

Wind Energy Science

Response to the comments made by the Associate Editor and the
Reviewers regarding the manuscript WES-2025-62:

Simulating run-to-failure SCADA time series to enhance wind turbine fault
detection and prognosis

by

Ali Eftekhari Milani, Donatella Zappalá, Francesco Castellani and Simon
Watson

Dear Dr. Nikolay Dimitrov,

We appreciate the opportunity to provide a revised version of our manuscript and appreciate the valuable feedback received. We are grateful to the Reviewers for their insightful and constructive comments. In response, we have made careful revisions to address each point raised and improve the overall quality of the manuscript.

In this document, we provide our point-by-point replies to the comments/suggestions (highlighted in *italics*) and outline the specific changes made to the manuscript. All these changes are highlighted in blue in the submitted revised version of the manuscript.

Yours sincerely,

Ali Eftekhari Milani, Donatella Zappalá, Francesco Castellani, and Simon
Watson

Dr. Nikolay Dimitrov - Associate Editor

The authors have successfully addressed the first of the two editor comments, which was also the more critical one.

I still think further discussion is needed on the second point, because I believe with its present definition the health index does not directly link the observations with the physical failure phenomenon. This poses challenges towards generalizing the approach, as it is not certain that for another turbine

(or even for the replaced component from the same turbine) a new failure event will occur exactly at health index of 1.

I suggest the paper now proceeds to review, and the authors are welcome to address this outstanding point together with the reviewer comments.

We are glad that we were able to successfully address your first comment. The definition of failure in the context of prognostics and RUL prediction is an important topic. In practice, a component is usually declared failed when the parameters monitored through sensors exceed predefined thresholds. Therefore, a complete failure is rarely reached, and the indicated failure times are generally based on the sensor signals rather than on the level of actual physical degradation.

When using datasets from laboratory scale experiments, failure is usually clearly defined, and there is consistency among realisations in terms of their failure time. For example, in [1], the HI model used in this work has been tested on two bearing testbed datasets, in both of which the failure is defined based on a threshold on the amplitude of the vibration signals.

However, in the field, a component is usually declared failed when the parameters monitored by SCADA and/or CMS systems exceed predefined thresholds, raising alarms and planning maintenance visits to confirm the presence of significant degradation in the component. In this case, many factors are involved in deciding the actual time of the maintenance visit, which is usually declared as the failure time. Therefore, as you correctly indicated, there is an irreducible inconsistency in the definition of failure, which can be thought of as a form of labelling noise.

In the proposed HI model, the HI is by definition in the range of 0 to 1, with 0 indicating the pristine state (0% degradation) and 1 indicating failure (100% degradation). Therefore, both the limitation in the availability of failure events for training purposes and the above-mentioned inconsistency in the definition of failure lead to an error in the prediction of the failure time, meaning the HI reaches 1 either before or after the true failure time.

The availability of only one failure event definitely limits the generalisation of any method for fault detection and RUL prediction, which leads to large errors in the test set. This work proposes a method that synthetically introduces variability in the training data in terms of the operational and environmental conditions and the degradation trajectory, and serves as a feasibility analysis showing it is able to improve the generalisation and lower the test set errors. To identify the extent of diversity in the failure patterns

that can be simulated using this method and assess its performance in the presence of various failure modes, further studies and experiments with multiple failure cases are required. This clarification is added to the conclusions in line 352.

Reviewer #1

The paper concerns the detection/prediction of faults and remaining useful life (RUL) estimation for wind turbines. It presents a methodology where synthetic data, produced by a cGAN algorithm, is used to improve the accuracy of failure detection and RUL algorithms. The methodology is validated on data from a wind farm with a single failure and one near failure (maintenance).

The problem statement is very relevant. Failure prediction and RUL on wind turbines is an important research topic. This paper can be an interesting and valuable contribution. This means that in my opinion the topic is in scope of the WES journal.

The paper is well written, reads fluent, and is to the point. The 18 pages of text (excluding the references) describes well the problem statement. The methodology is relatively well discussed, however, certain parts are unclear or missing (see remarks part). The results are well presented. The conclusion is concise and to the point but in my opinion somewhat too strong in its assertions (see remarks part). The abstract is concise and to the point and describes the content of the paper well. The figures that are added to the paper are useful to understand the text. The figures are also clear.

The paper shows that when synthetic data is used using a cGAN the number of misclassified labels (unhealthy, healthy) decreases by 84% on average. The paper indicates how the synthetic data methodology improves the performance in this case.

I do have some remarks/questions that in my opinion need clarification in the paper:

We thank the reviewer for showing interest in our work and for the time spent to provide an in-depth and high-quality review. The comments and suggestions are very relevant, and we hope that we have been able to address them successfully, improving the quality of the manuscript.

1. *On p.4 it is said “the weights of the f_R and f_{std} are experimentally set to 3 to balance the four terms”. How was this done? Which data was used for the experiment?*

We agree that more information is required in terms of how these weights were set experimentally. Thank you for pointing this out. The weights of the training cost function are set to harmonise the optimisation speed of the four terms. For this aim, we have performed a trial-and-error using the training failure case. A weight of 1 for all four terms resulted in slow convergence due to the slow minimisation of the reconstruction error, and the obtained training HI tended to be noisy. Increasing the weights of the two loss terms corresponding to these factors, i.e., f_R and f_{std} , resolved this issue. To clarify how these weights were set, an explanation is added from line 117 in the manuscript.

2. *On p.6-8 the methodology is discussed for generating the synthetic SCADA signals using cGAN. Can you state more clearly what the hyperparameter values were for this algorithm? For example the number of layers in the networks, the number of neurons, ... On p.12 it is mentioned that the learning rate of the Adam optimizer is 0.0005. How were the hyperparameter values selected? Was hyperparameter tuning used? If so, how was this done? Which data was used for this? If no hyperparameter tuning was done, then where do the hyperparameter values come from? From literature? If so, is it not surprising that those parameters work well on the case discussed in the paper? Can you please discuss this?*

The reviewer correctly points out that more explanation is required in terms of the hyperparameters of the cGAN model, and the visual introduction of the architecture in Figure 4 might not be sufficient. As mentioned in the introduction, the architecture of the cGAN is inspired by the models proposed in [2] and [3], combining elements from these works and adapting the architecture to the problem addressed in this work. The hyperparameters were set through trial-and-error using the training wind turbine data. The objective was to find a setting that ensures a stable training process, with the Generator and the Discriminator being trained in tandem and with consistent speeds. In line 185, a paragraph is added to explain the model architecture and the hyperparameters in detail and clarify how they are set. Furthermore,

the selection of the 0.0005 learning rate for the Adam optimiser is discussed with an additional sentence in line 250.

3. *On p.7 it is said that the window length w is experimentally set to 10. How was this experimentally done? Which data was used for the experiment? Which metric was used to decide which window length is optimal?*

The reviewer is correct in pointing out that this parameter needs clarification in the manuscript. The window length introduces a trade-off between model complexity and performance. A larger window allows more information to be captured at each time frame from earlier signal values, at the expense of increased computational burden. According to the findings in [4], temperature signals, which are the most important SCADA signals when dealing with mechanical component degradation, typically exhibit autocorrelation up to a maximum of two to three days in the past. Our experiments using the training failure case pointed to a similar result. The HI obtained from the training wind turbine essentially remained identical for the window lengths larger than 10 (60 hours). This clarification is added in line 178 in the manuscript.

4. *On p.9 it is said “Each turbine has a diameter of 100 m . . .”. Although it is clear from the context that “rotor diameter” is meant, it might be useful to explicitly specify it is the rotor diameter.*

This is amended in the manuscript in line 193.

5. *On p.9 it is said that WT9 is selected as validation dataset. Why was this wind turbine selected? How was it selected? This dataset is used to set the detection threshold. It is known that the component temperatures between different wind turbines can be structurally different even if they are healthy. It is surprising that this did not have an impact on the number of false positives in the test dataset. Did you notice differences between the different turbines? Please discuss this.*

The difference in component temperatures across different turbines in a wind farm is an important point. The varying number of misclassified labels in different healthy turbines reported in Table 2 and shown in Figure 7, corroborates the reviewer’s point. The robustness of the fault detection method to this variability lies in the fact that it has been trained to associate only the signal patterns close to the failure time

with the label of 1, i.e., faulty state. The difference in signal behaviour between the healthy and the faulty states in the training wind turbine is much higher than the variability of the signal behaviour among different wind turbines in the healthy state. It is important to note that the seasonal behaviour in the signals outweighs both of these factors, and the model is able to learn the fault features only when synthetic signals are introduced, allowing the model to isolate the seasonal features and learn the fault features.

WT9 was randomly selected as the validation set. We agree that the selection of the validation wind turbine will have an effect on the number of false positives in the remaining wind turbines. However, rather than demonstrating the performance of the fault detection method, the aim of this case study is to compare the fault detection with and without introducing the synthetic signals and demonstrate the effectiveness of the synthetic signal generation method. And, lowering or raising the detection threshold affects the performance in these two cases similarly. To clarify this, a paragraph is added in line 267.

6. *On p.10 you say the following: "... the minority (faulty) class is over-sampled using the SMOTEN method ...". Should it not be SMOTE? How was SMOTE applied during the training of the model? On which data? Please discuss this.*

Thank you for bringing this typo to our attention. We have fixed it and added some explanation in line 230.

7. *As a baseline a classification model is trained on data from a wind turbine that experienced a gearbox failure (p.10). No more information on the failure type is given. 12 months of data preceding the failure are used for training the model. Data less than 1 month before the failure is labeled as unhealthy, the remaining 11 months are considered healthy. It is not explained in the paper how this decision was taken. Why 1 month and not more or less data? It might be good to discuss this in the paper.*

As indicated by the reviewer, more information about the fault is added to line 195.

The length of the considered training data and its division into the healthy and faulty sections is done considering a trade-off between sev-

eral factors. Degradation is generally a gradual and monotonically increasing process, and a clean separation between the healthy and faulty states does not exist. Selecting a smaller portion of data close to the failure date as the faulty class can reduce false positives. However, it leads to a higher data imbalance and can reduce model performance. In this work, to minimise the risk of false positives, the smallest length possible was selected for the faulty training data that maintains an acceptable level of data imbalance, allowing a successful training of the fault detection model. This is clarified in line 219.

8. *It might also be good to add a bit more information on how this classification model was trained. Was a train-validation split used? If so, how much data was assigned to training and how much to validation? On p.10 the architecture is described. It is stated that this architecture is a good trade-off between performance and computational burden. How was this decided? Was hyperparameter tuning performed? If so, how was it done, on which data? If no hyperparameter tuning was done, how were the values for the hyperparameters found? Please discuss this.*

These details are indeed missing in the manuscript. Thank you for bringing it to our attention. The model is trained using the Adam optimiser with its default parameters and the binary cross-entropy loss function. During training, a train-validation split is performed, where 20% of the training data is randomly set aside for validation, and the training is stopped when the validation loss stops decreasing for 20 consecutive epochs. The model architecture is set using trial-and-error, where the model complexity is gradually increased in terms of the number of hidden layers and neurons per layer, until a significant performance improvement is not observed. These details are added in line 223.

9. *Furthermore, by using only one failure for training, does this not risk overly specializing the model to the degradation pattern of this single failure? My experience is that failures, even if they are of the same type, show themselves often quite differently in the data. There tends to be large variance in the degradation patterns. How does this method handle this? How do you guarantee that the model does not just memorized this degradation pattern (or certain properties of it)? Have you tested it on other failure types? The fact that there are no false positives for the*

“healthy” turbines is surprising. Gearbox temperatures are most likely influenced by many environmental conditions, which are not always easy to measure and use in a model. So I would expect there to be more noise on the results. Please discuss this.

This is an important point. Using only one failure case for training does increase the risk of overfitting. This, in fact, is the main problem statement of this work: Can we alleviate the overfitting when we are limited to only one failure case for training? As a solution, a method is proposed that can use the training failure case to simulate new failure cases with predefined operational and environmental conditions and degradation patterns. It is shown that the synthetic variability introduced to the training set using this method can reduce overfitting.

However, the training and test fault cases in this work are quite similar, both involving a fault in the gearbox that leads to elevated temperatures in the gearbox-related SCADA signals. Therefore, as the reviewer correctly indicates, the model might not perform well if the test failure mode is significantly different from the training one. This is now clarified in the conclusions in line 352.

10. *Figure 7 and the explanation on p.12 indicates that an anomaly zone was identified for WT6, and that this was most likely associated with a maintenance. An analysis of the pattern shows that there was an initial jump of the fault index, then it decreased sharply and then it stayed at a lower but elevated level. If it identified damage to the gearbox, why do we not see an increasing trend in the fault index over time? What causes the initial jump, and subsequent sharp decrease? It might be useful to add some discussion of this to the paper.*

This is an interesting observation. The fault index measures the density of the detected faulty time stamps in a weekly rolling time window, i.e., the ratio of the faulty time stamps to total time stamps in a week. While it is expected that a more severe fault should generally lead to more frequent alarms raised by the fault detection model, other factors play a role as well. For example, the operational mode and the ambient conditions. Therefore, this index can be noisy, and it is not expected to always reflect the actual level of degradation in the component. For this reason, the proposed HI construction method is used in the second case study. This method decomposes the signals and isolates the factor

corresponding to the degradation in the component. Therefore, the HI built corresponds to the severity of degradation through time.

Upon comparing the fault index and the HI built for the test case, it can be seen that a jump is also visible in the test HI around the time when the jump in the fault index is observed. This might be due to a sudden fault, such as a crack. However, this hypothesis cannot be asserted with confidence, since no in-depth details are available about this fault. A sentence is added to line 322 to discuss this.

11. *On p.14-15, it is explained how 4 trends are used for the synthetic HIs. These are all based on characteristics of the failure for WT8. Does this method not risk overly specializing the model to this single failure? Does this not mean that the good results achieved are limited to this single failure? What can be expected in conditions with multiple failures, how would the methodology be applied? What will be the impact of this? It might be useful to discuss this.*

This is a valid concern, and links with comment 9. The availability of only one failure case is a hard limit in this work, and it is shown that the proposed synthetic signal generation method can reduce the overfitting resulting from this limitation. However, it is not guaranteed that in all cases the results can be as good as the case study presented in this work. This work serves as a feasibility analysis that proves the proposed approach can generate entire sets of time series simulating new failure events that are able to mitigate the overfitting problem. To identify the extent of diversity in the failure patterns that can be simulated using this method and assess its performance in the presence of various failure modes, further studies and experiments with multiple failure cases are required. These points are clarified in the conclusions in line 354.

Considering the application with multiple failure events available for training, a similar approach can be taken, generating several synthetic failure cases based on each of the training failure cases.

12. *On p.16 the results of the RUL estimation are discussed. A second order polynomial is fit on the HI to predict the RUL up to the detection point. How was the decision taken to use a second order polynomial? What was the procedure? Was the decision taken by looking at the*

shape of the HIs? If this is the case don't you run the risk of overfitting this specific case? Is using a second order polynomial still valid when testing on other failures (degradation patterns)? Again the variance in the degradation patterns plays a role here. Please discuss this.

The method used was to fit a second-order polynomial to the HI, constraining the quadratic coefficient to be non-negative. This approach can model both linear and curved trends, using the fewest parameters possible, minimising the risk of overfitting. This explanation is added to line 331.

This function is fitted to an initial section of the test HI and is extrapolated to predict the RUL. Therefore, it is adapted to each failure case, and can be expected to perform similarly in other failure cases with reasonably consistent HI trends. However, it will not perform well if the trend is less consistent during the component's lifetime. This clarification is added to line 360.

Accurately forecasting the future trajectory of an HI for RUL prediction is an important topic which is out of the scope of this work. The simple method used in this work only serves as a tool to compare the performance of the HIs built with and without synthetic data.

13. *In my opinion the paper results can be seen as a proof of concept of a methodology. Quite some assumptions (a.o. how a degradation signal looks like, ...) are made during the construction of the pipeline. To know how well it would perform in general, a larger analysis is required on more failures (of different types). It might be useful to add this point more clearly to the conclusion of the paper.*

This is a correct remark, and is added to the conclusions in line 354.

14. *It might also be useful to add a schematic overview of which data was used by which part of the pipeline.*

A flowchart can indeed help in better explaining the pipeline and the data used in each step. Figures 7 and 10 are added to address this, showing the flowchart of the methodology with and without synthetic datasets and the data used.

15. *Conclusion: This paper is in scope of WES. It contains relevant research, and is in my opinion after addition of the discussions good for*

publication.

We are grateful for the time and effort put into this review. A lot of valuable comments are provided, and addressing them has contributed significantly to the quality of the manuscript.

Reviewer #2

This work addresses the failure detection and prognosis in the context of wind turbine operation. The paper introduces a synthetic data generation methodology for the training of failure detection and remaining useful life (RUL) prediction algorithms by using cGAN to generate SCADA data abiding to predefined conditions. The methodology aims to improve the prediction accuracies of the algorithms by providing more data samples that can better represent degradation trends of the wind turbine. SCADA dataset from a wind farm was used to validate the methodology.

In my opinion, the manuscript addresses an important research topic that is very relevant and within the scope of WES journal. The manuscript provided a sound methodology, and the content is a valuable contribution to the research area discussed.

Overall, the language of the paper is well written tonally, and the figures were clear and helped in presenting the findings. The manuscript provided a very good overview of the problem statement, while also providing relevant literature review to address the limitations of past works. The methodology is mostly well discussed and presented, but the structure of the result sections can be improved to provide more clarity for the steps taken to reach the conclusion. The conclusion is concise and summarised the findings well, but the limitations of the work can be elaborated.

My comments and questions will be listed below, points that in my opinion need clarification will be listed in Remarks and editorial suggestions will be listed in minor comments.

We thank the reviewer for taking the time to read the manuscript and provide valuable comments, and we are happy that he/she has found our contribution relevant and valuable for the field.

Remarks:

1. pg.4 - *“the weights of the fR and fstd terms are experimentally set to 3*

to balance the four terms”, what do you mean by “experimentally set”? Was it by iterations with a sample data-point?

We agree that more information is required in terms of how these weights were set experimentally. Thank you for pointing this out. The weights of the training cost function are set to harmonise the optimisation speed of the four terms. For this aim, we have performed a trial-and-error using the training failure case. A weight of 1 for all four terms resulted in slow convergence due to the slow minimisation of the reconstruction error, and the obtained training HI tended to be noisy. Increasing the weights of the two loss terms corresponding to these factors, i.e., f_R and f_{std} , resolved this issue. To clarify how these weights were set, an explanation is added from line 117 in the manuscript.

2. *Pg.9 – Section 3: Dataset, what is the test-train split strategy adopted? Why is only WT9 data used as validation set instead of sampling from all the wind turbines?*

This is an interesting point. Sampling from all wind turbines to obtain a validation set would have been a better approach if signal continuity and temporal ordering were not a requirement for the proposed method. In the fault detection method, the fault index measures the density of the misclassified labels in a rolling weekly window. Therefore, the validation set used to set the detection threshold for this fault index must maintain the continuity and temporal order between data instances. For this reason, the division is done based on turbines. One of the two wind turbines with a gearbox fault is used for training. One of the healthy wind turbines is selected as the validation wind turbine used to set the fault detection threshold, and the remaining wind turbines are used for testing.

It is important to note that the temporal ordering of data instances is relevant to building the fault index using the raw 0/1 labels, and not to the ANN generating these labels, since this model gets as input individual data instances and for each one predicts a label. Therefore, during its training using the WT8 data, a random 80%-20% train-validation split of the data instances is performed, and the training is stopped when the validation loss stops decreasing for 20 consecutive epochs. This validation set is also used for setting the ANN’s architecture. These details were initially missing in the manuscript and are

now added in line 225.

3. *pg.10 – “SMOTEN method”, I believe this is a typo. Which dataset was resampled? What value was resampled and why?*

Thank you for bringing this typo to our attention. We have fixed it and added some explanation in line 230. In the training dataset, i.e., WT8 data during the year leading to the gearbox failure, SMOTE was used to oversample the faulty class and resolve the class imbalance.

4. *Pg.12 – “The Adam optimiser”, it appears that hyperparameter fine-tuning was performed to optimise the algorithm, it will be good to include the fine-tuning strategy adopted to reach this conclusion.*

The reviewer correctly points out that more explanation is required in terms of the hyperparameters of the cGAN model and the strategy used to set them. The hyperparameters were set through trial-and-error using the training wind turbine data. The objective was to find a setting that ensures a stable training process, with the Generator and the Discriminator being trained in tandem and with consistent speeds. In line 185, a paragraph is added to explain the model architecture and the hyperparameters in detail and clarify how they are set. Furthermore, the selection of the 0.0005 learning rate for the Adam optimiser is discussed with an additional sentence in line 250.

5. *Pg.12-13 – Results on misclassified labels and false positives, it is interesting to see that despite having misclassified labels, most of the WTs had 0 false positives. Would you not consider this to be a sign of over-fitting?*

We agree that more discussion regarding these results is required in the manuscript. Since the state change from healthy to faulty is generally gradual, an individual time stamp predicted as faulty might be due to noise rather than an actual fault. For this reason, the fault detection in this work is based on a robust fault index which measures the density of the predicted faulty instances in a rolling weekly time window. A fault is detected when this fault index crosses a detection threshold set using the validation wind turbine. Hence, false positives are the time stamps at which the fault index is above the detection threshold while the turbine is healthy. In other words, although misclassified labels exist due to the noisy nature of the raw labels, a fault is only detected when

the density of the 1 labels in a weekly time window increases past a threshold value. It is important to note that this threshold is set based on the randomly selected validation wind turbine (WT9). Therefore, selecting a different wind turbine for validation can slightly change the number of false positives by lowering or raising the detection threshold. However, since this affects the number of false positives similarly in the two cases with and without synthetic datasets, it can act as a valid tool for performance comparison. These points are added in line 267.

6. *Section 5. RUL prediction case study, what is the baseline and structure of your performance measurement for the proposed method? Discussion on why the monotonicity value from the MK metric is relevant to the quality of degradation trend and how it can affect the RUL prediction should be included.*

This is an important context that needs to be added to the methodology section. Thank you for bringing it to our attention. Due to the generally irreversible nature of component degradation, an HI is expected to demonstrate a monotonic trend, and monotonicity has been widely used as one of the main criteria to build HIs. Therefore, maximising monotonicity leads to an HI that better represents the component degradation, leading to a more accurate RUL prediction. This clarification and relevant citations are added in line 101.

7. *Pg.16 – Justification to why the second-order polynomial function is used to predict RUL is needed. The method is only tested with WT6 dataset, this makes me wonder if the same method will be applicable and effective on a different failure case from a different wind turbine.*

The method used was to fit a second-order polynomial to the HI, constraining the quadratic coefficient to be non-negative. This approach can model both linear and curved trends, using the fewest parameters possible, minimising the risk of overfitting. This explanation is added to line 331.

This function is fitted to an initial section of the test HI and is extrapolated to predict the RUL. Therefore, it is adapted to each failure case, and can be expected to perform similarly in other failure cases with reasonably consistent HI trends. However, it will not perform well if the trend is less consistent during the component’s lifetime. This clarification is added to line 360.

Accurately forecasting the future trajectory of an HI for RUL prediction is an important topic which is out of the scope of this work. The simple method used in this work only serves as a tool to compare the performance of the HIs built with and without synthetic data.

Minor comments (suggestions):

1. In pg.1 line 13, “... to produce reliable RUL ~~estimates~~ estimations.”
2. In pg.1 line 24, “... improving their robustness and ~~practical applicability~~ practicality.”
3. In pg.10, line 200, “the effectiveness of the developed method in fault detection.” Can be clearer on which developed method this is referring to (e. of the developed synthetic data generation method).

The three suggestions above are implemented in the manuscript. Thank you for pointing them out.

4. For section 4., a short separation sentence can be included to clarify that the result of the HI produced from (a) SMOTE generated data and (b) cGAN generated data will be compared and discussed. I also find that separating [texts addressing model configurations and methods adopted] from [result presentation] into different paragraphs can improve the structure of the section.

A sentence is added to line 216 to clarify which two specific cases are compared in section 4. The explanation of these two cases was confusing in this section in the initial version of the manuscript. We have now fixed that. The two cases compared are a) with only the original WT8 dataset, and b) with both the original WT8 dataset and the synthetic datasets. The faulty class in both the original and the synthetic datasets are oversampled using SMOTE to balance the classes before using them to train the ANN. To clarify this, Figure 6 is edited to only show the oversampling using SMOTE, and Figures 7 and 10 are added to show the methodology flowchart without and with synthetic datasets.

References

- [1] Ali Eftekhari Milani, Donatella Zappalá, and Simon J. Watson. A hybrid convolutional autoencoder training algorithm for unsupervised bearing health indicator construction. *Engineering Applications of Artificial Intelligence*, 139:109477, January 2025. ISSN 0952-1976. doi: 10.1016/j.engappai.2024.109477. URL <http://dx.doi.org/10.1016/j.engappai.2024.109477>.
- [2] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets, 2014. URL <https://arxiv.org/abs/1411.1784>.
- [3] Jinsung Yoon, Daniel Jarrett, and Mihaela van der Schaar. Time-series generative adversarial networks. In H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, pages 5508–5518, 2019.
- [4] Ali Eftekhari Milani, Donatella Zappalá, Francesco Castellani, and Simon Watson. Boosting field data using synthetic scada datasets for wind turbine condition monitoring. *Journal of Physics: Conference Series*, 2024.