWOA, and Enhanced MOGOA. The algorithms were selected to provide different evolutionary and swarm-based search mechanisms while operating on the same predictive Digital Twin and the same conceptual objective framework.

NSGA-II. The Non-dominated Sorting Genetic Algorithm II (NSGA-II) is an elitist evolutionary multi-objective algorithm based on fast nondominated sorting and crowding-distance preservation (Deb et al., 2002). In the present framework, NSGA-II serves as a dominance-based evolutionary reference method for approximating the Pareto front.

MOPSO. Multi-Objective Particle Swarm Optimization extends the particle swarm optimization paradigm to multi-objective problems by using Pareto dominance and an external repository of nondominated solutions to guide particle movement (Coello Coello and Lechuga, 2002). In the present implementation, particles update their positions using inertia, cognitive, and social components. Leader selection is performed probabilistically using crowding-distance information, while velocity clamping and stochastic mutation are incorporated to regulate exploration and maintain solution diversity.

MOEA/D. The Multi-Objective Evolutionary Algorithm based on Decomposition (MOEA/D) decomposes a multi-objective optimization problem into a set of scalar subproblems and optimizes these subproblems simultaneously using neighborhood information (Zhang and Li, 2007). In the present implementation, real-coded evolutionary operators based on simulated binary crossover and polynomial mutation are used to generate candidate solutions. The underlying simulated binary crossover operator follows the formulation introduced by Deb and Agrawal (1995).

Hybrid MOGWO. The Hybrid Multi-Objective Grey Wolf Optimizer was developed in this study by extending the Grey Wolf Optimizer search mechanism into a multi-objective archive-based framework. The underlying Grey Wolf Optimizer models a leadership hierarchy through alpha, beta, and delta solutions and uses encircling and hunting mechanisms to update candidate positions (Mirjalili et al., 2014). In the present implementation, this core mechanism is combined with Differential Evolution mutation and Lévy-flight perturbation to enhance search diversity. The Differential Evolution component follows the mutation principle introduced by Storn and Price (1997), while the Lévy perturbation is based on Lévy-stable random-step generation (Mantegna, 1994). Because this hybrid configuration was specifically implemented in the present study, these references represent the methodological foundations of the incorporated operators rather than the exact proposed implementation.

Enhanced MOWOA. The Enhanced Multi-Objective Whale Optimization Algorithm was developed in this study from the Whale Optimization Algorithm framework. The original WOA models exploration and exploitation through whale-inspired encircling, search, and bubble-net hunting mechanisms (Mirjalili and Lewis, 2016). The present implementation extends this search mechanism through an external Pareto archive, crowding-distance-based selection of multiple leaders, stochastic Lévy perturbation, and population perturbation. The resulting algorithm is therefore referred to as **Enhanced MOWOA**, with the original WOA serving as its methodological basis rather than as an exact description of the implemented algorithm.

Enhanced MOGOA. The Enhanced Multi-Objective Grasshopper Optimization Algorithm was developed by extending the Grasshopper Optimization Algorithm (GOA) into a multi-objective archive-based search framework. GOA was originally introduced as a swarm-based optimization method inspired by grasshopper swarming behavior (Saremi et al., 2017). In the present implementation, the underlying social-interaction mechanism is combined with adaptive exploration–exploitation factors, multiple-leader guidance derived from the Pareto archive, Lévy perturbation, stochastic mutation, and explicit physics-aware repair. Accordingly, the cited GOA study represents the underlying optimization mechanism, whereas the multi-objective and enhancement mechanisms constitute the implementation developed in this study.

The six implementations share the same conceptual decision-support objective but may differ in implementation-specific numerical details such as penalty coefficients, objective scaling, repair ranges, and algorithm-specific exploration mechanisms. Such implementation details are reported separately in the experimental setup. This separation allows the present section to define the methodological structure of the proposed optimization framework without duplicating the experimental configuration.

For each operating scenario, each optimization algorithm generates a set of nondominated operating solutions that approximates the corresponding Pareto front. These solutions represent alternative feasible operating states with different trade-offs between predicted power performance and the uncertainty-related objective. A dedicated post-processing procedure is subsequently applied to these Pareto sets to extract representative operating policies for interpretation and comparative analysis. The policy-extraction procedure is described in Section 3.7.

3.7. Pareto-Based Operating Policies

The Pareto fronts obtained from the six optimization algorithms provide multiple nondominated operating solutions for each wind-speed scenario. Since these solutions represent different trade-offs between predicted power performance and the uncertainty-related objective, a post-processing procedure was applied to extract a small number of interpretable operating policies. Three representative policies were defined: Safe, Knee-Point, and Max-Power.

The policy extraction procedure was applied independently to the Pareto set generated by each optimization algorithm at each valid wind-speed scenario. Thus, the extracted policies preserve

the algorithm-specific Pareto solutions and do not impose a common optimization result across the six algorithms.

Safe Policy

The Safe policy was defined as the Pareto solution with the minimum reported Risk value within a given wind-speed scenario. The selection was therefore formulated as

$$x_{Safe} = R_{rep}(x) \qquad (3.32)$$

where $\mathcal{P}(V_w)$denotes the Pareto set obtained under wind-speed scenario V_w, and R_{rep}denotes the Risk value reported in the corresponding Pareto set.

The reported Risk should be distinguished from the raw Shannon entropy defined in Section 3.5. Its underlying uncertainty component is derived from the entropy of the MTL classification probability distribution, whereas the reported optimization objective may additionally contain the physics-consistency penalty according to the implementation of the corresponding optimization algorithm. Therefore, the Safe policy represents the Pareto solution with the lowest reported uncertainty-related objective value rather than a solution with a directly measured or calibrated probability of failure.

Max-Power Policy

The Max-Power policy was selected as the Pareto solution that provides the highest predicted power output:

$$x_{MaxP} = \hat{P}(x) \qquad (3.33)$$

where $\hat{P}(x)$denotes the power predicted by the Digital Twin for the corresponding operating condition.

This policy represents the performance-oriented extreme of the Pareto set. It does not imply that the selected operating point has the minimum uncertainty or the lowest reported Risk; rather, it identifies the nondominated solution associated with the highest predicted power under the specified wind-speed scenario.

Knee-Point Policy

To identify an operating point representing a compromise between predicted power and the reported Risk objective, the Pareto solutions were normalized independently within each wind-speed scenario.

For a Pareto solution i, the normalized Risk value was calculated as

$$R_i^N = \frac{R_i - R_{min}}{R_{max} - R_{min} + \varepsilon} \qquad (3.34)$$

and the normalized predicted power was calculated as

$$P_i^N = \frac{P_i - P_{min}}{P_{max} - P_{min} + \varepsilon} \qquad (3.35)$$

where $R_i$and $P_i$denote the reported Risk and predicted power of solution i, respectively. The quantities R_{min}, R_{max}, P_{min}, and $P_{max}$are calculated from the corresponding Pareto set for the same wind-speed scenario, and $\varepsilon > 0$is a small numerical stabilizer.

After normalization, the ideal point was defined as

$$z^* = (0, 1) \qquad (3.36)$$

where zero represents the minimum normalized Risk and one represents the maximum normalized power.

The Euclidean distance of each Pareto solution from this ideal point was then calculated as

$$D_i = \sqrt{(R_i^N)^2 + (1 - P_i^N)^2} \qquad (3.37)$$

The Knee-Point policy was selected as the Pareto solution with the minimum distance to the ideal point:

$$x_{Knee} = D_i. \qquad (3.38)$$

This procedure identifies a representative compromise between the two objectives within the normalized Pareto set. Importantly, the Knee-Point policy is not selected using an independently defined weighting coefficient between power and Risk; instead, it is determined geometrically from the normalized Pareto solutions.

Policy Extraction Across Wind-Speed Scenarios

The above procedure was repeated independently for every wind-speed scenario and for each of the six optimization algorithms. Scenarios containing fewer than three Pareto solutions were excluded from policy extraction because they did not provide a sufficient set of nondominated alternatives for meaningful three-policy representation.

For each valid scenario, the resulting policy record retained the complete operating state, including rotor speed, generator speed, generator temperature, wind speed, supply voltage, maximum recorded wind speed, wind-direction offset, and blade-pitch angle, together with the corresponding predicted power and reported Risk values. The extracted policies were subsequently organized according to wind speed and policy type to facilitate cross-algorithm comparison and independent validation.

The resulting three-policy representation provides an interpretable summary of the Pareto front while preserving the underlying multi-objective nature of the optimization. The Safe policy

emphasizes the minimum reported risk, the Max-Power policy emphasizes predicted power performance, and the Knee-Point policy provides a balanced geometric compromise between the two objectives within each wind-speed scenario. These policies were subsequently used for comparative analysis and cross-model consistency assessment of the optimization-derived power predictions.

Use of AI-Assisted Tools

ChatGPT (OpenAI) was occasionally used to assist with code review and debugging by identifying potential implementation issues and suggesting corrections. The authors evaluated and applied relevant suggestions and retained responsibility for the final code and computational results.

4. Experimental Setup

4.1. Aventa AV-7 SCADA Dataset

The experimental evaluation was conducted using the publicly available Aventa AV-7 (6 kW) IET-OST research wind turbine SCADA dataset (Barber et al., 2025). The Aventa AV-7 is a commercial wind turbine with a rated power of 6 kW. The dataset provides operational measurements related to turbine dynamics, environmental conditions, electrical variables, and operational status.

The original SCADA measurements were recorded at an approximately 1-s sampling interval. To reduce the data volume while retaining the relevant operational information, every tenth observation was retained, resulting in an effective sampling interval of approximately 10 s (0.1 Hz). After preprocessing, the resulting dataset contained 3,971,598 observations and 10 modeling variables, comprising eight predictive input variables and two target variables.

The eight input variables used by the predictive models were RotorSpeed, GeneratorSpeed, GeneratorTemperature, WindSpeed, SpeiseSpannung, MaxWindHeute, offsetWindDirection, and PitchDeg. These variables represent complementary mechanical, thermal, environmental, electrical, and control-related characteristics of the turbine. The first target, PowerOutput, represents generated electrical power and was treated as a continuous regression target. The second target, StatusAnlage, represents the operational status of the turbine and was formulated as a multiclass classification target.

Table 4.1 summarizes the variables used in the predictive experiments.

Table 4.1 Variables used for predictive modeling

Variable	Role	Type	Description
RotorSpeed	Input	Continuous	Rotor rotational speed
GeneratorSpeed	Input	Continuous	Generator rotational speed
GeneratorTemperature	Input	Continuous	Generator temperature

WindSpeed	Input	Continuous	Measured wind speed
SpeiseSpannung	Input	Continuous	Supply voltage
MaxWindHeute	Input	Continuous	Maximum recorded wind speed
offsetWindDirection	Input	Continuous	Wind-direction offset
PitchDeg	Input	Continuous	Blade-pitch angle
PowerOutput	Target	Continuous	Generated electrical power
StatusAnlage	Target	Categorical	Turbine operational-status class

The original StatusAnlage variable contained multiple categorical operating states. For the classification experiments, the valid operational classes were restricted to the following 11 status labels:

$$0,2,3,5,6,7,8,9,10,12,13.$$

These original labels were mapped to consecutive class indices from 0 to 10 for compatibility with the classification models. The mapping was retained throughout the experiments so that model predictions could subsequently be interpreted in terms of the original turbine-status codes.

Table 4.2 Mapping between original and encoded operational-status labels

Original StatusAnlage	Encoded Class
0	0
2	1
3	2
5	3
6	4
7	5
8	6
9	7
10	8
12	9
13	10

Before model development, the selected input and target variables were extracted from the SCADA records, and observations containing missing values among the selected modeling variables were

removed. The timestamp associated with each observation was retained during data loading and preprocessing to preserve the chronological reference of the measurements; however, the timestamp was not used as a predictive feature.

The resulting dataset was used consistently throughout the predictive modeling and subsequent optimization experiments. The eight input variables formed the feature vector supplied to the predictive models, whereas PowerOutput and StatusAnlage constituted the regression and classification targets, respectively. This dual-target structure provided the basis for the subsequent multi-task predictive architecture.

The principal characteristics of the experimental dataset are summarized in Table 4.3.

Table 4.3 Summary of the Aventa AV-7 SCADA dataset

Characteristic	Value
Turbine	Aventa AV-7
Turbine type	Commercial/industrial wind turbine
Rated power	6 kW
Sampling interval	Approximately 10 s
Effective sampling frequency	0.1 Hz
Number of observations	3,971,598
Number of input variables	8
Regression target	PowerOutput
Classification target	StatusAnlage
Number of valid status classes	11
Missing-value handling	Removal of incomplete observations
Timestamp	Retained as temporal reference, not used as model input

The resulting dataset provides a realistic basis for evaluating predictive performance and subsequent optimization behavior under diverse turbine operating conditions. In particular, the simultaneous availability of power-output and operational-status information allows the predictive layer to provide both performance estimation and probabilistic operational-state information for the downstream optimization process. The use of data-driven representations for monitoring and decision support is consistent with recent industrial Digital Twin research, in which data-driven models can complement conventional physical representations and support monitoring, prediction, and optimization tasks (Liu et al., 2024; Zhang et al., 2024).

4.2. Data Preprocessing

The SCADA dataset described in Section 4.1 was processed through a standardized preprocessing pipeline before model training. The purpose of this stage was to establish a consistent feature representation, remove incomplete observations, retain the predefined operational-status classes, construct independent training, validation, and test subsets, and prevent information leakage between the different subsets.

First, the eight input variables and two target variables defined in Section 4.1 were extracted. Observations containing missing values among these selected variables were removed. The resulting complete observations constituted the modeling dataset.

The PowerOutput variable was retained as a continuous regression target, whereas StatusAnlage was treated as a categorical target. Only the predefined valid operational-status classes were retained for classification. The 11 original classes were mapped to consecutive indices from 0 to 10. This encoding changed only the numerical representation required by the machine-learning implementations and did not alter the operational meaning of the original status labels.

The complete modeling dataset was divided into training, validation, and test subsets using a 70/15/15 split. Specifically, 70% of the observations were assigned to the training set, while the remaining 30% formed a temporary subset that was subsequently divided equally into validation and test sets. A fixed random state of 42 was used throughout the splitting procedure to ensure reproducibility. Stratification was applied to the classification pipeline to preserve the distribution of the valid operational-status classes across the subsets.

Feature standardization was performed using the StandardScaler transformation. To prevent information leakage, the standardization parameters were estimated exclusively from the training subset. For an input feature x, the standardized value was calculated as:

$$z = \frac{x - \mu_{\text{train}}}{\sigma_{\text{train}}} \quad (4.1)$$

where $\mu_{\text{train}}$and σ_{train}denote the mean and standard deviation estimated from the training data only. The same parameters were subsequently applied to the validation and test subsets.

The continuous PowerOutput target was also standardized in the regression training pipeline. The target scaler was fitted exclusively on the training portion of PowerOutput and then applied to the validation and test targets. For final interpretation, predicted values were transformed back to the original physical power scale before calculating the reported regression metrics.

The resulting dimensions of the main preprocessing pipeline were 2,780,117 observations for training, 595,740 observations for validation, and 595,740 observations for testing. The one-record difference relative to an exact arithmetic 70/15/15 partition results from the discrete allocation of observations during the sequential splitting procedure and does not represent missing data or a preprocessing inconsistency.

To improve robustness against sensor disturbances, the training data were additionally subjected to controlled perturbations during the robust-learning stage. Gaussian noise was introduced into standardized training features, while validation and test observations remained unperturbed. This separation ensured that robustness was learned from the training data without artificially modifying the evaluation distributions.

The overall preprocessing sequence was therefore:

$$RawSCADAdata \rightarrow selectionofmodelingvariables \rightarrow removalofincompleteobservations \rightarrow validationofoperational-statusclasses \rightarrow train/validation/testsplitting \rightarrow training-onlyfeaturestandardization \rightarrow training-onlytargetstandardization \rightarrow model-specifictrainingpreparation.$$

All transformations that estimate parameters from the data were fitted exclusively on the training subset. Model-specific procedures, including class balancing, perturbation, model architecture, hyperparameters, and robust-learning operations, are described in the following sections.

4.3. Predictive Models

Three predictive approaches were employed to construct the data-driven predictive core of the proposed framework: Extreme Gradient Boosting (XGBoost), Light Gradient Boosting Machine (LightGBM), and a Multi-Task Learning Neural Network (MTL-Net). XGBoost and LightGBM were used as tree-based reference models, whereas MTL-Net was designed to jointly model turbine power and operational status through shared representations and task-specific output heads.

XGBoost is a scalable gradient-boosting framework based on ensembles of decision trees and incorporates regularization and efficient tree construction mechanisms (Chen and Guestrin, 2016). LightGBM is another gradient-boosting decision-tree framework designed particularly for efficient learning on large datasets and employs mechanisms such as Gradient-based One-Side Sampling and Exclusive Feature Bundling (Ke et al., 2017).

All predictive approaches received the same eight SCADA input variables:

$$X = \begin{bmatrix} RotorSpeed, GeneratorSpeed, GeneratorTemperature, WindSpeed, \\ \text{SpeiseSpannung, MaxWindHeute, offsetWindDirection, PitchDeg} \end{bmatrix} \quad (4.2)$$

The regression target was PowerOutput, while the classification target consisted of the 11 encoded operational-status classes.

$$C = \{0,2,3,5,6,7,8,9,10,12,13\} \quad (4.3)$$

The original status-to-index mapping was retained so that classification outputs could be interpreted using the original SCADA status codes.

4.3.1. XGBoost

XGBoost was employed as a gradient-boosted decision-tree baseline for both regression and multiclass classification. Separate XGBoost models were trained for PowerOutput and StatusAnlage.

In Phase 1, both models were configured with 300 boosting estimators, a learning rate of 0.10, and a maximum tree depth of 8. The classification model employed the multi:softprob objective to generate class-probability distributions required for the subsequent operational-risk calculation.

Table 4.4 Phase-1 XGBoost configuration.

Parameter	Regression	Classification
Number of estimators	300	300
Learning rate	0.10	0.10
Maximum depth	8	8
Subsample	0.80	0.80
Column subsampling	0.80	0.80
Tree method	Histogram	Histogram
Random state	42	42

In Phase 2, XGBoost was retrained using the robust training procedure described in Section 4.4. The regression model employed 700 estimators, a learning rate of 0.03, a maximum depth of 6, and L1 and L2 regularization coefficients of 1 and 2, respectively. The classification model employed 600 estimators, a learning rate of 0.05, a maximum depth of 5, and the same regularization coefficients.

Table 4.5 Phase-2 XGBoost configuration.

Parameter	Regression	Classification
Number of estimators	700	600
Learning rate	0.03	0.05
Maximum depth	6	5
Subsample	0.80	0.80
Column subsampling	0.80	0.80
L1 regularization	1	1
L2 regularization	2	2

Tree method	Histogram	Histogram
Early-stopping patience	25	30
Random state	42	42

For regression, the standardized target was used during training and predictions were transformed back to the original physical scale before final evaluation. For classification, the encoded class labels were used directly.

4.3.2. LightGBM

LightGBM was employed as a second tree-based reference model. Its inclusion provided an independently implemented gradient-boosting architecture with an emphasis on computational efficiency and scalability for large datasets (Ke et al., 2017).

Separate LightGBM regressors and classifiers were trained for the two predictive tasks. The models received the same eight input variables and the same training, validation, and test subsets used for the other predictive approaches.

The LightGBM power model used 1200 trees, a learning rate of 0.03, 64 leaves, L1 regularization of 1, and L2 regularization of 2. The corresponding classification model followed the same general robust-learning pipeline, with model-specific optimization performed independently for the classification task.

The LightGBM model was also retained after training as an independent power evaluator in the validation of optimized operating states described in Section 4.6.4. Importantly, this model was not retrained on the optimized solutions. Consequently, the independent validation stage evaluated whether the operating states generated by the optimization process were consistent with a separately trained predictive representation.

4.3.3. Multi-Task Learning Neural Network

The MTL-Net was designed to jointly learn turbine power prediction and operational-status classification from a shared feature representation. Unlike the tree-based approaches, which employed separate models for the two tasks, the MTL-Net used a shared representation followed by task-specific regression and classification heads. This architecture enables the model to learn common operational features while simultaneously capturing information specific to power production and turbine operating status.

The overall architecture of the MTL-Net is illustrated in Figure 4.1. The network receives the eight SCADA-derived input variables and processes them through a shared feature-extraction pathway before branching into two task-specific heads. The shared pathway provides a common latent representation for both prediction tasks, while the separate heads allow each task to learn its corresponding output representation.

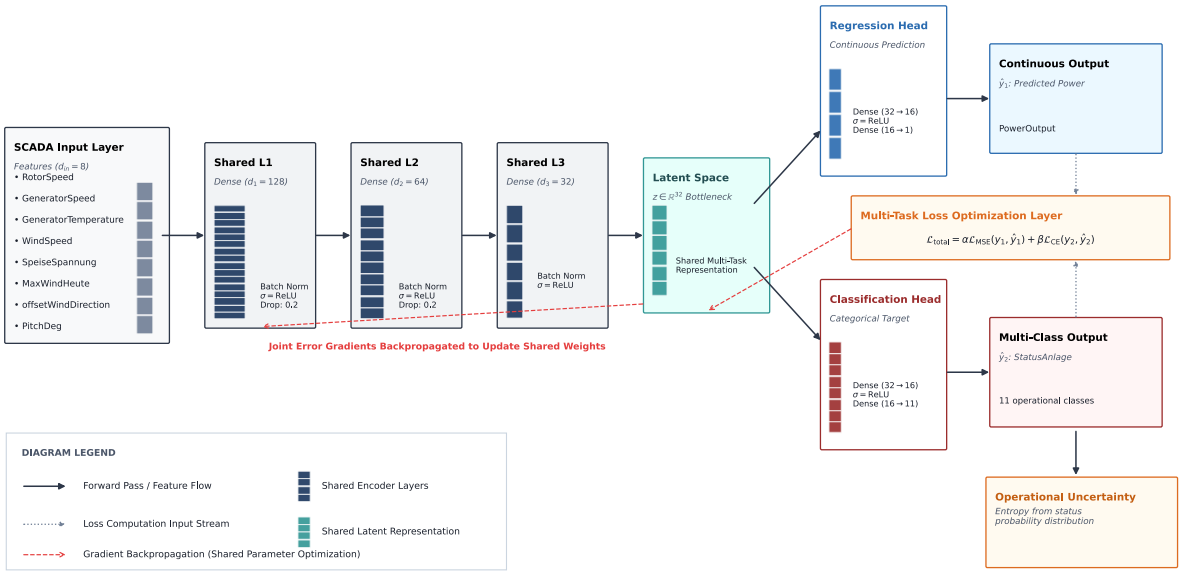


Figure 4.1 Architecture of the multi-task learning neural network, showing the shared representation and task-specific power regression and operational-status classification heads.

The shared network consisted of sequential fully connected layers with the implemented dimensional structure of $8 \rightarrow 128 \rightarrow 64 \rightarrow 32$. Batch normalization, ReLU activation, and dropout with a probability of 0.2 were incorporated into the shared representation to improve training stability and reduce overfitting. The final 32-dimensional shared representation was subsequently provided to the two task-specific branches.

The regression head consisted of fully connected layers that mapped the shared representation to a single continuous output corresponding to the predicted PowerOutput. This branch therefore provided the quantitative estimate of turbine power used for predictive evaluation and, subsequently, as the performance objective in the optimization stage.

The classification head mapped the shared representation to 11 output logits corresponding to the operational-status classes. A probability distribution over the 11 classes was subsequently obtained from these logits. Unlike the regression output, which provides a direct estimate of power performance, this probabilistic output provides information about the model's estimated operational state and forms the basis for the uncertainty characterization described in the subsequent optimization methodology.

The network was trained using the Adam optimizer with a learning rate of 0.001 and weight decay of 1e-5. Batch sizes of 2048/4096 were used according to the implemented training configuration, and early stopping with a patience of 15 epochs was employed.

The joint formulation enables the shared representation to capture operational information relevant to both power production and turbine status. Importantly, the MTL architecture itself does not

directly perform the subsequent uncertainty calculation or multi-objective optimization. Instead, its regression and classification outputs provide the predictive information required by the downstream decision-support layer. The predicted class-probability distribution is subsequently transformed into an entropy-based operational uncertainty index and incorporated into the physics-aware multi-objective optimization framework described in the following sections.

4.4. Robust Learning and Training-Set Preparation

The second modeling phase introduced controlled disturbances into the training data to evaluate whether the predictive models could maintain useful behavior under imperfect sensing conditions. The robust-training procedure consisted of three sequential operations: training-set balancing, Gaussian input perturbation, and random feature dropout.

All three operations were applied exclusively to the training data. Validation and test data remained unchanged.

4.4.1. Training-Set Balancing

Because the operational-status classes were not equally represented in the SCADA dataset, manual oversampling was applied to the training set in Phase 2. The purpose was to reduce the dominance of frequently occurring operating states and provide the models with additional exposure to less frequent classes.

For each class c, let $N_c$denote the number of training samples belonging to that class. A target minimum count of 1000 samples per class was used. Classes containing fewer than 1000 training samples were randomly resampled with replacement until the target count was reached, whereas classes containing at least 1000 samples were retained without additional duplication.

The resulting number of samples for class c was therefore defined as:

$$N_c^* = \begin{cases} N_c, & N_c \geq 1000 \\ 1000, & N_c < 1000 \end{cases} \quad (4.4)$$

Oversampling was performed exclusively on the training subset. The validation and test subsets retained their original class distributions.

4.4.2. Gaussian Input Perturbation

Following training-set balancing, Gaussian perturbations were introduced into the standardized training features. For an input vector x, the perturbed vector was defined as:

$$\tilde{\boldsymbol{x}} = \boldsymbol{x} + \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \boldsymbol{\mathcal{N}}(\mathbf{0}, \boldsymbol{\sigma^2 I}) \quad (4.5)$$

where ϵdenotes a zero-mean Gaussian perturbation vector, $I$is the identity matrix, and the standard deviation was set to:

$$\sigma = 0.01 \quad (4.6)$$

The perturbation was independently generated for the training samples and features. Because the perturbation was introduced after standardization, the magnitude of the disturbance was expressed relative to the standardized feature scale.

4.4.3. Random Feature Dropout

In addition to additive Gaussian noise, random feature dropout was applied to the perturbed training data. A Bernoulli mask was independently generated for each feature value with dropout probability:

$$p_{\text{drop}} = 0.05 \quad (4.7)$$

The robustified input was therefore expressed as:

$$x^{(r)} = m \odot (x + \epsilon), \quad m_j \sim \text{Bernoulli}\left(1 - p_{\text{drop}}\right) \quad (4.8)$$

where $\odot$ denotes element-wise multiplication and $m_j$is the binary mask associated with feature j.

Consequently, approximately 5% of individual feature values were randomly replaced by zero during each perturbation operation. This procedure was intended to emulate partial sensor information loss or temporary measurement dropout and to reduce excessive dependence on any individual sensor variable.

4.4.4. Robust Training Procedure

The complete robust-training procedure consisted of the following sequence:

$$D_{\text{train}} \xrightarrow{\text{Oversampling}} D_{\text{train}}^{*} \xrightarrow{\text{Gaussian Noise}} D_{\text{noise}} \xrightarrow{\text{Feature Dropout}} D_{\text{robust}} \quad (4.9)$$

where D_{robust}denotes the final robustified training set.

The validation and test data were deliberately excluded from these transformations. They were retained in their original form to provide an unbiased evaluation of predictive performance under unseen operating conditions.

The robust versions of XGBoost, LightGBM, and MTL-Net were subsequently used as the predictive components of the proposed framework. In the optimization stage, the robust MTL-Net supplied both predicted power and operational-status probabilities.

4.5. Multi-Objective Metaheuristic Optimization

Following construction of the robust predictive core, the optimization stage was used to identify operating states that simultaneously improve predicted power output and reduce operational uncertainty.

Six multi-objective optimization algorithms were employed: NSGA-II, MOPSO, MOEA/D, hybrid MOGWO, enhanced MOWOA, and enhanced MOGOA. The algorithms represent different search paradigms, including dominance-based evolutionary search, particle-based search, decomposition-based evolutionary optimization, and modified nature-inspired population search.

NSGA-II employs non-dominated sorting and crowding-distance mechanisms to preserve both convergence and diversity in multi-objective search (Deb et al., 2002). MOEA/D decomposes a multi-objective problem into a collection of scalar optimization subproblems associated with different search directions (Zhang and Li, 2007). MOGWO extends the Grey Wolf Optimizer to multi-objective optimization through an external archive and Pareto-based leader selection (Mirjalili et al., 2016). The underlying Whale Optimization Algorithm and Grasshopper Optimization Algorithm were originally introduced as population-based metaheuristics for nonlinear optimization (Mirjalili and Lewis, 2016; Saremi et al., 2017).

For all six algorithms, the optimization was performed over the same seven decision variables:

$$x = [\omega r, \omega g, Tg, Vs, Vmax, d, \Delta\theta, \beta] \quad (4.10)$$

where ωr denotes rotor speed, ωg denotes generator speed, Tg denotes generator temperature, Vs denotes supply voltage, $Vmax$,d denotes the maximum daily wind-speed indicator, $\Delta\theta$denotes wind-direction offset, and β denotes pitch angle.

The instantaneous wind speed, v, was not treated as a decision variable. Instead, it was externally specified and fixed for each optimization run. Thus, each optimizer solved a seven-dimensional decision problem independently for a prescribed wind-speed condition.

The optimization experiments were conducted over 23 operating conditions uniformly distributed between 3 and 25 m/s.

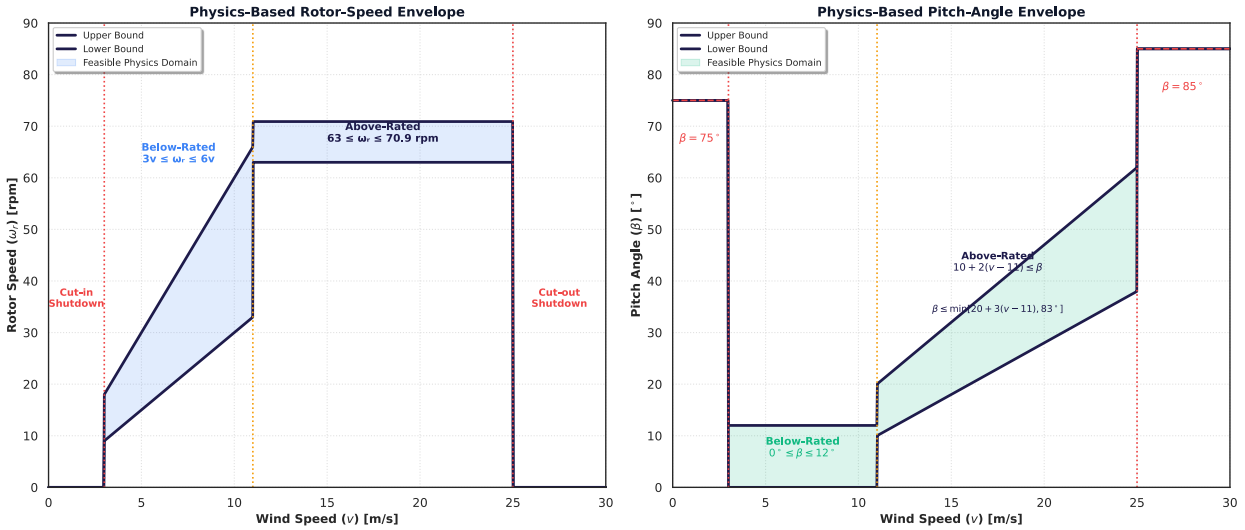


Figure 4.2 Physics-based feasible operating envelopes for rotor speed and pitch angle used in the optimization repair mechanism.

For every candidate solution, the same physics-based repair mechanism was first applied to enforce the prescribed operating envelope and physical constraints. The repair procedure adjusted rotor speed and pitch angle according to the wind-speed regime and enforced the generator-speed constraint through the specified gearbox-ratio bounds and generator-speed limit. The remaining bounded operational variables were also restricted to their prescribed physical ranges. The repaired input vector was subsequently standardized using the training-stage feature scaler and evaluated by the robust MTL-Net. The regression head supplied the predicted power, while the classification head generated the class-probability distribution used for uncertainty quantification.

The resulting evaluation pipeline was:

$$X \rightarrow PhysicsRepair(X) \rightarrow Standardization \rightarrow RobustMTL - Net \rightarrow P(X), R(X) \quad (4.11)$$

where $P(X)$denotes the predicted power output and $R(X)$denotes the $uncertainty/risk$ measure derived from the classification probabilities.

The two optimization objectives were formulated as minimizing negative predicted power and operational uncertainty:

$$minF(X) = [-P(X), R(X)] \quad (4.12)$$

where X represents the seven-dimensional decision vector.

Because all six optimizers interacted with the same surrogate model, objective functions, variable bounds, wind-speed conditions, and physics-based repair procedure, the comparison focused on differences arising from the search mechanisms and their algorithm-specific configurations.

4.5.1. Common Optimization Framework

The optimization was formulated as a two-objective minimization problem. Operational risk was minimized directly, while power maximization was represented through the negative predicted power objective. The complete mathematical formulation of the objectives, entropy-based risk measure, physical constraints, and penalty mechanism was established in the preceding methodological section and is therefore not repeated here.

The principal characteristics of the six optimization configurations are summarized in Table 4.6.

Table 4.6 Configuration of the multi-objective metaheuristic optimizers.

Optimizer	Search strategy	Population / swarm	Main iterations / evaluations	Diversity mechanism
NSGA-II	Non-dominated sorting and genetic evolution	200	8,000 evaluations	Crowding distance, SBX, polynomial mutation
MOPSO	Multi-objective particle swarm optimization	120	400 iterations	External Pareto archive and crowding-distance leader selection
MOEA/D	Decomposition-based evolutionary optimization	—	10,000 evaluations	Neighborhood-based decomposition
Hybrid MOGWO	Multi-objective Grey Wolf Optimization	50	100 iterations	DE mutation + Lévy flight + external archive
Enhanced MOWOA	Multi-objective Whale Optimization	60	80 iterations	Pareto archive, crowding-based leaders + Lévy perturbation
Enhanced MOGOA	Multi-objective Grasshopper Optimization	120	120 iterations	Adaptive exploration/exploitation + Lévy flight + mutation

The reported population sizes and iteration/evaluation settings correspond to the implemented configurations. In every case, the number of decision variables was seven, while wind speed remained externally fixed.

4.5.2. NSGA-II

NSGA-II was employed as a representative elitist evolutionary multi-objective optimizer. The population contained 200 candidate solutions, and the evolutionary search was conducted for 8,000 function evaluations.

Simulated binary crossover (SBX) was applied with a probability of 0.9 and a distribution index of 15. Polynomial mutation was applied with a probability of 0.2 and a distribution index of 20.

The algorithm ranked candidate solutions using Pareto dominance and employed crowding distance to preserve diversity along the non-dominated front. At every prescribed wind-speed condition, the seven decision variables were evolved while the wind-speed condition remained fixed. Each candidate was subsequently passed through the common physics-repair and predictive-evaluation pipeline.

4.5.3. MOPSO

A multi-objective particle swarm optimization framework was implemented using an external archive to preserve non-dominated solutions. The swarm consisted of 120 particles and was evolved for 400 iterations.

The inertia weight was set to $w = 0.72$, while the cognitive and social coefficients were:

$$c_1 = c_2 = 1.7 \quad (4.13)$$

The maximum velocity in each decision dimension was restricted to 15% of the corresponding variable range. The external archive was maintained according to Pareto dominance, while crowding distance was used to preferentially select leaders from less densely populated regions.

A mutation probability of 0.15 was additionally introduced to increase exploration and reduce premature convergence. The resulting archive represented an approximation of the Pareto solution set for each fixed wind-speed condition.

4.5.4. MOEA/D

MOEA/D was included to provide a decomposition-based optimization paradigm that differs from the dominance-based mechanism of NSGA-II. The method decomposes the multi-objective problem into a set of scalar subproblems associated with different weight directions (Zhang and Li, 2007).

A neighborhood size of 30 was employed, with 99 outer divisions used to generate the search directions. The same SBX crossover and polynomial mutation operators used in NSGA-II were adopted, with crossover probability 0.9, crossover distribution index 15, mutation probability 0.2, and mutation distribution index 20.

The algorithm was executed for 10,000 evaluations for each wind-speed condition.

4.5.5. Hybrid MOGWO

A hybrid Multi-Objective Grey Wolf Optimizer was implemented by combining the leadership and encircling mechanisms of MOGWO with additional diversity-enhancing operators. The underlying MOGWO framework uses an external archive of non-dominated solutions and Pareto-based leader selection (Mirjalili et al., 2016).

The implemented population consisted of 50 wolves and the optimization proceeded for 100 iterations. At each iteration, three leading non-dominated solutions were selected as the α, β, and δ leaders.

The conventional GWO control parameter was linearly reduced from 2 to zero during optimization. Differential-evolution mutation was applied with a probability of 0.5, while Lévy-flight perturbation was applied with a probability of 0.2.

Following each position update, candidate solutions were projected onto the predefined variable bounds and processed through the common physics-repair mechanism.

The resulting hybrid configuration was treated as a distinct algorithmic implementation and is referred to throughout the study as **hybrid MOGWO**.

4.5.6. Enhanced MOWOA

An enhanced Multi-Objective Whale Optimization Algorithm was implemented using an external non-dominated archive and crowding-distance-based leader selection. The underlying Whale Optimization Algorithm was introduced as a population-based optimization method inspired by the hunting behavior of humpback whales (Mirjalili and Lewis, 2016).

The population contained 60 candidate solutions and was evolved for 80 iterations.

At each iteration, three archive members were selected as leading solutions. Their positions were incorporated into the population-update mechanism. A Lévy-flight perturbation with stability parameter $\beta = 1.5$was applied with probability 0.25 to increase exploration.

The external archive was filtered according to Pareto dominance and limited to 200 solutions using crowding distance. The resulting implementation is referred to as **enhanced MOWOA** to distinguish it from the conventional single-objective WOA.

4.5.7. Enhanced MOGOA

An enhanced Multi-Objective Grasshopper Optimization Algorithm was employed as the sixth optimization strategy. The underlying Grasshopper Optimization Algorithm was originally introduced as a population-based optimization method for nonlinear engineering optimization (Saremi et al., 2017).

The implemented population consisted of 120 candidate solutions and the optimization proceeded for 120 iterations. The external archive had a capacity of 400 solutions.

The enhancement incorporated an adaptive exploration-exploitation mechanism. The exploration factor was progressively reduced during optimization, while the exploitation component increased as the search progressed.

Three archive leaders were selected probabilistically according to crowding distance, and their positions were combined to guide the population toward promising non-dominated regions.

A Lévy-flight perturbation with $\beta = 1.5$was incorporated during exploitation. In addition, a mutation operator with probability 0.15 was used to introduce local stochastic perturbations.

Following each position update, the wind-speed variable remained fixed at the prescribed operating condition and the resulting solution was passed through the common physics-repair mechanism.

The term **enhanced MOGOA** is used throughout this study to explicitly distinguish the implemented multi-objective and modified configuration from the conventional Grasshopper Optimization Algorithm.

4.5.8. Comparative Optimization Protocol

To ensure methodological consistency, all six optimizers shared the same:

1. seven-dimensional decision space;
2. decision-variable bounds;
3. 23 prescribed wind-speed conditions;
4. robust MTL-Net surrogate;
5. objective formulation;
6. operational-risk calculation;
7. physical-repair mechanism;
8. feature-scaling procedure; and
9. post-processing procedure.

Only the population-search mechanism and its algorithm-specific parameters differed.

For each of the 23 wind-speed conditions, every optimizer generated a set of non-dominated operating points. The resulting solutions were subsequently aggregated to construct the overall risk-power trade-off landscape.

Duplicate and physically invalid solutions were removed during post-processing. The retained solutions preserved the complete operating vector together with the corresponding predicted power and risk values.

The resulting Pareto solutions formed the basis for the subsequent operating-policy extraction stage. Representative SAFE, Knee-Point, and MAX-POWER operating points were subsequently identified for different wind-speed regimes. This policy layer converts the continuous Pareto solution space into interpretable operating states while preserving the underlying risk-performance trade-off.

4.6. Evaluation Metrics

The developed framework was evaluated at three complementary levels: predictive accuracy, Pareto-front quality, and independent consistency of optimized operating states.

Regression and classification metrics were calculated on the held-out test set. Multi-objective optimization was evaluated using the retained non-dominated solution sets. Finally, optimized operating states were independently evaluated using a separately trained LightGBM power regressor.

4.6.1. Regression Metrics

The predictive performance for turbine power was evaluated using Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Median Absolute Error (MedianAE), Maximum Error, Bias, coefficient of determination (R^2), adjusted coefficient of determination (R^2_{adj}), Explained Variance (EV), and Normalized Root Mean Squared Error (NRMSE).

For Nobservations, where $y_i$denotes the measured power and $\hat{y}_i$denotes the predicted power, MAE was calculated as:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \quad (4.14)$$

MSE was calculated as:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \quad (4.15)$$

RMSE was calculated as:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \quad (4.16)$$

MedianAE was defined as:

$$MedianAE = \text{median}(|y_i - \hat{y}_i|) \quad (4.17)$$

The maximum absolute prediction error was defined as:

$$MaxError = \max_i|y_i - \hat{y}_i| \quad (4.18)$$

Prediction bias was calculated as:

$$Bias = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i) \quad (4.19)$$

The coefficient of determination was calculated as:

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2} \quad (4.20)$$

where $\bar{y}$denotes the mean measured power.

The adjusted coefficient of determination was calculated as:

$$R^2_{\text{adj}} = 1 - (1 - R^2)\frac{N-1}{N-p-1} \quad (4.21)$$

where $p$denotes the number of predictor variables.

Explained Variance was calculated as:

$$EV = 1 - \frac{\text{Var}(y - \hat{y})}{\text{Var}(y)} \quad (4.22)$$

Finally, NRMSE was calculated using the observed power range:

$$NRMSE = \frac{RMSE}{y_{\max} - y_{\min}} \quad (4.23)$$

For the reported metrics, predictions were transformed back to the original physical power scale before metric calculation.

4.6.2. Classification Metrics

Classification performance was evaluated on the complete held-out test set using complementary metrics designed to characterize both overall performance and class-balanced behavior.

The primary metrics included Accuracy, Precision, Recall, F1-score, and Balanced Accuracy. Precision, Recall, and F1-score were reported using both macro and weighted averaging. Additional metrics included Matthews Correlation Coefficient (MCC), Cohen's Kappa, Jaccard Score, and Hamming Loss.

For Nevaluated samples, Accuracy was defined as:

$$\text{Accuracy} = \frac{1}{N}\sum_{i=1}^{N}\mathbb{I}\,(y_i = \hat{y}_i) \quad (4.24)$$

where $y_i$and $\hat{y}_i$denote the reference and predicted class labels, respectively.

For class k, Precision and Recall were calculated as:

$$\text{Precision}_k = \frac{TP_k}{TP_k + FP_k} \quad (4.25)$$

$$\text{Recall}_k = \frac{TP_k}{TP_k + FN_k} \quad (4.26)$$

The class-specific F1-score was calculated as:

$$F1_k = 2\frac{\text{Precision}_k \text{Recall}_k}{\text{Precision}_k + \text{Recall}_k} \quad (4.27)$$

For Koperational-status classes, the macro-average was calculated as:

$$\text{Metric}_{\text{macro}} = \frac{1}{K}\sum_{k=1}^{K} \text{Metric}_k \quad (4.28)$$

The weighted average was calculated as:

$$\text{Metric}_{\text{weighted}} = \sum_{k=1}^{K} \frac{n_k}{N} \text{Metric}_k \quad (4.29)$$

where $n_k$denotes the number of samples belonging to class k.

Balanced Accuracy was calculated as the arithmetic mean of class-specific recall values:

$$\text{Balanced Accuracy} = \frac{1}{K}\sum_{k=1}^{K} \text{Recall}_k \quad (4.30)$$

This metric was particularly relevant because the operational-status classes were substantially imbalanced.

MCC, Cohen's Kappa, Jaccard Score, and Hamming Loss were retained as complementary classification indicators. Hamming Loss was calculated as:

$$\text{Hamming Loss} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}\,(y_i \neq \hat{y}_i) \quad (4.31)$$

For the single-label multiclass setting considered here, Hamming Loss satisfies:

$$\text{Hamming Loss} = 1 - \text{Accuracy} \quad (4.32)$$

Because the operational-status distribution was imbalanced, macro-averaged Precision, Recall, and F1-score, together with Balanced Accuracy, were emphasized when interpreting class-balanced behavior. Overall Accuracy and weighted metrics were retained to characterize aggregate predictive behavior over the complete test population.

The additional metrics were treated as complementary outputs rather than mandatory comparison criteria for every robustness configuration.

4.6.3. Pareto-Front Quality Metrics

The quality of the non-dominated solution sets generated by the six optimization algorithms was evaluated using the Number of Nondominated Solutions (NOS), Spacing Metric (SM), and Maximum Spread (MS).

The Number of Nondominated Solutions represents the number of valid non-dominated solutions retained in the final Pareto set. A larger NOS indicates a larger retained set of trade-off solutions but was not interpreted independently as a direct measure of optimization quality.

The Spacing Metric was calculated according to the actual implementation used in this study. For each solution, the Manhattan distance to its nearest neighboring solution in objective space was first calculated. Let d_i denote this nearest-neighbor distance and $\bar{d}$ denote the mean of all nearest-neighbor distances. The implemented spacing measure was:

$$SM = \sqrt{\frac{\sum_{i=1}^{N}(\bar{d} - d_i)^2}{N-1}} \quad (4.33)$$

where the pairwise distance used to determine d_i was:

$$d_i = \min_{j \neq i} \sum_{k=1}^{m} |f_{ik} - f_{jk}| \quad (4.34)$$

A lower SM therefore indicates a more uniform distribution of neighboring solutions according to the implemented metric.

The Maximum Spread was calculated directly from the implementation as the Euclidean extension of the retained Pareto set across the objective dimensions:

$$MS = \sqrt{\sum_{j=1}^{m} \left(f_j^{\max} - f_j^{\min}\right)^2} \quad (4.35)$$

where m is the number of objectives, and $f_j^{\max}$ and $f_j^{\min}$ represent the maximum and minimum values of objective j among the retained non-dominated solutions.

A larger MS indicates a broader extension of the obtained solution set across objective space.

Because the optimization problem was internally formulated as a minimization problem, power was represented by its negative value. For presentation and interpretation, the objective values were converted back to the physical power representation.

Importantly, the definitions above correspond to the **implemented calculation procedures**, ensuring that the reported SM and MS values are directly reproducible from the experimental code.

4.6.4. Independent Validation of Optimized Operating States

The optimized operating states generated during Phase 3 do not necessarily correspond to previously observed SCADA samples. Consequently, a direct comparison with a measured power value is not possible for every optimized operating state.

An independent model-based validation procedure was therefore employed to assess the consistency of the optimized state space. The previously trained LightGBM power regressor was used as an independent evaluator without additional retraining.

Each optimized operating state generated by the optimization algorithms was supplied independently to the trained LightGBM evaluator. The resulting power estimate was denoted by P_{LGB}. This value was compared with the power predicted by the MTL-Net embedded within the optimization framework, denoted by P_{opt}.

The independent Mean Absolute Error was calculated as:

$$MAE_{\mathrm{ind}} = \frac{1}{N}\sum_{i=1}^{N}\left|P_{\mathrm{opt},i} - P_{\mathrm{LGB},i}\right| \quad (4.36)$$

The corresponding Mean Squared Error was:

$$MSE_{\mathrm{ind}} = \frac{1}{N}\sum_{i=1}^{N}\left(P_{\mathrm{opt},i} - P_{\mathrm{LGB},i}\right)^2 \quad (4.37)$$

The Pearson correlation coefficient was calculated as:

$$r = \frac{\sum_{i=1}^{N}\left(P_{\mathrm{opt},i} - \bar{P}_{\mathrm{opt}}\right)\left(P_{\mathrm{LGB},i} - \bar{P}_{\mathrm{LGB}}\right)}{\sqrt{\sum_{i=1}^{N}\left(P_{\mathrm{opt},i} - \bar{P}_{\mathrm{opt}}\right)^2}\sqrt{\sum_{i=1}^{N}\left(P_{\mathrm{LGB},i} - \bar{P}_{\mathrm{LGB}}\right)^2}} \quad (4.38)$$

where P_{opt}represents the power predicted by the MTL-Net embedded in the optimization process and P_{LGB}represents the independently obtained LightGBM prediction.

The maximum predicted power among the evaluated optimized states was also reported to characterize the upper operating envelope identified by each optimization configuration.

This independent evaluation was designed as a **cross-model consistency check** rather than as a conventional measurement-based validation against newly acquired SCADA observations. The distinction is important because optimized operating states may lie outside the exact states represented in the historical dataset. In contrast, the predictive performance metrics reported for the regression and classification tasks were calculated against actual held-out SCADA targets.

The independent evaluation therefore provides complementary evidence regarding the stability of the optimized operating states across predictive representations, while the held-out test-set results remain the primary basis for assessing predictive accuracy.

5. Results

5.1. Predictive Performance

The predictive performance of the three learning architectures was first evaluated under nominal operating conditions using the original SCADA data, prior to the introduction of the robustness-oriented perturbations. XGBoost and LightGBM were employed as single-task regression models for PowerOutput prediction, whereas the proposed Deep Multi-Task Learning (MTL) architecture jointly modeled PowerOutput and turbine operating status. To provide a consistent basis for comparison, the primary regression results are reported on the independent test set, as summarized in Table 5.1.

Table 5.1 Test-set regression performance of the predictive models under nominal operating conditions.

Model	MAE	RMSE	R^2
XGBoost	0.0135	0.0452	0.9995
LightGBM	0.0136	0.0403	0.9996
Deep MTL	0.0473	0.0750	0.9944

All three architectures demonstrated a high level of predictive accuracy under nominal operating conditions. XGBoost and LightGBM exhibited highly similar regression performance, with test-set R^2 values of 0.9995 and 0.9996, respectively. LightGBM achieved the lowest RMSE, with a value of 0.0403, whereas XGBoost produced a marginally lower MAE of 0.0135 compared with 0.0136 for LightGBM. The small difference between the two tree-based models indicates that both learned highly consistent mappings between the SCADA input variables and PowerOutput under the nominal operating conditions.

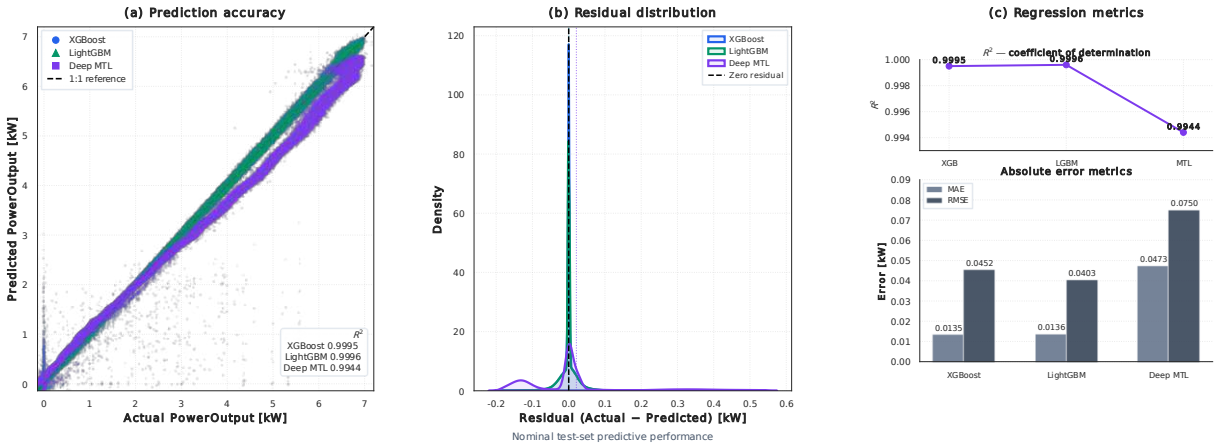


Figure 5.1 Nominal test-set predictive performance of XGBoost, LightGBM, and Deep MTL: (a) actual versus predicted PowerOutput, (b) residual distributions, and (c) regression-metric comparison.

The Deep MTL architecture achieved a test-set R^2 of 0.9944, with an MAE of 0.0473 and an RMSE of 0.0750. Its regression errors were higher than those obtained by the two single-task tree-based models. However, direct comparison of regression error alone does not fully characterize the role of the Deep MTL architecture because the model simultaneously performs turbine-status classification through its second output head. Accordingly, its predictive performance is considered jointly with the classification results reported in Section 5.2.

The consistency of the tree-based models across the available data partitions was also examined during model development. XGBoost achieved MAE values of 0.0130, 0.0133, and 0.0135 and RMSE values of 0.0389, 0.0430, and 0.0452 on the training, validation, and test sets, respectively. The corresponding LightGBM MAE values were 0.0134, 0.0134, and 0.0136, while the RMSE values were 0.0381, 0.0397, and 0.0403. The R^2 values remained approximately within the 0.9995–0.9996 range across the evaluated partitions. The relatively small changes in the error metrics between the training, validation, and test sets indicate consistent predictive behavior across the data partitions.

These nominal results establish the predictive baseline for the subsequent robustness analysis. The Deep MTL model selected for the optimization stage was subsequently used as the surrogate predictive engine. Importantly, the trained Deep MTL model was not retrained during the optimization procedure. Instead, the fixed predictive model was used to evaluate candidate operating states generated by the multi-objective optimization algorithms.

5.2. Classification Performance

The classification component of the predictive framework was evaluated using the turbine operating-status labels defined in the SCADA dataset. Because the operating-status classes exhibited a highly imbalanced distribution, overall accuracy was considered together with macro-averaged and weighted-averaged precision, recall, and F1-score. Macro-averaging assigns equal

importance to individual classes, whereas weighted averaging accounts for the number of observations belonging to each class. This distinction is particularly important for imbalanced multiclass classification, where a high overall accuracy can coexist with substantially weaker recognition of minority classes (Sokolova & Lapalme, 2009).

The resulting test-set classification performance is summarized in Table 5.2.

Table 5.2 Test-set classification performance under nominal operating conditions.

Model	Accuracy	Macro Precision	Macro Recall	Macro F1	Weighted Precision	Weighted Recall	Weighted F1
XGBoost	0.9995	0.6143	0.4938	0.5374	0.9994	0.9995	0.9994
LightGBM	0.8677	0.2610	0.2219	0.2007	0.9611	0.8677	0.8789
Deep MTL	0.9994	0.6052	0.4229	0.4767	0.9993	0.9994	0.9993

The results reveal a substantial difference between overall accuracy and macro-averaged performance, particularly for the imbalanced multiclass classification task. XGBoost and Deep MTL achieved overall accuracies of 0.9995 and 0.9994, respectively, together with weighted F1-scores of 0.9994 and 0.9993. In contrast, their macro-F1 values were 0.5374 and 0.4767. This discrepancy indicates that the high overall and weighted scores are strongly influenced by the dominant operating-status classes, whereas macro-averaging gives equal contribution to each class and therefore provides a more conservative assessment of minority-class recognition (Sokolova & Lapalme, 2009).

The class-wise prediction patterns are further illustrated by the normalized confusion matrices in Figure. 5.2. Normalization by the true class allows the recognition performance of individual operating-status classes to be examined independently of their different sample frequencies.

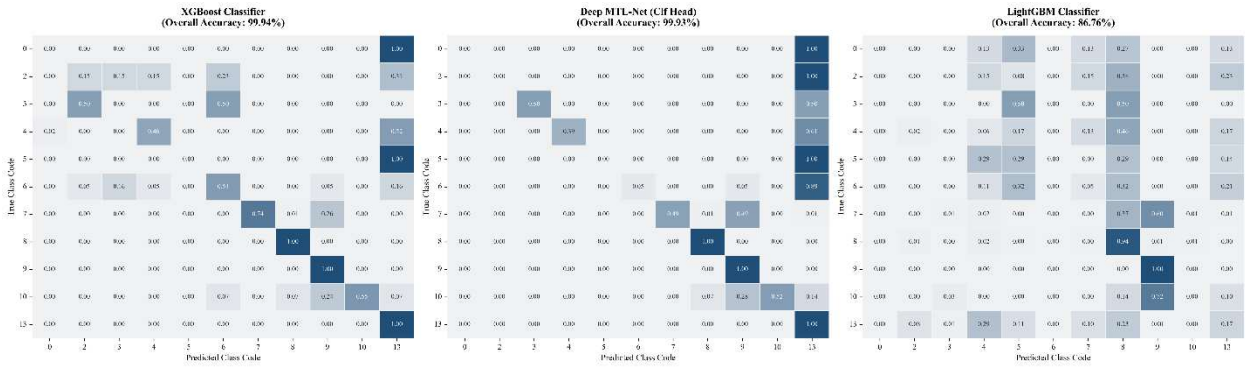


Figure 5.2 Normalized confusion matrices for test-set operational-status classification using XGBoost, LightGBM, and Deep MTL.

The class-level results further support this interpretation. For XGBoost, the dominant classes 9, 10, and 13 were classified with very high performance, whereas several low-frequency classes

exhibited substantially lower recall. In particular, the recall values for classes 2, 5, 7, 8, and 12 were 0.15, 0.46, 0.53, 0.74, and 0.55, respectively. A similar pattern was observed for Deep MTL, for which the dominant classes achieved near-perfect classification performance while several minority classes showed lower and more variable recall. For classes 5, 7, 8, and 12, the corresponding recall values were 0.39, 0.21, 0.54, and 0.52, respectively.

LightGBM exhibited a distinct classification behavior under the nominal training configuration. Its overall accuracy was 0.8677, while its macro-F1 and weighted-F1 values were 0.2007 and 0.8789, respectively. The substantially lower macro-averaged scores indicate that its predictions were less consistent across the full set of operating-status classes, particularly for minority states. This model-dependent behavior highlights that high-capacity tree-based classifiers do not necessarily exhibit identical class-recognition characteristics when trained on the same imbalanced multiclass dataset. In the present case, the observed difference is associated with the interaction between the class distribution, feature space, and model-specific learning characteristics. Therefore, the LightGBM result was retained as part of the nominal comparative evaluation rather than being excluded on the basis of its lower aggregate accuracy.

Importantly, the nominal classification results should be interpreted as a baseline rather than as the final robustness assessment of the proposed framework. The subsequent robust-training stage was specifically introduced to improve the resilience of the predictive models under perturbed and degraded operating conditions. Accordingly, the classification behavior under robustness-oriented training is evaluated separately in the corresponding robustness analysis.

These results demonstrate why accuracy and weighted metrics alone are insufficient for characterizing the present classification problem. The very high weighted scores of XGBoost and Deep MTL should not be interpreted as uniformly high performance across all operating-status classes. Instead, the macro-averaged measures expose the remaining difficulty associated with minority operating states and provide a complementary evaluation that is less dominated by the majority classes (Sokolova & Lapalme, 2009).

The classification capability of Deep MTL is particularly relevant to the subsequent optimization stage. Its classification head produces a probability distribution over the operating-status classes, which is subsequently used to construct the entropy-based risk objective. Consequently, the Deep MTL model provides not only a power prediction but also a model-derived measure of operating-state uncertainty that can be incorporated into the multi-objective optimization process.

5.3. Robustness Analysis

To evaluate the stability of the predictive framework under imperfect sensing conditions, the models were assessed using perturbed SCADA inputs containing simulated noise and data disturbances. The purpose of this experiment was to determine whether the learned predictive relationships remain stable when the input measurements deviate from their nominal values. The

analysis therefore focuses on predictive robustness under perturbed inputs rather than on a complete sensor-outage scenario.

The robustness results are summarized in Table 5.3. Regression performance was evaluated using Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), coefficient of determination (R^2), and normalized RMSE (NRMSE). The classification component was evaluated using overall accuracy together with macro- and weighted-averaged F1-scores.

Table 5.3 Predictive performance of the robustness-oriented models under perturbed input conditions.

Model	MAE	RMSE	R^2	NRMSE	Accuracy	F1-macro	F1-weighted
Robust XGBoost	0.0262	0.0628	0.9990	0.0087	0.9992	0.4720	0.9992
Robust LightGBM	0.0208	0.0577	0.9991	0.0080	0.9990	0.4490	0.9990
Robust Deep MTL	0.0616	0.1284	0.9957	0.0178	0.9982	0.4474	0.9986

All three robustness-oriented models retained high predictive performance under the evaluated perturbation conditions. Robust LightGBM achieved an MAE of 0.0208, RMSE of 0.0577, R^2 of 0.9991, and NRMSE of 0.0080. Robust XGBoost obtained an MAE of 0.0262, RMSE of 0.0628, R^2 of 0.9990, and NRMSE of 0.0087. Robust Deep MTL achieved an MAE of 0.0616, RMSE of 0.1284, R^2 of 0.9957, and NRMSE of 0.0178.

Although the absolute regression errors differed among the three models, all models maintained high coefficients of determination under the evaluated perturbations. In particular, the R^2 values remained above 0.995, indicating that the dominant variation in power output continued to be captured despite the introduced disturbances. The NRMSE values likewise remained relatively small, ranging from 0.0080 for Robust LightGBM to 0.0178 for Robust Deep MTL.

The classification results also indicate substantial stability under the perturbed conditions. Robust XGBoost achieved an overall accuracy of 0.9992, with macro-F1 and weighted-F1 values of 0.4720 and 0.9992, respectively. Robust LightGBM obtained an accuracy of 0.9990, macro-F1 of 0.4490, and weighted-F1 of 0.9990. Robust Deep MTL achieved an accuracy of 0.9982, macro-F1 of 0.4474, and weighted-F1 of 0.9986.

The persistent difference between macro- and weighted-averaged F1-scores is consistent with the highly imbalanced distribution of the operating-status classes. Thus, the high weighted-F1 values primarily reflect the strong recognition of the dominant classes, whereas the lower macro-F1 values indicate that recognition remains less uniform across the full set of operating states. This distinction is important when interpreting robustness under imbalanced multiclass conditions, since macro-averaged measures give greater visibility to the performance of less frequent classes. (Sokolova & Lapalme, 2009).

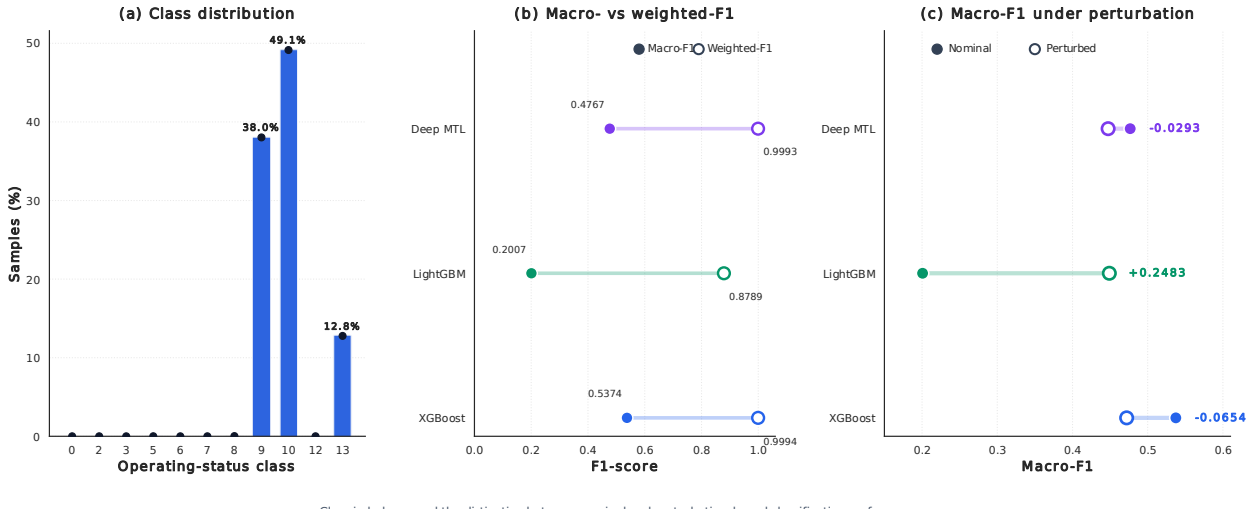


Figure 5.3 Classification imbalance and robustness. (a) Class distribution, (b) macro- versus weighted-F1, and (c) macro-F1 under perturbed conditions.

The observed robustness behavior is consistent with the respective training strategies. Robust XGBoost and Robust LightGBM were trained using perturbed observations, allowing the tree-based models to encounter deviations in the input space during model development. The Deep MTL architecture additionally incorporated Gaussian input noise and feature-level dropout within its neural architecture. These mechanisms expose the learned representation to variations in input measurements and reduce dependence on strictly nominal input conditions.

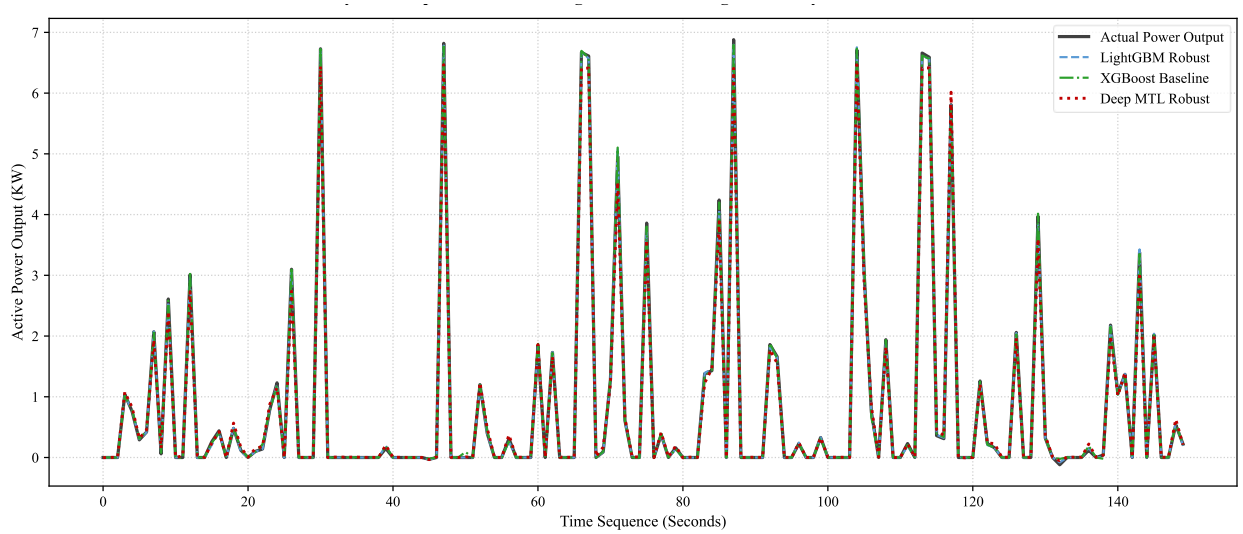


Figure 5.4 Dynamic tracking of power output under high-volatility transient conditions using the robustness-oriented predictive models.

An important distinction should therefore be made between nominal predictive accuracy and robustness. The purpose of this experiment was not to establish a universally superior predictive architecture, but to determine whether the learned predictive relationships remain stable when the input conditions depart from nominal SCADA measurements. The consistently high R^2 values and

low normalized regression errors, together with the high overall classification accuracies, indicate that the predictive framework retained substantial performance under the evaluated perturbations.

This robustness assessment provides an additional validation step before the predictive models are used within the subsequent optimization stage. In particular, it provides evidence that the selected surrogate relationships remain stable under the range of input disturbances represented in the robustness experiments. Nevertheless, the optimization stage remains subject to the physical operating constraints and repair mechanism defined in the methodology, which restrict candidate solutions to physically feasible operating envelopes.

5.4. Multi-Objective Optimization

The multi-objective optimization stage was conducted using six optimization algorithms: MOWOA, MOPSO, NSGA-II, MOEA/D, MOGOA, and MOGWO. The optimization problem simultaneously considered predicted power generation and the entropy-based operational-risk objective. The trained Deep MTL model was used as the surrogate predictive model within the optimization process, while the physical and operational constraints defined in Section 4 were imposed on candidate solutions.

The optimization procedure initially generated different numbers of candidate solutions for the six algorithms. Specifically, MOWOA generated 1282 candidate solutions, MOPSO generated 4294, NSGA-II generated 4590, MOEA/D generated 2214, MOGOA generated 429, and MOGWO generated 468 candidate solutions. These numbers correspond to the initial Pareto solution files produced by the optimization procedures before the subsequent operational wind-speed filtering.

A post-optimization feasibility filtering procedure was then applied to all generated Pareto solution sets. Specifically, only solutions satisfying the operational wind-speed interval of 3 to 14 m/s were retained. The filtering was implemented directly on the wind-speed variable, with solutions satisfying

$$3.0 \leq WindSpeed \leq 14.0 m/s \quad (5.1)$$

being retained and solutions outside this interval being removed. The retained datasets were subsequently sorted by wind speed and used for the final Pareto-front analysis.

After this filtering step, the numbers of valid retained solutions were 780 for MOWOA, 2685 for MOPSO, 2390 for NSGA-II, 1145 for MOEA/D, 314 for MOGOA, and 270 for MOGWO. Thus, the difference between the initially generated solution counts and the NOS values reported in Table 5.4 results from the explicit post-optimization operational wind-speed filtering rather than from a discrepancy between the optimization outputs.

The independent cross-model consistency assessment described in Section 5.6 was performed on the initial Pareto solution sets before the operational wind-speed filtering. The subsequent filtering was applied specifically to define the solution sets used for the final operational Pareto-front analysis and the extraction of representative operating policies.

Table 5.4 Summary of the retained Pareto solution sets generated by the six multi-objective optimization algorithms.

Algorithm	Number of Valid Solutions (NOS)	Spacing Measure (SM) ↓	Maximum Spread (MS) ↑
NSGA-II	2390	0.0065	7.3669
MOEA/D	1145	0.0180	7.3844
MOPSO	2685	0.0041	7.3293
MOGWO	270	0.0303	7.2281
MOWOA	780	0.0237	7.4019
MOGOA	314	0.0310	6.6966

The spacing measure (SM) was used to characterize the regularity of the distribution of solutions along the obtained Pareto sets, with lower values indicating more uniform spacing under the adopted implementation. The maximum-spread (MS) measure characterizes the extent of the obtained solution set in objective space. These indicators capture different geometric characteristics of a Pareto approximation and were therefore reported separately rather than combined into a single performance score (Zitzler and Thiele, 1999).

The obtained SM values ranged from 0.0041 to 0.0310, while the MS values ranged from 6.6966 to 7.4019. The variation in SM indicates differences in the regularity of the retained solutions along the corresponding Pareto fronts. Similarly, the variation in MS indicates differences in the extent of the solution sets across the considered objective space. These values are descriptive characteristics of the generated Pareto approximations and are not interpreted as an overall ranking criterion.

The Pareto solutions also exhibited different distributions across the evaluated wind-speed range. At low wind speeds, particularly between approximately 3 and 5 m/s, the solutions generated by the different algorithms showed relatively similar operating characteristics. As the operating condition approached and exceeded the rated wind-speed region, greater differences became apparent in the distribution of solutions across the power-risk objective space. This behavior is consistent with the increasing influence of the imposed operational constraints in the above-rated region.

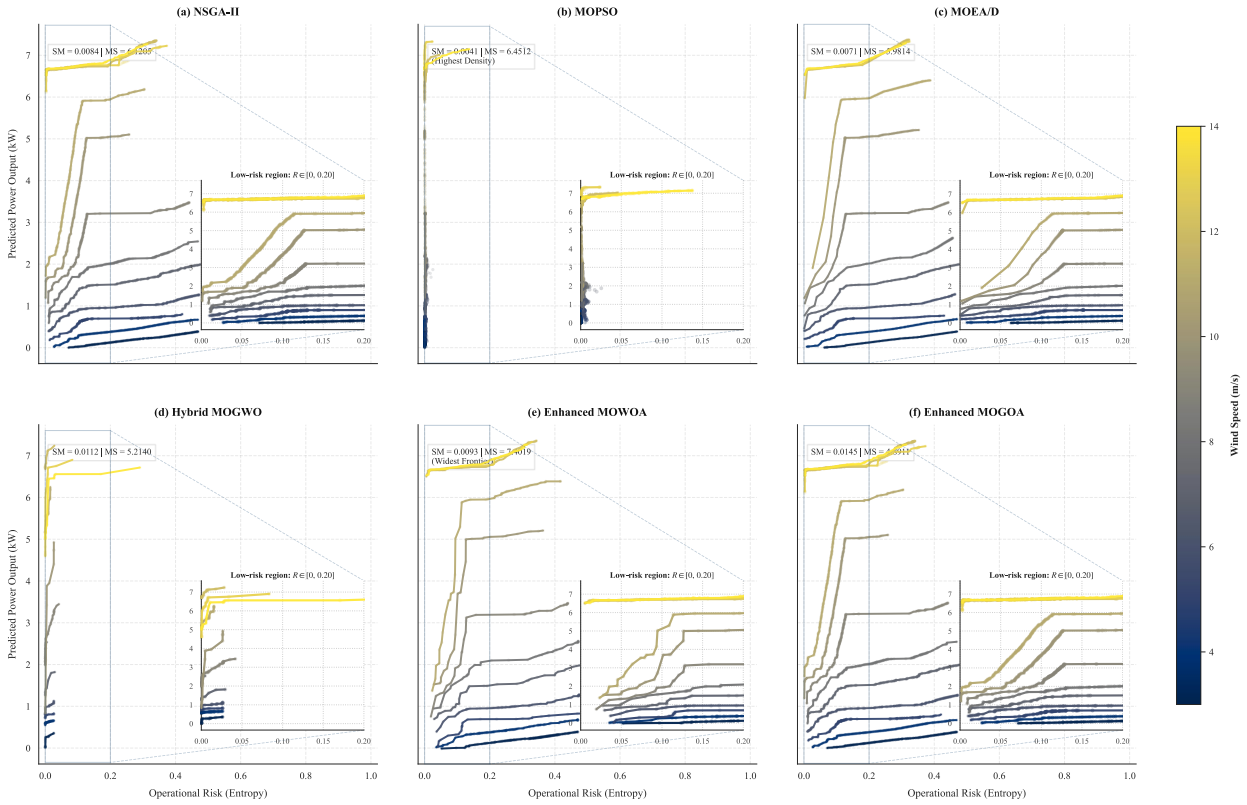


Figure 5.5 Multi-objective Pareto fronts obtained by the six optimization algorithms under different wind-speed conditions, illustrating the trade-off between predicted power output and operational risk.

The corresponding maximum predicted power values contained within the retained solution sets were 7.3580 kW for MOWOA, 7.3282 kW for MOPSO, 7.3526 kW for NSGA-II, 7.3729 kW for MOEA/D, 6.7399 kW for MOGOA, and 7.2404 kW for MOGWO.

These values represent **model-predicted power outputs generated by the surrogate-based optimization framework**. They should therefore be distinguished from the 6 kW rated power of the physical Aventa AV-7 turbine. In particular, values exceeding the rated power are retained exactly as produced by the predictive model and are not interpreted as direct measurements of physical turbine output. This distinction is important because the optimization results characterize the behavior of the learned surrogate model over the explored decision space rather than providing direct experimental measurements of turbine power at each optimized operating state.

The variation in the Pareto solutions was also reflected in the operating variables. In the below-rated region, the imposed physical constraints generally resulted in low pitch-angle values, while rotor and generator speeds provided the principal degrees of freedom for adjusting the operating point. In the above-rated region, the optimization process produced increased pitch-angle values in accordance with the imposed operational constraints, thereby regulating the aerodynamic operating condition while maintaining feasibility.

Overall, the six optimization algorithms generated feasible Pareto solution sets with different numbers of retained solutions, spacing characteristics, objective-space extensions, and maximum predicted power values. The results demonstrate that the proposed framework can generate multiple feasible operating conditions across the evaluated wind-speed range while explicitly representing the trade-off between predicted power and model-derived operational risk.

The representative operating policies extracted from these Pareto solutions are analyzed separately in Section 5.5.

5.5. Operational Policies

To translate the obtained Pareto-optimal solutions into interpretable operating strategies, three representative policies were extracted from the Pareto solutions at each evaluated wind-speed condition: **Safe**, **Knee-Point**, and **Max-Power**.

The **Safe** policy represents an operating point with emphasis on minimizing the model-derived risk objective. The **Max-Power** policy corresponds to the feasible Pareto solution providing the highest predicted power at the corresponding wind-speed condition. The **Knee-Point** policy represents an intermediate compromise between predicted power and operational risk. These policies provide an interpretable representation of the trade-off structure identified by the multi-objective optimization framework without imposing an overall ranking among the optimization algorithms.

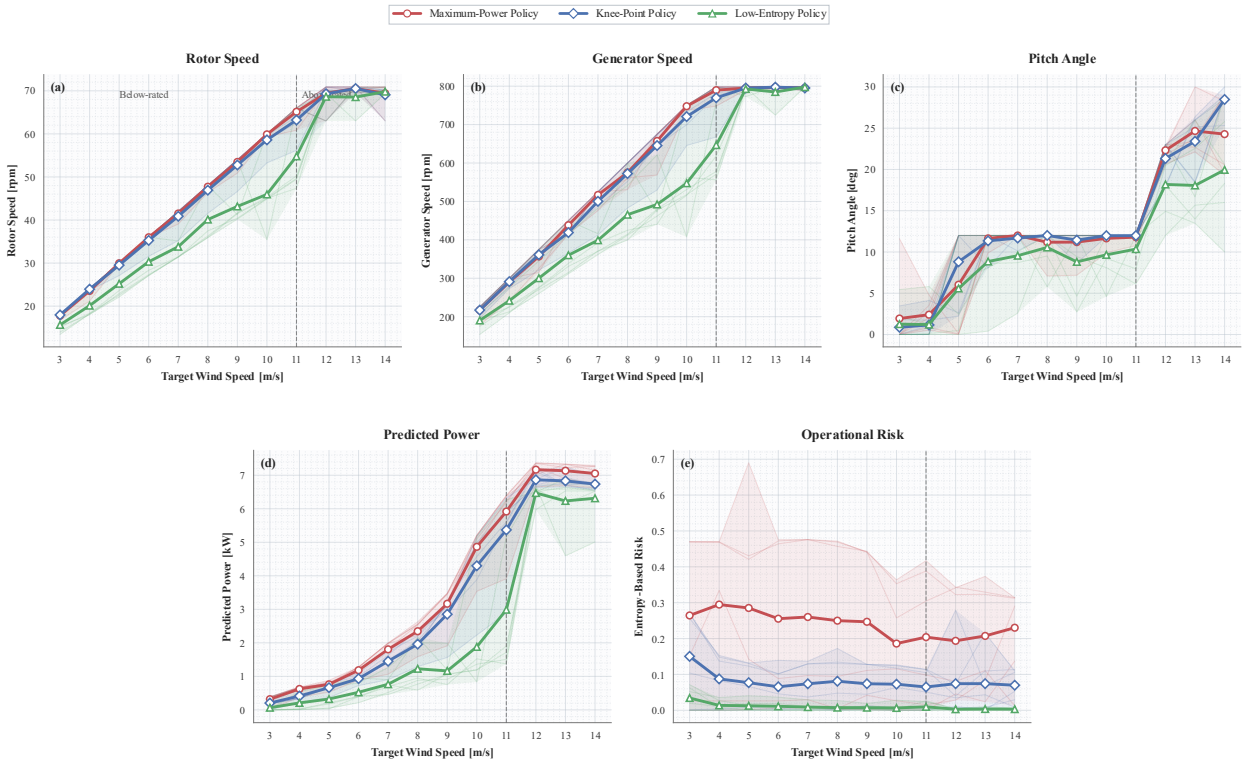


Figure 5.6 Operating characteristics of the representative Safe, Knee-Point, and Max-Power policies across wind-speed conditions: (a) rotor speed, (b) generator speed, (c) pitch angle, (d) predicted power output, and (e) model-derived operational risk.

As illustrated in Figure. 5.6, the three representative policies exhibit distinct operating characteristics across the evaluated wind-speed conditions, reflecting the trade-offs between predicted power output and the model-derived risk objective. At low wind speeds, the selected solutions generally operate with low pitch angles, while rotor and generator speeds increase with wind speed. For NSGA-II at 3 m/s, the Max-Power solution has a rotor speed of 18 and a generator speed of 225, with a pitch angle of approximately 0.0005°, whereas the corresponding Safe solution reduces the rotor and generator speeds to 15.38 and 192.29, respectively. The predicted power consequently decreases from 0.385 kW to 0.001 kW, while the risk objective decreases from 0.469 to 0.072.

At 6 m/s, the trade-off becomes more pronounced. The Max-Power policy produces 1.268 kW with a risk value of 0.472, whereas the Safe policy produces 0.396 kW with a risk value of 0.010. The Knee-Point solution provides an intermediate operating condition, yielding 0.916 kW of predicted power with a risk value of 0.102.

As wind speed increases toward and beyond the rated region, the operating policies exhibit a transition toward higher rotor and generator speeds and progressively larger pitch angles. For NSGA-II at 10 m/s, the Max-Power solution reaches a predicted power of 5.103 kW with a risk value of 0.258, whereas the Safe solution reduces the predicted power to 1.194 kW and the risk to

0.001. The Knee-Point solution provides an intermediate operating condition, yielding 4.575 kW with a risk value of 0.117.

At 14 m/s, the Max-Power, Safe, and Knee-Point policies produce predicted power values of 7.265, 6.523, and 6.747 kW, respectively. Their corresponding risk values are 0.314, 0.003, and 0.106. These results demonstrate that the extracted policies represent substantially different operating points in the power-risk objective space rather than merely different power levels.

The evolution of generator temperature further illustrates differences among the extracted policies. At 6 m/s, the Max-Power solution is associated with a generator temperature of 106.95, whereas the Safe and Knee-Point solutions maintain temperatures close to 50, with values of 49.85 and 50.06, respectively. At 10 m/s, the Safe and Knee-Point solutions similarly exhibit generator temperatures of approximately 50, while the Max-Power solution has a value of 28.10.

These temperature values should be interpreted cautiously. Generator temperature was not itself defined as the optimization objective; rather, the optimization objective was based on predicted power and the entropy-based risk measure derived from the classification output. Therefore, the observed temperature differences represent characteristics of the selected model-based operating solutions and should not be interpreted as evidence of a direct causal relationship between generator temperature and the calculated risk objective.

Overall, the extracted policies demonstrate how the Pareto solution sets can be translated into operationally interpretable decision points. The Max-Power policy emphasizes predicted power production, the Safe policy emphasizes reduction of the model-derived risk objective, and the Knee-Point policy represents an intermediate trade-off between these competing objectives. These policies are therefore better interpreted as alternative operating choices within the feasible solution space rather than as universally optimal operating modes.

5.6. Independent Cross-Model Validation

To assess the consistency of the optimized operating points, an independent cross-model validation was performed on the solutions generated during the multi-objective optimization stage. The purpose of this analysis was to determine whether an independently trained predictive model produced consistent power estimates when evaluated at the same optimized operating states.

A direct comparison with measured SCADA power at identical operating states was not possible for the complete optimization output because many optimized solutions correspond to combinations of operating variables that do not have an exact counterpart in the original SCADA observations. Accordingly, the present analysis was designed as an independent model-based validation rather than a direct measurement-based validation.

For this purpose, the previously trained LightGBM regression model (Power_LGB) was used as an independent evaluator. The model was not retrained or fine-tuned using the optimization results. Before operational wind-speed filtering, all initially generated Pareto solution points from each

optimization algorithm were supplied to the pretrained LightGBM model, and the resulting power predictions were recorded as Power_LGB. The resulting cross-model metrics therefore characterize agreement over the initial optimization-generated solution sets, whereas the final operational Pareto-front analysis and policy extraction were based on the subsequently filtered solutions satisfying the wind-speed interval of 3–14 m/s.

The power predicted by the Deep MTL surrogate used within the optimization process was then compared with the independently obtained LightGBM predictions. The comparison was performed using Mean Absolute Error (MAE), Mean Squared Error (MSE), and the Pearson correlation coefficient (r). These measures quantify the absolute deviation, squared prediction discrepancy, and linear correspondence between the two independently generated power estimates, respectively.

Table 5.5 Independent validation of optimized Pareto solutions using the pretrained LightGBM model.

Optimization Algorithm	Number of Tested Points	MAE Power	MSE Power	Pearson Correlation (r)	Maximum Extracted Power (kW)
MOWOA	1282	0.2650	0.1481	0.9925	7.3580
MOPSO	4294	0.2776	0.1462	0.9926	7.3282
NSGA-II	4590	0.3128	0.1764	0.9877	7.3526
MOEA/D	2214	0.3187	0.1819	0.9874	7.3729
MOGOA	429	0.3235	0.2748	0.9877	6.7399
MOGWO	468	0.7688	0.9483	0.9561	7.2404

The Pearson correlation coefficients ranged from 0.9561 to 0.9926, indicating a strong positive linear correspondence between the power estimates generated by the Deep MTL surrogate and those obtained independently from the pretrained LightGBM model. The MAE values ranged from 0.2650 to 0.7688 kW, while the corresponding MSE values ranged from 0.1462 to 0.9483 kW^2.

The high correlation coefficients indicate that the overall variation in predicted power across the optimized operating states was broadly consistent between the two independently trained predictive models. The magnitude of the absolute errors, however, varied across the optimization algorithms, indicating that agreement was not identical throughout the entire explored solution space.

This cross-model analysis is particularly relevant because the validation points are not randomly sampled observations from the original SCADA dataset. Instead, they are operating states generated by the optimization algorithms through the Deep MTL surrogate. Consequently, the analysis evaluates whether the power patterns identified by the optimization model remain reasonably consistent when the same candidate states are evaluated using an independently trained tree-based model.

The results therefore provide an independent consistency check on the optimization outputs without requiring an exact measured SCADA counterpart for every optimized operating state. The relatively high Pearson correlations across all six optimization outputs indicate that the predicted power patterns are not exclusively determined by the particular Deep MTL surrogate used during optimization.

At the same time, this analysis should not be interpreted as direct experimental validation of the optimized operating points. The LightGBM model is itself a learned predictive model rather than a physical measurement system. Accordingly, the results provide **cross-model predictive validation**, not measurement-based verification of the actual turbine response.

The maximum predicted power values across the retained solution sets ranged from 6.7399 to 7.3729 kW. These values are reported together with the validation metrics to characterize the operating range over which the cross-model consistency was evaluated. As noted in Section 5.4, these values are predictions generated by the computational surrogate framework and should not be interpreted as direct measurements of physical turbine output.

Taken together, the independent validation results provide additional evidence that the optimization-generated power patterns are not solely dependent on a single predictive architecture. The consistency between the Deep MTL and LightGBM predictions supports the use of the optimized solution sets as model-based candidate operating policies, while the distinction between model-based validation and physical measurement validation is retained.

5.7. Summary of Results

The results demonstrate the behavior of the proposed predictive and optimization framework across the successive experimental stages. Under nominal conditions, XGBoost and LightGBM achieved very high regression accuracy, while the Deep MTL architecture maintained a high coefficient of determination while simultaneously providing multiclass operating-status predictions. The classification analysis further showed that the high overall and weighted scores were strongly influenced by the dominant classes, emphasizing the importance of macro-averaged metrics for the imbalanced operating-status task.

Under perturbed input conditions, the robustness-oriented models retained high regression performance, with R^2 values of 0.9990, 0.9991, and 0.9957 for Robust XGBoost, Robust LightGBM, and Robust Deep MTL, respectively. The classification results likewise retained high weighted performance, while the macro-F1 values continued to reveal the difficulty associated with minority operating states.

The multi-objective optimization stage generated Pareto solution sets with different numbers of candidate and retained solutions across the six optimization algorithms. The post-optimization wind-speed filtering step reduced the initial candidate sets to the valid solution sets reported in Table 5.4. The resulting Pareto fronts exhibited different spacing and objective-space extension

characteristics, while all six algorithms generated feasible solutions across the evaluated operating range.

The extracted Safe, Knee-Point, and Max-Power policies further demonstrated the operational meaning of the Pareto trade-off. Increasing predicted power was generally associated with different model-derived risk levels and corresponding changes in the operating variables, allowing the optimization framework to represent multiple feasible operating choices rather than a single predetermined operating point.

Finally, the independent LightGBM validation produced Pearson correlation coefficients between 0.9561 and 0.9926 across the six optimization outputs. This cross-model agreement provides an additional consistency check for the predicted power patterns generated by the Deep MTL surrogate, while not being interpreted as direct physical validation.

Collectively, these findings support the computational validity of the proposed data-driven framework for predictive modeling, robustness assessment, and multi-objective generation of candidate operating policies. The limitations associated with model-based extrapolation, class imbalance, and the absence of direct experimental validation of every optimized operating state are considered in the subsequent discussion.

6. Discussion

6.1. Scientific Findings

This study presents an integrated data-driven framework for wind turbine operational decision support that combines power prediction, operational-status classification, perturbation-based robust learning, and multi-objective optimization. Rather than introducing a new foundational learning architecture or optimization algorithm, the main contribution lies in bringing these components together within a common computational workflow and evaluating their complementary roles in identifying candidate operating policies.

The framework addresses two related but distinct aspects of operational decision-making. First, the prediction component estimates continuous power output while simultaneously classifying the turbine's operational status. Second, the optimization component uses the resulting surrogate predictions to explore trade-offs between expected power generation and uncertainty in the predicted operational-status distribution. This separation allows the study to examine not only predictive performance but also how data-driven models can support the exploration of alternative operating strategies.

The experimental results highlight the complementary roles of the evaluated learning models. Under nominal conditions, LightGBM achieved an RMSE of 0.0403 and an (R^2) of 0.9996 for power prediction. Under the adopted robustness-oriented training and evaluation settings, LightGBM, XGBoost, and the deep multi-task learning model achieved (R^2) values of 0.9991, 0.9990, and 0.9957, respectively. These results indicate that high predictive performance was

retained across the evaluated models under the specified experimental conditions. However, the differences between their results also show why nominal predictive accuracy and performance under perturbations should be assessed separately.

A further contribution is the systematic evaluation of six multi-objective optimization algorithms—NSGA-II, MOPSO, MOEA/D, hybrid MOGWO, enhanced MOWOA, and enhanced MOGOA—under a common objective formulation and a shared set of operational-variable constraints. This design enables a comparison of the candidate Pareto solutions produced by different search strategies without conflating algorithmic differences with changes in the underlying optimization problem. The resulting solution sets support the examination of alternative trade-offs rather than prescribing a single universally optimal operating point.

Overall, the findings support the feasibility of integrating predictive learning and multi-objective search into a data-driven operational decision-support workflow. The contribution is therefore methodological and integrative: the framework connects joint prediction, perturbation-based robustness, uncertainty-aware optimization, and policy-oriented interpretation in a single experimental setting.

6.2. Prediction vs Optimization

The high predictive accuracy obtained by the learning models provides a useful basis for surrogate-assisted optimization, but it should not be interpreted as direct evidence that the resulting operating points are globally optimal or physically achievable. Prediction and optimization address different questions. Predictive evaluation measures how closely a model reproduces observed targets, whereas optimization searches the model's estimated objective space for solutions that satisfy the encoded constraints. Consequently, an optimizer may exploit local inaccuracies in a surrogate model, particularly when searching for solutions near the boundaries of the explored operating space.

To examine the consistency of the optimized power estimates, the initial Pareto solution sets were evaluated using a separately trained LightGBM regression model and the deep multi-task learning model. Across the evaluated solution sets, the Pearson correlation coefficients ranged from 0.9561 to 0.9926, while the mean absolute errors ranged from 0.2650 to 0.7688 kW. These results indicate a strong positive association between the power estimates produced by the two models across the evaluated candidate solutions.

Nevertheless, correlation should not be interpreted as exact agreement. Two models can exhibit a high correlation while retaining systematic differences in their predictions. Moreover, because both models are data-driven, their agreement does not establish that either prediction is physically accurate. The cross-model assessment therefore provides an additional consistency check on the optimization results, rather than independent physical validation or evidence of actual turbine performance.

The order of evaluation is also important. Cross-model consistency was assessed on the initial Pareto solution sets before the subsequent operational wind-speed filtering used to identify candidate policies. The reported statistics therefore characterize the initial optimization outputs and should not be interpreted as measurements of agreement exclusively for the final filtered policy set. This distinction is necessary for a transparent interpretation of the evaluation procedure.

Taken together, the results suggest that the optimization framework can identify candidate solutions whose predicted power values are broadly consistent across the two evaluated models. Establishing their actual operational value, however, requires additional validation against measured turbine behavior and a more comprehensive representation of physical feasibility.

6.3. Robust Learning Under Input Perturbations

Operational data-driven models may encounter input variations that differ from the conditions represented in their training data. To investigate model behavior under selected perturbations, the framework incorporates Gaussian input noise and random feature dropout during robustness-oriented training. These interventions are intended to reduce sensitivity to some input variations and to examine whether useful predictive performance can be retained under the specified experimental conditions.

The results show that the evaluated robust models maintained high power-prediction (R^2) values, ranging from 0.9957 to 0.9991. Although these values indicate strong predictive performance in the experiments conducted, they do not imply that robustness is independent of the perturbation type, magnitude, or operating conditions. In particular, the use of synthetic input perturbations does not establish resilience to every real-world sensor fault, sustained measurement bias, communication failure, or distribution shift.

The results should therefore be interpreted as evidence of performance under the adopted perturbation-based evaluation procedure, rather than as a general guarantee of fault tolerance. Further assessment using documented real sensor faults, temporally distinct operating periods, and data from additional turbines would be necessary to determine how well the learned models generalize to practical monitoring conditions.

6.4. Physical Constraints and Interpretation of Operational Feasibility

The optimization problem incorporates bounds on the decision variables, wind-speed-dependent rotor-speed conditions, pitch-angle behavior, and the gearbox relationship

$$\omega_g = N_{\text{gear}}\omega_r, \qquad N_{\text{gear}} = 11.5.$$

These constraints restrict the search to operating-variable combinations that satisfy the encoded conditions. However, satisfying these constraints does not by itself guarantee that the surrogate-

predicted power is physically feasible. In particular, the optimization formulation does not explicitly enforce the turbine's rated power of 6 kW as an upper bound on predicted output.

Accordingly, some optimized candidates produce raw surrogate power predictions above the rated capacity, reaching approximately 7.37 kW. These values have been retained as model outputs rather than clipped to the rated-power limit. This preserves the distinction between the optimization model's predictions and the turbine's specified operating capacity, while making an important limitation of the current formulation explicit. Predictions above rated power should not be interpreted as evidence that the Aventa AV-7 can deliver that level of actual power under the corresponding conditions.

This issue also illustrates a broader consideration in surrogate-based optimization: constraints on decision variables do not necessarily constrain every quantity predicted by a learned model. A model can produce an output outside a physically valid range even when its input variables satisfy the implemented constraints. Future formulations should therefore incorporate an explicit rated-power ceiling or a validated operating envelope, alongside any additional physical relationships needed to improve feasibility.

Consequently, the Pareto solutions in this study are best understood as candidate solutions within the encoded computational problem. Their use as operational recommendations would require additional feasibility checks and validation against measured turbine responses.

6.5. Interpretation of the Candidate Operating Policies

The Pareto solutions provide a basis for examining different trade-offs between predicted power generation and the uncertainty measure used in the optimization. To facilitate interpretation, three representative policies are considered: Safe, Knee-Point, and Max-Power. These policies represent different preferences within the objective space and are intended to make the trade-offs among candidate solutions more interpretable.

The Safe policy prioritizes minimizing the entropy of the predicted operational-status class probabilities. The Knee-Point policy represents a compromise between the two objectives, while the Max-Power policy prioritizes predicted power generation. Their distinction is therefore based on their relative positions and objective values within the evaluated solution space, rather than on an assumption that one policy is universally superior.

The term *Safe* requires particular care. In this framework, the second objective is Shannon entropy calculated from the predicted StatusAnlage class-probability distribution. Entropy measures the dispersion of that distribution and does not directly quantify the probability of failure, the severity of a hazardous event, or compliance with a certified safety criterion. A lower entropy value indicates a more concentrated predicted class distribution, but it does not necessarily indicate a safer physical operating condition. The Safe policy should thus be interpreted as an entropy-minimizing candidate policy, not as a validated safety guarantee.

Likewise, the Knee-Point policy should not automatically be considered the best operational choice for every application. Its practical suitability depends on the relative importance assigned to power generation and the entropy objective, as well as on the physical feasibility of the selected operating point. The Max-Power policy offers a different preference by emphasizing the model's predicted power output, but its predictions remain subject to surrogate uncertainty and the limitations of the encoded constraints.

These policies are therefore decision-support options rather than ready-to-deploy control commands. Their practical use would require a decision-maker to consider operating conditions, engineering requirements, physical feasibility, and the consequences of prediction errors before selecting an operating point.

6.6. Comparison with Previous Work

Previous research has investigated data-driven wind turbine performance prediction, condition monitoring, and optimization from complementary perspectives. Studies by Maldonado-Correa et al. (2020) and Pandit et al. (2023) provide context for data-driven approaches to wind energy performance analysis and operational assessment. The present study builds on this broader direction by linking predictive modeling to a multi-objective search procedure intended to support the interpretation of alternative operating decisions.

The use of multi-task learning is consistent with the general principle that related predictive tasks can benefit from a shared representation, as discussed by Caruana (1997). In the present framework, power regression and operational-status classification are addressed jointly, allowing continuous performance estimation and status inference to be incorporated into the same learning component. This integration is relevant to the subsequent optimization stage because the two outputs represent complementary aspects of the modeled operating condition. Nevertheless, the results do not establish that multi-task learning is universally superior to separate models; such a conclusion would require a broader comparative evaluation under consistent experimental settings.

Recent work has also explored learning-based and optimization-oriented methods for energy systems and industrial applications. Relevant examples include Ding et al. (2024), Li and Shen (2022), Kandemir et al. (2024), Hu et al. (2025), and Leon-Medina et al. (2025). In relation to this line of research, the present study emphasizes the combination of perturbation-based robust learning with a common multi-objective formulation and the comparative evaluation of six optimization algorithms. The purpose of this design is to examine the trade-offs identified by different search strategies under consistent modeling assumptions, rather than to claim that any one algorithm is inherently optimal for all operating conditions.

The uncertainty-oriented component also distinguishes the present formulation from approaches that optimize performance alone. Studies by Meng et al. (2026) and Daareyni et al. (2026) provide relevant context for research involving uncertainty-aware or multi-objective decision-making, while Ala et al. (2023), Lei et al. (2025), and Daaji et al. (2025) represent related work on

optimization and intelligent methods in engineering applications. In the present study, Shannon entropy is calculated from the predicted operational-status probabilities and used as an objective alongside predicted power. This provides a tractable way to explore the relationship between expected performance and the concentration of model-estimated status probabilities. It should not, however, be equated with a calibrated physical-risk measure.

A further distinction is the cross-model consistency assessment of the initial Pareto solutions. By comparing power predictions from LightGBM and the deep multi-task model, the study examines whether the optimized candidates receive broadly consistent estimates from two independently trained model architectures. This adds a model-consistency perspective to the evaluation, while leaving physical validation as a separate and necessary step.

The main distinction of the proposed framework is thus its integration of joint prediction, perturbation-based robustness, entropy-aware multi-objective optimization, and policy-oriented interpretation within a single data-driven workflow. Its contribution lies in the combined methodology and the empirical evaluation of its components on industrial SCADA data, rather than in the invention of a new foundational learning or metaheuristic algorithm. This positioning defines the scope of the findings and provides a basis for further evaluation on broader datasets and under stronger physical validation requirements.

6.7. Limitations and Future Work

Several limitations define the scope of the present findings and suggest directions for further research.

First, the evaluation uses industrial SCADA data from a single commercial Aventa AV-7 wind turbine. Although the results demonstrate the feasibility of the proposed workflow in this setting, they do not establish generalization across turbine designs, sites, environmental conditions, or maintenance regimes. Future studies should evaluate the framework on multiple turbines and temporally distinct datasets to examine its transferability and stability under changing operating conditions.

Second, the robustness assessment relies on specified input perturbations, including Gaussian noise and random feature dropout. These experiments provide evidence about model behavior under the adopted conditions but do not fully represent the range of real sensor faults and operational disturbances. Future work should incorporate documented sensor anomalies, realistic fault scenarios, and distribution-shift evaluation.

Third, the entropy objective is derived from the predicted operational-status class probabilities and has not been established as a calibrated measure of physical risk or safety. Further research should investigate probability calibration and compare entropy with operationally meaningful risk indicators, where suitable ground-truth evidence is available. This would help determine whether the uncertainty objective can be linked more directly to engineering decision criteria.

Fourth, the optimization formulation does not explicitly impose the 6 kW rated-power ceiling on surrogate predictions. The resulting outputs above rated capacity illustrate the need to strengthen the connection between the learned surrogate and physical feasibility. Future work should incorporate an explicit power ceiling or a validated power envelope and assess additional physical constraints against measured turbine behavior.

Finally, the cross-model consistency assessment is performed on the initial Pareto solution sets before the operational wind-speed filtering used to derive the representative policies. Further evaluation should examine the final filtered solutions and, where possible, compare candidate operating points with measured responses or validated simulation results. Such evaluation would help distinguish agreement between data-driven models from actual physical validity.

Addressing these limitations would support a more comprehensive assessment of generalization, uncertainty interpretation, and operational feasibility. In particular, combining multi-turbine validation, calibrated uncertainty estimates, and physically constrained optimization would provide a stronger basis for progressing from computational decision support toward practical operational evaluation.

7. Conclusion

This study developed and evaluated an integrated data-driven framework for wind turbine operational decision support, combining power prediction, operational-status classification, perturbation-based robust learning, and multi-objective optimization. Using industrial SCADA data from an Aventa AV-7 wind turbine, the study examined how predictive models and alternative optimization strategies can be combined to identify candidate operating policies under a shared objective formulation.

Under nominal conditions, LightGBM achieved an RMSE of 0.0403 and an (R^2) of 0.9996 for power prediction. Under the adopted robustness-oriented settings, the evaluated LightGBM, XGBoost, and deep multi-task learning models achieved (R^2) values of 0.9991, 0.9990, and 0.9957, respectively. These results demonstrate strong predictive performance within the evaluated dataset and experimental conditions, while not establishing universal robustness or generalization to unobserved operating environments.

The optimization component compared six multi-objective algorithms using predicted power and the Shannon entropy of operational-status class probabilities as objectives. The resulting Pareto solutions enabled the interpretation of three representative policies: Safe, Knee-Point, and Max-Power. These policies illustrate alternative trade-offs between predicted performance and the concentration of predicted status probabilities. The Safe policy, in particular, represents entropy minimization rather than a validated guarantee of physical safety.

Cross-model evaluation of the initial Pareto solution sets yielded Pearson correlation coefficients of 0.9561–0.9926 and mean absolute errors of 0.2650–0.7688 kW between LightGBM and deep

multi-task model power predictions. This provides evidence of consistency between the evaluated models, but it does not constitute independent physical validation. Similarly, the encoded operating-variable constraints do not fully ensure physical feasibility, as illustrated by surrogate predictions exceeding the turbine's 6 kW rated capacity.

The principal contribution of this work is the integration and empirical evaluation of predictive learning, perturbation-based robustness, entropy-aware multi-objective optimization, and policy-oriented interpretation within a unified data-driven workflow. The framework offers a computational basis for exploring alternative operating decisions, while clearly distinguishing model-based candidate solutions from verified operational outcomes.

Future research should extend the evaluation to multiple turbines and operating conditions, investigate realistic sensor-fault scenarios, calibrate the uncertainty objective against meaningful operational indicators, and enforce physically validated power limits. These developments would strengthen the framework's generalizability and provide a more rigorous basis for assessing its practical value in wind turbine operational decision support.

Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Data Availability

The SCADA dataset used in this study is publicly available on Zenodo at https://doi.org/10.5281/zenodo.15700928 (Barber et al., 2025).

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Declaration of Generative AI and AI-assisted Technologies in the Manuscript Preparation Process

During the preparation of this manuscript, ChatGPT (OpenAI) was occasionally used to assist with code review and debugging by identifying potential implementation issues and suggesting corrections, as well as with language refinement and manuscript organization. The authors evaluated the suggestions, applied relevant corrections, and reviewed and revised the relevant outputs. The authors retain full responsibility for the final code, computational results, manuscript, and its contents.

References

Ala, A., Mahmoudi, A., Mirjalili, S., Simic, V., & Pamucar, D. (2023). Evaluating the performance of various algorithms for wind energy optimization: A hybrid decision-making model. *Expert Systems with Applications, 221*, Article 119731. https://doi.org/10.1016/j.eswa.2023.119731

Amin, U., Kim, D., Ahmed, F. N., Ahmad, G., & Hossain, M. J. (2025). Digital twins for smart asset management in the energy industry: State-of-the-art. *Expert Systems with Applications, 289*, Article 128358. https://doi.org/10.1016/j.eswa.2025.128358

Barber, S., Hammer, F., Hilfiker, L., & Marykovskiy, Y. (2025). *Aventa AV-7 (6kW) IET-OST Research Wind Turbine SCADA* [Dataset]. Zenodo. https://doi.org/10.5281/zenodo.15700928

Caruana, R. (1997). Multitask learning. *Machine Learning, 28*(1), 41–75. https://doi.org/10.1023/A:1007379606734

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785–794). https://doi.org/10.1145/2939672.2939785

Coello Coello, C. A., & Lechuga, M. S. (2002). MOPSO: A proposal for multiple objective particle swarm optimization. In *Proceedings of the 2002 Congress on Evolutionary Computation (CEC 2002)* (pp. 1051–1056). https://doi.org/10.1109/CEC.2002.1004388

Daaji, M., Benatia, M.-A., Ouni, A., & Gammoudi, M. M. (2025). Predicting wind turbines faults using multi-objective genetic programming. *Expert Systems with Applications, 281*, Article 127487. https://doi.org/10.1016/j.eswa.2025.127487

Daareyni, A., Ylä-Autio, A., Martikkala, A., Mokhtarian, H., & Flores Ituarte, I. (2026). Combining data-driven and model-driven methods to support digital-twin engineering design: A case study for the (re)design optimization of a quadcopter. *Expert Systems with Applications, 296*, Article 129113. https://doi.org/10.1016/j.eswa.2025.129113

Deb, K., & Agrawal, R. B. (1995). Simulated binary crossover for continuous search space. *Complex Systems, 9*, 115–148.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation, 6*(2), 182–197. https://doi.org/10.1109/4235.996017

Ding, J., Deng, L., Li, Q., Gu, X., & Tang, B. (2024). Joint condition monitoring framework of wind turbines based on multi-task learning with poor-quality data. *ISA Transactions, 146*, 221–235. https://doi.org/10.1016/j.isatra.2024.01.008

Dubey, A. K., Kumar, A., Segovia Ramírez, I., & García Márquez, F. P. (2026). Machine learning and hybrid intelligence for wind energy optimization: A comprehensive state-of-the-art review. *Expert Systems with Applications, 296*, Article 128926. https://doi.org/10.1016/j.eswa.2025.128926

He, H., & Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering, 21*(9), 1263–1284. https://doi.org/10.1109/TKDE.2008.239

Hu, W., Fang, J., Zhang, Y., Liu, Z., Verma, A. S., Liu, H., Cong, F., & Tan, J. (2025). Digital twin of wind turbine surface damage detection based on deep learning-aided drone inspection. *Renewable Energy, 241*, Article 122332. https://doi.org/10.1016/j.renene.2024.122332

Kandemir, E., Hasan, A., Kvamsdal, T., & Alaliyat, S. A. (2024). Predictive digital twin for wind energy systems: A literature review. *Energy Informatics, 7*, Article 68. https://doi.org/10.1186/s42162-024-00373-9

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems* (Vol. 30, pp. 3146–3154).

Lei, B., Tang, H., Su, Y., Ru, Y., & Fei, S. (2025). Fuzzy adaptive power optimization control of wind turbine with improved whale optimization algorithm and kernel extreme learning machine. *Expert Systems with Applications, 272*, Article 126750. https://doi.org/10.1016/j.eswa.2025.126750

Leon-Medina, J. X., Tibaduiza, D. A., Parés, N., & Pozo, F. (2025). Digital twin technology in wind turbine components: A review. *Intelligent Systems with Applications, 26*, Article 200535. https://doi.org/10.1016/j.iswa.2025.200535

Li, Y., & Shen, X. (2022). A novel wind speed-sensing methodology for wind turbines based on digital twin technology. *IEEE Transactions on Instrumentation and Measurement, 71*, Article 2503213. https://doi.org/10.1109/TIM.2021.3139698

Liu, H., Sun, W., Bao, S., Xiao, L., & Jiang, L. (2024). Research on key technology of wind turbine drive train fault diagnosis system based on digital twin. *Applied Sciences, 14*(14), Article 5991. https://doi.org/10.3390/app14145991

Maldonado-Correa, J., Martín-Martínez, S., & Artigao, E. (2020). Using SCADA data for wind turbine condition monitoring: A systematic literature review. *Energies, 13*(12), Article 3132. https://doi.org/10.3390/en13123132

Mantegna, R. N. (1994). Fast, accurate algorithm for numerical simulation of Lévy stable stochastic processes. *Physical Review E, 49*(5), 4677–4683. https://doi.org/10.1103/PhysRevE.49.4677

Meng, Q., Hu, T., Zhou, S., & Li, J. (2026). Digital twin-driven dynamic prediction of operation status and process optimization for flexible production line. *Expert Systems with Applications, 308*, Article 131140. https://doi.org/10.1016/j.eswa.2026.131140

Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. *Advances in Engineering Software, 95*, 51–67. https://doi.org/10.1016/j.advengsoft.2016.01.008

Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software, 69*, 46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007

Mirjalili, S., Saremi, S., Mirjalili, S. M., & Coelho, L. dos S. (2016). Multi-objective grey wolf optimizer: A novel algorithm for multi-criterion optimization. *Expert Systems with Applications, 47*, 106–119. https://doi.org/10.1016/j.eswa.2015.10.039

Olatunji, O. O., Adedeji, P. A., Madushele, N., & Jen, T. C. (2021). Overview of digital twin technology in wind turbine fault diagnosis and condition monitoring. In *2021 IEEE 12th International Conference on Mechanical and Intelligent Manufacturing Technologies (ICMIMT)* (pp. 201–207). IEEE. https://doi.org/10.1109/ICMIMT52186.2021.9476186

Pandit, R., Astolfi, D., Hong, J., Infield, D., & Santos, M. (2023). SCADA data for wind turbine data-driven condition/performance monitoring: A review on state-of-art, challenges and future trends. *Wind Engineering, 47*(2), 422–441. https://doi.org/10.1177/0309524X221124031

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research, 12*(85), 2825–2830.

Qiao, L., Zhang, Y., Wang, Q., Li, D., & Peng, S. (2025). Fault diagnosis for wind turbine generators based on model-agnostic meta-learning: A few-shot learning method. *Expert Systems with Applications, 267*, Article 126171. https://doi.org/10.1016/j.eswa.2024.126171

Saremi, S., Mirjalili, S., & Lewis, A. (2017). Grasshopper optimisation algorithm: Theory and application. *Advances in Engineering Software, 105*, 30–47. https://doi.org/10.1016/j.advengsoft.2017.01.004

Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal, 27*(3), 379–423. https://doi.org/10.1002/j.1538-7305.1948.tb01338.x

Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management, 45*(4), 427–437. https://doi.org/10.1016/j.ipm.2009.03.002

Storn, R., & Price, K. (1997). Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization, 11*, 341–359. https://doi.org/10.1023/A:1008202821328

Zhang, Q., & Li, H. (2007). MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation, 11*(6), 712–731. https://doi.org/10.1109/TEVC.2007.892759

Zhang, Y., Qiao, L., & Zhao, M. (2024). Fault diagnosis for wind turbine generators using normal behavior model based on multi-task learning. *IEEE Transactions on Automation Science and Engineering, 21*(2), 1258–1270. https://doi.org/10.1109/TASE.2023.3293931

Zitzler, E., & Thiele, L. (1999). Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation, 3*(4), 257–271. https://doi.org/10.1109/4235.797969