List of Revisions: Manuscript "wes—2025-289"
Title: "Classification of Leading Edge Erosion Severity via Machine Learning Surrogate Models,"

The authors are grateful to the reviewers for their thoughtful comments which helped us to substantially improve the manuscript. Our responses to each comment are listed below and are reflected in the revised manuscript.

1 Reviewer 1

Revisions in response to comments made by Referee #1:

Reviewer #1 Major Comments:

1. **Comment:** The overall contribution is not yet clearly positioned with respect to the existing literature on surrogate modeling and wind turbine condition monitoring. Gaussian-process surrogates, sensitivity analysis, and random forest classifiers are all well-established techniques. The manuscript should clarify more explicitly what methodological advance is introduced beyond applying these tools to a specific erosion-monitoring scenario.

   Response: In Section 1 line 81, we clarify the overall contribution as follows:

   > Our primary contribution in this paper is the introduction of a framework to reduce the computational cost of training a SHM monitoring algorithm. Specifically, we present the first application of the PPzGP emulation methodology to wind turbine modeling and the classification of LEE. While this work involves simplifications to the physics and operating conditions including assuming steady-state wind, the approach could be extended to more physically realistic scenarios, including turbulent inflow. We compare the performance of LEE classifiers trained on datasets generated from the emulator to those obtained using OpenFAST simulations.

2. **Comment:** The claim of novelty regarding the PPzGP surrogate is not sufficiently demonstrated. While combining parallel partial emulation and range-censored Gaussian processes is technically interesting, the manuscript does not clearly show why this combination enables capabilities that standard GP surrogates would not provide for this problem.

   Response:

   In Section 1 line 71 we now highlight the specific advantages of our approach to WT emulation, namely we state that:

   > ... the existing GP literature primarily focuses on predicting scalar or low-dimensional quantities of interest. Emulation for SHM requires physically consistent, higher-dimensional outputs for downstream tasks such as damage classification. With the standard approach, emulating a WT simulator that generates higher-dimensional vector-valued output from multiple sensor channels requires separately fitting and storing multiple standard scalar GPs, one for each output dimension. In this paper we use partial parallel emulation to significantly accelerate the prediction of vector-valued outputs [9]. In addition, for outputs with constraints, such as generator power, a standard GP is unrealistic because its output is unbounded. Spiller et al. [17] developed the zero-censored GP emulator to enforce range-limited outputs. We show that the combination of parallel partial and zero-censored emulation (PPzGP) [9, 17] improves the fidelity and computational efficiency of higher dimensional WT emulation models.

Additionally, we added Section 4.3.1, in which we compare scalar and parallel emulation strategies to highlight the reduced computational effort and statistical benefits from using zero-censored and parallel-partial emulation [17, 10].

3. **Comment:** The erosion model used to generate the data is highly simplified. Blade erosion is represented through a parametric scaling of lift and drag coefficients across six blade regions. While this may be suitable for a proof-of-concept study, the manuscript should discuss the limitations of this representation and justify why it captures the key aerodynamic effects of real leading-edge erosion.

   Response: The description of the erosion model in Section 2.2.1 has been rewritten. We now begin with an overview of the model, which follows a similar model used in [7] (justified by wind tunnel experiments). In this section we justify our heuristic stochastic model for the trend and variability of erosion as a random function of position along the blade. We then describe three limitations of the model. The section ends with the following justification for why our simplified model captures the key aerodynamic effects of leading edge erosion in line 203:

   > However, our simplified stochastic parameterization is not intended to model the complete time evolution of LEE. Rather, it is designed to generate physically-motivated families of aerodynamic perturbations for evaluating whether emulator-generated data can support erosion-severity classification. For this purpose, the model preserves several important qualitative features. Namely, (1) erosion damage increases monotonically with class severity, (2) erosion accumulation is weighted toward the tip, and (3) the induced aerodynamic changes reduce lift and increase drag within reported ranges.

4. **Comment:** The erosion process is modeled as discrete severity classes rather than a continuous degradation process. In reality erosion evolves gradually and spatially across the blade surface. The use of five artificial classes may simplify the classification task and should be justified more clearly.

   Response: The simplified erosion model we use is not intended to provide a physically accurate model of erosion either over the course of time or along the blade. Instead, its purpose is described in the quote at the end of our response to the previous comment, line 206. Furthermore, in the discussion of the limitations of the erosion model at the end of Section 2.2.1 line 202, we state that

   > ... our use of five erosion classes necessarily coarsens a gradual physical process. This representation does not resolve subtle differences between early-stage erosion states or the full diversity of possible damage patterns along the blade.

   Additionally, in Section 2.2.1 in the paragraph beginning line 157, we state

   > As in [7], we quantify the severity of erosion using a finite ordered set of damage classes, which encode the overall erosion level of the blade. These classes provide a practical link between observed surface degradation, aerodynamic performance loss, and potential remedial actions [11, 15, 13].

5. **Comment:** The classification problem may be artificially easy because the erosion perturbations are directly embedded in the aerodynamic coefficients and the classifier is trained on outputs that are strongly linked to those coefficients (e.g., lift and drag sensor statistics). This raises the possibility that the model is learning the synthetic perturbation rather than identifying erosion signatures that would be observable in practice.

   Response:

   We thank the reviewer for this insightful comment. In our study, lift and drag sensors were placed only at the tip of the blade. Dynamic lift sensors could soon be commercially available since as we state in line 214 in Section 2.2.2:

> Although lift and drag measurements are not commonly available, new sensors under development [3] measure aerodynamic pressure over an aero-foil section to calculate the dynamic lift coefficient, which can be used to detect changes in airflow patterns due to surface roughness [1].

However, to account for the situation where lift and drag sensor data is unavailable, in Section 4.4.2, line 585, we state

> To further assess the dependence of the classifier on direct aerodynamic sensing, we repeat the classification study without the lift and drag coefficient channels.

We further state in the paragraph beginning line 603 that:

> In Fig. 12 we show that, after removing the aerodynamic lift and drag channels, classifier performance decreases substantially relative to the full-sensing case shown in Fig. 8, confirming that with less access to aerodynamics-based sensors the classification problem is more difficult. However, the simulator-trained and emulator-trained classifiers have comparable class-wise behavior. Nonetheless, the confusion matrices are dominated by their diagonal entries, indicating that that misclassifications typically occur between adjacent erosion classes rather than between widely separated damage states. The severe erosion class remains identifiable, with both classifiers correctly classifying 73% of $\alpha = 1.00$ samples. Consistent with the confusion matrices, in Table 19 we observe nearly identical balanced accuracy and macro-F1 values for the two classifiers, indicating that the emulator-trained classifier remains statistically comparable to the simulator-trained classifier in the case of no direct aerodynamic sensors.

6. **Comment:** The strong performance of a relatively simple random forest classifier suggests that the generated dataset may be too clean or too easily separable. In practice, leading-edge erosion detection is known to be challenging due to turbulence, operational variability, sensor noise, and confounding effects. The simulations appear to lack these disturbances, which may make the classification task unrealistically simple.

   Response: In Section 4.4.1, we have included new simulations with sensor noise and added sensor bias. However, a thorough study that includes turbulent wind conditions is beyond the scope of this paper. To partially address the reviewer's concerns we have added the following text to the Future Directions in Section 5 line 613.

   > In this paper we provide a proof-of-concept study for the use of wind-turbine emulators to generate data for SHM. Even though turbulent inflow makes distinguishing between different levels of LEE more difficult, we believe it is possible to extend the proposed modeling framework to more realistic operating conditions, including turbulent flow, controller transitions, and yaw adjustments. The OpenFAST turbulence model is parametrized by moments such as average wind speed and the mean and covariance of turbulence intensity, which could be used as inputs to an emulator. More complex operating conditions would require much longer OpenFAST simulations (on the order of tens of minutes rather than seconds) to extract reliable statistical features from time series information. In steady-state simulations, we achieved good results with a comparatively small dataset. However, in order to fit a GP on complex simulations, more simulations will likely be required to achieve high emulator accuracy, which may require different emulator fitting strategies [6]. However, in the case of long simulation runs, the speedup from using an emulator would be even greater than in the current study.

7. **Comment:** The simulations assume uniform wind conditions rather than turbulent inflow. Since turbulence strongly influences turbine loads and vibration signals, the use of uniform wind fields likely underestimates the variability present in real monitoring data. Including turbulent wind realizations would significantly improve realism.

Response: See the response to Reviewer 1, Comment 6.

8. **Comment:** The operational variability of the turbine is limited. Real turbines experience controller transitions, yaw adjustments, and varying operating regimes that influence measured signals. The current simulation setup may not capture these effects.

Response: See the response to Reviewer 1, Comment 6.

9. **Comment:** The sensor configuration used in the study may not reflect practical monitoring systems. In particular, lift and drag pressure sensors are rarely available in operational wind turbines. The manuscript should discuss the feasibility of the assumed sensing setup or consider signals more commonly available in SCADA or structural monitoring systems.

Response: In Section 2.2.2 line 214, we add the following discussion:

> Although lift and drag measurements are not commonly available, new sensors under development [3] measure aerodynamic pressure over an aero-foil section to calculate the dynamic lift coefficient, which can be used to detect changes in airflow patterns due to surface roughness [1]. Similar studies have used this data to track the progression of leading edge erosion using OpenFAST software [1, 7].

In Section 4.4.2 we have added an experiment were we remove the lift and drag pressure sensors from the potential inputs for damage classification.

10. **Comment:** The feature extraction strategy reduces time-series signals to simple statistical moments (mean, standard deviation, skewness, kurtosis). This discards potentially important information contained in the temporal and spectral structure of the signals. The authors should justify this choice or explore richer feature representations.

Response: In Section 2.2.2, line 225, we state

> Enríquez et al ([8]) also extracted richer features from time series data such as higher order crossings, mobility, and complexity features, but in Section 4.4 in Figure 8 and Figure 9 we show high classification accuracy without additional feature extraction. A plausible reason for this may be that (1) we look at a relatively coarse set of erosion levels that naturally have distinct features because they do not naturally overlap and (2) we are using bulk output statistical features which contain a large amount of information for steady-state wind flow.

Further, in Section 4.4.2, where we consider an experiment without lift and drag features, we considered additional time-series quantities shown in Table 18. We performed feature selection to reduce the number of features to 14 down from a total of 108.

11. **Comment:** The surrogate model is trained on a relatively small number of simulations compared to the dimensionality and nonlinearity of the system. Although the reported prediction errors are moderate, the manuscript should discuss potential model bias and the limits of extrapolation.

Response: [ES] In Section 2.5.1, line 293, we have added the text

> The GP predictive mean given in Eq (3) is a best linear unbiased predictor [14]. Also note that the rule of thumb for the minimum size of a GP design is $10\times$ the number of input dimensions ($10 \times p$) [5].

12. **Comment:** The reported classification improvement between simulator-trained and surrogate-trained models (83% vs 87%) is relatively modest. It would be helpful to provide statistical analysis or repeated experiments to assess whether this improvement is significant.

    Response: We use a paired repeated k-fold cross-validation protocol (10 repeats over 5 folds) to compare simulator-trained and emulator-trained classifiers. For each train/test split, both classifiers are evaluated on the same held-out simulated test fold, and paired confidence intervals are computed from the fold-wise score differences. This enabled us to generate the findings in Table 13 and Table 14 in Section 4.4 which support a small but statistically significant improvement in accuracy for the emulator trained model for all but the clean class. We added the following to this discussion at line 512:

    > In Table 12 we show the paired differences between the emulator-trained and simulator-trained classifier scores for each emulated training-set size. Positive values indicate that the emulator-trained classifier performs better on the same held-out simulated test sets, while confidence intervals that do not contain zero indicate statistically meaningful differences across repeated splits. The results show a clear dependence on emulated training-set size. With only 500 emulated samples, the emulator-trained classifier performs worse than the simulator-trained classifier across all three metrics. With 1,000 samples, the differences are small, indicating comparable performance. With 5,000, 10,000 and 50,000 samples, the paired differences are positive for the macro-AUC, balanced accuracy, and macro-F1, showing that the larger emulator-generated datasets improve downstream classifier performance. These results suggest that the main benefit of the emulator-based workflow is not that each emulated point exactly reproduces a simulator point, but that the emulator can generate substantially larger training datasets at negligible additional computational cost, improving aggregate classifier performance.

Reviewer #1 Additional Comments:

13. **Comment:** The manuscript frequently refers to digital twins, but the work primarily demonstrates surrogate modeling and classification using simulated data. Since essential digital twin elements such as data assimilation, state estimation, or online updating are not included, the connection to digital twins should be described more cautiously.

    Response:

    In the Discussion Section 5 of the revised manuscript we specify the connection to digital twins as a motivation for emulator development and future work beginning line 623:

    > The main motivation for development of the emulator is to reduce the computational cost of running a digital twin for monitoring leading edge erosion. We envisage that the PPzGP emulator in this paper would serve as the kernel of a probabilistic, graph-based digital twin for tracking damage progression over time [12].

14. **Comment:** The introduction is somewhat lengthy and could be shortened. Several sections summarizing wind-energy background material could be condensed to focus more directly on the methodological contribution.

    Response:

    We have shortened the introduction of the paper to focus more on the SHM/ML context of our framework and results.

15. **Comment:** Some terminology is used interchangeably throughout the manuscript (e.g., emulator, surrogate model). Consistent terminology would improve clarity.

Response: In the literature, the term surrogate model can refer to emulators, neural networks or reduced-order models. In the revised manuscript, we consistently use the term emulator for the surrogate model we developed.

16. **Comment:** Figures illustrating emulator predictions are informative but could be improved in readability, particularly through larger axis labels and clearer legends.

    Response: Figure readability has been improved in the revised paper.

17. **Comment:** The manuscript would benefit from a clearer discussion of the gap between simulation-based validation and deployment on real turbine monitoring data.

    Response: In Section 5.4 line 625 we state:

    > Areas in which this study could be extended include accounting for how real turbine monitoring data is affected by sensor noise, controller behavior, atmospheric variability, and maintenance history. Further, in this work we do not track erosion progression over time. To close the gap would require additional development of the surrogate model to include effects like turbulence, and validation of an improved surrogate model by comparison with field data (SCADA data and blade inspections) [7, 19].

18. **Comment:** Minor typographical issues and formatting inconsistencies appear throughout the manuscript and should be corrected during revision.

    Response: We have improved the typographical consistency. Minor typographical and formatting issues have been searched for and corrected throughout the manuscript, including inconsistent capitalization in figure and table captions, inconsistent hyphenation of compound terms, inconsistent use of mathematical notation and variable formatting, spacing around units and symbols, and minor punctuation and grammar errors.

2 Reviewer 2

Revisions in response to comments made by Referee #2:

Reviewer #2 Comments:

1. **Comment:** The scientific contribution should be stated more precisely. The claimed novelty appears to be the combination of PPzGP emulation with leading-edge erosion classification, rather than a fundamentally new classifier or erosion model. The authors should distinguish clearly between methodological novelty, application novelty, and engineering proof-of-concept. The present text sometimes makes the contribution appear broader than what has actually been demonstrated.

   Response:

   Section 1 has been improved to clearly explain the contribution of the paper, see Reviewer #1 Comment #1. In Section 6, the conclusions have been moderated, focusing separating conclusions of this proof-of-concept study from future directions.

2. **Comment:** The erosion model is highly heuristic and needs stronger justification. The severity classes are generated through a multivariate normal erosion-level model and then translated into lift and drag changes through simple linear scaling. The covariance structure in Eq. (9), the assumed spatial correlation along the blade, and the linear use of the 53 % lift-loss and 500 % drag-increase limits

should be justified with experimental, CFD, or literature-based evidence. If this is only a synthetic benchmark, this limitation should be stated explicitly throughout.

Response:

See the response to Reviewer 1, Comment 3.

In particular, starting in line 161 we justify the linear dependence of the mean erosion as follows.

> In addition, it is reasonable to assume that, to first order, the amount of erosion depends linearly on position along the blade. This erosion is expected to initiate and progress more rapidly in blade regions that are further from the hub [4, 13, 16, 18].

Furthermore, in line 177 we justify the heuristic covariance model as follows.

> This heuristic covariance model imposes stronger correlation between the erosion level in nearby blade regions, while allowing erosion variability to increase both toward the blade tip and with erosion severity. The increase in erosion variability with erosion severity accounts for the fact that severe erosion can represent a wide range of physical states, including coating loss, roughness patches, pits, gouges, and localized defects, whereas mild erosion is comparatively more uniform [13].

Finally, in line 183 we justify the lift-loss and drag-decrease limits as follows.

> ...we model the aerodynamic effect of erosion through a simplified spatial perturbation of the lift and drag polars. This approach is consistent with prior erosion models that interpolate or perturb aerodynamic polars according to local damage severity [13, 2]. Based on reported severe-erosion effects, including lift losses up to 53% [11] and drag increases up to 500% [15]...

3. **Comment:** The manuscript states that regional erosion levels lie in [0, 1], but the multivariate normal distribution in Eq. (8) is not naturally bounded. The authors should explain whether samples are clipped, rejected, transformed, or otherwise constrained. This is important because the erosion labels and the lift/drag perturbations depend directly on these sampled values.

   Response: In Section 2.2.2 line 181 of the revised paper we state that

   > This model takes values sampled from the normal distribution that lie in the interval $[0, 1]$, replacing values that are < 0 by 0 and values > 1 by 1.

4. **Comment:** The simulation setting is considerably simplified. The study uses steady uniform wind files, a fixed nacelle, constant environmental inputs, and ultimately fixes wind shear after sensitivity screening. Turbulence, inflow variability, wave effects, yaw-control behavior, sensor noise, and operational transients are not considered. This is acceptable for an initial proof-of-concept, but the conclusions should not be framed as evidence for operational digital-twin deployment unless these limitations are addressed or explicitly qualified.

   Response:

   See the response to Reviewer 1, Comment 6. Turbulent inflow simulations are beyond the scope of the current study. We frame the conclusions around a promising proof-of-concept study that illustrates key advantages of our emulation framework and discuss future research directions in Section 5.4.

5. **Comment:** The observability of the proposed damage predictors requires more discussion. Several highly ranked inputs are lift and drag coefficient quantities extracted from OpenFAST at a blade node. In a real turbine these are not directly available in the same form and would require pressure instrumentation, calibration, and noise handling. The authors should explain how such measurements

would be obtained in practice, and should test the classifier under realistic sensor noise, bias, missing data, or reduced sensor availability.

Response:

For a discussion of obtaining aerodynamic data from blade sensors, see the response to Reviewer#1 Comment#9. We perform an additional study to investigate the robustness of the framework to added sensor noise and bias in Section 4.4.1. In Section 4.4.2, we describe an experiment in which we removed the lift and drag pressure sensors.

6. **Comment:** The data-partitioning strategy is not sufficiently transparent. The manuscript refers to a primary dataset, a separate emulator-training dataset, 10-fold cross-validation, feature selection with an initial random forest, and emulator-generated data. The exact relationship between these datasets should be described with a table or schematic. It must be clear which simulations are used for Morris screening, feature ranking, emulator training, random-forest training, hyperparameter tuning, and final testing.

   Response: Table 5, which has been added to Section 3, describes the purpose and relationship between each dataset used in this study. The revised table and accompanying text specify which simulations are used for Morris screening, feature ranking, emulator training, random-forest training, hyperparameter tuning, and final testing. In particular, we now explicitly distinguish the primary simulator dataset, the separate emulator-training dataset, and the emulator-generated classifier-training data. We also clarify that final testing is performed only on held-out simulator-generated data and that these test samples are not used during feature selection, emulator fitting, classifier training, or hyperparameter tuning.

7. **Comment:** The feature-selection and hyperparameter-optimization procedures may introduce optimistic bias if they are not nested within the cross-validation loop. The authors should state whether predictor ranking, hyperparameter selection, and model evaluation were performed inside each training fold. If not, the classification results should be recomputed using a fully nested evaluation protocol.

   Response: In Section 3 line 368, we explain the feature-selection and hyperparameter-optimization:

   > In Fig. 2, we show the workflow we used to compare two classifiers for leading-edge erosion, one trained directly on simulated data and the other trained on emulated data. ... In order to eliminate potential bias, we perform feature ranking and selection on an independent dataset in Section 4.2 to rank the importance of the sensor outputs for discriminating between erosion classes and select a shorter list of predictors in order to improve classifier accuracy. In Section 4.3, we use the third dataset to train and test the emulator. ... In Section 4.4, we use the fifth dataset as the ground-truth against which to compare the emulator-trained classifiers' performance. Classifier hyperparameter tuning is done within each of the repeated 5-fold cross-validation sets. Predictions are made on the testing portion of each split, which is not seen by the model during hyper-parameter tuning. The emulator-trained classification models do not train on simulation data.

8. **Comment:** The reported improvement from the emulator-trained classifier is modest and may not be statistically meaningful. Table 9 reports an accuracy increase from 83.00 % to 86.74 %, but the reported standard deviations are 4.55 % and 6.69 %. The authors should provide paired confidence intervals, repeated cross-validation, or an appropriate statistical test. Macro-F1, balanced accuracy, per-class precision and recall, and calibration metrics would also be more informative than accuracy and AUC alone.

   Response:

In Section 4.4, we added Tables 12, 13, and 14 as well as the text lines 512–532 to compare Macro-F1, balanced accuracy, and recall between the emulator-trained and simulator-trained classifiers. We further provide paired confidence intervals computed using 10 repeated 5-fold cross validation studies.

9. **Comment:** The comparison between classifiers is confounded by training-set size. The simulation-trained classifier uses 500 simulated samples, while the emulator-trained classifier uses 5000 emulator-generated samples. The observed gain could be due to more training data rather than the surrogate approach itself. The paper would benefit from learning curves and equal-size comparisons, for example, 500 emulated samples versus 500 simulated samples, and progressively larger emulator-generated datasets.

   Response: We agree that using a larger training set for the emulator-trained classifier than for the simulator-trained classifier indeed improves classification accuracy. However, one of our main points in this work is that once the emulator is trained, obtaining as much training data as one would like is essentially free. Additionally, in Section 4.4 we add a learning-curve study, varying the number of samples from 500 to 50,000 in the emulated dataset. We compare Macro-AUC and Macro-F1 metrics in Figure 7. We compute the paired confidence intervals and display the results in Table 12, see Comment #8 above.

10. **Comment:** There are inconsistencies between the abstract, the results, and the tables. The abstract states that NRMSE values are below 10% and credible-interval coverage exceeds 88%. However, Table 7 shows NRMSE values above 10% for at least two outputs, and Table 6 shows coverage of 80% for root moment mean and tip acceleration standard deviation. These claims should be corrected, and the implications of undercoverage and higher errors should be discussed.

    Response: These inconsistencies have been corrected in Section 4.3.

11. **Comment:** The computational-speed claims need to distinguish single-prediction speed from full-workflow speed. Table 8 indicates that a single PPzGP prediction is much faster than one OpenFAST run, but the full surrogate workflow includes 173 simulations and imputation. The authors should report the total cost needed to generate the 5000-sample classifier-training dataset and compare it with the cost of generating the same dataset directly. Hardware, parallelization, software versions, and wall-clock versus CPU time should be specified. The text currently alternates between about 1000x, 4 orders of magnitude, and other implied speedups.

    Response: Table 11 (old Table 8) has been updated, and text has been addedstarting on line 456 in Section 4.3 to address the reviewer's comments.

    > we distinguish between the cost of a single simulation, training, and generating samples with the GP emulator. All timings are wall-clock times measured on a machine with an Intel(R) Core™ Ultra 7 155H processor (3.80 GHz) and 32 GB RAM using OpenFAST-v3.5.0. Simulator timings were estimated from 25 sequential OpenFAST runs at randomly selected inputs. Emulator timings were estimated from five repeated five-fold splits on 210 data points; prediction timings correspond to generating 10,000 emulator samples and were repeated 25 times.
    >
    > Our OpenFAST simulation requires 28.4 ± 0.3 s, whereas generating 10,000 PPzGP samples requires 0.93 ± 0.01 s, corresponding to approximately 9.3×10^{-5} s per emulator sample. Thus, evaluating the fitted PPzGP is roughly 3×10^5 times faster than running OpenFAST once. This prediction-only speedup does not include the cost of constructing the emulator. The full PPzGP workflow includes the 210 OpenFAST simulations used to train the emulator, zero-censored preprocessing for range-limited outputs, PPzGP fitting, and generation of the emulator-based classifier-training data. Using the measured OpenFAST timing, the simulator portion of this workflow costs approximately 5964 s. Zero-censored preprocessing is the most

expensive part of fitting the PPzGP, but it can be sped up by preparing each output in parallel, approximately 280 s on our laptop. It takes 3.7 s for PPzGP fitting, and 0.93 s to generate 10,000 samples, the full surrogate workflow requires approximately 6.2×10^3 s. In contrast, directly generating 10,000 OpenFAST samples would require approximately 2.84×10^5 s, giving a full-workflow speedup of about 46×. The prediction-only speedup and full-workflow speedup indicate that querying the fitted emulator is extremely fast and the full workflow remains dominated by the initial OpenFAST simulations needed to train the emulator.

12. **Comment:** The sensitivity analysis justifies fixing wind shear only within the simplified scenario tested. Since wind shear may interact with turbulence, yaw, and operating regime in realistic conditions, the authors should avoid generalizing this conclusion. They should also clarify whether the Morris screening was performed on raw time-series outputs or on the statistical moments later used for classification.

    Response: To address these concerns, in Section 4.1 line 389 we state that

    > However, the limited impact of wind shear may be due to our assumption of steady-state wind conditions. Whether wind shear is influential in the turbulent case would have to be analyzed.

    In Section 4.1 line 381 we also state that

    > The elementary effects study was carried out on the mean of the five sensor output time series described in Section 2.2.

13. **Comment:** The severe false-negative cases should be investigated. The text notes that the most serious misclassification occurs when a fully eroded blade is classified as clean. Even if rare, this is critical from a maintenance and risk perspective. The authors should report per-class recall, confusion costs, and whether the classifier can be tuned to reduce dangerous false negatives at the expense of less critical false positives.

    Response: Throughout Section 4.5, we report now report recall to assess if the model is missing critical severe classes, see Tables 14 and 15. In addition, at line 548 we added an experiment addressing confusion cost and avoiding false negatives.

    > In Figure 11 we show the row-normalized confusion matrices obtained using the cost-minimizing thresholds identified in Figure 10. Compared with the default decision rule ($\tau = 0.5$), the tuned classifiers predict severe damage more often. This conservative tuning reduces the rate of severe underprediction for both the simulator-trained and emulator-trained classifiers, although it also increases overpredictions. Figure 11(b) shows a noticeable reduction in recall for the $\alpha = 0.75$ class under the tuned rule compared to Fig. 8(b), while the emulator-trained classifier maintains more consistent recall across the severe classes. Overall, these results show that the emulator-trained classifier can be tuned effectively to reduce high-risk severe false negatives, with a risk-reduction tradeoff comparable to that of the simulator-trained classifier.

14. **Comment:** Several tables and figures need correction or improvement. Table 6 appears to repeat the label "Coefficient of Lift standard deviation" twice (rows 2 and 6). Figure 1 refers to regions described in Table 4, although, if I am not wrong, the region definitions are in Table 3. Figures 7 and 8 are too small for the confusion-matrix values and ROC labels to be read comfortably.

    Response: The tables and figures have been corrected as suggested by the reviewer.

15. **Comment:** The literature review contextualizing this work should be better defined in terms of general SHM/NDT applications to wind turbines (see, e.g., https://doi.org/10.3390/s22041627 and similar ones).

    Response: We have shortened the introduction of the paper to better contextualize our framework and results, namely ML based SHM methods for detecting LEE and emulation in the context of WT modeling.

16. **Comment:** The conclusions should be moderated. The present study establishes a synthetic proof-of-concept for surrogate-assisted classifier training under simplified conditions. It does not yet demonstrate real-time decision-making, field deployment, damage progression tracking, or robust digital-twin operation under realistic sensor and environmental variability. These points can be framed as future work rather than current achievements.

    Response: In the Future Direction section 5 line 623, we moderate the conclusions of the paper as follows.

    > The main motivation for development of the emulator is to reduce the computational cost of running a digital twin for monitoring leading edge erosion. We envisage that the PPzGP emulator in this paper would serve as the kernel of a probabilistic, graph-based digital twin for tracking damage progression over time [12]. Areas in which this study could be extended include accounting for how real turbine monitoring data is affected by sensor noise, controller behavior, atmospheric variability, and maintenance history. Further, in this work we do not track erosion progression over time. To close the gap would require additional development of the surrogate model to include effects like turbulence, and validation of an improved surrogate model by comparison with field data (SCADA data and blade inspections)[7, 19].

17. **Comment:** The English is generally understandable, but the manuscript needs careful proofreading. Examples include "classifers", "leafs" instead of "leaves", inconsistent use of GP/GPs, inconsistent capitalization of OpenFAST module names, and occasional awkward or missing articles. The notation and unit formatting should also be made consistent throughout.

    Response:

    We have improved the revised manuscript via careful reading/editing.

References

[1] I. Abdallah, G. Duthé, S. Barber, and E. Chatzi. Identifying evolving leading edge erosion by tracking clusters of lift coefficients. J. Phys. Conf. Ser., 2265(3):032089, 2022.

[2] I. Abdallah, A. Natarajan, and J. D. Sørensen. Impact of uncertainty in airfoil characteristics on wind turbine extreme loads. Renew. Energ., 75:283–300, 2015.

[3] S. Barber, J. Deparday, Y. Marykovskiy, E. Chatzi, I. Abdallah, G. Duthé, M. Magno, T. Polonelli, R. Fischer, and H. Müller. Development of a wireless, non-intrusive, mems-based pressure and acoustic measurement system for large-scale operating wind turbine blades. Wind Energy Science Discussions, 2022:1–25, 2022.

[4] J. I. Bech, C. B. Hasager, and C. Bak. Extending the life of wind turbine blade leading edges by reducing the tip speed during extreme precipitation events. Wind Energy Science, 3(2):729–748, 2018.

[5] J. O. Berger and L. A. Smith. On the statistical formalism of uncertainty quantification. Annual review of statistics and its application, 6(1):433–460, 2019.

[6] A. C. Clark and C. E. Clark. Employing bayesian quadrature to improve fitting of surrogate models to wind turbine loads. J. Phys. Conf. Ser., 2265(4):042045, 2022.

[7] G. Duthé, I. Abdallah, S. Barber, and E. Chatzi. Modeling and monitoring erosion of the leading edge of wind turbine blades. Energies, 14(21):7262, 2021.

[8] J. Enríquez Zárate, M. d. l. Á. Gómez López, J. A. Carmona Troyo, and L. Trujillo. Analysis and detection of erosion in wind turbine blades. Mathematical and Computational Applications, 27(1):5, 2022.

[9] M. Gu and J. O. Berger. Parallel partial gaussian process emulation for computer models with massive output. Ann. Appl. Stat., pages 1317–1347, 2016.

[10] M. Gu, J. Palomo, and J. O. Berger. Robustgasp: Robust gaussian stochastic process emulation in r. The R Journal, 11(1):112–136, 2019.

[11] W. Han, J. Kim, and B. Kim. Effects of contamination and erosion at the leading edge of blade tip airfoils on the annual energy production of wind turbines. Renew. Energ., 115:817–823, 2018.

[12] M. G. Kapteyn, J. V. Pretorius, and K. E. Willcox. A probabilistic graphical model foundation for enabling predictive digital twins at scale. Nat. Computational Science, 1(5):337–347, 2021.

[13] D. C. Maniaci, H. MacDonald, J. Paquette, and R. J. Clarke. Leading edge erosion classification system. Technical report, Sandia National Lab.(SNL-NM), Albuquerque, NM (United States), 2022.

[14] T. J. Santner, B. J. Williams, W. I. Notz, and B. J. Williams. The design and analysis of computer experiments, volume 1 of Springer Series in Statistics. Springer, 2 edition, 2003.

[15] A. Sareen, C. A. Sapre, and M. S. Selig. Effects of leading edge erosion on wind turbine blade performance. Wind energy, 17(10):1531–1542, 2014.

[16] M. Schramm, H. Rahimi, B. Stoevesandt, and K. Tangager. The influence of eroded blades on wind turbine performance using numerical simulations. Energies, 10(9):1420, 2017.

[17] E. T. Spiller, R. L. Wolpert, P. Tierz, and T. G. Asher. The zero problem: Gaussian process emulators for range-constrained computer models. SIAM/ASA Journal on Uncertainty Quantification, 11(2):540–566, 2023.

[18] A. S. Verma, S. G. Castro, Z. Jiang, and J. J. Teuwen. Numerical investigation of rain droplet impact on offshore wind turbine blades under different rainfall conditions: A parametric study. Composite structures, 241:112096, 2020.

[19] J. Visbech, T. Göçmen, C. B. Hasager, H. Shkalov, M. Handberg, and K. P. Nielsen. Introducing a data-driven approach to predict site-specific leading-edge erosion from mesoscale weather simulations. Wind Energy Science, 8(2):173–191, 2023.