Pontifícia Universidade Católica
do Rio de Janeiro

Fernanda Carvalho Marinho Filizzola

Crack Detection in Wind Turbines Blades and Plastic Strain Accumulation Prediction based on Machine Learning

Dissertação de Mestrado

Dissertation presented to the Programa de Pós-graduação em Engenharia Mecânica of PUC-Rio in partial fulfillment of the requirements for the degree of Mestre em Engenharia Mecânica.

Advisor : Prof. Florian Alain Yannick Pradelle
Co-advisor: Dra. Paula Aida Sesini
Co-advisor: Prof. Helon Vicente Hultmann Ayala

Rio de Janeiro
October 2024

Pontifícia Universidade Católica
do Rio de Janeiro

Fernanda Carvalho Marinho Filizzola

Crack Detection in Wind Turbines Blades and Plastic Strain Accumulation Prediction based on Machine Learning

Dissertation presented to the Programa de Pós-graduação em Engenharia Mecânica of PUC-Rio in partial fulfillment of the requirements for the degree of Mestre em Engenharia Mecânica. Approved by the Examination Committee:

Prof. Florian Alain Yannick Pradelle
Advisor
Departamento de Engenharia Mecânica – PUC-Rio

Dra. Paula Aida Sesini
Co-advisor
Departamento de Engenharia Mecânica – PUC-Rio

Prof. Helon Vicente Hultmann Ayala
Co-advisor
PUC-PR

Prof. Renato Bichara Vieira
Departamento de Engenharia Mecânica – PUC-Rio

Dr. Sinvaldo Rodrigues Moreno
EDF Renewables do Brasil

Rio de Janeiro, October the 24th, 2024

Ficha Catalográfica

Filizzola, Fernanda Carvalho Marinho

Crack Detection in wind turbines blades and plastic strain accumulation prediction based on machine learning / Fernanda Carvalho Marinho Filizzola ; advisor: Florian Alain Yannick Pradelle ; co-advisors: Paula Aida Sesini, Helon Vicente Hultmann Ayala. – 2024.

80 f. : il. color. ; 30 cm

Dissertação (mestrado)–Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Engenharia Mecânica, 2024.

Inclui bibliografia

1. Engenharia Mecânica – Teses. 2. Turbina Eólica. 3. Detecção de falhas. 4. Monitoramento estrutural. 5. Machine learning. 6. Deep learning. I. Pradelle, Florian Alain Yannick. II. Aida Sesini, Paula. III. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Engenharia Mecânica. IV. Ayala, Helon Vicente Hultmann. V. Título.

CDD: 621

To my father who never saw this adventure

Acknowledgments

I would like to thank God and the Virgin Mary for giving me strength and encouragement throughout all the challenging moments of completing this dissertation.

To my father, for teaching me to believe in myself and for loving me so profoundly that I still feel it, even in your absence from this world.

To my mother, for teaching me strength and the importance of pursuing my goals, and to my grandmother, for being a second mother to me.

To my husband, for his support and encouragement. Thank you for believing in me, for pushing me to persevere, and for standing by my side through every challenge.

To my advisors, Florian, Helon, and Paula Sesini, thank you for your guidance, support, and encouragement throughout this journey. I am profoundly grateful for all the time and effort you have invested in my success.

To PUC-RIO, thank you for granting me a scholarship that was essential for the completion of this work.

To all my friends and family, your presence and belief in me have been invaluable, and I am deeply grateful for each and every one of you.

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

Abstract

Filizzola, Fernanda Carvalho Marinho; Pradelle, Florian Alain Yannick (Advisor); Aida Sesini, Paula (Co-Advisor); Ayala, Helon Vicente Hultmann (Co-Advisor). **Crack Detection in Wind Turbines Blades and Plastic Strain Accumulation Prediction based on Machine Learning**. Rio de Janeiro, 2024. 80p. Dissertação de Mestrado – Departamento de Mechanical Engineering, Pontifícia Universidade Católica do Rio de Janeiro.

The background machine learning approaches for wind turbine blade monitoring rely mostly on complete and labeled datasets, which are costly and often impractical to obtain. Additionally, in material science, most methods depend on expensive experimental data, with limited exploration of data augmentation techniques to reduce the cost and effort of model training. The objective of this research addresses these significant gaps by exploring one-class classification for anomaly detection in wind turbine blades and by developing methods to augment existing data for more cost-effective model training in material science. The methods applied in this work for anomaly detection include One Class Support Vector Machine (OCSVM), Support Vector Data Descriptio (SVDD), and Long Short-Term Memory (LSTM) autoencoders. For augmenting experimental material data, noise addition and image manipulation techniques were used. For anomaly detection, the LSTM autoencoder achieved an accuracy of 97.4% with approximately 100% recall, while OCSVM achieved 89% accuracy and 97% recall. OCSVM was deemed more suitable due to lower training costs and similar performance. The results for experimental data augmentation showed a 20% improvement over previously trained models, with the augmentation technique significantly enhancing performance, especially in models trained with data from different experimental samples. In conclusion, this research demonstrates the effectiveness of one-class classification for anomaly detection in wind turbine blades and highlights the benefits of data augmentation techniques for cost-effective model training in material science.

Keywords

Mechanical Engineering; Wind Turbine; Blades; Structural Health Monitoring; Deep Learning; Failure Detection.

Resumo

Filizzola, Fernanda Carvalho Marinho; Pradelle, Florian Alain Yannick; Aida Sesini, Paula; Ayala, Helon Vicente Hultmann. **Detecção de trincas em pás de aerogeradores e previsão de acúmulo de deformação plástica com base no aprendizado de máquina**. Rio de Janeiro, 2024. 80p. Dissertação de Mestrado – Departamento de Mechanical Engineering, Pontifícia Universidade Católica do Rio de Janeiro.

Os métodos de aprendizado de máquina para monitoramento de pás de turbinas eólicas dependem principalmente de conjuntos de dados completos e rotulados, que são caros e muitas vezes impraticáveis de obter. Além disso, na ciência dos materiais, a maioria dos métodos depende de dados experimentais caros, com exploração limitada de técnicas de aumento de dados para reduzir o custo do treinamento de modelos. O objetivo desta pesquisa é abordar essas lacunas significativas explorando a classificação de uma classe para a detecção de anomalias em pás de turbinas eólicas e desenvolvendo métodos para aumentar os dados existentes para um treinamento de modelos mais econômico na ciência dos materiais. Os métodos aplicados neste trabalho para a detecção de anomalias incluem One Class Support Vector Machine (OCSVM), Support Vector Data Description (SVDD) e autoencoders Long Short-Term Memory (LSTM). Para aumentar os dados experimentais de materiais, foram utilizadas técnicas de adição de ruído e manipulação de imagens. Para a detecção de anomalias, o autoencoder LSTM atingiu uma precisão de 97.4% com aproximadamente 100% de recall, enquanto o OCSVM atingiu 89% de precisão e 97% de recall. O OCSVM foi considerado mais adequado devido aos custos de treinamento mais baixos e desempenho semelhante. Os resultados para aumento de dados experimentais mostraram uma melhoria de 20% em relação aos modelos previamente treinados, com a técnica de aumento melhorando significativamente o desempenho, especialmente em modelos treinados com dados de amostras experimentais diferentes. Em conclusão, esta pesquisa demonstra a eficácia da classificação de uma classe para a detecção de anomalias em pás de turbinas eólicas e destaca os benefícios das técnicas de aumento de dados para o treinamento de modelos econômicos na ciência dos materiais.

Palavras-chave

Engenharia Mecânica; Turbina Eólica; Pá; Monitoramento de Saúde Estrutural; Aprendizado Profundo; Detecção de Falhas.

Table of contents

List of figures

List of tables

List of Abreviations

AE – Acoustic Emission

AMFMD – Adaptive Multivariate Feature Mode Decomposition

ANN – Artificial Neural Networks

AR – Autoregressive

BiConvLSTM – Bidirectional Convolutional Long Short-Term Memory

CNN – Convolutional Neural Network

FEM – Finite Element Method

GNN – Graph Neural Networks

HBCE – Hinged Binary Cross-Entropy

KNN – K-Nearest Neighbors

LSTM – Long Short-Term Memory

MAFResCNN – Multi-Attention Fusion Residual Convolutional Neural Network

ML – Machine Learning

OCSVM – One Class Support Vector Machine

O&M – Operation & Maintenance

PCA – Principal Component Analysis

PSD – Power Spectral Density

PSO – Particle Swarm Optimization

RNN – Recurrent Neural Network

RMSE – Root Mean Square Error

SCADA – Supervisory Control and Data Acquisition

SHM – Structural Health Monitoring

SVDD – Support Vector Data Description

TCVAE-GAN – Temporal Convolutional Variational Autoencoder - Genera-

tive Adversarial Network

XAI – Explainable AI

1
Introduction

Climate change poses one of the greatest threats to humanity, as highlighted by numerous world conferences culminating in the signing of agreements aiming for net-zero emissions by 2050 [1]. Additionally, the global energy crisis of 2022 further emphasized the importance of investing in new energy sources to ensure both energy independence and security, reducing the overall reliance on fossil fuels [2].

In response, wind energy is of high interest due to the world's wind resources, both onshore and offshore, and the technological advancements. In 2021, global wind electricity generation increased by 55% compared to 2020 [3], which is key in the energy transition. In Brazil, the wind generation capacity stands at 32 GWh, contributing 14% to the country's energetic matrix [4] and 6.7% to the global wind generation [5].

This surge is vital for the energy transition but brings challenges, especially in operation and maintenance (O&M). Wind turbines, which combine structural, electrical, and mechanical components [6], require careful maintenance to prevent failures that could be environmentally and financially catastrophic [7].

To mitigate such risks, investments are being made in Structural Health Monitoring (SHM) to detect potential damages early. This technology, which involves the analysis of data from sensors, allows for a transition from time-based to condition-based maintenance strategies [8]. Early detection by O&M teams enables proactive maintenance, reducing downtime and costs while enhancing safety and reliability [9, 10], as shown in Figure 1.1.

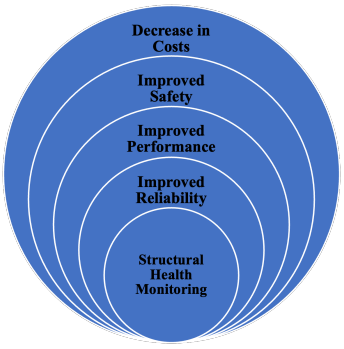


Figure 1.1: Structural Health Monitoring

Blades are one of the most critical parts of a wind turbine since they are a crucial component in the conversion of wind to rotational force in the main axis. Not only are the blades crucial for generation, but they also accounted for about 19.4% of wind turbine incidents in 2012 [11],[12] and 15% to 20% of the total cost of on-shore turbines [11].

Beyond identifying failures, understanding the mechanical behavior of materials is crucial for the design and optimization of components. At the microstructural level, the interfaces between different crystalline grains, known as grain boundaries, are especially significant [13]. However, each grain has a unique orientation; consequently, the grain boundaries end up being lattice imperfections that influence the properties of materials [14][15]. One example of this influence is the preference of the boundaries for the nucleation of micro-cracks due to stress concentrations in those regions [16].

Traditionally, material science research was heavily reliant on time-intensive experimental methods that required substantial investment. However, the advent of advanced computing power and machine learning has transformed this field, enabling the use of large datasets and algorithms to significantly enhance research efficiency and reduce costs [17].

This comprehensive approach from global agreements to microscopic material behaviors illustrates the multifaceted strategies required to address the challenges of modern energy production and turbine maintenance.

In summary, effective SHM systems contribute to maximizing energy production by ensuring turbines operate at optimal performance levels and meet safety standards through the early detection of failures. Additionally, this approach facilitates condition-based maintenance strategies, reducing unplanned downtime and maintenance costs while extending the lifespan of wind turbines. As the global demand for renewable energy sources grows, advancing SHM techniques will play a critical role in meeting energy goals and mitigating the environmental impact of energy production. Hence, the importance of the study of SHM techniques for wind turbines.

1.1 Literature Review

The article from Xiang et al. [21] used a fully labeled dataset that integrates vibrational data from wind turbine tower-mounted accelerometers, operational metrics from Supervisory Control and Data Acquisition (SCADA) systems, and marine meteorological data to train shallow models for regression tasks in the context of structural health monitoring. The study's improved SHM strategy for offshore wind turbine towers, using ML models with Bayesian

optimization, reduced the standard deviation of natural frequencies by up to 71%, significantly enhancing monitoring accuracy.

The few articles that explored anomaly detection methods typically focused on deep learning approaches, which were more computationally expensive. For instance, Fan et al. [22] presented an unsupervised anomaly detection method using a Temporal Convolutional Variational Autoencoder - Generative Adversarial Network (TCVAE-GAN) model with bearings healthy data, achieving an 89% true positive rate.

Zhang et al. [23] explored fault diagnosis methods for wind power converters using a combination of Particle Swarm Optimization (PSO) and Support Vector Machine (SVM) algorithms. The study utilized fully labeled data sets and achieved an accuracy of 98.4%.

The approach outlined in Yan et al. [25] uses a condition monitoring method called Hinged Binary Cross-Entropy (HBCE), which utilizes a Hybrid-attention Bidirectional Convolutional Long Short-Term Memory (BiConvLSTM) model for wind turbine gearbox monitoring. The training is performed on labeled health Supervisory Control and Data Acquisition (SCADA) data for normal operation and fault SCADA data for gearbox oil over-temperature faults. The model achieved a 48.80% reduction in RMSE, a 50.00% reduction in MAE and MAPE, and a 4.04% increase in R^2 compared to traditional models like CNN-LSTM.

The approach outlined in Owolabi et al. [26] explored condition monitoring methods using both Finite Element Method (FEM) and Artificial Neural Networks (ANN) models for wind turbine gearbox monitoring. The FEM approach was utilized for structural analysis and fault investigation, while the ANN models were used for intelligent fault prediction and classification. The training was performed on labeled vibration signal data, with various ANN variants such as Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), and Long Short-Term Memory (LSTM) networks being employed for accurate fault diagnosis. The FEM approach showed an average accuracy of 99% for mesh stiffnesses of planetary gear trains, while among the ANN models, CNN achieved the best results with classification accuracy of up to 98% in some cases.

The approach outlined in Ding et al. [27] explored damage detection methods using acoustic signals for wind turbine blade monitoring. The study focused on both acoustic emission (AE) signals and aerodynamic noise signals for detecting and locating blade damage. Various machine learning algorithms are employed, but all used a fully labeled data set. The study achieved a detection accuracy of 95% and a localization error of less than 10 cm.

The approach described by Xiaoan et al. [28] used a machinery fault simulator equipped with multiple accelerometers mounted on the bearing testing module to detect faults in machinery components such as bearings, gears, and rotors through an Adaptive Multivariate Feature Mode Decomposition (AMFMD) and Multi-Attention Fusion Residual Convolutional Neural Network (MAFResCNN). This deep learning approach effectively identified fault conditions, achieving 98% of accuracy, although it required labeled fault data for accurate fault diagnosis.

As an alternative to acquiring real damage data, the approach proposed by Khazaee et al. [29] involved using simulations to produce fault signals that affect the dynamic behavior of the wind turbine blade. These signals would then be used as input on a Convolutional Neural Network (CNN), which would utilize accelerometers located on the tower for fatigue detection. The approach achieved an accuracy level of about 90% in Wind Turbulence Intensity (WTI) of 10%.

The approach described in Calderano et al. [30] suggested using accelerometers mounted on the blade to detect cracks and ice through a K-Nearest neighbors (KNN) algorithm. While this shallow learning approach was computationally efficient and performed well, it also required labeled fault data. The achieved results demonstrated that using ARMAX for feature extraction, even with simple classifiers, yielded high accuracy (acc $>$ 0.99) in isolating fault cases in wind turbine blades, outperforming traditional PCA methods across all classifiers employed.

In contrast to the other two articles, Reddy et al. [31] employ CNN that are fed with drone images of wind turbine blades. The network is trained using annotated data, including both cracked and uncracked samples. The CNN model achieved an accuracy of around 94.94% for binary fault classification and 91% for multiple class fault classification.

The approach outlined in Liu et al. [32] suggests using an ensemble model that utilizes a deep autoencoder for feature extraction to detect ice, with training performed on labeled "normal" and "icing" SCADA data. The model achieved a detection accuracy of 99% and demonstrated superior performance compared to traditional machine learning models.

In Regan et al. [33], the author proposes an application of data augmentation to generate pseudo-unhealthy data to train a self-supervised model and detect damage. The study conducted on wind turbine blade damage detection using supervised machine learning algorithms found that both logistic regression and support vector machines (SVMs) were effective in classifying the presence of damage with an accuracy greater than 98% for stationary blades.

In literature, there are many proposed instruments and methodologies to monitor wind turbine blades, such as vibration [18] and acoustic-based [19] methods. However, most of the proposed prognostic machine learning approaches require a complete and labeled dataset, including a variety of data from damaged equipment from different failure modes. Unfortunately, obtaining all this labeled data is expensive and, usually, not economically feasible [20]. While several studies address similar issues in other components, such as bearings and gearboxes, the models proposed might not effectively generalize to data from blades.

To predict the mechanical reaction of elastoplastic materials to impact, Long et al. [34] used Graph Neural Networks (GNN) to predict material fracture strengh and plastic strength using experimental datasets available online though not applying any data augmentation technique. The R^2 values for the optimal model in predicting the properties of bulk metallic glasses (BMGs) were 0.963 for fracture strength and 0.801 for plastic strain. Additionally, the fine-tuned model showed an improvement in R^2 scores by 23.8% for the plastic strain dataset, demonstrating significant enhancements in predictive performance.

The study by Forest et al. [35] utilized a labeled dataset containing image patches for crack classification and also used explainable AI (XAI) methods to generate segmentation masks. The analysis was primarily macrostructural; the primary objective of the models was to detect the presence of these surface cracks, rather than assessing deeper structural issues such as residual strain. The methods were able to roughly estimate the number of cracks, crack area, and maximum crack width. DeepLift and LRP were among the best performers, with DeepLift achieving a mean absolute error of 0.72 in estimating the number of cracks per patch.

Lin et al. [36] employed stress as the input for a two-hidden-layer Artificial Neural Network (ANN), but with datasets fully generated through numerical simulation instead of experimental. The prediction errors, expressed in terms of ballistic limit velocity (BLV), were 1.16% for the data-driven approach.

Hasan et al. [37] utilized both deep and shallow learning models to predict material fracture critical stress levels using microstructural features as input. The study achieved a high coefficient of determination with the Multi-Layer Perceptron. However, the dataset used was generated through computational simulation due to the costs and difficulties associated with acquiring experimental data. The neural network models achieved R^2 values of 0.86, 0.88, and 0.93 at strain levels of 0.75%, 2.25%, and 3.25%, respectively.

Zhang et al. [38] employed the XGBoost algorithm to predict grain boundary (GB) damages in magnesium alloy specimens using various crystallographic and geometric features as inputs. The model achieved AUC scores of 89% with a dataset fully derived from experimental observations. The machine learning model achieved AUC (area under the receiver operating characteristic curve) scores of 73.7% for E-0 and 84.1% for E-90 in predicting grain boundary damages.

Frankel et al. [39] employed a hybrid neural network model combining convolutional neural networks and long short-term memory networks to predict the evolution of the stress field in polycrystals undergoing elastic-plastic deformation. The study achieved high-fidelity predictions that matched well with crystal plasticity data. However, the dataset used was generated through computational simulations due to the high computational cost and complexity of obtaining three-dimensional experimental data. In this study, the normalized mean RMSE values were 0.0258 (median), indicating that on average, the predictions are very close to the actual values.

Thomas et al. [40] utilized graph neural networks to predict fatigue damage within polycrystals using microstructure representations as input. The study achieved a balanced accuracy of 0.72 with the Graph Convolutional Network (GCN). However, the dataset used was derived from multimodal experimental data.

Qu et al. [41] developed a micromechanics-informed deep learning model for constitutive behavior prediction of granular materials. The models were created using three training strategies with deep neural networks to predict stress responses based on external and internal variables identified through an incremental stress-strain relationship analysis. The data set used was created through discrete element modeling. The tested models achieved an average prediction score of 0.977, which used the scaled mean absolute error (SMAE) and an empirical cumulative distribution function (eCDF) of the scaled squared errors (SSE) to evaluate prediction accuracy.

Vieira & Lambros [42] obtained experimental data and employed neural networks, along with the geometric angle of the grain, to predict residual plastic strain from each angle value. As a result, the best dataset achieved an average Pearson correlation coefficient (R) of 0.65 between the measured and predicted values.

Araujo et al. [43] used the same experimental dataset from [42] and input parameters to employ random forest, with the difference of transforming the dataset to predict the residual plastic strain from the center of a 35x35 square utilized as input. As a result, the best dataset achieved an average Pearson

correlation coefficient (R) of 0.72 between the measured and predicted values

In literature, there are not many proposed methods to correlate grain boundary geometry with plastic strain accumulation. Also, most of the proposed approaches hinge solely on experimental datasets, which restricts the full potential of the models due to the lack of data.

1.2 Objectives

The primary objective of this thesis is to advance the application of machine learning and deep learning techniques in structural health monitoring (SHM) and material science. The secundary goals are as follows: i) to establish a framework for real-time anomaly detection; ii) to augment the experimental dataset generated by Vieira & Lambros [42]; and iii) to illustrate the potential of Deep Learning methods for predicting plastic strain in grain boundaries of polycrystalline materials.

To attain the overarching objectives of the thesis, the following steps were undertaken: i) proposing a deep learning methodology that optimizes accuracy to formulate best practices for real-time anomaly detection; ii) suggesting the application of deep learning algorithms for microscopic plastic strain prediction in grain boundaries; iii) generating quantitatively and qualitatively valid results for both prediction and anomaly detection models.

1.3 Contribution

In the anomaly detection approach, the following contributions have been made in order to address the gap in literature regarding deep and shallow models trained exclusively on healthy labeled datasets:

(i) A data ingestion workflow utilizing autoregressive models (AR) and Principal Component Analysis (PCA) to extract characteristics from accelerometer signals, which are then used to feed one-class classification shallow models. (ii) The application of Monte-Carlo holdout with a repeated cross-validation model validation procedure for the evaluation and comparison of shallow learning models. (iii) The utilization of a Long Short-Term Memory autoencoder model to learn from normal data and detect anomalies by identifying deviations.

In the prediction approach, the following contribution have been made to bridge the literature gap on micro-structural strain prediction datasets and to extend the work of Vieira & Lambros [42] and Araujo et al. [43]:

(i) The application of Multi-Layer Perceptrons and Convolutional Neural Networks, coupled with data augmentation techniques, to predict the accumulation of plastic strain at grain boundaries.

1.4 Organization

The remainder of the thesis is organized as follows: Chapter 2 provides an introduction to the fundamental concepts essential for comprehending the data source and understanding the machine learning models.

In Chapter 3, the construction and validation of both shallow and deep learning models for anomaly detection are presented. This includes feature extraction using Autoregressive (AR) models and Principal Component Analysis (PCA), Monte-Carlo validation with k-folds for shallow models, and the utilization of the LSTM autoencoder model. The resulting plots for all procedures are displayed, followed by a brief discussion of the obtained results.

Chapter 4 focuses on the deep learning models employed to predict plastic strain at grain boundaries in polycrystalline materials, along with the data augmentation techniques. Similar to Chapter 3, resulting plots for all procedures are showcased, followed by a concise discussion of the obtained results.

Finally, Chapter 5 provides a summary of the research, presenting conclusions and offering guidelines for further implementations to enhance the proposed method.

2
Theoretical Background

This chapter provides the theoretical foundation for the experimental procedures utilized in this work. Section 2.1 elucidades fundamental concepts of material microstructure and fatigue behavior. Section 2.2 details the preprocessing techniques, starting with power spectral density (subsection 2.2.1), followed by autoregressive models (AR) in subsection 2.2.2, principal component analysis (PCA) in subsection 2.2.3, and data augmentation in subsection 2.2.4. Section 2.5 covers unsupervised learning methods for anomaly detection, including one-class support vector machines (OCSVM) in subsection 2.5.1, support vector data descriptor (SVDD) in subsection 2.5.2, and LSTM autoencoders in subsection 2.5.3. Section 2.6 discusses predictive approaches, with a focus on multi-layer perceptron in subsection 2.6.1 and convolutional neural networks in subsection 2.6.2. Finally, Section 2.7 outlines the methodologies employed for model evaluation.

2.1
Fundamentals of Material Microstructure and Fatigue Behavior

Material microstructure refers to the arrangement and organization of phases and defects within a material on a microscopic scale. It is composed of various elements such as grain size, orientation, and the presence of grain boundaries and dislocations that, even though these are microscopic elements, they are determinants of material macroscopic properties and behavior [44].

Grains are individual crystalline regions within a material, separated by grain boundaries, which are interfaces where there is a misorientation between adjacent grains. The angle of misorientation, known as the grain boundary angle, influences material properties such as ductility and toughness by contributing to atomic disorder. Also, the boundaries act as physical barriers to the movement of dislocations, increasing material strength but often decreasing its ductility. Also, the orientation of grains can lead to anisotropic behavior, where properties vary with direction [45].

When materials are subjected to static loads, they deform elastically up to a certain point, when the material has the ability to recover its original shape after the load is removed. Beyond this point, plastic deformation occurs

and the shape is permanently affected [46]. The material load history affects the microscopic structure, inducing memory effects and often leaving residual plastic strain after the load is removed [46]. This residual strain results from the rearrangement and movement of dislocations within the material's microstructure, which can result from the magnitude and duration of the load, as well as the material properties, causing internal stresses and permanent deformation at a microscopic level [46].

Also, often materials operate under varying load conditions, in the case of wind turbine blades for example, the constantly changing wind speeds and directions create repeated cycles of stress [47]. While the loads may primarily cause elastic deformation, the cumulative effect of these repeated stress cycles can lead to fatigue over extended periods due to microstructural residual plastic strains [46]. The microstructural changes and the presence of residual plastic strain can lead to the initiation of cracks at stress concentrators, such as grain boundaries or defects. With continued cyclic loading, these cracks can propagate leading to macroscopic failures [46].

2.2 Data Preprocessing

Data preprocessing is the first step of data workflow, it involves all steps necessary to prepare raw data for analysis and training of the machine learning models. Effective preprocessing enhances the performance of machine learning algorithms by improving data quality and ensuring that the dataset is both comprehensive and representative of the problem domain. This foundational step not only improves the accuracy and efficiency of subsequent analysis but also helps to mitigate biases and errors, ultimately leading to more robust and generalizable models [48].

Data preprocessing is also important to avoid the "curse of dimensionality," which occurs when data with many features makes training extremely slow and complicates finding a good solution. The challenge in reducing dimensionality is that some information is inevitably lost [49]. Sections 2.2.1, 2.2.2, and 2.2.3 explore the methods used to address this problem while minimizing information loss.

On the other hand, having too little data presents another challenge due to the difficulty and costs associated with acquiring a dataset that is representative of the entire problem [50]. In such cases, it is important to find ways to augment the dataset to improve model training and accuracy. Section 2.2.4 explores the methods used for data augmentation.

2.2.1 Power Spectral Density (PSD)

The act of acquiring acceleration values at each time $x(t)$ indicates that the acceleration data is time-series data or, in other words, data in the time domain. Another complementary way of analyzing a time-series signal is by transforming it from the time domain to the frequency domain $X(f)$ (or spectrum), often represented by the Power Spectral Density (PSD). This transformation allows for the analysis of the frequency characteristics of a system, which is crucial for developing accurate models [51].

This transformation between time-domain and frequency-domain is achieved through the Fourier Transform which decomposes a signal into its constituent sinusoidal components, revealing the amplitude and phase of each frequency component [52].

$$X_T(f) = \int_{-\inf}^{\inf} x(t) \exp^{-j2\pi ft}\, dt \tag{2-1}$$

Where $\exp^{-j2\pi ft}$ is the complex exponential function that decomposes the signal into its constituent sinusoidal components.

The Fourier Transform is well-suited for analyzing vibration signals characterized by a finite number of dominant frequency components. However, real-world vibration analysis often involves additional "random" vibrations that affect measurements. This is why the Power Spectral Density (PSD) is calculated using the FFT. The PSD provides a comprehensive view of the power distribution across all frequency components present in the signal, including both dominant and random vibrations [53].

Mathematically, the PSD $S_x(f)$ is defined as [52]:

$$S_x(f) = \lim_{T\to\infty} \frac{1}{T}\mathbb{E}[|X_T(f)|^2] \tag{2-2}$$

Power Spectral Density represents a fundamental tool for understanding signal characteristics, aiding in the development of accurate models and facilitating effective vibration analysis.

2.2.2 Autoregressive Models (AR)

Autoregressive (AR) models are highly effective for time series analysis as they explicitly account for temporal dependencies by utilizing past values to predict future outcomes. By incorporating these past observations, AR models can detect and capture trends, seasonality, and other temporal patterns in the data, leading to precise and understandable forecasts [54].

Autoregressive models are based on the concept that the current value of the series, x_t, can be expressed as a function of p previous values, also referred to as lags, $x_{t-1}, x_{t-2}, \ldots, x_{t-p}$. Mathematically, the AR model function can be denoted as [55].

$$x_t = \phi_1 x_{t-1} + \phi_2 x_{t-2} + \cdots + \phi_p x_{t-p} + \epsilon_t \tag{2-3}$$

Where ϕ_1,$\phi_2, \ldots, \phi_p$ are the model coefficients and ϵ_t is a white noise error term with zero mean and constant variance.

However, to utilize AR models effectively for time series analysis, certain assumptions must be satisfied. Firstly, the time series must demonstrate stationarity, indicating that its statistical properties such as mean and variance remain constant over time. Stationarity ensures that the relationships between past and present values remain consistent, thus enabling reliable predictions [56].

Additionally, AR models presume a linear relationship between the current value and its past values. This assumption simplifies the modeling process by suggesting that changes in the independent variables (lags) result in proportional changes in the dependent variable (current value) [55].

Moreover, AR models require the absence of autocorrelation in residuals. The error term in the AR model equation, representing the difference between observed and predicted values, is assumed to be white noise. This entails a zero mean, constant variance, and no autocorrelation in the residuals. Autocorrelated residuals suggest inadequacies in the model or the presence of additional explanatory variables, compromising the model's accuracy [55].

2.2.3 Principal Component Analysis (PCA)

As the number of features or dimensions increases within a dataset, the data space expands exponentially. This expansion results in sparsity within the data, where data points are increasingly scattered across the high-dimensional space. Consequently, traditional algorithms and models struggle to capture meaningful patterns, relationships, and structures in such sparse and high-dimensional data, leading to the curse of dimensionality, which adversely affects model's performance and interpretability [49].

Principal Component Analysis (PCA) serves as a widely employed statistical technique to address the curse of dimensionality, leading to dimensionality reduction and data compression. At its core, PCA aims to transform high-dimensional data into a lower-dimensional space while retaining the essential information present in the original dataset [57].

PCA operates by identifying the axis that explains the highest variance in the training set, known as the first principal component. Subsequently, it identifies the second axis, termed the second principal component, and continues this process until the i_{th} axis, referred to as the i_{th} component. The mechanism by which PCA determines each of these components involves employing a matrix factorization technique called Singular Value Decomposition (SVD), after standardizing the data to a mean equal to zero and a standard deviation of one [58].

Right after representing the dataset with j data points and i variables using the matrix A_{ij}, the SVD factorization begins by standardizing the data, ensuring a mean of zero and a standard deviation of one for each variable. Following standardization, the SVD is computed for the standardized data matrix A. This involves decomposing the matrix A into three separate matrices: the left singular vectors matrix U, the diagonal matrix of singular values Σ, and the right singular vectors matrix V^T. These matrices satisfy the equation [58]:

$$A = U\Sigma V^T \tag{2-4}$$

The columns of U and V represent the principal components of the data, and the singular values in Σ indicate the importance of each component. This SVD factorization allows for dimensionality reduction and reconstruction of the original dataset using a subset of principal components [58].

2.2.4 Data Augmentation

Data augmentation represents a widely used technique in datasets utilized for training both shallow and deep learning models, particularly within the realm of computer vision. It involves creating synthetic data from the original dataset through the application of various transformations to the existing data. Initially, the primary aim of this technique was to address class imbalances, yet it has since been applied for other purposes, such as diminishing over-fitting and enhancing the model's capacity for generalization [59].

Data augmentation techniques can be categorized into two main types: basic and advanced techniques. Basic techniques include simple image transformations, while advanced involve more sophisticated methods such as generative models. In order to define the best technique for each model, it is important to take into consideration the specific characteristics of the dataset, the model, and also involves experimentation to properly evaluate the impact of the techniques in the use case. In this work, the following techniques were

used: Gaussian noise injection and image manipulation [60].

The technique of injecting gaussian noise to the data consists of adding to a portion of the dataset a random value drawn from a normal distribution. Mathematically it can be represented as [61]:

$$x' = x + n \tag{2-5}$$

Where n is the noise taken from a normal distribution with mean μ and variance σ^2, and x the true value taken from the dataset [61].

An image is a matrix of pixel values, where each value represents the intensity of the pixel at a specific location. So, similarly to gaussian noise injection, image manipulation consists of mathematically manipulating those pixels to create variations of the original image and, therefore, augmenting the dataset [62].

Some common methods are horizontal and vertical flips, rearranging the matrix columns and rows, respectively. Mathematically, a horizontal flip can be represented as [62]:

$$\text{Original matrix: } \begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \end{bmatrix}$$

$$\text{Flipped matrix: } \begin{bmatrix} p_{13} & p_{12} & p_{11} \\ p_{23} & p_{22} & p_{21} \\ p_{33} & p_{32} & p_{31} \end{bmatrix}$$

Other common technique is zooming, which consists in scaling the image up or down, either focusing on a central portion or the entire image, to simulate different levels of magnification. When zooming in a image by a factor of k each element from the original image I is replicated $k \times k$ times to fill the zoomed in matrix. Mathematically, each element can be represented as [62]:

$$I_{\text{zoom\_in}}(i,j) = I\left(\left\lfloor \frac{i}{k} \right\rfloor, \left\lfloor \frac{j}{k} \right\rfloor\right) \tag{2-6}$$

Where $i = 0, 1, \ldots, k \cdot m - 1$ and $j = 0, 1, \ldots, k \cdot n - 1$.

2.3 Hyperparameter Optimization

In machine learning, a hyperparameter is a parameter whose value is set before the model learning process begins and has the goal to control the behavior of the training algorithm. Unlike model parameters, which are learned from the training data, hyperparameters are predefined and can significantly affect the performance of the model [63].

Random Search CV is an optimization technique that randomly samples a wide range of hyperparameter combinations to find the best set for the model. Instead of exhaustively searching through all possible combinations, Random Search CV evaluates a fixed number of randomly chosen configurations, which can be more efficient and often yields comparable results to more exhaustive methods, and select a random value for each hyperparameter in each iteration [58].

This approach allows for a more extensive exploration of the hyperparameter space within a reasonable time, improving the model's performance by identifying optimal parameter settings. However, the method has its limitations. For instance, it relies on the assumption that randomly sampled hyperparameter values are representative of the entire hyperparameter space, which may not hold if the hyperparameter space is highly irregular or if certain regions contain the optimal configurations [64].

Despite its limitations, Random Search CV often outperforms other hyperparameter optimization techniques, such as full grid search, making it a crucial tool for model performance optimization. Bergstra and Bengio [64] demonstrated that Random Search CV could achieve better results with fewer evaluations compared to grid search, especially in high-dimensional hyperparameter spaces.

2.4 Cross Validation Techniques

Cross-validation is a fundamental technique in machine learning and statistical modeling used to assess the generalizability of a model. It involves splitting the dataset into multiple subsets, where the model is trained and tested on different random splits of the original dataset, thus addressing several challenges in model development [64]. Firstly, in cases where the dataset is small, it helps making the use of data more efficient by allowing every data point to be used for both training and validation [64]. Additionally, as stated in the **no free lunch theorem** [65], no machine learning algorithm is universally better than the other, so cross-validation aids in model selection by providing a robust framework for comparing different models or hyperparameter settings, ensuring that the chosen model or parameters are not biased by a particular train-test split [58].

Monte Carlo method is a broad class of computational algorithms that rely on repeated random sampling to solve a problem. In this work, it was applied Monte Carlo Cross-Validation to create n random splits of the training and testing data and, therefore, n models for each method. In this way, each

tested method can be evaluated statistically to understand how it performs in terms of not only accuracy but also consistency [66].

To ensure the robustness of the validation process, k-fold cross-validation is used. K-fold cross-validation is a method where the training data is divided into k equally sized folds. The model is trained k times, each time using a different fold as the validation set and the remaining $k-1$ folds as the training set, ensuring that every data point is used for validation exactly once, and for training $k-1$ times [63].

2.5 Anomaly Detection Methods

Anomaly detection is the process of identifying patterns in data that deviate from the expected behavior, often referred to as outliers. These anomalies can represent rare events, errors, or novel occurrences that significantly differ from the majority of the data. The main goal of anomaly detection techniques is to have a preventative approach to potentially harmful conditions, allowing for early intervention and mitigation [67]. Anomaly detection methods can be classified into three categories: supervised, where the data is labeled; unsupervised, where the dataset lacks labels; and semi-supervised, where the training data includes only labeled normal (positive) instances [67]. Sections 2.5.1, 2.5.2 and 2.5.3 explore unsupervised methods used for anomaly detection, all using a semi-supervised approach.

2.5.1 One Class Support Vector Machine (OCSVM)

One-Class Support Vector Machine (OCSVM) is a powerful machine learning algorithm for anomaly detection and outlier identification in datasets. It builds on the principles of the Support Vector Machine (SVM) algorithm, which seeks to separate classes by finding the widest possible margin between them, a technique known as large margin classification, as shown in Figure 2.1 [49].

However, unlike traditional SVM, which requires data from multiple classes, OCSVM focuses on distinguishing normal data from anomalies within a single class. It works by finding a decision boundary that surrounds the majority of the data points in the feature space, effectively learning the characteristics of the normal data. Any data points that fall outside this boundary are considered anomalies [68].

Mathematically, the OCSVM is defined as [68]:

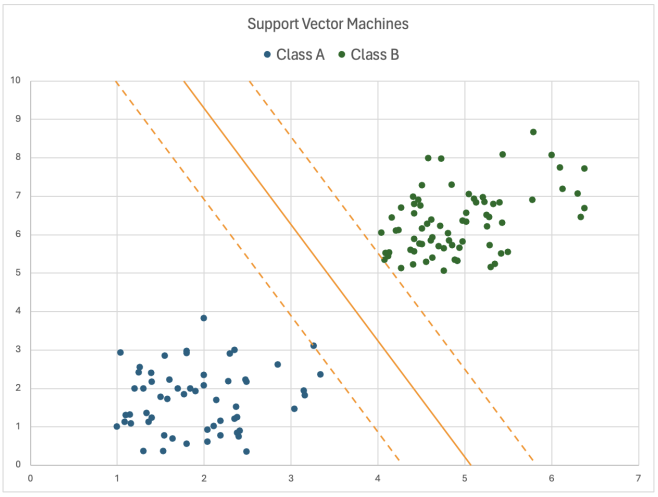


Figure 2.1: Illustration of margins used to separate classes in SVM models inspired by [58].

$$\min_{\mathbf{w},\xi_i,\rho} \left(\frac{1}{2}|\mathbf{w}|^2 + \frac{1}{\nu n} \sum_{i=1}^{n} \xi_i - \rho \right) \tag{2-7}$$

With the constraints:

$$(\mathbf{w} \cdot \phi(\mathbf{x}_i)) \geq \rho - \xi_i, \quad \xi_i \geq 0 \tag{2-8}$$

Where $\mathbf{w}$ represents the weight vector, $\phi(\mathbf{x}_i)$ is the feature mapping of the input data $\mathbf{x}_i$, ξ_i are the slack variables that allow some data points to lie outside the boundary to make the training phase less sensitive to outliers, ρ is the offset, ν is a parameter controlling the trade-off between the size of the margin and the number of allowed outliers, and n is the number of data points [68].

2.5.2 Support Vector Data Descriptor (SVDD)

Support Vector Data Description (SVDD) is another robust technique for anomaly detection, closely related to One-Class Support Vector Machine (OCSVM). SVDD is an algorithm to identify outliers in datasets by learning the properties of the normal data and distinguishing these from anomalous data points [69].

The goal of SVDD is to find the smallest sphere in the feature space containing most of the data points. Any data points lying outside this sphere are classified as anomalies. This approach can be effective when dealing with datasets where the normal instances form a cluster, as the hypersphere provides a simple description of the data's distribution [70].

Mathematically, the SVDD problem can be formulated as:

$$\min_{R,\mathbf{a},\xi_i} \left(R^2 + C \sum_{i=1}^{n} \xi_i \right) \tag{2-9}$$

With the constraints:

$$||\phi(\mathbf{x}_i) - \mathbf{a}||^2 \leq R^2 + \xi_i, \quad \xi_i \geq 0 \tag{2-10}$$

Where R is the radius of the hypersphere, a is the center of the hypersphere, $\phi(\mathbf{x}_i)$ is the feature mapping of $\mathbf{x}_i$ and ξ_i is the slack variable that allows the presence of outliers. C controls the trade-off between the volume of the sphere and the penalty for data points outside the sphere (outliers) [70].

2.5.3 Long Short-Term Memory

A Recurrent Neural Network (RNN) is a type of artificial neural network designed to process sequential data, such as time series, in a way that allows previous time steps to be fed back into the network, enabling it to maintain a memory of previous inputs [71].

Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN) that is designed to handle the vanishing gradient problem in RNNs, which occurs when the gradients used to update the weights of the network during training become too small, causing the network to learn slowly or not at all [71].

LSTMs use a structure called a memory cell, illustrated in Figure 2.2, allowing the network to sellectivelly remember or forget information from previous time steps. The memory cell is controlled by three gates: the input gate, the forget gate, and the output gate. The input gate controls how much new information is added to the memory cell, the forget gate controls how much old information is forgotten, and the output gate controls how much of the memory cell is used to make predictions at each time step [72].

The equations governing these gates and the memory cell in an LSTM are as follows [72]:

– Input Gate:

$$\begin{aligned} i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\ \tilde{C}_t &= \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \end{aligned} \tag{2-11}$$

– Forget Gate:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{2-12}$$

– Output Gate:

$$\begin{aligned} o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\ h_t &= o_t \odot \tanh(C_t) \end{aligned} \tag{2-13}$$

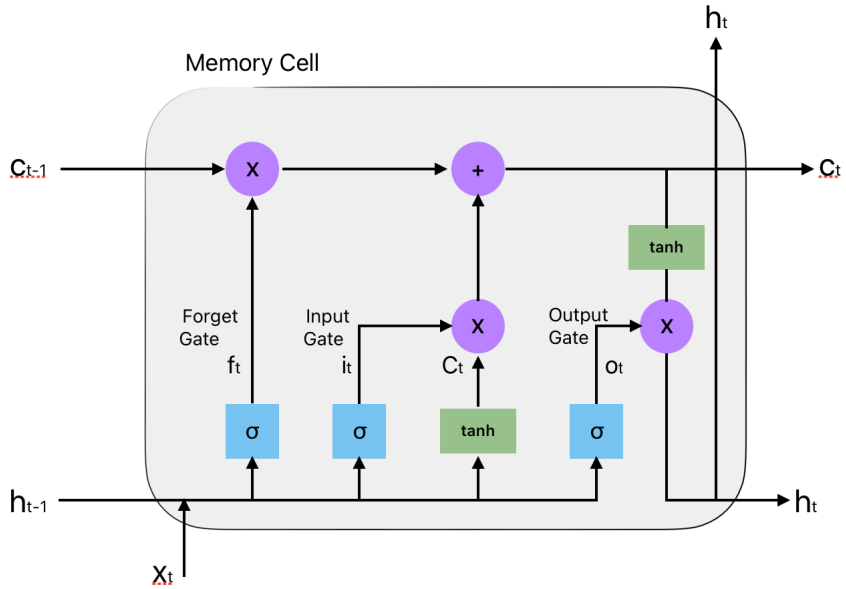


Figure 2.2: Structure of LSTM memory cell inspired by the concepts presented in [72]

The LSTM architecture is effective in various applications, including speech recognition, language translation, and image captioning [72]. In their article, Hochreiter and Schmidhuber [72] introduce the LSTM architecture and demonstrate that it outperforms other RNN architectures.

2.6 Regression Methods

Different from anomaly detection,which is a type of classification approach, regression methods involve predicting continuous outcomes based on input data. This is achieved through modeling the relationship between dependent and independent variables in data. Regression methods can be used for various purposes, such as forecasting and replacing missing values in datasets. Sections 2.6.1 and 2.6.2 explore the supervised methods used for regression.

2.6.1 Multi-Layer Perceptrons

A Multi-Layer Perceptron (MLP) is a type of artificial neural network designed to model complex relationships and patterns in data. As illustrated in Figure 2.3, it consists of multiple layers of interconnected nodes, known as neurons or perceptrons, organized into an input layer, one or more hidden layers, and an output layer [73].

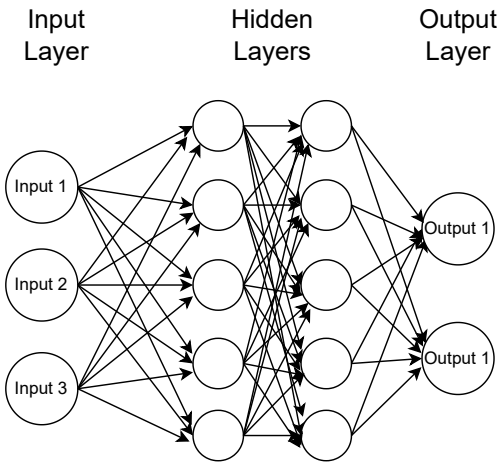


Figure 2.3: Multi-Layer Perceptron Schema inspired by the concepts presented in [58]

A MLP works based on a feedforward mechanism, meaning that the data flows from the input layer through the hidden layers to the output layer. In each layer, the node connections are associated with weights, and the network learns by adjusting these weights during training to minimize the difference between predicted and actual outcomes. Additionally, to introduce non-linearities to the model, each node can also incorporate activation functions [58].

Mathematically, each neuron calculates the weighted sum and has an activation function [74]:

– Weighted Sum:

$$z_j^l = \sum_{i=1}^{n_{l-1}} w_{ij}^l a_i^{l-1} + b_j^l \tag{2-14}$$

Where z_j^l is the weighted sum at neuron j in layer l, w_{ij}^l is the weight connecting neuron i in layer $l-1$ to neuron j in layer l, a_i^{l-1} is the output (activation) of neuron i in layer $l-1$ and b_j^l is the bias term for neuron j in layer l [74].

– Activation Function:

$$a_j^l = \sigma(z_j^l) \tag{2-15}$$

Where σ is most commonly sigmoid, tanh, ReLU (Rectified Linear Unit), and softmax (as the output layer in classification tasks) [58].

The weights w_{ij}^l and biases b_{ij}^l are iteratively adjusted using backpropagation combined with a optimization algorithm with the main goal of minimizing the loss function, which is the difference between the real and the predicted outputs [58].

2.6.2 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are a specialized type of artificial neural network particularly well-suited for processing data that has a grid-like topology, such as images. CNNs leverage the spatial structure in data to extract hierarchical patterns and features through convolutional layers, pooling layers, and fully connected layers [75].

A typical CNN architecture consists of several key components, as illustrated in Figure 2.4:

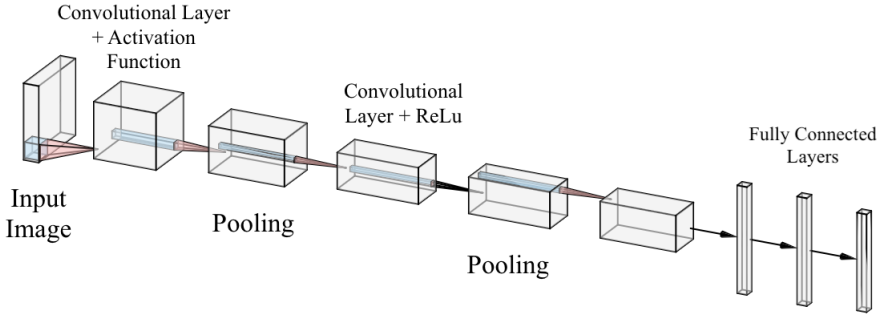


Figure 2.4: Convolutional Neural Network Schema inspired by the concepts presented in [76]

- **Convolutional layers:** they aim to extract important features through various linear convolution operations on the input image. These operations take as inputs the image array and a filter (or kernel), which is also represented by a matrix with smaller height and width than the input image but with the same number of channels, resulting in a two-dimensional convolution operation. Thus, through linear combinations, the filters model the relationships between pixels and their neighbors [76].

  The filters have a set of adjustable attributes to control the model training called hyperparameters. These are [76]:

  - Depth: determines the number of filters in the convolutional layer, each responsible for extracting a different feature. The higher the number of filters, the greater the number of features and also the higher the operational cost.
  - Stride: determines the number of pixels to skip in the input matrix when applying the filter; typically, the maximum stride is 2.

  - Zero-padding: fills the borders of the input matrix with zeros to adjust its size, thus making it possible to fit the filters across the entire image and produce an output with the same volume as the input.

After applying the filters, the output passes through an activation function (e.g., ReLU, LReLU, PReLU, tanh, Sigmoid, etc.) responsible for introducing non-linearities into the system, thereby increasing the network's accuracy. During the network training, weights and biases are adjusted in neurons to minimize errors [76].

- **Pooling**: pooling layers aim to reduce the dimensionality of the output from convolutional layers, thereby preventing overfitting and reducing the computational cost of the network, while retaining important information. Dimensionality reduction is achieved by aggregating multiple features into one, which can take various forms such as max-pooling, L2 norm, L1 norm, average pooling, among others. It is important to emphasize that this layer does not involve learning parameters [77].
- **Fully Connected Layer (FC)**: This layer aims to vectorize the output tensor from the feature extraction layers, combining the features extracted from convolutional layers, and feeding them into an activation function responsible for classifying the model's outputs [77].

  All neurons from the previous layer are connected to every neuron in the following layer. In more recent architectures, FC layers with ReLU and softmax outputs are commonly used [76].

  Additionally, the dropout technique can be implemented in fully connected layers to prevent overfitting and reduce the number of training iterations, randomly dropping out a percentage of neurons in each iteration [76].

2.7 Model Evaluation

To evaluate classification models, it is necessary to understand the various metrics that provide insights into their performance [78]. This section discusses the evaluation metrics used in this work, including confusion matrix in 2.7.1, accuracy and balanced accuracy in 2.7.2, coefficient of determination (R^2) in 2.7.3, root mean square error (RMSE) in 2.7.4, and Pearson correlation in 2.7.5.

2.7.1 Confusion Matrix

A confusion matrix is a table used to describe the performance of a classification model, it compares the actual target values with those predicted by the model. The matrix provides a detailed breakdown of the classification performance by showing the number of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) [79].

As illustrated in Figure 2.5:

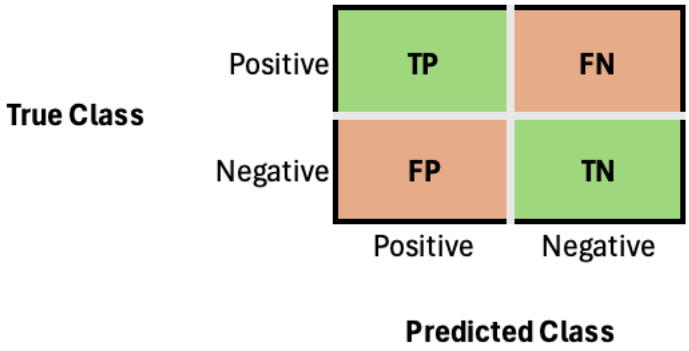


Figure 2.5: Confusion Matrix Schema inspired by the concepts presented in [79]

- True Positives (TP): The number of instances correctly predicted as positive.
- True Negatives (TN): The number of instances correctly predicted as negative.
- False Positives (FP): The number of instances incorrectly predicted as positive.
- False Negatives (FN): The number of instances incorrectly predicted as negative.

2.7.2 Accuracy

Accuracy is a performance metric commonly used in classification tasks to calculate the proportion of correctly predicted instances, including both true positives and true negatives (as described in Section 2.7.1), out of the total instances. It is calculated as [80]:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2-16}$$

However the accuracy calculation can be misleading in cases of inbalanced datasets, where one class significantly outnumbers the others. So, in order to accurately measure the performance of the model in those cases, the balanced accuracy is calculated as [81]:

$$\text{Balanced Accuracy} = \frac{1}{2}\left(\frac{TP}{TP+FN} + \frac{TN}{TN+FP}\right) \tag{2-17}$$

2.7.3 Coefficient of determination (R^2)

The coefficient of determination (R^2) is a key metric for regression tasks. It measures the proportion of the variance in the dependent variable (model outputs) that is predictable from the independent variables (model inputs). In other words, it shows how well the model's predictions match the actual data [82].

The coefficient of determination is calculated as:

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} \tag{2-18}$$

Where SS_{res} is the differences between observed and predicted values (sum of squares of residuals) and SS_{tot} is the differences between observed values and their mean (total sum of squares).

The R^2 values can range from 0 to 1, with higher values indicating better model performance and lower values indicating worse performance.

2.7.4 Root Mean Square Error (RMSE)

The Root Mean Square Error (RMSE) is a measure of the differences between predicted and observed values. It is the square root of the average of squared differences between prediction and actual observation. RMSE is defined as [83]:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{2-19}$$

Where y_i is the actual value and $\hat{y}_i$ is the predicted value. RMSE goal is to assess the magnitude of prediction errors, but it should be used with care since it is particularly sensitive to outliers [58].

2.7.5 Pearson Correlation Coefficient

Pearson correlation measures the linear relationship between two variables, providing a value between -1 and 1. A value of 1 indicates a strong

positive correlation, -1 indicates a strong negative correlation, and 0 indicates no linear relationship [58].

Mathematically it can be represented as [83]:

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{2-20}$$

Where x_i and y_i are the sample points and $\bar{x}$ and $\bar{y}$ are the mean values of x and y, respectively.

3
Blade Anomaly Detection

This chapter unveils the outcomes of the anomaly detection approach. The section is structured as follows: Section 3.1 provides details on the accelerometer measurement setup on the wind turbine blade; Section 3.2 outlines the procedures to train Shallow Learning Models, the subsection 3.2.1 describes the feature extraction process using AR, PSD and PCA; Section 3.2.2 details the construction of OCSVM and SVDD; Section 3.3 covers the construction of the Deep Learning Model, LSTM autoencoder; Section 3.4 presents the results obtained from both shallow and deep learning models while Section 3.5 presents a final discussion of the results; In this chapter, the main emphasis is placed on identifying anomalous operations by analyzing deviations from measured accelerometer signals in comparison to what is considered normal behavior. All methods descibed bellow were created using Python.

3.1
Wind Turbine Blade Vibration Measurements

The experimental benchmark data set [84] used in this work is based on the 1.5m-long blade of a Sonkyo Energy Windspot 3.5kW wind turbine with a total mass of 5kg. The properties of the wind turbine are described in Table 3.1.

Table 3.1: Windspot 3.5 kW WT - Sonkyo Energy

Rating:	3.5 kW
Type:	Upwind horizontal axis
Rotor Diameter:	4.05 m
Hub height:	12, 14 and 18 m
Cut-in wind speed:	3 m/s
Rated wind speed:	12 m/s
Cut-out wind speed:	30 m/s
Rated rotor speed:	250 rpm
Rotor mass:	185 kg

The blade underwent dynamic testing in a laboratory on a fixed-free setup inside a climate chamber for temperature control purposes. As shown in Figure 3.1, Node 1 and the signal generator are responsible for creating two different types of dynamic excitation signals, each with a duration of 120 seconds, determined by finite element analysis:

- White noise signal with a frequency bandwidth between 0 and 400 Hz.
- Sine sweep signal with frequencies ranging from 1 to 300 Hz.

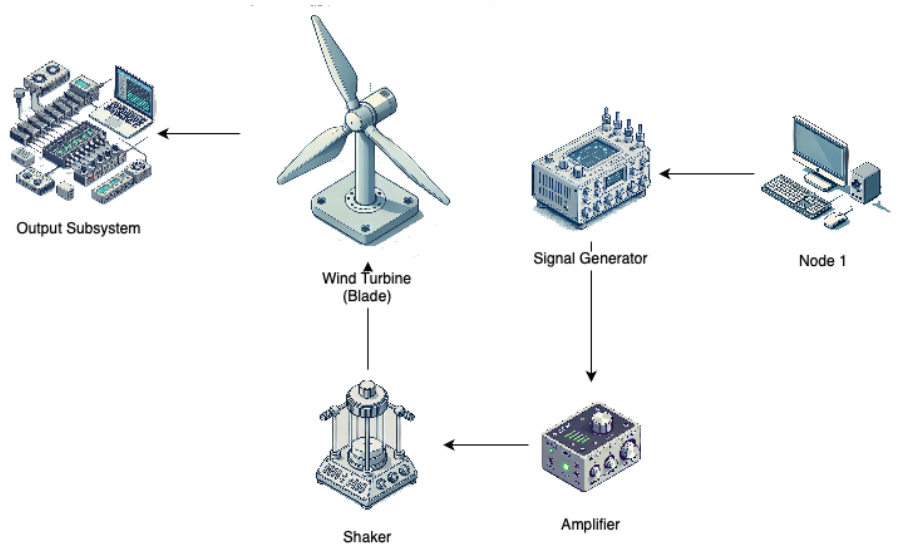


Figure 3.1: Illustration of the data acquisition system inspired by the concepts presented in [84]

The excitation signal was transferred to the wind turbine blade shaker through the power amplifier. To ensure the shaker operated effectively across the entire temperature range, it was placed in an insulation box with an electrical bulb to generate heat during low temperatures.

While the blade was subjected to the shaker forces, its response was retrieved and recorded by the installed instrumentation directly through the output system. The instrumentation setup consisted of humidity, temperature, and force sensors, 8 piezoelectric accelerometers, and 22 strain gauges arranged in two different settings. However, this study only uses signals from the accelerometers since they are less expensive, less sensitive to the installation location, and are already used to monitor other equipment on the wind turbine.

The blade was subjected to examination in various temperature settings, covering both healthy and damaged scenarios, as elaborated in Table 3.2. Case 'R' was generated with the blade in its initial healthy state. After case "R", cases "A", "B", and "C" were created to simulate icing accretion to the blades,

which is not considered a damage scenario but a change in the structure. Since the goal of this work is to assess damages, those cases were deemed healthy for these purposes.

Table 3.2: List of Experimental Cases

Case Label	**Description**	**Number of Experiments**
R	Healthy State	21 per temperature
A	$Mass_1 = 44g$	6 per temperature
B	$Mass_1 = 44g$ and $Mass_2 = 44g$	6 per temperature
C	$Mass_1 = 44g$, $Mass_2 = 44g$ and $Mass_3 = 44g$	6 per temperature
D	$Crack_1 = 5cm$	6 per temperature
E	$Crack_1 = 5cm$ and $Crack_2 = 5cm$	6 per temperature
F	$Crack_1 = 5cm$, $Crack_2 = 5cm$ and $Crack_3 = 5cm$	6 per temperature
G	$Crack_1 = 10cm$, $Crack_2 = 5cm$ and $Crack_3 = 5cm$	6 per temperature
H	$Crack_1 = 10cm$, $Crack_2 = 10cm$ and $Crack_3 = 5cm$	6 per temperature
I	$Crack_1 = 10cm$, $Crack_2 = 10cm$ and $Crack_3 = 10cm$	6 per temperature
J	$Crack_1 = 15cm$, $Crack_2 = 10cm$ and $Crack_3 = 10cm$	6 per temperature
K	$Crack_1 = 15cm$, $Crack_2 = 15cm$ and $Crack_3 = 10cm$	6 per temperature
L	$Crack_1 = 15cm$, $Crack_2 = 15cm$ and $Crack_3 = 15cm$	6 per temperature

Cases from "D" to "L" simulate fatigue scenarios with physically induced cracks of different lengths and locations on the structure, as seen in Table 3.2.

The simulated scenarios were chosen based on experimental and simulation-based contributions to wind turbine damage.

Figure 3.2 illustrates the accelerometer measurements excited by a sine sweep in both the time and frequency domains. Analyzing the data in the time domain reveals that the presence of cracks (case L) causes a shift in the natural frequencies of the blade, resulting in stronger vibrations at those frequencies.

The frequency domain analysis demonstrates the effect of cracks on both bending and torsional modes, as indicated by [84]. Considering both the frequency and amplitude of the signal, it is possible to conclude that the influence of the cracks is more pronounced in the torsional mode.

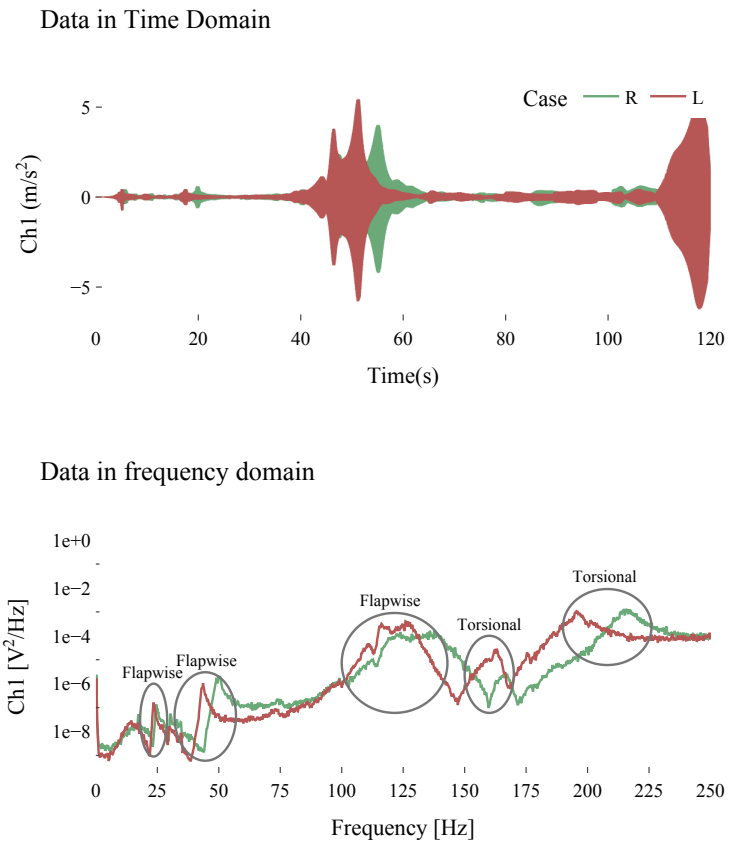


Figure 3.2: Time history of acceleration and power spectral density for accelerometer 1 in case L

The publicly available dataset contains accelerometer and strain gauge data separated for each experiment and for each temperature, all in the time domain.

3.2 Shallow Learning Models

A shallow model, in the context of machine learning and neural networks, typically refers to a model with a limited number of layers or parameters compared to deep neural networks. Shallow models are characterized by their simplicity and computational efficiency, often consisting of only one or two layers of neurons. Overall, shallow models provide a trade-off between computational efficiency and modeling complexity, making them a valuable tool in various machine learning applications.

To train the shallow learning models, the dataset includes raw accelerometer data collected from all 8 accelerometer channels, as elaborated in Section 3.1. The healthy dataset incorporates data from cases R, A, B, and C. Conversely, the unhealthy dataset encompasses data from cases D, E, F, G, H, I, J, K, and L, corresponding to instances where cracks were detected.

Figure 3.3 illustrates the process of ingesting data to create the models.:

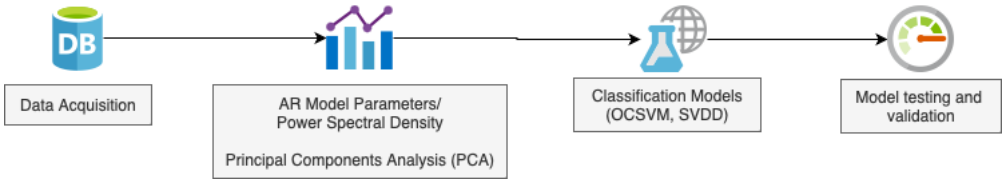


Figure 3.3: Shallow learning models data ingestion pipeline

3.2.1 Feature Extraction

To enhance the preprocessing of data and improve the accuracy of the trained Shallow Learning models (Support Vector Machine and Support Vector Data Descriptor), two methods were applied:

- AutoRegressive model (AR) and Principal Component Analysis (PCA)
- Power Spectral Density (PSD) and Principal Component Analysis (PCA)

The choice to examine both AutoRegressive models and Power Spectral Density was driven by the aim to assess the time series data from two distinct perspectives. AR models emphasize the temporal relationships between data points, facilitating an understanding of sequential patterns and trends [85]. On the other hand, PSD emphasizes the frequency domain characteristics of the data. By utilizing both approaches it is possible to select which method better captures the characteristics of the dataset.

In order to generate the AR models, it is important to define the number of data points that will be taken into consideration, known as lag. For the purpose of this work, it was tested lags of 5, 10, 15, 20, 30, 40 and 50.

From Figure 3.4 it is possible to analyse the accuracy of each model"s predictions based on RMSE and assess the complexity of each model using AIC. Observing the median RMSE values, except for the AR(5) model, all other models exhibit similar accuracy. However, the AR(10) model appears slightly less consistent than the others, displaying more outliers and a larger interquartile interval. Since there were no obvious improvements in RMSE for models with lags greater than 15, it is possible to conclude, from the RMSE perspective, that the AR(10) and AR(15) models are the most suitable.

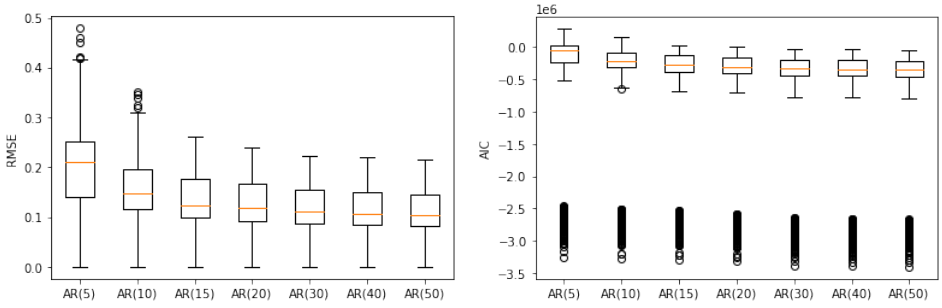


Figure 3.4: Comparison of AR Model Performance: RMSE and AIC Analysis for Different Lags

When evaluating the complexity of the models based on the AIC boxplots depicted in Figure 3.4, it is evident that the AR(10) model exhibits a lower median AIC value compared to AR(15). Therefore, the AR(10) model is selected as the preferred model for training the shallow learning models.

After implementing AR and PSD on the data, Principal Component Analysis (PCA) is applied to reduce the dimensionality of the new dataset with the goal to eliminate dimensions that exhibit correlation while retaining the variance of the data. In this study, the decision was made to retain 95% of the variance of the data, which reduced the number of components from 80 to 8 and 7 for AR(10) and PCA, respectively (see Figure 3.5).

Figure 3.6 illustrates the outcomes following the application of PCA to both AR(10) and PSD models. In the upper plots, it is possible to observe the capacity of each variable, represented on the x-axis, to distinguish between the healthy and anomalous states. Meanwhile, the lower plots display boxplots for each variable and each state. Upon analyzing the plots, it becomes evident that there is no distinct separation between the states across any variable, highlighting the challenge associated with classifying the turbine state as anomalous or healthy.

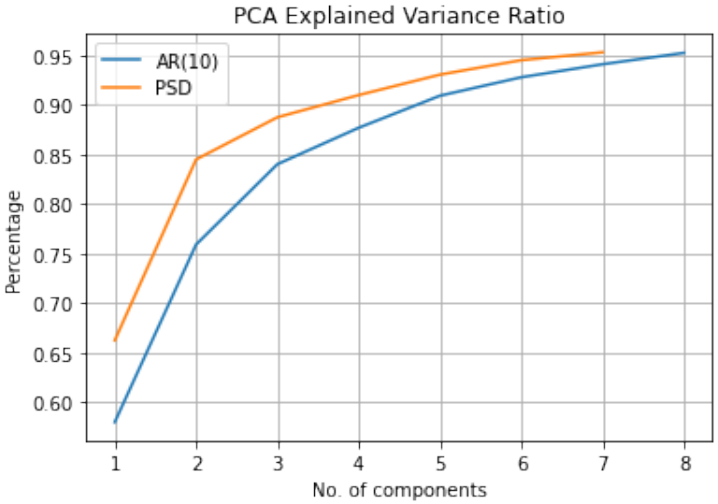


Figure 3.5: Percentage of Explained Variance vs. Number of Components in PCA Analysis

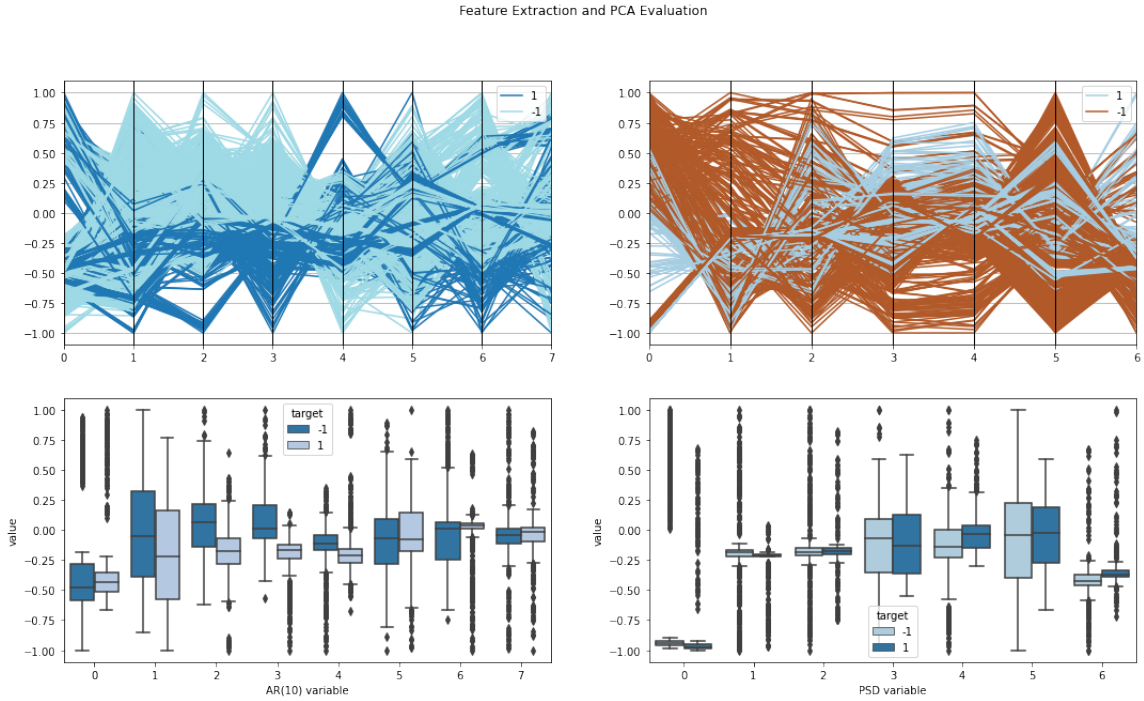


Figure 3.6: Comparative Visualization of AR(10) + PCA and PSD+PCA Models using Parallel Coordinates Plotting

3.2.2 Model Training

For training both OCSVM and SVDD models, the Monte Carlo method was used to create 100 random splits of the dataset into training and test sets. The training set was comprised of 70% of the healthy dataset, while the test set included 30% of the healthy dataset along with 30% of the unhealthy

dataset. In the end, the training dataset is 93% of the test dataset. Additionally, the split between training and validation within the training data was made using k-fold cross-validation. Lastly, the model parameters were selected using Random Search Cross-Validation (Random Search CV).

3.3 Deep Learning Models

3.3.1 LSTM Autoencoder

The initial segment of the model employs an LSTM autoencoder to learn essential features and reproduce the input in the output [86]. The subsequent part performs classification, determining whether the input represents a "healthy" or "damaged" signal based on the error in the reconstructed signal.

The primary goal of an autoencoder is to replicate the input in the output. This first section of the network consists of a series of LSTM layers, referred to as the encoder, responsible for extracting crucial features. Each LSTM layer comprises as many cells as timesteps, with each cell connected to the corresponding cell from the previous timestep in the subsequent layer [86].

The encoded feature vector (depicted as latent in Figure 3.7) serves as input to the second section of the network, the decoder. The decoder's architecture mirrors that of the encoder but in reverse order, enabling the reconstruction of the input.

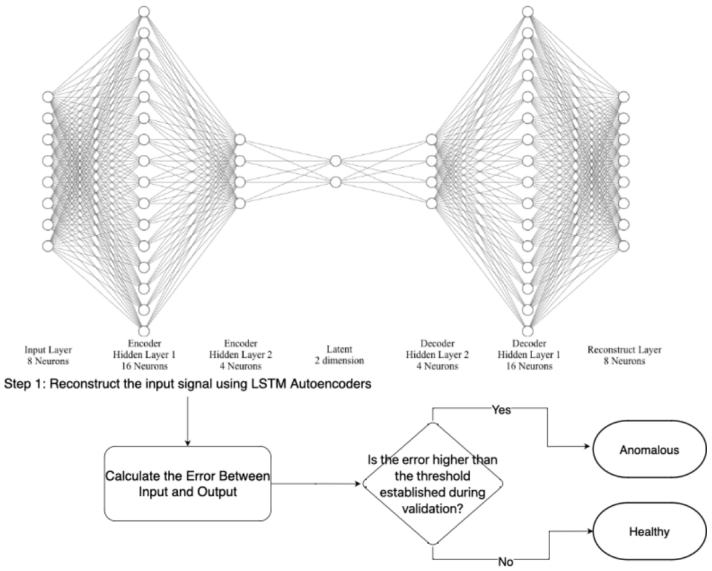


Figure 3.7: Graphical overview of the proposed anomaly detection model (LSTM Autoencoder)

In the context of anomaly detection, the model is trained solely on healthy data, resulting in a small expected reconstruction error. However, when tested with damage data, which it has not been trained to predict, the error tends to increase. Therefore, during the model validation phase with healthy data, a threshold for the error between the input and reconstructed output is established to detect anomalies.

The training dataset includes raw data from cases R, A, and B (see Table 3.2 from Section 3.1), covering all temperature settings, identified as healthy data. Specifically, measurements from the eight accelerometers are utilized, excluding temperature and strain data, with a 20% split for validation.

The test dataset includes data from cases C, designated as healthy for this study, and cracked data from both less damaging and worst-case scenarios (cases "D" and "L" listed in Table 3.2 from Section 3.1).

The measured signals were downsampled from 1666 Hz to 166 Hz and then standardized using a Gaussian distribution with a mean of 0 and a variance of 1. As a result, the training dataset comprised $14,332,131$ samples, while the test dataset comprised $6,142,341$ samples.

The model structure was initialized with 4 LSTM layers: 2 for encoding and 2 for decoding. The input consisted of 8 features, one for each accelerometer time series, and ten timesteps. The first LSTM layer produced 16 features for each timestep, which were then reduced to 4 in the second layer. The model was instantiated and compiled using Adam as the neural network optimizer and mean absolute error to calculate the loss function.

The number of parameters required to train the model described in Table 3.3 constitutes only 0.02% of the total number available in the training dataset.

Table 3.3: Summary of the LSTM Autoencoder Model Architecture

Layer (type)	**Output Shape**	**Parameters #**
Input Layer	[(None, 10, 8)]	0
LSTM 1	(None, 10, 16)	1600
LSTM 2	(None, 4)	336
Repeat Vector	(None, 10, 4)	0
LSTM 3	(None, 10, 4)	144
LSTM 4	(None, 10, 16)	1344
Time Distributed	(None, 10, 8)	136
Total Params: 3,560		
Trainable Params: 3,560		
Non-Trainable Params:0		

3.4 Results

3.4.1 One Class Support Vector Machine (OCSVM)

The model employs a specialized variant of Support Vector Machines (SVM) known as One-Class SVM (OCSVM). To ensure robustness and reliability in the evaluation of the One-Class SVM model, a Monte Carlo simulation approach was employed to split the train and test data 100 times, which helps in understanding the variability and stability of the model"s performance under different data conditions. This method provided a more comprehensive set of accuracy values, allowing for a detailed statistical analysis of model performance across multiple scenarios, as shown in Figure 3.8.

In Figure 3.8, the accuracy of each feature extraction method, PSD and AR, is compared to evaluate the consistency and performance of the models. By examining the median accuracy for each method, it is evident that the OCSVM models using AR parameters with PCA achieved better overall results than those using PSD, even in the presence of outliers. The median accuracy was 0.83 for the AR method, compared to 0.63 for the PSD method, representing a 30% higher accuracy for the AR method.

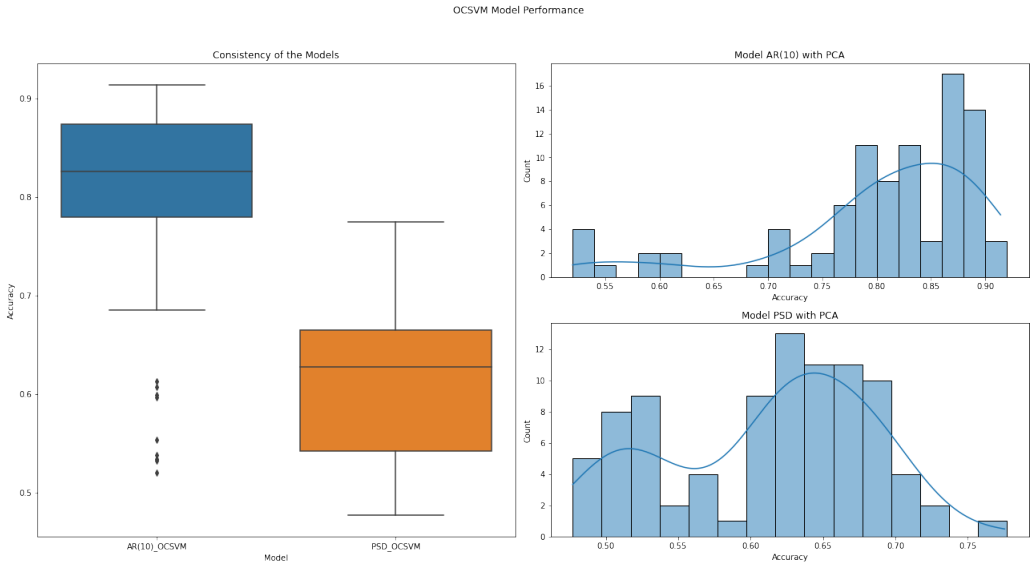


Figure 3.8: Evaluation of the One Class Support Vector Machine model"s performance was conducted across various hyperparameter combinations and splits of the test and training data.

Additionally, comparing the interquartile range of both models and their histograms, it can be concluded that the AR models consistently produced

higher accuracy results than the PSD models. Specifically, 75% of the AR models had accuracy values higher than 78%, whereas only 75% of the PSD models had accuracy values higher than 52%, with the PSD results showing much greater dispersion.

3.4.2 Support Vector Data Description (SVDD)

As described in 3.4.1, a Monte Carlo simulation approach was employed to split the train and test data 100 times in order to help in understanding the variability and stability of the model"s performance under different data conditions. This method provided a more comprehensive set of accuracy values, allowing for a detailed statistical analysis of model performance across multiple scenarios, as shown in Figure 3.9.

In Figure 3.9, the accuracy of the feature extraction methods, Power Spectral Density (PSD) and AutoRegressive (AR) modeling, is compared to evaluate the consistency and performance of the SVDD models. The analysis reveals that SVDD models utilizing PSD parameters with Principal Component Analysis (PCA) achieve better overall results than those using AR parameters, which contrasts with the findings in 3.4.1. Specifically, the median accuracy for the AR method is 0.60, while the PSD method achieves a median accuracy of 0.66, indicating an 11% higher accuracy for the PSD method.

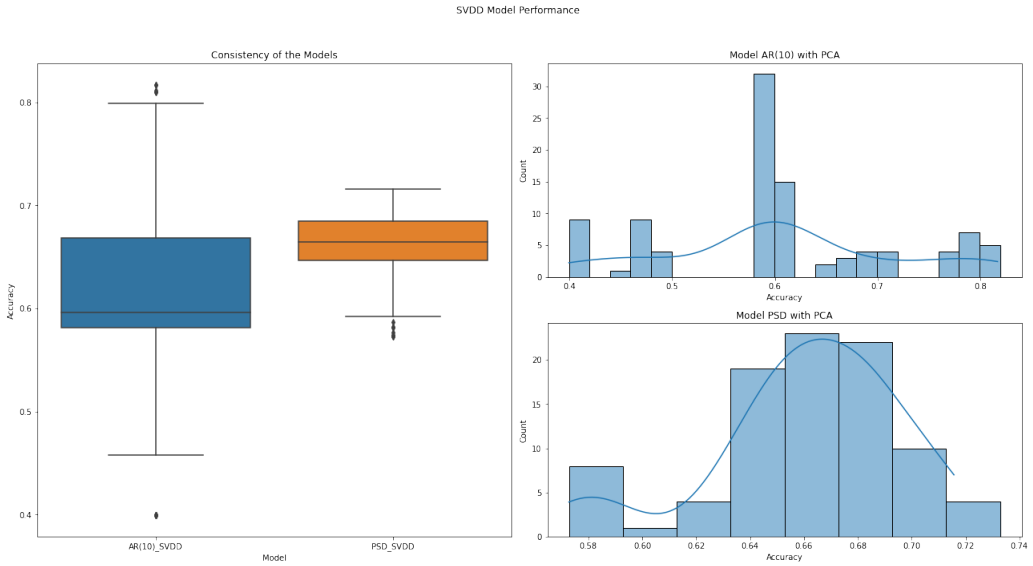


Figure 3.9: Evaluation of the Support Vector Data Description model"s performance was conducted across various hyperparameter combinations and splits of the test and training data.

Additionally, comparing the interquartile range and histograms of both

models, it is evident that the PSD models consistently yielded higher accuracy results than the AR models. Specifically, 75% of the AR models had accuracy values above 58%, while 75% of the PSD models had accuracy values exceeding 64%. Moreover, the AR results exhibited greater dispersion, indicating a wider range of accuracy values compared to the PSD models.

3.4.3 LSTM Autoencoder

The LSTM Autoencoder model was trained for ten epochs with a batch size of 100 and L2 regularization. Training was performed on a GPU, and the entire training process took approximately 25 hours to complete.

Since the model was trained using healthy data, a small error is expected when comparing the autoencoder output with the actual measured accelerometer values for cases where the blade is operating in a normal state. However, the autoencoder should not accurately reconstruct the measurements for a blade with cracks, resulting in higher error values. Thus, for the classification portion of the model, a threshold is set. The threshold is chosen based on the distribution of the mean squared error of the validation set, as shown in Figure 3.10.

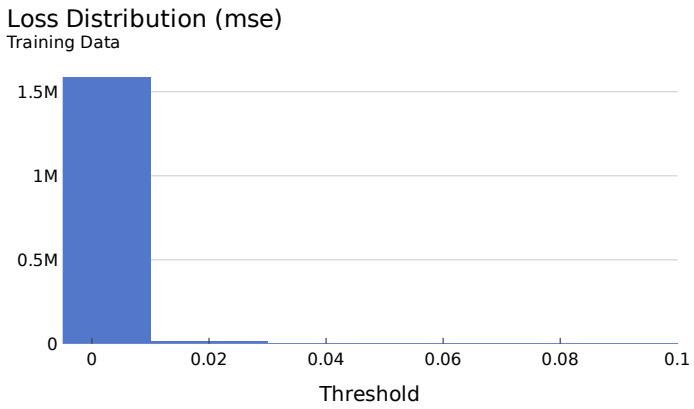


Figure 3.10: Loss distribution of the validation data

Due to the much higher cost of blade repair and replacement compared to the cost of an inspection to detect the cracks, this model prioritizes false positives over false negatives, therefore chosen threshold value is 0.005. Thus, the threshold is set slightly below the "noise level," and it is expected to trigger some false positives.

After classifying the samples based on the reconstruction error, the model achieved an accuracy of 97.4%. Given the choice to prioritize false crack classifications over false normal classifications, the number of false positives

was 5 times higher than false negatives. However, this proportion is still small (2.21%), indicating that the vast majority of the data was correctly classified. This demonstrates the model"s ability to accurately identify healthy (uncracked) blades in various conditions, as seen in Figure 3.11.

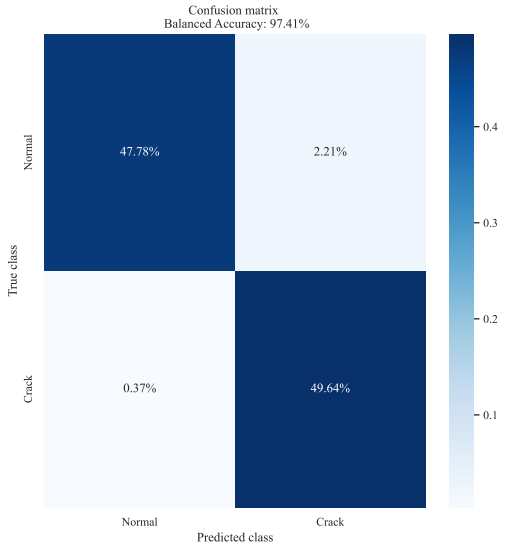


Figure 3.11: Confusion Matrix

3.5 Discussion

To compare the models trained above, the best-performing parameters for each shallow learning model were selected, trained and tested on a arbitrary selection of the random state used to split the train and test data. So, the OCSVM model used the dataset with Auto Regressive parameters as described in 3.4.1, the SVDD model uses the dataset with PSD feature extraction as described in 3.4.2 and the LSTM Autoencoder model uses raw data as described in 3.4.3.

Table 3.4 shows both the accuracy and the recall for each one of the trained models. In the use case of a wind turbine blade, failing to detect a crack in the early stages is significantly more costly than inspecting the blade and finding no damage, hence the importance of also evaluating the recall.

Table 3.4: Comparison of Accuracy and Recall Between OCSVM, SVDD, and LSTM Autoencoder Models

Model	**Accuracy**	**Recall**
LSTM Autoencoder	97%	100%
OCSVM	89%	97%
SVDD	70%	54%

Comparing the models, it is evident that the SVDD has the worst performance in both accuracy and recall, making it unsuitable for the proposed application. When comparing the LSTM autoencoder with OCSVM, the LSTM shows better overall performance. However, this improvement is more pronounced in terms of accuracy than recall, indicating that the OCSVM model has a higher proportion of false positives compared to the LSTM but a similar proportion of true positives and false negatives, as can be seen in Figure 3.12.

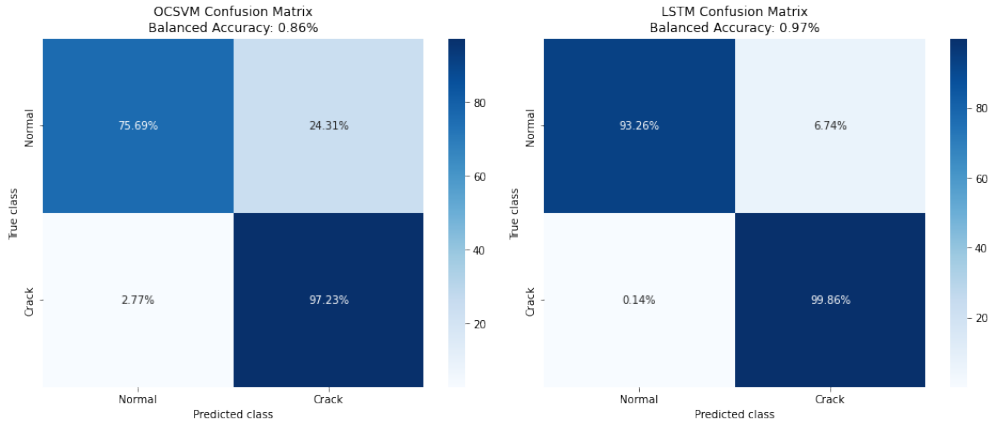


Figure 3.12: Comparison of OCSVM and LSTM Autoencoder Models Confusion Matrices

Two of the three methods evaluated proved successful for the studied dataset, with the LSTM Autoencoder model achieving 97% accuracy and the OCSVM model achieving 89% accuracy. The LSTM Autoencoder results are comparable to the 99% accuracy achieved by Calderano et al. [30], which used a fully labeled dataset. However, while the overall performance of the LSTM Autoencoder model was better than that of the OCSVM, the improvement is not sufficient to justify the higher costs of training a deep learning model, which requires GPUs and more memory. Therefore, the most suitable model for this use case is the OCSVM.

4
Plastic Strain Accumulation Prediction

Section 4.1 outlines the experimental arrangement for measuring plastic strain accumulation in polycrystalline materials. Section 4.2 details the preprocessing techniques while Sections sec: mlp covers the architecture and training of Multilayer Perceptron (MLP) models and 4.4 discusses the development of Convolutional Neural Network (CNN). Finally, Section 4.5 presents the results of the predictive models, comparing their performance against previous works. Subsections 4.5.1, 4.5.2 and 4.5.3 detail the results for each sample, with a comprehensive discussion provided in section 4.6. In this chapter, the main emphasis is placed on predicting residual strain in material microstructures using data augmentation to reduce the number of experiments necessary for training reliable models.

4.1
Polycrystalline Plastic Strain Accumulation Measurements

The data set used in this study was created through experimental research by Vieira & Lambros [42] using three hourglass samples of alloy 709 that underwent a heat treatment process at 1200^oC for 48 hours to increase grain size. Each sample was loaded to 250MPa for 1 hour at room temperature.

To measure residual mantle strains, the Digital Image Correlation (DIC) technique was applied. Digital Image Correlation is a non-destructive optical technique employed for determining the displacement and strain fields within a material. This method involves capturing a series of images to analyze the shifting patterns post-deformation. To ensure accurate measurements, high-resolution images are essential, achieved here using a microscope with 40-times magnification. Despite each image having a resolution of approximately $0.09\,\mu$m/pixel, the field of view is insufficiently covered. Consequently, overlapping images were captured and subsequently stitched together to encompass the entire area of interest.

In order to obtain grain boundary angles, it was applied a technique called electron backscatter diffraction (EBSD) scans. EBSD is a technique based on scanning electron microscopy, meaning that a focused electron beam is directed on the sample surface and the orientation of the crystal grains is

obtained based on the diffraction pattern. Lastly, both datasets were aligned using five fiducial Vickers markers, at the edges of the regions of interest. Each one of the three samples composes one dataset with the local geometric angle of the grain boundary (α) and the residual plastic strains at the three directions (ϵ_{nn} normal, ϵ_{tn} tangential-normal and ϵ_{tt} tangential).

4.2 Data Preprocessing

As described in Section 4.1, the experiment conducted by Vieira & Lambros [42] generated one image depicting the geometric angles of the grain boundaries, and another image showing the residual strain values in each of the three principal stress directions (ϵ_{nn} normal, ϵ_{tn} tangential-normal and ϵ_{tt} tangential) for each of the three samples.

As a first pre-processing step the input data (geometric angle of the grain boundary) was divided into 35x35 squares to augment the dataset for training and testing purposes. The predicted variables were the strains at the center of each square in all three directions.

After generating the dataset, each individual dataset was randomly split into training and testing sets at a 30/70 ratio, and used to train models **MD1** (training dataset composed of Sample 1 data), **MD2** (training dataset composed of Sample 2 data) and **MD3** (training dataset composed of Sample 3 data). Furthermore, a fourth model was constructed using 33% of the training data from each sample, denoted as **MD4** (training dataset containing data from all samples).

Subsequently, a second version of each data set was created using data augmentation techniques to generate new syntetic data based on the measured angles and residual stresses for each sample. The data augmentation method was chosen according to the type of input required by the model being trained: for MLPs, which require individual data points for each neuron, the data was augmented using Gaussian noise; for CNNs, which take images represented as matrices, augmentation was performed using image manipulation techniques. The details of the augmentation process will be described in subsections 4.3 and 4.4, respectively.

In summary, each deep learning model was trained using the following training data set, experimental and with data augmentation:

- Model 1 (MD1): Sample 1 data
- Model 2 (MD2): Sample 2 data
- Model 3 (MD3): Sample 3 data
- Model 4 (MD4): 33% of data from Samples 1, 2 and 3 training data set

4.3 Multilayer Perceptron Models

As described in Section 4.2, the MLP model was trained using both experimental and augmented data sets. Since the MLP network takes individual datapoints as input, the 35x35 squares were reshaped as vectors, ending up with size 1225. For the MLP network, the data augmentation procedure involved adding Gaussian noise to the training data with a 30% probability. This means that, on average, 30 out of every 100 data points were augmented with noise.

The added Gaussian noise was determined based on the standard deviation parameter. The standard deviation determines how much the augmented data points deviates from their original values and it should be carefully selected to avoid causing excessive distortions to the data. In this model, the standard deviation was set to 0.2, representing 25% of the mean angle value. Therefore, the experimental training data sets have input shape of $(x, 1225)$ and target shape of $(x, 3)$ while the augmented training data sets have input of $(1.3x, 1225)$ and target $(1.3x, 3)$.

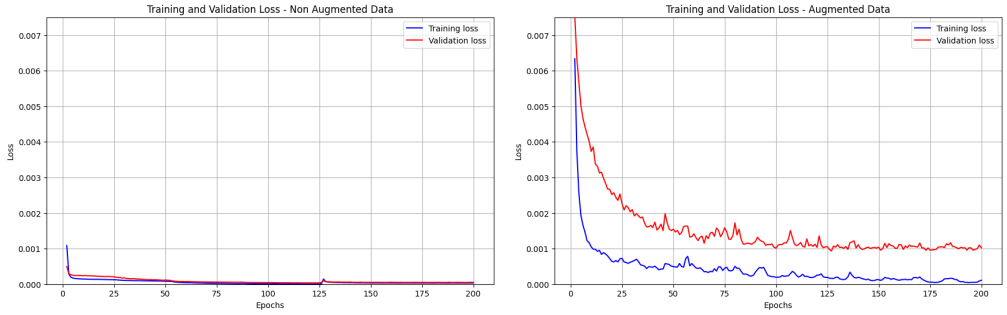
After creating the data sets, the MLP architecture is described in Table 4.1:

Table 4.1: Multilayer Perceptron Models Architecture

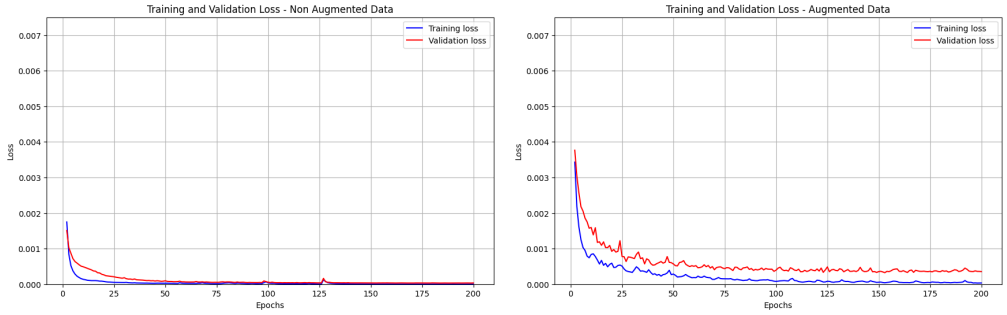
Layer (Type)	**Output Shape**
Input (Flatten)	(1225,)
Dense 1 (512, relu)	(512,)
Dense 2 (256, relu)	(256,)
Dense 3 (128, relu)	(128,)
Output (Dense, no activation)	(3,)

The first layer, a dense layer, flattens the input into a 1-dimensional array, where each value from the 35x35 square is individually fed as input to each neuron. Subsequently, the data is fully connected to dense layers that employ the ReLU activation function. Finally, a dense layer with a three neurons is used to output residual stress in each direction. Both the experimental and the augmented dataset is trained in the same MLP architecture.

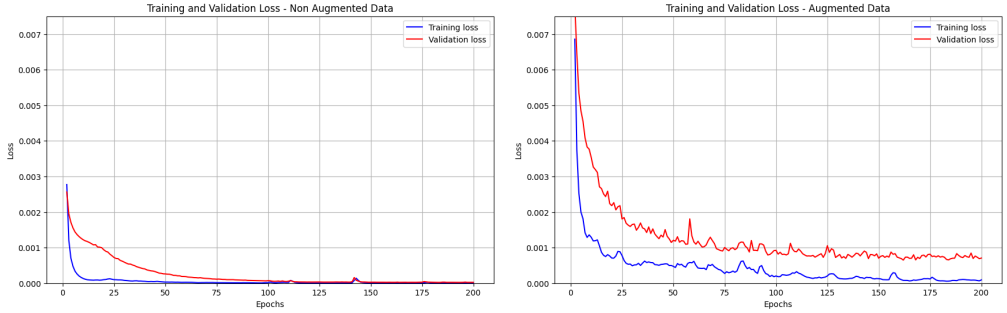
As illustrated in Figure 4.1, training was conducted in batches of size 64 over 200 epochs. It is evident that in both cases, there is no significant performance improvement beyond 100 epochs. Throughout all epochs, the validation loss consistently exceeds the training loss, suggesting that the model is not overfitting.



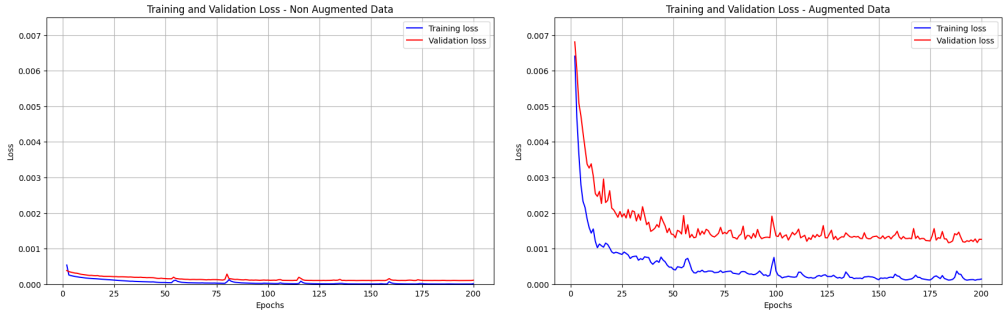
(a)



(b)



(c)



(d)

Figure 4.1: Experimental and Augmented Data Models Performance. (a) Model 1 (Sample 1 training data) , (b) Model 2 (Sample 2 training data), (c) Model 3 (Sample 3 training data), (d) Model 4 (33% of training data from each sample).

When comparing the performance of the training between the original and augmented datasets, it is possible to notice small peaks in both training and validation loss. This change in performance behaviour is mainly due to

the increased variablity in data from the augmentation process, which makes the learning process more challenging. The same can also be noticed when comparing the delta in loss between training and validation data between both original and augmented datasets.

4.4 Convolutional Neural Network Models

Different from a Multi-Layer Perceptron (MLP) model, a Convolutional Neural Network (CNN) model takes images as inputs. Therefore the data augmentation process used involved image manipulation on the 35x35 squares that were generated as described in Section 4.2. To improve the model generalization, the data augmentation was done using horizontal and vertical flipping, creating upside-down and left-right mirroing.

Additionally, a zoom of 20% was used to randomly zoom in or out of the images, enabling the model to recognize objects at different scales. The 'nearest' fill mode was utilized to fill empty areas created by these transformations, filling any new areas created in the image with pixel values that closely match the surroundings of the original image.

Also, to remain consistent with the process used on the MLP model, described in Section 4.3, 30% of the inputs were randomly chosen to be augmentated. Therefore, the experimental training data sets have input shape of $(x, 35, 35)$ and target shape of $(x, 3)$ while the augmented training data sets have input of $(1.3x, 35, 35)$ and target $(1.3x, 3)$.

The CNN network was defined as described in Table 4.2:

Table 4.2: Convolutional Neural Network Architecture

Layer (Type)	**Output Shape**
Input (InputLayer)	(None, 35, 35, 1)
Conv2D (Conv2D)	(None, 33, 33, 32)
MaxPooling2D (MaxPooling2D)	(None, 16, 16, 32)
Conv2D (Conv2D)	(None, 14, 14, 64)
MaxPooling2D (MaxPooling2D)	(None, 7, 7, 64)
Conv2D (Conv2D)	(None, 5, 5, 128)
MaxPooling2D (MaxPooling2D)	(None, 2, 2, 128)
Flatten (Flatten)	(None, 512)
Dense (Dense)	(None, 128)
Dense (Dense)	(None, 64)
Dense (Dense)	(None, 3)

The CNN model begins with an input layer that accepts images of size 35×35 pixels and a single channel (grayscale). This input is then processed through a series of convolutional and pooling layers to extract hierarchical features from the images. After the convolutional layers, the Flatten layer converts the 2-dimensional output from the last pooling layer into a 1-dimensional vector. This flattened representation serves as the input to three fully connected Dense layers, employing ReLU activation function.

As illustrated in Figure 4.2, and to remain consistent with the MLP model, training was conducted in batches of size 64 over 200 epochs. In both cases, there is no significant performance improvement beyond 100 epochs. Throughout all epochs, the validation loss consistently exceeds the training loss, suggesting that the model is not overfitting.

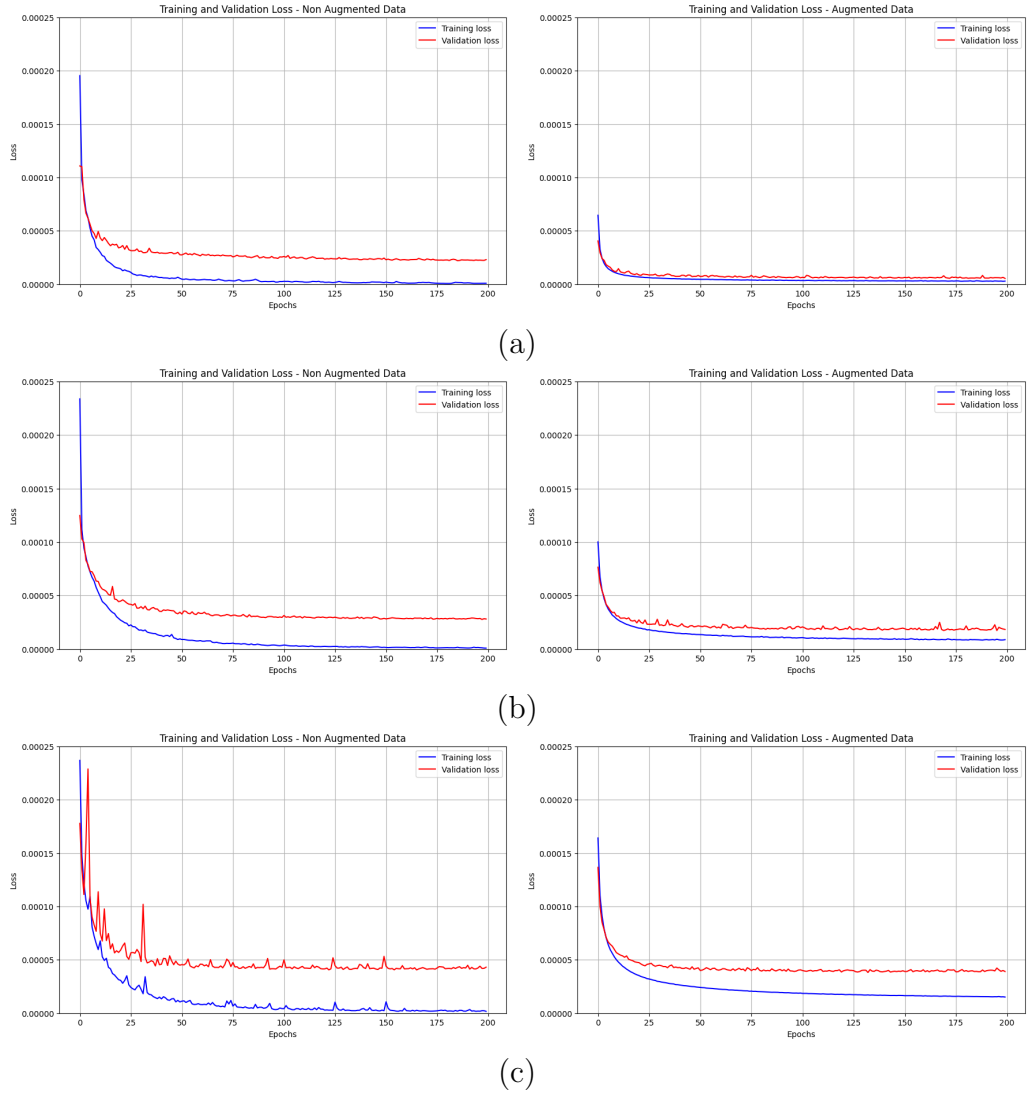
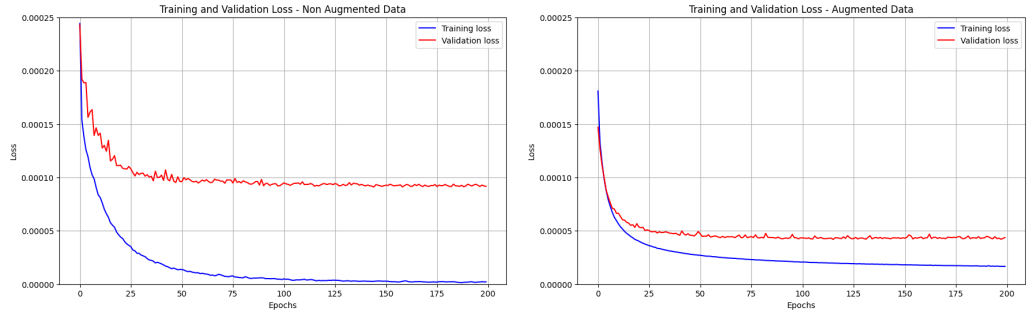


Figure 4.2: Experimental and Augmented Data Models Performance. (a) Model 1 (Sample 1 training data) , (b) Model 2 (Sample 2 training data), (c) Model 3 (Sample 3 training data).



(d)

Figure 4.3: Experimental and Augmented Data Models Performance. (d) Model 4 (33% of training data from each sample).

4.5 Results

In this section, the results of the proposed models will be presented alongside a comparison to the findings of Vieira & Lambros [42], who utilized a Multi-Layer Perceptron (MLP) with each angle value predicting a residual plastic strain, and Araujo et al. [43], who employed a shallow learning model predicting the residual plastic strain utilizing a 35x35 square centered around the location of each predicted value as input, similar to the approach adopted in this work.

Figure 4.4 shows the results for coefficient of determination of the models described on Vieira & Lambros [42], Araujo et al. [43] and the ones described on Sections 4.3 and 4.4. In order to make accurate comparisons, the coefficient of determination was computed using models trained and tested on the same sample dataset.

		Coefficient of Determination					
		Random Forest [13]	MLP [6]	MLP	MLP with data augmentation	CNN	CNN with data augmentation
	Predictor Variables	35x35 matrix with angle variables	Single angle value	35x35 matrix with angle variables	35x35 matrix with angle variables	35x35 matrix with angle variables	35x35 matrix with angle variables
	Strain						
Sample 1	Enn	0,72	0,13	0,88	0,37	0,89	0,90
	Etn	0,69	0,00	0,86	0,26	0,87	0,89
	Ett	0,78	0,15	0,90	0,40	0,90	0,91
Sample 2	Enn	0,60	0,00	0,75	0,46	0,80	0,83
	Etn	0,81	-0,35	0,88	0,71	0,93	0,93
	Ett	0,62	0,00	0,72	0,37	0,79	0,80
Sample 3	Enn	0,76	0,07	0,88	0,32	0,8	0,88
	Etn	0,7	0	0,87	0,25	0,93	0,86
	Ett	0,72	0,16	0,9	0,43	0,78	0,85

Figure 4.4: Coefficient of determination (R^2) values found for each model

The MLP without data augmentation and with matrix input had, on average, the same R^2 as the CNN without data augmentation. This value was

20% higher than the average of the random forest model and 5000 times higher than the MLP with single-angle input. The significantly poorer performance of the single-input model, compared to those utilizing more input information, demonstrates the value of the data preprocessing described in Section 4.2.

Regarding data augmentation, it reduced the performance of the MLP model by an average of 53%. In the case of the CNN, it increased performance by 2%. Among the presented results, the Convolutional Neural Network (CNN) with data augmentation exhibited the highest coefficient of determination (R^2).

4.5.1 Sample 1

In order to compare the results of residual plastic strain accumulation prediction with Vieira & Lambros [42] and Araujo et al. [43], the choosen metric was the Pearson correlation coefficient. This metric assesses whether the predicted locations of higher and lower residual strain align with actual sample data.

When comparing CNN models without data augmentation to those with augmentation, it is evident that there are no significant impacts on correlation values when testing within the same training data, as seen in Figure 4.5. However, performance on tests using different samples is significantly higher (60%), as seen on Figure 4.6. Also, for the majority of cases, the CNN model with data augmentation presented a better performance.

		Pearson Correlation					
		Random Forest [13]	MLP [6]	MLP	MLP with data augmentation	CNN	CNN with data augmentation
	Predictor Variables	35x35 matrix with angle variables	Single angle value	35x35 matrix with angle variables	35x35 matrix with angle variables	35x35 matrix with angle variables	35x35 matrix with angle variables
	Model	Sample 1					
	Strain						
Model 1 (training dataset composed of Sample 1 data)	Enn	0.87	0.37	0.37	-0.18	0.95	0.95
	Etn	0.86	0.10	0.29	-0.31	0.94	0.94
	Ett	0.90	0.39	0.51	0.22	0.95	0.95
Model 2 (training dataset composed of Sample 2 data)	Enn	-0.06	0.37	0.41	-0.10	-0.06	0.31
	Etn	-0.26	0.10	0.58	0.12	0.00	0.54
	Ett	-0.17	0.39	0.42	0.27	0.25	0.50
Model 3 (training dataset composed of Sample 3 data)	Enn	-0.33	0.37	0.33	-0.19	0.25	0.41
	Etn	0.15	-0.04	0.31	-0.34	0.05	0.40
	Ett	-0.38	0.38	0.48	0.22	0.21	0.55
Model 4 (training dataset containing data from all	Enn	-0.30	0.37	0.81	-0.21	0.85	0.83
	Etn	0.19	-0.04	0.82	0.16	0.836	0.82
	Ett	-0.20	0.38	0.79	0.32	0.88	0.86
Average		0.11	0.26	0.51	0.00	0.51	0.67

Figure 4.5: Pearson Correlation for Sample 1 predictions

As expected, models M1 (trained using Sample 1 data) and M4 (trained using Sample 4 data), which were trained on their respective datasets, achieved the best results. However, models trained on other samples showed moderately

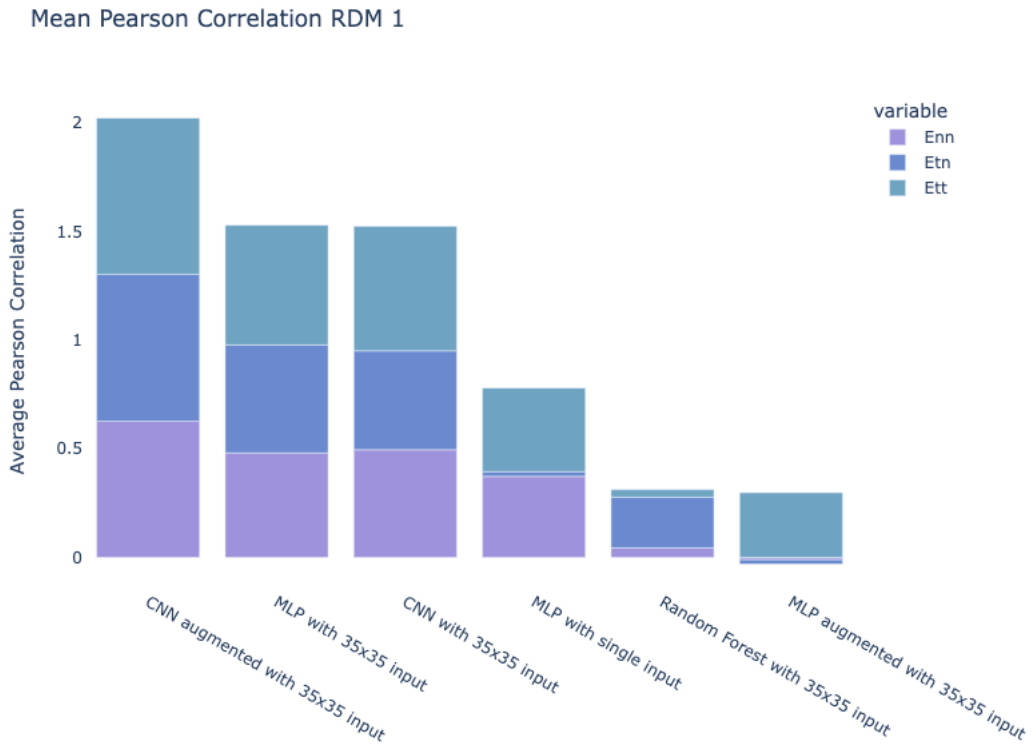


Figure 4.6: Comparison of Mean Pearson Correlation for Sample 1 predictions

positive correlations, making them suitable for evaluating the location of critical stress concentration points in cases where some precision can be sacrificed.

When evaluating the actual residual strain predictions against the true values, as shown in Figure 4.7, both Model 1 and Model 4 demonstrated accurate performance. Although Model 1, developed solely from the sample dataset, exhibited higher accuracy, Model 4 also yielded acceptable R^2 values. This underscores the significance of constructing a dataset comprising diverse samples for more accurate predictions.

Figure 4.8 presents a heatmap of microstructural strain accumulation results across different CNN models with data augmentation.

All models exhibit errors with a median very close to zero, indicating that, on average, their predictions are accurate, but with varying degrees of error dispersion. While some models display slight asymmetry, there is no strong indication that they consistently predict values significantly above or below the actual outcomes.

As seen in Figures 4.7 and 4.9, models trained with datasets consisting of data from the same sample consistently had the best predictions. The low variability in errors suggests that these models are the most precise and

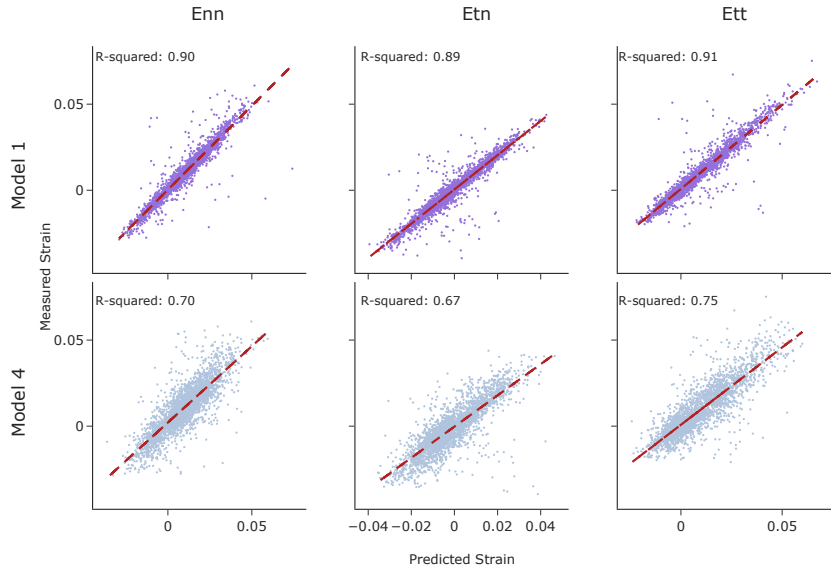


Figure 4.7: CNN testing results for Sample 1 with Models 1 and 4

consistent in the quality of their predictions.

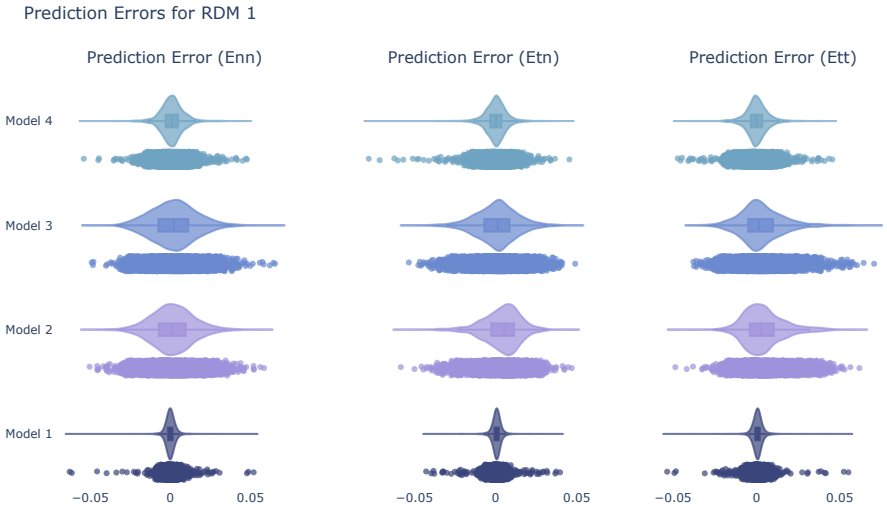


Figure 4.9: Raincloud Plots Showing Prediction Errors for RDM1 for Each Model

Among the models, Models 2 and 3 stand out for having the highest dispersion of errors. This variability indicates that these models may struggle with consistency in predicting residual plastic strain, potentially affecting their reliability in practical applications and resulting in R^2 values close to zero.

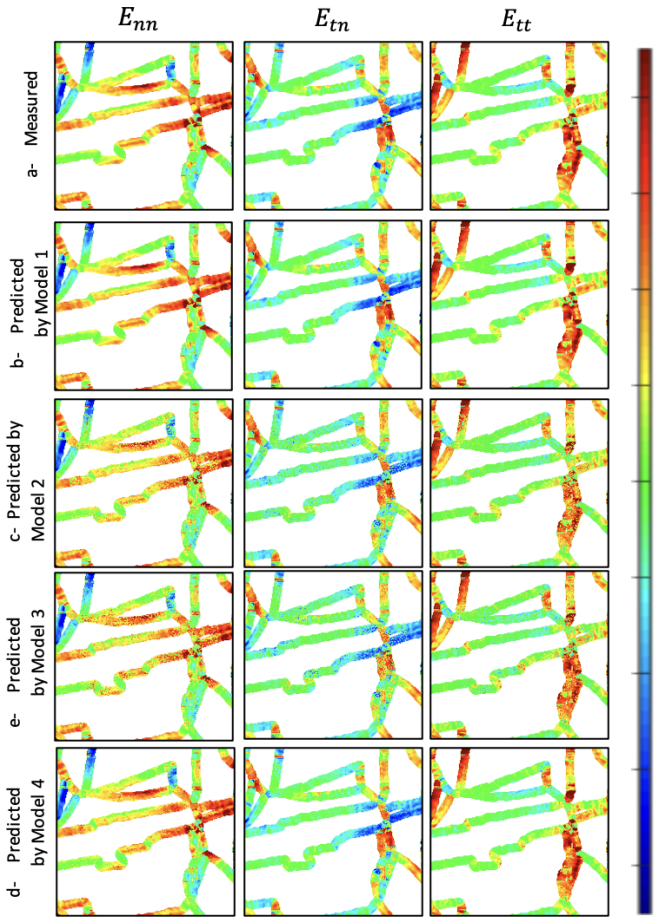


Figure 4.8: Heatmap showing microstructural strain accumulation results for various CNN models with data augmentation

4.5.2 Sample 2

As shown in Section 4.5.1, to compare results of this work with Vieira & Lambros [42] and Araujo et al. [43], the choosen metric was the Pearson correlation coefficient.

When comparing CNN models without data augmentation to those with augmentation, it is evident that there are no impacts on correlation values when testing within the same training data, as seen in Figure 4.5.2. When comparing the performance on tests using different samples, CNN with data augmentation performs 22% (Figure 4.5.2) better, however, not as significant as the 60% increase found in Sample 1 (Section 4.5.2). Nevertheless, the CNN model with data augmentation continues to exhibits superior performance across most tested scenarios.

		Pearson Correlation					
		Random Forest [16]	MLP [6]	MLP	MLP with data augmentation	CNN	CNN with data augmentation
	Predictor Variables	35x35 matrix with angle variables	Single angle value	35x35 matrix with angle variables	35x35 matrix with angle variables	35x35 matrix with angle variables	35x35 matrix with angle variables
	Model	Sample 2					
	Strain						
Model 1 (training data set composed of Sample 1 data)	Enn	0.00	0.13	0.37	-0.19	0.14	0.34
	Etn	-0.38	0.01	0.29	-0.33	0.31	0.72
	Ett	-0.10	0.11	0.51	0.23	0.26	0.42
Model 2 (training data set composed of Sample 2 data)	Enn	0.82	0.13	0.41	-0.1	0.90	0.90
	Etn	0.91	0.17	0.58	0.12	0.97	0.97
	Ett	0.82	0.11	0.42	0.27	0.89	0.89
Model 3 (training data set composed of Sample 3 data)	Enn	0.07	0.13	0.33	-0.19	0.33	0.41
	Etn	-0.47	-0.01	0.31	-0.34	0.56	0.58
	Ett	0.16	0.11	0.48	0.22	0.41	0.45
Model 4 (training data set composed of Sample 4 data)	Enn	0.07	0.13	0.81	0.45	0.77	0.80
	Etn	-0.45	-0.01	0.82	0.47	0.92	0.92
	Ett	0.15	0.11	0.79	0.47	0.78	0.81
Average		0.13	0.09	0.51	0.09	0.60	0.68

Figure 4.10: Pearson Correlation for Sample 1 predictions

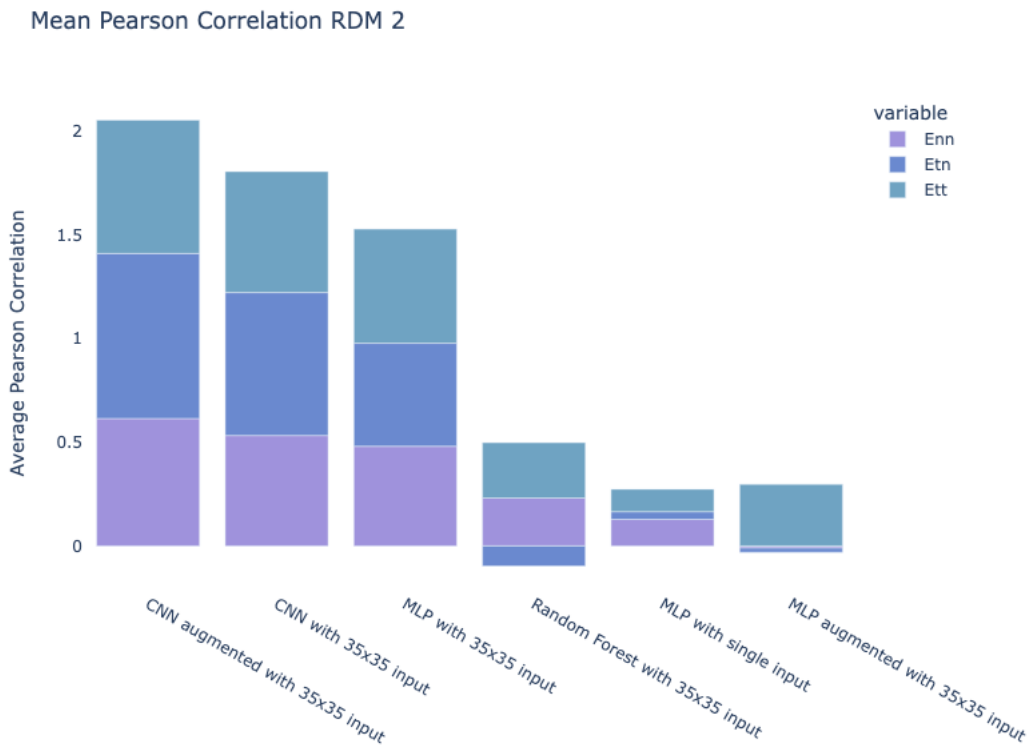


Figure 4.11: Comparison of Mean Pearson Correlation for Sample 1 predictions

In line with observations from Sample 1, models M2 and M4, trained using data from Samples 2 and using data from sample 2 respectively, consistently delivered the best performance. However, models trained on alternative samples also demonstrated moderately positive to strong correlations. This

suggests their utility in identifying critical stress concentration points where a degree of precision can be reasonably compromised.

When evaluating the actual residual strain predictions against the true values, as shown in Figure 4.12, both Model 2 and Model 4 demonstrated accurate performance. Although Model 2, developed solely from the sample dataset, exhibited higher accuracy, Model 4 also yielded acceptable R^2 values.

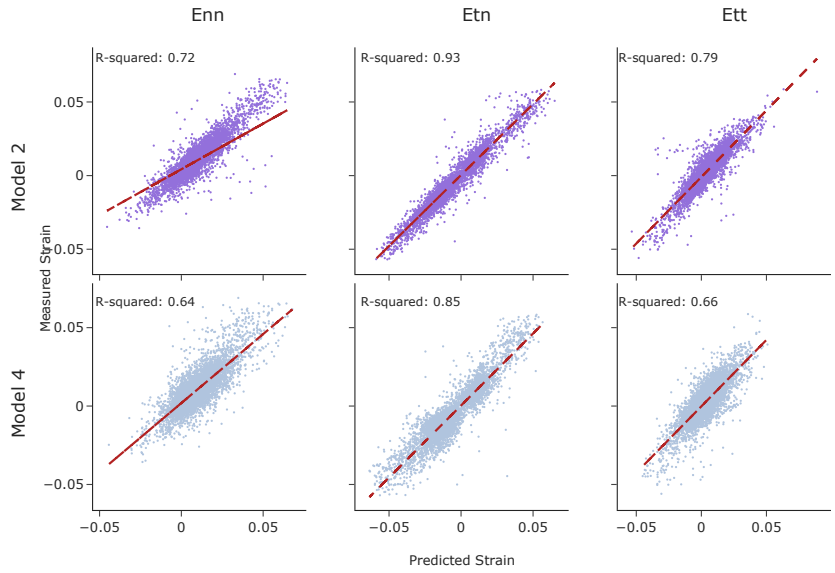


Figure 4.12: CNN testing results for Sample 2 with Models 2 and 4

Figure 4.13 presents a heatmap of microstructural strain accumulation results across different CNN models with data augmentation.

As shown in the previous section and reaffirmed in the current results depicted in Figure 4.14, models developed with dataset-specific training continue to achieve the most accurate predictions. The consistency of this finding across different samples further supports the efficacy of tailored datasets in improving model performance.

Just as in Section 4.5.1, all models showed a median error very close to zero. No model consistently over- or under-predicted beyond acceptable margins. Models 1 and 3 exhibited the highest error dispersion among all models, as highlighted in the last section. This dispersion suggests potential challenges in maintaining consistent prediction quality for residual plastic strain, which may compromise their reliability in certain applications and result in lower R^2 values.

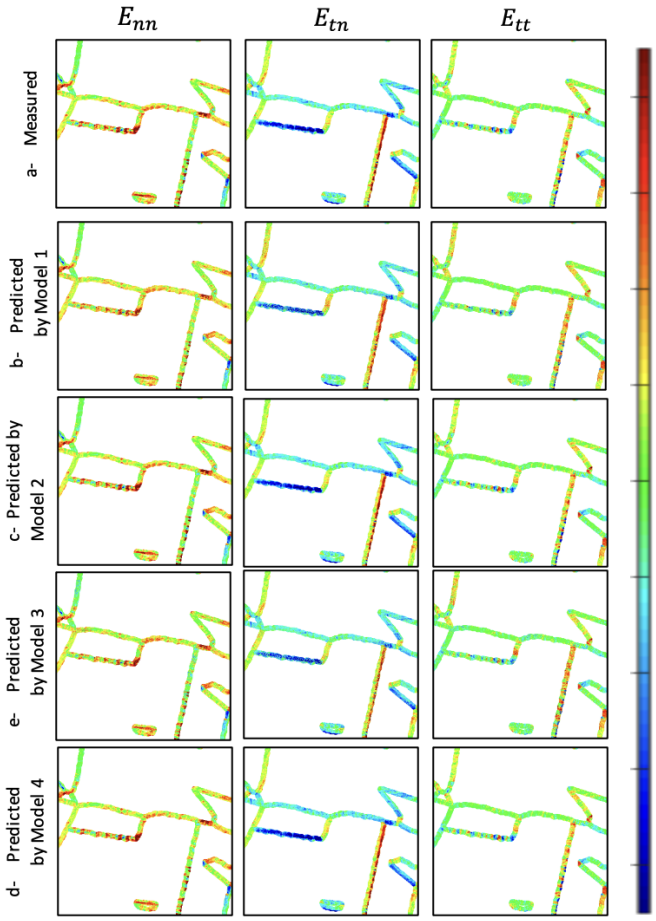


Figure 4.13: Heatmap showing microstructural strain accumulation results for various CNN models with data augmentation

4.5.3 Sample 3

As shown in Section 4.5.1 and 4.5.2, to compare results of this work with Vieira & Lambros [42] and Araujo et al. [43], this section compares the Pearson correlation coefficient for the trained models.

As highlighted earlier, CNN models with and without data augmentation showed comparable performances when tested on datasets from the same sample as their training data, as seen on Figure 4.5.3. However, on average, the model with data augmentation outperformed the one without by 17%.

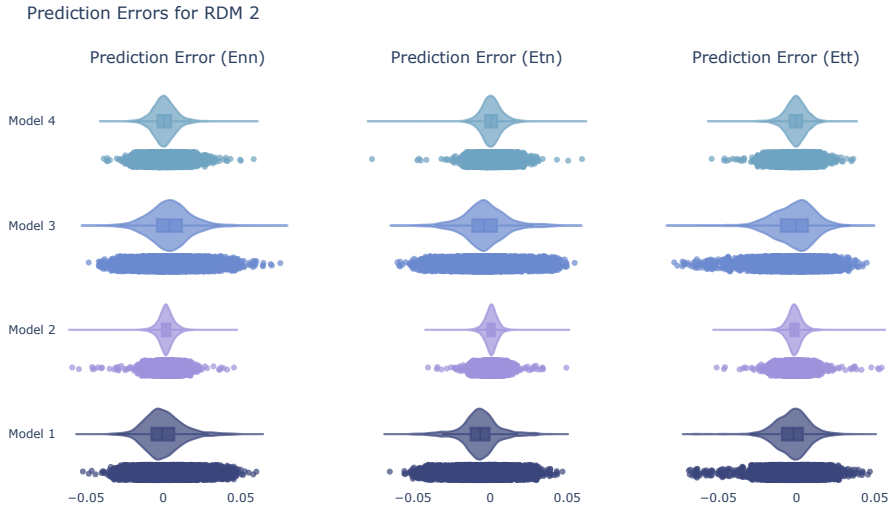


Figure 4.14: Raincloud Plots Showing Prediction Errors for RDM2 for Each Model

		Pearson Correlation					
		Random Forest [16]	MLP [6]	MLP	MLP with data augmentation	CNN	CNN with data augmentation
	Predictor Variables	35x35 matrix with angle variables	Single angle value	35x35 matrix with angle variables	35x35 matrix with angle variables	35x35 matrix with angle variables	35x35 matrix with angle variables
	Model	Sample 3					
	Strain						
Model 1 (training data set composed of Sample 1 data)	Enn	-0.34	0.33	0.37	-0.19	0.11	0.53
	Etn	0.09	0.00	0.29	-0.33	0.04	0.46
	Ett	-0.33	0.41	0.51	0.23	0.24	0.59
Model 2 (training data set composed of Sample 2 data)	Enn	0.07	0.33	0.41	-0.10	0.49	0.46
	Etn	-0.13	0.05	0.58	0.12	0.38	0.35
	Ett	0.28	0.41	0.42	0.27	0.50	0.50
Model 3 (training data set composed of Sample 3 data)	Enn	0.89	0.33	0.33	-0.19	0.95	0.94
	Etn	0.86	0.01	0.31	-0.34	0.95	0.93
	Ett	0.86	0.41	0.48	0.22	0.93	0.92
Model 4 (training data set composed of Sample 4 data)	Enn	0.48	0.33	0.81	0.45	0.86	0.91
	Etn	0.42	0.01	0.82	0.47	0.85	0.90
	Ett	0.50	0.41	0.79	0.47	0.83	0.88
Average		0.30	0.25	0.51	0.09	0.59	0.70

Figure 4.15: Pearson Correlation for Sample 1 predictions

Consistent with findings from Sample 1 and Sample 2, models trained using data from the same sample as their test data consistently achieved the highest performance. Conversely, models trained on different samples also demonstrated moderately positive to strong correlations.

When assessing the accuracy of residual strain predictions against actual values, as illustrated in Figure 4.17, both Model 3 and Model 4 showed precise performance. Model 3, made specifically for its sample dataset, achieved slightly higher R^2 values when compared to Model 4 but still not significant.

Figure 4.18 presents a heatmap of microstructural strain accumulation

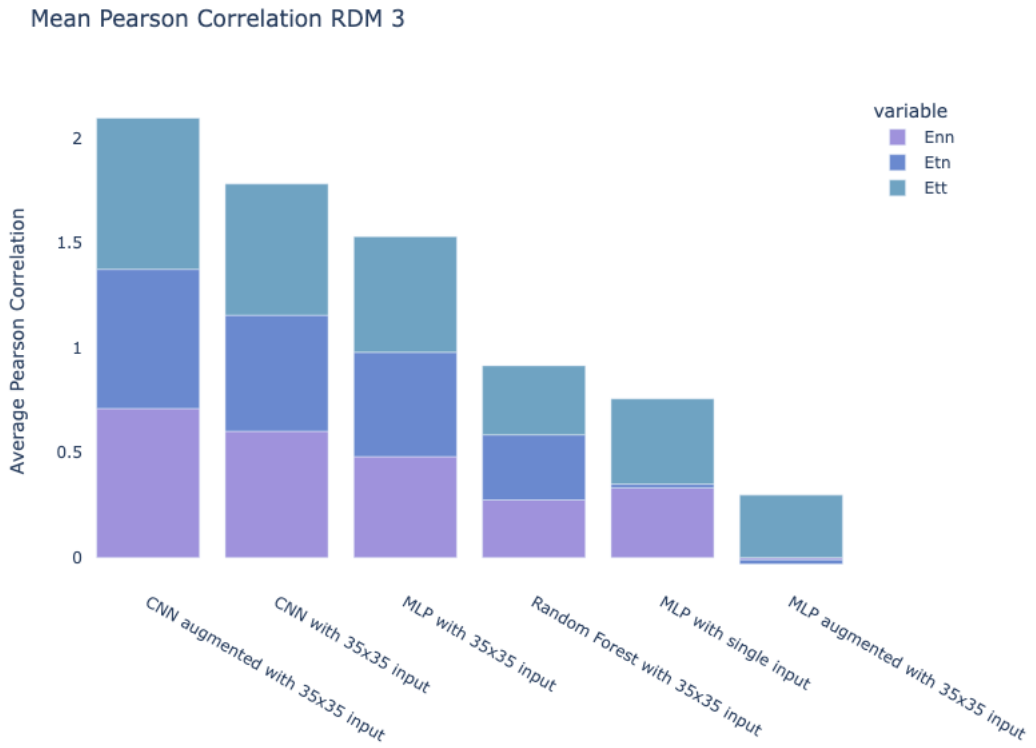


Figure 4.16: Comparison of Mean Pearson Correlation for Sample 1 predictions

results across different CNN models with data augmentation.

All models show errors with a median very close to zero, demonstrating that their predictions are generally accurate, albeit with varying levels of error dispersion. Some models exhibit slight asymmetry, but none consistently predict values significantly higher or lower than the actual outcomes.

As demonstrated in Figures 4.17 and 4.19, models trained on datasets from the same sample consistently yielded the most accurate predictions. The minimal variability in errors indicates that these models are the most reliable and precise in their predictions. Models 1 and 2 exhibited the greatest dispersion of errors among the models.

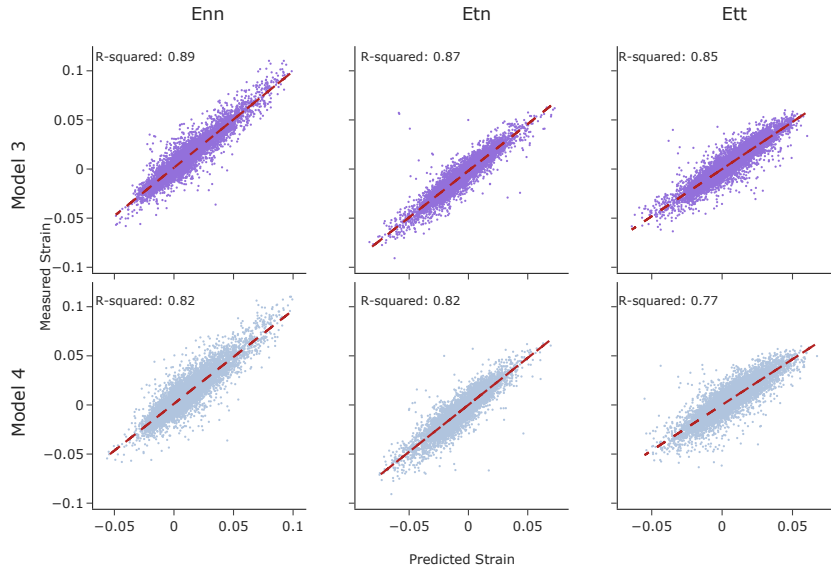


Figure 4.17: CNN testing results for Sample 3 with Models 2 and 4

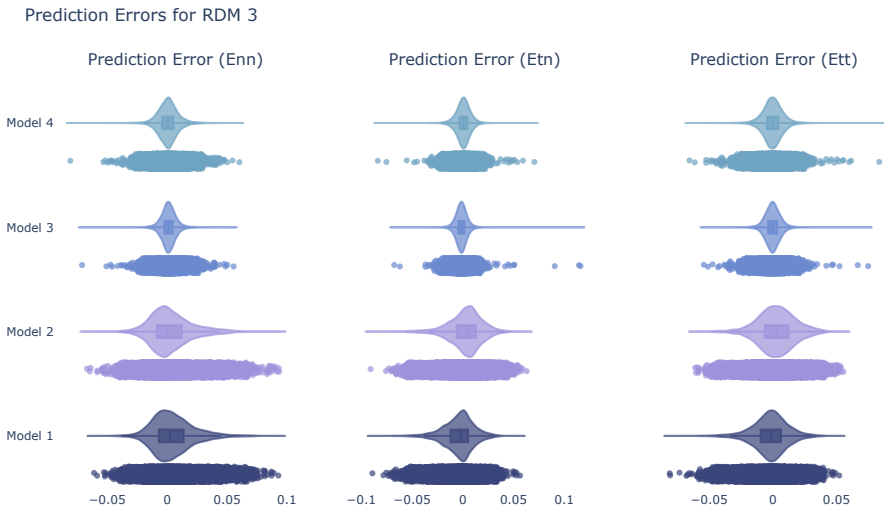


Figure 4.19: Raincloud Plots Showing Prediction Errors for RDM3 for Each Model

4.6 Discussion

This study compares augmented and non-augmented CNN and MLP models with the ones from Vieira & Lambros [42], who used a Multi-Layer Perceptron (MLP) with each angle value predicting residual plastic strain,

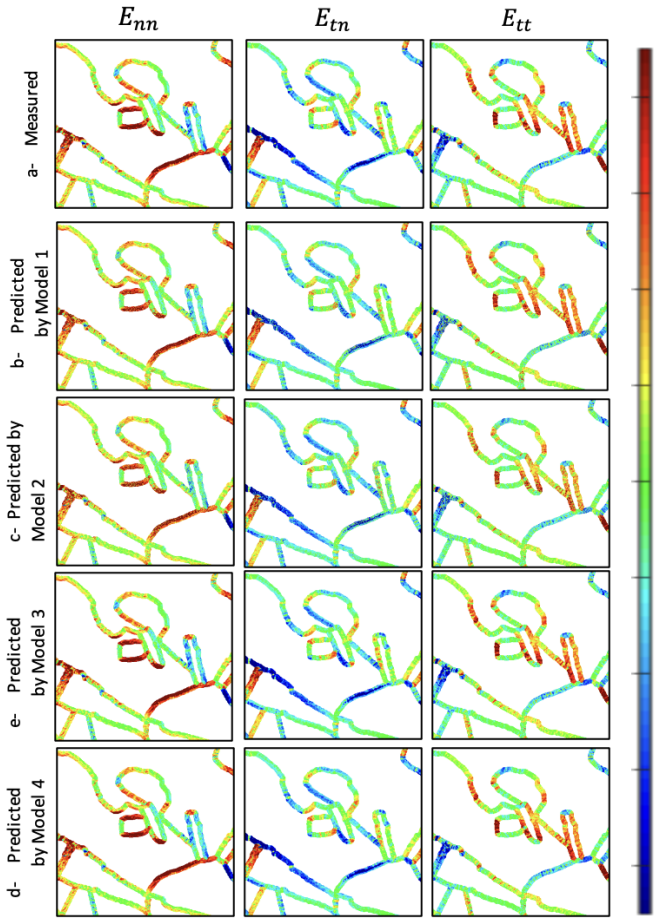


Figure 4.18: Heatmap showing microstructural strain accumulation results for various CNN models with data augmentation

and Araujo et al. [43], who used a shallow learning model with a 35x35 square input centered on each predicted value.

When comparing the coefficient of determination (R^2) the results indicated that the MLP model without data augmentation but with matrix input had a comparable R^2 value to the Convolutional Neural Network (CNN) with and without data augmentation. Both outperformed the random forest model by approximately 20% and exceeded the performance of the single-angle input MLP by 5000 times, underscoring the importance of the preprocessing steps detailed in Section 4.2.

The study's comparative analysis with previous works by Vieira & Lambros and Araujo et al. provides valuable insights into the advancements in model performance achieved through enhanced preprocessing and data augmentation techniques. Data augmentation showed mixed effects on model performance. For the MLP, data augmentation decreased performance by an average of 53%, while it slightly increased the performance of the CNN by 2%, when comparing R^2 values from models trained on the same samples as test

data. But, as highlighted in Sections 4.5.1, 4.5.2 and 4.5.3, when comparing models trained on different or mixed sample datasets, the CNN model with data augmentation achieved a higher performance, approximately 33% better than the non-augmented CNN in terms of Pearson correlation. In conclusion, the CNN model with data augmentation has proven to be the superior method for the proposed application.

5
Conclusion and Future Works

In conclusion, this work has demonstrated the effectiveness of various machine learning and deep learning models in detecting anomalies in wind turbine blades and predicting residual plastic strain.

The key findings for data preprocess when evaluating anomaly detection techniques for wind turbine blades are as follows: AR models and PSD were employed to analyze time series data from different perspectives. Neither method proved universally superior; the choice of method depends on the specific model being used. In the case studied, OCSVM achieved 30% higher accuracy using AR compared to PSD, whereas SVDD demonstrated a median accuracy that was 11% higher with the PSD model than with the AR model. When predicting residual plastic strain accumulation, the significant improvement of 5000 times in R^2 values for models with matrix input over single-angle input highlights the critical role of data preprocessing and input format in enhancing model performance.

When evaluating the impact of data augmentation for residual plastic strain accumulation prediction, it was found that while data augmentation had mixed effects on MLP models, reducing their performance by 53%, it proved beneficial for CNN models. Particularly, when tested on different or mixed datasets, CNN models achieved a performance approximately 33% higher than non-augmented models. This finding underscores the potential of data augmentation in improving the generalization capabilities of deep learning models.

In terms of model performance, various techniques were evaluated for different applications. For anomaly detection in wind turbine blades, despite the LSTM Autoencoder having a higher accuracy of 97% and a 100% recall, OCSVM was chosen for its balance of performance, with 89% accuracy and 97% recall, and computational efficiency, making it more practical for real-world applications. For predicting residual plastic strain accumulation, CNNs, particularly those with data augmentation, consistently outperformed other models, demonstrating their robustness and accuracy. The superior performance of these models in cross-sample testing further validates their effectiveness.

Overall, the findings suggest that, for blade anomaly detection, both shallow learning models like OCSVM and deep learning models such as LSTM Autoencoder are effective. Also, the research demonstrated that for applications involving the prediction of residual plastic strain, CNNs with data augmentation offer the most reliable and accurate predictions.

Future work could explore:

- Applying this method to other types of measurements, such as turbulence on wind turbine blades.
- The application of this method to other equipment and renewable sources, such as solar plants, which are subjected to different stress histories.
- Implementing real-time monitoring systems using the developed models to continuously assess the health of turbine blades.
- Developing of hybrid models that combine the strengths of shallow learning and deep learning approaches, such as using random forests with CNN for enhanced plastic strain predictions.
- Enhancing in data augmentation techniques.
- Applying transfer learning techniques to leverage pre-trained models on similar tasks, reducing the training time and computational resources.

Bibliography

1 UNITED NATIONS. **COP26: Together for our planet**. 2023.

2 Nature. The global energy crisis: Three ways to keep the lights on. **Nature**, v. 610, p. 250–253, 2022.

3 BOJEK, P. Wind electricity – analysis. 2022.

4 Associação Brasileira de Energia Eólica. **ABEEólica**. 2024.

5 REN21. **Renewables 2024 Global Status Report**. 2024.

6 BADUREK, C. A. **Wind Turbine**. 2022.

7 PENG, H. et al. Analysis of wind turbine equipment failure and intelligent operation and maintenance research. **Sustainability**, v. 15, n. 10, 2023.

8 KAEWNIAM, P. et al. Recent advances in damage detection of wind turbine blades: A state-of-the-art review. **Renewable and Sustainable Energy Reviews**, v. 167, p. 112723, 2022.

9 AZIMI, M.; ESLAMLOU, A. D.; PEKCAN, G. Data-driven structural health monitoring and damage detection through deep learning: State-of-the-art review. **Sensors**, v. 20, n. 10, 2020.

10 MARTINEZ-LUENGO, M.; SHAFIEE, M. Guidelines and cost-benefit analysis of the structural health monitoring implementation in offshore wind turbine support structures. **Energies**, v. 12, n. 6, 2019.

11 LIU, W. et al. The structure healthy condition monitoring and fault diagnosis methods in wind turbines: A review. **Renewable and Sustainable Energy Reviews**, v. 44, p. 466–472, 2015.

12 CHOU, J.-S. et al. Failure analysis of wind turbine blade under critical wind loads. **Engineering Failure Analysis**, v. 27, p. 99–118, 2013.

13 WATANABE, T. Grain boundary engineering: historical perspective and future prospects. **Journal of Materials Science**, v. 46, p. 4095–4115, 2011.

14 ROESLER, J.; HARDERS, H.; BAEKER, M. **Mechanical Behaviour of Engineering Materials**. Berlin, Heidelberg: Springer, 2007.

15 CALLISTER, W. D.; RETHWISCH, D. G. **Materials Science and Engineering: An Introduction**. Hoboken, NJ: Wiley, 2020.

16 CASTRO, J. T. P. D.; MEGGIOLARO, M. A. **Fatigue Design Techniques**. North Charleston, SC: Createspace Independent Publishing Platform, 2016.

17 SCHMIDT, J. et al. Recent advances and applications of machine learning in solid-state materials science. **npj Computational Materials**, v. 5, p. 83, 2019.

18 LI, D. et al. A review of damage detection methods for wind turbine blades. **Smart Materials and Structures**, v. 24, n. 3, p. 033001, 2015.

19 JARAMILLO, F. et al. A bayesian approach for fatigue damage diagnosis and prognosis of wind turbine blades. **Mechanical Systems and Signal Processing**, v. 174, p. 109067, 2022.

20 SUN, S. et al. Condition monitoring of wind turbine blades based on self-supervised health representation learning: A conducive technique to effective and reliable utilization of wind energy. **Applied Energy**, v. 313, p. 118882, 2022.

21 XIANG, Z.-Q. et al. Vibration-based health monitoring of offshore wind turbine towers using machine learning with bayesian optimisation. **Ocean Engineering**, v. 292, p. 116513, 2024.

22 FAN, Z. et al. Unsupervised anomaly detection method for bearing based on vae-gan and time-series data correlation enhancement (june 2023). **IEEE Sensors Journal**, v. 23, n. 23, p. 29345–29356, 2023.

23 ZHANG, H.; GUO, X.; ZHANG, P. Improved pso-svm-based fault diagnosis algorithm for wind power converter. **IEEE Transactions on Industry Applications**, v. 60, n. 2, p. 3492–3501, 2024.

24 FREMMELEV, M. A. et al. A full-scale wind turbine blade monitoring campaign: detection of damage initiation and progression using medium-frequency active vibrations. **Structural Health Monitoring**, v. 22, n. 6, p. 4171 – 4193, 2023.

25 YAN, J. et al. Wind turbine gearbox condition monitoring using hybrid attentions and spatio-temporal biconvlstm network. **Energies**, v. 16, n. 19, 2023.

26 OWOLABI, O. I. et al. Fem and ann approaches to wind turbine gearbox monitoring and diagnosis: a mini review. **Journal of Reliable Intelligent Environments**, v. 9, n. 4, p. 399 – 419, 2023.

27 DING, S.; YANG, C.; ZHANG, S. Acoustic-signal-based damage detection of wind turbine blades—a review. **Sensors**, MDPI, v. 23, n. 11, p. 4987, 2023.

28 YAN, X. et al. Machinery multi-sensor fault diagnosis based on adaptive multivariate feature mode decomposition and multi-attention fusion residual convolutional neural network. **Mechanical Systems and Signal Processing**, v. 202, 2023.

29 KHAZAEE, M.; DERIAN, P.; MOURAUD, A. A comprehensive study on structural health monitoring (shm) of wind turbine blades by instrumenting tower using machine learning methods. **Renewable Energy**, v. 199, p. 1568–1579, 2022.

30 CALDERANO, P.; MARINS, D. B. D.; AYALA, H. A comparison of feature extraction methods for crack and ice monitoring in wind turbine blades: System identification and matrix decomposition. In: **2022 30th Mediterranean Conference on Control and Automation (MED)**. [S.l.: s.n.], 2022, Paphos, Cyprus.

31 REDDY, A. et al. Detection of cracks and damage in wind turbine blades using artificial intelligence-based image analytics. **Measurement: Journal of the International Measurement Confederation**, v. 147, p. 106823, 2019.

32 LIU, Y. et al. Intelligent wind turbine blade icing detection using supervisory control and data acquisition data and ensemble deep learning. **Energy Science and Engineering**, v. 7, n. 6, p. 2633–2645, 2019.

33 REGAN, T.; BEALE, C.; INALPOLAT, M. Wind turbine blade damage detection using supervised machine learning algorithms. **Journal of Vibration and Acoustics, Transactions of the ASME**, v. 139, n. 6, p. 1–14, 2017.

34 LONG, T.; LONG, Z.; PANG, B. An end-to-end explainable graph neural networks-based composition to mechanical properties prediction framework for bulk metallic glasses. **Mechanics of Materials**, v. 191, 2024.

35 FOREST, F. et al. From classification to segmentation with explainable ai: A study on crack detection and growth monitoring. **Automation in Construction**, v. 165, p. 105497, 2024.

36 LI, X. et al. A data-driven approach for predicting the ballistic resistance of elastoplastic materials. **Engineering Fracture Mechanics**, v. 293, 2023.

37 HASAN, T.; CAPOLUNGO, L.; ZIKRY, M. A. Predictive machine learning approaches for the microstructural behavior of multiphase zirconium alloys. **Scientific Reports**, v. 13, n. 1, 2023.

38 ZHANG, S. et al. A machine learning study of grain boundary damage in mg alloy. **Materials Science and Engineering: A**, v. 867, p. 144721, 2023.

39 FRANKEL, A.; TACHIDA, K.; JONES, R. Prediction of the evolution of the stress field of polycrystals undergoing elastic-plastic deformation with a hybrid neural network model. **Machine Learning: Science and Technology**, IOP Publishing, v. 1, n. 3, p. 035005, jul 2020.

40 THOMAS, A. et al. Materials fatigue prediction using graph neural networks on microstructure representations. **Scientific Reports**, v. 13, n. 1, 2023.

41 QU, T. et al. Towards data-driven constitutive modelling for granular materials via micromechanics-informed deep learning. **International Journal of Plasticity**, v. 144, p. 103046, 2021.

42 VIEIRA, R. B.; LAMBROS, J. Machine learning neural-network predictions for grain-boundary strain accumulation in a polycrystalline metal. **Experimental Mechanics**, v. 61, n. 4, p. 627–639, 2021.

43 ARAúJO, L.; AYALA, H.; VIEIRA, R. Previsão do acúmulo de deformações plásticas em contornos de grãos de metais polycristalynelinos baseado em aprendizado de máquina. 2023.

44 CALLISTER, W. D.; RETHWISCH, D. G. **Materials Science and Engineering: An Introduction**. 10th. ed. Hoboken, NJ: Wiley, 2020.

45 SHACKELFORD, J. F. **Introduction to Materials Science for Engineers**. 8th. ed. Upper Saddle River, NJ: Pearson, 2015.

46 CASTRO, J. T. P. D.; MEGGIOLARO, M. A. **Fatigue Design Techniques**. North Charleston, SC: Createspace Independent Publishing Platform, 2016.

47 SAMAREH-MOUSAVI, S. S. et al. Monitoring fatigue delamination growth in a wind turbine blade using passive thermography and acoustic emission. **Structural Health Monitoring**, v. 0, n. 0, p. 14759217231217179, 0.

48 KDNUGGETS. Importance of pre-processing in machine learning. **KDnuggets**, 2023.

49 GéRON, A. **Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems**. [S.l.]: O'Reilly Media, 2017.

50 Neptune.ai. **Data Augmentation in Python**. 2023.

51 PANDIT, V. S. **Power integrity for I/O interfaces: With signal integrity/power integrity co-design**. [S.l.]: O'Reilly, 2011.

52 JAMES, J. F. **A Student's Guide to Fourier Transforms: With Applications in Physics and Engineering**. 3. ed. [S.l.]: Cambridge University Press, 2011. (Student's Guides).

53 ENDAQ. **Vibration Analysis: FFT, PSD, and Spectrogram Basics**. Year of Retrieval.

54 QUANTINSTI. **Autoregression: Time Series, Models, Trading, Python and more**. 2024.

55 SHUMWAY, R. H.; STOFFER, D. S. **Time series analysis and its applications: With R examples**. 4. ed. Cham, Switzerland: Springer International Publishing, 2017.

56 CHATFIELD, C.; XING, H. **The Analysis of Time Series: An Introduction with R**. 7. ed. [S.l.]: Chapman and Hall/CRC, 2019.

57 JOLLIFFE, I. T. **Principal Component Analysis for Special Types of Data**. [S.l.]: Springer, 1986. 199–222 p.

58 GéRON, A. **Hands-on Machine Learning with Scikit-Learn, Keras and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems**. 2. ed. [S.l.]: O'Reilly Media, 2019.

59 SHORTEN, C.; KHOSHGOFTAAR, T. M. A survey on image data augmentation for deep learning. **Journal of Big Data**, v. 6, p. 60, 2019.

60 DataCamp. **A Complete Guide to Data Augmentation**. 2023.

61 DOU, H.-X. et al. Patchmask: A data augmentation strategy with gaussian noise in hyperspectral images. **Remote Sensing**, v. 14, n. 24, 2022.

62 BROWNLEE, J. **How to Configure Image Data Augmentation When Training Deep Learning Neural Networks**. 2020.

63 GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. Cambridge, MA: MIT Press, 2016.

64 BERGSTRA, J.; BENGIO, Y. Random search for hyper-parameter optimization. **Journal of Machine Learning Research**, v. 13, p. 281 – 305, 2012.

65 WOLPERT, D. H. The lack of a priori distinctions between learning algorithms. **Neural Computation**, MIT Press, v. 8, n. 7, p. 1341–1390, 1996.

66 XU, Q. S.; LIANG, Y. Z. Monte carlo cross validation. **Chemometrics and Intelligent Laboratory Systems**, Elsevier, v. 56, n. 1, p. 1–11, 2001.

67 PERERA, P.; OZA, P.; PATEL, V. M. One-Class Classification: A Survey. jan 2021.

68 SCHOLKOPF, B. et al. Estimating the support of a high-dimensional distribution. In: **Neural Information Processing Systems**. [S.l.]: MIT Press, 2001. p. 582–588.

69 BERGMANN, P. et al. **Patch SVDD: Patch-Level SVDD for Anomaly Detection and Segmentation**. 2020.

70 TAX, D. M. J.; DUIN, R. P. W. Support vector data description. **Machine Learning**, v. 54, n. 1, p. 45–66, 2024.

71 VIDHYA, A. **Introduction to Long Short-Term Memory (LSTM)**. 2021.

72 HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. **Neural Networks**, v. 9, n. 8, p. 1735–1780, 1997.

73 BROWNLEE, J. **Crash Course on Multi-Layer Perceptron Neural Networks**. 2017.

74 AI, T. **Tutorial on LSTMs: A Computational Perspective**. 2020.

75 LECUN, Y. et al. Gradient-based learning applied to document recognition. **Proceedings of the IEEE**, IEEE, v. 86, n. 11, p. 2278–2324, 1998.

76 PRABHU, R. **Understanding Convolutional Neural Network (CNN) — Deep Learning**. 2024.

77 ZEILER, M. D.; FERGUS, R. Visualizing and understanding convolutional networks. **CoRR**, abs/1311.2901, 2013.

78 POWERS, D. M. W. Evaluation: From precision, recall and f-measure to roc, informedness, markedness & correlation. **Journal of Machine Learning Technologies**, v. 2, n. 1, p. 37–63, 2020.

79 IBM. **What is a confusion matrix?** 2023.

80 HE, H.; GARCIA, E. A. Learning from imbalanced data. **IEEE Transactions on Knowledge and Data Engineering**, IEEE, v. 21, n. 9, p. 1263–1284, 2009.

81 NEPTUNE.AI. **Balanced Accuracy: When Should You Use It?** 2023.

82 SCRIBBR. **Coefficient of Determination (R2) | Calculation & Interpretation**. 2023.

83 HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. **The Elements of Statistical Learning: Data Mining, Inference, and Prediction**. [S.l.]: Springer, 2009.

84 OU, Y. et al. Vibration-based monitoring of a small-scale wind turbine blade under varying climate conditions. part i: An experimental benchmark. **Structural Control and Health Monitoring**, v. 28, n. 6, p. 1–18, 2021.

85 BROCKWELL, P. J.; DAVIS, R. A. **Introduction to time series and forecasting**. 3. ed. Cham, Switzerland: Springer International Publishing, 2016.

86 MALHOTRA, P. et al. Long short term memory networks for anomaly detection in time series. In: **Proceedings of the 23rd European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)**. [S.l.: s.n.], 2016, Bruges, Belgium.