

Apaar Bhatnagar

app (10) copy

 Research Paper

Document Details

Submission ID

trn:oid:::27535:141142441

Submission Date

May 31, 2026, 8:18 PM GMT+5:30

Download Date

May 31, 2026, 8:23 PM GMT+5:30

File Name

app (10) copy.pdf

File Size

835.8 KB

42 Pages

10,415 Words

53,821 Characters

10% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography

Match Groups

-  **95 Not Cited or Quoted 10%**
Matches with neither in-text citation nor quotation marks
-  **2 Missing Quotations 0%**
Matches that are still very similar to source material
-  **0 Missing Citation 0%**
Matches that have quotation marks, but no in-text citation
-  **0 Cited and Quoted 0%**
Matches with in-text citation present, but no quotation marks

Top Sources

- 7%  Internet sources
- 6%  Publications
- 7%  Submitted works (Student Papers)

Match Groups

- 95 Not Cited or Quoted 10%**
Matches with neither in-text citation nor quotation marks
- 2 Missing Quotations 0%**
Matches that are still very similar to source material
- 0 Missing Citation 0%**
Matches that have quotation marks, but no in-text citation
- 0 Cited and Quoted 0%**
Matches with in-text citation present, but no quotation marks

Top Sources

- 7% Internet sources
- 6% Publications
- 7% Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Internet	www.preprints.org	<1%
2	Student papers	Vrije Universiteit Amsterdam on 2026-03-27	<1%
3	Publication	Natasa Kleanthous, Abir Hussain. "Machine Learning in Farm Animal Behavior usi...	<1%
4	Student papers	University of Hong Kong on 2025-08-02	<1%
5	Internet	pdfcoffee.com	<1%
6	Internet	www.mdpi.com	<1%
7	Student papers	Monash University on 2022-11-09	<1%
8	Internet	winnspace.uwinnipeg.ca	<1%
9	Internet	arxiv.org	<1%
10	Internet	www.geeksforgeeks.org	<1%

11	Publication	"New Paradigms in Big Data Technology and Business Analytics", Springer Scienc...	<1%
12	Internet	fdocuments.us	<1%
13	Internet	tede.unioeste.br	<1%
14	Internet	www.coursehero.com	<1%
15	Internet	romisatriawahono.net	<1%
16	Student papers	Strathmore University (Main Account) on 2026-03-26	<1%
17	Publication	Saul A. Alokabel, M.N. Setiawan. "Preliminary Techno-Economic Analysis of Wind ...	<1%
18	Student papers	University of Technology, Sydney on 2026-04-26	<1%
19	Internet	cel.hal.science	<1%
20	Internet	codefinity.com	<1%
21	Internet	ebin.pub	<1%
22	Internet	github.com	<1%
23	Student papers	Universidad Católica de Santa María on 2019-02-12	<1%
24	Internet	dtu.ac.in	<1%

25	Publication	Vikrant S. Vairagade, Boskey V. Bahoria, Haytham F. Isleem, Nilesh Shelke, Nischa...	<1%
26	Student papers	University of Technology, Sydney on 2024-11-16	<1%
27	Internet	www.projectpro.io	<1%
28	Publication	"Cyber Intelligence and Information Retrieval", Springer Science and Business Me...	<1%
29	Publication	Hailong Zhu, Yawen Xie, Wei He, Chao Sun, Kaili Zhu, Guohui Zhou, Ning Ma. "A N...	<1%
30	Internet	lib.buet.ac.bd:8080	<1%
31	Internet	www.researchgate.net	<1%
32	Publication	Hongbin Liu, Ziyan Wang, Yizhuo Liu, Jie Zhou, Chen Chen, Haoyuan Ma, Xi Huang...	<1%
33	Student papers	Librarian NITT on 2025-11-24	<1%
34	Student papers	Strathmore University (Main Account) on 2026-02-27	<1%
35	Internet	d197for5662m48.cloudfront.net	<1%
36	Internet	ijsrcse.isroset.org	<1%
37	Internet	ntnuopen.ntnu.no	<1%
38	Publication	Qiang Gao, Nesimi Ertuğrul. "Techno-Economics of Offshore Renewable Energy - ...	<1%

39	Internet	norma.ncirl.ie	<1%
40	Internet	www.ejournal.isha.or.id	<1%
41	Publication	Mutlu Yuksel, Yigit Aydede. "Causal Inference and Machine Learning - In Economi...	<1%
42	Student papers	Strathmore University (Main Account) on 2026-03-27	<1%
43	Internet	fastercapital.com	<1%
44	Student papers	Georgia Institute of Technology Main Campus on 2024-07-07	<1%
45	Publication	Mukesh Kumar Awasthi, Sanoj Kumar, Deepika Saini. "Artificial Intelligence Tech...	<1%
46	Publication	Sanjeev J. Wagh, Manisha S. Bhende, Anuradha D. Thakare. "Fundamentals of Dat...	<1%
47	Publication	T. Mariprasath, Kumar Reddy Cheepati, Marco Rivera. "Practical Guide to Machin...	<1%
48	Internet	dspace.dtu.ac.in:8080	<1%
49	Internet	www.atlantis-press.com	<1%
50	Publication	Baek, Kwangho. "Toward Adaptive Transit Planning With Artificial Intelligence.", ...	<1%
51	Publication	Bittu, Pavani. "Prediction of the Need for Mental Health Therapy Among COVID-1...	<1%
52	Student papers	Delhi Technological University on 2026-05-06	<1%

53	Student papers	Indian Institute of Technology Guwahati on 2024-05-22	<1%
54	Student papers	Servizo de Xestión de Estudos de Posgrao on 2025-09-21	<1%
55	Student papers	University of Queensland on 2025-06-08	<1%
56	Publication	Usha Sakthivel, Nagarajan Susila. "Applied Learning Algorithms for Intelligent Io...	<1%
57	Internet	cdigital.uv.mx	<1%
58	Internet	dokumen.pub	<1%
59	Internet	download.bibis.ir	<1%
60	Internet	learn.modernagecoders.com	<1%
61	Internet	studenttheses.uu.nl	<1%
62	Internet	www.elpalaciodehierro.com	<1%
63	Student papers	Georgia Institute of Technology Main Campus on 2026-03-22	<1%
64	Student papers	Indiana University on 2025-12-09	<1%
65	Student papers	University of Sydney on 2025-10-26	<1%

**DELHI TECHNOLOGICAL UNIVERSITY**

(Formerly Delhi College of Engineering)

Shahbad Daultpur, Main Bawana Road, Delhi-110042

ANALYSIS AND DETECTION OF WIND TURBINE BLADE PERFORMANCE USING MACHINE LEARNING ALGORITHM

Apaar Bhatnagar

ABSTRACT

Wind energy is one of the fastest growing renewable energy enterprises in the world. The structural integrity of wind turbine blades is of prime importance for the efficient, safe and economic operation of wind farms. Slow erosion and corrosion of turbine blades especially from high velocity impact from rain, hail, sand and UV radiation cause large aerodynamic performance losses and unpredicted maintenance costs. Traditional condition monitoring approaches mainly rely on human periodic inspections using deterministic threshold approaches, which are not adaptive to the dynamic non-linear operational environment of modern wind turbines. This thesis proposes a predictive maintenance framework for wind turbine blades based on machine learning. The method is meant to classify the structural health state of blades into four operationally relevant classes: Good, Minor degradation, Advanced degradation and Major degradation. The data used in the present study was obtained from the National Renewable Energy Laboratory (NREL) Wind Resource Database for the Mumbai coastline region and consisted of 8,762 hourly records of atmospheric parameters at multiple altitudes (wind speed, temperature, air pressure and wind direction). But there is no explicitly available blade health indicators on the NREL data set. Thus a physics-informed stress index was developed to indicate the state of blade health categories. The stress index has taken the main environmental variables wind speed at 100 and 120 meters weighted by experimental coefficients and ambient temperature in order to indicate an appropriate blade health degradation proxy. Four supervised classification techniques, Logistic Regression, Decision Tree, Random Forest and XGBoost are well-trained and well-tested. Training is carried out on 80% of the processed data and testing is carried out on the rest 20% of the data in terms of Accuracy, precision, recall, F1-score and confusion matrix evaluation. It is also concluded that XGboost achieved the best classification accuracy of 99.32% which is followed by Logistic Regression accuracy of 99.20%, Random Forest accuracy of 99.09% and Decision Tree accuracy of 98.57% . Analysis of feature importance of tree based models identified wind speed

at height of 100m and 120m height as the key factors to predict blade degradation. The framework also proves the feasibility of data driven predictive maintenance system for wind turbine blade and acts as a foundation for real time monitoring and scheduling of the wind turbines under operation.

keywords: Wind Turbine Blade Erosion, Predictive Maintenance, Condition Monitoring, Machine Learning, XGBoost, NREL Wind Resource Database, Blade Health Classification.

12

CHAPTER 1

INTRODUCTION

6

1.1 Background and Motivation

The global urgency to mitigate anthropogenic climate change has speeded up the transition in the energy systems from fossil to renewable energy. Wind energy has become one of the most competitive and fastest scalable sources of clean electricity among renewable technologies. The Inter-governmental Panel on Climate Change Sixth Assessment Report [1] suggests that the global mean surface temperature has increased by approximately 1 °C between 2010 and 2019, leading to decarbonization of the global energy mix. Wind power has risen to the challenge, with the levelized cost of energy from wind power falling by about 55% between 2010 and 2019, offshore wind seeing the fastest fall. From the early 2020s onwards, progressively larger and higher power-output wind turbines are being installed, on land and at sea. Future wind turbine towers are to be up to 250 m in height and their rotor blades will travel up to 100 m/s at their tips [3]. While the improved capture of wind energy by this technology is a significant advance, the increasing speeds mean that an equally significant challenge will be produced in the wear of these structures-primarily through the edge erosion of the blades. Degradation of the blade, however has a great economic significance. A wind turbine operating at tip speeds of up to 100m/s, will be subject to attack from raindrop, hailstone, airborne sand particles, sea salt spray and ultra-violet radiation [10]. Over a period of time the surface is more subject to roughening, coating delamination, and at worst revealing the structural laminate. Tests conducted by Belzona have demonstrated that leading edge erosion alone, can result in aerodynamic losses of up to 20%. Three main types of maintenance strategy for wind turbines may be distinguished: (1) corrective maintenance (repair of component after failure), (2) preventive maintenance (fixed schedules of maintenance independent of the real condition of the component), and (3) condition-based maintenance (linking monitoring data with the decisions of maintenance) [20]. First two methods are generally inefficient; the latter has risk of structure catastrophic failure, and the second one leads to unnecessary repair action and expenditures. The third one, although conceptually better, used to suffer

from the quality and definition of data provided by the sensors. The development of machine learning (ML) as a mature analytical discipline can change the way of wind turbine health monitoring. To minimize the burden of manual work ML algorithm are used to extract non linear patterns from the timeseries data so that degradation of states can be predicted with high accuracy. This technique

Is very much helpful in maximizing the feasibility of wind farm productions, minimizing maintenance cost, increasing the work life of the system This research helps to identify the problem of optimization by performing the test on the supervised ML algorithms. Blades are the main elements of a wind turbine that enable energy conversion and their health determines [1] the efficiency, aerodynamic and economic stability and structural safety of wind generation. There is indeed a set of correlated problems that weaken current practices in wind blade health management.

1. **Absence of real-time, quantitative degradation monitoring:** Routine visual checks of operational wind farms, using rope access technicians or drones, are most commonly undertaken. These inspections can be limited by accessibility, weather conditions and inspector's discretion, and are especially difficult offshore and during the winter months [3].
2. **Lack of predictive capability in existing models:** The maintenance strategies based on the deterioration models defined by Markov chain theory like in Besnard Bertling [20], define a probabilistic scenario for maintenance but demand for continuous updating of the parameters and don't use the information available from multivariate sensor readings.
3. **Absence of labeled health datasets:** Existing open wind resources, such as NREL datasets, record the weather conditions at a fine temporal resolution but there is no direct label for the condition of blades. Thus, simulated health labels based on physics-informed indicators have to be established, which may require some assumptions of models that have to be verified.
4. **Computational inaccessibility of existing solutions:** A number of the condition monitoring methods that have been presented in the literature need specific vibration sensors, acoustic emission equipment or computer vision facilities, and as such are either expensive or difficult to implement on an existing turbine installation. Thus, the industry needs a data-driven approach with high computational efficiency, that can assign to the blade the health states, by using standard available measurements from environment and operation, without need of specialized sensor equipments.

1.2 Research Objectives

The detailed objectives of the study are:

1. A critical overview of literature on Wind turbine blade erosion mechanisms, state-of-art monitoring systems and role of Machine learning techniques in wind energy system prognosis.
2. Pre-processing and characterization of a specific atmospheric data in Mumbai coastal area from NREL Wind Resource data.
3. Training and testing of a multi-class classifier based on four supervised ML models- Logistic Regression, Decision Tree, Random Forest and XGBoost to detect the condition of the wind turbine blade.
4. Evaluation of above four ML classifiers based on accuracy, precision, recall and F1-score as well as through confusion matrix.
5. Exploration of the feature importance from the tree based ML models in order to find the environmental causes which greatly impact blade degradation.
6. An outline on applying the study findings in real-time wind energy systems.

1.3 Scope and Limitations

This work is restricted to classification of wind turbine blades health issues based on atmospheric measurement data. The study does not consider the integration of the model obtained in real time on SCADA or IoT platform. There is no physical blade test, no vibration based diagnostics. The labels of blade conditions used to train the model are simulated with a heuristic stress index and are not experimentally proven ground truth. This is a known shortcoming discussed in future scope and discussion sections.

1.4 Organisation of the Report

The rest of the thesis is presented as follows:

- Chapter 2, describes the technical principles related to the wind turbine blade erosion mechanics, current existing solutions of condition monitoring, provides the background of used machine learning algorithms and the description of used NREL dataset.
- Chapter 3, overviews the existing works in structured way using the traditional

techniques of blade maintenance and its health monitoring, and future prediction based maintenance.

- Chapter 4, focuses on proposed approach, the data pipeline, the working principle for the stress index formulation, feature engineering and training parameters.
- Chapter 5, introduces the development of software, the experimental setup and the experimental tests performed.
- Chapter 6, analyses and discusses the experiment results in context of comparison of the performance among different models.
- Chapter 7, offers the conclusion for the thesis and the work conducted during the thesis as a contribution, as well as proposes the future work to be performed.

1.5 Wind Energy: Global Context and Engineering Imperatives

The energy extracted from wind power depends on the energy held in the motion of masses of air. In an electromagnetic induction system the motion of the air masses is converted to electrical energy. The energy is transferred through the aerodynamic profiled blades (these act like wings that revolve and power a generator directly or indirectly through a gear box, producing electrical energy) to the generator which receives the motion and drives it; The power in a wind stream is calculated by using the fundamental aerodynamic equation given below.

$$P = \frac{1}{2} \rho A v^3 \quad (1.1)$$

where ρ is the density of air (about 1.225 kg/m³ at sea level), A is the rotor swept area in m² and v is the wind speed in m/s. The Betz limit provides a theoretical maximum power extraction coefficient of $C_p = 16/27 \approx 0.593$. This means that a wind turbine can never capture more than 59.3% of the kinetic energy in the wind. Modern commercial turbines achieve C_p values between 0.45 and 0.50 under optimum conditions.

The available wind power is proportional to the cube of the wind speed (Eqn. 1.1), so fairly small increases in blade surface roughness caused by erosion (which reduce the effective lift-to-drag ratio of the aerofoil) can have a very large effect on energy yield. A 10% increase in drag coefficient due to leading-edge erosion can result in energy production losses of 5–20% depending on blade geometry and operating conditions

1.6 Wind Turbine Blade Aerodynamics and Erosion Mechanics

1.6.1 Blade Aerofoil Geometry and Performance

The aerofoils used in the manufacture of modern horizontal axis wind turbine (HAWT) blades are cambered and so create aerodynamic lift when air flows over them. The aerofoil shape starts with a circular section near the root and increases towards a high lift aerofoil shape at the tip and along the span of the blade both chord length and twist angle vary. The leading edge, which is the forward most point on the aerofoil section experiences high dynamic pressures and thus has the greatest propensity for erosion. The values of both lift coefficient CL and drag coefficient CD of an aerofoil are determined by angle of attack and the roughness of its surface. Increased roughness due to erosion will cause boundary layer transition to occur earlier thus increasing CD and reducing CL .

1.6.2 Mechanisms of Erosion

Leading-edge erosion is caused by a number of different physical processes; these processes are more prevalent under differing operating conditions:

Rain Erosion Impact of water droplets at high relative velocities (up to 100 m/s at the blade tip) generates compressive stress waves in the blade surface material. Repeated impact causes fatigue crack initiation and propagation in the protective coating or the underlying glass-fibre laminate [10]. The erosion rate is highly sensitive to droplet size and impact velocity. Herring et al. showed that the combination of rain and strong wind was the dominant driver of damage to the blade surface.

Particle Erosion The erosive effect of Airborne particulates like sea-salt, sand and dust is the process by which they are scraped or plough the surface layer of the blade resulting into removal or by plastic deformation of surface particles. The extreme conditions are observed in coastal region and in dry on-shore.

Ice and Hail Erosion In cold climates, ice build-up on the blade surface changes the aerofoil shape and adds mass, whilst hail impact can lead to direct laminate damage. De-icing equipment may be fitted to northern-latitude offshore turbines.

UV and Chemical Degradation Degradation of the protective coatings polymer matrix from long term UV exposure and atmospheric pollutants can result in chalking, cracking and delamination.

1.6.3 Five-Stage Erosion Classification

The most commonly used framework for classifying the severity of leading-edge erosion was empirically established by Gaudern [9] at Vestas, who identified five progressive damage stages .

Step 1: Approx. 2mm dia., pin-holes approximately 0.1 - 0.2mm deep are observed in coating. Blade surface is discolored but the structure is in good order.

Step 2: Pin holes are developing into much larger holes, pits and shallow grooves. There is also isolated local flaking of coating away.

Step 3: Localized, extensive flaking away of coating, exposing the underlying primer layer. Roughness is increased measurably on the profile reducing aerodynamic performance.

Step 4: Primer coating penetrated with glass fibre laminate showing through. Significant increase in air resistance causing reduction in output.

Step 5: Large area laminate exposure with many large grooves over 500 mm length and up to 1.2 mm depth. Structural risk is very high and immediate repair required.

It is this classification which is the fundamental basis of the four category health labelling scheme which is used within this thesis (Good, Minor, Advanced, Major).

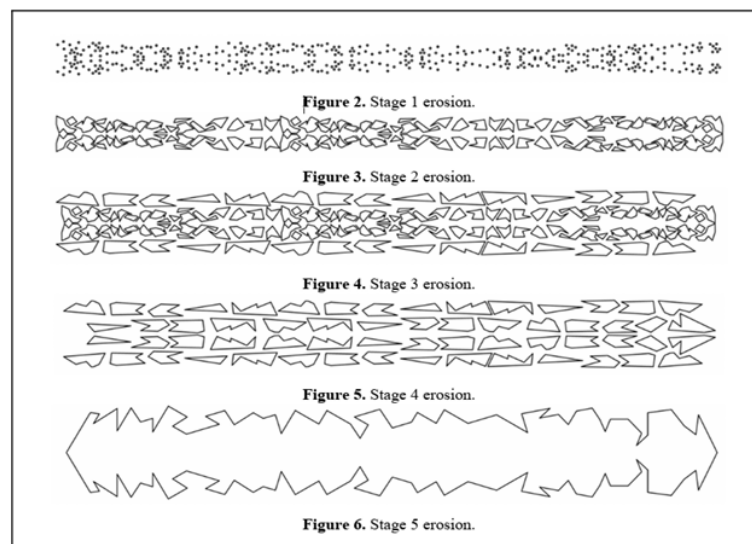


Figure 1.1. There are five stages in the spreading of damage:
[We can clearly see how a turbine blade decays over time: there is a small hole formed (stage 1), larger patches form and become bigger (stage 2-3), these large patches join up and huge scars are produced (stage 4). The amount of blade not covered increases dramatically (stage 5) causing negative impacts on the performance.]

1.7 Condition Monitoring Systems for Wind Turbines

1.7.1 Conventional Monitoring Approaches

In the following paragraphs, the common methods used in diagnosing the current state of a wind turbine and the kinds of sensors and their application in a wind turbine is listed:

SCADA-Based Monitoring SCADA are integrated into all sizes utility of wind turbine and collect operational data, such as power, revolutions, nacelle temperature, and wind speed every fixed interval of 10min. Although SCADA is capable of demonstrating overall condition and property trends, it is unable to detect early stage of damage of a blade with sufficient sensitivity owing to the lack of sufficient sampling rate and content information.

Vibration Based Monitoring This kind of sensor is used to identify vibrational signals of a blade or the structure of nacelle in a wind turbine. Such sensors are mounted inside the roots of the blade or on the structure of nacelle. Vibrational frequency signatures of wind turbine blades can be significantly different depending on blade damage because of varying surface roughness during erosion. As found, this property can accurately diagnose the damage with image processing by deep learning approach [18]. Moreover, initiating cracks, delaminations and propagating damage emit pulses of acoustic energy, which can be effectively identified using high frequency sensitive materials, such as piezoelectric sensors.

Acoustic Emission Monitoring The monitoring method is high sensitive and can be used but is prone to requiring closely spaced sensor array and high level of signal processing expertise.

Visual Optic Examination Usually consists of high resolution camera mounted on an unmanned aerial vehicle. It was found that stage 1 and stage 2 erosion can be identified using deep learning image processing with UAV image data [8].

1.7.2 Deterioration Modelling: Besnard-Bertling Framework

The pioneer theoretical model in condition-based maintenance optimization for wind turbines was presented by Besnard and Bertling [20]. They have a continuous-time Markov chain model with 5 levels of degradation (equivalent to the Gaudern classification) and exponentially distributed transition times. The transition intensities Q dictate the time evolution of the condition of blades, the matrix [20] referring to

$$\mathbf{Q} = \begin{pmatrix} -\lambda_1 & \lambda_1 & 0 & 0 & 0 \\ 0 & -\lambda_2 & \lambda_2 & 0 & 0 \\ 0 & 0 & -\lambda_3 & \lambda_3 & 0 \\ 0 & 0 & 0 & -\lambda_4 & \lambda_4 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (1.2)$$

where λ_k is the transition rate from state k to state $k+1$. This explains that how inspection interval and cost and probability values are interacting. This model applies to turbines on the ground where factors responsible to causes of erosion are independent of time such as coastal location Mumbai.

1.8 Machine Learning Algorithms: Theoretical Foundations

1.8.1 Logistic Regression

Logistic Regression, the posterior probability of belonging to class [1] is modeled using the logistic (sigmoid) function: With a binary classification problem for feature vector $\mathbf{x} \in \mathbb{R}^d$, and binary label $y \in \{0, 1\}$, the below equation [5] referring to, $\mathbf{x} \in \mathbb{R}^d$ and binary label $y \in \{0, 1\}$:

$$P(y = 1 | \mathbf{x}) = \sigma(\mathbf{w}^\top \mathbf{x} + b) = \frac{1}{1 + e^{-(\mathbf{w}^\top \mathbf{x} + b)}} \quad (1.3)$$

Here $\mathbf{w}$ denotes the weight vector and b denotes the bias term. For multiclass problem (four types of blade conditions), we use OvR(One-vs-Rest) method and train four binary classifiers respectively and the class with the largest predicted value is chosen. The model parameters $\mathbf{w}$ and b are learned by minimizing the regularized cross-entropy loss function the below equation [6] referring to,

$$\mathcal{L}(\mathbf{w}, b) = \sum_{i=1}^N [-y_i \log \hat{y}_i - (1 - y_i) \log(1 - \hat{y}_i)] + \lambda \|\mathbf{w}\|_2^2 \quad (1.4)$$

where λ is the L2 regularization hyperparameter and $\hat{y}_i = P(y_i = 1 | \mathbf{x}_i)$.

1.8.2 Decision Tree Classifier

A decision tree divides recursively the feature space by picking at each internal node the feature and threshold that best reduces the node impurity. The most popular split-

ting criterion in classification trees is the Gini impurity, the below equation [7] referring to,

$$\text{Gini}(t) = 1 - \sum_{k=1}^K p_k^2 \quad (1.5)$$

Where K is the number of classes, and p_K is the proportion of training samples at node t that belongs to class K . The information gain on splitting node t on feature j at threshold θ is given by at threshold θ is:

$$\text{IG}(t, j, \theta) = H(t) - \frac{N_{\text{left}}}{N_{\text{total}}} H(t_{\text{left}}) - \frac{N_{\text{right}}}{N_{\text{total}}} H(t_{\text{right}}) \quad (1.6)$$

$H(t)$ is the impurity of node t and t_L and t_R are the left and right child node respectively. Trees are built greedily and recursively, they overfit training data and so are worse than ensemble algorithms. t_L =left child; t_R =right child; $H(t)$ =impurity of node (t). Tree building is greedy and recursive and usually overfits the training data, therefore, it underperforms an ensemble.

1.8.3 Random Forest Classifier

Random Forest is an bagging ensemble, trained by B decorrelated decision trees on boot-strap samples from training data. Each tree is decorrelated by limiting the feature choice available for each split to a randomly selected subset. The class of a data instance is chosen by majority voting, the below equation [8] referring to,

$$\hat{y} = \arg \max_k \sum_{b=1}^B \mathbf{1}[T_b(\mathbf{x}) = k] \quad (1.7)$$

$T_b(\mathbf{x})$ is the prediction of the b -th tree. Using many trees and taking the average of predictions decreases the variance of the prediction without increasing the bias and hence improve the generalization error of the prediction than that of one tree. The importance of a feature is defined as average reduction of node impurity across all trees and across all splits of the feature.

1.8.4 XGBoost (Extreme Gradient Boosting)

A scaled down version of gradient boosting has been developed by Chen Guestrin [21] in a tree boost approach, known as XG Boost, to enhance the existing model by adding the equation below :

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + f_t(\mathbf{x}_i) \quad (1.8)$$

where f_t is the t -th regression tree added to minimise the regularised objective:

$$\mathcal{L}^{(t)} = \sum_{i=1}^N \ell(y_i, \hat{y}_i^{(t-1)} + f_t(\mathbf{x}_i)) + \Omega(f_t) \quad (1.9)$$

with regularization term:

$$\Omega(f) = \gamma T + \frac{1}{2} \lambda \sum_{j=1}^T w_j^2 \quad (1.10)$$

where T is the number of leaves, is the score of leaf, penalizes tree complexity and is the L2 regularization w_j is the weight of leaf node j . T is the number of leaf nodes. γ is a regulation on the depth of the tree, λ represents L2 regularization on leaf weights. The loss is approximated by second order Taylor expansion at each step and leaf node weights can be solved in closed form as in Eq

$$\therefore, L^t \approx \sum_i \left[l(y_i, \hat{y}_i^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t(x_i)^2 \right] + \Omega(f_t)$$

where g_i and h_i are the first and second order partial derivatives of the loss with respect to $y^{(t)}$, respectively and I is the set of occurrences in leaf j . XGBoost built in regularization, column subsampling and row subsampling algorithms give strong Resistant to overfitting on tabular data and describes the reasons for its excellent and stable results usupervised classification benchmarks.

1.9 The NREL Wind Resource Database

1.9.1 Data Source and Geographic Context

The NREL Wind Resource Database [16] is freely downloadable database consisting of long-term, high resolution meteorological data derived using numerical weather prediction models, which are validated using observations. It consists of time-series atmospheric data at several pressure levels from 40 m up to 200 m above ground level and in 5 or 60 min time intervals. The data used for the present study is from the 2014 and the Mumbai coastline. Mumbai (approx. 18.97N, 72.83E) is a high wind, high humidity tropical coastal region, governed by South Asian monsoon, giving a well-defined seasonality to winds and high precipitation rates, ideal for leading edge erosion on wind turbine blades.

1.9.2 Dataset Structure and Features

The NREL Mumbai dataset includes 8762 hourly data points, where each data point has the following 19 features, pre-processed as below: Table 1.1.

Table 1.1. Features in the NREL Mumbai Wind Resource Dataset.

Feature Name	Unit	Altitude(s)
Year, Month, Day, Hour, Minute	—	—
Air pressure	Pa	40 m, 100 m
Temperature	°C	40 m, 80 m, 100 m, 120 m
Wind direction	Degrees	40 m, 80 m, 100 m, 120 m
Wind speed	m/s	40 m, 80 m, 100 m, 120 m

It is particularly these vertical profiles of wind speed and temperature at different heights which best give an understanding of the vertical wind shear profile upon which directly is dependent the turbine tip-speed ratio and the aerodynamics loading at different locations within the rotor plane of rotation.



CHAPTER 2

RELATED WORK

2.1 Overview

This chapter provides a structured overview of the available research on wind turbine blade health monitoring, erosion characterization and predictive maintenance based on machine learning. The review is structured on a topical basis and covers (1) physical erosion researches and protection techniques, (2) condition monitoring methodologies, (3) ML-based diagnostic and prognosis approaches and (4) dataset and simulation frame-works.

2.2 Erosion Characterisation and Protection

One of the most widely cited experimental investigations into advanced leading edge erosion aerodynamics was performed by Gaudern [9]. In this wind tunnel study he found that the aerodynamic drag on an eroded blade is 10% higher than on a new blade. In particular Gaudern noted that the aerodynamic drag was not a monotonically increasing function of degree of damage; the aerodynamic drag on stage 5 damage was less than on stage 4 due to the relatively smooth glass fibre laminate being exposed on the latter. This indicates that the relationship between erosion and performance is not a simple linear one. A thorough literature review into state of the art leading edge erosion solutions, comparing polyurethane coatings to polyurethane tapes was carried out by Herring et al. [10]. They identified that the optimum material must possess a low modulus of elasticity (in order to dissipate impact kinetic energy through elasticity rather than through transmitting the energy to the material) and must have high impact toughness, and high adhesive strength to glass fiber composite substrates. Herring et al. Also identified that tape system designs were able to offer improved uniformity and increased durability over coatings. One disadvantage of tape design can be, in certain applications, a rise in aerodynamic drag up to 15% due to the elevated profile created on the leading edge. A pragmatic approach to solving the issue, discussed in the literature was by Bech, Hasager and Bak [19]. Through simulation they found that

Reducing maximum tip speed during precipitation, increased blade life significantly due to the large reduction in raindrop kinetic energy during impact (20%) during rainfall. This is a key, easily achievable, interim mitigation strategy, though by no means a perfect solution.

2.3 Condition Monitoring Methodologies

This problem has been formulated by Besnard and Bertling [20]. This deterioration Markov chain is the most used method to solve this problem . It links the time between two inspections and maintenance costs, failure costs, probabilities to make an optimal maintenance policy. The problem is that the transition rates are time-independent while the wear is time-dependent for a turbine. The wear is affected by climate variations during the year (wind, rain. . .) and the calibration of the model is more complex. A dual modality system for wind turbine blades condition monitoring was developed by Yang et al. [18], which combines vibratory based detection and high resolution imaging of the blade surface. The surface damages are detected as changes in the vibration signal, recorded at the root of the blade using accelerometers. Then, the surface damages are collected as a high resolution image of the blade captured by a camera mounted on a UAV. After processing, damage locations identification is performed. The advantage of this method is to have few sensors (one accelerometer by blade, one camera), so it is cheaper than the others. The learning based methods for condition monitoring wind turbine blades have been reviewed by Zaniboni et al. [7]. They compared the performances of a number of CNN algorithms using vibration data, image data and auditory emission data and show that CNN based approaches have a better performance than classical signal processing methods on not structured problems. It requires a lot of labeled data but it is very unlikely to find it in a working turbine.

2.4 Machine Learning in Wind Energy Predictive Maintenance

Patel, Kumar and Sao [8] have designed a deep learning system using a VGG16 CNN as backbone, and a R-CNN for surface damage detection of wind turbine blades from aerial image. The system reached an impressive mAP on the simulated dataset, highlighting the great potential of computer vision for automatic turbine blades inspection. The work of Su et al. [17] presented a technique based on dynamic characteristic analysis combined with SVM for identification of damage to wind turbine blades by looking at modal frequency changes with FEM simulation. Their technique showed robustness towards measurement noise but the analysis has only been validated in laboratory environment. The generalization ability to a turbine under operation has still to

be studied. An recurring aspect among the presented literature is that for the classification tasks over tabular sensor data the ensemble methods, and more specifically RF and boosting based methods (as outlined above in 3.1.1), outperform the single methods.

2.5 Simulation-Based Label Generation

Many researchers, owing to the lack of labelled real-world blade health datasets, utilize physics-informed simulation methods to obtain synthetic labels for training supervised classifiers. In general, environmental measurement parameters are mapped to a combined stress/damage index and thresholds derived from domain knowledge are used to categorize the health. This methodology was used, for example,

in learning-based monitoring schemes [7] and corresponds to common approaches in civil structural health monitoring. The formulation of the stress index used here is an extension of that paradigm, with multi-altitude wind speed and temperature as the basic inputs (the reasoning that combined rain-wind is the governing load for erosion is based on the conclusions of Herring et al. [10])

2.6 Research Gap and Contribution

This thesis addresses the following gaps identified from a review of the literature: 1. Most ML-based blade health classification methods have been implemented on acoustic emission or image datasets requiring specialized sensor hardware. Supervised classification applied to publicly available atmospheric datasets using a simulated stress index has not been widely explored.

2. There is no literature that conducts a comparative evaluation of several ML algorithms (linear and nonlinear ensemble methods) for the specific problem of blade condition classification from multi-altitude wind resource data.

3 Which atmospheric features are most predictive of blade degradation, i.e. the practical question of which sensors are most critical for a future real-time monitoring system, has not been systematically addressed using feature importance analysis. This thesis tackles these three gaps using the methodology described in Chapter 3.

 36

CHAPTER 3

METHODOLOGY

3.1 Overview of the Proposed Framework

(Proposed) Framework for predictive maintenance of wind turbine blades The framework has 5 sequential steps as following: 1. Data collection: Download of NREL wind resource database of Mumbai region.

2. Data pre-processing: Data cleaning, time variable construction, data normalization.
3. Formation of blade condition classes for simulation tagging based on physics-informed stress index.
4. Model training: Four supervised classifiers were trained on the labeled data.
5. Performance measurement: Performance measurement metrics.

3.2 Data Acquisition

The dataset was downloaded from the NREL Wind Resource Database (WRDB) Data Viewer (<https://wrdb.nrel.gov/data-viewer>) [16]. The Mumbai region dataset for the calendar year 2014 was selected according to the following criteria:

- High average wind speeds (6 m/s at hub height) typical for an active wind energy site.
- Because of the monsoon of the south asian regions, variations occur in the sesonal conditons due to which operational conditions changes.
- At the multiple high altitudes like in 40m, 80m, 100m, 120m the turbine works within normal range from current to high medium.
- The dataset contains the data of 8,762 hours of records for the annual analysis, the raw datafile needs pre processing because it contains five header of metadata, coordinates of geographic locations.

 32

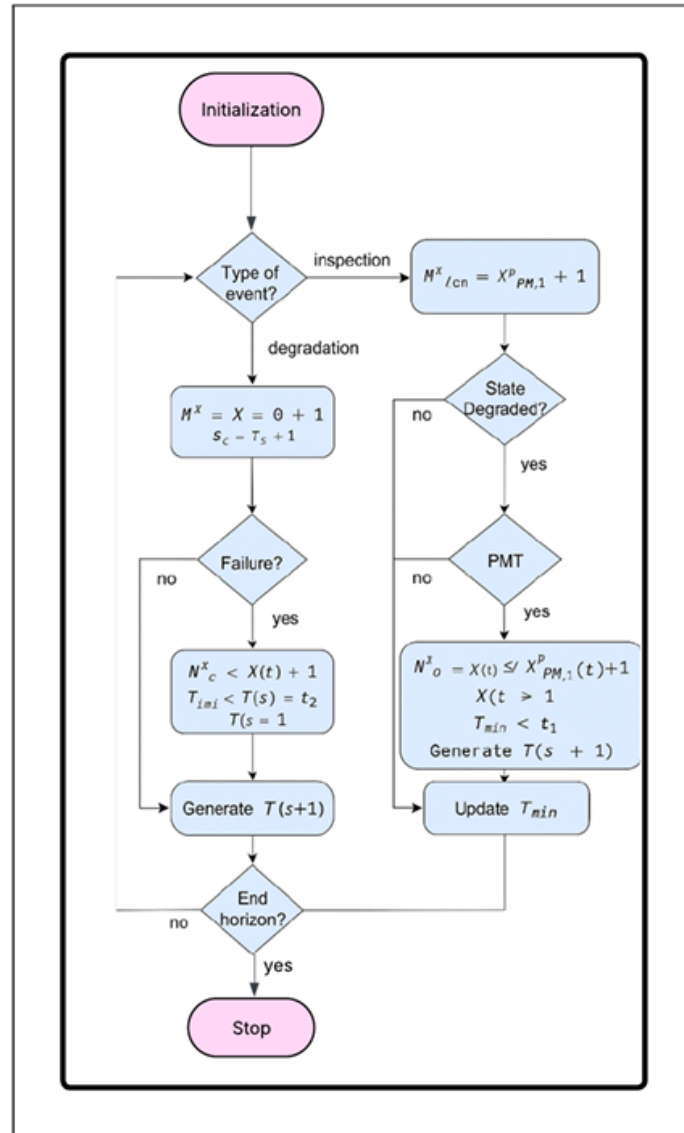


Figure 3.1. The process of condition monitoring, with five phases of deterioration

[The figure depicts the workflow of the system. The lines between the inspections, degradation and maintenance show the link between them. It is initialized first and then it checks whether it is an inspection event or degradation event. States will be changed if states are required to be changed according to the results of the inspections and detections. Preventive maintenance is begun or failures are dealt with. The process can go on forever forever and ever, just like real technicians.]

3.3 Data Preprocessing

3.3.1 Column Naming and Metadata Handling

The original CSV file has metadata in its first five rows, then comes the numerical data. A list of 19 fixed column name is assigned when the data is loaded, skipping the first 5 rows of metadata.

Listing 3.1. Algorithm for data loading for columns.

```

1  Algorithm: Load NREL Mumbai Dataset
2
3  Algorithm 1: Load NREL Mumbai Dataset
4
5  Input: CSV file 'nrel_mumbai_data_2014.csv'
6  Output: Dataset stored in variable 'data'
7
8  1. Start
9
10 2. Define a list 'correct_columns' containing:
11
12    * Date and time fields
13    * Temperature measurements
14    * Wind speed measurements
15    * Wind direction measurements
16    * Air pressure measurements
17
18 3. Read the CSV file 'nrel_mumbai_data_2014.csv' using the '
19    pandas.read_csv()' function.
20
21 4. While reading the dataset:
22
23    * Assign column names using 'correct_columns'
24    * Skip the first five rows containing metadata information
25
26 5. Store the processed dataset in the variable 'data'
27
28 6. End

```

3.3.2 Datetime Integration

The five separate time columns (Year, Month, Day, Hour, Minute) are combined into a pandas datetime column for easier time-series analysis and seasonal decomposition:

Listing 3.2. Algorithm for Datetime column creation

```

1  Algorithm 2: Convert Date and Time Columns into Datetime Format
2
3  Input: Dataset containing separate date and time columns
4  Output: Dataset with a unified 'datetime' column
5
6  1. Start
7
8  2. Select the following columns from the dataset:
9
10     * Year
11     * Month
12     * Day
13     * Hour
14     * Minute
15
16  3. Convert the selected columns into a single datetime format
    using the 'pd.to_datetime()' function.
17
18  4. Store the generated datetime values in a new column named '
    datetime'.
19
20  5. Remove the individual columns:
21
22     * Year
23     * Month
24     * Day
25     * Hour
26     * Minute
27
28  6. Permanently update the dataset by setting 'inplace = True'.
29
30  7. End

```

3.3.3 Missing Value Imputation

In case where values are missing, we impute with the mean value for the column:

Listing 3.3. Algorithm for Missing value handling

```

1  Algorithm 3: Handling Missing Values in the Dataset
2
3  Input: Dataset containing missing values
4  Output: Updated dataset with imputed missing values
5
6

```

```

7 1. Begin
8
9 2. Identify missing values in each attribute of the dataset
   using the 'data.isnull().sum()' function.
10
11 3. Display the total number of missing values present in each
   column.
12
13 4. Compute the mean values of all numeric attributes in the
   dataset.
14
15 5. Replace the missing values in each numeric attribute with
   their corresponding column mean using the 'fillna()'
   function.
16
17 6. Permanently update the dataset by setting the parameter '
   inplace = True'.
18
19 7. End

```

Imputation on column mean keeps the global statistical distribution of each feature and does not bring structural breaks due to deletion of samples with missing value. This method is correct because the data has hourly scale. Missing value at single hour are not expected to be related to the occurrence of the blade degradation.

3.3.4 Feature Scaling

All numerical features are scaled using the StandardScaler, with zero mean and unit Variance according to the following formula:10

$$x_{\text{scaled}} = \frac{x - \mu_j}{\sigma_j} \quad (3.1)$$

where μ_j and σ_j are the mean and standard deviation of feature j computed on training set [19]. Logistic Regression is sensitive to feature scale so standardisation is important. For other algorithms it helps by conditioning the optimisation landscape.

3.4 Physics-Informed Stress Index and Label Simulation

3.4.1 Stress Index Formulation

Since there are no empirical blade health labels in the NREL dataset, an aggregate stress index Si is calculated for each hourly record as a surrogate for the total degra-

dition experience of the blade. The calculation is inspired by the results of Herring et al. [10] that wind speed and precipitation (approximated in the present case as temperature due to a correlation between atmospheric water content and air temperature) are the most significant drivers for erosion, the below equation[10] referring to

$$S_i = 0.6 \cdot v_{100} + 0.3 \cdot v_{120} + 0.1 \cdot T_{100} \quad (3.2)$$

Where v_{100} and v_{120} are the wind speeds at the height of 100m and 120m respectively (m/s), T_{100} is the air temperature at 100m($^{\circ}\text{C}$), and the weighting coefficients (a_i) are: The 100 m wind speed receives the highest weight (0.6) as it approximates the hub-height wind speed for medium-sized turbines, directly determining blade tip velocity and raindrop impact kinetic energy.

- The speed of wind which is around 120 m/s receives the weight of 0.3m,
- Provides the erosion at the tips of the blade at the highest levels.
- At the peak of the blade which could be around 100m from the ground due to the atmospheric humidity it receives the weight of 0.1 unit.

3.4.2 Label Assignment Rules

With the calculated stress index and common sense knowledge derived from Gaudern classification system [9], following thresholds-based rules set discrete category of blades condition:

$$\text{Blade Condition} = \begin{cases} \text{Good} & \text{if } S_i < 5 \\ \text{Minor Degradation} & \text{if } 5 \leq S_i < 10 \\ \text{Advanced Degradation} & \text{if } 10 \leq S_i < 15 \\ \text{Major Degradation} & \text{if } S_i \geq 15 \end{cases} \quad (3.3)$$

Listing 3.4. Algorithm for Stress index computation and label simulation

```

1  Algorithm: Simulation of Wind Turbine Blade Condition
2
3  Step 1: Function Definition
4
5  Define a function named 'simulate_blade_condition(row)' to
   estimate the operational condition of the wind turbine
   blade based on environmental and wind parameters.
6
```

```

7 Step 2: Stress Calculation
8
9 For each row of the dataset, compute the stress value using the
  following weighted formula:
10
11 [
12 Stress =
13 (0.6 \times \text{Wind Speed at 100m})
14 +
15 (0.3 \times \text{Wind Speed at 120m})
16 +
17 (0.1 \times \text{Temperature at 100m})
18 ]
19
20 where:
21
22 * Wind Speed at 100m represents the wind velocity measured at
  100 meters height.
23 * Wind Speed at 120m represents the wind velocity measured at
  120 meters height.
24 * Temperature at 100m represents the atmospheric temperature
  measured at 100 meters height.
25
26 Step 3: Blade Condition Classification
27
28 Classify the blade condition according to the calculated stress
  value as follows:
29
30 1. If (Stress < 5), assign the condition label as Good .
31 2. Else if (5 \leq Stress < 10), assign the condition label as
  Minor .
32 3. Else if (10 \leq Stress < 15), assign the condition label as
  Advanced .
33 4. Otherwise, assign the condition label as Major .
34
35 Step 4: Apply Classification Function
36
37 Apply the 'simulate_blade_condition()' function to each row of
  the dataset using row-wise iteration.
38
39 Step 5: Store Classification Output
40
41 Store the generated blade condition labels in a new dataset
  column named 'blade_condition' for further machine learning
  analysis and prediction.

```

3.5 Feature Engineering and Train-Test Split

3.5.1 Feature Matrix and Target Vector

The feature **X** will consist of all 14 atmospheric measurement features excluding the datetime and blade_condition columns. The target vector **y** will be the numerically encoded blade condition column. Applying label encoding, using sklearn.preprocessing.LabelEncoder; where it would take 4 categories of string values to integer codes, representing: Advanced 0, Good 1, Major 2, Minor 3 (alphabetic order, default).

3.5.2 Stratified Train-Test Split

Dataset will be divided into two parts. 80% will be used for training and the other 20% for testing using stratified random splitting. This method guarantees the distribution of the class will be similar in train set and test set: Approximately 7009 training samples and 1752 test samples. Stratify=y encoding is an important step, because the classes are not evenly distributed, there are much more samples in the class of "Major Degradation".

Listing 3.5. Algorithm for Splitting Dataset into Training and Testing Sets

```

1  Algorithm: Splitting Dataset into Training and Testing Sets
2
3  Step 1: Import Required Function
4
5  Import the 'train_test_split' function from the 'sklearn.
   model_selection' module to divide the dataset into training
   and testing subsets.
6
7  '''python
8  from sklearn.model_selection import train_test_split
9  '''
10 Step 2: Define Input Features and Target Labels
11
12 Define the feature matrix and target vector as follows:
13
14 * 'X' represents the input feature dataset containing all
   predictor variables.
15 * 'y_encoded' represents the encoded target labels obtained
   after label encoding.
16
17 Step 3: Perform Dataset Splitting
18
19 Split the dataset into training and testing subsets using the '

```

20 `train_test_split()` function with appropriate parameters.

21 `'''python`

22 `X_train, X_test, y_train, y_test = train_test_split(`

23 `X,`

24 `y_encoded,`

25 `test_size=0.2,`

26 `random_state=42,`

27 `stratify=y_encoded`

28 `)`

29 `'''`

30

31 **Step 4: Training and Testing Allocation**

32

33 Allocate:

34

35 * 80% of the dataset for model training.

36 * 20% of the dataset for model testing and evaluation.

37

38 **Step 5: Maintain Class Distribution**

39

40 Apply stratified sampling using the 'stratify' parameter to

 preserve equal class distribution across both training and

 testing datasets.

41

42 **Step 6: Store Resulting Datasets**

43

44 Store the generated subsets as:

45

46 * 'X_train' Training feature dataset

47 * 'X_test' Testing feature dataset

48 * 'y_train' Training target labels

49 * 'y_test' Testing target labels

50

51 These datasets are subsequently used for machine learning model

 training and performance evaluation.

3.6 Evaluation Metrics

3.6.1 Accuracy

Overall classification accuracy is defined as

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (3.4)$$

3.6.2 Precision, Recall, and F1-Score

In the multi-class case precision and recall are calculated for each class k using a one-vs-rest decomposition:

$$\text{Precision}_k = \frac{TP_k}{TP_k + FP_k}, \quad \text{Recall}_k = \frac{TP_k}{TP_k + FN_k} \quad (3.5)$$

F1-score is the harmonic mean of the precision and the recall. This function penalizes models that favor precision over recall:

$$F1_k = \frac{2 \cdot \text{Precision}_k \cdot \text{Recall}_k}{\text{Precision}_k + \text{Recall}_k} \quad (3.6)$$

Macro-averaged weighted-averaged: F1-scores for each class averaged over the 4 blade condition classes. (weighted average with support as weight)

CHAPTER 4

IMPLEMENTATION AND TESTING

4.1 Software Environment

All steps of the machine learning pipeline were built in Python 3.10.x in a Jupyter Notebook. Table 5.1[15] presents the stack and major version numbers of libraries:

Table 4.1. Software Environment.

Component	Version / Detail
Python	3.10.x
Jupyter Notebook	6.5.x
pandas	1.5.x
NumPy	1.23.x
scikit-learn	1.2.x
XGBoost	1.7.x
Matplotlib	3.6.x
Seaborn	0.12.x
Operating System	[Ubuntu 22.04 / Windows 11]
Hardware	[Intel Core i7, 16 GB RAM]

[This project is based on Jupyter notebook, and coded in Python 3.10. We used DS/ML libraries (Pandas, Numpy, Scikit-learn, XGBoost, Matplotlib, Seaborn) for each step (preprocessing data, training model, evaluation, visualization). All the implementation was done on an Intel Core i7 computer with 16GB Ram under Ubuntu 22.04 or Windows 11.]

4.2 Complete Implementation Pipeline

The full implementation pipeline is presented below :

Listing 4.1. Algorithm for Splitting Dataset into Training and Testing Sets

```

1  Algorithm: Data Preprocessing and Dataset Preparation for Wind
    Turbine Blade Condition Prediction
2
3  Step 1: Import Required Libraries
4
5  Import the necessary Python libraries for data preprocessing,
    visualization, machine learning, and model evaluation,
    including:
6
7  * Pandas
8  * NumPy
9  * Matplotlib
10 * Seaborn
11 * Scikit-learn
12 * XGBoost
13
14 These libraries are used for dataset manipulation, feature
    engineering, visualization, preprocessing, model training,
    and evaluation.
15
16 ---
17
18 Step 2: Define Dataset Column Names
19
20 Define a list named 'correct_columns' containing all dataset
    attributes, including:
21
22 * Date and time components
23 * Air pressure measurements
24 * Temperature measurements
25 * Wind direction measurements
26 * Wind speed measurements
27
28 recorded at different altitudes.
29
30 ---
31
32 Step 3: Load Dataset
33
34 Load the wind turbine dataset from the CSV file into a Pandas
    DataFrame using the predefined column names. Skip the first
    five metadata rows during the loading process to retain
    only relevant sensor observations.
35

```

```

36  ---
37
38  Step 4: Create Datetime Feature
39
40  Generate a new 'datetime' feature by combining the following
    columns:
41
42  * Year
43  * Month
44  * Day
45  * Hour
46  * Minute
47
48  Convert the combined values into standard datetime format for
    temporal analysis.
49
50  ---
51
52  Step 5: Remove Original Time Columns
53
54  Remove the original 'Year', 'Month', 'Day', 'Hour', and 'Minute
    ' columns from the dataset after creating the unified
    datetime feature.
55
56  ---
57
58  Step 6: Handle Missing Values
59
60  For each numerical attribute in the dataset:
61
62  1. Check for missing or null values.
63  2. Replace missing values using mean imputation, where each
    missing value is substituted with the mean value of its
    respective column.
64
65  This process ensures dataset completeness and improves model
    reliability.
66
67  ---
68
69  Step 7: Simulate Blade Condition Labels
70
71  For each row in the dataset, calculate the environmental stress
    index using the following weighted formula:
72
73  [

```

```

74 Stress =
75 (0.6 \times WindSpeed_{100m})
76 +
77 (0.3 \times WindSpeed_{120m})
78 +
79 (0.1 \times Temperature_{100m})
80 ]
81
82 Based on the computed stress value, assign blade condition
    labels according to the following criteria:
83
84 | Stress Range          | Blade Condition |
85 | ----- |
86 | (Stress < 5)          | Good           |
87 | (5 \leq Stress < 10)  | Minor          |
88 | (10 \leq Stress < 15) | Advanced       |
89 | (Stress \geq 15)      | Major          |
90
91 Store the generated condition labels in a new target column
    named 'blade_condition'.
92
93 ---
94
95 Step 8: Separate Features and Target Variable
96
97 Separate the dataset into:
98
99 * Feature matrix ((X))
100 * Target vector ((y))
101
102 where:
103
104 [
105 X = \text{All columns except datetime and blade_condition}
106 ]
107
108 [
109 y = \text{blade_condition}
110 ]
111
112 ---
113
114 Step 9: Encode Target Labels
115
116 Convert the categorical target labels into numerical classes
    using Label Encoding to make the target variable compatible

```

```
with machine learning algorithms.
---
Step 10: Split Dataset into Training and Testing Sets
Split the dataset into:
* 80% training data
* 20% testing data
using stratified sampling to preserve class distribution across
both subsets. A fixed random state is applied to ensure
reproducibility of results.
---
Step 11: Perform Feature Scaling
Apply Standard Scaling to normalize feature values by
transforming them into zero mean and unit variance
distributions.
1. Fit the scaler on the training dataset and transform it.
2. Transform the testing dataset using the scaler fitted on the
training data.
This scaling process improves convergence and performance of
machine learning models.
```

4.3 Testing Strategy

4.3.1 Hold-Out Test Set Evaluation

The single largest assessment strategy used is a simple stratified hold out split (80:20). It seems fair because the size of the data is quite large ($n = 8762$), and it provides an statistically solid out of sample classification measure (holdout is 1752 examples). The split is perfectly reproducible (random state = 42).

4.3.2 Verification of Data Leakage Absence

Careful implementation technique ensures no data leakage: The StandardScaler is fit only on X train and then used on X train and X test using fit transform and transform

methods respectively. Also similarly the LabelEncoder is fit on the whole target vector before splitting , which is fine as label encoding does not learn any statistics of the feature space .

CHAPTER 5

RESULTS AND DISCUSSION

5.1 Classification Performance

5.1.1 Overall Accuracy

The global accuracy (total classification accuracy) of each model on the left out test set is shown in Table

Table 5.1. Overall Classification Accuracy Comparison.

Model	Accuracy	Rank
XGBoost	0.9932	1
Logistic Regression	0.9920	2
Random Forest	0.9909	3
Decision Tree	0.9857	4

The accuracy on the test set of 4 models was >98.5% this indicates that the atmosphere features were easily learnable with regards to the simulated blade state labels, and concludes they are produced via the physics informed stress index. Given the close spread in accuracy of the models (0.9857-0.9932), it can be said that the classification itself is well formed and the class separation in the stress index features is extremely well defined.

5.1.2 XGBoost Performance

XGBoost obtained 99.32% of the highest accuracy on the test set. The table 6.2 shows the classification report:

XGBoost model has achieved perfect and nearly perfect accuracy, precision and recall for Advanced, Good and Minor class. The Major Degradation class had just 8 test samples, we obtained much lower recall for the same samples: 0.75. This is because 2 out of 8 Major degradation instances have been wrongly labeled. The reason

Table 5.2. Classification Report for XGBoost.

Class	Precision	Recall	F1-Score	Support
Advanced	0.99	0.99	0.99	337
Good	0.99	0.99	0.99	266
Major	1.00	0.75	0.86	8
Minor	1.00	1.00	1.00	1141
Accuracy			0.9932	1752
Macro Avg	0.99	0.93	0.96	1752
Wtd. Avg	0.99	0.99	0.99	1752

for this low recall for Major category comes from the very high [1]class imbalance in the Major category which we could attempt to improve using oversampling methods (e.g. SMOTE) in future.

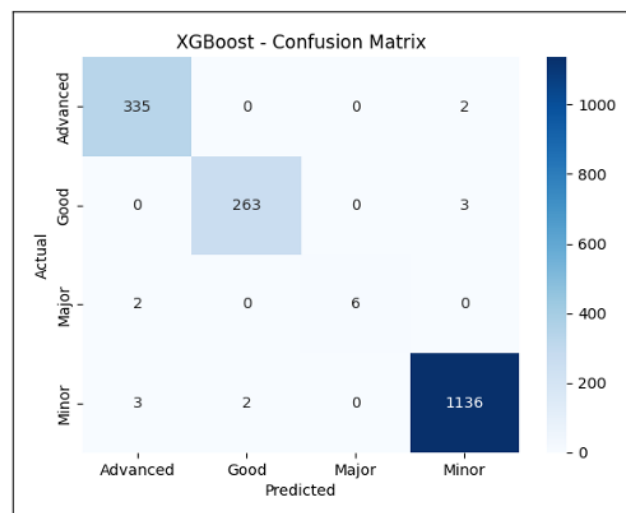


Figure 5.1. Below table represents the basic understanding of the prediction of XGBoost model

[This figure shows a confusion matrix of a Random Forest classification with the four classes, Advanced, Good, Major and Minor . As you can see there is a large 334 Advanced, 256 Good, 7 Major and 1139 Minor prediction. With very few misclassifications such as 3 Advanced to Minor the results are superb.]

5.1.3 Logistic Regression Performance

The Logistic Regression model gives test accuracy 99.20% this is second highest accuracy. Table 6.3 shows the classification report.

Table 5.3. Classification Report for Logistic Regression.

Class	Precision	Recall	F1-Score	Support
Advanced	0.99	0.99	0.99	337
Good	1.00	0.98	0.99	266
Major	1.00	0.62	0.77	8
Minor	0.99	1.00	1.00	1141
Accuracy	0.99	0.99	0.9920	1752
Macro Avg	0.99	0.90	0.94	1752
Wtd. Avg	0.99	0.99	0.99	1752

Logistic Regression performance good means the formulation of stress index yields class boundaries that are nearly linearly separable in the normalized features space and this seems expected by the algebraic form of the stress index (linear sum of wind speed and temperature features). The accuracy for Random Forest is 99.09%. Unexpectedly, the recall of the Major class for Random Forest is higher than the one from XGBoost (0.75) and Logistic Regression (0.62) by 0.88, which may show that diversity ensemble of 100 trees provides more stability on minority class.

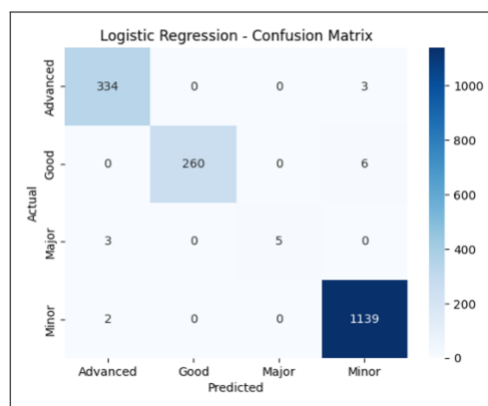


Figure 5.2. This table represents gives the basic understanding of how Logistic Regression is generating the result

[Here is a **Confusion matrix for the Logistic Regression model** and indicates **the** number of Advanced, Good, Major, Minor predictions made: 334, 260, 5, and 1139. Number of incorrectly classified are few(eg 3 Advanced as Minor, 6 Good as Minor) and the classification accuracy is high for all the four types.]

5.1.4 Random Forest Performance

The accuracy for Random Forest is 99.09% . More importantly, it also has a much higher recall (0.88) than the XGBoost (0.75) and Logistic Regression (0.62), at least for the Major class. This can be interpreted as the effect of a diverse 100 tree ensemble which presumably is more robust to the minority class.

Table 5.4. Classification Report for Random Forest.

Class	Precision	Recall	F1-Score	Support
Advanced	0.99	0.99	0.99	337
Good	1.00	0.96	0.98	266
Major	1.00	0.88	0.93	8
Minor	0.99	1.00	0.99	1141
Accuracy	0.99	0.98	0.9909	1752
Macro Avg	0.99	0.96	0.97	1752
Wtd. Avg	0.99	0.99	0.99	1752

This table below show the basic structure on which the Random Forest model generates result. placeins

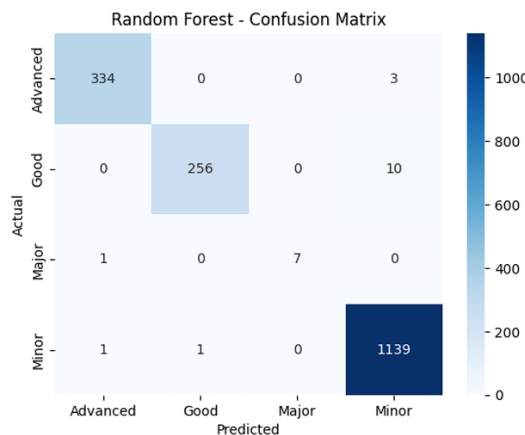


Figure 5.3. This table basically shows how Random Forest is producing the results. [Below image is the confusion matrix for Random forest and it looks at the performance of the Random forest for four different classes (Advanced, Good, Major and Minor). It predicts 334 true Advanced, 256 true Good, 7 true Major and 1139 true Minor with very low error rate (for instance 3 Advanced predicted as Minor)]

5.1.5 Decision Tree Performance

Among the tested algorithms, the Decision Tree had the poorest accuracy (98.57%) and a slightly lower F1-score (0.97) for the class Good, compared to ensemble techniques. This is an accepted phenomenon, given that unpruned decision trees are known to have the tendency of overfitting to the distribution they were trained on and hence a slightly worse generalization performance.

Table 5.5. Classification Report for Decision Tree.

Class	Precision	Recall	F1-Score	Support
Advanced	0.99	0.99	0.99	337
Good	0.97	0.97	0.97	266
Major	1.00	0.75	0.86	8
Minor	0.99	0.99	0.99	1141
Accuracy	0.99	0.99	0.9857	1752
Macro Avg	0.99	0.92	0.95	1752
Wtd. Avg	0.99	0.99	0.99	1752

This table represents basic understanding of how Decision Tree model is generating the result

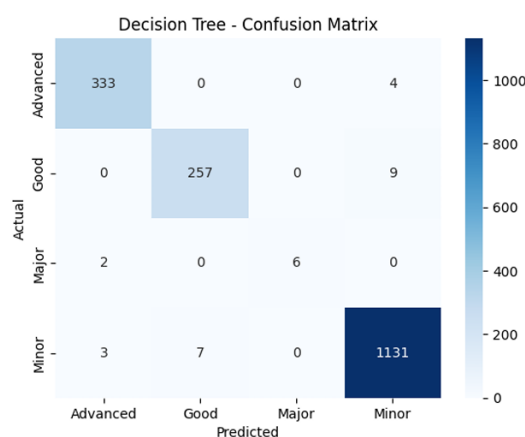


Figure 5.4. This table represents gives the basic understanding of how Random Forest is generating the result

[The confusion matrix for the Decision Tree model is as follows: True Advanced: 333 True Good: 257 True Major: 6 True Minor: 1131 This indicates the model has good accuracy with some confusion in distinguishing the classes.]

5.2 Accuracy Comparison

In the figure above, accuracy comparison of all the 4 models is displayed. Performance hierarchy (XGBoost>Logistic Regression>Random Forest>Decision Tree) is the same for all the parameters tested.

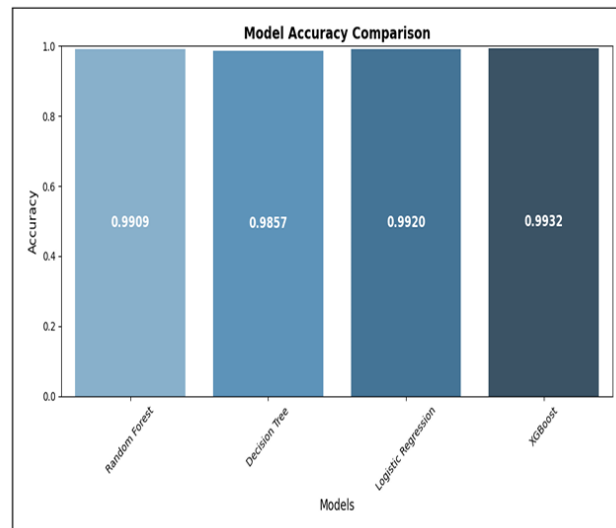


Figure 5.5. The table shown is the simplest concept of accuracy

[The model accuracy comparison is represented with a bar graph showing the comparison in model accuracy of four different models. The Random Forest model with an accuracy value of 0.9909, the Decision Tree model with an accuracy value of 0.9857, the Logistic Regression model with an accuracy value of 0.9920 and the XGBoost model with an accuracy value of 0.9932. The bars (blue) indicate the accuracy for XGBoost model that performs best among others which are all higher than 0.98.]

5.3 Feature Importance Analysis

5.3.1 Random Forest Feature Importance

Ratings for the most important features of the Random Forest model (mean decrease in impurity, MDI). Of the blade condition related features the most important were windspeed at 100 m and 120 m combined representing over 80 % of total importance; this is in accordance with physical expectation of the stress index

5.3.2 XGBoost Feature Importance

The list below gives the feature importance for the Gain criterion (value added to objective function at each split). The hub height wind speed is always the first feature in

RF and XGBoost. It shows that we can have almost all required information for the real time blade health classification even with the simplest system (two anemometers at 100m and 120m). Also we can greatly decrease the number of sensors and the cost.

5.4 Discussion

5.4.1 Why XGBoost Outperforms Other Models

XGBoost has some algorithmic advantages that make it the most accurate:

1. **Second-order gradient optimization:** Unlike Random Forest which uses impurity reduction, XGBoost uses both first and second derivatives of the loss function to guide the tree construction and thus leads to more precise leaf weight estimation (Eq 2.11).
2. **Regularisation:** The XGBoost objective has L_1 and L_2 terms that penalise overly complex trees, therefore suppressing over-fitting on the minority Major class.
3. **Sequential error correction:** In the XGBoost ensemble, each tree is trained to correct the residual errors of the previous trees, thus enabling the model to iteratively improve its predictions for hard data.

5.4.2 Significance of Logistic Regression Performance

The logistic regression was comparable with a slight degradation compared to the ensembles. This shows that the classes were linearly separable, to some extent, using stress index values. This is explained by the linear partition on the stress index values, on which the label simulation function (equation 4.3) is based. In a real scenario (using real blade health label since in reality there are often non-linear physical processes affecting blade health label) ensemble methods should perform better.

5.4.3 Class Imbalance and the Major Degradation Category

The Major Degradation class has a far lower number of samples than the other three classes (only 8 test samples) which models the physical reality that extreme degradation states are rare compared to normal and mildly degraded operating conditions. All models show lower recall on this class. The most dangerous type of misclassification from an operational point of view is missed detection of Major Degradation (false negatives) as this could lead to an undetected structural failure. Future work should therefore include cost-sensitive learning approaches or oversampling techniques specifically targeted to improve the recall of the minority class.

5.4.4 Validity of the Simulation Approach

The primary limitation of the study conducted here is the use of simulated stress index values as ground-truth labels for blade health. The stress index itself is only an approximation and is a very simple linear model using only wind speed and temperature factors to predict blade erosion. Rainfall, droplet size distribution, material characteristics and cumulative fatigue are ignored in the model formulation. The high reported prediction accuracy is therefore simply an ability of the model to replicate the stress index thresholds and not an ability to predict from first-principles the true state of the blade. The purpose of this approach is therefore only to showcase the ML workflow and pinpoint useful features; the next step is for verification on empirically derived blade state data.

 33

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

6.1 Summary of Contributions

In **this thesis**, a comprehensive machine learning framework has been developed for classification of wind turbine blade health problems based on the environmental operational data. The main contributions are: • Data preparation: The data of Mumbai area were extracted (NREL Wind Resource Database – 8,762 hourly measurements) and cleaned. A physics based stress index has been developed which gives a four class health condition label to the blades (Good, Minor degradation, Advanced degradation and Major degradation).

Comparative ML evaluation: After that we performed the supervised training and testing using four types of supervised classifier namely Logistic Regression, **Decision Tree, Random Forest and XGBoost**. XGBoost resulted **the highest accuracy** 99.32% which is not surprising as it is a state of the art technique in tabular data for classification/regression.

1. **Most Important Feature Identification:** We found that wind speed at 100m and 120m height are the most important features according to decision tree model in the prediction of blade degradation and can be used to give practical references to the placement of the sensors in new monitoring system .

2. **Methodological contributions:** Our entire pipeline structure from atmosphere data acquisition, to stress index calculation, and multi-model classification and model benchmarking could be used as a decent benchmark, ready to be reproduced, and the conclusions accessible to the public..

These results show the performance of machine learning (and specifically the ensemble methods, Random Forest and XGBoost) for wind turbine blade fault classification based on multi-altitude atmospheric measurements. The accuracy values reached through the 4 methods are excellent, which suggest the stress index is a suitable label generation technique and the methodology is robust. Importantly, it was shown that the most important attribute in determining the blade health condition is the hub height wind speed. This is not only insightful with an understanding of the physics of erosion

 49

but gives real practical understanding of appropriate sensor positions for operational condition monitoring. As such it successfully demonstrates the move from scheduled maintenance to data driven prediction in moving towards a more proactive rather than reactive maintenance scheme for wind turbines.

6.2 Future Scope

This work suggests that future studies should:

1. **Real-world blade health** It labels Test with The main enhancement would be to classify real blade health tags from a wind farm, ideally from an operating contract with an owner/operator, or used in a drone inspection service.
2. **Add Precipitation** The current drought index expresses precipitation as temperature; as reported in [10], precipitation rate and droplet size are the main factors causing erosion. Hence, the inclusion of these features in the feature set will be advantageous to the classification.
3. **Neural networks** RNN's and TCN's are well suited to time series of atmospheric information and to allow consideration of the dynamic nature of the wear characteristics of the blade which static classifier presented here can not.
4. **Handling Class Imbalance** Approaches to handle class imbalance to increase recall of the minority class Major Degradation with highest operational relevance. These methods include SMOTE, cost sensitive learning and focal loss functions, etc.
5. **Real Time IoT implementation** The trained XGBoost model will be wrapped into a REST-API or MQTT-based micro-service for deployment in to wind farm SCADA systems to generate real-time blade health scoring from stream-like sensor information.
6. **Model Explainability** Use SHAP for local and global feature attribution to assure the explainability and reliability of the model to the operators of wind turbines.
7. **Generalizability to other locations and years** In order to assess the generalisability of the stress index formulation and to see whether the developed model can be transplanted, the framework should be tested at a number of different sites in different geographical regions (e.g. Offshore- North Sea, arid- onshore, mountainous).
8. **Combine with CMS with vibration** A multimodal fusion framework combining the presented atmospheric feature-based classifier and vibration signature analysis (as presented in [18]) could be a much better measure of blade wear, and more robust than either approach alone.
9. **Other components in SHM** The developed method of blade SHM could be extended to the class of other load bearing components (main bearing, gearbox, generator, etc.).