A hybrid approach to implement the Digital Twin concept into a damage evolution prediction for composite structures

Master of Science Thesis

R.S.C. Coenen

Delft University of Technology

TU Delft

This page is intentionally left blank

A hybrid approach to implement the Digital Twin concept into a damage evolution prediction for composite structures

MASTER OF SCIENCE THESIS

For obtaining the degree of Master of Science in Aerospace Engineering
at the Delft University of Technology

R.S.C. Coenen

November 5, 2021

Faculty of Aerospace Engineering · Delft University of Technology

TU Delft

The work in this thesis was supported by Dimitrios Zarouchas and the faculty of Aerospace Engineering at Delft University of Technology. Their cooperation is gratefully acknowledged.

DELFT UNIVERSITY OF TECHNOLOGY
FACULTY OF AEROSPACE ENGINEERING
DEPARTMENT OF AEROSPACE STRUCTURES AND MATERIALS

GRADUATION COMMITTEE

Dated: November 5, 2021

Chair holder:

Supervisor: Dr.Ir. D. Zarouchas

Committee members:

Dr.Ir. B. Chen

Dr.Ir. J.A. Pascoe

Dr. C. Nastos Konstantopoulos

Abstract

In the last decades, fiber-reinforced polymers (FRP) have become increasingly popular within many engineering fields. Having superior specific properties makes them valuable in industries where weight is of utmost importance, in particular the aerospace industry. However, composites are vulnerable to damage that is developed during various life-cycle stages and complex to model. Structural Health Monitoring (SHM) is a means to identify and charactere the damage, which brings advantages like improving the design or rescheduling for maintenance. The need for reliable and accurate SHM of composite materials is still existent in the present after many years, in which the high level of complexity of the damage mechanisms plays a dominant role. Thus, to achieve an improvement in the SHM of composite materials, an enhanced understanding of the damage accumulation processes is essential and facilitates the way towards lighter, more optimized, and more sustainable aerospace structures.

Another trend, especially in the last decade, is the growth in the availability of data, having introduced and slowly manifested itself in society in the form of the Internet of Things, Big Data, and Industry 4.0. It has become increasingly popular amongst researchers to use this increase in available data to increase the reliability of SHM. Two particularly useful topics show to be promising tools that further stimulate the ongoing trend in the use of data: machine learning (ML) algorithms and the Digital Twin (DT). concept. One of the main advantages of using ML is that it is much faster to operate than a computationally expensive physics-based model, such as a finite element model. The Digital Twin concept entails, in short, creating a virtual copy of a system (its physical twin) that makes use of an integrated, heterogeneous information database and is updated accordingly in real-time, with the purpose of increasing the user's knowledge of the system and allowing to act accordingly.

There are three trends ongoing that, once converged, may achieve a synergy effect: development of SHM using composite materials, the Digital Twin concept, and machine learning algorithms. By making a first step towards the synergy effect, this thesis aims to enhance our understanding of damage evolution in composites and improve on health assessment during its service life by constructing a digital twin of a composite material. The presented methodology overcomes the need for expensive testing to make use of the benefits of a data-driven digital twin, which focuses on the real-time prediction of damage accumulation within a composite specimen. Successful integration of models and data to create a digital twin with an accurate prediction of damage contributes to the research to damage accumulation in composite materials and validation of structural health monitoring techniques.

A relevant case study object for the implementation of the digital twin in the damage accumulation process of composites is transverse matrix cracking in a cross-ply specimen. Recent research of Li et al. investigates damage accumulation in cross-plies, both under quasi-static and fatigue loading, and provides data to use as a case study. The experimental data provides the opportunity to validate the digital twin and advance the ongoing research. Even though the cross-ply specimen is a suitable case study object, augmentation of the existing experimental data set is still essential to make the data set compatible with a data-driven technique.

A numerical FEM model was opted to provide the augmentation, where two main challenges arise: 1) the complexity of modelling the interaction between matrix cracks and delaminations, and 2) overcoming the deterministic nature of FEM. XFEM-CE was deemed as a suitable method to model the interaction between the two damage mechanisms. Furthermore, to overcome the deterministic nature of FEM, material variability was implemented within the model. Next to the advantage of being able to simulate various damage states with the same model, including the material variability accounts for the inherent variability of composite materials, which should improve the reliability of the simu-

lations.

With regards to the real-time application of the model that processes complex information, machine learning algorithms were assessed as interesting tools. Based on the literature review, a recurrent neural network (RNN) with Long Short-Term Memory (LSTM) cells was opted to serve as a digital twin that processes information and predicts the next crack in real-time. Recurrent neural networks are popular tools in predicting sequences across various disciplines. Even more so, it allows combining classification and regression tasks into a single model. Predicting the next crack location based on a crack history in the form of classification results in a probability per location, which essentially returns a list of the most likely crack locations rather than a single location. The next crack load was predicted using regression, in the unit of strain. After all, that is the measured entity by most sensors.

To synthesize the research project, a digital twin can serve as a tool to solve the problem of modeling complex damage accumulation processes with the final purpose of structural health monitoring or validation of other structural health monitoring techniques. Key building blocks of the Digital Twin concept are developed to address the need to capture complex underlying patterns of the damage accumulation process in composite materials. By focusing on the processing of heterogeneous data in real-time to predict the next crack location and load within a cross-ply, a fundamental model is built that covers the research objective and may form the basis for the next steps in further research, which may be focusing on including different life stages, remaining useful life, and/or other topics that relate to damage accumulation within a composite material.

Material variability was implemented per element in the FEM model, which was configured in Abaqus. All engineering constants were assigned a distribution from which per element a material was drawn. The distribution for the transverse matrix strength Y_T was experimentally obtained. The other distributions were assigned as normal or Gaussian, with mean values from the datasheet and coefficients of variance of 2%. It showed the model capable of generating various crack patterns, although some simulations ran into convergence issues caused by numerical problems.

Damage was implemented using simple damage initiation and propagation laws. Most damage parameters were estimated based on trial and error within a range of values that were found in literature. To model the transverse matrix cracks, elements were enriched in columns with a spacing of 0.5 mm. A maximum principal stress damage initiation was used with linear softening. In general, cracks showed to appear and propagate instantaneously, similar to what was observed in experiments. To model delamination, cohesive surfaces were adopted. A quadratic traction damage initiation criterion was implemented combined with linear softening. The crack patterns that were observed in FEM showed to be similar to the experiments.

After augmenting the experimental data set with FEM data, the data set consists of 277 FEM patterns and 8 experimental patterns. Further augmentation was performed by mirroring the locations and shifting the patterns 1 mm to the left and to the right. The last augmentation procedures increase the size of the data set by a factor 6, although it should be noted that the quality of the data is less than that of new patterns from experiments and FEM.

An architecture with 2 LSTM layers was chosen based on exploring various architectures. The locations turned out to be too stochastic to predict. On the contrary, predicting the strain was more successful. After approximately 5 cracks, it shows that the error decreases and stabilizes within relatively small bounds: a 95% confidence bound of around ± 379 $\mu\varepsilon$ or ± 20 MPa (multiplying the strain by a typical Young's modulus of 52.8 GPa). Increasing the prediction horizon up to 5 cracks ahead increases the confidence bounds by 229%. It is shown that experimental specimens with higher loading rates are more accurately predicted than the specimens of a lower loading rate.

The digital twin as proposed seems to provide a stepping stone for further research: it is proven that without many costly experiments, a digital twin can be established that can provide the user relevant information on its damage state via integration of heterogeneous data sources in real-time. Predictions on damage accumulation can be used in a later stage to take the next step in developing a digital twin of a composite specimen. For example, the number of cracks may form, together with other features such as a strain field, the basis for accurate modeling and prediction of stiffness degradation as

damage progresses. Proposed key recommendations that are expected to facilitate further developments are:

- To include correlation between material properties within the distributions, which can be implemented by linking a micro-mechanical model to the FEM model.
- To adopt a UMAT subroutine in the model to unlock the possibilities of using more sophisticated failure criteria and/or including a fatigue model.
- To increase the size of the data set to improve the performance of the model as proposed.
- To change the problem definition in the neural network for predicting the location to regression or change the type of model to a Bayesian neural network.

Preface

Dear reader,

After a year of participating in courses about the designing, modelling, and manufacturing of composite aerospace structures, I wanted to use this knowledge to do something new, something that was not done before. The Digital Twin concept is a fairly new concept that is currently under development in a wide range of engineering and scientific disciplines, but not for much longer than a decade or so. No exhaustive database of literature on Digital Twins within the field of aerospace engineering and composite materials exists yet. A major advantage of that is that there is still lots of room for development and the opportunity to contribute to the fundamentals of a promising field of research; in other words, the opportunity to do something new. On the other hand, a major challenge is to scope the project to an amount of work that is doable by one master student, because there are countless aspects of the concept that require further development and could be worked out as a thesis. In the end, a paper by NASA on 'the Airframe Digital Twin' turned out to be the starting point of approximately a year's work: my Master thesis. This thesis marks the end of a challenging year, as well as my time as a student at the faculty of Aerospace Engineering at Delft University of Technology.

Many people have helped me to get where I am throughout the last year. First and foremost, I would like to express my sincerest gratitude to my supervisor Dimitrios Zarouchas for keeping me motivated and guiding me during this project. Furthermore, thanks go out to Xi for facilitating the experimental data and giving advice. I am very grateful towards Eva for facilitating the use of Abaqus in times where remote working was the only way to work for a long time. Moreover, I would like to thank Mahir, Thomas and Thom for proofreading and their feedback that improved the quality of the report. Last but not least, I would like to thank my friends, family, and girlfriend for being there for me through thick and thin.

Delft University of Technology
November 5, 2021

Robert Coenen

List of Figures

List of Tables

List of Abbreviations

Abbreviation	Meaning
ALLIE	Internal Strain Energy
ALLVD	Viscous Dissipation Energy
ANN	Artificial Neural Network
BK	Benzeggagh-Kenane
BPTT	Backpropagation Through Time
CDM	Continuum Damage Mechanics
CDS	Characteristic Damage State
CE	Cross-entropy
CLT	Classical Laminate Theory
CV	Coefficient of Variance
DCB	Double Cantilever Beam
DOF	Degree of Freedom
DT	Digital Twin concept
FEM	Finite Element Method
FFC	Compressive Fiber Failure
FFT	Tensile Fiber Failure
FRP	Fiber-Reinforced Polymers
GPR	Gaussian Process Regression
LEFM	Linear Elastic Fracture Mechanics
LRF	Load rate factor
LSTM	Long Short-Term Memory
MAXE	Maximum Strain
MAXPE	Maximum Principal Strain
MAXPS	Maximum Principal Stress
MAXS	Maximum Stress
MFC	Compressive Matrix Failure
MFT	Tensile Matrix Failure
ML	Machine Learning
MSE	Mean squared error
MSLE	Mean squared logarithmic error
PNM	Phantom Node Method
QUADE	Quadratic Nominal Strain
QUADS	Quadratic Nominal Stress
ReLU	Rectified Linear Function
RNN	Recurrent Neural Network
RVM	Relevance Vector Machine
SEM	Scanning Electron Microscope
SERR	Strain Energy Release Rate
SGD	Stochastic Gradient Descent
SHM	Structural Health Monitoring
SVM	Support Vector Machine
UD	Uni-Directional
VCCT	Virtual Crack Closing Technique
WWFE	World-Wide Failure Exercise
XFEM	Extended Finite Element Method

List of Symbols

Latin symbols

Symbol	Definition	Unit
b	Node bias	-
c	Long term state	-
d_{vr}	Damage variable	-
E	Young's modulus	GPa
f	Fraction	-
f_1	Fraction of total load for which specimen is undamaged	-
G	Shear modulus	MPa
G_c	Critical strain energy release rate	J/m^2
G_f	Fracture energy	N/mm
G_{mc}	Mixed mode strain energy	J
G_{Ic}	Critical strain energy release rate mode I fracture	J/m^2
G_{IIc}	Critical strain energy releaste rate mode II fracture	J/m^2
h	Hidden state	-
K_p	Penalty stiffness	N/mm^3
l_{cz}	Cohesive zone length	mm
l_0	Characteristic element length	mm
L	Loss	-
L_{CE}	Cross-entropy loss	-
L_{MSE}	Mean squared error loss	-
$\dot{u}$	Nodal velocity	mm/s
X_C	Compressive longitudinal strength	MPa
X_T	Tensile longitudinal strength	MPa
$\hat{y}$	True value	-
y	Prediction	-
Y_C	Compressive transverse strength	MPa
Y_T	Tensile transverse strength	MPa
w	Node weight	-

Greek symbols

Symbol	Definition	Unit
β_1	Momentum decay rate	-
β_2	Variance decay rate	-
γ	Shear strain	-
δ^0	Smallest length parameter	mm
$\Delta\delta_{crit}$	Smallest displacement increment that causes damage initiation	mm
ε	Normal strain	-
ε_{eq}^f	Equivalent peak strain	-
ε_n	Normal strain	-
ε_s	Shear strain	-
ε_t	Transverse strain	-
η	Learning rate	-
η_{vr}	Viscous regularization term	s
θ	Model parameters	-
κ	Shape parameter of Weibull distribution	-
λ	Scale parameter of Weibull distribution	-
μ	Mean value	-
ν	Poisson's ratio	-
ξ	Regularization parameter	-
σ	Standard deviation	-
σ_{eq}^f	Equivalent peak stress	MPa
σ_n	Normal stress	MPa
σ_s	Shear stress	MPa
σ_t	Transverse stress	MPa
σ_y	Yield stress	MPa
τ_0	Strength	MPa
τ_I	Strength mode I	MPa
τ_{II}	Strength mode II	MPa
τ_{III}	Strength mode III	MPa
ϕ	Activation function	-
ω	Regularization function	-

Contents

1 Introduction

In the last decades, fibre-reinforced polymers (FRP) have become increasingly popular within many engineering fields. Having superior specific properties makes them valuable in industries where weight is of utmost importance, in particular the aerospace industry. However, composites are vulnerable to damage developed during various life-cycle stages, which, may compromise the structural integrity. To assess the effect of damage on a structure, the damage can be identified and characterized via structural health monitoring (SHM). With proper SHM, advantages emerge like, for example, improving the design or rescheduling for maintenance.

The damage mechanisms that appear in composites are complex to model. Therefore, the need for reliable and accurate structural health monitoring (SHM) is still existent in the present after many years. To achieve improvements in the SHM of composite materials, an enhanced understanding of the damage accumulation processes is essential. Moreover, it facilitates the way towards lighter, more optimized, and more sustainable aerospace structures.

Another trend, especially in the last decade, is the growth in the availability of data, having introduced and slowly manifested itself in society in the form of the Internet of Things, Big Data, and Industry 4.0 [1]. It has become increasingly popular amongst researchers to use this increase in available data to increase the reliability of SHM. Augmenting current failure prediction methods for composites with the immense amount of available data might bring the prediction of the structural integrity of composite components to the next level. Two topics are identified to support the implementation of the growing availability of data in SHM: machine learning algorithms and the Digital Twin concept.

The increasing availability of data that comes from sensors that are monitoring a structure allows an improvement in enhanced decision making regarding the diagnosis of a structure's condition and prognosis of damage within a structure. Using ML in SHM has been found particularly useful in three use cases: 1) a large amount of available data, 2) physical characteristics that are too complex to model with physics-based models, 3) required reduction in computational efforts [2]. However, opposing views exist on the use of ML in the field of SHM. Its power has shown to be a welcoming addition to SHM to some, while others are hesitant because it lacks a physical foundation and is thereby considered to be a black box [2]. It is evident that generally speaking, a ML algorithm is much faster to operate than a computationally expensive physics-based model, such as a finite element model. Reduced computational efforts is especially an advantage in applications where fast operations of an algorithm are critical, such as the SHM of a fighter jet that accumulates damage during a fight.

Using the current state of a structure combined with its available operating and failure history data shows to provide improvements in predicting the future state, which is one of many capabilities that the Digital Twin concept introduces. The Digital Twin concept is a particularly interesting concept that potentially provides a paradigm shift and advances further enhancement of big data in SHM. Interest in the concept has grown in the past years along with the growth of available data: it was identified as one of the key concepts behind the rise of Industry 4.0 [3].

In short, the main characteristic of a digital twin is to create a virtual copy of a system (its physical twin) that makes use of an integrated, heterogeneous information database and is updated accordingly in real-time, to increase the user's knowledge of the system and allow to act accordingly. A virtual copy does not necessarily entail a detailed, exhaustive model of the mechanical behavior of every square millimeter of the system. In fact, it can also be to virtually copy and predict the damage

state based on a limited quantity of state variables of the physical twin [4].

The Digital Twin concept has the potential to address the issue of required improvements on SHM of composite materials and ultimately may revolutionize the designing, certifying, maintaining, and operating of systems and their components [5]. However, being a relatively young concept, there is considerable room for development: metal specimens have been the preferred test subject in many Digital Twin studies, which, consequently, address the need for more complex test subjects such as composite materials [5–9].

To summarize, there are three trends ongoing that, once converged, may achieve a synergy effect: development of SHM using composite materials, the Digital Twin concept, and machine learning algorithms. By making a first step towards the synergy effect, this thesis aims to enhance our understanding of damage evolution in composites and improve on health assessment during its service life by constructing a digital twin of a composite material.

A methodology is presented that overcomes the need for expensive testing to make use of the benefits of a data-driven digital twin, which focuses on the real-time prediction of damage accumulation within a composite specimen. A prediction method to predict the damage accumulation is aimed at contributing to the research to damage accumulation in composite materials and validation of structural health monitoring techniques.

Before presenting the research objectives and questions, chapter 2 presents a literature review on the Digital Twin concept, damage mechanics of composite materials, material variability, and machine learning in the context of SHM. Next, the research objectives and questions are defined in chapter 3. The experiments performed by Li et al. that provide validation data for the finite element model and form the basis of the data set to train a data-driven digital twin with is explained in chapter 4. The finite element model that was built is presented in chapter 5 and the data-driven digital twin in chapter 6. Finally, a conclusion and recommendations are provided in chapter 7.

2
Literature Review

The purpose of this literature review is to gain more insight into relevant research fields that are related to the research aim. First of all, the digital twin concept is discussed in section 2.1. The appearance and modelling of damage of composite materials is discussed afterward in section 2.2. Afterward, material variability of composite materials is covered in section 2.3. Lastly, the field of machine learning is discussed in section 2.4.

2.1. Digital Twin Concept

The Digital Twin[1] concept was first thought of as a concept for Product Lifecycle Management by Dr Michael Grieves in 2002. The concept was further developed by Grieves in the years to follow until it was first cited "Digital Twin" [10]. The model described by Grieves and Vickers (the co-author on behalf of NASA) consists of three main elements: a digital entity, a physical entity, and a data and information interface. The idea is that the Digital Twin allows to design, test, manufacture, and use the virtual version of a product, which is considered a fundamental paradigm shift in many industries, amongst which the field of aerospace engineering [11]. In general, the majority of DT research has been done in the field of manufacturing and *industry 4.0* [12, 13], but also in other engineering industries such as the automotive industry [14] or off-shore engineering [15].

One example of the implementation of the DT is NASA's "Airframe Digital Twin". NASA is developing a solution for the way in which heuristics and similitude based certification and maintenance results in aircraft structures being heavier than necessary. They defined the Digital Twin as follows: "*An integrated multi-physics, multi-scale, probabilistic simulation of an as-built vehicle or system that uses the best available physical models, sensor updates, fleet history, etc., to mirror the life of its corresponding flying twin*" [11]. Consequently, on behalf of the U.S. Air Force, Tuegel et al. stated that in 2025 the first aircraft of a new model type is delivered that comes along with a digital twin of the same aircraft [16]. Four categories of challenges are identified by Tuegel on the concept in 2012: modelling of as-manufactured aircraft, modelling of complex flight loads such as dynamic aeroelastic effects, selection and integration of submodels (e.g. additional material state evolution submodels are required), and managing uncertainty, i.e. reducing differences between the physical aircraft and its digital twin [11].

A study performed by Majumdar, Haider, and Reifsnider in 2013 [4] presents a Digital Twin that is suitable for future integration with other types of digital twins to model a multi-physical environment. Specifically, the authors examined how an electrical field causes microstructural changes to simulate the damage tolerance of composite materials due to a lightning strike; experiments showed how electrical current (conducted through the fibers) affects the dielectric matrix material. 3D X-ray image processing was used to obtain the actual geometry of the composite material such that it can be used to solve multi-physics equations. For instance, by using finite element software: a segmented volume or area is captured by the 3D imaging and consequently used to generate a mesh and solve accordingly.

Majumdar and Reifsnider performed another study in 2013 [17] in which they address the need for a model that can be used for composite materials. Changes in the material properties as a result of

[1] "Digital Twin" or "DT" refers to the conceptual approach, "digital twin" refers to a specific digital model or replica.

damage are essential to understand in order for a prognosis of future performance. It is shown that measuring changes in electrical permittivity of a glass-epoxy fabric during tensile testing allows for data to be generated in which early warnings are shown of impending failure once approaching the end of life. Crack initiation, accumulation and interaction are identifiable based on the rate of change of the global permittivity. The work is presented as foundation for a flying digital twin.

In 2017, Li et al. [18] proposed a concept of a probabilistic model to realise the Digital Twin vision that was initially proposed by NASA [11]. More follow up research on the concept by NASA and the U.S. Air Force has been done in the past years [5–8]. Outside of aerospace, the US Department of Energy sponsored research that presents the Digital Twin as a predictive probabilistic model that was trained using a combination of experimental and computational data by using machine learning [19]. This digital twin was used to characterise and estimate statistical dependence of parameters within a composite material. Next to the U.S., other governments of countries such as China and Canada are also researching the possibilities of the Digital Twin in (aerospace) applications [20, 21], as well as conglomerates such as Airbus, that is working on a Digital Twin to fulfil the never-ending need of aircraft operators to increase efficiency [21], or General Electric [22].

Back when Grieves first came up with the idea, it did not attract much attention among scientists because of three main reasons: difficulties in real-time processing of big data, limited developments in options to digitally describe products, and a limited amount of technical means to collect information in a production process [23]. Linking these difficulties that Wang described to damage mechanics in composite materials: deep learning algorithms such as neural networks have proven to be capable to handle big data in real-time applications [24]; damage mechanics can be described both analytically by a plural of models and numerically via finite element software; and damage information collection or damage diagnostics can be done via several methods that can also be used outside the laboratory in real aerospace applications.

Therefore, the digital twin is a viable solution nowadays that could advance ongoing research on damage accumulation in composite materials with the purpose of further development of SHM. Correspondingly, the damage mechanisms in composite materials are more elaborately discussed.

2.2. Damage in Composite Materials

Composite materials are notorious for the complexity in analysis that arises once different types of damage mechanisms occur simultaneously and/or interact with each other. The sequence of damage accumulation can be attributed to various causes, such as the environmental conditions, the load case, layup of the laminate, manufacturing defects, and material properties. modelling this process can be challenging. Therefore, in this section it is established what types of damage can be found in composite materials and how they can be modelled. Firstly, a qualitative discussion of damage phenomena is provided in subsection 2.2.1. Next, the two main categories in damage mechanics models are discussed in subsection 2.2.2. An overview is given of the development and availability of failure criteria for the initiation and progression of damage in subsection 2.2.3.

2.2.1. Damage Phenomena

A typical evolution of damage under static or fatigue loading within a composite specimen's life is depicted in Figure 2.1. Damage occurs from a microscopic to macroscopic scale – causing multi-scale modelling difficulties, sometimes referred to as *the tyranny of scales* [25] – and can be divided into two groups: interlaminar and intralaminar damage. Interlaminar damage would be delamination, i.e. the separation of two laminas due to interlaminar matrix cracking. Intralaminar damage mechanisms are for example matrix cracks or fiber breakage. Intralaminar damage phenomena are discussed first, followed by interlaminar damage phenomena. In the last part of this section, the ability of the material to withstand loading once damage is present, the damage tolerance, is described.

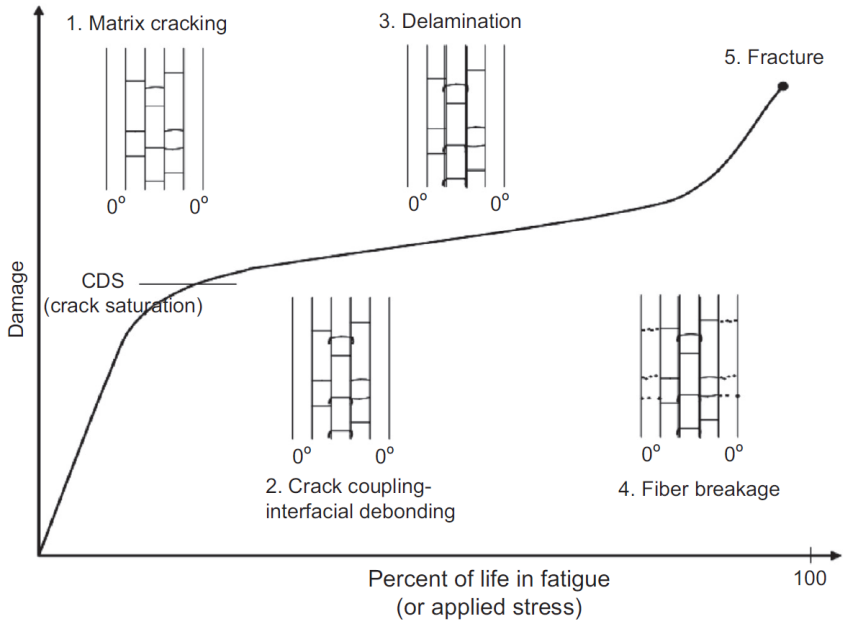

Figure 2.1: Typical damage evolution within a composite specimen's life [26].

Intralaminar Damage

Cracks in the matrix initiate in many forms and due to various scenarios. In case of damage in composite materials, cracks in the matrix are rarely the sole damage that is present, but still affect the residual strength of the material [27]. Intralaminar crack growth is often irregular and constrained by the surrounding fibers. These surrounding fibers generally act as crack stoppers which is the reason why often cracks between plies (delamination) do not easily branch out but mostly only grow between different plies. Crack growth through the thickness can analytically be modelled by several models, of which the Mar-Lin prediction method is shown to be more accurate than the Whitney-Nuismer, Poe-Suva, and Linear Elastic Fracture Mechanics models in estimating crack size [27]. Numerical models such as the finite element method (FEM) provide alternative methods.

Transverse matrix cracks occur in off-axis plies under both quasi-static and fatigue loading. Its degree is characterised by the crack density, which is defined as the number of cracks per unit length [28]. Once cracking begins, the crack density increases until a saturated state is reached, a so called characteristic damage state (CDS). The CDS (or "crack saturation" point as shown in Figure 2.1) has been proposed to be independent of the applied loads and instead is determined by the layup, next to geometry and material properties [29], while experimental studies report specimens showing differences in the crack density value corresponding with CDS under the same loading conditions [30]. The overwhelming majority of studies on (transverse) matrix cracks assumes that the crack spacing is equal between the cracks and that as such, analysis can be narrowed down to a representative laminate segment [28]. However, parameters such as the applied loading rate affect the amount of scatter in the uniformity of the matrix crack distribution [31]. Figure 2.2 shows various crack patterns that were observed in cross-ply laminates, including other damage mechanisms induced by matrix cracking, such as matrix micro damage (d), delamination (e), or crack kinking through the interface (f).

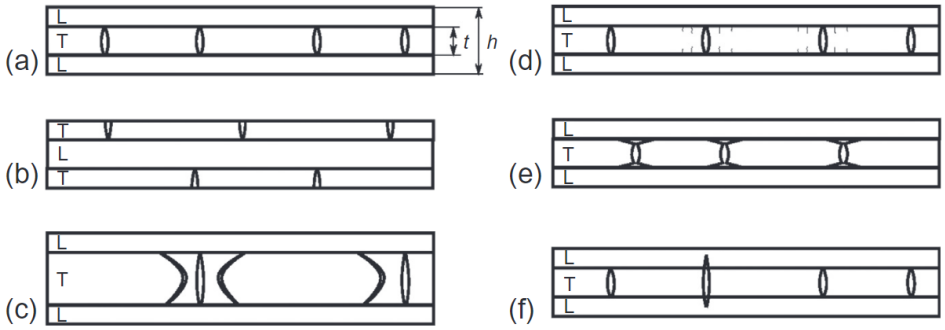

Figure 2.2: (a-c) Various crack patterns in cross-ply laminates (d-f) and damage mechanisms induced by matrix cracking. L - longitudinal, T - transversal [32].

Other type of intralaminar damage modes are micromechanical failure modes that relate to the interface between fiber and matrix, such as fiber-matrix debonding. The mechanisms behind debonding between the fiber and matrix depends on the loading type. In case of transverse loading, it can be observed in broken specimens on a microscale that the damage pattern can be traced to the presence of debonds; meaning the development of this type of debonding can turn out to be very significant in the global damage accumulation of in the composite [33]. This can be considered as mode I failure, which is further explained in subsection 2.2.2. In case of axial loading, mode II failure is occurring. To determine the interfacial fracture toughness, the single-fiber fracture toughness test can be used [34]. Fiber pull-out is another example of failure of the fiber-matrix interface. Examples of micromechanical tests that are used to determine the interface strength are illustrated in Figure 2.3.

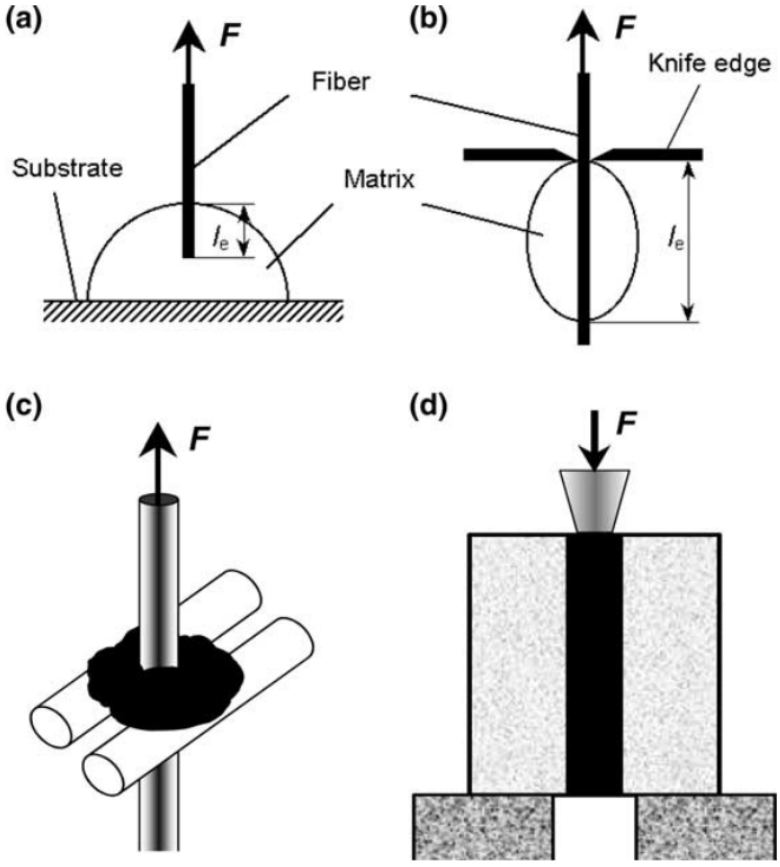

Figure 2.3: Micromechanical failure mode tests where a load is directly applied to the fiber: (a) fiber pull-out (b) microbond testing (c) three-fiber test (d) push-out [35].

Different loading types generally induce different types of failure modes. Under tensile loading, fiber breakage and matrix cracking most commonly occur, while under compressive loading the primary failure mechanism is fiber kinking due to matrix cracking or yielding and rotation of fibers [36]. The first campaign of the World-Wide Failure Exercise (WWFE) stressed the need for dedicated experimental campaigns that investigate the mechanical response of composite laminates in the presence of multiaxial stress states [37]. A study by Kalteremidou et al. [38] was aimed at providing such a dedicated experiment and considers the effect of multiaxial stress state, as well as the effect of stacking sequence and number of off-axis layers on the mechanical response and damage sequence in carbon/epoxy laminates. In case of shear loading being dominant in one of the off-axis layers, delamination occurred even prior to matrix cracking.

Intralaminar Failure Modes

The failure modes within a ply of a FRP are generally characterised as Tensile Matrix Failure (MFT), Compressive Matrix Failure (MFC), Tensile Fiber Failure (FFT), and Compressive Fiber Failure (FFC).

FFT – Fibers failing due to tensile loading results in failure of the main load-carrying components within a ply, without the load being able to be redistributed. Therefore,it is generally considered to be a catastrophic failure mode. FFT is an explosive failure mode because of the energy that is released [39].

FFC – Compressive loading on the fibers is influenced by the resin's shear load-carrying capabilities and imperfections in the laminate. Ultimately, fiber kinking can be observed as well as microbuckling of the fiber or shear failure of the matrix [39].

MFT – Transverse tension that causes matrix cracks is denoted as MFT. The cracks usually grow through the thickness of a ply until stopped by the ply boundary. In general, the direct effects of this failure mode result in minimal stiffness losses. However, once these cracks will grow they may induce more sever detrimental effects and activate other damage modes such as delamination at CDS [39, 40].

MFC – Transverse compression that causes cracks in the matrix may lead to cracks that do grow through the thickness. Even though it is commonly referred to compressive failure of the matrix, it is actually shear failure of the matrix, as shown by the angle between the fracture and applied load [39].

Interlaminar Damage

Delamination is an interlaminar failure mode of a composite material of which the onset and propagation has been investigated by several reviews [28, 41, 42]. It can be difficult to detect delaminations because they are frequently embedded in the material due to various causes such as manufacturing defects [41]. Development in understanding the process of delamination started with the theory of fracture mechanics, which is more elaborately discussed in subsection 2.2.2. Even though modelling this failure mode is of great importance, it cannot be considered to be isolated from less critical damage types, such as transverse matrix cracks. For instance, layup and the applied loading profile play a role in the damage interaction [43]. The onset of delaminations of, for example, cross-ply laminates is often preceded and caused by cracking of the matrix [44]. Delaminations caused by the tips of microcracks within a cross-ply laminate have been analysed by Nairn and Hu [45]. The authors show here that at a certain critical crack density, the required energy release rate for the initiation of delamination becomes lower than that for microcracking, i.e. delamination will be initiated rather than microcracking.

Damage Tolerance

Damage of the material results in degraded residual properties, such as strength or stiffness. The damage tolerance of a structure indicates its capability to remain its original material or structural properties despite being damaged. The degradation of a material can be modelled both instantaneous and sudden. modelling how the degradation of material affects its residual properties is one of

the main challenges in progressive failure analysis [46]. Being able to assess the effect of damage on the material response is an essential step before prognosis. Vice-versa, the reduction of mechanical properties can be measured and used to define damage [47].

Dependent on the type of damage, different degradation effects can be attributed to the composite material. For example, fiber failure in case of a tensile on-axis load using a cross-ply will effect a more significant degradation to the specimen than transverse cracking in the 90-plies. In other words, a single, fixed damage mechanism is generally not the cause of failure of a composite laminate, which has been shown in extensive experimental work [48]. Therefore, failure mode-dependent failure criteria are more suitable to use in progressive failure analysis [49]. Such failure criteria are discussed in subsection 2.2.3.

2.2.2. Damage Mechanics

The nucleation and growth of defects on a micro-scale to their coalescence into cracks on a macroscale was first used to predict the creep rupture [50]. Afterwards, the the concept of Damage Factor was developed by Robotnov [51]. With damage mechanics providing for a measure of material degradation on a microscale, damage variables were introduced to reflect the degradation of the material on average on a macroscale, which is why Continuum Damage Mechanics (CDM) was developed. CDM and fracture mechanics are discussed here.

Continuum Damage Mechanics

Continuum Damage Mechanics (CDM) has been widely used over the past six decades to predict damage evolution of composite laminates and the corresponding stiffness degradation [52]. In CDM models, the accumulation of damage is modelled by reducing certain material properties. As such, the loss of load-carrying capability is simulated. Once the values of specified damage values reaches a certain value, it is assumed that the material has failed and can no longer withstand any loads. Failure criteria exist that describe the onset and progression of damage using such a strength-based principle, which are provided in subsection 2.2.3. Correspondingly, the stress is used as measure, which can be obtained by e.g. measuring strain via experiments. Because of this, these methods are sometimes also referred to as stress-strain methods.

Several aspects make composites difficult to model damage using this principle. Due to the morphology of the material, there will be a preferred direction for crack growth. This is not merely resulting from the geometry, loads or boundary condition, but from e.g. the fact that the interface between the fiber and matrix is weaker than surrounding material [53]. Another example is the difference in material properties such as the coefficient of thermal expansion of the fiber and matrix that may cause internal stresses. This inhomogeneity of the composite material simply does not agree with homogeneous modelling of reduced material properties.

Nonetheless, in the early 1990's, CDM degradation models were first applied to composite materials [54]. From a mesoscale perspective, a composite lamina is more suited to treat as a homogeneous material [53]. Plane stress conditions can be assumed for a lamina, which simplifies the stress vector from 3D to 2D. Another advantage of using lamina properties rather than material properties, is that using ply properties prevents redoing experiments every time the layup of the laminate changes.

CDM approaches are preferred in case of a more diffuse and spread damage pattern that is typical for intralaminar damage forms [36, 48]. Combining damage and fracture mechanics approaches to solve progressive damage analysis problems in composite materials have been proposed in the last years [55]. In modern approaches, progressive failure analysis is performed using FEM-based numerical methods, which is more elaborately discussed in subsection 2.2.4.

Fracture Mechanics

Fracture Mechanics is an energy based method and was developed during the first world war by Griffith to explain failure of brittle materials [56]. The change in energy during crack propagation under

a constant load was related to the creation of crack surface. This rate is defined as the strain energy release rate and the required energy to crack a surface is defined as the fracture toughness G_c. Plasticity is not taken into account in Griffith's theory and a stress singularity at the crack tip exists here as well. Irwin introduced a plastic zone, which also removed the stress singularity and makes the method valid for both brittle and ductile materials. As shown in Figure 2.4, it makes a distinction between three fracture modes: Mode I – opening or peeling; Mode II – In-plane shear; and Mode III – Out-of-plane shear. Mode I fracture has generally the lowest fracture resistance, making it the most likely to occur, although a mixed mode fracture also often occurs.

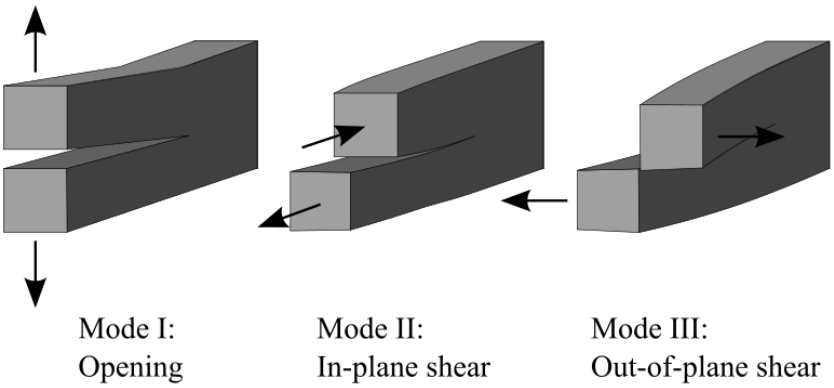

Figure 2.4: Fracture modes I, II, and III.

In composite materials, a plastic zone may not be (completely) developed, e.g. due to fiber bridging between two crack surfaces. The fiber bridging may be more dense towards the crack tip, resulting in a higher resistance to separation there. This zone is called a cohesive zone and based on two independent concepts by Dugdale [57] and Barenblatt [58]. The main advantage of this model is that cracks can be modelled in uncracked structures, whereas classical linear elastic fracture mechanics (LEFM) requires the presence of an initial crack. The cohesive zone model forms the basis of the cohesive zone element. Delaminations are generally simulated using cohesive cracks, because of the discrete nature of the cohesive model [36, 48]. modelling cracks in finite element software and is further explained in subsection 2.2.4. An example of the use of fracture mechanics in FEM is the Fracture Analysis Code 3D (FRANC3D), which enables 3D arbitrary, non-planar crack growth and was coupled to a prognostic model that was built as a digital twin [8].

Another approach in modelling delamination in composite materials is the use of an approach called finite fracture mechanics [59]. A crack will initiate once a certain amount of energy is released, at the point of the so called critical energy release rate, which can be explained as follows. Linear elastic anisotropic materials can be analysed using the strain energy release rate. For self-similar crack growth like delamination, the SERR can be determined via local forces and moments [60]. The basic assumption here is that the energy per unit area required to create a new cracked surface area is equal to R, the crack resistance, which depends on e.g. the type of fracture mode.

At fracture, $G = R$ holds. G_I and G_{II}, the energy release rates for mode I and mode II fracture, together make up the total SERR. Both G_I and G_{II} can be expressed as a function of the geometry and loading, which means that already from relatively simple experiments the energy release rate can be obtained without directly measuring energy.

The critical energy release rate is experimentally obtained, based on the type of material and it is reasonable to consider it a material property independent of stacking sequence [61]. It should be noted that in case of large displacements the load-displacement curve may become non-linear. An advantage of using a double cantilever beam (DCB) specimen is that the fracture toughness is relatively simple to determine based on the described principles; novel methods are described in which only the load-displacement curve is required to determine the mode I and mixed modes I/II interlaminar fracture toughnesses [62]. The fracture toughness of a material is an indication of a material's sensitivity for cracks under static loading and is used for residual strength calculations [63].

2.2.3. Modelling Damage

Modelling the onset and progression of damage can be done by using failure criteria. These criteria can be applied both analytically and numerically. Lots of different criteria exist, while there is no single criterion that is universally accepted. An overview of initiation criteria which are still implemented in numerical models nowadays is provided here first. For example, Hashin damage initiation criteria are still the default setting in the widely-used commercial FEM software *Abaqus*. Next, damage propagation and corresponding material degradation models are discussed.

Initiation Criteria

The simplest initiation criteria are the maximum stress and maximum strain criteria, which make no distinction between different types of failure mode, i.e. mode-independent, and compare the stress or strain in a certain direction with the material strength property in the corresponding direction. Once the measured stress or strain exceeds the material strength properties, the ply will fail. The Tsai-Wu failure criterion is an example of a criterion that is mode-independent [64] and was the first to be formalized in 1971 [65]. The Tsai-Wu criterion makes use of seven polynomial invariants based on the stress state of the material.

A few years later, Hashin and Rotem were the first to make a distinction between different failure modes within a material: fiber and matrix failure under a cyclic tensile normal stress with plane stress conditions [66]. Hashin continued working on the topic and established failure criteria that were based on the hypothesis that fracture is provoked by stresses acting on a fracture surface (or action plane) [67]. The World-Wide Failure Exercise (WWFE) resulted in 19 failure theories [68], of which Puck and Schurmann's criteria were found to be one of the most noticeable ones in terms of accuracy [69]. Puck and Schurmann developed an approach that continues the work of Hashin and has a reasonably physical basis, namely the Mohr-Coulomb fracture hypothesis [70]. Additional parameters next to mechanical material properties are required, some which are experimentally determined [71]. Cuntze's failure criteria resemble that of Puck, including the failure envelopes, but included a probabilistic method to model interaction between damage mechanisms [72]. The LaRC 03 extended Puck and Hashin failure criteria and makes use of in-situ strength values [73] which can be quite different from experimentally obtained strength values [69]. The in-stu strength relates to the fact that the strength of a ply that is embedded in a laminate is actually larger than the experimentally obtained strengths of a single UD ply [74].

LaRC04 and LaRC05 are again expansions of the LaRC03 failure criteria. LaRC04 uses three-dimensional criteria and is strongly physically based: it is devoid of empirical parameters [75], while LaRC05 provides further computational advancements and was submitted for WWFE-II [76], which is targeted at developments regarding three-dimensional and through-the-thickness failure criteria. The WWFE-II provided two rankings of 12 failure theories, however, the assessment of failure theories is critised by Christensen because of several reasons, some of which are: the unconservativeness; partial incorrectness on hydrostatic pressure load cases; contradictory theories within the winning four theories; personal views influencing the outcome of the organizers; and a high number of input parameters (50-75 for the winning theories) which implies the outcome might as well be due to the power of parameters rather than a physical basis [77]. It should be noted that Christensen's own failure theory participated and ended 11th out of 12, about which he stated "it was viewed by the organizers as being the very poorest of quality". Nonetheless, Christensen acknowledges the WWFE-II to be a useful step toward a robust and reliable failure criterion for FRP.

Even recent (2020) research [78] shows that Hashin and Puck criteria are able to predict first ply failure (FPF) within a error margin of 5%. Such early theories like Hashin, but also Tsai-Wu are still widely used because they are easy and simple to understand and implement, despite their shortcomings [79].

Many criteria succeed in predicting the remaining strength of a composite component, but struggle with predicting the initial failure stress, the corresponding damage mode, and the significance of this initial damage mode for the subsequent performance after FPF [37]. However, it is argued here that

accurate prediction of damage initiation is of paramount importance to reliably implement damage-based failure criteria.

All in all, plenty of failure criteria exist but none has shown to be sufficiently reliable and accurate to become universally accepted [49]. The mode-dependent criteria lack the homogeneity of the mode-independent criteria because different failure modes are considered amongst the various criteria [49]. For an analytical model to consider all variables upon which complex damage phenomena rely, such as delamination due to buckling, is rather challenging, which is why numerical finite element methods have been developed in the past years.

Damage Propagation

Models for the damage propagation of composites are widely available, from relatively simple stiffness degradation to more complex CDM or fracture mechanics based models. CDM-based models have shown to be the most accurate to predict failure strengths of composite materials [80, 81]. On the other hand, delamination is more accurately modelled using LEFM based approaches [36, 48]. Modelling sudden material degradation upon damage onset can be done by using one of the initiation criteria as stated above, which are easy to use [46]. Another option is gradual degradation which allows for more accuracy in modelling and the implementation of the underlying physical principle [46]. Models for sudden material degradation have been used and are briefly discussed per failure mode. Afterwards, gradual degradation models are discussed.

Fiber Failure In case of fiber failure, the main load-carrying components of a ply have failed. Additionally, in case of FFT there is a significant energy release that potentially damages adjacent material [39]. Sudden fiber failure has been implemented in the form of total-ply discount methods, such as by Lee [82]. Here, in case failure is observed using a failure criterion, the entire ply assumed to have lost all of its material properties. Using common sense, it can be established that it is not realistic that a ply has no material properties left at all in case of failure. To address this problem, several models have been proposed to select more carefully what properties are degraded and to what values. One example of such a model was presented by McCarthy et al. [83]. Next to modelling not all material properties to zero, a distinction was made between FFC and FFT. This research was followed up by Camanho and Matthews [84]. More examples of sudden fiber failure modes can be found in a degradation model review by Garnich and Akula [46].

Matrix Failure Generally, matrix failure is associated with the loss of transverse stiffness properties. Camanho and Matthews distinguish compressive and tensile matrix failure, where it is assumed that a compressive crack still allows load trasnder and therefore compressive failure is deemed less critical [84]. Because matrix cracking is associated with shear failure of the matrix, the shear moduli are degraded by this model as well.

Gradual Degradation Models Numerous examples exist of gradual degradation models, but unfortunately, those generally still lack physical understanding and reliable experimental data. One example of a model that has a more physical basis is those that are based on bi-linear softening, as proposed by e.g. Pinho [39, 76] and Zhang [85]. The bi-linear softening law assumes a linear softening relation after failure. The strain softening of the material is dependent on the failure mode. Separate fracture toughnesses for each of the failure modes have to be obtained; how to do so is described in more detail by Pinho [39]. Another approach in gradual material degradation was proposed by Puck and Schurmann [70]. This approach can be used in case of matrix failure and proposes the use of a degradation factor which allows the reduction of transverse properties, dependent on the identified failure mode of the matrix. With most of the available gradual degradation models following a heuristic approach and having limited physical basis, recent numerical FEM methods provide a solution in which stress-strain based models and fracture mechanics based models are integrated, which is discussed in subsection 2.2.4.

2.2.4. Finite Element Method

The Finite Element Method (FEM) is a tool that has developed over the years into a indispensable technology for the simulation and modelling in a wide variety of advanced engineering fields, such as thermal analysis and of course structural analysis. FEM can be used to obtain the distribution of field variables within a specified domain of a problem that is difficult to solve using analytical solutions. Solving a structural analysis problem via FEM usually starts by modelling the structure's geometry and mesh it, i.e. discretize the structure into smaller pieces called elements. Next, material properties have to be specified, as well as the boundary, initial, and loading conditions. The complexity in all the associated calculations make FEM difficult to implement in in-situ damage prediction [86]. However, it has been shown to be effective in the training of a prognostic model [87]. Another example where FEM has been used, to train an artificial neural network (ANN), is provided by Mucha [88]. It was shown that the computational effort was reduced by one order of magnitude while the accuracy was maintained at the same high level. However, it should be noted that obtaining those results required many attempts and was not trivial.

In case of damage, failure criteria and damage evolution laws can be selected and determine the way in which the structure's material properties are degraded. Depending on the type of software that is used, failure criteria and damage progression as described in subsection 2.2.3 are available. CDM and fracture mechanics based approaches can be used to model damage using FEM software. Figure 2.5 illustrates a taxonomy of the computational approaches that exist to simulate fracture and the progression of damage in composites. Relevant approaches are discussed in this section.

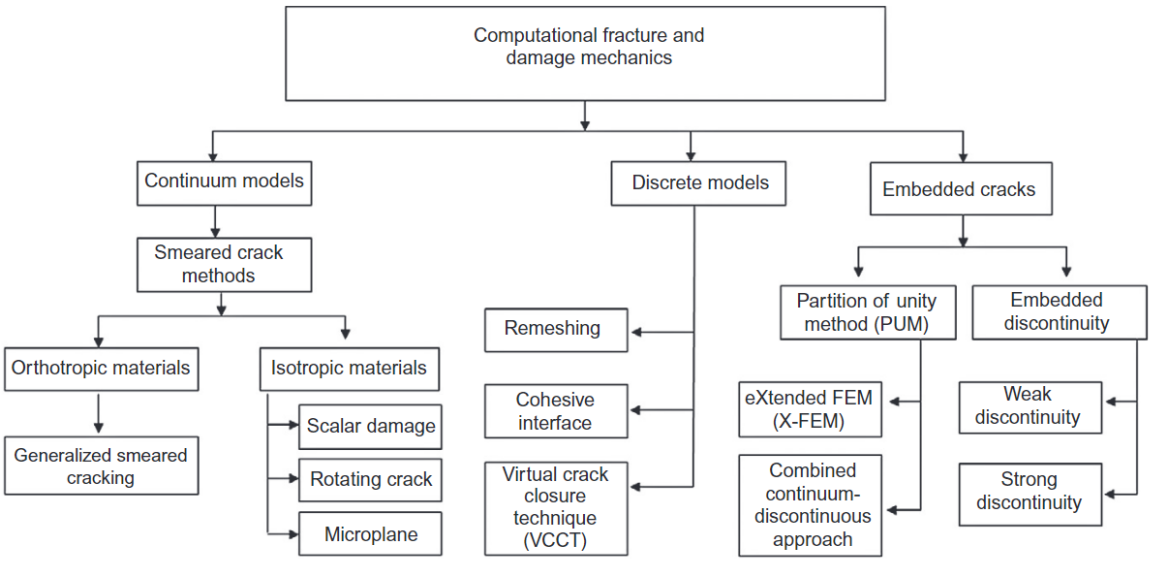

Figure 2.5: A taxonomy of computational damage and fracture mechanics approaches [36].

One of the simplest ways to model progressive damage and failure is through stiffness degradation of the elements using the stiffness degradation method. By using failure criteria such as specified in subsection 2.2.3, damage onset and progression can be determined and subsequently, the stiffness can be degraded. Figure 2.6 schematically illustrates the process.

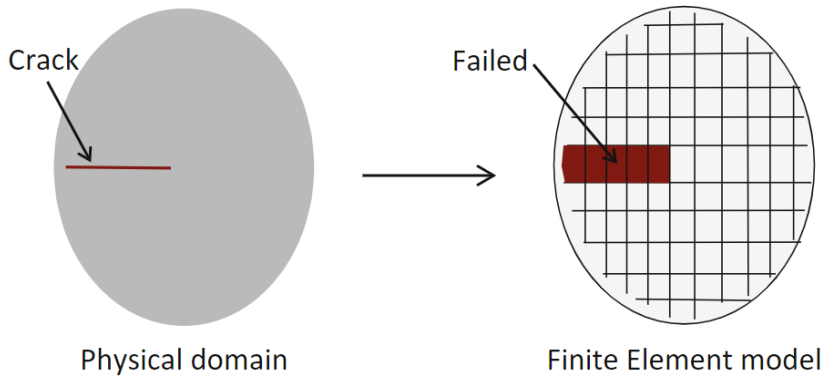

Figure 2.6: Modelling a sharp crack in a FEM [48].

Cohesive Elements

The cohesive zone as described in subsection 2.2.2 formed the basis of the cohesive element in FEM. The cohesive zone method assumes that once damage is initiated and the two faces are separated, the load carrying capability is not instantly lost [47]. Instead, the stiffness loss progresses more gradually, as illustrated in Figure 2.7, using a constitutive traction-separation law. Cohesive elements are interface-type elements with zero thickness that can be modelled inside the mesh along a potential crack path [48]. Such elements are assigned zero thickness or, in some cases, a small, finite thickness. It can be particularly useful to model delaminations. However, once it is unclear where a crack will initiate, it can be difficult to decide the number and corresponding locations of cohesive elements that should be modelled. To overcome this problem, alternatives exist and are discussed hereafter.

Three parameters often determine the accuracy of using cohesive elements to model delamination in composite materials: the fracture toughness, penalty stiffness, and interfacial strength [89]. The fracture toughness, or fracture toughnesses of multiple modes in case of mixed mode loading when relevant, is used as input for the failure criterion and can be experimentally obtained. In case very high stress concentrations, e.g. at crack tips, are present, the interfacial strength was found to have no effect on the results. However, in cases where those high concentrations were lacking, the results turned out to be sensitive to the modelled strength. The penalty stiffness describes the behaviour of the interface before any damage is initiated. Low values decreases the structural stiffness and delamination initiation, while high values ensure correct modelling of the structural stiffness at the cost of an increase in required computational efforts.

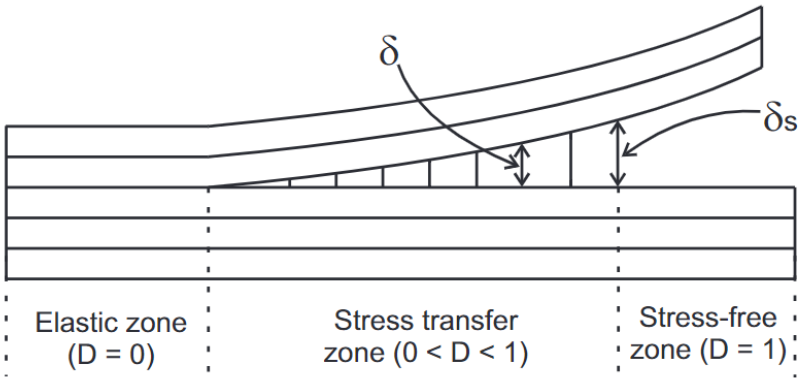

Figure 2.7: Cohesive zone model [47].

Smeared Crack Model

The smeared crack model comes down to an improved stiffness degradation method, using a cohesive law to degrade element stiffnesses. Instead of modelling cohesive elements straight into the mesh, a cohesive crack is smeared in a ply element following a cohesive law, and stiffness is of the element is degraded accordingly [90]. An advantage is that because a cohesive law is used, the energy dissipation is correct. A disadvantage is that the direction of the mesh dictates the elements that will fail next, which affects the prediction of fracture paths [91].

Multiple possible solutions exist to overcome this mesh bias. One of which is remeshing around the crack tip to achieve a better accuracy in the solution. However, this can still be complicated to model, due to the complex crack geometry in composites, as well as the fact that cracks in all plies would have to be remeshed properly. Additionally, remeshing is quite expensive from a computational point of view. Another solution is to evaluate damage based on the weighted average strain within the region surrounding a crack tip, rather than evaluating the sole crack tip element. Other solutions have been developed as well in the form of different methods.

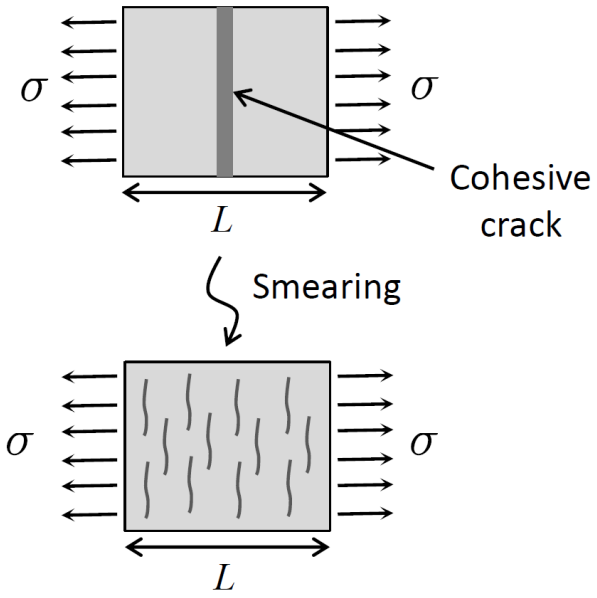

Figure 2.8: Smeared crack model [92].

Virtual Crack Closing Technique

Virtual crack closing technique (VCCT) is used for computing the strain energy release rate (SERR) from FEM simulations in order to allow the use of the mixed-mode fracture criterion [93]. The method is based on fracture mechanics; it is based on the assumption that the energy required to create a crack is equal to the energy required to close it. VCCT was used as basis for a computationally efficient approach to simulate discrete crack growth [94]. Here, the FEM analysis and crack growth module were decoupled to increase the computational efficiency. The role of VCCT was to provide the SERRs, which were then used to convert into stress intensity factors. It is stated that other fracture analysis methods could have been used as well, but VCCT was selected due to its capability of providing accurate SERRs with a coarse mesh.

eXtended Finite Element Method

The eXtended Finite Element Method (XFEM) was proposed by Belytschko and Black [95]. In XFEM, the elements are enriched with additional degrees of freedom (DOFs). Once a crack occurs, these additional DOFs are used to represent a displacement discontinuity. This is illustrated by Figure 2.9. The crack tip itself can be enriched as well, such that analytical information can be used to model the shape functions around the crack tip. However, this information is not always available for composite

materials. Bouhala et al. proposed an approach to allow crack tip enrichment in composite materials [96].

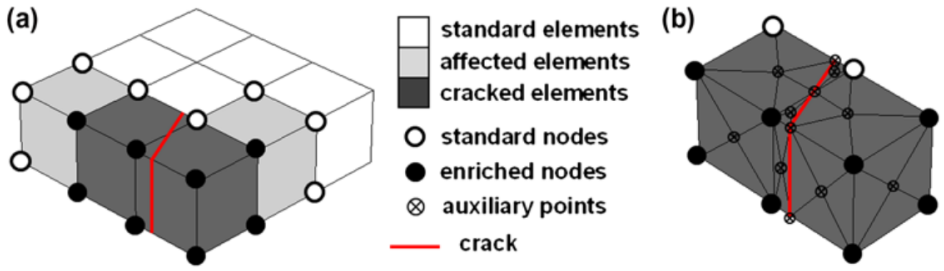

Figure 2.9: Enrichment and partition of elements in XFEM. (a) enriched elements (b) partitioned elements. [97]

Several modifications have been proposed and reviewed already [98]. The main advantage of using XFEM is that crack propagation can be simulated without predefining a crack path or need to update the mesh [99]. Van der Meer et al. have shown that an XFEM based approach was capable of simulating transverse matrix cracks in cross-plies in good agreement with experiments [91]. The performance of XFEM in application to the cracking process in cross-ply laminates was assessed by Petrov, Gorbatikh, and Lomov [99] using *Abaqus*. It was found that XFEM was capable of predicting the crack propagation and corresponding crack density in a cross-ply laminate. A physical trend was used between crack density and observed strain, which was in agreement with presented experimental data. Nonetheless, the biggest identified challenge was proper interpretation of the input parameters that describe microscale phenomena.

One of the modifications that has been made on the XFEM model is Hansbo's method, or the Phantom Node Method (PNM) [100]. Elements are assigned both real and phantom nodes, where the latter are created as a double of the real nodes. Once a crack appears, the element is split into two independent element by the crack. With the presence of the phantom nodes, two separate displacement fields are created. In a study by Van der Meer et al. PNM is used to model matrix cracking, cohesive interface elements are used for delamination, and CDM is used to model fiber failure [101]. The predictions that were made were found to be in good agreement with experimental values.

Another modification of XFEM is to integrate the cohesive elements with XFEM, the XFEM-CE approach [55]. This approach was addressed at the limitations of the smeared crack method. Both XFEM-CE and PNM are aimed at capturing the interactions between matrix cracks and delaminations more accurately. In the XFEM-CE method, XFEM is used to model matrix cracks and cohesive elements are used to model delaminations, as illustrated in Figure 2.10. With XFEM, smearing of the crack is no longer required and the associated mesh bias is eliminated. Cohesive elements are placed between plies that have different fiber orientations. Furthermore, those cohesive elements are enriched to ensure continuity with the adjacent elements.

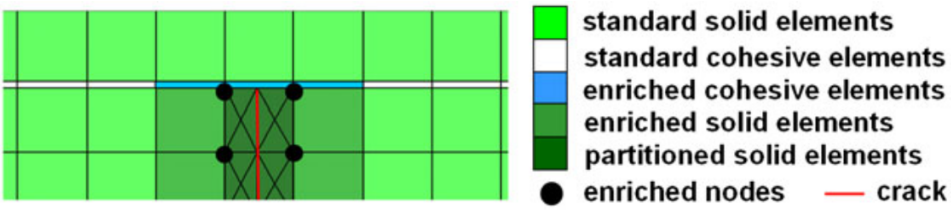

Figure 2.10: Interaction between partitioned solid elements and cohesive elements [97].

A parametric study was performed by Petrov, Gorbatikh, and Lomov that assesses the performance of XFEM in modelling cracks in a cross-ply laminate [99]. It is shown that the strength distribution strongly affects damage initiation. In addition, it affects the damage propagation when using a traction-separation law, because it changes the maximum separation for a given critical energy re-

lease rate, which is illustrated in Figure 2.11. The crack pattern seems to be sensitive to whether delamination is modelled or not in terms of the evolution of cracks and the final crack density. The graph of a single crack evolution plot is compared to the model and agrees well.

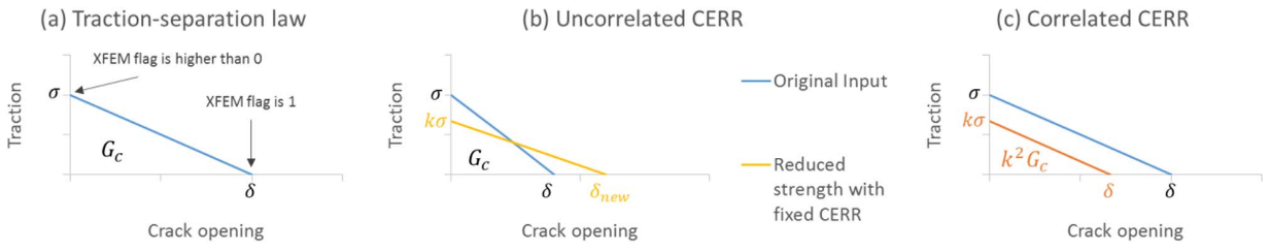

Figure 2.11: Traction-separation behavior after damage initiation: (a) linear softening law with XFEM status, (b) effect of the coupling between CERR and strength, (c) effect of correlating CERR and strength [99].

The next section covers how the material variability that is inherent to composite materials is introduced into modelling damage using FEM.

2.3. Material Variability

Simulating varying material properties to model the mechanical behavior of composites has been done already, one of the main reasons being the need to overcome the deterministic nature of models, such as FEM models. The probability distributions that describe the variability of material properties is ideally obtained via experiments, but unfortunately experiments are costly. Literature presents distributions that were obtained via either experiments or alternative approaches (or both). An overview is presented in this section.

Two decades ago, Okabe and Takeda proposed an approach to estimate a strength distribution for a single fiber embedded in a composite [102]. An adopted shear lag model takes into account elasto-plastic deformation of the matrix in the interface region, based on which a Monte Carlo simulation[2] is used to estimate parameters of a two parameter Weibull distribution. Sun et al. make use of a shear lag model and Monte Carlo simulation as well to model transverse matrix cracks in a cross-ply under fatigue loading [103]. Figure 2.12 shows the probability density function that was obtained. From the shear lag model, it was found that the longer a uncracked segment's length, i.e. between two cracks, the higher the maximum stress amplitude and the shorter its life.

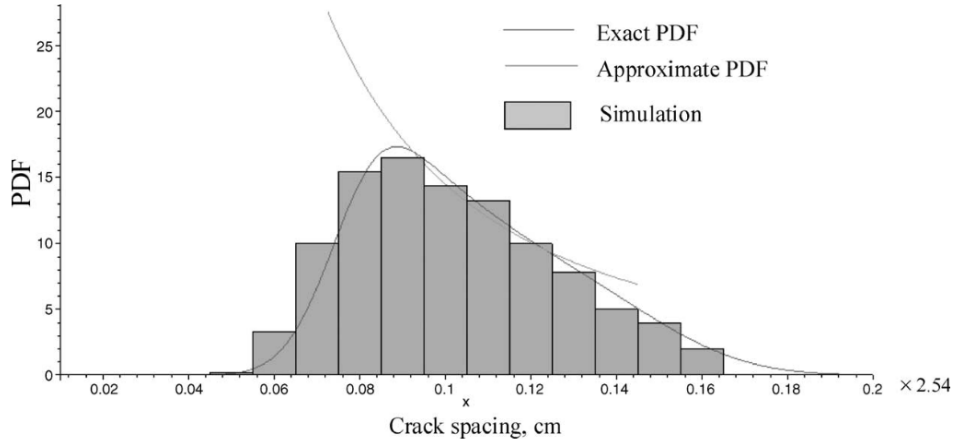

Figure 2.12: PDF of transverse crack spacing in $[0/90_4]$ from simulation and two theoretical models [103].

Lian and Yao use a similar strategy in an approach to modelling fatigue life prediction of compos-

[2] A Monte Carlo simulation essentially simulates a physical process multiple times with different starting conditions each time. As such, the set of simulations represents a probability distribution.

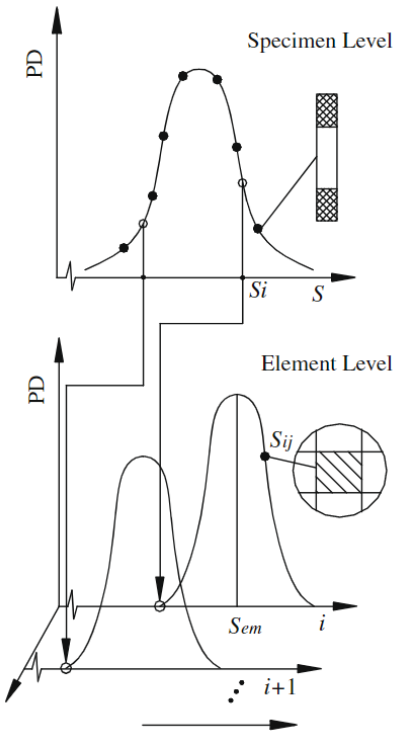

Figure 2.13: Probability distribution (PD) of the strength on the element and specimen scale [104].

ite laminates [104]. It was found that the engineering constants and strength properties of the UD glass-epoxy laminates that were used followed normal distributions. Consequently, a finite element model was created where each of the 24,000 elements was assigned a different value for each of the engineering constants and strength values. The strength in element's scale was argued to be different from the specimen's scale, because failure in the specimen's scale can be attributed to the weakest region in the specimen. Therefore, the authors propose to generate a specimen strength based on an experimental distribution, after which an imaginary mean strength value of the elements is created to form the basis of a different, synthetic normal distribution for the element strengths, as illustrated in Figure 2.13. Both stiffness degradation and fatigue life predictions show to be accurate.

A normal distribution is used as well by Naderi and Maligno to model initial flaws and impurities in carbon-epoxy laminates [105]. The coefficient of variance (CV) that is used for all properties is 0.3%. To mimic localized defects, the properties of five random elements are set to 90%. As such, good agreement is obtained with experimental results in predicting the fatigue life in the form of S-N curves. Van der Meer and Dávila used a similar approach by including one initial defect to trigger damage initiation [106]. The authors state here state that such a method shows to provide similar results as when a distribution is assigned to mechanical material properties.

On the contrary, Lekou and Philippidis show that other distributions than a normal distribution may be more suitable to describe the variability of specific parameters [107]. The authors show a case study for which none of the engineering constants or strength values is best fitted by a normal distriubtion. However, when comparing normal, log-normal, Weibull, Gamma, and extreme value asymptotic distributions, the normal distribution scores decent overall over the different properties. It is concluded that reliability of composite structures is seriously overestimated when the stochastic nature of the material properties is not taken into account.

Jin et al. show an approach where analyses on micro-scale were coupled to the macro-scale [108]. Based on the fiber volume fraction, distributions for the engineering constants were obtained. Fiber and matrix properties followed a normal distribution and were modelled with a CV of 5%. It results in the macro-scale engineering constants following a normal distribution as well. A correlation analysis is presented as well between the fiber and matrix properties and the macro-scale properties: e.g. E_{11}

is under a dominant influence of the modulus of the fiber, whereas E_{22} and E_{33} are most dominantly influenced by the modulus of the matrix and the Poisson's ratio. Analysis of Zhang, Zhang, and Chen is focused on the effect of modelling this correlation between the engineering constants and strengths and underlines that the failure probability can be largely overestimated when correlation is not taken into account [109].

Another example of a multi-scale approach is proposed by Wu et al. [110], where a stochastic non-linear micro-mechanical model is characterized via scanning electron microscope images. The model serves as a surrogate model to generate random mechanical properties as input for a stochastic finite element model. A UD composite ply is used as case study, where stress-strain curves are properly reconstructed up until the point of 'softening onset', after which the proposed solution diverges.

Martinez and Bishay performed a stochastic analysis of first-ply failure of composites under in-plane static tensile loading via Monte Carlo simulation and conclude that the coefficients of variance of material properties highly impact the failure ply, failure load, and probability of failure [111]. CVs of all mechanical material properties are equally set and varied from 1% up to 20% with steps of 1%. Higher coefficients show a reduction in strength, implying that materials with high CVs should take a correction factor into account during design to account for this reduction in strength compared to the theoretical, deterministic model where all CVs are zero.

Table 2.1 provides an overview of some of the standard deviations that were found in mentioned literature.

Table 2.1: Coefficients of variance found in literature for relevant engineering constants and strengths. (*a*): Carbon-epoxy weave [108], (*b*): Glass-polyester UD [107], (*c*): Glass-epoxy UD [104], (*d*): Carbon-epoxy UD, assumed values [105], (*e*): Carbon-epoxy UD [109], *: Weibull distribution

COV [%]	(a)	(b)	(c)	(d)	(e)
E_{11}	3.28	8.94	3.6	0.3	7.0
E_{22}	7.13	14.81	5.5	0.3	5.6
G_{12}	4.72	24.90	4.2	0.3	9.0
ν_{12}	7.32	18.68	1.4	-	4.1
X_T	-	15.07	5.6	0.3	7.0*
X_C	-	10.09	5.6	0.3	-
Y_C	-	13.49	6.8	0.3	-
Y_T	-	14.99	6.8	0.3	5.7*
S_{12}	-	18.06	7.9	0.3	-

Conclusion

Successful modelling of material variability has proven to be both possible and necessary to obtain good results. Furthermore, introducing material variability via FEM allows to overcome the deterministic nature of such a simulation. It means that, given material properties and distributions, an infinite amount of new experiments can be simulated, mainly limited by the required computational resources and the available time. To integrate the data that potentially becomes available from such simulations with other data sources such as experimental data, machine learning is opted as a tool and discussed more elaborately hereafter.

2.4. Machine Learning

Machine learning (ML) is a data-driven approach that covers a broad spectrum of algorithms which have been have been impacting various industrial areas and scientific disciplines [112], from fatigue life predictions up to medical diagnosis of diseases. In general, ML is used to process large amounts of data with the goal of recognizing patterns and regression of models. This section elaborates on its application in SHM of composite materials and damage mechanics of composite materials. Firstly, an overview is provided of the different types of relevant machine learning algorithms and their working principles, with a conclusion on what type of algorithm is most suited to study further in-depth. Next, the selected type of algorithm, deep learning and its backbone: neural networks, are explained together with some examples.

2.4.1. Machine Learning in SHM

The increasing availability of data from sensors that monitor a structure allows for enhanced decision making regarding diagnosis of a structure's condition and prognosis of damage within a structure. Using ML in SHM has been found particularly useful in three scenario's: 1) large amount of available data, 2) physical characteristics that are too complex to model with physics-based models, 3) required reduction in computational efforts [2]. Nonetheless, opposing views exist on the use of ML in SHM: its power has shown to be a welcoming addition to SHM to some (especially during one of the three aforementioned scenario's), while others are hesitant due to the fact that it can be very much like a black box, without underlying physical models. After the presented overview, examples are discussed.

Several ways to classify ML techniques exist. A common first classification of ML techniques is made via the way of learning of the algorithms. Learning is essentially the process of fitting training data to your model. It can be done in four ways: 1) supervised learning, 2) unsupervised learning, 3) semi-supervised learning, or 4) reinforcement learning [113]. Alternative classifications are the way in which it learns "on the fly", i.e. online versus offline, or if they work by comparing data points (instance-based) or by detecting patterns (model-based) [114].

Supervised learning is done by feeding a set of "labeled" training data, i.e. it is specified what the correct output is for a certain input. Supervised learning problems are generally described as classification or regression, dependent on whether the output is categorical (discrete) or numerical (continuous). Unsupervised learning does not use such labels and is aimed at revealing underlying patterns in an unlabeled dataset. Semi-supervised learning uses a mix of labeled and unlabeled training data. Reinforcement learning uses its environment to learn, so it does not have a dataset of fixed shape, and is less popular than supervised learning and unsupervised learning [113]. Supervised learning is the most appropriate for most SHM problems, which requires the fitting of labeled data, such as historical or synthetic damage state data [2].

For supervised learning, the goal is to solve a generalized optimization problem by minimizing the loss over a training dataset, shown in Equation 2.1 [113]. The loss is a function that measures the difference between the true value y and the predicted value $\hat{y}$. The predicted value is a function of the model, which depends on the input x and the model parameters θ. The loss is minimized by tuning the model parameters θ. A regularization term $\xi\Omega(\theta)$ is added to put constraints on the amount that the model parameters are varied when exposed to training data, balancing between letting the model parameters hardly change and letting the model change so much it only fits the most recent training sample it has seen. Optimizing ξ is part of the training process.

$$\underset{\theta}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \varphi\left(y_i, f\left(\boldsymbol{x}_i; \theta\right)\right) + \xi\Omega(\theta) \tag{2.1}$$

As discussed, the objective function of unsupervised learning does not include output values, as can be seen in Equation 2.2, where α represents the model parameters [113].

$$\underset{\alpha}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \varphi\left(\boldsymbol{x}_i; \alpha\right) \tag{2.2}$$

For supervised learning, the input values of the datasets are shaped $n \times p$, where n is the number of datapoints or observations, and p the number of features. Features are also known as attributes, covariates, or independent variables. The output value is shaped $n \times 1$: one prediction for each observation. Before training, feature selection and extraction from the data must be done, keeping in mind which features are the most influential to improve training efficiency of the model, increase its flexibility, or enhance the model's performance. It should be kept in mind that increasing the number of features, or increasing the dimensionality of the problem, increases the challenge of generalization, i.e. makes it more difficult for an algorithm to perform well on new data.

Evaluation Metrics

In general, two factors are important in any machine learning algorithm: 1) a small error between the prediction and the training data and 2) a small gap between the training error and the test error. The latter is also known as the generalization error, which indicates the ability to perform well on data that the model has not seen before. Whenever the gap between the training and test error is too big, the model is overfitting. On the contrary, the model underfits in case the gap between training and testing error is too small. Feeding a model more data will decrease the training error. At some point, it will start overfitting such that the training error keeps decreasing but the testing error starts to increase. Stopping at the right time is thus important to prevent either of the phenomena to occur.

Evaluating the model's performance depends on what metrics you use. Generally, performance is assessed in terms of accuracy and error rate [114], but it depends on the type of problem that is solved. For example, the performance of a binary classification model is generally evaluated by evaluating its accuracy (fraction of all predictions that is correct), precision (fraction of true positive predictions over the sum of true and false positive predictions), and recall (fraction of true positive predictions over the sum of true positive and false negative predictions). The measure of performance needs to be clear before training initiates, because via training, the model should optimze its performance corresponding to the selected metric.

A simple example to illustrate this process is provided using linear regression and the mean squared error, which is a popular performance metric:

$$\mathrm{MSE}_{\mathrm{test}} = \frac{1}{n} \sum_{i=1}^{n} \left(y_i^{\mathrm{test}} - \hat{y}_i^{\mathrm{test}}\right)^2 \tag{2.3}$$

where y_i^{test} is the prediction and $\tilde{y}_i^{\mathrm{test}}$ the true test value in a set of n features. It can also be written as a function of the model parameters and training input:

$$\mathrm{MSE}_{\mathrm{test}} = \frac{1}{n} \sum_{i=1}^{n} \left(y_i^{\mathrm{test}} - \boldsymbol{\theta}^T \boldsymbol{x}_i\right)^2 \tag{2.4}$$

Consequently, one wants to find a set of model parameters for which the MSE is minimal. For this simple example, a closed-form or direct solution can be obtained, which starts with setting the gradient of the MSE with respect to the model parameters to zero:

$$\nabla_\theta \mathrm{MSE}_{\mathrm{train}} = 0 \tag{2.5}$$

Finally, the normal equation can be derived, which is defined as follows:

$$\hat{\boldsymbol{\theta}} = \left(\boldsymbol{X}^{\boldsymbol{T}} \boldsymbol{X}\right)^{-1} \boldsymbol{X}^{\boldsymbol{T}} \boldsymbol{y} \tag{2.6}$$

which is the function that minimizes the MSE, or minimizes the cost function. One can imagine that finding the gradient for a linear regression model is not that complicated. It gets more complex when the algorithm models non-linear behavior, as is the case with neural networks. Training for such algorithms is explained in subsection 2.4.2.

Examples

Popular supervised learning algorithms include [114]:

- Gaussian Process Regression
- Support Vector Machines (SVMs)
- k-Nearest Neighbors
- Decision Trees and Random Forests
- Logistic Regression
- Artificial Neural Networks

The first two are briefly explained here: one type of regression and one type of classification. Examples of implementation in SHM or damage mechanics of composite materials are included. In the section hereafter, neural networks are discussed.

Gaussian Process Regression The Gaussian Process Regression (GPR) is a method that is based on linear regression. The difference lies in the fact that GPR considers errors to be correlated and linear regression does not. The correlation between variables describes the degree in which variables are related to each other: in case the correlation between X and Y is 1, their values move synchronously, while X an Y are completely independent in case their correlation is 0.

Figure 2.14 illustrates the principle of GPR. GPR assumes that all the measured data points are correct and fits all data. During prediction, the uncertainty is described by using the so-called local departure of the global function. The global function, often in the form of a constant or polynomial, has to be provided by the user for both the mean value and covariance. Together, the global function and local departure make up the simulated value of the GPR. Based on the correlation between data points, a prediction is made for which usually the uncertainty increases as the predicted point is further away from the latest measured data point, because the prediction converges towards the selected global function. Therefore, extrapolating with GPR is not that different from "standard" linear regression [115].

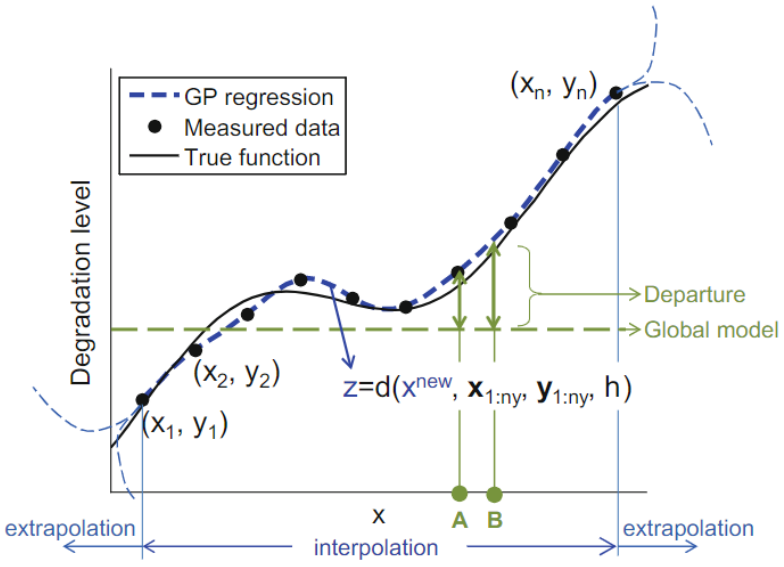

Figure 2.14: Illustration of the working principle of Gaussian Process Regression in the context of a structure's degradation level [115].

Goebel et al. include GPR in a comparison between Relevance Vector Machine (RVM), GPR and a neural network-based approach in the task of damage diagnostics and remaining life estimates [116]. The dataset consists of time series data of the damage state and has high noise content and a sparse measurement interval. It is found that provides a sound framework for the task of prediction but has some limitations as well. One of which is to choose a correct covariance function to interrelate the data. A key ingredient in GPR prediction is that it requires some prior knowledge about the form of the global mean function and covariance function, which should be derived based on the context of the application. Therefore, in case of limited knowledge about the mean and/or covariance function, selecting an appropriate function is difficult. Another observed limitation is that the computational efforts can become a problem in case the training set becomes larger.

Surrogate models are based on the same principle as GPR and mimic complex calculations in a simplistic way to alleviate the computational burden of complex simulations. A surrogate model $\mathscr{F}$ can be generated by exploiting e.g. a FEM model to generate data. Input data is taken from a representative set of samples of the health state where the output can consequently be calculated using a FEM model. The mapping of the input into a data set for the surrogate model can be optimised such that the error is minimised for training data. One example of successful implementation of a surrogate model was by Cristiani, Sbarufatti, and Giglio [87], where the model was used to reduce the computational burden that is associated with real-time updating of a FEM model. Another example is where a surrogate model was used in a DT approach by Karve et al. to alleviate the computational burden of LEFM based fatigue damage predictions [7].

It should be noted, however, that surrogate models can behave quite differently in the inter- and extrapolation regions [115]. In most applications, surrogate models are used for interpolation, while in prognostics the future damage state is predicted, i.e. the model needs to extrapolate. When setting up a surrogate model with a database, the range of values should therefore be sufficienty wide such that the values that are to be predicted are within interpolation range and require only little or no extrapolation.

Support Vector Machine The Support Vector Machine (SVM) is most popularly used as classification algorithm that is used in various pattern recognition tasks. The working principle is easiest explained when looking at Figure 2.15[3]. Here, a binary classification problem is plotted, where the dots are either blue or red. A SVM simply searches for the line that separates these two classes best. It does that by searching for support vectors that have the greatest distance to the line (decision boundary) and both classes separated on opposite sides of it. The decision boundary is used for classification: in this problem, an unknown dot that exists at the left-upper-hand of the decision boundary is classified as blue, whereas on the opposite side it is classified as red. The decision boundary is defined as a hyperplane. In an N-dimensional problem, the hyperplane has N-1 dimensions: in this 2D problem, the hyperplane is 1D, but for a 3D problem the hyperplane becomes a 2D plane. In the example, the decision boundary is a straight line that separates two classes, but in more complex classification problems that are not linearly separable, new dimensions can be added to the data (so by definition, the hyperplane's dimension is also increased) to obtain other shapes of the hyperplane. This trick is known as the "kernel trick".

[3] Retrieved from: https://learnopencv.com/support-vector-machines-svm/ (Consulted on: 24-6-2021)

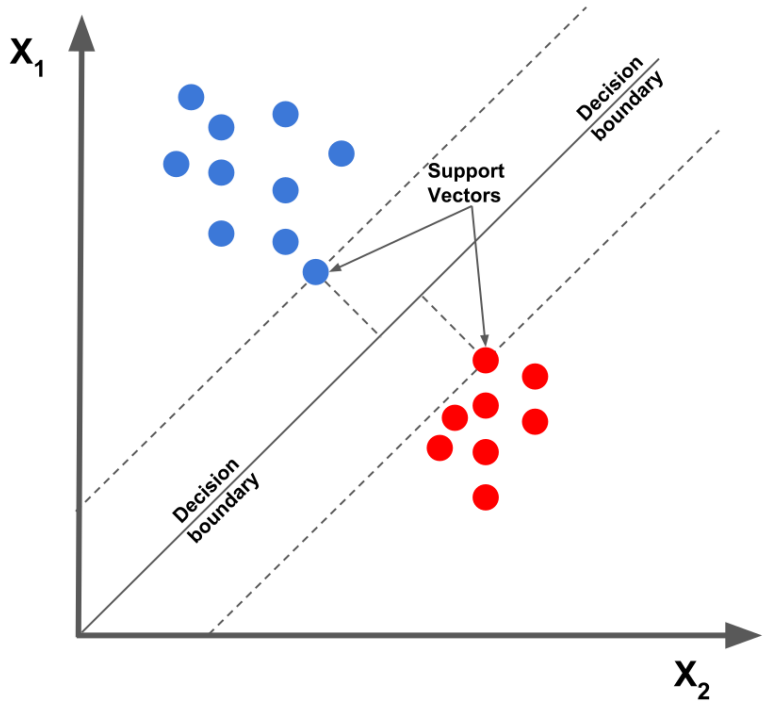

Figure 2.15: Working principle of a classification SVM.

Damage diagnosis within a stiffened panel made of carbon-epoxy prepregs was done using SVMs by Loutas et al. [117]. Data was acquired via Fiber Bragg grating sensors when the panel was loaded under vibrational loads. Different damage states could be distinguished by using a SVM with a non-linear kernel. Several advantages are provided as motivation to use SVM in this context, of which the most interesting are: 1) good generalization capabilities, 2) a single global minimum exists for the objective function (as opposed to e.g. neural networks, which sometimes have multiple local minima), so an optimal set of model parameters can be obtained, and 3) no a priori knowledge of the distribution of the data is required, as is the case with for example GPR.

He et al. present a comparison of three ML techniques to assess delamination in carbon-epoxy beams [118]. Via vibration-based monitoring, the interface, location, and size of delaminations was predicted. Predicting the interface was a classification problem, whereas predicting the location and size of the delamination was predicted using regression. A SVM turned out to outperform a back propagation neural network and an extreme learning machine in terms of prediction accuracy. A split of 80-20 was found to be optimal to split the dataset in training data and validation data.

Conclusion

In selecting the right ML approach, the underlying problem plays a dominant role. In predicting the location and load of the next crack, a distinction was opted: turning the prediction of the location into a classification problem and the prediction of the next crack into a regression problem. By turning the prediction into a classification problem, the crack probability per location could be outputted. Furthermore, discretization of the specimen into separate 'bins' is relatively easily done; discretization in the form of meshing is what FEM does by definition. Considering the stochastic nature of the crack locations, a probability per location was found to provide more information about the underlying relationship than a single continuous numerical output. Regarding the prediction of the next crack load, discretization is not that straightforward. To output a single continuous number for prediction of the next crack load seems more suitable, for which regression is the most suitable option. Even though the nature of the problems is different, the information flow within a digital twin should be integrated, meaning the ML algorithm should be able to perform both classification and regression.

Examples of models that can perform both types of solving are more scarce. Other than artificial neural networks, decision trees are capable of performing both regression and classification. Decision trees are found to not perform as well as ANNs when it comes to nonlinear, noisy data by Curram and Mingers [119]. Furthermore, it was found by Tso and Yau that decision trees are more suitable for predicting simply a categorical outcome rather than time series data [120]. Predicting the crack pattern based on a crack or damage history makes the problem time-dependent. As such, an ANN was found to be the right algorithm for the task. The next section describes the working principle of artificial neural networks and what specific type of network was chosen.

2.4.2. Artificial Neural Networks

With the non-linearities that are assocatiated with damage mechanics in composite materials, the ML algorithm that is used to predict damage accumulation should be able to cope with non-linear behavior. An artificial neural network (ANN) is an algorithm that is well suited for such a task. After a brief description of its evolution throughout time, it is explained how a neural network works, followed by some relevant examples of implementation in SHM of composite materials or modelling damage mechanics of composite materials.

ANNs have been around for almost 80 years now after which their popularity has evolved in waves. In the 1960s funding for research vanished once it became clear that ANNs did not found the basis for truly intelligent machines. Next, in the 1980s the interest was revived when new architectures were developed, after which in the 1990s other ML algorithms such as the SVM were developed and offered better performance, which once again halted further development of the ANNs. Nowadays, its popularity is at a peak again that might stay for a while, because of the huge amount of available data, increased computing power, and improved training algorithms [114].

Recurrent Neural Network

With the goal of predicting crack pattern characteristics, the challenge lies in predicting future crack characteristics based on the current crack state. Therefore, it is evident this requires sequential modelling of the data, which is what a recurrent neural network is designed for. This section explains the working principle of a recurrent neural network. To start with the fundamentals, the general working principle of 'normal' neural network is explained first.

Working Principle - Neuron (Node)

Figure 2.16 illustrates a schematic of a single neuron (or node), firstly introduced as perceptron, with three inputs. These inputs x_i are multiplied with a specific weight w_i, where the sum of these multiplications is added to the bias b. Next, an activation function ϕ is applied to the total sum, which results in the output y of the neuron. An example of an activation function is the rectified linear function, which sets negative input values to zero:

$$\phi_{\text{ReLU}}(x) = \max(0, x) \tag{2.7}$$

More types of activation functions are discussed in the next section.

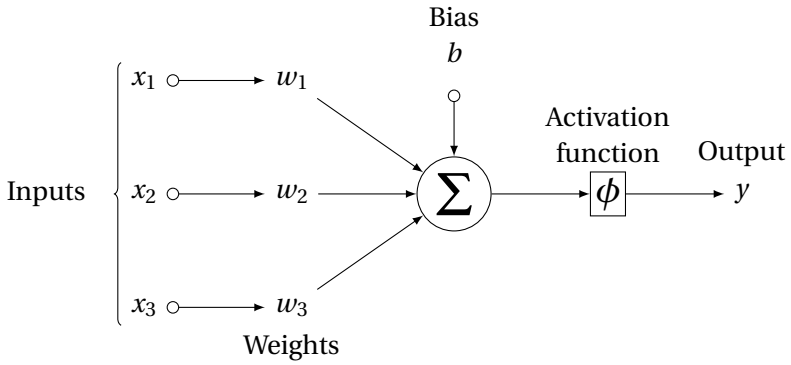

Figure 2.16: Schematic of single neuron with three inputs.

Aligning multiple neurons that are fed by the same inputs is called a layer of neurons. Computing the output y of such a layer is done quite efficiently, simply by using linear algebra on the input matrix $\mathbf{X}$, the weight matrix $\mathbf{W}$ and the bias vector $\mathbf{b}$:

$$y_{\boldsymbol{W},\boldsymbol{b}} = \phi(\boldsymbol{X}\boldsymbol{W} + \boldsymbol{b}) \tag{2.8}$$

A network can include many layers in between an input and output layer. Those layers are called 'hidden layers'. A network with multiple hidden layers is called a deep neural network [121]. The highest accuracy is achieved when the loss is lowest. A popular way to minimize loss is by using a Gradient Descent algorithm, which is a slightly more complex method than using the normal equation, but uses the same underlying principle.

Training - Gradient Descent algorithm

A gradient descent algorithm calculates the weights and biases of the neurons. As described, the weights and biases determine the output of the neuron, which means the weights and biases also determine the loss, which in its turn depends on the loss function that is selected. Random values of the weights and biases are initialized before training and can be used to make a first prediction when applied to an input value, from which a loss can be computed. The loss can be computed for different model parameter settings (and using different input values) and finally, a combination needs to be found for which the loss is minimum. The gradient descent algorithm initializes a random set of model parameters and from there, finds its way down to a minimum loss by calculating the local slope of the loss (or cost) function, as is illustrated in Figure 2.17. It makes gradient descent very powerful, especially in scenarios where setting the derivative of the loss function with respect to the model parameters to zero cannot be solved.

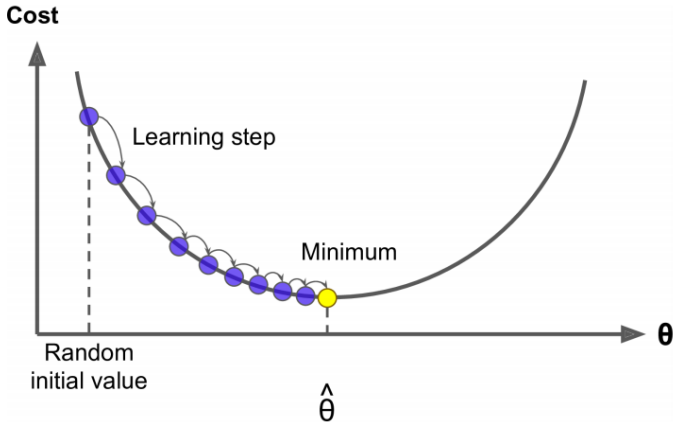

Figure 2.17: Gradient descent [114].

Training is essentially tuning the weights and biases from Equation 2.8 until the network is able to

accurately produce the desired output from input that it has never seen before. When a neuron is fed with training input, it will make a prediction. The degree in which the weights are corrected is dependent on the gradient of a weight with respect to the loss. In case the gradient is steep, it means the weight is far off from its optimal value meaning a higher correction is required than in case of a small, flat gradient. In other words, a small difference between the prediction and target value yields a small change in weights, whereas a large difference yields a larger change in weights. As depicted in Figure 2.17, the learning rate determines the level in which weights are adjusted during a learning step.

In case of more complicated multidimensional spaces, a learning step that is too small may get the algorithm stuck into false local minima and show slow convergence. On the contrary, if it is too big, the algorithm may overshoot, become unstable, and diverge. Figure 2.18 illustrates the idea.

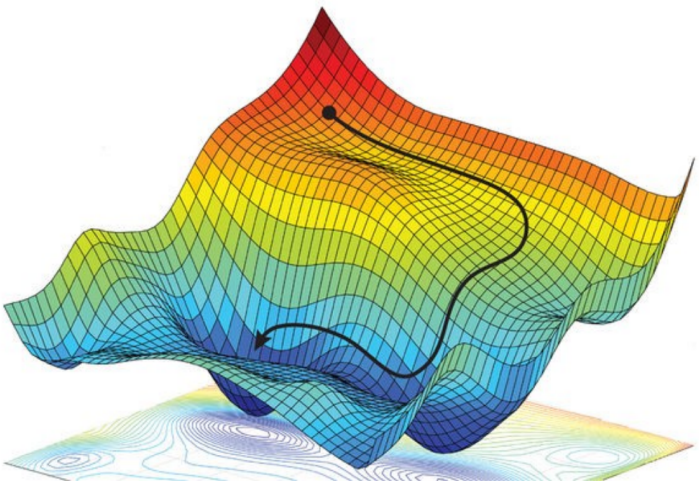

Figure 2.18: Using gradient descent to minimize loss in a 3D space[4].

Most optimizers are based on the stochastic gradient descent algorithm (SGD), which is a modification of the gradient descent algorithm as described earlier. The 'basic' algorithm uses an entire batch of training data to update the weights and biases for each epoch (each round of passing data forward and backward through) and is therefore called 'batch gradient descent'. One can imagine that in case of a very large data set and/or many layers with many neurons, using the entire data set for each epoch can become computationally exhausting.

Stochastic gradient descent tackles this problem. Instead of using the entire batch per epoch, a randomly picked subset from the training set is used to compute the gradients. Naturally, this speeds up training significantly. Disadvantages are, however, that the cost function tends to decrease in a 'bouncy' manner: on average the loss is decreasing during training but not in a straight line. However, this disadvantage has an upside: it allows the model to escape local minima. Altering the batch size allows to control this behavior. Using SGD or SGD-based optimizers can be particularly useful when training via GPU's, which allows more effective matrix operations than running via a CPU.

The described gradient descent-based algorithms have in common that when updating the weights, previous steps in updating are not taken into account. Momentum optimization does take the previous gradients into account. An analogy to explain the principle: if the gradients are the velocity of changing in the weights in gradient descent, the gradients become the acceleration in momentum optimization and momentum itself becomes the velocity, i.e. a vector that changes the weights and biases. An alternative analogy: consider a ball rolling down a hill. If it keeps rolling, it will pick up velocity and reach the bottom of wherever it is rolling faster than if it would keep its speed. Using momentum means the learning rate is not fixed, as is the case with stochastic gradient descent. To prevent momentum from overshooting, friction is added to the momentum vector in the form of a

[4]Obtained from: https://towardsdatascience.com/coding-deep-learning-for-beginners-linear-regression-gradient-descent-fcd5e0fc077d (consulted on 25-8-2021).

decay rate, 0 being high friction and 1 being no friction.

Backpropagation algorithm

Training can be done by using the backpropagation algorithm, which in essence uses a very efficient gradient descent algorithm. The gradients are automatically calculated in two steps: feeding data a step forward and feeding the outcome a step backward. Based on the output that is produced during the forward pass, the weights and biases are adapted during the backward pass such that the error is decreased. Each time that data (either in a batch or the whole data set at once) is passed forward and backward through the neurons is called an epoch. In case of multiple hidden layers, the data is passed forward through all layers first, which results in a final output and corresponding error. Next, it is calculated how much each connection in the network contributed to the error (or cost) by using the chain rule:

$$\frac{dL}{d\theta} = \frac{dL}{dy} \cdot \frac{dy}{d\theta} \tag{2.9}$$

θ representing the weight and/or bias of a node. The backward pass calculates the error gradient with respect to θ by passing it through the network backwards, layer by layer. It should be noted here that in case of multiple hidden layers, the gradient $\frac{dy}{d\theta}$ becomes a function of all other gradients of the nodes that data passes through from the output during the backward pass to reach a node closer to the input layer.

At last, a gradient descent step is taken to adapt the network's weights and biases corresponding to the determined error gradients that were computed from w and b to w^{nextstep} and b^{nextstep}:

$$w^{\text{nextstep}} = w - \eta \frac{dL}{dw} \tag{2.10}$$

$$b^{\text{nextstep}} = b - \eta \frac{dL}{db} \tag{2.11}$$

where η is the learning rate. In other words, the values of the gradients are calculated and based on those values, it is determined how close a parameter is towards minimizing the loss function, just like what is illustrated in Figure 2.17. An increase in dimensionality of the problem, i.e. including more features, makes it more complex to find the optimal set of model parameters to minimize the loss function.

Activation functions

A key feature of ANNs is that they can fit non-linear data. The reason for that is the option for an activation function that is not linear. Looking at Equation 2.8, it is evident that for a layer of "standard" neurons one would sum functions like: $y_1(x) = 2x - 3$, $y_2(x) = 4x + 1$, and so on, which would result in the hidden layer to act as a linear function as well. Thus, a non-linear activation function must be used to be able to model non-linear behavior. Figure 2.19 shows three popular activation functions and their derivatives that can be used for it: the sigmoid function (Equation 2.12), hyperbolic tangent function (Equation 2.13) and rectified linear (ReLU) function (Equation 2.14). The activation functions are tailored to the desired shape by multiplication with the weights or summation with the biases. The activation function that one selects highly depends on the application. For example, it is an important setting in the vanishing gradient problem, which is explained later in this section, and also determines the output format in the output layer.

$$\phi_{\text{Sigmoid}}(x) = \frac{1}{1+e^{-x}} \tag{2.12}$$

$$\phi_{\text{Tanh}}(x) = \frac{2}{1+e^{-2x}} \tag{2.13}$$

$$\phi_{\text{ReLU}}(x) = \max(0, x) \tag{2.14}$$

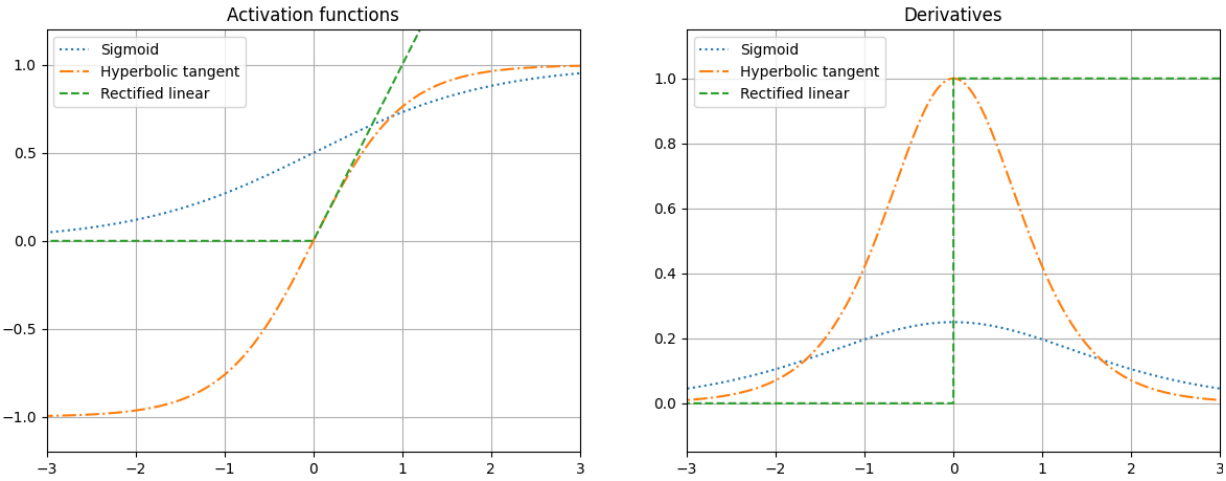

Figure 2.19: Popular activation functions and their derivatives.

Recurrent neuron

Recurrent neural networks (RNNs) are focused at modelling sequenced data, which can come in many forms: audio, text, images, etc. Like other neural networks, it consists of neurons, which are nodes through which data flows. Figure 2.20a illustrates the simplest form of a RNN: a single neuron that gets input $\mathbf{x}_{(t)}$, produces an output $\mathbf{y}_{(t)}$, and sends the produced output back to itself. This process happens within the same neuron for each timestep, as illustrated in Figure 2.20b. The same principle is illustrated for a layer of recurrent neurons in Figure 2.21, where the input and output of a layer are now a vector rather than a single scalar.

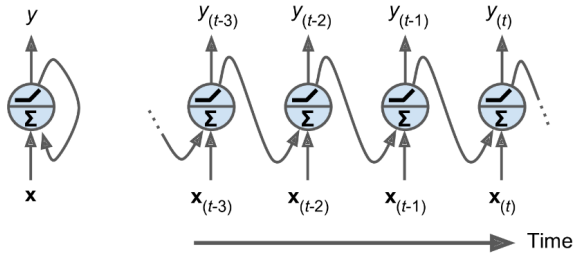

Figure 2.20: *a)* a recurrent neuron *b)* an unrolled recurrent neuron through time [114].

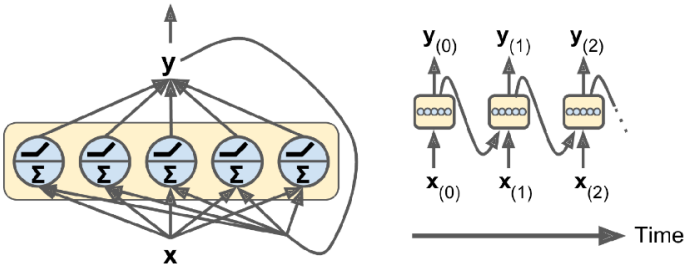

Figure 2.21: *a)* a RNN layer *b)* an unrolled RNN layer [114].

Both the input $\mathbf{x}_{(t)}$ and the output of the previous step $\mathbf{y}_{(t-1)}$ are assigned weight vectors, $\mathbf{w}_{(x)}$ and $\mathbf{w}_{(y)}$. In case the whole layer is considered, the weight vectors of the single neurons can be combined in to a single matrix, $\mathbf{W}_{(x)}$ and $\mathbf{W}_{(y)}$. The equation for the output vector becomes:

$$\mathbf{y}_{(t)} = \phi\left(\mathbf{W}_x{}^{\top}\mathbf{x}_{(t)} + \mathbf{W}_y{}^{\top}\mathbf{y}_{(t-1)} + \mathbf{b}\right) \tag{2.15}$$

where **b** is the bias vector and ϕ the activation function. In a typical display of a general neural network, the nodes would represent the biases and the connecting lines between the nodes represent the weights. The weights are preferred to have values between 1 and -1 during initializing of the model's layers to prevent an explosion of the numerical values.

As mentioned before, the activation function is called after the weights and biases are applied. In case no activation function is used, it will become impossible to fit nonlinear behavior because the model behavior will be a result of simple, linear multiplication and addition of the weights and biases respectively. Similar to a plain or 'vanilla' neural network, at least two hidden layers are required for a RNN to fit nonlinear shapes, as well as nonlinear activation functions, such as one of the listed functions shown in Figure 2.19.

It should be noted that $\mathbf{y}_{(t)}$ is a function of all inputs since $t = 0$. The part of the network that memorizes its state is a (memory) layer. Such a layer includes a state function $\mathbf{h}_{(t)}$, which is a function of both the inputs at that time step and the previous states. The "h" stands for hidden. This hidden state is generally equal to the output, but in more complex cases they can differ. Typically, a single neuron in a RNN can preserve this hidden state for about 10 steps because of vanishing gradients, although this depends on the task. This limitation is addressed in the section hereafter.

You can input a sequence of inputs into an RNN while it simultaneously produces a sequence of outputs, which is called a sequence-to-sequence network. A sequence-to-vector network ignores all outputs but the final one while feeded with a sequence of inputs. Alternatively, you could input a vector while it outputs a sequence, called a vector-to-sequence network. Lastly, you can combine a sequence-to-vector network with a vector-to-sequence network, or encoder-decoder. This two-step model generally is preferred over a simple sequence-to-sequence model, but is more complex to use and generally serves a different purpose, for example language processing. The different forms are schematically depicted in Figure 2.22.

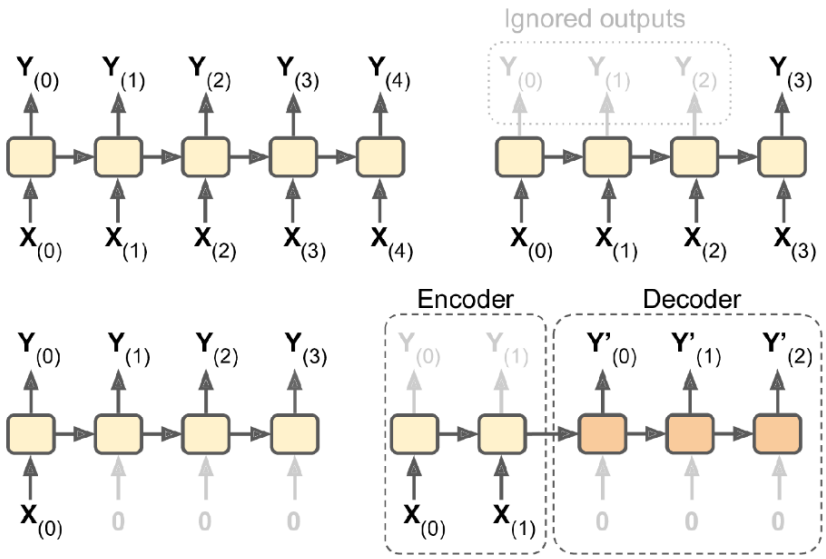

Figure 2.22: Starting top left clockwise direction: sequence-to-sequence, sequence-to-vector, vector-to-sequence, encoder-decoder [114].

Training of RNN based layers

During the training of a RNN(-based) layer, it is first unrolled through time and then backpropagated, which is called backpropagation through time (BPTT). It is visualised in Figure 2.23, where the dashed arrows indicate the initial forward pass through the unrolled network and the solid arrows represent the gradients that results from the loss function that are propagated backward through the unrolled network.

Compared to a standard feed forward model, backpropagation becomes more complicated and ex-

pensive due to the relevance of time: backpropagation in RNNs is done at each individual timestep and across all the timesteps. This requires many repeated multiplications and calling of activation functions. However, it should be noted that the training of a LSTM layer can be done by using parallel processing of GPU's, which accelerates training with respect to CPU's significantly.

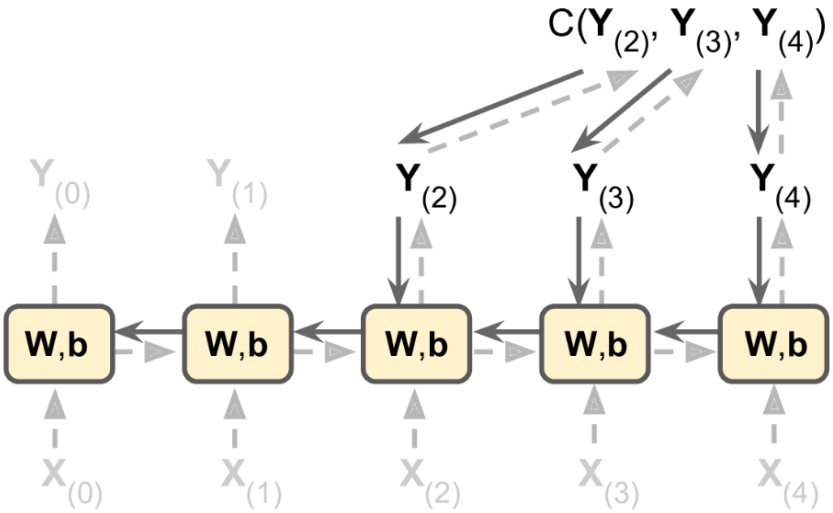

Figure 2.23: Backpropagation through time, subscripts indicate the time step [114].

Limitation — Vanishing/exploding gradient problem

A limitation of the RNN that is important in the context of this project is its limited amount of information that is contained over time: approximately 10 time steps. Considering that the number of matrix cracks in a 100 mm specimen generally exceeds 10 cracks, it does not "possess" sufficient memory for predicting e.g. the 15th or 20th crack based on the entire damage accumulation history. The reason for this limited memory is caused by the vanishing gradient problem, which is not exclusive to RNNs.

The gradient here is the gradient of the loss function with respect to the weights, which is calculated by using backpropagation as described earlier. The gradient of the loss with respect to the weights of a layer is determined by gradients of layers that are placed later in the network. For example, considering a neural network of five layers between the input and output layer: the gradients of the first layer after the input layer are calculated as a multiplication function of the gradients of the four layers that are placed after it. In case the four layers after it have gradients that are smaller than 1 and/or close to zero, you essentially multiply fractions close to zero with each other which results in a gradient that is also close to zero, i.e. the gradients vanish.

Since the updating of the weights is done proportionally to these gradients, it means that for a very small gradient the weights are hardly updated. As a consequence of the weights not being changed during training, the corresponding neurons can no longer significantly contribute to minimizing the loss function, i.e. improve the network's performance. On the contrary, if the gradients of layers in are larger than 1, gradients of early layers in the network risk exploding to huge values which would correspond to exploding weights as well during training.

Solution — Long Short-Term Memory

Several solutions have been proposed over the years to overcome this limitation. The Long Short-Term Memory (LSTM) cell is the most popular solution to this problem for a RNN. Even though it was first proposed already in 1995, none of the variants manage to significantly improve on the original LSTM's performance [122]. LSTM based ANNs have been applied in a wide variety of applications: from predicting Bitcoin's price [123] to epileptic seizure detection [124], implying a proper level of generalization.

Figure 2.24 illustrates the architecture of a LSTM cell within each time step. It has both a short-term state $\mathbf{h}(t)$ and long-term state $\mathbf{c}(t)$. Simply put, the long-term state first forgets some memories and afterward adds some memories. Next, it is passed on to the next time step and copied to pass through a activation function (a tanh activation function in the example), after which the result is filtered by the output gate. The result is the short-term state. The long-term state of the LSTM cell allows gradients to flow through different time steps or layers without vanishing or exploding.

The input $\mathbf{x}_{(t)}$ and previous short-term state $\mathbf{h}_{(t-1)}$ (which is the same as the output of the previous time step $\mathbf{y}_{(t-1)}$) are fed into four functions: $\mathbf{f}(t)$, $\mathbf{g}_{(t)}$, $\mathbf{i}_{(t)}$, and $\mathbf{o}_{(t)}$. $\mathbf{g}_{(t)}$ is the main function from which the most important information is stored in the long-term state. A standard RNN cell only consists of this function. The other three layers are gate controllers with output values between 0 and 1: closing the gate at 0 and opening it at 1. The forget gate $\mathbf{f}_{(t)}$ controls what information is forgotten, $\mathbf{i}_{(t)}$ controls what information of $\mathbf{g}_{(t)}$ should be added to $\mathbf{c}_{(t)}$, and $\mathbf{o}_{(t)}$ controls what information should be outputted to $\mathbf{y}_{(t)}$ and $\mathbf{h}_{(t)}$. The forget gate $\mathbf{f}_{(t)}$ and output activation function are the most critical components with regards to the performance [122].

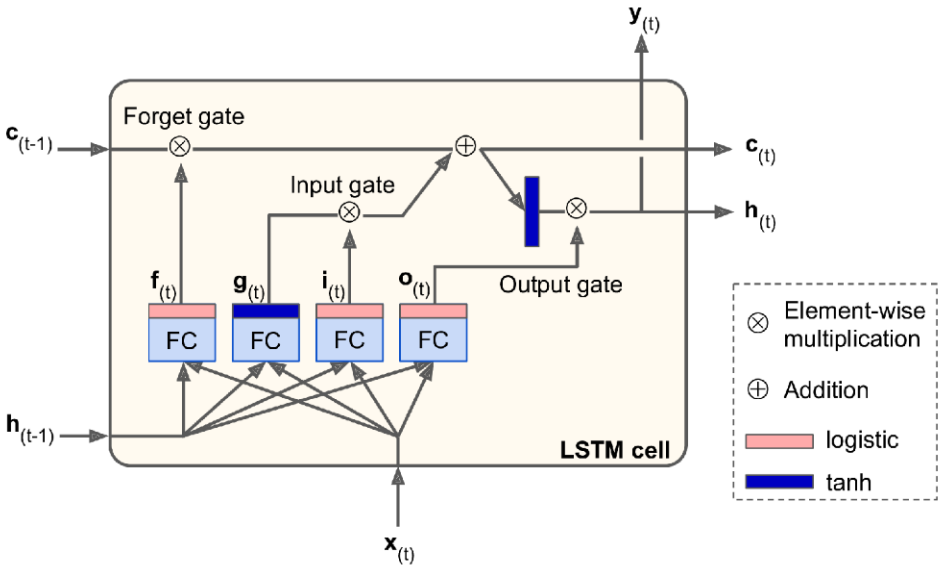

Figure 2.24: LSTM cell [114].

Examples

Califano et al. used an ANN to perform strain-based SHM on composite structure to perform damage identification [24]. Strain data was acquired by using optical fibre sensors that were embedded in the material of a specimen that was subjected to several four-point bending tests. It is proposed that the advantage of using ANN with respect to other machine learning techniques is that it only distinguishes healthy/positive or damaged/negative samples, making it computationally efficient and thereby less time-consuming.

Mucha et al. presented an approach in which an ANN is used to verify and validate strain gauges that were used for the online load monitoring of a complex composite plate (a curved plate stiffened by omega-stiffeners) [125]. An ANN was trained using FEM showing real-time implementation is possible.

Danish, Taghipour, and Lee constructed a digital twin of turbofan engine performance degradation and prognostics using an LSTM-based neural network [3]. They demonstrated accurate RUL estimations when compared to previous models that were used on the same data set, which consists of 100 run-to-failure information samples. Mentioned advantages of using LSTM-based neural networks in deep learning based prognostics are accurate non-linear data processing and the fact that no underlying assumptions about a degradation curve or model are required to be able to generate a state-of-art RUL estimation.

3
Research Project

This chapter elaborates on the research project in terms of the objective, research questions, hypotheses and overall approach. The main research objective of this thesis is to enhance our understanding of damage evolution in composites and improve on health assessment during its service life by constructing a digital twin of a cross-ply specimen loaded in quasi-static tension.

A relevant case study object for an implementation of the digital twin in the damage accumulation process of composites is transverse matrix cracking in a cross-ply specimen, as becomes clear from section 2.2. Recent research of Li et al. is aimed at investigating damage accumulation in cross-ply specimens, both under quasi-static and fatigue loading, and provides data to use as case study [31, 126]. The experimental data provides the opportunity to validate the digital twin and advance the ongoing research. Even though the cross-ply specimen is a suitable case study object, augmentation of the existing experimental data set is still essential to make the data set compatible with a data-driven technique such as a recurrent neural network.

A numerical FEM model was opted to provide the augmentation, where two main challenges arise: 1) the complexity of modelling the interaction between matrix cracks and delaminations, and 2) overcoming the deterministic nature of FEM. From subsection 2.2.4, XFEM-CE was deemed as a suitable method to model the interaction between the two damage mechanisms. Furthermore, to overcome the deterministic nature of FEM, the options that were listed in section 2.3 may form a basis to implement material variability into the FEM model. Next to the advantage of being able to generate various damage states with the same model, including the material variability accounts for the inherent variability of composite materials, which should improve the reliability of the simulations.

With regards to the real-time application of the model that processes complex information, machine learning algorithms seem interesting tools that have found their way increasingly in (aerospace) structures. It has been found particularly useful in three cases: 1) large amount of available data, 2) physical characteristics that are too complex to model with physics-based models, 3) required reduction in computational efforts [2]. Especially the last two descriptions are quite applicable to the case of real-time monitoring of the complex damage mechanisms. As mentioned, the only thing that lacks here is an exhaustive data set, which is why augmentation of the data set is addressed via the FEM model in chapter 6, but may require further augmentation procedures. Based on section 2.4, a RNN with LSTM cells was opted to serve as digital twin that processes information and predicts the next crack in real-time.

Example features of advanced digital twins such as presented in section 2.1 include multi-scale, multi-physics, probabilistic simulations that integrate heterogeneous information sources and process it with the purpose of decision making in a fully autonomous manner. Creating such an exhaustive, fully-equipped digital twin for all lifecycle stages is too ambitious for the scope of a 9-month project by one master student. When considering the (im)maturity of the concept as well, smaller steps have to be taken first. Key aspects that make such a model a digital twin are the identification and processing of changing state variables, such as damage parameters of the test subject, in real-time. These aspects fit in well within many engineering and scientific disciplines next to structural health monitoring; its versatility is what makes the digital twin so interesting from a broader perspective.

To synthesize, a digital twin can serve as a tool to solve the problem of modeling complex damage accumulation processes with the final purpose of structural health monitoring or validation of other

structural health monitoring techniques. Key building blocks of the DT concept are developed to address the need to capture complex underlying patterns of the damage accumulation process in composite materials. By focusing on the processing of heterogeneous data in real-time to predict the next crack location and load within a cross-ply, a fundamental model is built that covers the research objective and may form the basis for next steps in further research, which may be focusing on including different life stages, remaining useful life, and/or other topics that relate to damage accumulation within a composite material.

3.1. Research Questions

The main research question of this thesis is:

> To what extent can the crack pattern of transverse matrix cracks in a cross-ply be predicted using a digital twin?

With the sub-questions:

1. How can the material variability that is inherent to composite specimens be modelled into a FEM model?
2. To what extent can XFEM-CE simulate the interaction between transverse matrix cracks and delaminations in a carbon-epoxy cross-ply specimen under tensile quasi-static loading?
3. How well can a neural network be trained with an augmented dataset to predict the location and load of the next crack of a carbon-epoxy cross-ply under quasi-static tensile loading?

3.2. Hypotheses

Via a hybrid approach where a physics-based model and experiments form the basis of data-driven predictions, the crack pattern in a cross-ply specimen under quasi-static tensile loading is expected to be predicted reasonably well. Specifically, the crack pattern of transverse matrix cracks is hypothesized to be predicted well from a certain point on. After all, the first few cracks will always be caused by randomness of the distribution of material properties, which is not predictable unless the (micro)mechanical structure of the material is analyzed and included in the analysis. It would be interesting to find out when the 'stochasticity' in crack appearance is more or less modelled out.

In case of a positive outcome, XFEM-CE would show to be fast enough to sufficiently augment a database, in spite of a uniform and relatively fine mesh requirement due to the fact that cracks may occur anywhere in the matrix. Furthermore, fast and successful integration of the heterogeneous data with a neural network underlines its performance when it comes to fast processing of highly non-linear data. Moreover, in case the models integrate well, it may form a basis of the corresponding predictive model of another specimen. In other words, if the FEM model succeeds in sufficiently augmenting the data set, the hybrid combination of a FEM and data-driven model can relatively easy be adapted to be used for a specimen that is made of a different material or a different layup. It would make it easier to use the models as validation tools for SHM techniques.

In case of a negative outcome, it should be noted that the models are not dependent on each other. Another method than XFEM-CE may be found to model the damage mechanisms, or even a different type of numerical model than FEM to augment the data set. Similarly, a different type of neural network or machine learning algorithm in general, such as decision trees, may turn out to be more suitable.

3.3. Project Approach — Flowchart

Figure 3.1 presents a flowchart that links the different subjects and steps that were taken in this thesis. The red boxes show how the experimental set-up and the corresponding crack patterns are linked to various other parts of the project. A description of the cross-ply and the experimental set-up is described in chapter 4. Within the FEM environment, the properties and geometries as derived from the experiments form the basis for the Abaqus model. Once the model was configured and data was

post-processed, it was assessed whether the model sufficiently fitted the experimental data. Once it did, synthetic crack patterns were generated. Actions and results directly related to FEM are displayed in blue boxes and discussed in chapter 5. Lastly, the neural network or NN environment, which starts with merging the synthetic and experimental crack patterns, followed by further augmentation in the purple triangle. The green boxes indicate neural network steps and are explained in chapter 6. Data preparation was done firstly, to obtain the data set that the network was trained with. During model exploration, various architectures were explored, after which only the best performing ones were finetuned until the crack prediction was satisfactory.

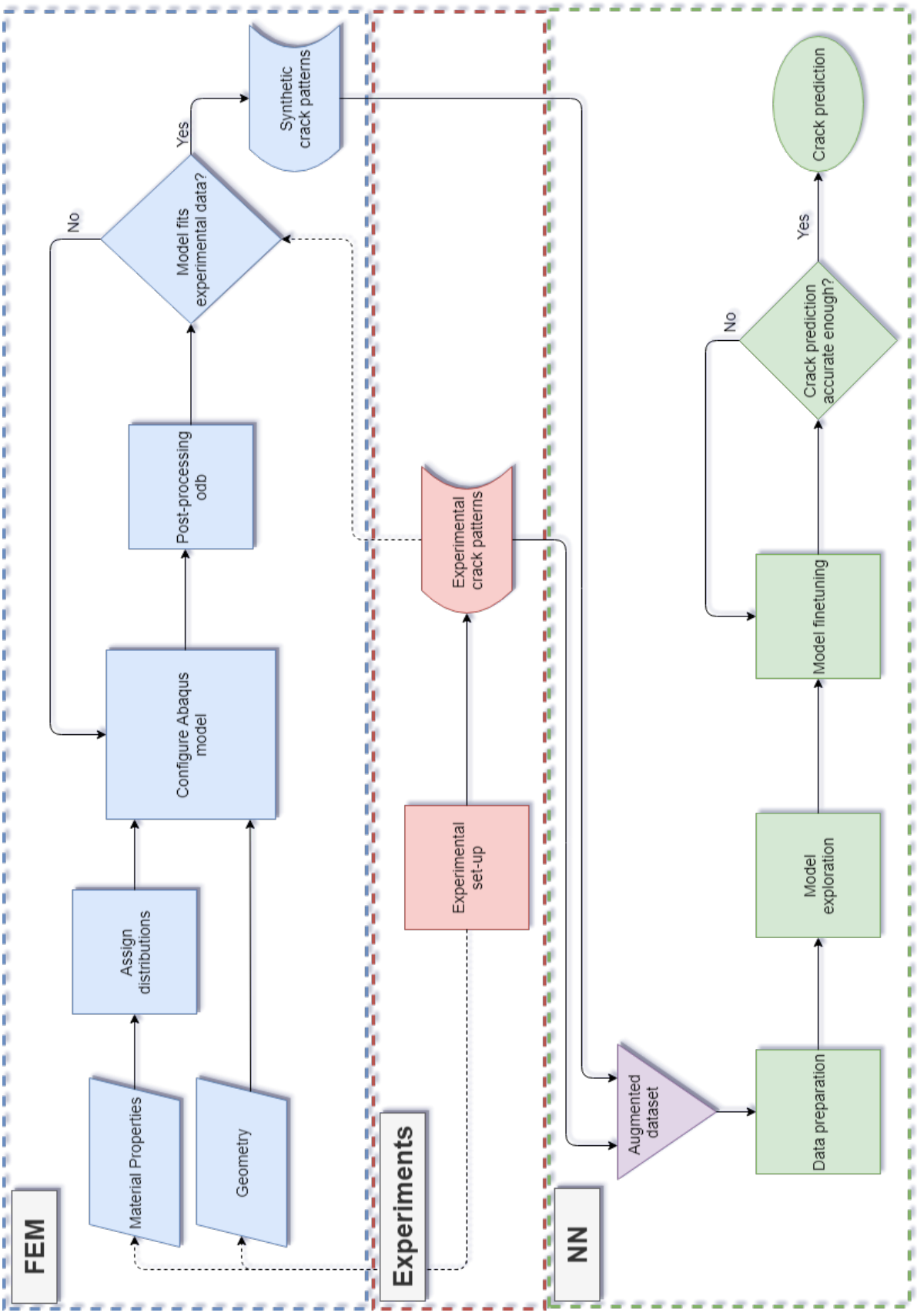

Figure 3.1: Flowchart of the project approach.

4

Physical Twin - Case Study Object

This chapter explains the specifications of the physical twin in this thesis, i.e. the case study object. The physical twin is the cross-ply specimen that was used by Li et al. to test under a quasi-static load with different load rates [31], meaning that this chapter essentially describes the work of Li et al. The specimen itself is described in section 4.1, followed by an explanation of the experimental set-up in section 4.2. The final section in this chapter displays the data that forms the basis of the data set that is ultimately used to construct a data-driven digital twin.

4.1. Cross-ply Specimen

A cross-ply was used to study the effect of the loading rate on damage accumulation [31] and to study early fatigue damage [126]. The specimens are made of Hexply F6376C-HTS(12K)-5-35%, which is a UD carbon fiber prepeg with high-tenacity Tenax-E-HTS45 fibers and a tough epoxy matrix: Hexply 6376. The properties of the material are shown in Table 4.1. The nominal fiber volume fraction is 65% and a single ply has a thickness of 0.125 mm. Specimens were cut from autoclaved panels into rectangular shapes with a size of 250 x 25 mm and a layup of $[0_2/90_4]_s$.

Table 4.1: Material properties of the used Prepreg obtained from the datasheet [127].

Property	Symbol	Value	Unit
Longitudinal Modulus (tensile)	E_{11}	142	GPa
Transverse Modulus (tensile)	$E_{22} = E_{33}$	9.1	GPa
Transverse shear modulus	$G_{12} = G_{13}$	5.2	GPa
In-plane Poisson's ratio	$\nu_{12} = \nu_{13}$	0.27	-
Transverse Poisson's ratio	ν_{23}	0.3	-
Longitudinal strength (tensile)	X_T	2274	MPa
Longitudinal strength (compressive)	X_C	1849	MPa
Transverse strength (tensile)	Y_T	102	MPa
Transverse strength (compressive)	Y_C	255	MPa
In-plane shear strength	$S_{12} = S_{13}$	63	MPa
Transverse shear strength	S_{23}	35	MPa

4.2. Experimental Set-Up

To improve clamping grip, paper tabs were glued on the ends of the specimen. Specimens were placed in a 60 kN fatigue machine with hydraulic grips. The setup was monitored during testing with two edge camera's, two acoustic emission sensors, and two camera's that were used for DIC monitoring. The region that was monitored by the edge camera's and the acoustic emission sensors spans 100 mm, which was taken as the length of the FEM model that is described in chapter 5. Figure 4.1 illustrates a schematic diagram of the specimen within the experimental set-up, which also formed the basis of the sketches of the FEM model.

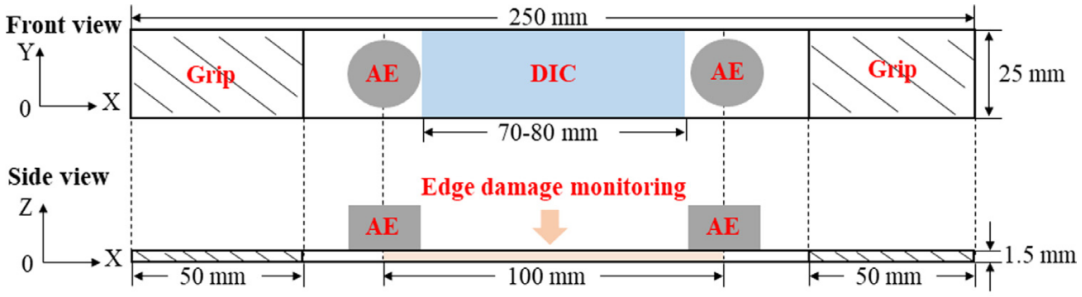

Figure 4.1: Schematic diagram of the experimental setup and specimen [31].

4.3. Data Acquisition

The positions and loads of the cracks of eight specimens that were used during the mentioned experiment [31] were available to form the basis of the data set. The different loading rates showed different damage accumulation evolutions, which is illustrated in Figure 4.2. The finite element model that was built takes into account stochasticity of material properties, but not of the loading rate effects. Discrepancies in results may partially be explained by this. Nonetheless, the physical experiments provide a scatter which ideally is captured by the finite model as well, even though the cause of scatter may differ.

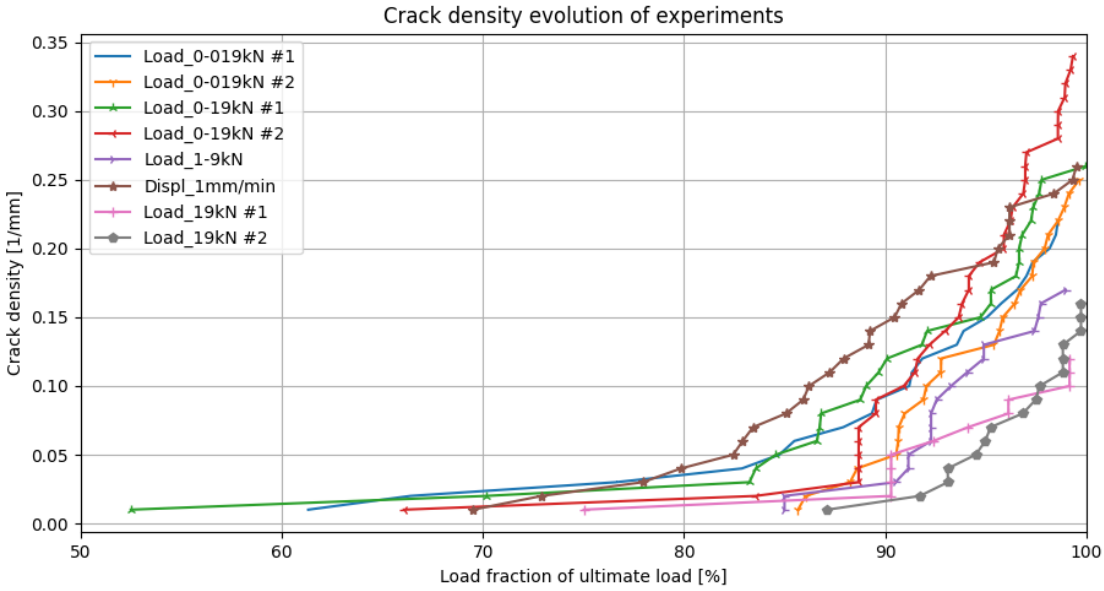

Figure 4.2: Crack density evolution curves of the experiments. 'Load' or 'Displ' indicates load-controlled or displacement controlled, followed by the loading rate ('-' indicates the range of loading). In case the same load rate was tested twice the different specimens are numbered by '#'.

5
Augmenting the Dataset via Abaqus

To augment the experimental dataset that is described in chapter 4, a finite element model was simulated in Abaqus [128] via a script made in Python [129]. To overcome the deterministic nature of a finite element model, the material properties were varied per element. Two damage mechanisms were modelled as observed in experiments: intralaminar cracks in matrix and fiber and interlaminar cracks between the interfaces of the longitudinal and transverse blocks, i.e. delamination. After a simulation, data is collected, analysed, verified, validated, and processed into input for the data-driven model that is described in the next chapter.

Firstly, underlying assumptions of the model and damage theory of Abaqus are discussed before going into the details of the model in section 5.1. The model itself is discussed in section 5.2. Next, the results are shown and discussed in section 5.3 and a sensitivity analysis is presented in section 5.4. Finally, the chapter is concluded with a conclusion and recommendations in section 5.5.

5.1. Preliminary Considerations

In subsection 5.1.1, the assumptions that were made before setting up the finite element model are presented. Next, the options to model initiation and propagation of transverse matrix cracking and delamination are explained in subsection 5.1.2.

5.1.1. Assumptions

Assumptions that were made for this model are stated and discussed hereafter.

1. The cross-ply specimen can be modelled as 2D.
   (a) Transverse strength of the matrix is constant in the width direction.
   (b) Transverse matrix cracks propagate instantaneous through the width direction.
2. Plane strain conditions apply.
   (a) Strains in the width direction are zero (length: x, thickness: y, width: z): $\varepsilon_{zz} = \gamma_{xz} = \gamma_{yz} = 0$.
3. The mass of the specimen does not play a significant role.
4. The only ply-interface bonds that can fail are those on the two 0-90 interfaces.
5. Failure behavior of both the 90-block and 0-block is brittle.
6. Loading is perfectly aligned.
7. All transverse matrix cracks propagate perpendicular to the length direction.
8. Delaminated interfaces slide over each other in a smooth, frictionless manner.

Assumption 1: Assumption 1b was observed during the experiments [31] and in literature [103], from which assumption 1a was concluded: that the strength in the width direction can be considered to be constant, or at least not varying significantly enough to affect the direction of the crack propagation. Therefore, the width direction was considered to be irrelevant to model, which is in line with the findings of, amongst others, van der Meer [130]. Besides that, the computational cost of adding the width direction was expected to be very large, leading either to a coarser mesh size (and hence, less accuracy in the solution in the plane of interest) or too much runtime of the model.

Assumption 2: Due to the presence of the thick 90-block, a lot of stiffness is present in the width direction. The deformation in width direction is in reality not zero, but neglibigle. Furthermore, the assumed load distribution becomes uniform. In theory, that is valid, because of the purely on-axis axial load that is applied and the layup. The load is applied along a principal axis without any out-of-plane stresses, which is confirmed by experimental observations, apart from a very small out-of-plane deformation at the moment of failure at the point of failure, as is presented in section 5.3. Lastly, Wang et al. showed that using plane strain conditions provides good predictions of the load-displacement response and damage evolution in a 2D FEM model that uses integrated XFEM-CE to simulate multiple failure modes in a stiffened panel [131].

Assumption 3: The mass of a single specimen is approximately 15 g, meaning the weight would be about 0.147 N. The applied stress right before failure is around 700 MPa, or 18.6 kN, which is about 125,000 times larger than the weight. Therefore, it is safe to say gravity does not play a role in this experiment.

Assumption 4: This assumption is based on the experiments. During the experiments, there was no delamination observed at a different location than the interfaces at connections between a longitudinal ply and a transverse ply before failure of the specimen.

Assumption 5: The failure behavior is important with respect to the failure criteria that can be picked from. It was observed during the experiments that fiber failure occurred instantaneous and resulted into immediate failure of the specimen. Similarly, the transverse cracks in the matrix initiated and propagated instantaneously as well. Little to none (local) ductile yielding behavior was observed, which is why this assumption was made.

Assumption 6: In case loading is not aligned, a off-axis loading is applied instead of a purely axial loading. Different mechanical behavior and damage mechanics are to be expected in such a case. That is also why a cross-ply is not considered a practical layup for applications other than experiments. Furthermore, statistical correlation between properties no longer influences the failure probability when the loading is perfectly aligned [109].

Assumption 7: In fact, the majority of the cracks is observed to be perpendicular to the length direction, but two exceptions are observed: 1) a curved crack that initiates close to an already existing crack, 2) two cracks that initiate from opposing 0-blocks at a slightly different position along the length of the specimen and then propagate towards each other to *meet in the middle*. That means that due to this assumption, a few cracks are modelled differently than what may happen in real life. However, severe convergence issues arise when this assumption is not implemented, which will be further explained in subsection 5.4.4.

Assumption 8: Due to the fact that the expected dominant fracture mode of delamination at the interface is mode II, friction will play a role. Not including friction therefore may result in local inaccuracies of the FEM model during damage evolution or after complete failure of the interface with regards to delamination. Nonetheless, the main effect of delamination is that it initiates at a crack tip and relaxes the surrounding region. The effect of friction on the delamination evolution is therefore expected not to have a great effect on the overall solution.

5.1.2. Damage Modelling

The damage modelling of transverse matrix cracks and delaminations was based on the work of Hu et al. [55]. The transverse matrix cracks were modelled using the eXtended Finite Element Method (XFEM) and cohesive zone theory was used to model delamination. The damage theory behind implementing both methods is discussed in this section.

Transverse Matrix Cracks

Modelling the transverse matrix cracks via XFEM requires both initiation and propagation criteria. Via the Python script, the options are limited by the default options that come with Abaqus which are discussed hereafter. Two main types of modelling damage with XFEM can be distinguished: based on cohesive zone theory or based on an LEFM approach [132].

The approach based on cohesive zone theory uses traction-separation laws and can be used for both brittle and ductile failure. Cohesive behavior is used to simulate the stresses in the contact when the crack opens, whereas a pressure-overclosure relationship models the behavior in case of crack-closure. To implement this approach, a damage initiation criterion and damage evolution law need to be specified. Depending on what settings are chosen for initiation, the propagation direction of the crack need to be specified as well.

The approach based on LEFM is more appropriate for brittle failure and makes use of VCCT. A fracture criterion based on critical strain energy releaste rates of different failure modes. Additionally, power law parameters need to be specified as well, dependent on the chosen mixed-mode behavior. Using VCCT requires specification of crack locations a priori. When the fracture criterion is satisfied, damage initiates. Due to the fact that crack locations are not known a priori, VCCT surfaces are required throughout the entire model.

Initiation Three stress-based and three strain-based initiation criteria were considered based on what was available in Abaqus and compatible with XFEM: maximum principal stress, maximum principal strain, maximum nominal stress, maximum nominal strain, quadratic nominal stress, and quadratic nominal strain. The effect of compressive normal stresses and strains is not taken into account:

$$\langle \sigma_n \rangle = \begin{cases} \sigma_n & \text{for } \sigma_n > 0 \\ 0 & \text{for } \sigma_n < 0 \end{cases} \tag{5.1}$$

$$\langle \varepsilon_n \rangle = \begin{cases} \varepsilon_n & \text{for } \varepsilon_n > 0 \\ 0 & \text{for } \varepsilon_n < 0 \end{cases} \tag{5.2}$$

For maximum principal stress (MAXPS) and maximum principal strain (MAXPE), damage initiation occurs when the stress or strain value exceeds a critical value, as shown in Equation 5.3 and Equation 5.4 respectively. At initiation, f is $1.0 \leq f \leq 1.0 + f_{tol}$, where f_{tol} is a tolerance value. Furthermore, the crack-plane is set to be perpendicular to the maximum principal stress or strain, which is why this criterion is most suitable when damage is caused by a tensile load. As the crack propagates, this direction can change.

$$f = \frac{\langle \sigma_n \rangle}{\sigma^0_{\max}} \tag{5.3}$$

$$f = \frac{\langle \varepsilon_n \rangle}{\varepsilon^0_{\max}} \tag{5.4}$$

Maximum nominal stress (MAXS) and maximum nominal strain (MAXE) take transverse and shear stresses into account on top of the normal stress. The respective criteria are defined as Equation 5.5 and Equation 5.6. A local material direction can be specified as crack plane normal when one of these initiation criteria is used.

$$f = MAX \left\{ \frac{\langle \sigma_n \rangle}{N_{\max}}, \frac{\sigma_t}{T_{\max}}, \frac{\sigma_s}{S_{\max}} \right\} \tag{5.5}$$

$$f = MAX \left\{ \frac{\langle \varepsilon_n \rangle}{\varepsilon_n^{\max}}, \frac{\varepsilon_t}{\varepsilon_t^{\max}}, \frac{\varepsilon_s}{\varepsilon_s^{\max}} \right\} \tag{5.6}$$

Quadratic nominal stress (QUADS) and quadratic nominal strain (QUADE) are similar to MAXS and MAXE. The main difference lies in the different criteria definitions as shown in Equation 5.7 and Equation 5.8, but apart from that a local crack plane can be selected as well and both normal and shear stresses are taken into account.

$$f = \left(\frac{\langle \sigma_n \rangle}{N_{\max}}\right)^2 + \left(\frac{\sigma_t}{T_{\max}}\right)^2 + \left(\frac{\sigma_s}{S_{\max}}\right)^2 \tag{5.7}$$

$$f = \left(\frac{\langle \varepsilon_n \rangle}{\varepsilon_n^{\max}}\right)^2 + \left(\frac{\varepsilon_s}{\varepsilon_s^{\max}}\right)^2 + \left(\frac{\varepsilon_t}{\varepsilon_t^{\max}}\right)^2 \tag{5.8}$$

Propagation After damage initiation, stiffness is degraded until the point where the material fails, which is also described as softening. The softening behavior can be described by several functions: linear, exponential, or tabular (manual input of points). The functions describe the evolution of the damage parameter, which is 0 for no damage and 1 for complete failure:

The function is based on the after initiation on the traction-separation plot: Figure 5.1 and Figure 5.2 illustrate linear and exponential softening respectively.

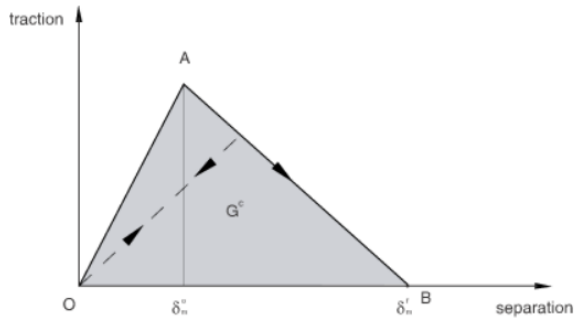

Figure 5.1: Linear softening

Figure 5.2: Exponential softening

Damage evolution models in Abaqus are either energy- or displacement-based. Energy based damage evolution is based on the fracture energy G_f, which is defined as the energy required from the point of damage initiation to the point of failure. Zhang et al. define the fracture energy as:

$$G_f = \frac{1}{2} \cdot \sigma_{eq}^f \cdot \varepsilon_{eq}^f \cdot l_c \tag{5.9}$$

where σ_{eq}^f is the equivalent peak stress and ε_{eq}^f the equivalent failure strain [85]. l_c is the characteristic length of the element that is automatically calculated by Abaqus in order to make damage evolution independent of the mesh size [133]. The fracture energy is similar to, but smaller than the fracture toughness G_c, considering that the fracture toughness takes the energy before damage initiation into account as well.

Displacement-based damage evolution requires the equivalent plastic displacement $\bar{u}_f^{pl}$ as input, which is the remaining displacement after damage initiation up to failure and relates to G_f as follows:

$$\bar{u}_f^{pl} = \frac{2G_f}{\sigma_y} \tag{5.10}$$

where σ_y represents the yield strength.

Stabilization Softening behavior of the element may result in convergence issues. Due to the negative tangent of the matrix after damage initiation, instabilities are introduced. Viscous regularization offers a solution to such problems and is done via the use of a viscous parameter η_{vr}, which works as follows.

Consider the speed of change of the viscous regularized damage variable:

$$\dot{d}_{\text{vr}} = \frac{1}{\eta_{\text{vr}}} (d - d_{\text{vr}}) \tag{5.11}$$

By setting η_{vr}, it can be controlled how fast damage grows. Rewriting Equation 5.11 into a discretized format yields:

$$d_{\text{vr}}^{t+\Delta t} - d_{\text{vr}}^{t} = \frac{\Delta t}{\eta_{\text{vr}}} \left(d^{t+\Delta t} - d_{\text{vr}}^{t} \right) \tag{5.12}$$

Isolating $d_{\text{vr}}^{t+\Delta t}$, the damage variable at the new timestep, the influence of η_{vr} becomes more evident. If the time increment Δt becomes small compared to η_{vr}, the first term on the right-hand side of Equation 5.12 becomes small and the actual damage variable contributes more significantly. This way, the new damage variable is forced to be similar to the previous one, i.e. its evolution is stabilized.

Setting η_{vr} too large results in slow growth of the damage variable and a minimal effect on the overall solution. On the other hand, setting it too small may pollute the overall solution. A practical rule of thumb when assigning η_{vr} is to ensure [92]:

$$\eta_{\text{vr}} \dot{u} < \delta^0 \tag{5.13}$$

where $\dot{u}$ is the nodal velocity, e.g. loading rate, and δ^0 the smallest length parameter in the cohesive law. The rationale behind this relation is that in case the viscous time parameter multiplied with the nodal velocity (resulting in a displacement) is larger than δ^0, overshoot may occur of the peak in the traction-separation law. So by ensuring that Equation 5.13 holds, the traction-separation law is not invalidated.

Generally, it requires trial and error to set this viscous parameter correctly. To verify if viscous effects did or did not pollute the solution, the artificial viscous dissipation energy (ALLVD) must be compared to the internal strain energy (ALLIE). In case ALLVD is too large with respect to ALLIE, the viscous parameter must be tuned. There is no universal definition on what is considered too large, but generally 1-2% is deemed acceptable. In subsection 5.3.3, the ALLVD and ALLIE are compared for the simulations.

Delamination

Modelling delamination with cohesive zone theory can be done via cohesive elements or cohesive contact surfaces. The cohesive surfaces are more simple to implement and have been successfully implemented before with the purpose of modelling delamination [134]. Using a traction-separation law, cohesive behavior is simulated via an interaction property that includes mechanical material properties, damage properties, and contact properties.

The cohesive law requires to specify artificial penalty stiffness K_p values to define the mechanical behavior of the surface bond. A literature review on numerical loading parameters in modelling delamination in various carbon-epoxy laminates is presented by Lu et al. and shows that typical values for K_p are in the range of 10^5-10^6 N/mm^3 [89]. Setting K_p too low results in the predicted delamination being too small or even disappears, whereas setting it too high results can result in a significant increase in computational efforts. Rules of thumb for the penalty stiffness in the different loading directions are [135]:

$$K_{nn}^{\min} = \frac{50E_3}{h}, K_{ss}^{\min} = \frac{50G_{13}}{h}, K_{tt}^{\min} = \frac{50G_{23}}{h} \tag{5.14}$$

Initiation Abaqus offers four damage initiation criteria for surface-based cohesive behavior: maximum traction, quadratic traction, maximum separation, and quadratic separation.

Maximum traction and quadratic traction initiation are the same as MAXS (Equation 5.5) and QUADS (Equation 5.7). Maximum separation and quadratic separation are similar to MAXE and QUADE:

$$f = MAX \left\{ \frac{\langle \delta_n \rangle}{\delta_n^{\max}}, \frac{\delta_t}{\delta_t^{\max}}, \frac{\delta_s}{\delta_s^{\max}} \right\} \tag{5.15}$$

$$f = \left(\frac{\langle \delta_n \rangle}{\delta_n^{\max}}\right)^2 + \left(\frac{\delta_s}{\delta_s^{\max}}\right)^2 + \left(\frac{\delta_t}{\delta_t^{\max}}\right)^2 \tag{5.16}$$

Propagation For cohesive contact surfaces, damage evolution describes the degradation of the stiffness of the cohesive bond, whereas for cohesive elements the material stiffness is degraded. Other than that, damage evolution options are the same, which were mostly described already in the previous section. The main difference lies in the option of modelling mode-mix dependence.

When using energy-based damage evolution, the mode-mix ratio can be defined in tabular form (manually providing datapoints of the desired function), a power law criterion, or a Benzeggagh-Kenane (BK) criterion.

The mixed mode criterion based on a power law in Equation 5.17 is specified based on the fracture toughnesses of fracture modes I (normal), II (in-plane shear), and III (out-of-plane shear). Furthermore, a material-dependent power law exponent α is included to relate the different fracture modes. λ_i represents the fraction of the strain energy of a certain mode of the total strain energy.

$$G_{mc} = \left[\left(\frac{\lambda_I}{G_{Ic}}\right)^{\alpha} + \left(\frac{\lambda_{II}}{G_{IIc}}\right)^{\alpha} + \left(\frac{\lambda_{III}}{G_{IIIc}}\right)^{\alpha}\right]^{-1/\alpha} \tag{5.17}$$

A BK-criterion requires only G_{Ic} and G_{IIc}, because it assumes that we can say that the behavior and fracture toughness of the two shear directions are the same. It means one less parameter is required to set up the mixed-mode criterion. Apart from the two fracture toughnesses, the criterion in Equation 5.18 requires a value for η, the BK exponent, which is material-dependent and obtained via experiments.

$$G_{mc} = G_{Ic} + (G_{IIc} - G_{IC})\left(\frac{G_{II}}{G_I + G_{II}}\right)^{\eta} = G_{Ic} + (G_{IIc} - G_{IC})\, B^{\eta} \tag{5.18}$$

Stabilization Similar to the transverse matrix cracks, the cohesive laws that were implemented to model delamination can show convergence issues that can be (partially) solved by adding artificial viscosity. The working principle here is exactly the same as described in the previous section.

5.1.3. Verification & Validation procedures

Mechanical behavior is verified and validated by comparing the load-displacement curves with both experiments and an analytical solution that is obtained via CLT. The mathematical foundation of the CLT is discussed in Appendix A. To simulate damage using CLT, failure of the matrix degraded inply stiffnesses to 18% of its original value and a second matrix failure or fiber failure degraded the ply properties to zero, which is in line with the sudden degradation model parameters as proposed by Camanho and Matthews [84]. The damage parameters that were used for the FEM model were verified by the range of values was established when looking at comparable materials from literature and validated by comparing results with experiments.

5.2. Finite Element Model

A finite element model was constructed in Abaqus to augment the existing experimental dataset. A specimen length of 100 mm was simulated, corresponding to the area that was monitored during testing. Initial settings of the model are explained in this chapter. Firstly, the specimen's geometrical dimensions are discussed in subsection 5.2.1, followed by the material properties and their distributions in subsection 5.2.2. Damage parameters are explained in subsection 5.2.3, and the mesh, boundary conditions, and step settings are explained in subsection 5.2.4, subsection 5.2.5, and subsection 5.2.6 respectively.

5.2.1. Geometry

The specimen was modelled as three 2D planar shell parts with the same point of view as the schematic drawing in Figure 4.1. The three parts consist of the two longitudinal blocks and the middle transverse block. The parts were connected via a general contact with cohesive properties that are further discussed in subsection 5.2.3. The parts were partitioned such that each portion of the specimen was assigned its own separate element, section, and material. The properties of the elements were calculated based on the distributions that are discussed in the next section. Table 5.1 lists the dimensions of the 2D model.

Table 5.1: Dimensions of the 2D model

Parameter	Value	Unit
Length	100	mm
Ply thickness	0.125	mm
Layup	$[0_2/90_4]_s$	-
Total thickness	1.5	mm

5.2.2. Material Properties

The engineering constants and strength values are varied per element. The goal of this was to model the inhomogeneity that is inherent to composite materials, caused by e.g. defects. This was done by creating a different material and section per element and assigning it to the partitioned elements in the parts. Taking into account spatial correlation between all different mechanical material properties was considered beyond the scope of this thesis, considering the additional required research and modelling of the micro-mechanical scale. Only the correlation between Y_T and G_f is taken into account, because of its dominant effect on the crack pattern characteristics (further explained in subsection 5.2.3). Each property is assigned a distribution, which was assigned as follows.

In literature, a normal distribution was found to be a popular tool with, in general, promising results [104, 105, 108, 111]. Therefore, a normal distribution was adopted here as well. It is characterized by a mean value μ and standard deviation σ (or coefficient of variance (CV), which is defined as CV= $\frac{\sigma}{\mu}$), which indicates the spread of the distribution. It was implemented in Python by using Numpy's *numpy.random.normal()* function. The standard deviations were not experimentally determined and thus may differ from what one would observe during experiments. The mean values are provided by the data sheet [127], except for the transverse strength Y_T, which was experimentally observed.

A Weibull distribution for the transverse strength Y_T of the 90-plies was observed from the experiments performed by Xi Li, which was modelled into Python using the *numpy.random.weibull()* function. The distribution was obtained by mapping the stresses at which cracks initiate. As the cracks propagate instantaneously through the width and through the thickness direction, the Weibull distribution was used to model the strength over the length direction only. To minimize the differences in strength in the through-the-thickness direction, a normal distribution was used with the Weibull strength as mean value with a CV of 0.3%, i.e. Y_T is essentially only varied along the length direction.

Similar to Y_T, the CV of X_T values in through-the-thickness direction was set to 0.3%. Thus, the variation in X_T was essentially modelled as a function of only the length direction as well. Implementing this adjustment in the distribution allowed for smoother convergence at the point of failure of the 0-blocks, without significantly affecting any of the results such as the failure strength.

Tests were conducted where the CVs of all material properties that are listed in Table 5.2 were varied from 0.01% to 20%. To validate the CV setting, the results were compared to the experimental data, with a focus on the crack pattern characteristics, and are presented and discussed in section 5.4. Ultimately, a CV of 2% was chosen and validated with experimental results. Figure 5.3 highlights the effect of modelling the varying properties in the elements. The plot shows the maximum principal stress in the matrix under an arbitrary loading that causes a different stress per element.

Table 5.2: Distributions of the material properties of the specimen (Table 4.1), *: Experimental observations from Li.

Property	Symbol	Distribution	Parameters	Values	Unit
Longitudinal Modulus (tensile)	E_{11}	Normal	μ, σ	142, 2.84	GPa
Transverse Modulus (tensile)	$E_{22} = E_{33}$	Normal	μ, σ	9.1, 0.182	GPa
Transverse shear modulus	$G_{12} = G_{13}$	Normal	μ, σ	5.2, 0.104	GPa
In-plane Poisson's ratio	$\nu_{12} = \nu_{13}$	Normal	μ, σ	0.27, 0.0054	-
Transverse Poisson's ratio	ν_{23}	Normal	μ, σ	0.3, 0.006	-
Longitudinal strength (tensile)	X_T	Normal	μ, σ	2274, 45.48	MPa
Longitudinal strength (compressive)	X_C	Normal	μ, σ	1849, 36.92	MPa
Transverse strength (tensile)	Y_T	Weibull*	λ, k	113.7, 14.85	-
Transverse strength (compressive)	Y_C	Normal	μ, σ	102, 2.04	MPa
In-plane shear strength	$S_{12} = S_{13}$	Normal	μ, σ	104, 2.08	MPa

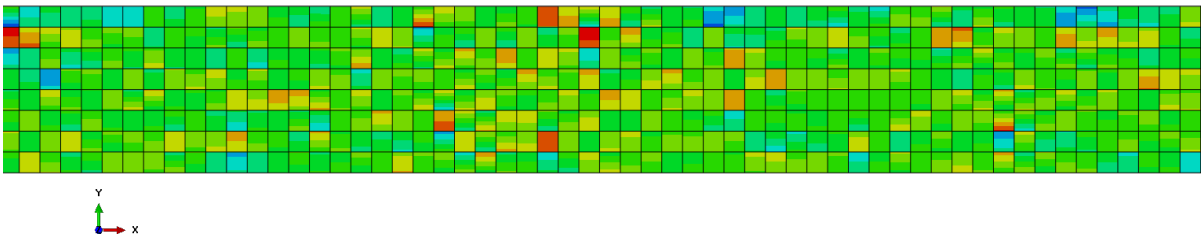

Figure 5.3: Illustration of the effect of the varying properties in the matrix. The plot shows the maximum principal stress within the elements under loading. The colors rank the stress from high to low as: red - yellow - green - blue.

5.2.3. Damage Parameters

Argumentation on the settings of damage parameters that were made is provided, followed by a summary of the final values that were used. Transverse matrix cracks and fiber failure were modelled by using XFEM and delamination was modelled using cohesive surfaces. A summary of the chosen damage parameters is presented in Table 5.5. Verification of the values is done in the process of selecting the values as described in this section by using literature and the data sheet. Validation is done afterward where the results are discussed in section 5.3. Sensitivity analyses of the parameter settings are presented in section 5.4.

Transverse Matrix Cracks For the cross-ply specimen, the load that will cause transverse matrix cracks is close to purely tensile. Not only does that make sense when considering the applied tensile load to the block of 90-plies, but also is that validated by the perpendicular shapes of the cracks (both in width direction and through the thickness direction) in experiments. With the strength values measured as stresses, a MAXPS criterion was considered to be the most logical choice from these six criteria. Adopting a MAXPS criterion has already shown to provide reasonable results in modelling failure by using XFEM [99, 131, 136], despite the fact that shear stresses that are introduced to the 0-plies at the interface with the transverse block are not taken into account. The expectation is that thereby the strength of the longitudinal plies is slightly overestimated.

With the transverse tension being the predominant load on the 90-block, the fracture is assumed to be purely mode I. It should also be noted that the case study subject has quite a thick 90-block, thus the effect of shear loads in the interface regions on the matrix cracks is expected to be even smaller. As such, a mode-independent fracture of the matrix is assumed.

However, as mentioned before, it should be noted that G_{Ic} is in fact the fracture toughness of a material, i.e. the area under the entire traction-separation curve of the entire displacement, *including* the phase before damage initiation. That means using G_{Ic} directly as G_f for the mode I fracture energy of the 90-block theoretically results in an overestimation of G_f. G_{Ic} is therefore interpreted as an upper

limit for the fracture energy, rather than the precise value.

A correlation between G_f and Y_T was included because of the following. When Y_T is lowered but G_f is kept the same, the crack opening displacement at complete failure of the element would increase, which does not make sense from a physical point of view: one would not suddenly expect a rather ductile damage response in case the strength is decreased due to defects such as voids. Not correlating these parameters may result in unrealistic situations that go hand in hand with convergence issues, which has been acknowledged before.

Van der Meer and Dávila [106] assigned random scalar fields to scale both the strength and fracture toughness proportionally, which comes down to linearly scaling the fracture toughness with the strength. Petrov, Gorbatikh, and Lomov correlated the fracture energy with the strength [99] in a quadratic manner, as illustrated in Figure 2.11. Both types of correlation were looked into: linear and quadratic scaling.

Scaling G_f linearly with strength would result in the same displacement for all elements:

$$G_f = \frac{Y_T}{Y_{T,\mu}} \cdot G_{f,\mu}$$

where subscript μ indicates the mean values. Scaling G_f quadratically with the strength would result in a proportional change of the crack opening displacement at failure:

$$G_f = \left(\frac{Y_T}{Y_{T,\mu}}\right)^2 \cdot G_{f,\mu}$$

The quadratic scaling option was considered to make the most sense from a physical point of view. An increase in strength means that the element can cope with a higher amount of stress. However, as mentioned, correlating the engineering constants and strengths is considered beyond the scope of this thesis. Thus, assuming that the modulus is not correlated with the strength, an increase in strength also increases the failure strain and vice versa.

$Y_{T,0}$ was based on the mean value of the Weibull distribution: 110 MPa. Compared to the datasheet, the mean value of the Weibull distributin is 10 MPa higher, which can be explained by in-situ effects caused by the embedding in between the two blocks of 0-plies.

To come up with a reasonable value for the fracture energy, G_f values for tensile matrix failure in other carbon-epoxy specimens were found to provide an indication of a reasonable range of values. Table 5.3 presents the found values for these fracture energies. Based on these findings, the range of values that were tested for $(G_f)_{90}$ was set to 0.15 - 0.35, with steps of 0.1. Ultimately, $(G_f)_{90}$ was set to 0.25 N/mm.

Table 5.3: Fracture energies for tensile matrix failure in carbon-epoxy laminates from literature.

Material	$(G_f)_{mft}$ [N/mm]
IM7/8552 [81, 137]	0.28
AS4/8552 [138]	0.2
T300-1034C [53]	0.13

Fiber Failure In view of the longitudinal plies being aligned with the loading, fiber failure was considered to be the failure mode of the longitudinal plies and thereby the specimen, which is confirmed by experimental observations. Therefore, the MAXPS initiation criterion was chosen for the longitudinal plies as well. Ideally, a more advanced criterion was implemented to model the fiber failure to account for the shear that is introduced in the interface region by the difference in moduli between the 0- and 90-block, for example by Hashin's criteria. However, this is not an option for the type of elements that are used in this 2D approach. Via a subroutine like UDMGINI or UMAT, custom initiation and propagation criteria could be defined in Fortran code, but considering the limited time resources

due to the data-driven model, this was considered beyond the scope of this thesis. The considered alternative criterion was a MAXS criterion, which takes into account the transverse and shear stress as well. Nonetheless, the contribution of transverse and shear stresses is significantly smaller than the axial normal stress when considering the purely axial loading and alignment of the longitudinal plies. Additionally, delaminations that initiate at the transverse matrix crack tips are expected to introduce some relaxation of the shear stress. X_T was assigned as strength value. A correction factor of 1.1 was applied to increase the failure strength of the specimen to the level of the experimental values.

Similar to the transverse block and in line with the assumption that fiber failure is dominated by the axial normal stress, it is assumed that the fracture is purely mode I. The corresponding fracture energy $(G_f)_0$ was found by testing simulations within a range that was based on G_f values for tensile fiber failure of the same carbon-epoxies as mentioned in Table 5.3. These fracture energies are summarised in Table 5.4. The range was set from 10 - 100 N/mm. Within this range, the only parameter that $G_{f,0}$ was found to affect was the failure strength of the specimen. Cracking of the matrix seemed to occur more or less independently from this parameter.

Nonetheless, tuning the fracture energy of the 0-plies turned out to affect convergence, whereas too low values resulted in a single block of 0-plies to fail with scenarios such as depicted in Figure 5.4. Also, the load at which the 0-plies fail was found to be decreased. On the contrary, the higher values showed difficulty in convergence in a different way: simulations would just keep on iterating when damage initiates in one of the elements without successfully continuing damage evolution until failure. Values equal to or lower than 25 N/mm turned out to result in the exact same damage accumulations. In an attempt to strike a balance between a too brittle and abrupt failure and difficulties in convergence, the fracture energy was set to 75 N/mm. To enforce smoother failure, the artificial viscosity coefficient was set relatively high: 10^{-4} 1/s.

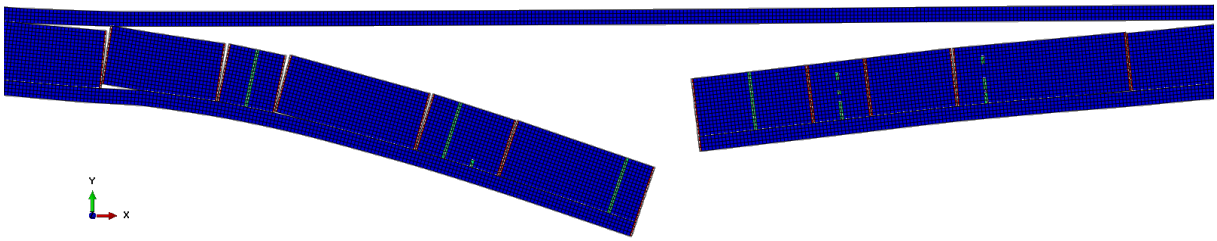

Figure 5.4: Fiber failure in only one of the 0-blocks, resulting in close to 7,000 iterations after failure of the lower 0-block to attempt to break the upper block of 0-plies. Red blocks indicate fully failed elements (STATUSXFEM=1.0) and green blocks indicate elements that are in between damage initiation and complete failure (0 < STATUSXFEM < 1). Maximum displacement is 3 mm.

Table 5.4: Fracture energies for tensile matrix failure in carbon-epoxy laminates from literature.

Material	$(G_f)_{fft}$ [N/mm]
IM7/8552 [81, 137]	81.5
AS4/8552 [138]	92.0
T300-1034C [53]	52.5

Crack Spacing A spacing between the cracks was introduced to alleviate computational efforts and to prevent an infinite number of cracks initiating. A transverse matrix crack that is modeled as a discontinuity with XFEM does not necessarily result in unloading of the surrounding elements. In other words, in case no delamination is present yet, the stress will keep increasing in the region of a "new" matrix crack in every uncracked element around it until the strength is exceeded and those elements fail as well. It can be concluded that not only a crack spacing is required, but also accurate delamination modelling to prevent an infinite number of cracks initiating. A crack spacing parameter was introduced by van der Meer and Sluys because of the same issue [139] and was used afterward by Petrov, Gorbatikh, and Lomov too [99].

Elements were enriched in columns, as shown in Figure 5.5. The reason for the chosen spacing of the columns was based on the smallest spacing that was observed in experiments, which was 0.44 mm. A spacing of 0.5 mm was chosen, where the output format of the crack pattern that serves as input for the neural network was also taken into account: choosing a spacing of 0.5 mm makes discretization of the specimen more convenient and was considered close enough to the observed minimum spacing. Furthermore, it was verified that decreasing the spacing does not significantly affect the number of cracks that appear, but does significantly increase computational efforts, which is in line with findings in literature [99, 139]. The sensitivity of the spacing parameter is further assessed in section 5.4.

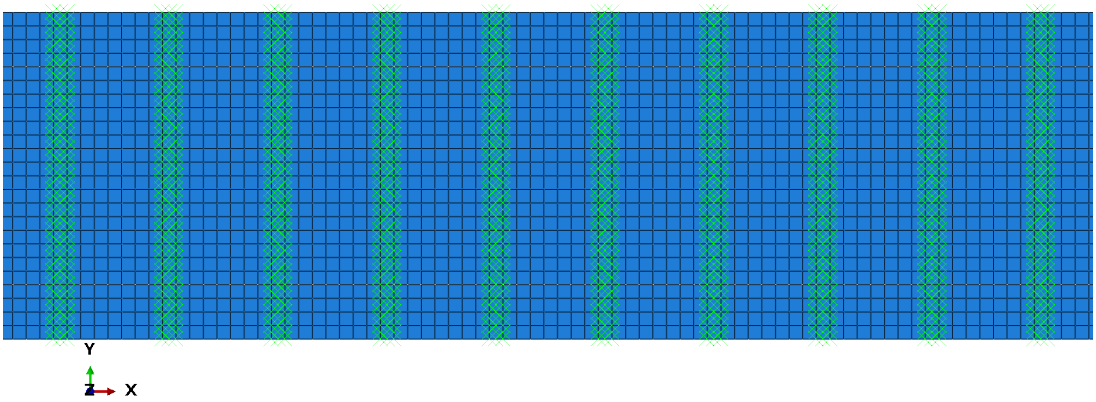

Figure 5.5: Illustration of crack spacing principle in a portion of the specimen: X-direction is along the width, Y-direction is through-the-thickness. Green X patterns mark enriched element columns.

Table 5.5: Damage parameters of the enriched elements.

Parameter	Value	Unit
$(G_f)_0$	75	N/mm
$(\eta_{vr})_0$	10^{-4}	1/s
$(G_f)_{90}$	0.25	N/mm
$(\eta_{vr})_{90}$	10^{-5}	1/s
Minimum crack spacing	0.5	mm

Delamination The additional computational cost of implementing crack locations throughout the model compared to the cohesive based approach resulted in a preference for the cohesive based approach over the LEFM based approach. A cohesive surface was chosen over cohesive elements because of simplicity and reduced computational cost (cohesive elements require "double nodes" at the interface).

Firstly, the cohesive behavior was defined via the uncoupled penalty stiffnesses. For simplicity, the penalty stiffness is assumed to be the same in all three directions. The value for this penalty stiffness was set to 10^5 N/mm^3, which is the minimum value that is recommended to use for carbon-epoxy laminates [89]. Increasing the value increases the number of iterations that is required for convergence, thereby the computational time, without drastic changes in the outcome. In addition, increasing the penalty stiffness postpones delamination and as such matrix cracking, which results in a shift of the majority of the matrix cracks toward a higher load when they occur all at the same time.

Furthermore, a frictionless tangential behavior was assumed for delaminated surfaces. The block of 90-plies was assigned as slave-surface on both ends, because it is the softer material. A simple hard pressure-overclosure contact was defined in case of compressive loads through the thickness.

Delamination Initiation A quadratic traction law was adopted for damage initiation in the cohesive surface contact. Literature presents good results with the criterion [55, 89, 131, 140]. A mixed-mode B-K criterion was adopted, which requires the normal strength (mode I) and shear strength (mode

II = mode III) of the interface. The interface strength value for mode I fracture was based on the transverse strength of the ply from the datasheet: 102 MPa. The mode II strength value was based on the interlaminar short beam shear strength that is provided by the data sheet of the material: 108 MPa. Turon et al. report it is acceptable to reduce the interface strength values in case large cohesive elements are used to promote convergence [135]. As such, the tested ply strengths of the normal/shear strength of the interface were scaled based on the data sheet values: 102/110 MPa, 76.5/81 MPa, and 51/55 MPa.

The lowest pair of interface strengths showed to experience an increased difficulty in converging. This is attributed due to the significant increase of total delamination initiations and thus the total delaminated area, which sparks convergence issues related to contact modelling. The highest pair showed delaminations to occur only around the crack tips, without extending much further to adjacent elements. The middle pair of values, 76.5 MPa for the normal strength and 81 MPa for the shear strength, showed the most reasonable results by using the balance.

The cohesive zone length is an important parameter concerning the minimum mesh size to provide sufficient accuracy. It is a measure of the distance between the crack and the location of maximum traction. Turon et al. state that the importance of having at least 3 to 5 elements in this zone to obtain sufficient accuracy of the cohesive zone modelling [135]. To estimate its length, Equation 5.19 can be used:

$$l_{cz} = ME\frac{G_c}{(\tau_0)^2} \tag{5.19}$$

where M is a parameter that depends on the cohesive model and is typically close or equal to 1. For orthotropic materials with plane strain conditions, the transverse modulus E_2 can be used for E. To calculate the length of the cohesive zone, it is assumed that it is a resin-rich region, meaning data from the resin's datasheet was used: G_{Ic} = 0.432 kJ/m^2 and τ_0 = 102 MPa. It results in a a cohesive zone length of 0.378 mm. Having at least 3 to 5 elements in this zone means a minimum element size of 0.0756 - 0.126 mm. It should be noted here that the upper limit of an allowable size exceeds the largest element size that was considered: one square element through the thickness of a single ply of 0.125 mm.

Delamination Evolution A study focused on cohesive zone modeling in FE performed by Alfano [141] showed that exponential softening behavior was optimal in terms of accuracy, whereas a bilinear law would be the best compromise in terms of accuracy versus computational cost. Keeping the computational cost in mind due to the high number of degrees of freedom, the bilinear setting was chosen.

A B-K criterion was adopted to model the evolution of delamination. No experimental values were obtained for the material that was used in the experiments. Therefore, small ranges of values were found based on the used material and literature.

Regarding the fracture energies, mode I was based on the found value for matrix cracking and the data sheet of the matrix and mode II was based on literature. The values that were tested for mode I are 0.25 N/mm and 0.4 N/mm and for mode II 0.6 N/mm and 1.0 N/mm. The total number of combinations that was tested is 12, including the various strength parameters, from which the final values are shown in Table 5.6. A more elaborate discussion on these results is presented in section 5.3. A viscous parameter was calculated by using Equation 5.13, to be approximately $6 \cdot 10^{-5}$. Because the matrix is tough and not ductile, a B-K exponent of 2 was assumed.

It was found that the mode I fracture hardly affects the results, which makes sense when looking at the load case: the longitudinal blocks and matrix block are shearing over each other. Thus, mode I effects seem negligible. The mode I fracture energy was set equal to the fracture energy of the matrix.

The lower fracture energy of 0.6 N/mm showed a smoother evolution of the crack density curve, compared to a rather steep curve caused by simultaneous cracking at different locations. Delamination progresses more easily at lower values for fracture energy and as such, relaxation in the stress field

around matrix cracks is facilitated.

Table 5.6: Cohesive interface parameters

Parameter	Value	Unit
K_p	10^5	N/mm^3
τ_I	76.5	MPa
τ_{II}	81	MPa
τ_{III}	81	MPa
$(G_f)_I$)	0.25	N/mm
$(G_f)_{II}$)	0.6	N/mm
B-K exponent	2	-
$(\eta_{vr})_{coh}$	10^{-5}	1/s

5.2.4. Mesh

At the time of this project, XFEM's compatibility in Abaqus is limited to linear elements. Hence, linear CPE4I plane strain elements were used on all elements. The setting of incompatible modes was toggled on to account for potential issues in case of local bending of the specimen at the point of failure of the 0-blocks. The shape and size of the mesh was configured during the partitioning of the parts into squares. Because cracks can initiate at essentially any location, the same accuracy is required everywhere, so the same element size was set uniform over the entire model.

The largest element size that was taken into account corresponds with one single square element through the thickness of a single ply, i.e. 0.125 mm. Two more element sizes were considered as well, corresponding with two and three elements through the thickness of a single ply: 0.0625 mm and 0.03125 mm respectively. The result of a mesh convergence study is presented in section 5.4.

5.2.5. Boundary Conditions

The clamps that were placed on the specimen during the tests were modelled as a constraint in movement in through-the-thickness direction ($U2 = 0$) and a constraint in rotation ($UR3 = 0$). The left side of the specimen was constrained to move in the loading direction as well ($U1 = 0$), i.e. was fully clamped, as shown in Figure 5.6. Constraining of $U2$ and $UR3$ showed to improve stability the system, which was particularly necessary when modeling the full length of the specimen. Boundary conditions were applied per edge of the element as a result of keeping the element size and layup variable in the script. The boundary conditions at the loaded edge are shown in Figure 5.7.

However, constraining through-the-thickness displacement at the clamped and loaded edges introduces additional stresses. As a result, despite the variation in strength per simulation, the specimen consistently failed prematurely near the edges. Therefore, a "crack-free" region of 2.5 mm near the edge was introduced, meaning that elements near the edge were not enriched and could not fail. The value of 2.5 mm was based on observations of the stress field near the boundary condition, similar to the stress field observations that are discussed in the mesh convergence study in subsection 5.4.1.

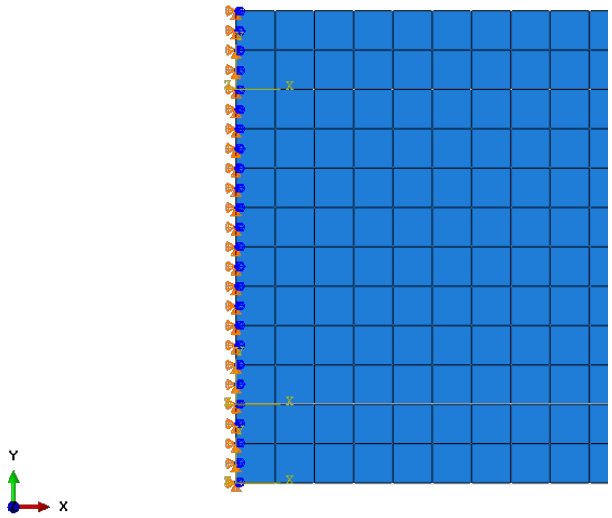

Figure 5.6: Boundary conditions at the clamped edge: $U1 = U2 = UR3 = 0$. The coordinate systems that coincide with the clamped edge are those of the three different parts: the two longitudinal blocks and the transverse block.

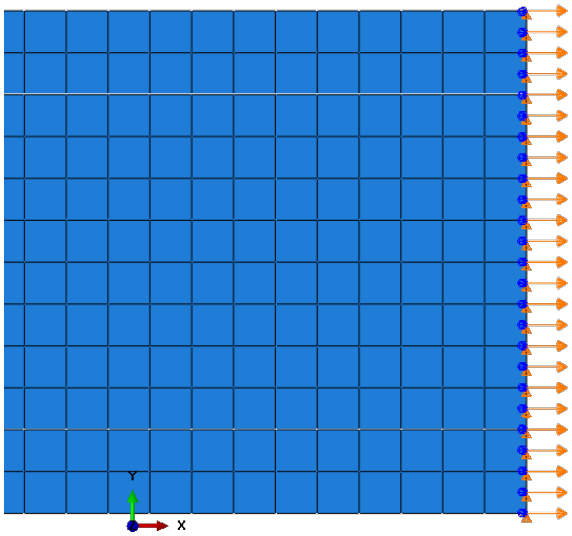

Figure 5.7: Boundary conditions at the loaded edge: $U2 = UR3 = 0$, $U1 = \Delta U$.

Loading

Loading was applied as a displacement boundary condition, where corresponding reaction forces are a result of the specimen's stiffness. These reaction forces were converted to a stress to validate results with experimental data. Displacement controlled loading was in this case preferred over applying a force controlled loading because of stability: force controlled loading turned out to have great difficulties to converge because of the many cracks that cause disruptions on the load-displacement curve. On the contrary, displacement controlled loading showed small, vertical drops on the load-displacement curve when cracks appear, that are handled more adequately by Abaqus. The failure strength of a specimen was obtained by taking the peak stress in the stress-strain curve, the corresponding strain was taken as failure strain.

To speed up the runtime, the load was applied by using a tabular amplitude loading, as illustrated in Figure 5.8. It can be estimated beforehand what portion of the load can safely be applied without sudden simultaneous failure at several locations in the matrix. This fraction of the load is indicated

in the figure as f_1 and is applied in the time t_1. Accordingly, the loading rate of this first load portion can be increased: per time increment, a larger displacement is applied which decreases the runtime. Once the displacement is getting close to a level where matrix cracks are expected, the load rate is again decreased to obtain sufficient accuracy.

Depending on the desired loading rates in the two stages, t_1 can be adjusted. t_1 was made into a function of ε_0, ε_t, and a load rate factor (LRF), as shown in Equation 5.20. ε_0 is the strain at which the specimen is still undamaged, ε_t the total applied strain, and the LRF is the ratio between the load rates of the first loading block ($0 < \varepsilon \leq \varepsilon_0$) and the second loading block ($\varepsilon_0 < \varepsilon \leq \varepsilon_t$).

$$t_1 = \frac{1}{\text{LRF} \cdot \frac{\varepsilon_t - \varepsilon_0}{\varepsilon_0} + 1} \tag{5.20}$$

Table 5.7 shows the parameter settings that were used to come up with the chosen value of t_1. ε_0 and ε_t were observed from several simulations and LRF was arbitrarily set.

Table 5.7: Parameter settings to calculate t_1 by using Equation 5.20.

Parameter	Value	Unit
ε_0	0.007	-
ε_t	0.015	-
LRF	10	-
f_1	0.467	-
t_1	0.08	sec

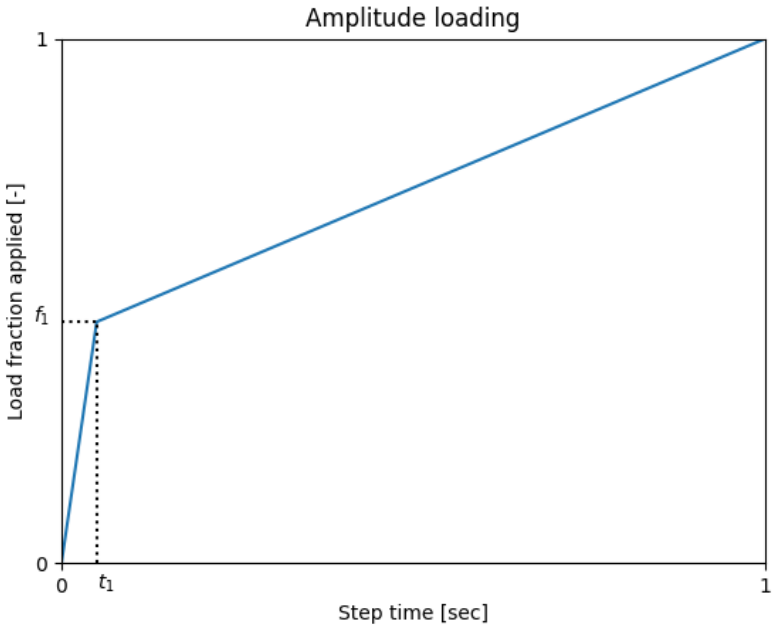

Figure 5.8: Amplitude that was applied to the displacement controlled loading. f_1 corresponds with the fraction of the total applied load until which no failure is expected. t_1 is the corresponding step time.

5.2.6. Step

A dynamic implicit quasi-static step was implemented to obtain convergence of this relatively unstable problem. Due to the highly non-linear nature of the problem, time incrementation settings were changed to the values that are shown in Table 5.8. An explanation is provided on why the values were changed.

Δt_{init} is the setting for the first time increment, which, if set too large can cause convergence issues.

Δt_{min} determines the lowest increment for which it is tried to obtain convergence. Setting this value too high could result in a prematurely stopped simulation, whereas setting it too low could result in endless cutbacks and prevent the simulation from stopping once it gets stuck.

The maximum time increment during a step Δt_{max} is an important parameter that needs to be balanced in between accuracy of the solution and number of iterations to convergence, i.e. computational efforts. Setting Δt_{max} too low results in a significant increase in runtime, whereas setting it too high results in simultaneous failure of the matrix at multiple different locations. The latter results in a polluted solution. To determine Δt_{max}, the following relation was derived:

$$\Delta t_{max} = \frac{\Delta \delta_{crit}}{\text{LR}} \tag{5.21}$$

where $\Delta \delta_{crit}$ is the smallest increment in applied displacement that causes damage initiation in the first element, defined as $\varepsilon_{crit} \cdot l_0$, where l_0 is the characteristic length or in case of square elements: element length. Dependent on the mesh size, $\Delta \delta_{crit}$ is recalculated based on ε_{crit}. LR is the applied loading rate in the second loading block. With none of the parameters (apart from the artificial viscosity coefficients) being dependent on time, it was argued that increasing the runtime or decreasing the value for Δt_{max} would make little difference. The latter was option was picked. A value for δ_{crit} was derived by observing the moment of first element failure for several simulations. The loading rate was calculated based on the settings for t_1 and does not change much when LRF is varied. With settings for t_1 as in Table 5.7, a specimen length of 100 mm and an element size of 0.0625 mm, Δt_{max} becomes 0.00115 sec.

I_0, I_R, I_P, I_C, and I_L are parameters that involve the number of consecutive equilibrium equations before certain checks are performed. Due to the highly non-linear nature of the problem, keeping these settings to default may cause convergence issues. Thus, increasing these values prevents the simulation from non-convergence and premature aborting of the job.

Table 5.8: Time incrementation settings.

Parameter	Default	Setting	Unit
Δt_{init}	0.01	0.001	sec
Δt_{min}	0.001	1e-12	sec
Δt_{max}	0.01	0.00115	sec
I_0	4	40	-
I_R	8	80	-
I_P	9	90	-
I_C	16	160	-
I_L	10	100	-

5.3. Results and Discussion

The results of the analyses are discussed in this section. Post-processing of the results was done in the Python environment as well. Several text files were generated based on the output data base generated by Abaqus, from which the most important information was extracted. 277 patterns were simulated. The results are compared to the experimental results and an analytical solution based on CLT. The comparison forms the basis of the discussion and conclusion on the validity of the model and corresponding recommendations.

Firstly, load-displacement curves are presented and compared to experiments and literature in subsection 5.3.1. Next, the appearance of matrix cracks and delamination is discussed in subsection 5.3.2. In subsection 5.3.3, using artificial viscosity is justified and limitations of the model are discussed in subsection 5.3.4.

5.3.1. Mechanical Behavior

The mechanical behavior of the specimen was compared to the experiment in which a displacement based load was applied with a loading rate of 1 mm/min [31] and a CLT based solution.

The experimental data shows to 'jump around' a straight line, with the jumps becoming larger at a later stage. This is explained by the fact that the experiment was displacement-controlled: as displacement is further increased, instabilities are introduced by the fact that the required resulting reaction force to displace the specimen further drops for a moment at the point of cracking.

The Young's modulus of the pristine specimen that was simulated via CLT was 53.65 GPa. Failure of the transverse block occurs at a stress of 649 MPa, or a strain of 0.0121, after which the modulus is (instantaneously) degraded to 48.65 GPa. The CLT specimen fails at 775 MPa.

The stress-strain curves of the three methods are shown in Figure 5.9. 10 FEM curves were included to provide an indication of the variability within the simulations. The curves are dotted to prevent complete cluttering of the graph. It is evident that the analytical CLT method overestimates the failure strength the most significantly, which makes sense when considering all of the assumptions that are made and how damage progression is simulated: without the presence of delamination and without taking into account any shear stresses that are introduced near the interface region.

Some FEM simulations slightly overestimate the strength, which can be partially attributed to the maximum principal stress damage initiation criterion that was used to model fiber failure. After all, the strength of the fibers in the longitudinal blocks determines the global strength of the specimen. Furthermore, it can be seen that the non-linearity in the stress-strain curve is different between the FEM model and experiments: the FEM specimen fails at a higher stress but lower strain. Not only is this due to the initial difference in stiffness of the specimen as a whole, but also can it be attributed to the 'simple' bilinear softening law. An exponential softening law might provide more accuracy.

The plot also shows that in some cases FEM simulations underestimate the strength and failure strain, which could be the result of material properties having arranged itself in an unfavorable manner. Figure 5.10 zooms in on the region where the specimens fail in Figure 5.9. It can be seen that the modulus of the FEM simulations lies in between the CLT specimen and the experiment. This implies that the coefficient of variance could have been increased, which as discussed in subsection 5.2.2, degrades the modulus. Furthermore, the failure points of the specimen are quite similar to the experimental specimen, although the failure strength is consistently higher, presumably caused by the higher moduli.

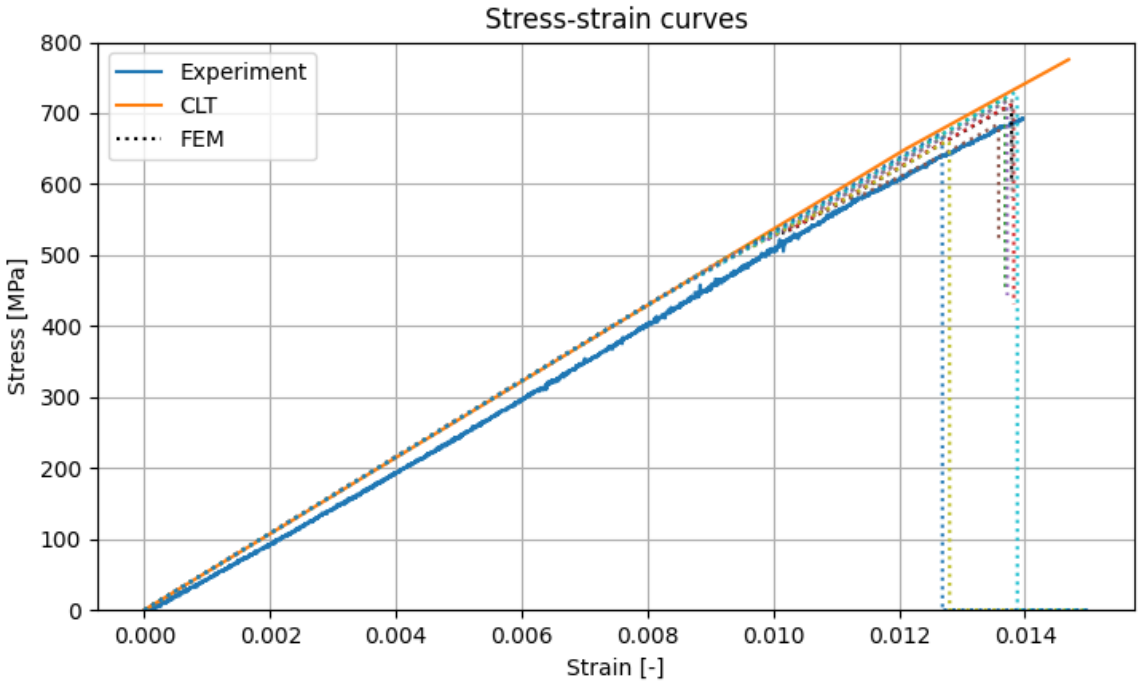

Figure 5.9: Stress-strain curves of a displacement-based experiment [31], analytical solution, and 10 FEM dotted FEM curves indicating the variability.

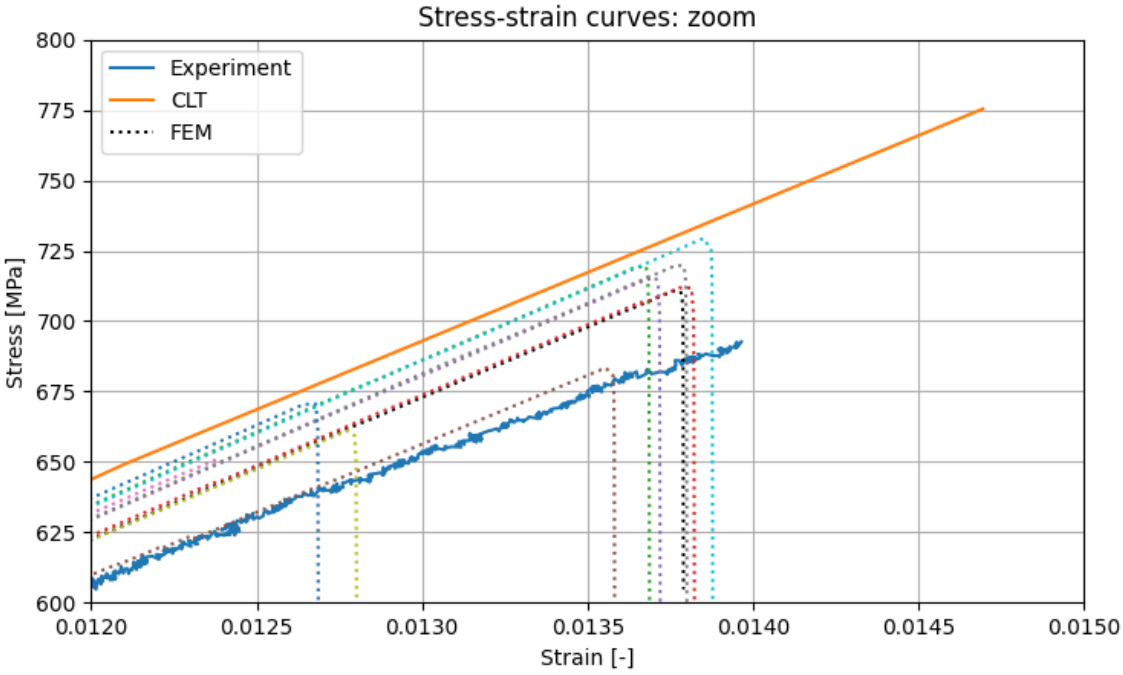

Figure 5.10: Zoom of Figure 5.9 at the region where the specimens fail.

5.3.2. Damage Results

It was found that the majority of the matrix cracks initiated at elements located at the interface. This is explained by the fact that the elements near the interface are constrained via the cohesive bond to the longitudinal plies, which causes a local increase in the observed principal stresses. Time between the moment of initiation and propagation differs per crack: it was both observed that some elements failed almost immediately after damage initiated, whereas other elements were damaged (i.e. STATUSXFEM=0.4) more gradually towards failure when the load was increased.

The curved cracks that were mentioned earlier were also encountered in the results, although rarely. An example is illustrated in Figure 5.11. Images of the sequential progression of the cracks of this portion of the specimen are presented in Appendix B.

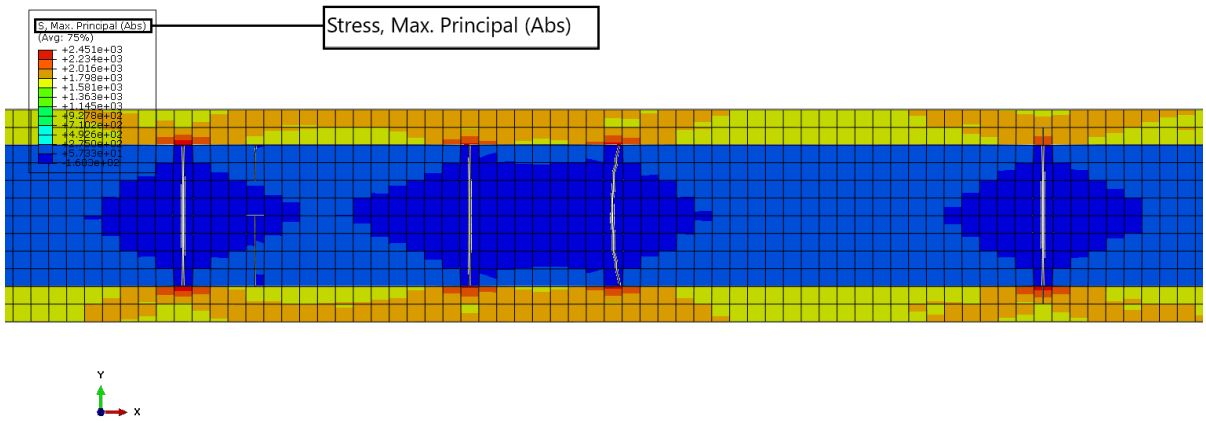

Figure 5.11: Maximum principal stress field around a curved crack that was found in the results of the simulations. The red color indicates a stress of 245 MPa, whereas the dark blue indicates a compressive stress: -16 MPa.

Remarkably, the location of the first crack showed not to occur at the exact same place as where the lowest strength values of the elements were placed. This is explained by the difference in mechanical properties per element: first failure occurs at the element which firstly experiences a stress that exceeds a strength value. In case the moduli of elements that are assigned the lowest strength are lower as well, this means the stress in the element is also lowered and another element could fail first. In reality, one would expect the first crack to show at the same place as the lowest strength. The reason that the reality differs from the FEM simulation here, is the lack of correlation between the different material properties. Thus, it is expected that introducing correlation between the material properties will fix this phenomenon.

Within the Python environment, crack patterns were plotted based on the output files of the Abaqus simulations. Examples of such plots are shown in Figure 5.12 - Figure 5.14. The red dots indicate failure of enriched elements and the green diamond shapes indicate delamination (CSDMG > 0). It should be noted that the aspect ratio of these plots is drastically different than the actual aspect ratio of the cross-ply specimen; the current aspect ratio was purely chosen to clarify the content.

Figure 5.12 depicts a crack pattern with similar characteristics as the experiments and convergence of fiber failure. Figure 5.13 shows a pattern of a simulation that stopped convergence because of difficulties in modeling fiber failure. Figure 5.14 shows a crack pattern that results from initial struggles of the model to converge fiber failure, during which significant delamination is caused. The corresponding Abaqus plot before final failure of the fiber is shown in Figure 5.18. Fiber failure can easily be spotted from the crack pattern graphs: elements with a y-coordinate smaller than 0 mm or larger than 1.0 mm correspond to elements within the longitudinal plies.

Failure of the longitudinal plies occurred at locations were a matrix crack already existed, which is in line with findings during the experiments. Nevertheless, damage initiation within the plies regularly showed to cause convergence issues, after which simulations were aborted before complete failure occurred.

Comparing Figure 5.12 - Figure 5.14 to the experimentally observed crack patterns that are depicted in Figure 5.15 - Figure 5.17 show similarities: the range of total number of cracks is similar, as well as the fact that this range can be rather wide. The experimental specimens range from 12 transverse matrix cracks to 34. The FEM simulations show a similar, with 13 matrix cracks in Figure 5.14 and the highest number of cracks being 39. The precise lowest number of cracks for fully-run simulations is difficult to obtain, because of the simulations that encounter convergence issues, as is further discussed in Figure 5.18. Another similarity is the non-homogeneous crack spacing, with sometimes portions of

5-25 mm of a pristine matrix combined alternated clusters of cracks at failure.

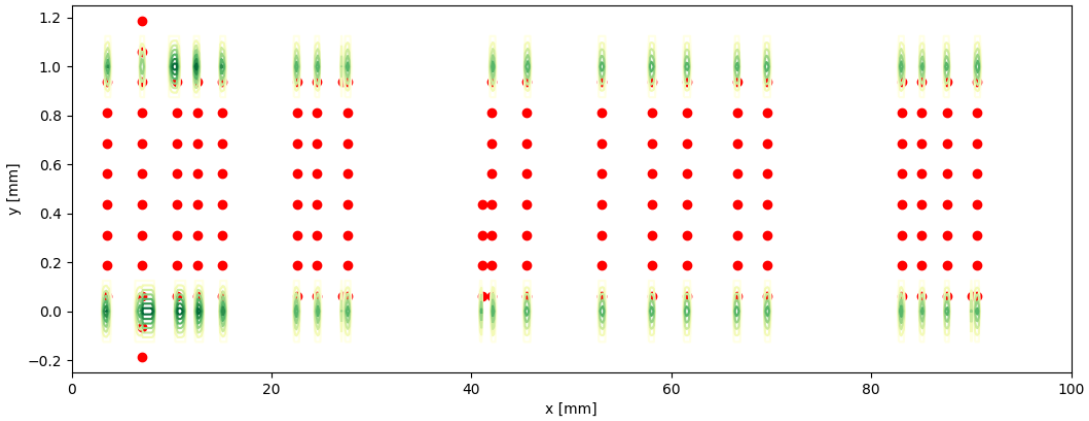

Figure 5.12: Crack pattern of a specimen that converges.

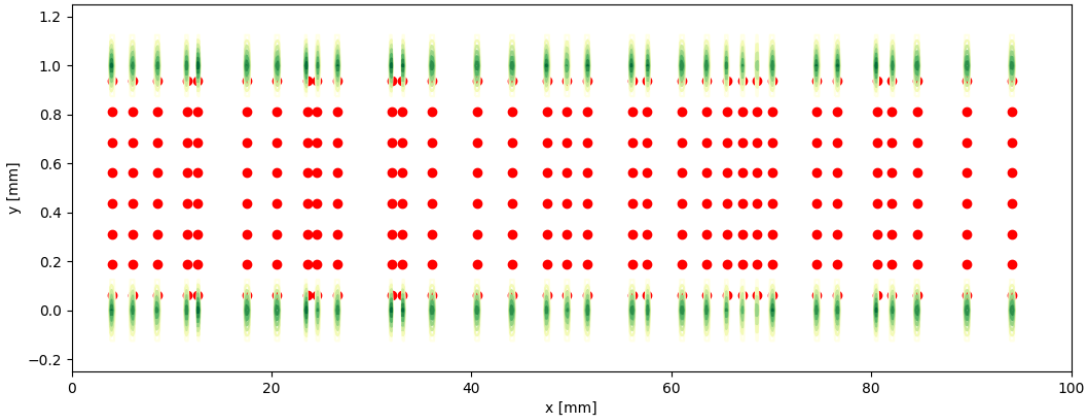

Figure 5.13: Crack pattern of a specimen that does not converge beyond fiber failure.

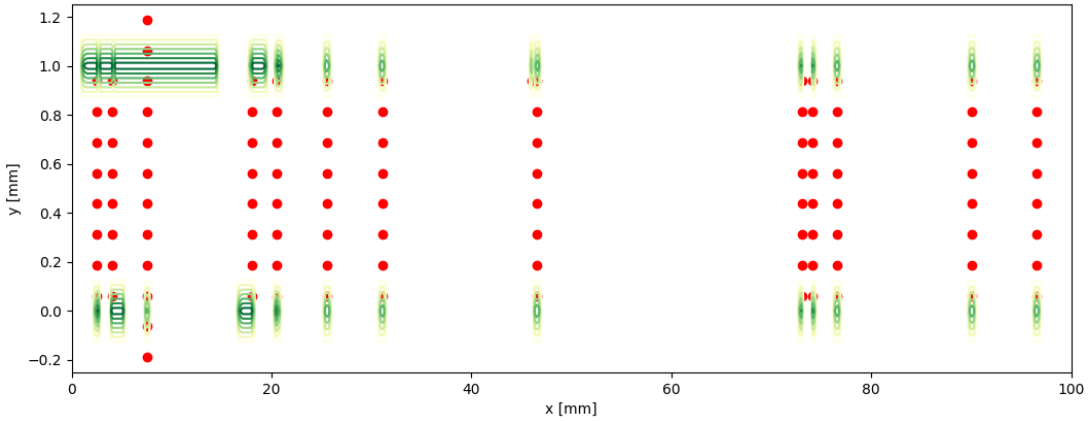

Figure 5.14: Crack pattern of Figure 5.18, a specimen that converges after many iterations.

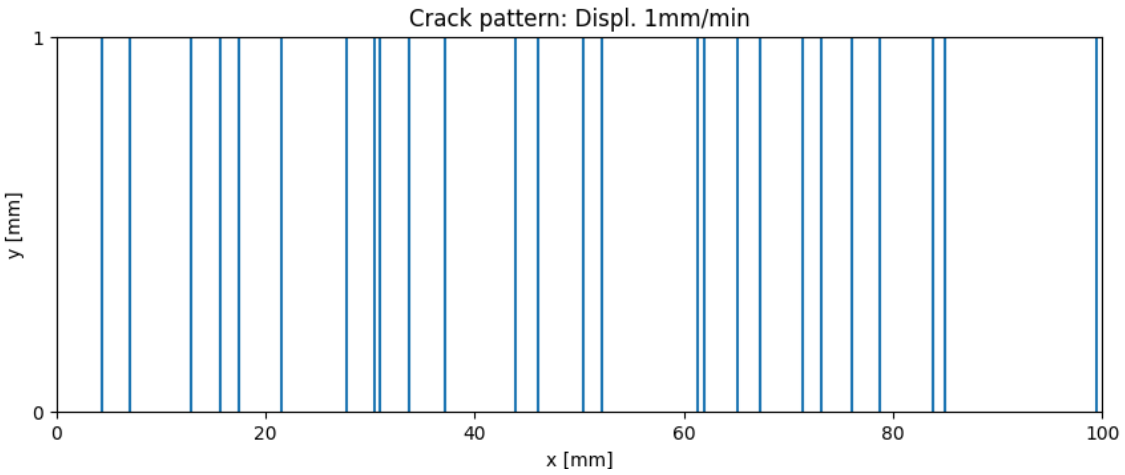

Figure 5.15: Crack pattern of a displacement-controlled experiment where the applied displacement rate was 1 mm/min.

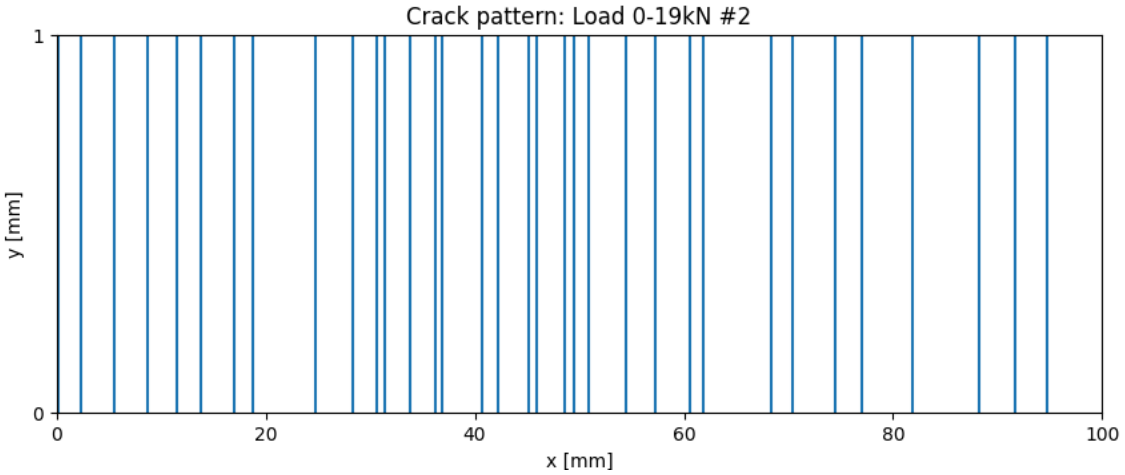

Figure 5.16: Crack pattern of a load-controlled experiment where the load was gradually increased from 0 to 19 kN.

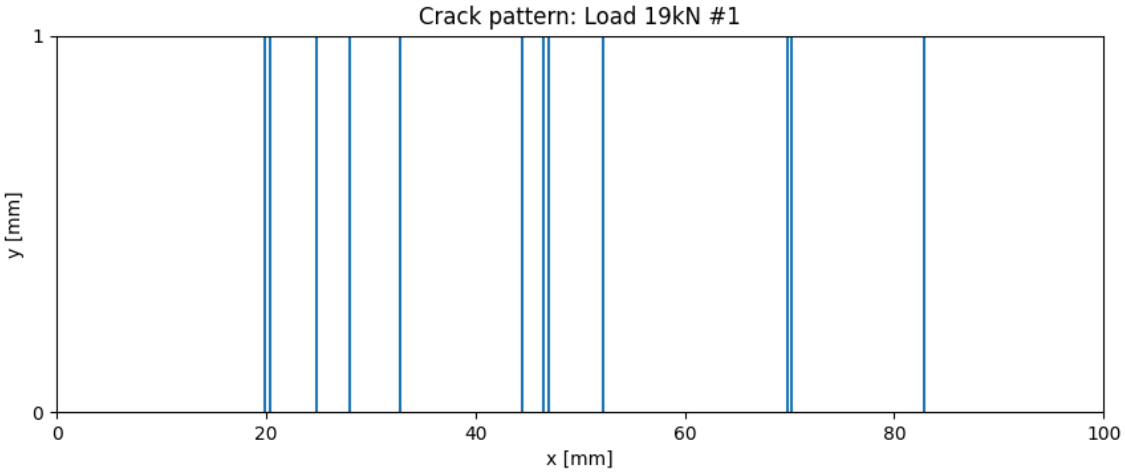

Figure 5.17: Crack pattern of a load-controlled experiment where the applied load was 19 kN.

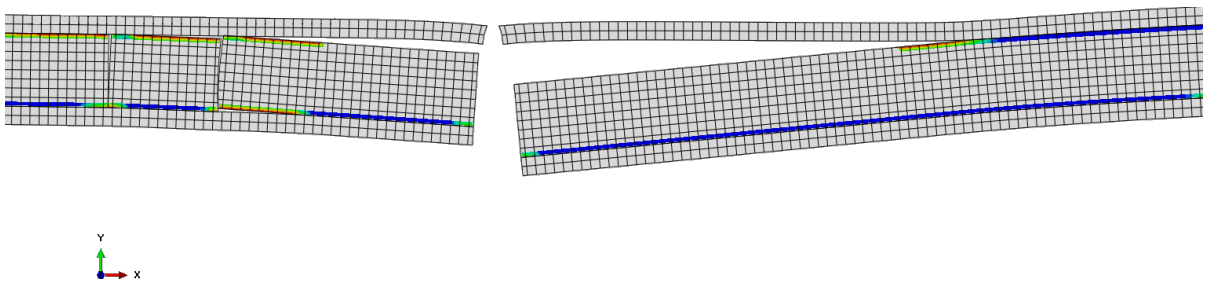

Figure 5.18: Stiffness degradation plot of the cohesive surface (CSDMG) in a simulation that encountered difficulties in modelling fiber failure, resulting in locally large delaminations.

5.3.3. Effect of Artifical Viscosity Coefficient Settings

Figure 5.19 shows the evolution of the internal strain energy (ALLIE) and viscous dissipation energy (ALLVD) in a representative specimen. To verify that the chosen settings for the artificial viscosity parameters in damage stabilization did not pollute a solution, ALLVD needs to remain small. In the example below, the maximum value of ALLVD expressed in % of ALLIE is 1.5%. This maximum value of 1.5% was found not to vary significantly between various simulations. As such, it was concluded that the artificial viscosity parameter settings did not pollute the solutions. The steeper part of the curve early on in the loading is attributed to the amplitude loading.

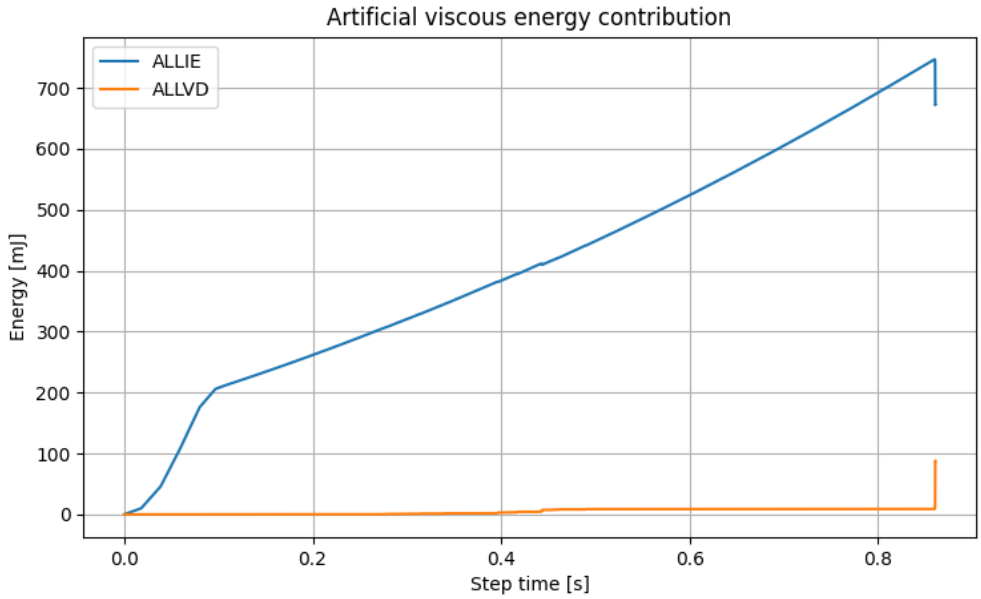

Figure 5.19: Viscous dissipation energy (ALLVD) compared to the internal strain energy (ALLIE) during a simulation. Right before the vertical jumps near the end of the graphs, the specimen fails. Before the specimen fails, ALLVD is never above 1.5% of ALLIE.

5.3.4. Convergence Issues — Limitations of the Model

The number of maximum iterations was limited based on simulations that ran smoothly. Convergence issues resulted in not every simulation running up until fiber failure. Requiring significantly more iterations than that number of iterations was considered to be a waste of the available resources, because the additional knowledge of the exact point of fiber failure was deemed less important than generating more matrix crack patterns, albeit focusing on early damage development. Several simulations were prematurely aborted, caused by one of or a combination of the phenomena as described below.

- **Non-failing 0-plies**: quite often, it was found that damage initiated at the 0-plies, but struggled to evolve from the point of damage initiation to failure. This could be attributed to the chosen

setting of the fracture energy. An attempt to solve this problem was made by setting the artificial viscosity in the 0-plies to a relatively high value (10^{-4}), but the problem kept occurring, causing thousands of additional required iterations only to break the fibers. Therefore, a loss was accepted in terms of accuracy: not every simulation that was used to provide data for the database included all matrix cracks up until failure. It is hypothesized that the accumulation before the point of fiber failure is independent of this convergence error.

- **Limitation due to the enriched columns**: Some of the cracks initiate within an element along a curved path. However, the elements are enriched in columns. Therefore, a crack sometimes hits the "sides" of its column and cannot recover its path from that, causing a loop of iterations without the possibility of the crack to evolve following its initiated path. However, this phenomenon was observed rarely.
- **Asymmetric failure of 0-plies**: as shown in Figure 5.4, it occurred that only one of the 0-plies failed and the other one would not break, resulting in a significant increase in required iterations. Nonetheless, it should be noted that in reality, in case one of the longitudinal plies breaks, the other one breaks instantaneously as well. Therefore, the damage accumulation pattern obtained after asymmetric failure was deemed to be still useful.

5.4. Sensitivity Analyses

This section presents the sensitivity analysis of four model parameters: mesh size in subsection 5.4.1, the CV of the material properties that were modelled via a normal distribution in subsection 5.4.2, the Y^T distribution in subsection 5.4.3, and the spacing between enriched columns in subsection 5.4.4.

5.4.1. Mesh Convergence study

Three different meshes were tested: 1, 2, and 3 elements through the thickness of a single ply. The corresponding DOF and runtime are presented in Table 5.9.

Table 5.9: Mesh convergence study parameters.

Mesh	Element size [mm]	Number of nodes [-]	Number of elements [-]	Total DOF [-]
1	0.125	66,155	16,000	107,630
2	0.0625	207,967	51,200	347,374
3	0.03125	417,837	105,600	705,726

Crack initiation and propagation behavior seemed relatively independent of the mesh. The stress field around a crack was further analysed to assess the effect of the mesh size. Specifically, it was determined to what range around a crack the stress field was affected by its presence. Furthermore, the shape of the stress field was taken into account.

Figure 5.20 - Figure 5.22 depict the stress fields around a matrix crack for the different mesh sizes. The limit values of the color plots were set equally for all plots and were based on the stresses that were observed with the finest mesh. Naturally, a finer mesh results in a smoother stress fields. Nonetheless, the 'coarse' mesh of elements with size 0.125 mm shows to be able to capture the overall shape of the stress fields. Furthermore, it was found that the size of the stress field region around a crack that is affected by the presence of a crack varies in the order of 0.01 mm, i.e. not significantly: up to about 1.2 mm on both sides of the crack, the stress field is differs from the observed stresses in uncracked regions.

For the layup that was used in the experiments, using the coarsest mesh was deemed acceptable. Because of the thick block of transverse plies, 8 elements are present through the thickness of this transverse block and 12 elements are present in total throughout the thickness. Having 8 elements through the thickness of the transverse block allows sufficiently accurate stress (re)distributions to model the transverse matrix cracks.

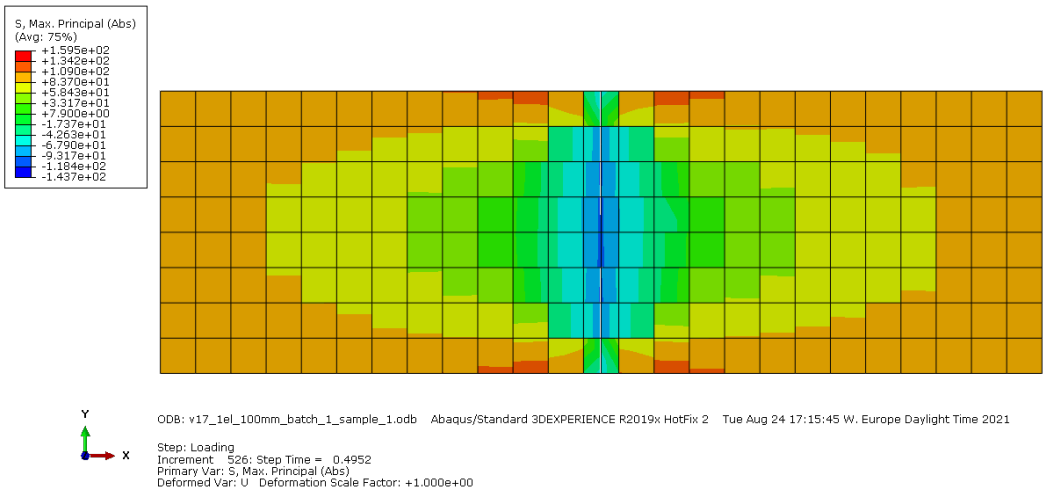

Figure 5.20: Max. principal stress field around a matrix crack. Coarse mesh — 1 element though the thickness of a ply — 0.125 mm. Total width displayed: 3.125 mm.

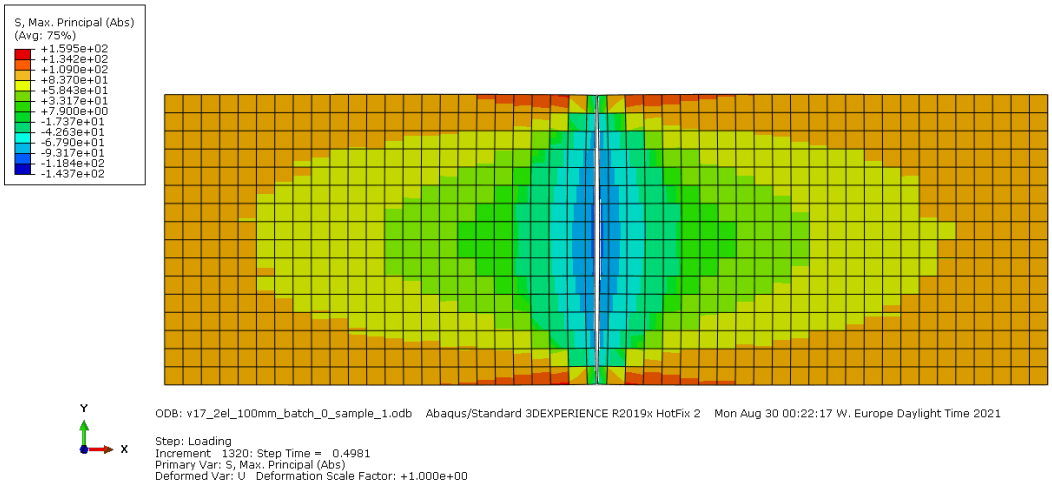

Figure 5.21: Max. principal stress field around a matrix crack. Medium mesh — 2 elements though the thickness of a ply — 0.06125 mm. Total width displayed: 3.175 mm.

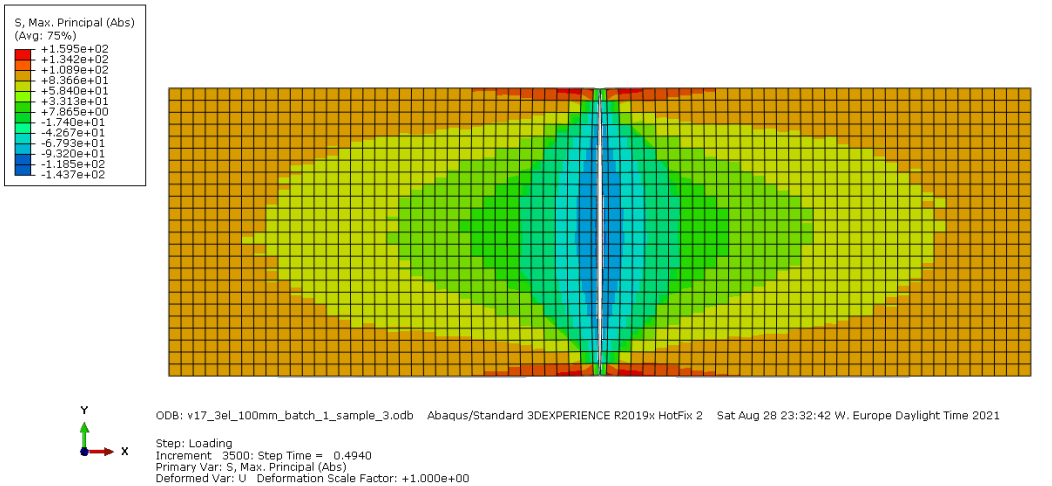

Figure 5.22: Max. principal stress field around a matrix crack. Fine mesh — 3 elements though the thickness of a ply — 0.03 mm. Total width displayed: 3.0 mm.

5.4.2. CV of Material Properties

The CV's of the engineering constants and strengths (except for X_T and Y_T) are varied from 0.01% up to 20%. As shown in Table 5.10, the failure strength decreases overall when the coefficient of variance of the engineering constants and strength values increases. These findings are in line with a recent study that investigates the effect of including stochasticity into FEM [111]. The undamaged modulus was calculated based on the global load-displacement curve up to before damage initiation at a total applied strain of 0.005, corresponding to an applied stress of approximately 200 MPa. The overall trend is that the global modulus decreases with an increasing CV, which too is in line with the findings of [111].

Even though the mean value of the stiffness properties remains the same when increasing the CV, it makes sense that the strength and stiffness are affected when the CV is decreased. The differences in properties between adjacent elements increases and causes local peaks in stress, which explains the decrease in observed failure strength. Furthermore, it is hypothesized that due to the growing local differences in case of increasing CVs, the load transfer between elements is affected and causes the global stiffness to decrease. Nonetheless, the decrease in global stiffness is significantly less affected than the failure strength. This can be attributed to the strength being dependent on only the lowest values within a drawn set of samples from a distribution, whereas low stiffness elements are compensated for by an approximate equal amount of high stiffness values.

Looking more closely at the individual results, it is observed that for an increase in CV, the damage accumulation process up to failure becomes more abrupt. For lower values of CV, several transverse matrix cracks initiate from which delaminations starts, until at some point the longitudinal blocks fail at a location of an existing transverse matrix crack. At higher values of CV, only one or a few matrix cracks initiate, after which the longitudinal blocks fail at a "random" location that does not extend from a matrix crack. This phenomenon makes sense from the point of view that e.g. a CV of 15% for the longitudinal strength X^T drastically reduces the local strength, such that the fibers fail already before the matrix has failed. Moreover, considering that no correlation was adopted between the engineering constants and the strengths, a reduction in strength could be combined with a higher longitudinal modulus within the same element. Such a scenario is deemed unphysical and would result in an even more drastic reduction in strength, as such facilitating the observed phenomena.

Table 5.10: Sensitivity study of the coefficient of variance that is applied to all engineering constants and strength values, except for Y^T. *: convergence issues encountered.

CV [%]	Failure strength [MPa]	Difference [%]	Modulus (undamaged) [GPa]	Difference [%]
0.01*	720	0.075	53.85	3.37
1	739	5.73	53.85	3.36
2*	657	-6.0	53.81	3.28
3*	682	-4.7	53.76	3.19
5	696	-0.47	53.72	3.11
10	602	-17.0	53.21	2.14
15	497	-28.9	52.68	1.11
20	518	-29.2	52.31	0.4

5.4.3. Transverse Strength distribution

The transverse strength distribution dominantly affects the progression of matrix cracks. A new set of properties was drawn from the experimental Weibull distribution for each simulation. The lowest strength values logically affect the load at which matrix cracks start to appear: the lower the strength values, the earlier the cracks showed and vice versa. Therefore, it was found that for a part of the specimens, cracks started initiating earlier, but also stopped to initiate earlier. In other words, the crack density evolution curve seemed to have been shifted backwards.

Figure 5.23 shows what happened when all input variables are kept the same, except for adding a bias of 5, 10, 15, and 20 MPa to strength values that were drawn from the same Weibull distribution. In

other words, one set of strength values was drawn from one Weibull distribution and to assess the effect of increasing transverse strength, the same simulation was done five times: adding 0, 5, 10, 15, and 20 MPa to the set of strength values, while keeping all other variables the same.

The overall trend that can be derived from the plot is that increasing the strength values manually (sometimes literally) shifts the crack density curves towards the right. The orange curve of $Y_T + 5$ is somewhat of an outlier because it is the only simulation that does not have a final crack density of 0.18 cracks per mm. Nonetheless, it can be seen that all curves have a similar pattern in the evolution of matrix cracks, proven by the parallel lines for the majority of the regions of crack density evolution.

It was opted to include the crack patterns that were generated with these 'low outliers', because of two reasons. Firstly, it is not unrealistic to expect a decrease of strength to 60% of the mean value in a poorly produced specimen, which means the dataset includes specimens of different qualities, making it more robust. Secondly, the crack locations that are generated with these low outliers hardly change when one would increase the strength: the evolution is almost identical for all of the curves. All specimens but the orange curve of show cracking at the exact same locations, except for the additional cracks of orange curve of $Y_T + 5$ with respect to the total number of cracks of the other four curves.

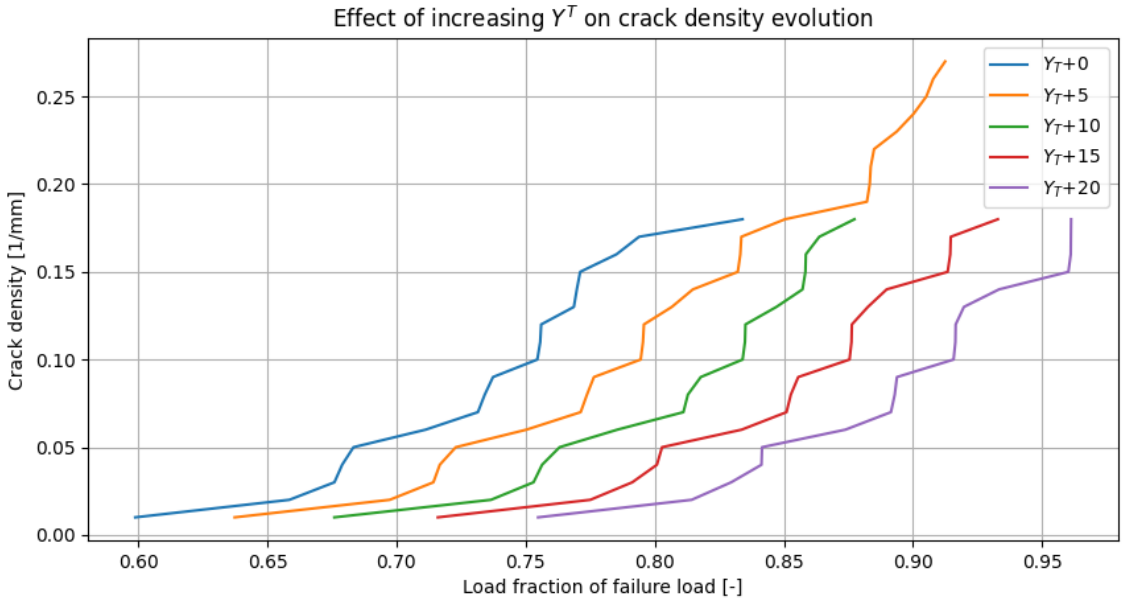

Figure 5.23: Crack density evolution when increasing the strength distribution.

5.4.4. Enrichment Spacing

Enrichment spacing values were varied from 0.25 mm to 1.0 mm. It was observed that a higher spacing postpones crack initiation, as can be seen in Figure 5.24. A logical reason for that would be that an increase in spacing means more columns are taken into account for failure that could have a lower strength value and thereby reduce the required load for the first crack. Furthermore, increased crack spacing seems to promote simultaneous failure at several locations, which shows in the plots as vertical jumps, whereas lower spacing values result in a more smooth crack density curve.

Table 5.11: Sensitivity analysis of the spacing between the enriched columns.

Enrichment spacing [mm]	Failure strength [MPa]	Crack density [1/mm]	Crack initiation [$\mu\varepsilon$]
0.25	664	0.21	8162
0.5	659	0.26	8531
0.75	652	0.2	8602
1	647	0.29	9269

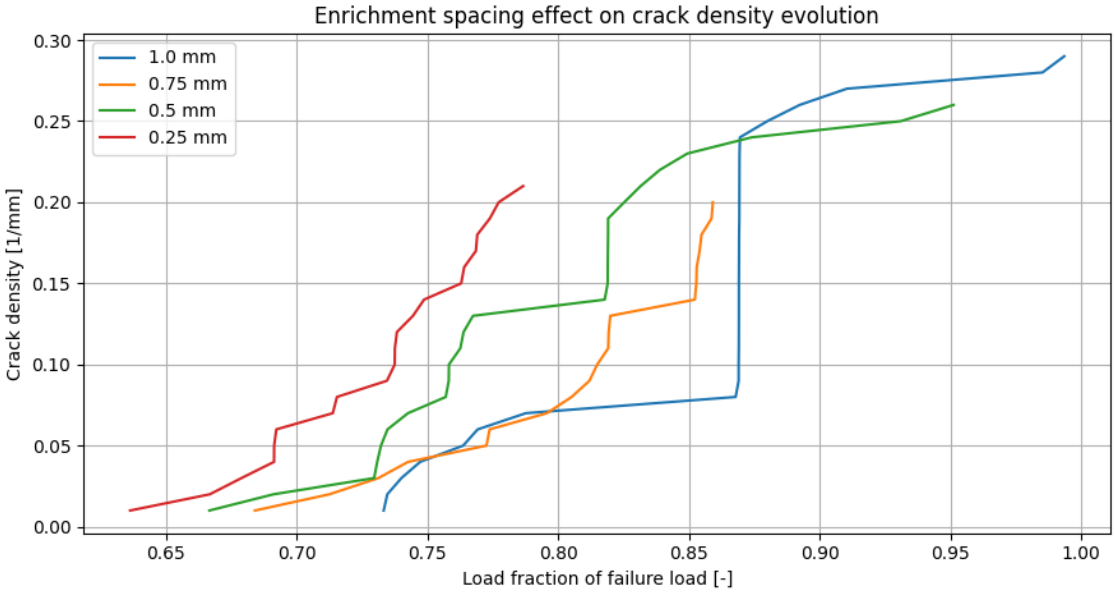

Figure 5.24: Effect of spacing between enriched columns on the crack density evolution curves, simulated using 40 mm specimens.

5.5. Conclusion and Recommendations

A model with varying material properties per element has shown to overcome the deterministic nature of a FEM model and augment the existing experimental data set. As a result of the 2D simplification and limited time resources, straightforward damage parameters were implemented. Verification and validation of the input parameters was done by comparing input values to similar materials found in literature and comparing results with an analytical solution, literature and the experimental data set.

A Weibull distribution of the strength of the matrix was obtained by Li based on the performed experiments. This distribution was directly implemented to generate strength values for the elements. The tensile strength of the transverse plies was correlated with the fracture energy to prevent invalidation of the physical nature of the traction-separation law that was adopted to model the evolution of the matrix cracks after initiation. It provided reasonable results that mostly compare well with experimental data. Deviating crack patterns can be caused because of too low strength values that are drawn from the distribution: 60% of the mean value or lower. Nonetheless, these crack patterns are still deemed relevant to train the data-driven model with, because increasing the strength values produces similar or sometimes even the same patterns in terms of the crack locations and also to account for poorly produced specimens.

The engineering constants and other strength values were modelled using a normal distribution, taking mean values from the data sheet and setting the coefficient of variation to 2% of the mean value. This allows sufficient variation in element properties to generate different crack patterns without affecting the strength and stiffness of the specimen too much with respect to experimental data. A CV of 2% can be considered as quite low, however, the experiments were performed using prepregs, so fewer defects and a higher degree of uniformity of the material properties can be expected.

Furthermore, XFEM-CE turned out to provide reasonable results in modelling the interaction between transverse matrix cracks and delamination with square elements of 0.125 x 0.125 mm. Elements were enriched in columns to prevent convergence issues caused by interacting cracks that deviate from a straight path. The spacing between these columns was based on experimental observations and set to 0.5 mm. Different spacings show similar evolution curves, with the main difference in the point of crack initiation: lower spacings showed earlier crack initiation caused by the inclusion of lower strength values from columns that otherwise would not have been enriched, i.e. allowed to fail. Enriched columns in the matrix extended towards the longitudinal plies, to allow evolution of a matrix crack to the longitudinal plies and ultimately cause failure of the specimen, as was observed to be the failure mode in experiments.

A cohesive surface was adopted to model delamination using traction-separation laws for damage initiation and evolution. Allowing for delamination to occur around the tips of the matrix cracks showed to play an important role in the evolution of the crack pattern. Setting the penalty stiffness and fracture energy of the cohesive surface too high prevented delamination to occur and as such prevented stress relaxation around a matrix crack, which caused premature failure of surrounding elements.

Both for the enriched elements and cohesive surfaces, damage evolution parameters were estimated based on trial and error within a range of values that were found in literature. The established parameters show to be able to model the damage mechanics deservingly.

Convergence issues were encountered that potentially could be solved with a finer mesh, but would increase the computational efforts by too much to generate a sufficiently large data set within the available time.

The following recommendations are made to further develop the proposed Abaqus model:

- Material property distributions:
  - The transverse strength Y_T is the only parameter for which the distribution was experimentally observed. The distribution of other parameters, especially X_T, E_{11}, E_{22}, are expected to most significantly affect the outcome of the FEM simulations. Experiments to obtain those distributions are likely to improve the accuracy of the results.

  - A spatial correlation between distributions of several parameters exist that could be modelled into a multivariate distribution. As such, properties such as the tensile modulus and tensile strength can become interlinked, which would be in line with the research by Sriramula and Chryssathopoulos [142]. The spatial correlation could be obtained by relying on the micro-mechanics scale, for example by characterizing the stochastic behavior based on simulations of stochastic representative volume elements [110].
  - Using the obtained distribution for the transverse strength values sometimes results in an underestimation of the strength of the matrix for the adopted damage initiation and evolution laws. Improvements can be made in terms of the adopted damage initiation and evolution laws (as discussed in upcoming bullets). Consequently, the compatibility between chosen damage laws and the strength distribution should be investigated more closely. It could be investigated what the minimum strength of the matrix would be and adjust the Weibull distribution accordingly.

- Generalization:
  - The failure initiation criterion of maximum principal stress seems sufficiently accurate for this layup and load case, but severely limits the range of other types of load cases and layups to which the model can be applied as it is built. Nonetheless, altering the failure criterion can relatively easy be done by using the Python script in case one wants to use one of the other "standard" failure criteria that is used. Otherwise, the script is compatible to use in combination with a subroutine, such as UDMGINI or UMAT. This extends to the current damage evolution laws: more complex load cases and layups require the adoption of mixed-mode fracture.
  - Including fatigue to assess the effect of variability on predicted life and stiffness degradation would be very interesting. Using a UMAT subroutine seems as a viable and suitable solution to implement it. Because the same experiments using the same material have been performed already by Li, one could use the model that was made for this thesis as a starting point and can focus completely on implementing a fatigue loading.
- Loading
  - The specimens that formed the basis of the data set were tested under varying loading rates, which has been shown to affect damage evolution [31]. By including this effect in the FEM model, the model becomes more representative for the experimental data.
  - Both experiments and simulations were damaged by gradually increasing the load until failure, without unloading in between. It should be assessed how much the damage pattern changes when unloading in between. In case it does change significantly, the applied loading in the experiments and FEM model could be tailored such that the digital twin is also valuable when its physical twin is completely unloaded at a point where cracks already exist, after which it is brought back into operation before repairs.
- Damage modelling - XFEM
  - A more sophisticated initiation criterion to model the fiber failure is expected to improve accuracy, because with the MAXPS criterion, the shear stress in the longitudinal plies near the interface region is not taken into account. Including a more sophisticated initiation criterion is expected to reduce the strength of the fibers, because the introduced shear stress is essentially added to the normal stress that is already applied to the fibers.
  - The crack spacing as is limits the exact locations where cracks appear as well as any interaction between two adjacent cracks that are close to each other. Thus, by shifting the starting point and varying the spacing, new types of crack patterns can be simulated.
- Damage modelling - Delamination
  - More extensive research and experiments should be performed to verify and validate the mechanics more accurately. Frictionless surfaces are now assumed, whereas including friction will affect the results. Thus, to simulate the physics more accurately, modelling the friction should be looked into more carefully.

  - The use of cohesive elements rather than a cohesive surface would allow to vary properties of the interface as well. End-notched flexure experiments were performed by Motamendi et al. from which it was concluded that the variation in fracture properties of specimens from the same batch affects the result [143].

- 2D model
  - Extending the model to 3D would greatly increase the computational efforts, but in its turn would allow to model and validate crack patterns that are observed in experiments, as well as in 2D model. Free-edge delaminations are currently not taken into account whereas out-of-plane stresses at the free edges are not zero. Furthermore, by adding the width direction, the propagation direction of both delaminations and matrix cracks can be observed and taken into account.

- Convergence
  - A relatively large portion of the samples in database that was used for training the neural network consists of simulations that were not finished. Underlying issues that have been identified should be solved, as well as further analysis of other unidentified issues.

6
Data-Driven Digital Twin - A Deep Learning Approach

The data-driven digital twin that predicts the location of the next crack and its corresponding load based on the crack state is proposed as an artifical neural network in the form of RNN-based LSTM layers. It is defined in Python [129], using the Keras library [144]. For an explanation on the theory behind the working principle of the neural network, the reader is referred to subsection 2.4.2.

The preparation of the data set is discussed in section 6.1, followed by a description of augmentation procedures that extend the data set further in section 6.2. In section 6.3, the hyperparameters are explained. Model evaluation metrics are discussed in section 6.4 and the training procedures are depicted in section 6.5. After the results and a discussion are presented in section 6.6, section 6.7 includes a helicopter view of the digital twin within its intended environment. section 6.8 provides a conclusion and recommendations on the data-driven digital twin.

6.1. Data Preparation

A first step in data preparation is deciding what and how many input values should be used in predicting the target value(s), or put differently: feature selection. Increasing the number of features potentially benefits the performance of the model because it is fed with more information, but on the other hand more features increases the computational cost and potentially increases the required size of the training data set, which in its turn would degrade the model's performance. Therefore, in the step of feature selection, the number of input variables is reduced to the minimum.

A lot of information about the various (damage) state variables of a specimen can be extracted via sensors during its life. However, this project focuses on predicting characteristics of the crack pattern, which is why the input features per specimen are limited to those that directly relate to the crack pattern: a vector with locations of all the cracks and a vector of the corresponding loads at which they initiated at time t. To elaborate: when the first crack initiates, its location and initiation load are fed into the model, which then provides a prediction on the location and load of where the next crack is going to be. This process repeats until the specimen is completely failed.

The crack locations that are generated from the FEM model are discretized, caused both by the fact that FEM discretizes a structure by definition and the induced crack spacing between the enriched columns. This makes the prediction of crack locations a multi-class classification problem, where all possible locations are considered to be the classes. An advantage of such a problem is that the output specifies a probability per bin. If, for example, the next crack is likely to occur at either 64 mm or 23 mm, then it should show higher probability on the output at those locations, whereas regression would result in a single answer and would only show the single most likely location.

Loading is applied as displacement-controlled, which is why the loading is expressed in strain. Being numerical values, the problem of predicting the strain at which the next crack initiates is a regression problem.

Before locations and loads could be fed as input to the model, the data needs to be transformed into the right format. Because the nature of the problem differs between the locations and loads, they are discussed separately.

Crack Locations When modelling the crack locations as a classification problem, the crack locations are provided as binary arrays that represent the specimen over the length. Each value corresponds to the crack state of a portion of the specimen that equals the discretization width. Each portion is either cracked (1) or uncracked (0).

Increasing the discretization width results in a less accurate prediction of the location, whereas setting this width too small results in too many possibilities for the model to accurately predict the next crack. The minimum spacing observed in experiments was approximately 0.5 mm. Using 0.5 mm results in 200 'bins'. Increasing it to 1 mm could improve the performance of the model, but requires additional handling of cracks that appear within the same bin, e.g. when cracks appear at 50.1 mm and 50.8 mm. By using this discretization, the prediction is expressed as a probability distribution over the length of the specimen. It was considered to provide a more nuanced prediction than a single location.

Crack Loads The crack loads are specified in global applied strain. Regression problems require scaling of the input features, which can be done either by normalization or standardization. Normalization scales all values from a minimum to a maximum, commonly from 0 to 1, which is done by subtracting the minimum value from all values and divide the values by the maximum minus the minimum:

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{6.1}$$

Alternatively, standardization subtracts the mean value and divides by the standard deviation:

$$x_{scaled} = \frac{x - \mu}{\sigma} \tag{6.2}$$

Standardization can be helpful in case the input or output variables have a Gaussian distribution or when there are outliers in the data set that would crush a normalization scale. Because such outliers are not expected in the crack loads, normalization is applied. The predicted load is expressed as a numeric value.

6.2. Data Augmentation

Due to the high runtime of the FEM model, the total number of samples that could be obtained was limited. Approximately 100-150 specimens were simulated using FEM, whereas the training of a deep neural network generally is done with a larger training set, such as in [145], where 500 instances of data were used to train a surrogate model for fracture analysis of composite materials. Therefore, further augmentation of those obtained was deemed necessary.

If a crack pattern has n cracks, it is fed to the network $n-1$ times. A pattern is split after each crack, such that each crack is predicted once. For example, consider a pattern that consists of five cracks: the second crack gets predicted based on the first crack, the third crack is predicted based on the first two cracks, and so on until the final crack is predicted. This increases the number of samples that the neural network sees by approximately a factor 20, depending on the average number of cracks per crack pattern.

Next, a single pattern is used to create two more by shifting all crack locations x mm to the right and x mm to the left. The idea here is that the physical relationship between the cracks is maintained by keeping the distance between cracks the same, but different numerical values are linked to the cracks. The loads are kept the same to refrain from invalidating the physics. This converts a single pattern into three.

Important to note here is that after these modifications of the data set, the patterns are shuffled. Otherwise, the model risks to overfit each pattern after it is being fed the same data right after each other, meaning the final model parameters would be overfitted to the last pattern the model has seen.

Finally, data instances were 'padded' such that each pattern that is fed to the neural network consists

of arrays of the same length. Based on the longest array in the data set (corresponding to the pattern with the most cracks), all other arrays were concatenated with zeros until they had the same length. To prevent the neural network from interpreting the zeros as data, masking was required. Masking means that each array that is inputted to the network is accompanied by an array of the same length with boolean value that inform the network about which values to interpret and which values to ignore. In other words, when the network encounters a zero, nothing is done.

Before discussing the model parameters in the next section, a top level overview of the process of processing data from Abaqus simulations and the experimental data towards the final training and testing data is provided in Figure 6.1.

6.3. Model Hyperparameters

This section explains the hyperparameters of the neural network that was used for crack predictions. The majority of the theory behind the working mechanisms has been explained in subsection 2.4.2. Otherwise, it is briefly explained here. The loss functions and optimizers are explained in subsection 6.3.1 and subsection 6.3.2 respectively. The model was trained via *Google Colab*, which allows to use GPU's that have been made available by Google. Using the GPU's speeds up training immensely, but does introduce certain constraints, which are explained in subsection 6.3.3. Hyperparameters that were introduced to overcome overfitting specifically are described in subsection 6.3.4, followed by the different types of layers in the Keras environment that were adopted in the final architecture of the model as listed in subsection 6.3.5.

6.3.1. Loss Functions

Different types of loss functions are required for the locations and loads due to the difference in nature of the problem. The prediction of the location of the next crack is probabilistic, whereas the prediction of the load of the next crack is numeric.

Multi-class Classification

For multi-class classification problems, categorical cross-entropy is the default loss function. It calculates a score based on the average differences between predicted probability distributions for all the classes. As mentioned before, the definition of the optimization problem is to find a set of model (hyper)parameters that minimizes the loss. The mathematical formulation of cross-entropy is:

$$L_{CE} = -\sum_{i=1}^{n} T_i \log(S_i) \tag{6.3}$$

where T_i is the label, or truth, and S_i the predicted probability.

Regression

The mean squared error is the default loss function for regression problems. It was defined previously in subsection 2.4.2 as:

$$L_{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(y_i^{\text{test}} - \hat{y}_i^{\text{test}}\right)^2 \tag{2.3}$$

It calculates the average of the differences between the prediction and target value squared. By squaring this error, large errors are penalized more severely than small errors. Ideally, its value is zero.

In case one does not want severe punishment for large errors, an additional step can be included in calculating the error by taking the natural logarithm of the values:

$$L_{MSLE} = \frac{1}{n}\sum_{i=1}^{n}\left(\log(y_i^{test}+1) - \log(\hat{y}_i^{test}+1)\right)^2 = \frac{1}{n}\sum_{i=1}^{n}\left(\log\left(\frac{y_i^{test}}{\hat{y}_i^{test}}\right)\right)^2 \tag{6.4}$$

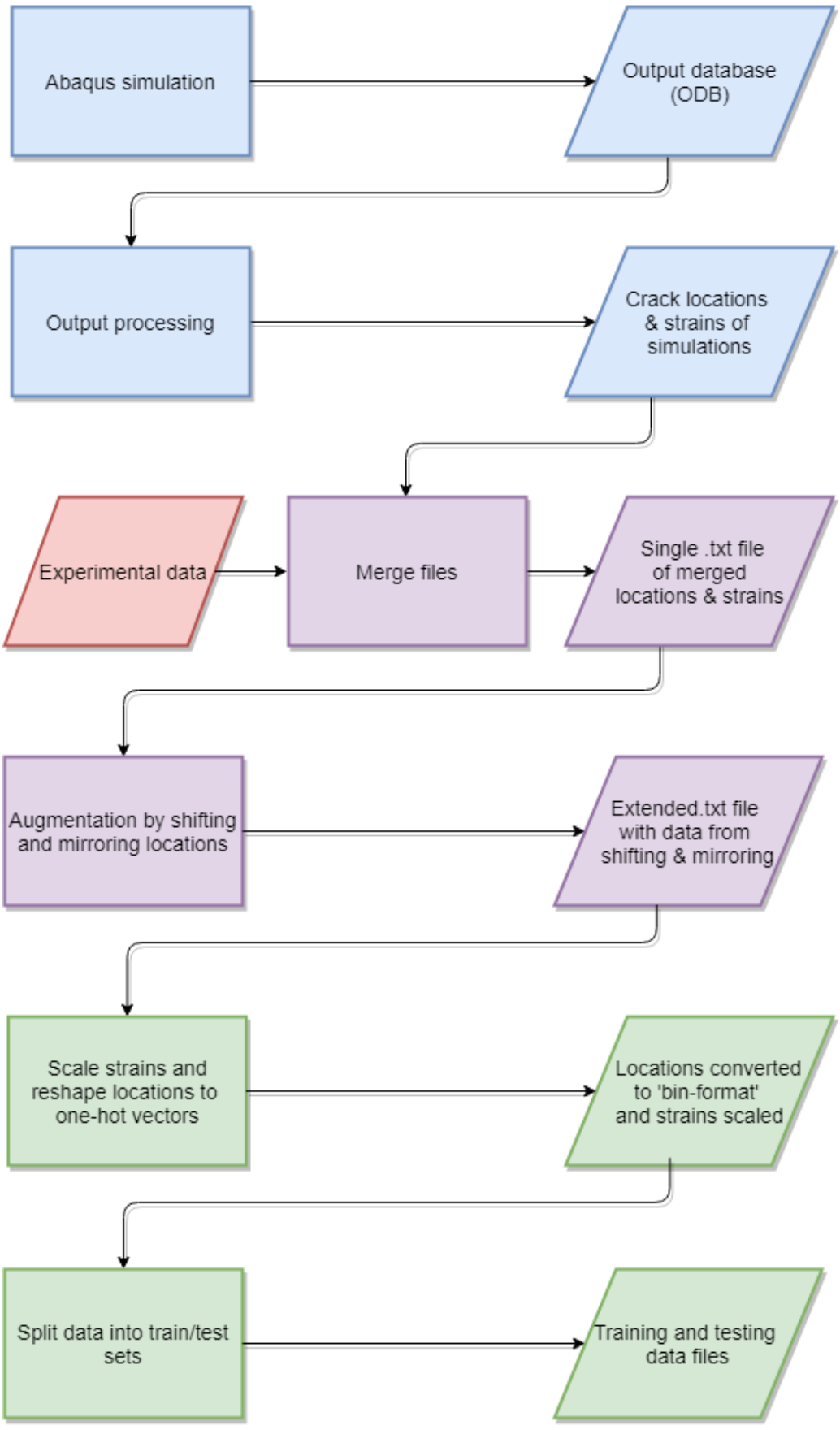

Figure 6.1: Data flowchart from Abaqus and the experiments towards the train and test files that were used as input to the neural network. Square boxes indicate a process, parallelograms indicate data.

6.3.2. Optimizers

The optimizer specifies the algorithm that updates the weights and biases of the network during training. A popular optimizer that uses momentum is the so-called 'Adam' optimizer [146], which stands for adaptive moment estimation. Not only does it calculate the exponential moving average of past gradients, but also does it calculate the moving average of the variance of those gradients. It introduces two additional parameters that are linked to these two properties: β_1 and β_2, which control the decay rates of the momentum and variance. The Adam optimizer has become a benchmark in literature [145]. It will be compared to the earlier discussed SGD algorithm.

Learning rate The learning rate determines the steps that are taken during gradient descent, as described earlier. The learning rate can be configured according to a schedule, meaning it can change during training by becoming a function of the epochs. As such, early steps in training can be relatively large, while smaller steps are taken once the algorithm is already closer to a minimum.

A popular learning rate schedule makes use of exponential decay, such that the learning rate decreases exponentially as training progresses and, possibly, as it comes closer to the minimum. Alternatively, a cyclic learning rate schedule can be used, which alternates linearly between a minimum and maximum learning rate, as is shown in Figure 6.2. It was proposed by Smith in 2015 [147] and is said to eliminate the need to experimentally find the best values for global learning rates and monotonically decrease the learning rate during training. The cyclic learning rate shows an increases in classification accuracy without lots of tuning. As such, a cyclic learning rate is adopted here.

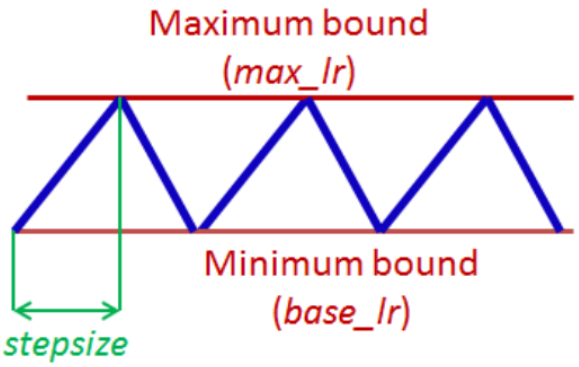

Figure 6.2: Triangular learning rate. Blue line indicates the learning rate that is changing between the maximum and minimum learning rate bound. The stepsize is the number of iterations in half a cycle [147].

6.3.3. LSTM Layers in Keras

To be able to make use of the computational efficiency of GPU's, *Keras* has the following constraints on its architecture with LSTM's:

- Activation function must be tanh
- Activation function of hidden state must be sigmoid
- Drop-out of hidden state must be zero
- Bias has to be included
- If masking is used, inputs must be right-padded

As such, these settings were implemented and limited the options regarding the activation functions for the hidden layers. These setting restrict flexibility in terms of the activation functions of the LSTM layers, but these activation functions are popular for their performance and have shown proper results in the context of damage prediction in composites [145].

6.3.4. Overfitting

In general, overfitting is one of the most prominent issues in deep neural networks [148]. It means the model is memorizing the training data but performs poorly when it is inputted data that the model

has never seen before. As will be more elaborately be discussed in section 6.5, overfitting was a regularly encountered problem during training. A brief overview is presented here on hyperparameters that were implemented to attempt to overcome overfitting.

Dropout Dropout improves regularization of the model by disconnecting nodes during each training epoch. These nodes are randomly picked and could be part of either the input layer or one of the hidden layers. It can be configured how many of the neurons are dropped out each epoch. For an RNN, this is typically 20-30% of the nodes [114]. When 20-30% of the nodes are not included during a training step, it comes down to training a 'new' neural network during each epoch. Including dropout makes the network less dependent on a small group of nodes, thereby making it more robust.

Batch size A smaller batch size splits up the training data into more subsets, meaning the model is exposed to more data. A result of this is that training takes longer per epoch, with the upside that the model learns in less epochs. A smaller batch size also corresponds with the earlier described 'bouncing around', which has the upside of having the possibility to jump out of a local minimum, but the downside of jumping around or away from the global minimum. Increasing the batch size generally slows down learning and thereby smooths the loss function curves, however, if it is increased by too much, the model may stop learning early.

Early stopping Early stopping means to stop the training of a model once the loss is at a minimum or before the training and validation error start to diverge, i.e. before overfitting starts. It was implemented in the form of a callback in Keras. A callback allows to monitor model parameters, such as the validation loss, during training after each epoch. As such, in case the validation loss stops to decrease, the model is either overfitting or has stopped learning. In practice, you set the parameter you want to monitor, the minimum change of that parameter that is still considered to be an improvement, and the number of epochs that the callback remains patient, i.e. a slow-down of loss for one epoch should ideally not stop training.

Regularizer A regularizer adds artificial loss to a certain layer, which increases the total loss of the model. Consequently, the model keeps changing its parameters and as such it should regularize. L1 and L2 regularizers are popularly used and relatively simple to implement via Keras.

6.3.5. Types of Layers

The functional API of Keras was used, allowing multi-input and multi-output modelling. The location and strain arrays were inputted separately in different input layers and output separately in different output layers. In between, LSTM, Dense, and Concatenate layers were used. Only the arguments that were used are discussed.

Input — *tf.keras.Input()* — The layer in which the data is inputted.

- *shape*: specifying the shape of the input data. In case of locations, this setting was set to (None, n_bins). 'None' allows a variable input length of data. 'n_bins' specifies the number of bins that was used to discretize over the length of the specimen.

Masking — *tf.keras.layers.Masking()* — The masking layer that was used to learn the network to ignore the padded zero input values and is placed directly after the Input layer.

- *mask_value*: the value which should be masked, i.e. ignored, being 0 in this case.
- *input_shape*: similar to the Input layer, the input shape is matched to the shape of the input data.

LSTM — *tf.keras.layers.LSTM()* — A LSTM layer consisting of the LSTM cells as described in subsection 2.4.2. Implementations that were listed already in subsection 6.3.3 are repeated here for the sake of completeness.

- *units*: specifying the number of nodes within the layer.

- *activation*: the activation function of the layer.
- *recurrent_activation*: the activation function of the hidden state.
- *return_sequences*: boolean that specifies whether to return the last output or the entire sequence.
- *dropout*: the fraction of all nodes in a layer that is dropped each layer to increase robustness of the network, which is further explained in section 6.5.
- *recurrent_dropout*: dropout fraction of the hidden state.

Concatenate — *tf.keras.layers.concatenate()* — A layer that was used to merge the crack locations and strain values. The layer concatenates the values from the layers from which it receives input and outputs a single array, without applying any functions to these values.

Dense — *tf.keras.layers.Dense()* — A dense layer that was used in between LSTM layers and as output layer.

- *units*: number of nodes in the layer, which in case of the output layer should correspond with the preferred number of outputs. In case of the locations, this value is set to the number of bins. For the strains, it is set to 1.
- *activation*: the activation function for the layer. A ReLU activation function was used for all Dense layers that were not the output layers. The activation functions for the output layers is discussed below.

6.4. Model Evaluation

Metrics are used to assess the performance of the model, but are, in contrast to loss functions, not used in the training of the model. The mean squared error is used both as loss function and as metric to assess the accuracy of the strain predictions, its mathematical formulation is shown in Equation 2.3. Regarding the assessment of accurately predicting the of the location, a "top-k accuracy" metric was adopted. It is a popular metric for classification problems. A prediction is labeled as accurate when the target value appears in the top k predictions. The default setting is k=5, meaning a prediction is labeled as accurate if the target is in the bins with the 5 highest predicted probabilities.

Train-validate-test split The entire data set is split up into a training set, validation set, and testing set. The training data is used to train the model, i.e. adapt the model parameters based on the chosen optimizer. The validation data is used for model selection (e.g. configuring the number of layers or nodes per layer) and tuning the hyperparameters (e.g. the learning rate). Per epoch, the newly updated model is evaluated by using the validation data, which is particularly useful in assessing if the model overfits and from what point. The testing data is a separate data set that can be used to assess the performance of the final model once training is finished. The reason that testing data was deemed necessary is because of the model selection: it allows a fair comparison in model performance between different models (e.g. 1 hidden layer compared to 5 hidden layer) on the same set of completely unseen data. By making use of the testing data set, there is no bias, which would have been there if validation sets were used to evaluate performance, because tuning the hyperparameters is partially based on the validation sets.

The split in data that was made is: train/validate/test $\tilde{7}0/20/10$. The test set was created before augmenting through mirroring or shifting, or modification of the data. 4 experimental crack patterns were included in the test, together with 27 crack patterns from FEM. No augmentation through mirroring or shifting of the data was performed on the test set. The remaining 4 experimental patterns and 259 FEM patterns were combined into a training set. Once the patterns are fully augmented and split into patterns per crack, the set is shuffled. Using *validation_split* within the *Keras* environment, the set is split into a training and validation set corresponding to the 70/20 split.

It is given that the appearance of at least the first crack is random and fully determined by the the distribution of the properties. The second crack is most likely also randomly distributed, perhaps the

third as well. It is investigated to see if the model is able to predict the crack location given a crack history. To investigate that, a new testing data set was created by changing the format of the initial one by splitting it up and sorting by number of cracks. As such, the model's performance can be evaluated per crack number, i.e. a model performance can be obtained for predicting the second crack based on the first crack, the third crack based on the first two cracks, and so on.

6.5. Training

The final architecture was configured by training models with different architectures and selecting the one that performs best. This section explains the systematic process of training and finetuning that led to the final architecture. Two types of networks were configured: a network where the data of crack locations and strains is combined into a single "coupled" model and networks where the locations and strains are predicted separately. As such, it can be assessed if combining the information of location and strain improves the performance. To obtain the decoupled networks, the coupled network was simply decoupled by removing the concatenation layer without further modification of the parameters, to allow for a fair comparison. Furthermore, a prediction horizon was set up, which predicts multiple cracks ahead instead of just one. As such, the model's ability to predict further into the future can be assessed.

After a brief description of techniques to overcome overfitting in subsection 6.5.1, the training of the coupled model is discussed in subsection 6.5.2. In subsection 6.5.3, it is explained how the models were set up that were used to generate the prediction horizon.

6.5.1. Overfitting

Overfitting was found to be a prominent issue, for which several techniques exist to cope with it. A brief description of the techniques with the effect is provided before the systematic training procedure is presented that was followed to result in the configuration as is, because the described techniques were an an integrate part of the process of training in an iterative manner. Overfitting of the data was constantly encountered during training, which is, in general, one of the most prominent issues in deep neural networks [148]. It means the model is memorizing the training data but performs poorly when it is inputted data that the model has never seen before. Besides increasing the size of the data set by augmentation via FEM and mirroring and shifting of the patterns, one of or a combination of the following actions were taken when overfitting was encountered:

- Reduce the complexity of the model, i.e. reduce the number of layers and/or nodes.
- Use early stopping, so stop training at the moment that the training and validation error start to diverge.
- Adjust the learning rate.
- Increase the drop-out rate.
- Increase the batch size.
- Try a regularizer, which essentially penalizes the loss during a training step for the whole model. Regularizers can be used for the weights, biases, output.

The working principle of these hyperparameters is discussed in subsection 6.3.4.

6.5.2. Training of the coupled network

An infinite number of possibilities of neural networks exist, considering the options for number of layers, activation functions, optimizers, learning rates, and so on. To come up with a network, the architecture was explored first, i.e. the number of layers and number of nodes per layer. To gradually narrow down the many options to a single model to finetune further, the following procedure was followed:

1. Explore architectures, i.e. the number of layers and nodes per layer — narrow down to three architectures
2. Find range of suitable learning rates for cyclic learning rate schedule

3. Implement learning rate schedule and compare — compare performance metrics across models and move on with one model for further finetuning
4. Adapt batch size
5. Implement drop out

1. Explore architectures A variety of configurations was tested with mostly default settings to get a grasp of what combination of layers and nodes per layer is most suitable. The following choices were made regarding the basics of the architecture. Separate input layers are used for the location and strain data and right after these input layers, masking layers are placed to make sure the padded zeros are ignored. Within the model, the data is concatenated to include dependency between the location and load. It was opted to vary the number of LSTM layers before concatenation of data and have one LSTM layer after the concatenation layer. Initially, the number of nodes in this final layer was fixed to be equal to the number of bins + 1 that was implemented, such that one node is available per output at the final hidden layer. The number of bins was quickly determined to be 100; setting it to 50 and 200 showed lower accuracy for initial models that were tried.

Accordingly, the dense output layers are also separated, for which the activation functions determine the format of the predictions. For a classification problem, either a softmax activation function or a sigmoid activation function can be used. Softmax activation function is used in case the probabilities are dependent (i.e. the probabilities sum to 1) and a sigmoid activation function is used when probabilities are independent (i.e. the probabilities do not sum to 1). The purpose of the digital twin in the first place is to reveal a relationship between the different cracks, which implies the assumption of dependency between cracks. Therefore, the activation function of the output layer for classification of the next crack location was implemented with a softmax activation function. The activation function of the output layer for the regression problem of predicting the next crack load was set to rectified linear to allow for non-linear behavior and output a real, continuous number.

Determining the number of layers and number of nodes per layer before concatenation was the first step in configuring the network. Regarding the number of LSTM layers and number of nodes per layer the following combinations were tried:

- Number of layers before concatenation - 1/2/3/5
- Number of nodes per layer - 2/8/16/32/64/128/256

These networks were trained with a learning rate of either 10^{-3}, 10^{-4} or 10^{-5}; a fully augmented data set, i.e. including the patterns from mirroring and shifting; and a dropout setting of either 0% or 30% (recommended value from Géron for RNNs [114]). Furthermore, constraints were imposed on the LSTM layers as explained in subsection 6.3.3. For each combination of number of layers and number of nodes, the model with the lowest loss was picked. The resulting loss & accuracy of the models from this initial training are shown in Appendix C. Due to the overfitting, an early stopping callback was implemented to stop diverging models from training to save time. The early stopping was configured such that if the validation loss would start to increase by 0.0001 without decreasing for 10 epochs, training was stopped and the model parameters associated with the lowest validation loss were adopted as final.

From the initial configurations, it becomes clear that models with less degrees of freedom are preferable with the current data set:

Table 6.1: Preliminary model selection.

Number of layers before concatenation	Number of nodes per LSTM layer
1	8
2	8
2	16

It makes sense that these models were most successful, considering that overfitting was constantly encountered, which indicates the need for less complexity in the model.

2. Find range of suitable learning rates for cyclic learning rate schedule The next step that was taken was to tune the learning rate. A scan of the 'loss landscape' was made by varying the learning rate between 10^{-8} up to 10^{0} and calculating the loss after 10 epochs of training. From these scans, the learning rates at which the model learns best were derived. These learning rates form the basis for a learning rate schedule. Loss landscapes were made for both the Adam optimizer and a SGD optimizer with default settings for momentum. To generate these plots, the data from augmentation via shifting and mirroring of the basic patterns was excluded from the training set to reduce computational efforts. Because these plots are aimed at providing an overview of a range of proper learning rates, excluding the shifting and mirroring data is not deemed to affect the result. The plots are shown in Figure 6.3 and the resulting learning rate bound are presented in Table 6.2.

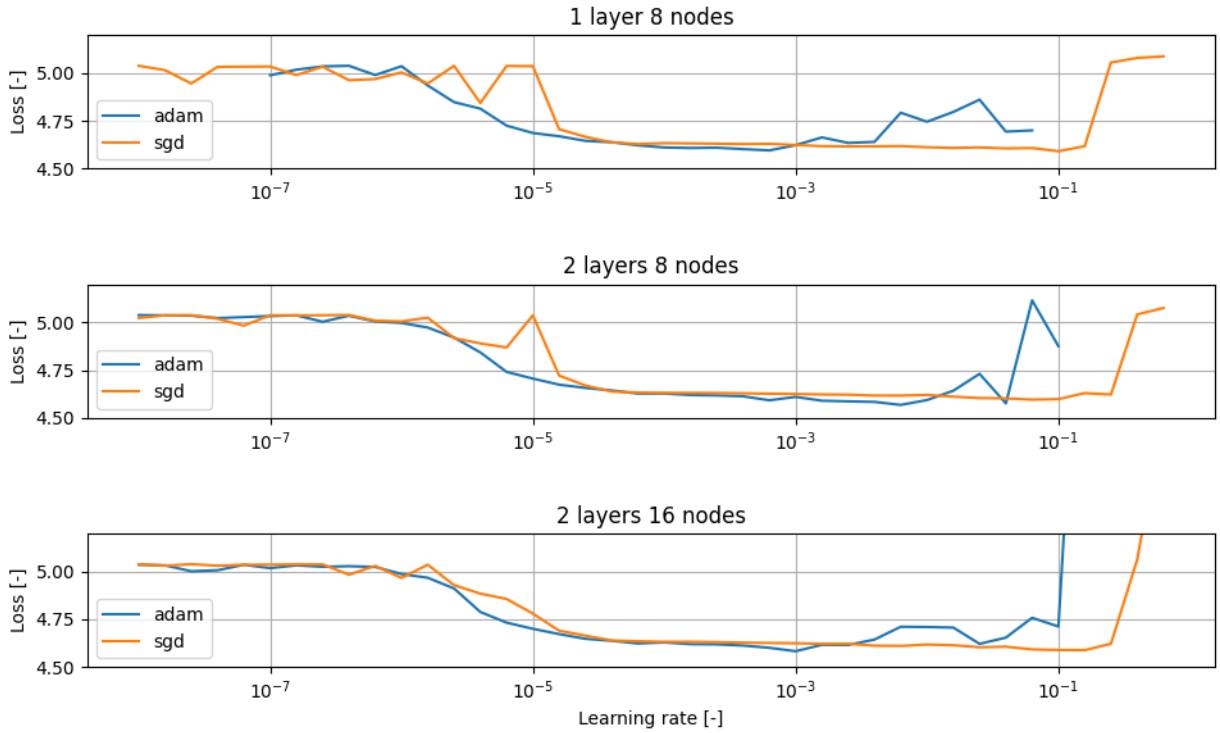

Figure 6.3: Loss landscapes of the three different architectures for both the adam and SGD optimizer with momentum. Generated by varying the learning rate and the resulting loss after training for 10 epochs.

Table 6.2: Learning rate boundaries for the cyclic learning rate schedule, obtained from loss landscape plots.

Ref.	Layers	Nodes	Optimizer	Min. learning rate	Max. learning rate
1	1	8	Adam	$2{\cdot}10^{-5}$	$6{\cdot}10^{-4}$
2	1	8	SGD	$4{\cdot}10^{-5}$	$6{\cdot}10^{-2}$
3	2	8	Adam	10^{-5}	$4{\cdot}10^{-2}$
4	2	8	SGD	$2{\cdot}10^{-5}$	10^{-2}
5	2	16	Adam	10^{-5}	$6{\cdot}10^{-3}$
6	2	16	SGD	$2{\cdot}10^{-5}$	$3{\cdot}10^{-2}$

3. Implement learning rate schedule and compare A cyclic learning rate schedule as explained in subsection 6.3.2 was adopted. It requires four inputs: the base or minimum learning rate, the maximum learning rate, the step size, and the scale function. The minimum and maximum learning rates were the result of the loss landscapes scan as described in Table 6.2.

The stepsize needs to be determined in terms of number of iterations. The number of iterations is a function of the total number of samples and the batch size. In case of all training samples are used with the default batch size of 32, it results in one epoch having 117 iterations. Smith recommends setting the stepsize to a factor in the range of 2-10 of the number of iterations per epoch, such that

the end of a learning rate cycle coincides with the end of an epoch [147]. Setting it to 2 results in a relatively 'wavy' behavior of the loss due to the quick fluctuation of learning rates. It was opted to set the stepsize to 10 to reduce the waviness and maintain the early stopping callback. The scale function determines the shape of the learning rate cycle function. Smith obtains the best results using a triangular shape function (Figure 6.2), which was also adopted here.

The results were obtained with a fully augmented data set. During training, models with Adam optimizers showed more bouncy and unstable training than the SGD optimizers, as shown in Figure 6.4. The earlier mentioned 'waviness' due to the cyclic learning rate is also clearly visible for the training loss of the SGD optimizer. Also, it can be seen that the Adam optimizer results in quite fast overfitting, whereas the SGD optimizer remains fairly stable.

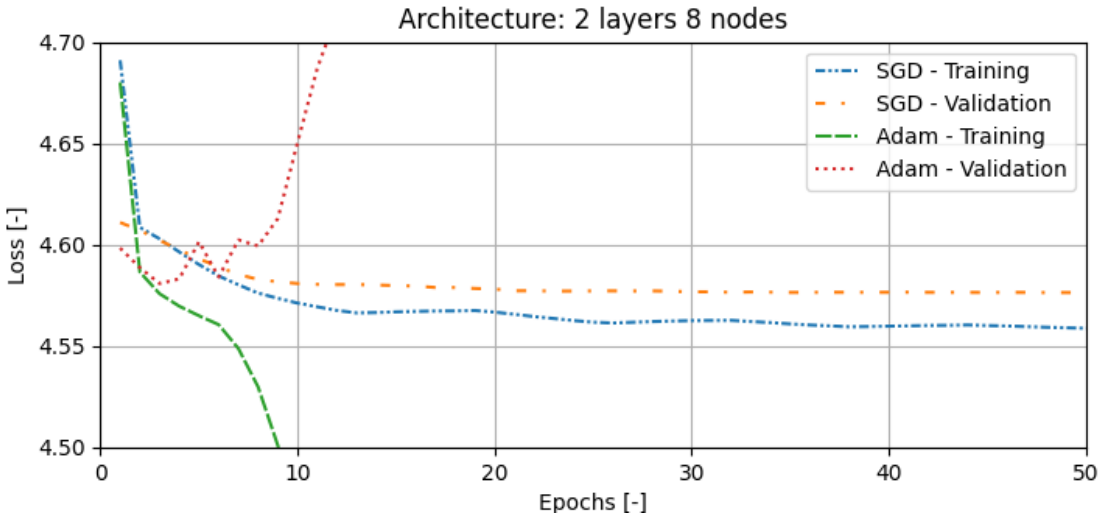

Figure 6.4: Training of architectures with 2 layers before concatenation and 8 nodes per layer including the cyclic learning rate schedule with boundaries as shown in Table 6.2.

The resulting loss, MSE, and top-5 accuracy of the settings as shown in Table 6.2 is presented in Table 6.3. The best performing model, both in terms of MSE and top-5 accuracy, is model 2, i.e. 1 layer before concatenation, 8 nodes in the layer, and a SGD optimizer with momentum.

Table 6.3: Performance of the models from Table 6.2.

Ref.	Loss	MSE	Top-5 accuracy
1	4.6085	0.0079	0.051
2	4.6021	0.0049	0.0743
3	4.6082	0.0102	0.0403
4	4.6113	0.0125	0.0658
5	4.594	0.0061	0.0701
6	4.6043	0.0114	0.0637

4. Adapt batch size Different batch sizes between 4 and 128 were adopted to evaluate differences. The final outcome did not vary significantly between different settings. The main difference showed early on in training, as shown in Figure 6.5, and in the required training time per epoch: lower batch sizes increased the training time because the training set was split into more batches.

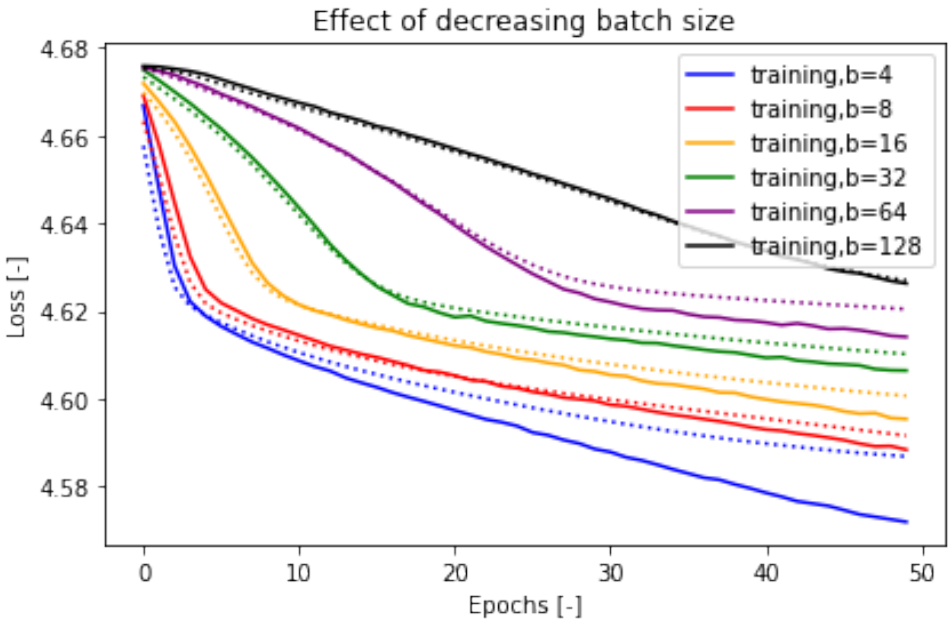

Figure 6.5: Effect on the training loss of various batch size settings.

5. Implement drop out To evaluate the effect of implementing drop out, six settings were attempted: 0-50% with steps of 10%. Drop-out was applied at the LSTM layer after concatenation. The layers before concatenation only consist of 8 nodes, which is why applying drop out in those layers would leave too few nodes active and destabilize training of the model in an unfavorable manner. Table 6.4 shows that the top-5 accuracy is fluctuating between all different drop out settings but the MSE steadily decreases with decreasing drop out rate. Therefore, no drop out was implemented after all.

Table 6.4: Performance of the model when using drop out.

Drop out	Loss	MSE	Top-5 accuracy
0	4.6150	0.0060	0.0573
10%	4.6160	0.0121	0.0594
20%	4.6233	0.0145	0.0637
30%	4.6296	0.0189	0.0446
40%	4.6226	0.0187	0.0510
50%	4.6286	0.0200	0.0594

6.5.3. Prediction Horizon

Setting up the prediction horizon provides an indication on the model's capability to predict further ahead into the future. The goal here is to establish a prediction horizon in the form error evolution as a function of predicting several cracks further ahead into the future and relate the errors to a confidence interval.

Predicting several cracks ahead into the future can be done by two ways: 1) predicting the next crack and adding that to the crack history to predict the following crack, and so on; 2) training the model to predict several steps ahead in one go.

The first option would imply accumulation of errors that are associated with each prediction, sometimes referred to as the 'naive' approach. To elaborate, predicting one crack ahead introduces an error to the crack history once the prediction is appended to it. Using multiple predictions as input to predict an upcoming crack only further increases this error.

Thus, the second option was implemented. It required retraining of the model, because the format of the data changes if multiple cracks ahead are predicted at once. Instead of feeding the model a

sequence and learning it to output a single value, it should output a sequence with the length of the number of cracks one wishes to predict ahead.

The prediction horizon was analyzed up to 5 cracks in the future, which was deemed a reasonable number given the size of the data set and the number of samples that are available to test. To predict 2, 3, 4, and 5 cracks ahead at once theoretically requires the training of four additional models: each model being tailored and specifically trained to predict a certain number of cracks ahead. For each of the models, the same architecture and hyperparameter settings were used. Furthermore, the training and testing data was reshaped to match the output data to the number of cracks that was predicted.

As a consequence, the size of the data set decreased as a higher number of cracks was predicted at once. Taking for example a pattern of 20 cracks: based on the 1st crack, up to the 6th crack is predicted, and so on until the first 20 cracks are used to predict up to the 25th. However, not every sample in the training or test set has the same crack pattern length. As a higher number of cracks is predicted, less samples are available for training and evaluation. Accordingly, predictions of the 25th crack are less reliable than predictions of the 8th crack. For this reason, the prediction horizon was established by predictions based on no more than 20 cracks as input.

6.6. Results and Discussion

The architecture is presented in Figure 6.6, where the type of layer is preceded by its name, so 'name: type of layer'. The format of the data shape is added to the graph, which is [batchsize, timesteps, features]. Setting timesteps to 'None' allows variable input in terms of timesteps. 'x' indicate the crack location layers and 'e' the strain layers.

After the input layer, masking is required to ignore the zeros as discussed before, after which one LSTM layer is placed. These layers return full sequences, meaning all time steps are returned in each pass of data. Next, the data of the location and strain values is concatenated in a concatenate layer, after which the data is passed through one more LSTM layer. Finally, the data is passed to the output layers.

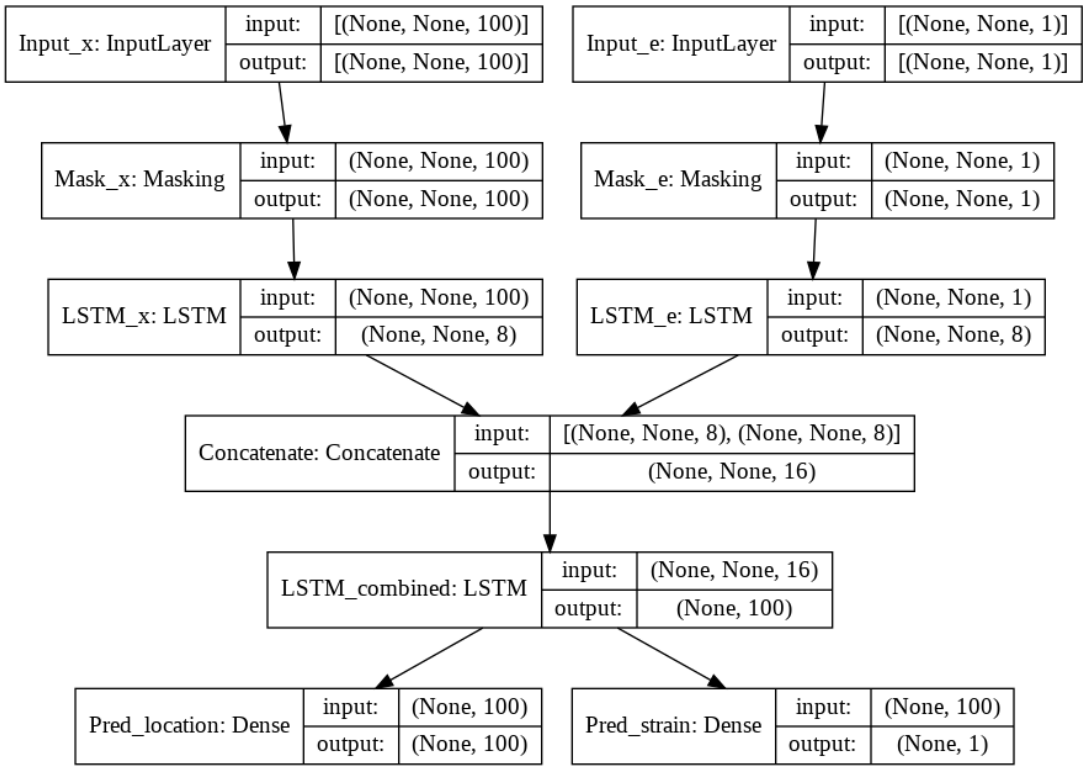

Figure 6.6: Final architecture of the neural network.

The results are presented and discussed in three parts: the overall model performance in subsection 6.6.1, the performance per crack based on the error evolution in subsection 6.6.2, and the prediction horizon in subsection 6.6.3.

6.6.1. Overall Model Performance

Table 6.5 shows the performance metrics for the coupled and decoupled models. Figure 6.7 shows an example prediction of the location of the 15th crack, where the blue dotted lines indicate the inputted crack history, the orange line the outputted prediction, and the green line the ground truth. Similarly, Figure 6.8, shows the prediction of the corresponding strain. A random crack history length was used to illustrate the information that becomes available within a prediction.

Predicting the strain of the next crack was more successful than predicting the next location of the crack. The additional information of the strain at which the next crack occurs does not significantly affect the top-5 accuracy. Throughout training, the top-5 accuracy never exceeded 10% and was close to 5%. With 100 bins and a top-5 accuracy, this is hardly an improvement compared to a blind guess of the next location. There are believed to be two main causes for this: the problem definition and the size of the data set.

Table 6.5: Model performance of the coupled and decoupled models.

Model	Loss	MSE	Location loss	Top-5 accuracy
Coupled	4.6021	0.0049	4.5972	0.0743
Decoupled - Strain	0.0031	0.0031	-	-
Decoupled - Location	4.5977	-	4.5977	0.0531

Classification of the next crack locations into 100 bins may be too complex for the network to properly learn. Nonetheless, using a lower amount of bins showed a lower accuracy of the model metrics. Furthermore, lowering the amount of bins increases the discretization width, whereas with 1 mm it is already larger than the smallest observed crack spacing. As such, details will be lost for lowering the number of bins, especially regarding dependent cracks that initiate close to an existing crack. Changing the problem to a regression problem could potentially improve results and would allow to combine the crack load and location into a single layer.

Alternatively, the underlying relationship between crack locations may simply be too complex to grasp for the network with the current data set. This could again be the result of the data set not being large enough or the relationship between the crack locations being too complex to accurately model. Furthermore, as can be seen in Figure 6.7, the weights for each bin are still close to 1%, which means that, when assuming there should indeed be some detectable pattern for the location of cracks, the model did not learn sufficiently. With a larger training set, it is expected that the weight will be able to form a more significant shape in prediction, i.e. providing a less uniform prediction than as it is now: around 1% per bin.

In summary, it is argued that the underlying causes for the location predictions to be off are most likely in both the problem definition and the size of the data set. The relationship between crack locations sure is complex, but a higher accuracy than 7.43% should be able to be obtained with current level of technology. Luckily, predicting the strain was more successful. It is shown that the decoupled model is able to obtain a MSE of 0.0031, which is almost 40% lower compared to the coupled model. The strain plot highlights the non-linear relationship between the strain and crack number, confirming the need for a model that is capable of addressing such complex relationships.

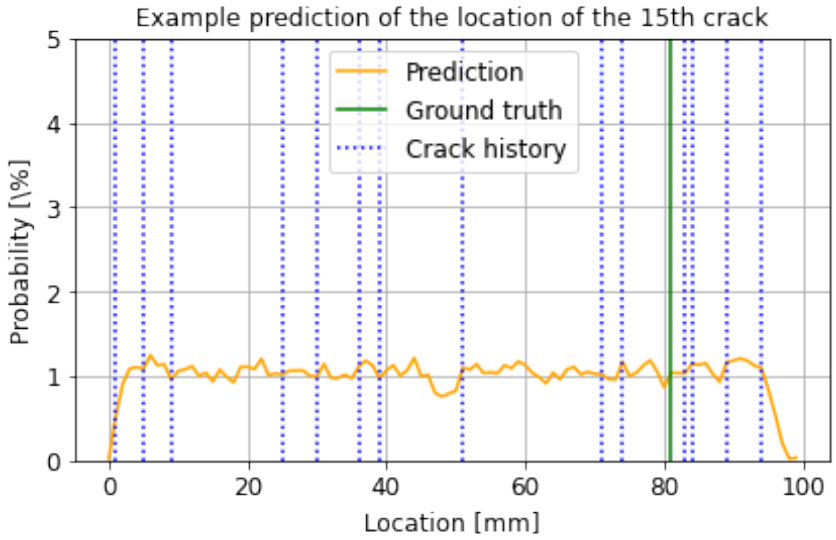

Figure 6.7: Example prediction of the location of the 15th crack using the coupled model.

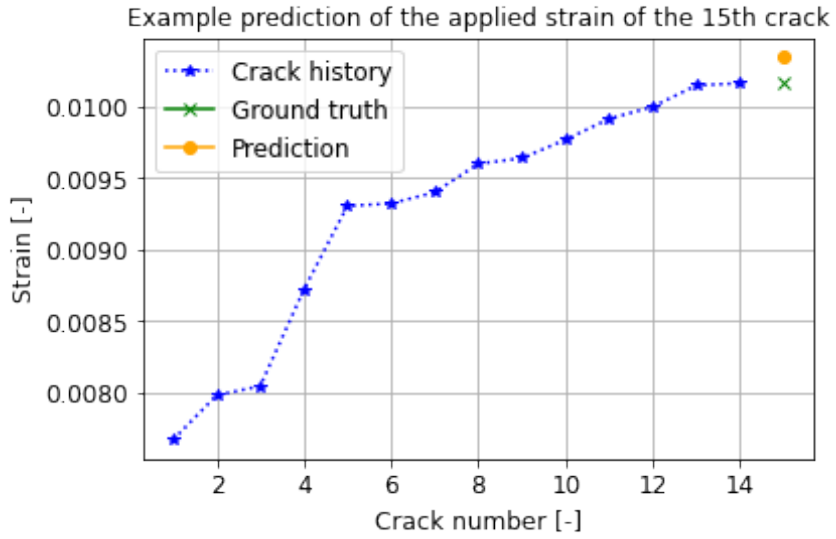

Figure 6.8: Example prediction of the applied strain that causes the 15th crack using the coupled model.

6.6.2. Error Evolution

The error evolution of the prediction of the strain for the next crack is presented here for the decoupled network, because of its 40% lower MSE than the coupled model. As explained in section 6.2, each pattern in the testing database (4 experimental patterns, 27 FEM patterns) was split into more patterns in the same way as was done to the training and validation set, although the testing set was not further augmented via mirroring and shifting of the locations.

The errors were calculated per crack number, from which both a mean value and a standard deviation was found. The mean values and standard deviations are plotted in Figure 6.9, with the number of samples in the test set that was used to determine the error and standard deviation plotted in Figure 6.10. The number of samples with a crack history length of more than 20 cracks keeps decreasing after 20 cracks, which is why the error increases again thereafter. To provide a more intuitive judgment of the errors in the same unit as the strength values, the strain was converted to stress. Non-linearities in stress-strain behavior of the specimen during damage accumulation are ignored for simplicity: the strain is multiplied by a Young's modulus of 52.8 GPa, a typical value for the specimens.

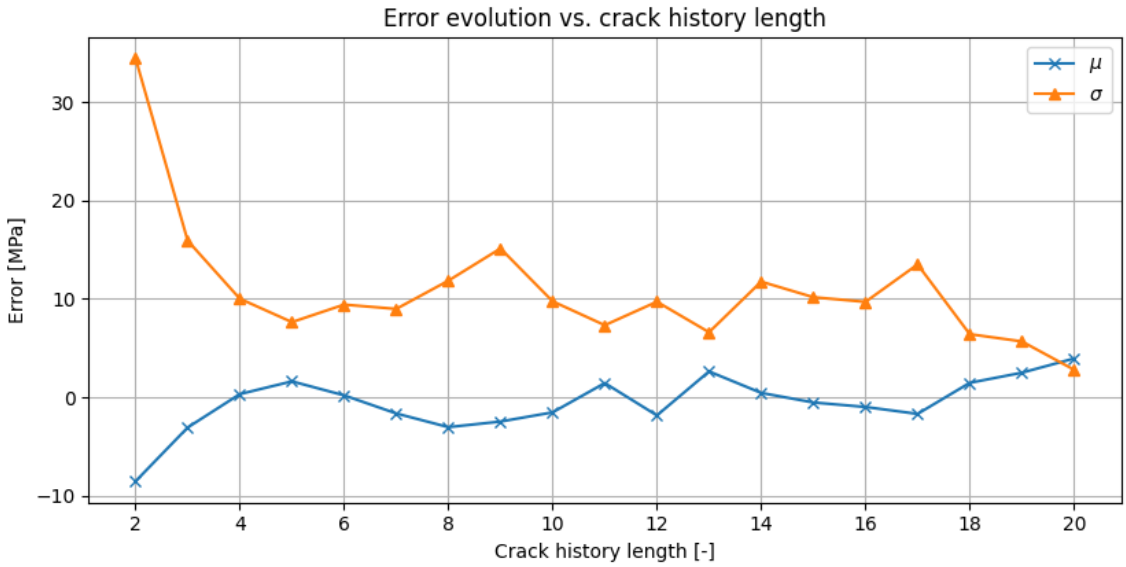

Figure 6.9: Error evolution as a function of the predicted crack number.

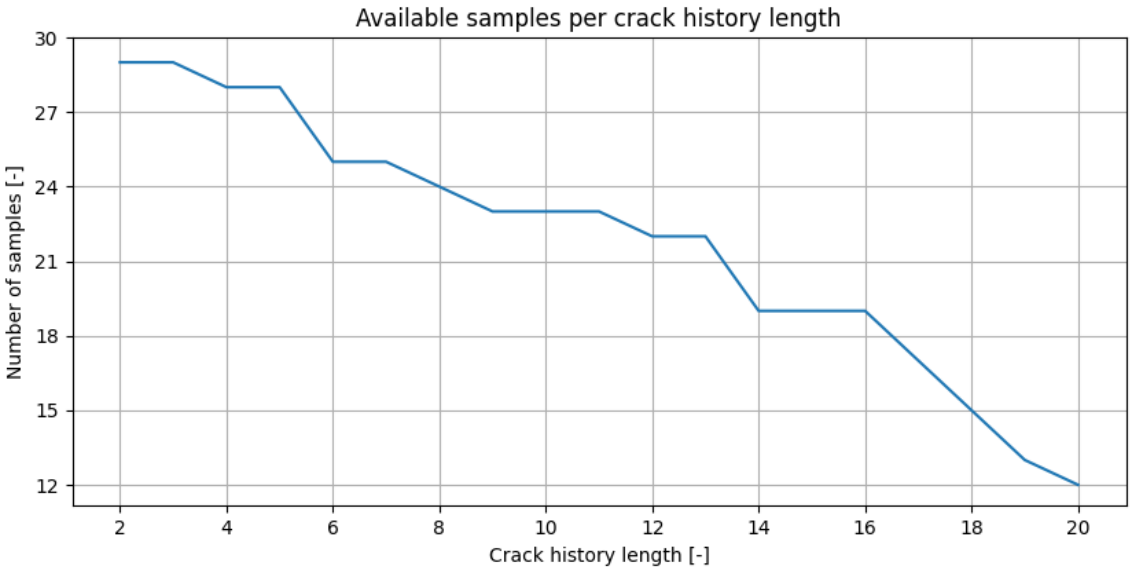

Figure 6.10: The available number of samples per crack history length within the test set.

From Figure 6.9, it becomes clear that the mean value of the error and standard deviation decrease with an increasing crack history length. It shows that from a crack history length of 5 cracks, the mean error and standard deviation remain within the same region of values. As such, it confirms the hypothesis that the first cracks are random, and from a certain point on, it gets more predictable. This suggests that the stochasticity steadily decreases until about 5 cracks, from which a more reliable prediction can be made. It can be seen that the curve is not entirely smooth, which is explained by the decrease in available samples as a higher crack number is predicted. More testing data should improve the accuracy of the error evolution and smoothness of the curves. Ideally, more testing data is obtained from more experiments, to further validate the model.

Multiplying the standard deviation by 1.96 provides a 95% confidence interval. Taking for example the error at a crack history of 6: the mean value of the errors is 0.16 MPa and the standard deviation is 9 MPa. It means that a 95% confidence bound would be ±17.6 MPa (±333 $\mu\varepsilon$ or 0.0333 mm for the 100 specimen) around the mean error of predicting the load of the 7th crack. Intuitively, that seems like a relevant number to a user in terms of damage tolerance of the structure. Figure 6.11 illustrates three different confidence interval bounds: 68%, 95%, and 99.7%. The bounds were calculated by

multiplying the standard deviation with 1, 1.96, and 3 respectively and summing and subtracting those values from the mean value. The limit of what is acceptable highly depends on the use case and in what loading stage the specimen is in: for example, ±50 MPa might be acceptable when the load is quite low, but unacceptable in case the specimen is loaded close to its strength.

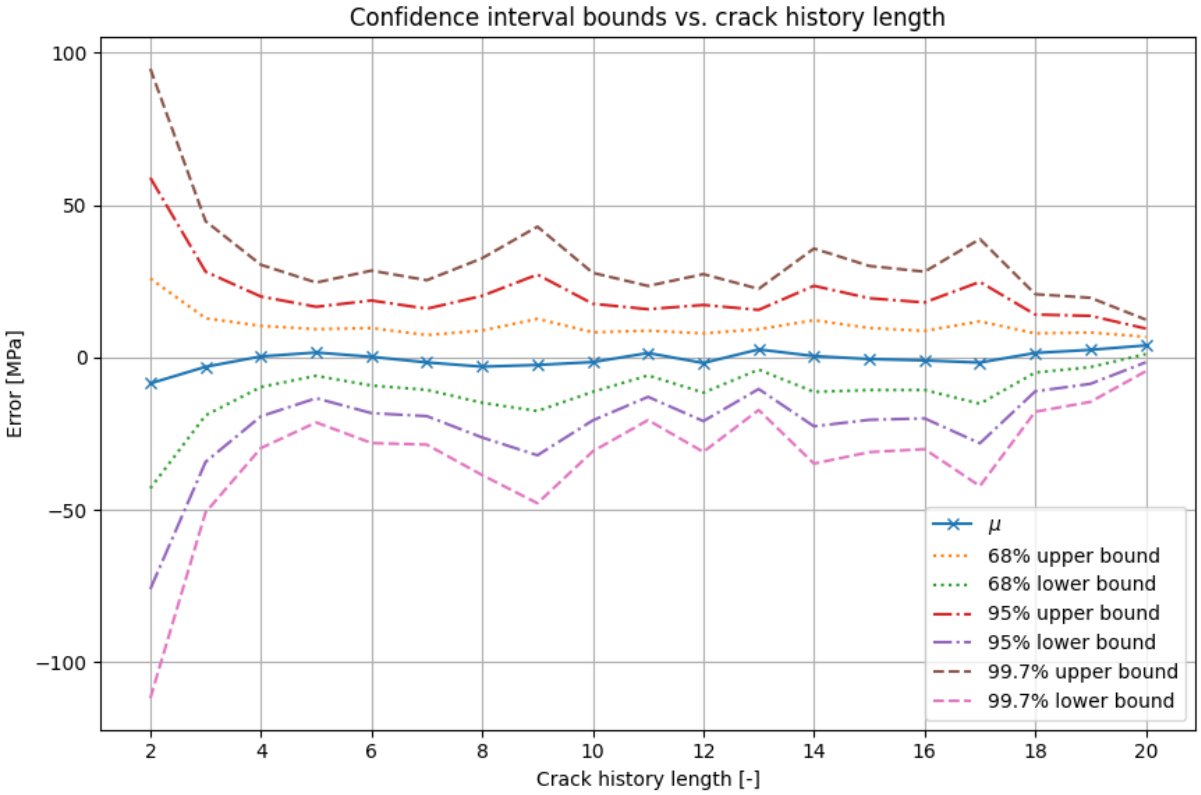

Figure 6.11: Confidence interval bounds of predicting the next crack load per crack history length.

6.6.3. Prediction Horizon of Strain

With the crack locations being too stochastic to extract a pattern from, the prediction horizon was set up for solely the prediction of the strains. Based on the results and discussion in the previous section, it was opted to establish a prediction horizon from the 5th crack onward up to 5 cracks ahead. Due to the improved results in predicting the strain, the decoupled strain prediction model was used. As discussed in subsection 6.5.3, four additional models were set up which are analyzed first before presenting their performance based on a 5-long crack history.

Table 6.6 shows the total losses for each of the models that was trained. It should be noted that there is some variability in all of the models. Training a model with the exact same parameters results in a slight variation each time, which is mainly caused by the random manner of initializing the weights within each layer and the random selection of validation data via the Keras function. Nonetheless, more global trends have shown to be fairly constant throughout training, such as the trend discussed around Figure 6.12.

Table 6.6: Losses of the models that were trained to determine the prediction horizon.

Ref.	Prediction horizon	Loss (MSE)
P1	1	0.0031
P2	2	0.00098
P3	3	0.0011
P4	4	0.0013
P5	5	0.0016

To further analyze the different models, the mean error and its standard deviation in predicting a specific crack number into the future was compared. For example, model P2 is trained to predict the next two cracks based on a certain crack history and therefore predicts 'crack 1' and 'crack 2' into the future. Those individual predictions are associated with a certain error and standard deviation of the error. Predictions were made based on the first 2-20 cracks in a crack history, so 19 predictions per model per crack number.

Table 6.7 shows the average mean error and standard deviation in predicting a specific crack number into the future for each of the models that was trained. Figure 6.12 shows a plot on the average mean value and standard deviation of the errors per crack number as a function of the prediction horizon.

Table 6.7: Average mean value and standard deviation error in **MPa** per predicted crack for the models that were trained to determine the prediction horizon. 'Crack 1' indicates the average error and standard deviation of each of the models to predict the first crack ahead, 'Crack 2' indicates the average error and standard deviation to predict the second crack ahead, and so on.

Ref.	Crack 1		Crack 2		Crack 3		Crack 4		Crack 5	
	Error	σ	Error	σ	Error	σ	Error	σ	Error	σ
P1	-0.54	10.89	-	-	-	-	-	-	-	-
P2	-1.67	11.48	-1.88	16.95	-	-	-	-	-	-
P3	-0.74	11.32	-1.06	16.15	-0.87	20.50	-	-	-	-
P4	-1.45	11.33	-2.11	16.23	-1.76	19.93	-1.76	23.07	-	-
P5	-1.45	11.53	-2.16	16.49	-2.53	20.18	-2.31	22.70	-3.26	25.94
Average	-1.17	11.31	-1.80	16.45	-1.72	20.20	-2.03	22.89	-3.26	25.94

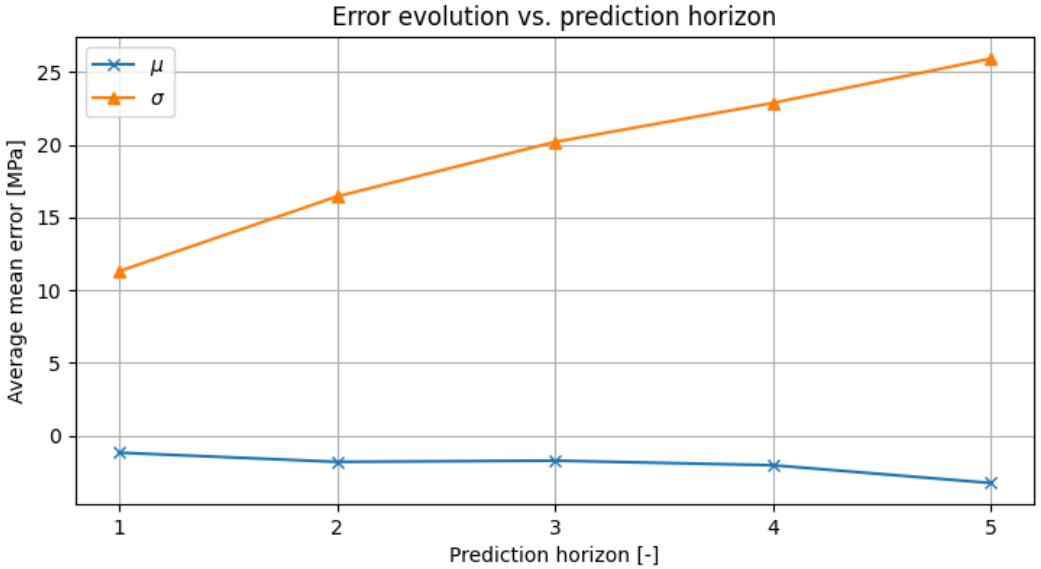

Figure 6.12: Error evolution for a prediction horizon of up to 5 cracks ahead into the future, based on the average mean errors and corresponding standard deviations from Table 6.7.

A clear trend is observed in Table 6.7 and highlighted in Figure 6.12: cracks further ahead into the future are predicted with a larger (absolute) error and a higher standard deviation, which makes sense intuitively. It should be noted here that the absolute value of the average mean error is not a proper evaluation metric on its own. After all, an average error of 0 MPa might as well be achieved with luck and a standard deviation of 200 MPa, which, obviously, is an indicator for a very unreliable model.

The average mean error seems to deviate further from 0 as the prediction horizon is increased, albeit that it is still relatively close to 0. On the other hand, the standard deviation is seemingly growing linearly. As such, the associated confidence intervals are also growing, as is shown in Figure 6.13. Here, the confidence bounds were calculated in the same way as for Figure 6.11.

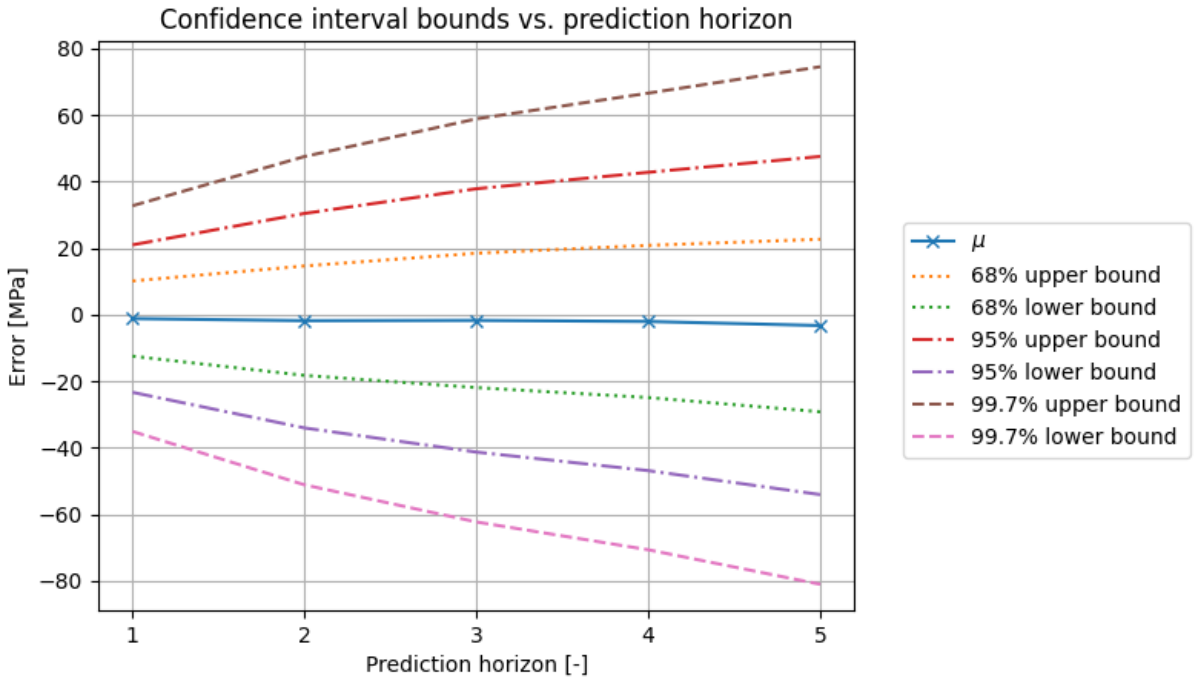

Figure 6.13: Confidence interval bounds of prediction horizon, based on the error evolution of Figure 6.12.

Figure 6.14 and Figure 6.15 provide more nuance in the discussion on the prediction horizon, where all models listed in Table 6.6 were used to predict a crack. It should be noted that generating this plot with model P5, which predicts the next 5 cracks, results in very similar plots. From Figure 6.14, it becomes clear that the error is very similar for all prediction horizons. Only the error in predicting the fifth crack has quite a drastic jump at a crack history length of 16, where the others remain more alike. This is attributed to the decrease in available number of training samples for longer crack patterns, meaning the model has seen less of longer crack patterns.

Figure 6.15 shows how the standard deviation, thereby the confidence interval and reliability of the prediction, steadily increases as the prediction horizon increases. Up until about 16-18 cracks, the standard deviation is staying within the same range of values, after which it starts to deviate from the trend. This is partially attributed to the decrease in available number of training samples as well. Moreover, the decrease in available testing samples for longer patterns means a decrease in number of errors. The standard deviation is a function of the number of errors that was used to calculate it, because a lower number of values means less reliability in the distribution that is aimed to be mapped.

After discussing the overall trends of the prediction horizon, experimental data of four specimens was used as test subject. Five models were generated, which showed similar performance in predicting certain crack numbers with a minor variation in error evolution, as illustrated in Table 6.7. Ideally, a certain crack is predicted with a model that is optimized for that specific crack, but to keep the number of plots and prevent the plotting of 5 overlapping lines, predictions plots were generated by using the model with a prediction horizon of 5 cracks.

The prediction horizon will be based on the performance of predicting a number of cracks into the future based on a crack history of 5 cracks. Experimental data of four specimens was used for each of the five models. The experimental data consists of the two specimens that were tested under an increasing loading rate from 0 to 0.19 kN and the two specimens that were tested under an increasing loading rate from 0 to 19 kN (the blue, orange, green, and red curves that are showing as the first four curves in legend in Figure 4.2).

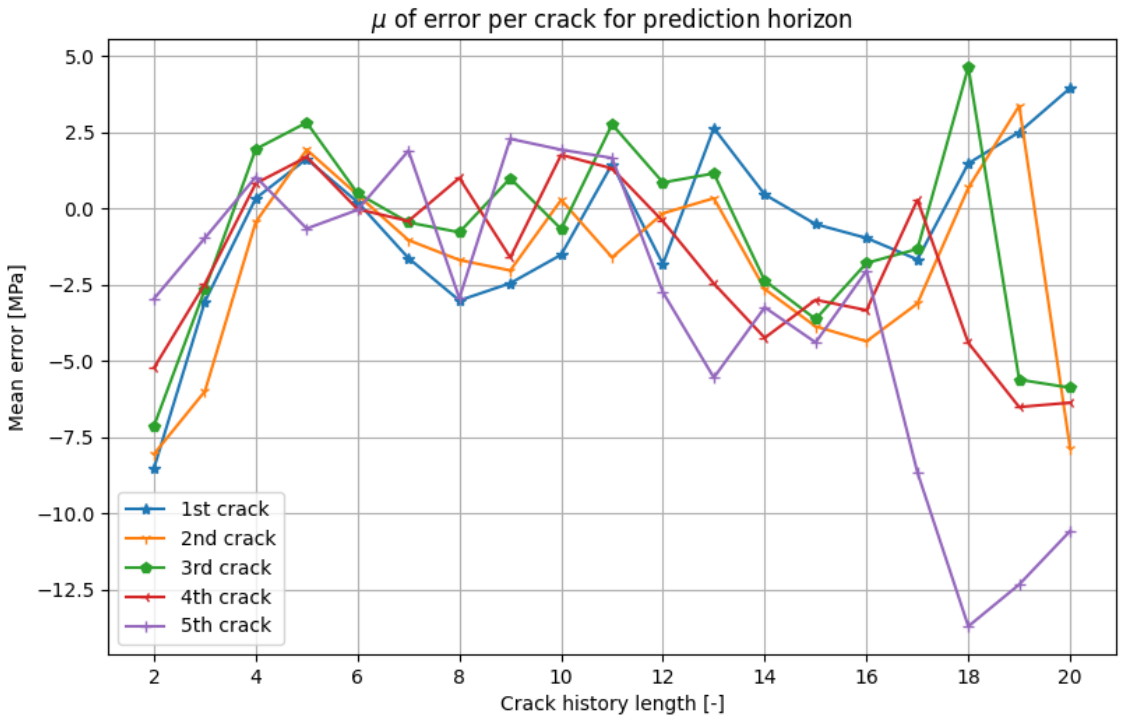

Figure 6.14: Mean error vs. crack history length for a prediction horizon of 1, 2, 3, 4, or 5 cracks. Each line was generated using the corresponding model as listed in Table 6.6.

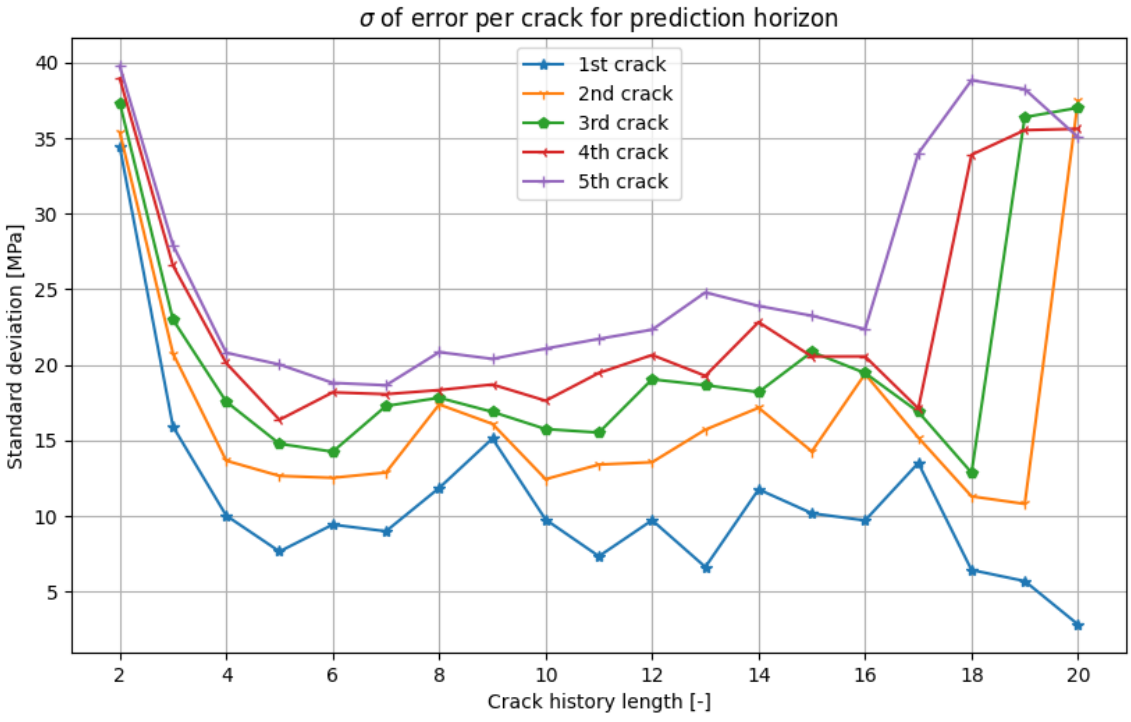

Figure 6.15: Standard deviation vs. crack history length for a prediction horizon of 1, 2, 3, 4, or 5 cracks. Each line was generated using the corresponding model as listed in Table 6.6.

Figure 6.16 shows the predictions for both specimens under the 0-19 kN load based on a crack history of 4 cracks (t_1) and 5 cracks (t_2) and Figure 6.17 shows the predictions for the specimens under the 0-0.19 kN load. Comparing both figures suggests that the specimens under the higher loading rate are more accurately predicted. That makes sense, considering that the loading rate in the FEM models was also rather high, as discussed in subsection 5.2.5.

Zooming in further on the individual plots in Figure 6.16 and Figure 6.17, three things are noted: 1) overall, the first crack is more accurately predicted than the other cracks, which is in line with what was discussed earlier in this section; 2) within the set of predictions, there seems to be cracks that are predicted at a lower load than its predecessors, such as for the plot of specimen 0-0.019 kN #2 at t_2 in the bottom right of Figure 6.17; and 3) several predictions, including the first appearing crack, are lower than the applied the stress.

The last two phenomena are interlinked but not the same, however, it is hypothesized that both phenomena are solved by the same fix: an increase in the size of the data set. Similar to the cause of difficulties in predicting the locations, this is believed to be caused by the weights having not enough data to be properly trained. Additional training data will result in the model recognizing that each prediction load should at least be higher than the latest prediction.

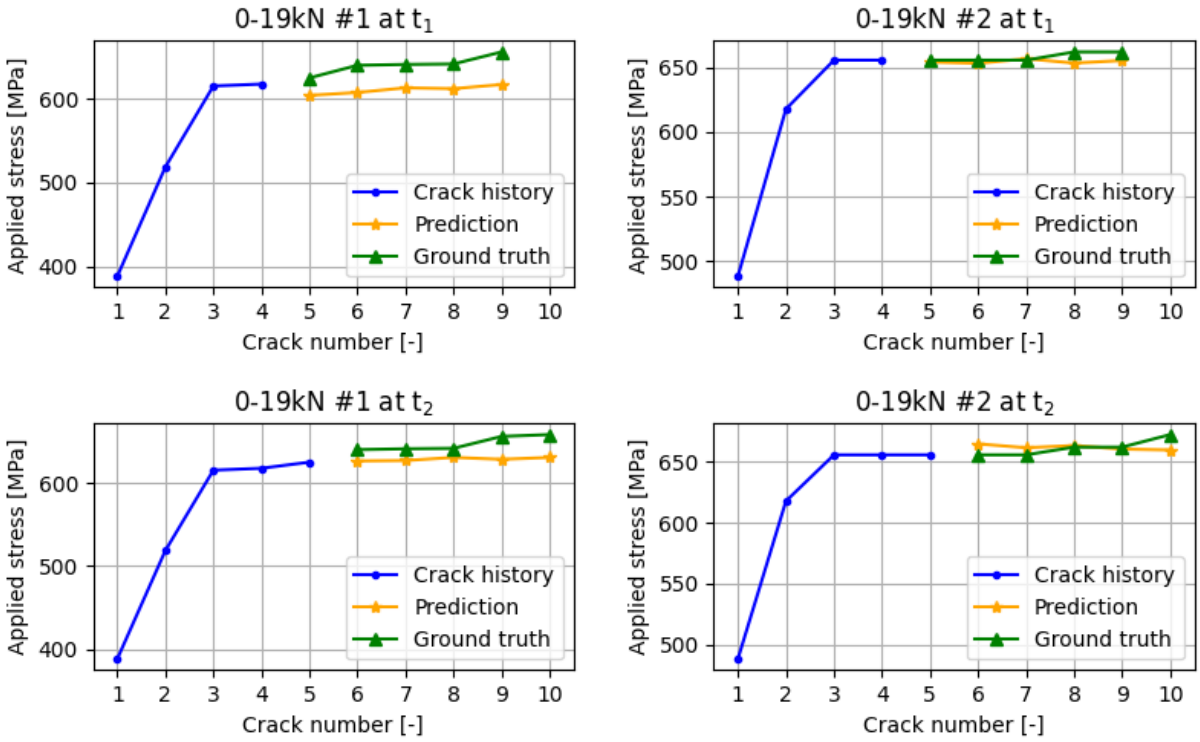

Figure 6.16: Predictions of the next crack load, where the strain was converted to stress again by multiplying the strain values with a typical modulus of 52.8 GPa. The two specimens were loaded under an increasing load of 0-19 kN.

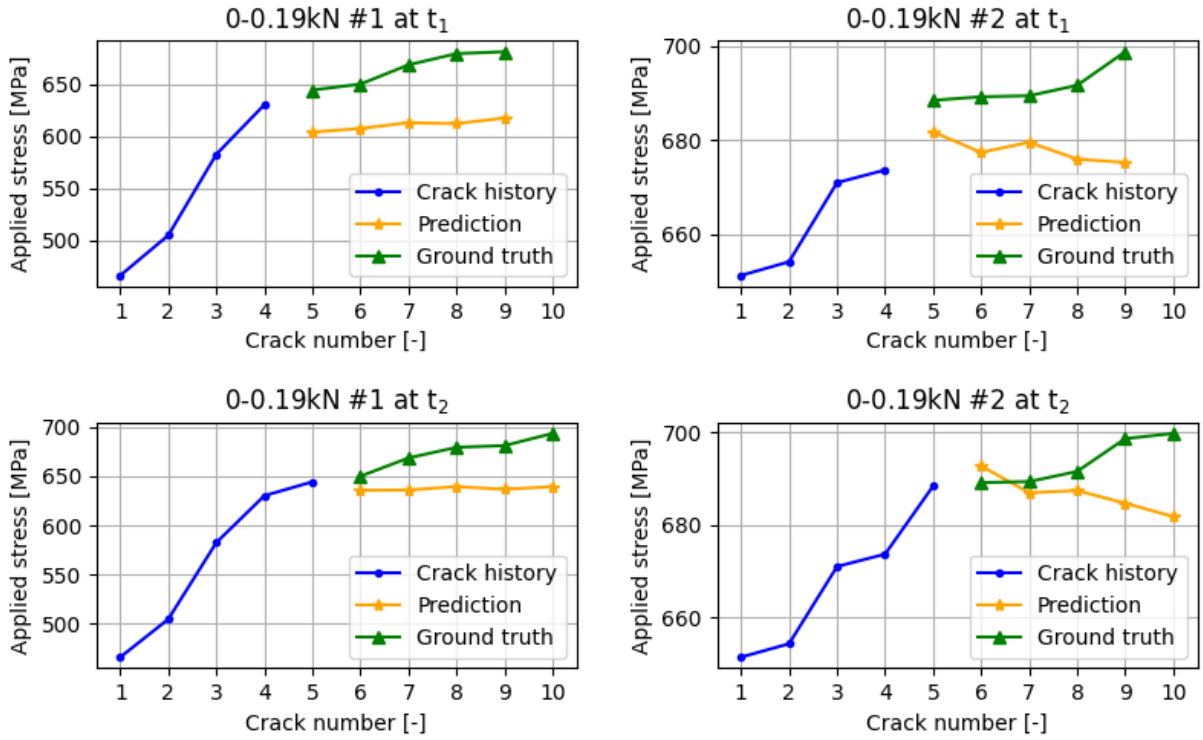

Figure 6.17: Predictions of the next crack load, where the strain was converted to stress again by multiplying the strain values with a typical modulus of 52.8 GPa. The two specimens were loaded under an increasing load of 0-0.019 kN.

6.7. Helicopter View Digital Twin

After detailed discussions on the finite element model and neural network, this section zooms out to place the digital twin into context again. Figure 6.18 illustrates the proposed digital twin in the environment in which it operates. The dotted lines indicate one of the main interfaces: the user, the physical twin, and the digital twin. The user controls the mechanical environment which affects the physical twin.

Damage diagnosis was not covered in this thesis but is performed via acoustic emission sensors and digital image correlation monitoring on the physical twin. In case of a new crack, the damage state of the digital twin is updated, i.e. the neural network receives a new location and load of a crack. In case the digital twin is operable, new cracks could be saved in the operational history and the network could learn online, i.e. 'on the fly'. The updated damage state is fed into the network which outputs a new prediction of a crack location and loading. The latest prediction is passed to the user, which could use this information to, for example, adjust the loading conditions.

The digital twin as proposed seems to provide a stepping stone for further research: it is proven that without many costly experiments, a digital twin can be established that is able to provide the user relevant information on its damage state via integration of heterogeneous data sources in real-time. Predictions on damage accumulation can be used in a later stage to take the next step in developing a digital twin of a composite specimen. For example, the number of cracks may form, together with other features such as a strain field, the basis on accurate modeling and prediction of stiffness degradation as damage progresses.

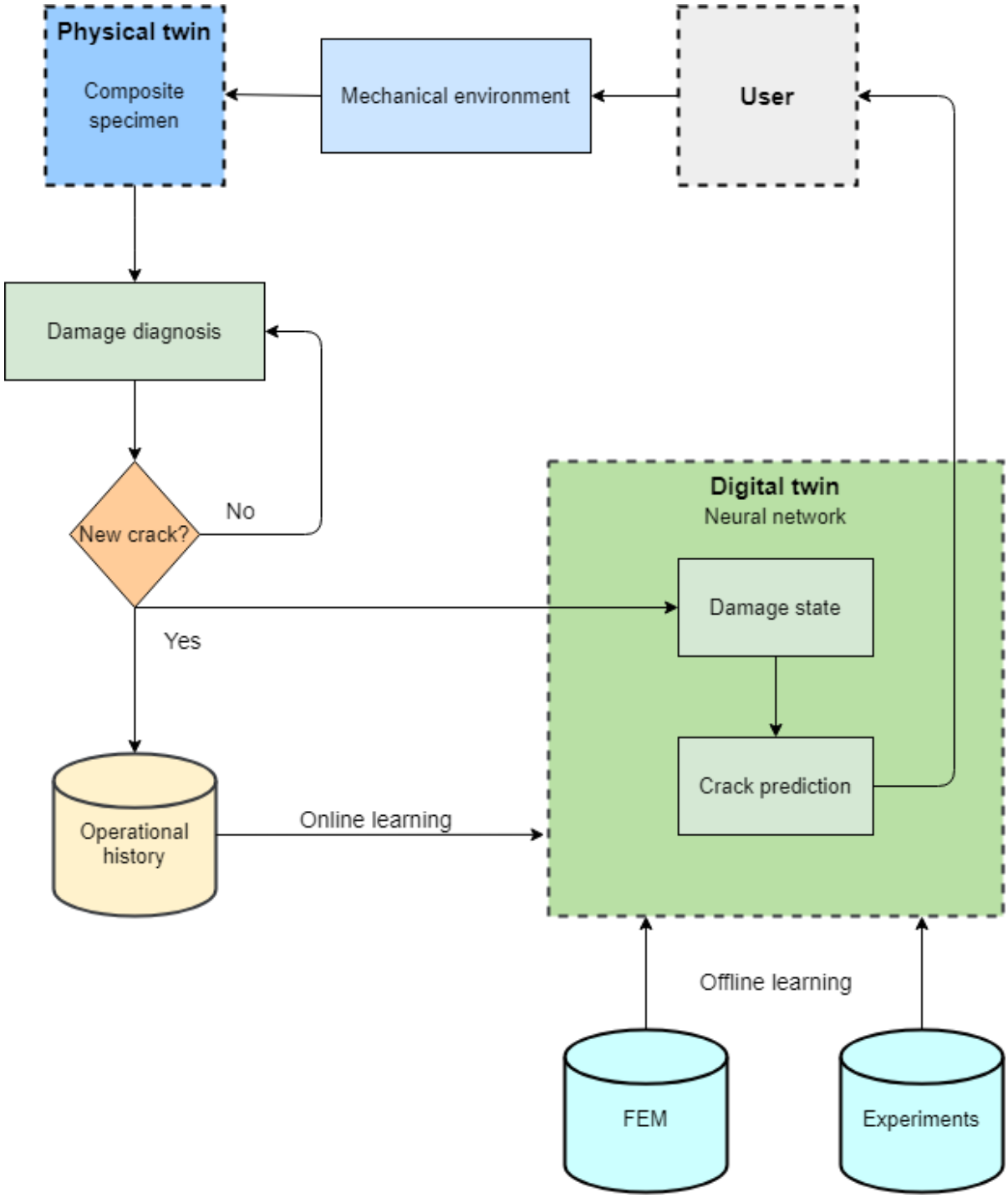

Figure 6.18: Environment in which the digital twin operates.

6.8. Conclusion and Recommendations

The data-driven digital twin was proposed in the form of an artificial neural network made out of LSTM layers, because it is the benchmark in the prediction of sequences within the field of deep learning. Using LSTM layers allows to include the crack history of a pattern as input for a prediction, thereby including a time component. The basis of the data set was formed by experimental data as described in chapter 4 and augmented by the FEM model described in chapter 5. Further augmentation was performed by shifting locations of the data set all by the same amount, to maintain physical relationship between them.

To come up with an architecture, both the number of LSTM layers and nodes per layer were varied. Overfitting was a prominent issue, which is why out of all tested architectures, the architecture with relatively few degrees of freedom performed best. Performance was measured in mean squared error for predicting the next crack load and a top-5 accuracy metric was adopted to measure the accuracy of predicting the next crack location.

The crack location was found difficult to predict, which was attributed to the problem definition, classifying the next crack out of 100 classes, and the size of the data set, it is hypothesized that the relationship between crack locations is too complex to be modelled with the current data set size.

The strains at which the next crack appears were predicted more successfully. To introduce more intuitive numbers, strains were converted to stress by assuming linear elasticity for simplicity and a Young's modulus of 52.8 GPa. After approximately 5 cracks, the average error of close to 30 predictions of different specimens is close to 0 with a standard deviation of the errors of around 10 MPa. It confirms the initial hypothesis that the first few cracks in the crack pattern are random and from a certain point onwards, it gets predictable.

Increasing the prediction horizon for predicting the next crack load up to 5 cracks ahead shows that the error steadily increases when more cracks ahead are predicted. The average mean error and standard deviation for predicting 1 crack ahead by using a crack history length of 2-20 are -1.2 MPa and 11.3 MPa respectively, which becomes -3.3 MPa and 25.9 MPa at predicting the 5th crack ahead: an increase of 229%. It underlines the fact that improvements in the model are necessary to be able to increase the prediction horizon without severe loss in accuracy.

All in all, it was found that the digital twin as proposed is able to predict the crack pattern reasonably well in terms of at what load the next crack is going to appear. It seems a suitable stepping stone to take the next steps in the development of a digital twin of a composite specimen. The following bullet points are recommendations for this further research:

- Data set
  - The data set should be expanded both in terms of experimental data and FEM data to improve the performance. In case a split was made between experimental data and FEM data to perform model evaluation, only four experimental patterns were used to evaluate and four experimental patterns were used to train the model. Especially when discrepancies between the FEM data and experimental data exist, such as the bias between crack density curves as discussed in subsection 5.4.3, a higher ratio of experimental data over FEM data is expected to improve performance of the model.
- Feature selection
  - The crack load could be linked to the mechanical state of the specimen, for example the stiffness. As such, a more exhaustive digital twin could be constructed. It also provides the user with additional relevant information. The stiffness of the specimen could form the basis of a remaining useful life prediction.
  - By including more features into the model, it could be extended to predict damage initiation. A relatively simple manner to predict damage initiation could be to see if the global stiffness evolution of a specimen can be monitored and inputted to provide for a prediction of damage initiation. Alternatively, if the strain fields of the model could be mapped in

real-time, that information could be used as additional features to provide inputs to predict damage initiation. Cristiani et al. provide an example where the strain field was used as input to an ANN surrogate model [87], meaning

- Problem definition
  - The model that was built assumes the appearance of cracks to be a time-dependent problem. It would be interesting to see how for example a neural network built out of non-RNN based layers would perform. In that case, one would input simply the current crack locations and applied load based on which it predicts the next crack location and load, without taking into account the history of evolution.
  - In case a larger data set can be obtained, it could be investigated what parameters can be monitored in real-time and be added as input features, for example by including the (degraded) stiffness state.
- Loss fuction
  - The loss function that was used for regression is the mean squared error, which 'punishes' severe errors more than small errors. An improvement for the cases presented in Figure 6.17 may be accomplished via a different loss function, such as the mean absolute error loss function.
- Pruning
  - A different approach in constructing the network that could improve results by preventing overfitting is presented by You et al. [149]. The authors propose to take a set of randomly-initialized networks from which subsets may match or even exceed test accuracy of the original network: 'winning tickets'. Identification of these winning tickets involves pruning, which is a process of setting connections to unimportant weights to zero, thereby eliminating the neuron. Generally, it is costly to first train a fully dense network and start pruning afterwards. You et al. present a technique called 'early-bird tickets', which allows to draw winning tickets earlier in the training process with low-cost training algorithms.
- Black box effect
  - Whether or not the "black box" nature of deep learning is worth it is partially dependent on the allowable time for a prediction: in case an abrupt and critical load is applied to a structure, one wants to know about it as soon as possible to act accordingly to save the structure, whereas if it is a matter of failure after 10,000+ cycles under a fatigue loading, it is less of a problem if the prediction time is more than a couple seconds.
- Uncertainties
  - Proper representation, quantification, and propagation of uncertainties is critical when it comes to accurary in predictions. Two major types of uncertainties exist: aleatoric uncertainties — statistical uncertainties caused by inherently random effects, such as flipping a coin, which is always influenced by randomness — and epistemic uncertainties — systematic uncertainties caused by the used model, which may be caused by erroneous settings or a general lack in knowledge. Both types of uncertainties are addressed in an extensive paper by Hüllermeier and Waegeman, with many ways to address uncertainties [150]. One particular model is encouraged to look deeper into to handle uncertainties within the context of this thesis: a Bayesian neural network.
  - The nature of the network could be changed to a Bayesian Neural Network, where weights are substituted for probabilities. A similar type of outcome is outputted — it outputs a posterior distribution of the prediction, (typically Gaussian, but non-Gaussian is possible as well) rather than a single value — but now with possibly an improved result. As described in the problem definition of classification of the locations, such a posterior distribution would have three advantages: 1) it provides the user with more nuance in the prediction to make a better decision in case the digital twin is operative; 2) accurately predicting posterior distribution may reveal the complex underlying patterns between for example

the location of the next crack; 3) it allows to capture epistemic uncertainties revolving the weights of a 'normal' neural network.

- Verification & Validation
  - The validation set that was used during draining to tune the hyperparameters was split automatically from the training data that was inputted, under the assumption that after shuffling the training data, the samples are distributed in an unbiased manner. Therefore, the performance of the model could vary once another part of the training set is assigned as validation set. Doing such an analysis where the entire training set is 'folded' k times is known as k-fold cross validation. It comes down to assigning a different part of the training set as validation set while retraining the model several time, until all samples have been part of the validation set once. If the set is split into 10 different validation sets, it would be 10-fold cross validation. As such, the metrics across iterations could be analyzed via an average, range, standard deviation, and so on. It allows more accurate verification of the model performance.
  - The model could be trained on a different data set that is already validated to assess the quality of the current data set. For example, it could be trained with a homogeneous data set coming from either experiments or a synthetic source of which underlying patterns are already known and verified. In doing so, the root cause of encountered problems could be attributed to the data set if it turns out the problems disappear once the data set is changed.
  - To assess the accuracy of the model, it could be compared to alternative predictive methods. One of which would be to implement a certain crack pattern into a FEM model using pre-cracked elements. Alternatively, the type of problem could be changed. It was found most difficult to predict the next location rather than the strain at which the next load occurs and it was approached as a classification problem. To assess the performance of the architecture, the input and output of the location could be converted to numerical values, changing the problem into a regression problem. As such, the effect of problem definition on the model performance can be further assessed.

7
Conclusion

Improvement in the SHM of composite materials requires an enhanced understanding of the damage accumulation processes and helps in the way towards lighter, more optimized, and more sustainable aerospace structures. A digital twin can serve as a tool to solve the problem of modelling complex damage accumulation processes with the final purpose of structural health monitoring or validation of other structural health monitoring techniques. By mirroring the current state of its physical twin in real-time, it provides a fast manner of processing new information by integration of heterogeneous information sources.

In this thesis, a hybrid approach is proposed by combining experimental data and physics-based finite element simulations as a basis for data-driven predictions. A digital twin is presented as a data-driven approach in the form of a recurrent neural network with LSTM cells. The physical twin is a cross-ply specimen for which experimental data under various quasi-static tensile load rates is available. The finite element model that was built was used to augment the experimental data set. The initial hypotheses were fulfilled, as it was confirmed that the first few cracks were random but after a certain point, approximately 5 cracks, the error decreased. Furthermore, fast and successful integration of the heterogeneous data was proven to be capable with the recurrent neural network with LSTM cells. As such, a methodology is presented that overcomes the need for expensive testing to make use of the benefits of a data-driven approach such as machine learning.

This thesis aimed to answer the following main research question:

> *To what extent can the crack pattern of transverse matrix cracks in a cross-ply be predicted using a digital twin?*

The short answer is that with the proposed approach, the crack pattern can be predicted reasonably well. The location turned out to be too stochastic to predict, whereas the strain at which the next crack appears was predicted more successfully. A more detailed answer is found via answering the sub-questions:

1. *How can the material variability that is inherent to composite specimens be modelled into a FEM model?*
   Material variability was successfully implemented in the finite element model by partitioning the model parts into the desired element shape and assigning a separate section with each element having its own material. Material properties were defined in distributions, from which a different set of properties was drawn per element. Naturally, the model is sensitive to the adopted material distributions. The transverse matrix strength has the most dominant effect on the progression of transverse matrix cracks, which also was the only distribution that was experimentally observed. A normal distribution was adopted for the other material properties by taking the mean values from the data sheet and setting the coefficients of variance to be small (2%), taking into account that the used prepregs generally hold a higher quality than e.g. hand lay-ups. The distributions showed to result in various damage evolution patterns, which, overall, match experimental observations. The main difference was found to be in the initiation strain of the first matrix crack and the initiation strain of the last matrix crack; the ones of the finite element simulations seemed to have the same shape but shifted to lower loads, possibly caused by a discrepancy in the Weibull distribution.

Results are expected to be further improved by implementing a correlation between the distributions of parameters. One option to do so would be to link the finite element model to a micromechanics model. Furthermore, more distributions of material properties than the transverse strength Y_T are expected to advance the model.

2. *To what extent can XFEM-CE simulate the interaction between transverse matrix cracks and delaminations in a carbon-epoxy cross-ply specimen under tensile quasi-static loading?*
Based on comparing the results of the finite element model with experimental observations, XFEM-CE seems to be able to capture the evolution of transverse matrix cracks well, even though most damage parameters were derived from trial and error within ranges that were established based on other carbon-epoxy specimens. Both transverse matrix cracks and fiber failure were modelled using enriched elements and delamination was modelled using cohesive based surfaces. However, introducing an enrichment spacing was required to facilitate convergence. Different spacings show similar evolution curves, with the main difference in the point of crack initiation: lower spacings showed earlier crack initiation caused by the inclusion of lower strength values from columns that otherwise would not have been enriched, i.e. allowed to fail. Convergence issues were encountered that are believed to be caused by troubles of the model in damage evolution in the longitudinal plies. It resulted in a significant amount of the specimens in the augmented data set to have crack patterns that do not run until failure of fibers, but stop before it already. Nonetheless, patterns resulting from prematurely aborted simulations were included in the data set, because those essentially include all cracks except for the last one until failure. Lastly, no validation on delamination was done, other than assessing its effect on the evolution of transverse matrix cracks, which was the focus of this thesis. Setting the penalty stiffness and fracture energy of the cohesive surfaces too high prevented the required stress relaxation around a matrix crack, which causes premature failure of surrounding elements.

   To improve of the model's overall performance and generalization, several adjustments are proposed. Firstly, the failure initiation criterion of maximum principal stress could be changed to a more sophisticated one, such that the shear stresses in the region of the interface between the longitudinal plies and transverse plies are taken into account more accurately. Furthermore, via a UMAT subroutine, one could implement fatigue loading to assess the effect of material variability on the predicted remaining useful life and stiffness degradation to build a digital twin that focuses on the fatigue life. The main step that needs to be taken to implement fatigue here would be to successfully implement a fatigue model in the Abaqus model; experimental fatigue data of the same specimen already exists for which the same experimental setups and researchers as explained in chapter 4. Lastly, extending the 2D model to 3D would unlocks two things: 1) validation of the 2D model, 2) implementation of other layups than cross-plies, because with the current definitions, only 0° or 90° can be modelled.

3. *How well can a neural network be trained with an augmented data set to predict the location and load of the next crack of a carbon-epoxy cross-ply under quasi-static tensile loading?*
The data-driven digital twin was proposed in the form of a recurrent neural network with LSTM cells. The crack location was found difficult to predict, which was attributed to the problem definition, classifying the next crack out of 100 classes, and the size of the data set, the relationship is probably too complex to be modelled with the current size. The strains at which the next crack appears were predicted more successfully. After approximately 5 cracks, the mean error of close to 30 predictions of different specimens is near 0 MPa with a standard deviation in errors of around 10 MPa. Increasing the prediction horizon to 5 cracks into the future hardly affects the mean error but increases the average standard deviation (taken over the use of crack histories ranging from 2-20 cracks) by 229% to 25.9 MPa. Thus, it was concluded that improvements in the model are necessary to be able to increase the prediction horizon without severe loss in accuracy. All in all, it was found that the digital twin as proposed is able to predict the crack pattern reasonably well in terms of when the next crack is going to appear.

By adding data from more simulations and experiments, the performance is expected to further improve. Not only because of more training data, but also to increase the testing data set, to gain a better insight on the error evolution. The network itself could also be improved by a change in problem definition for the locations: changing it from a classification problem to a regression problem. Another method to change the problem definition would be to change the entire network type to a Bayesian neural network, where weights substituted for probabilities. An advantage of shifting to a Bayesian neural network is that epistemic uncertainties revolving the weights a 'normal' neural network are dealt with.

Placing the digital twin in a broader context, the presented methodology seems suitable for further research. Without costly experiments, a first step was taken towards a digital twin of a composite specimen that mimics its damage mechanics. The neural network showed capable of fast processing of a large amount of information from heterogeneous sources to provide a reasonable prediction of the loading at which the next crack occurs in real-time.

Using this thesis as a stepping stone, further steps could entail a larger specimen, a different layup, or a different load case to start with, as well as the above-mentioned recommendations. It would allow for the digital twin to develop and aid in further enhancement of our understanding of damage accumulation in composite materials. In conclusion, with the expectation that the current trends in availability in data and use of composite materials will continue to grow, the digital twin will, in the proposed form or another, find its way into structural health monitoring of composite structures.

Appendices

A

Classical Laminate Theory

In this appendix, the classical laminate theory (CLT) that was used for verification of the mechanical behavior of the FEM simulations (Figure 5.9 and Figure 5.10) is explained. CLT is an extension of the classical plate theory of Kirchoff for homogeneous plates [151]. Using CLT, the mechanical properties and stresses of a laminate composed of UD plies can be calculated, both per ply and of the global laminate.

The generalized stress-strain relationships or Hooke's law describe the mechanical behavior and state of deformation by relating the normal and shear stresses to the normal and shear strains by using 21 independent constants, as shown in Equation A.1.

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \\ \tau_{yz} \\ \tau_{xz} \\ \tau_{xy} \end{Bmatrix} = \begin{bmatrix} E_{11} & E_{12} & E_{13} & E_{14} & E_{15} & E_{16} \\ E_{21} & E_{22} & E_{23} & E_{24} & E_{25} & E_{26} \\ E_{31} & E_{32} & E_{33} & E_{34} & E_{35} & E_{36} \\ E_{41} & E_{42} & E_{43} & E_{44} & E_{45} & E_{46} \\ E_{51} & E_{52} & E_{53} & E_{54} & E_{55} & E_{56} \\ E_{61} & E_{62} & E_{63} & E_{64} & E_{65} & E_{66} \end{bmatrix} \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_z \\ \gamma_{yz} \\ \gamma_{xz} \\ \gamma_{xy} \end{Bmatrix} \tag{A.1}$$

The UD plies that are dealt with in this thesis can be assumed to be orthotropic, meaning it has two axes of symmetry. This assumption sets many coupling terms to zero and simplifies Equation A.1 to Equation A.2:

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \\ \tau_{yz} \\ \tau_{xz} \\ \tau_{xy} \end{Bmatrix} = \begin{bmatrix} E_{11} & E_{12} & E_{13} & 0 & 0 & 0 \\ E_{12} & E_{22} & E_{23} & 0 & 0 & 0 \\ E_{13} & E_{23} & E_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & E_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & E_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & E_{66} \end{bmatrix} \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_z \\ \gamma_{yz} \\ \gamma_{xz} \\ \gamma_{xy} \end{Bmatrix} \tag{A.2}$$

To calculate stresses within a laminate and its plies, the problem is reduced from three dimensions to two dimensions. By assuming plane stress conditions, the out-of-plane stresses are assumed to be negligible compared to the in-plane stresses. Filling this in into Equation A.2:

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ 0 \\ 0 \\ 0 \\ \tau_{xy} \end{Bmatrix} = \begin{bmatrix} E_{11} & E_{12} & E_{13} & 0 & 0 & 0 \\ E_{12} & E_{22} & E_{23} & 0 & 0 & 0 \\ E_{13} & E_{23} & E_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & E_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & E_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & E_{66} \end{bmatrix} \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_z \\ \gamma_{yz} \\ \gamma_{xz} \\ \gamma_{xy} \end{Bmatrix} \tag{A.3}$$

Hence:

$$\gamma_{yz} = \gamma_{xz} = 0 \tag{A.4}$$

As such, the system of equations is reduced to the following:

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} = \begin{bmatrix} Q_{xx} & Q_{xy} & 0 \\ Q_{xy} & Q_{yy} & 0 \\ 0 & 0 & Q_{ss} \end{bmatrix} \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} \tag{A.5}$$

where

$$\begin{aligned} Q_{xx} &= E_{11} - \frac{E_{13}^2}{E_{33}} \\ Q_{xy} &= E_{12} - \frac{E_{13}E_{23}}{E_{33}} \\ Q_{yy} &= E_{22} - \frac{E_{23}^2}{E_{33}} \\ Q_{ss} &= E_{66} \end{aligned} \tag{A.6}$$

Besides the ply thickness, based on the following five elastic constants Equation A.5 can be defined for each ply and ultimately, the entire laminate:

- E_L: Longitudinal Young's modulus
- E_T: Transverse Young's modulus
- G_{LT}: In-plane shear modulus
- ν_{LT}: In-plane major Poisson's ratio
- ν_{TL}: In-plane minor Poisson's ratio (is calculated as: $\frac{E2 \cdot \nu_{LT}}{E_L}$)

In Equation A.7, the matrix in Equation A.5 is rewritten in terms of the constants that are listed above.

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} = \begin{bmatrix} \frac{E_L}{1-\nu_{LT}\nu_{TL}} & \frac{\nu_{LT}E_T}{1-\nu_{LT}\nu_{TL}} & 0 \\ \frac{\nu_{LT}E_T}{1-\nu_{LT}\nu_{TL}} & \frac{E_T}{1-\nu_{LT}\nu_{TL}} & 0 \\ 0 & 0 & G_{LT} \end{bmatrix} \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} \tag{A.7}$$

If necessary, the obtained stress-strain relationships can be rewritten to account for rotated plies, for which the zeros in Equation A.5 and Equation A.7 become nonzero again. However, because this thesis deals with a cross-ply and in-plane longitudinal loading, the reader is referred to the book of Kassapoglou on design and analysis of composite structures for further reference on how to transform stresses and strains to a different angle [152].

Once having defined the mechanical properties of each ply by applying Equation A.7, the stresses and strains per ply are integrated through the thickness to obtain the global response. These integrations are a summation of the stresses per ply, while taking into account each ply's relative position to account for curvature effects. It is common practice to define forces and moments through integration through the thickness of the entire laminate. For example, in longitudinal direction: $N_x = \int_{h/2}^{h/2} \sigma_x dz$. Obtaining all forces and moments by integrating the stresses as defined by Equation A.7 yields Equation A.12.

The A_{ij} components are defined as shown in Equation A.8 and represent the extension/compression response of the laminate. The summation is simply the summation of the A_{ij} for each ply. Similarly, B_{ij} components are calculated using Equation A.9 and represents the coupling between extension/-compression and bending. The B-matrix is zero for symmetric laminates such as a cross-ply. The bending response of the laminate is represented by the D_{ij} components, which are calculated via Equation A.10.

$$A_{ij} = \sum_{k=1}^{n} Q_{ij} \left(z_k - z_{k-1}\right) \tag{A.8}$$

$$B_{ij} = \sum_{k=1}^{n} \frac{Q_{ij}}{2} \left(z_k^2 - z_{k-1}^2\right) \tag{A.9}$$

$$D_{ij} = \sum_{k=1}^{n} \frac{Q_{ij}}{3} \left(z_k^3 - z_{k-1}^3 \right) \tag{A.10}$$

To account for the through-the-thickness position of the plies, the local curvature is include in calculating the strain as shown in Equation A.11.

$$\begin{aligned} \varepsilon_x &= \varepsilon_{xo} + z\kappa_x \\ \varepsilon_y &= \varepsilon_{yo} + z\kappa_y \\ \gamma_{xy} &= \gamma_{xyo} + z\kappa_{xy} \end{aligned} \tag{A.11}$$

In summary, Equation A.12 shows a system of equations that allows the modelling of the mechanical response of a composite laminate based on four material characteristics and the thickness of the plies. The stress per ply can also be calculated.

$$\begin{Bmatrix} N_x \\ N_y \\ N_{xy} \\ M_x \\ M_y \\ M_{xy} \end{Bmatrix} = \begin{bmatrix} A_{11} & A_{12} & A_{16} & B_{11} & B_{12} & B_{16} \\ A_{12} & A_{22} & A_{26} & B_{12} & B_{22} & B_{26} \\ A_{16} & A_{26} & A_{66} & B_{16} & B_{26} & B_{66} \\ B_{11} & B_{12} & B_{16} & D_{11} & D_{12} & D_{16} \\ B_{12} & B_{22} & B_{26} & D_{12} & D_{22} & D_{26} \\ B_{16} & B_{26} & B_{66} & D_{16} & D_{26} & D_{66} \end{bmatrix} \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \\ \kappa_x \\ \kappa_y \\ \kappa_{xy} \end{Bmatrix} \tag{A.12}$$

Via further mathematical manipulation of the matrix above, it can be shown that for a symmetric, balanced laminate the Young's modulus is calculated via Equation A.13.

$$E_{1m} = \frac{1}{h} \frac{A_{11}A_{22} - A_{12}^2}{A_{22}} \tag{A.13}$$

Alternatively, one could plot the stresses in a laminate while increasing the applied strain to generate a stress-strain curve, as shown in Figure 5.9. This was simulated using a Python script, where both the global response and stresses per ply were updated under an increasing strain.

To apply damage, a Hashin failure criterion was adopted. A distinction was made between failure of the matrix and failure of the fibers. Failure of the matrix within a ply results in a reduction of the in-plane ply stiffnesses to 18% of its original value, which is in line with Camanho and Matthews [84]. A second failure of the matrix is interpreted as fiber failure, which degrades all properties to approximately zero (setting it to zero enforces numerical errors).

B

Damage Accumulation Around a Curved Crack

All images show the maximum principal stress field in a portion of the specimen at which a curved crack appears. The color bars are set to the same limit in each plot for fair comparison of the stress fields, ranging from 245 MPa (red) to -16 MPa (dark blue). With the specimen length of 100 mm, a global applied strain of 1000 $\mu\varepsilon$ corresponds with a total elongation of the specimen of 0.1 mm, measured at the boundary condition where the displacement load is applied.

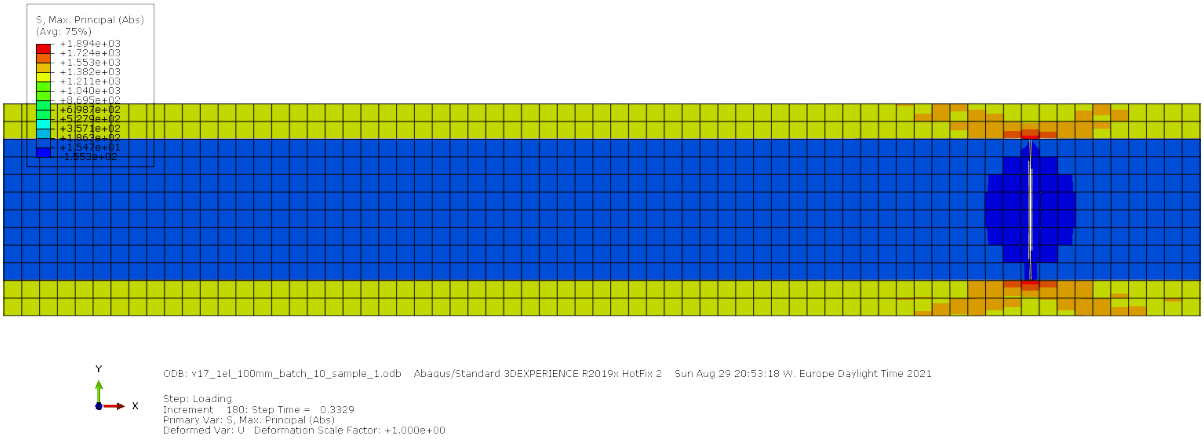

Figure B.1: Applied strain: 4994 $\mu\varepsilon$ — Increment: 180 — Step Time: 0.3329. Initiation of the first crack in this portion of the specimen.

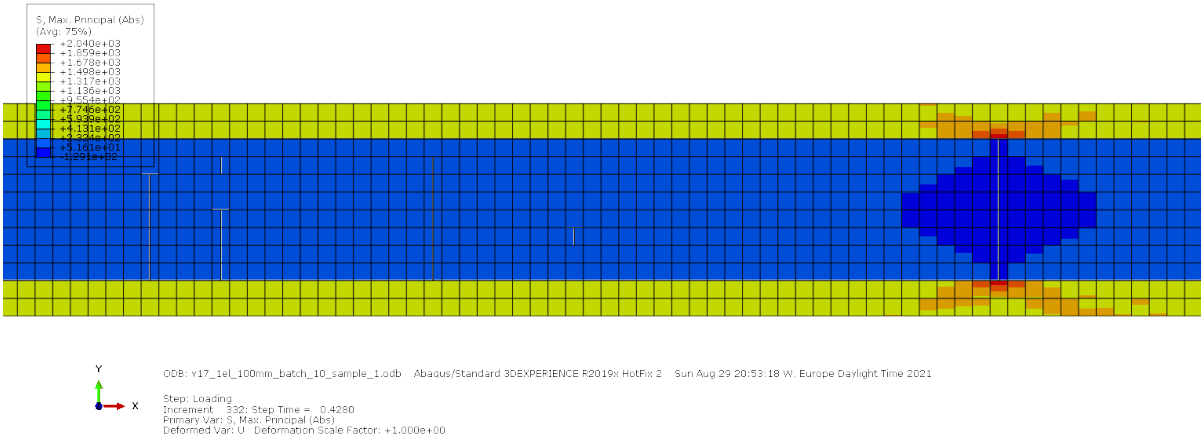

Figure B.2: Applied strain: 6420 $\mu\varepsilon$ — Increment: 332 — Step Time: 0.4280. Simultaneous damage initiation at four locations.

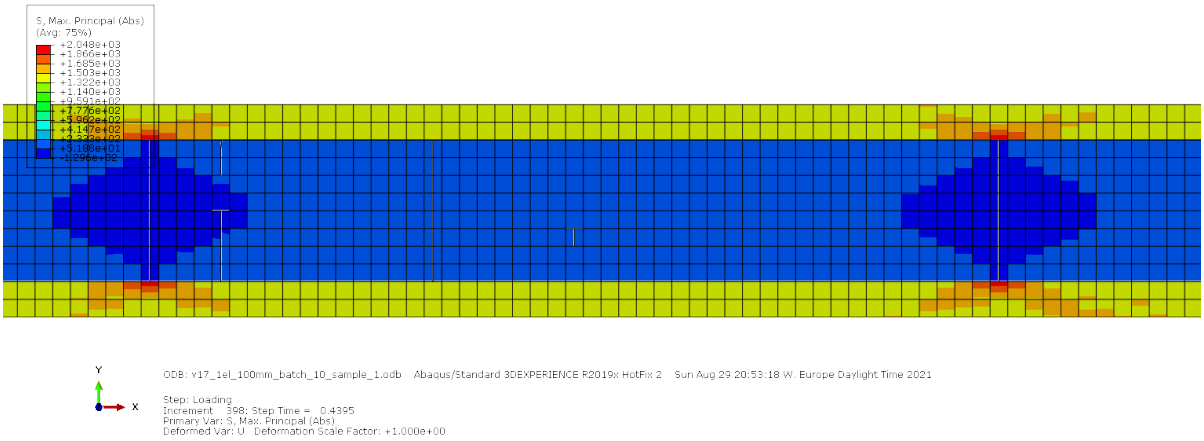

Figure B.3: Applied strain: 6593 $\mu\varepsilon$ — Increment: 398 — Step Time: 0.4395. Initiation of second crack.

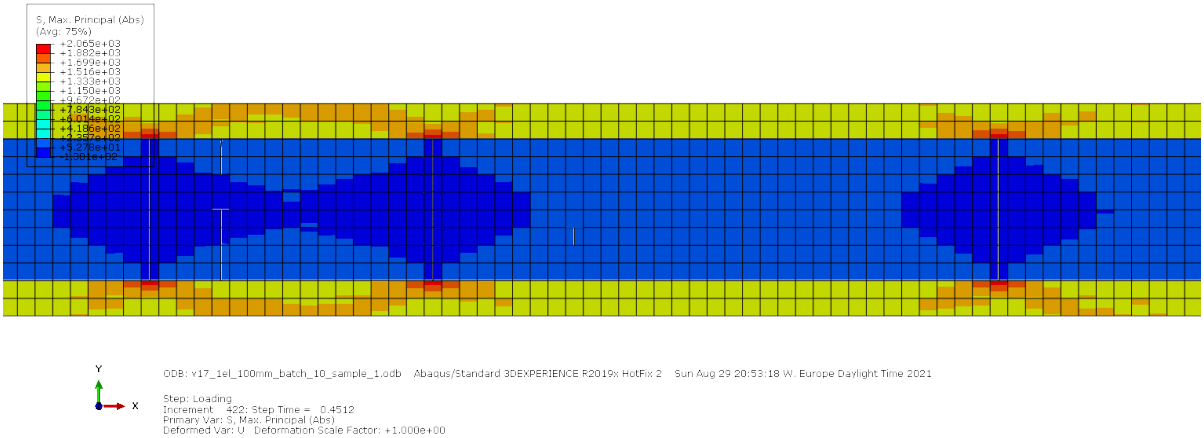

Figure B.4: Applied strain: 6768 $\mu\varepsilon$ — Increment: 422 — Step Time: 0.4512. Initiation of third crack.

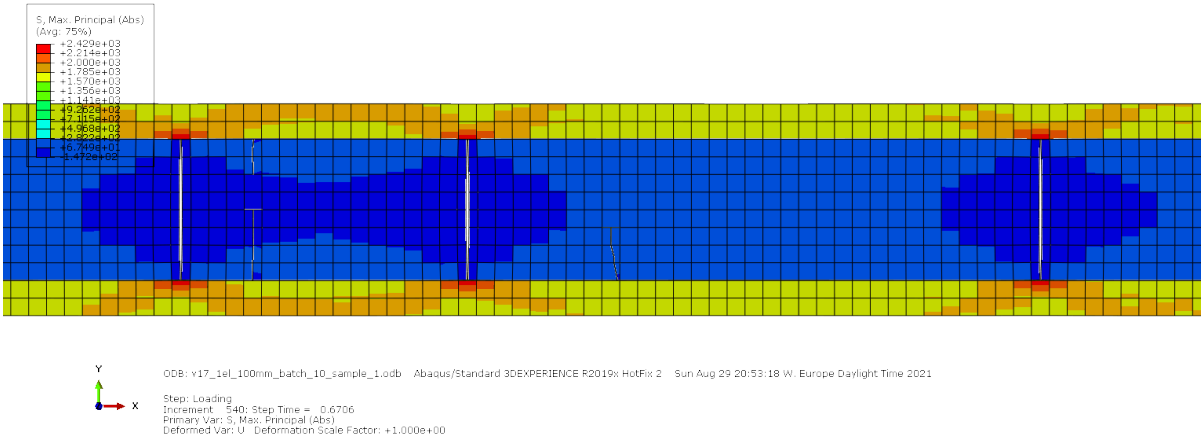

Figure B.5: Applied strain: 10059 $\mu\varepsilon$ — Increment: 540 — Step Time: 0.6706. Evolution of damage at the location where the curved crack initiates.

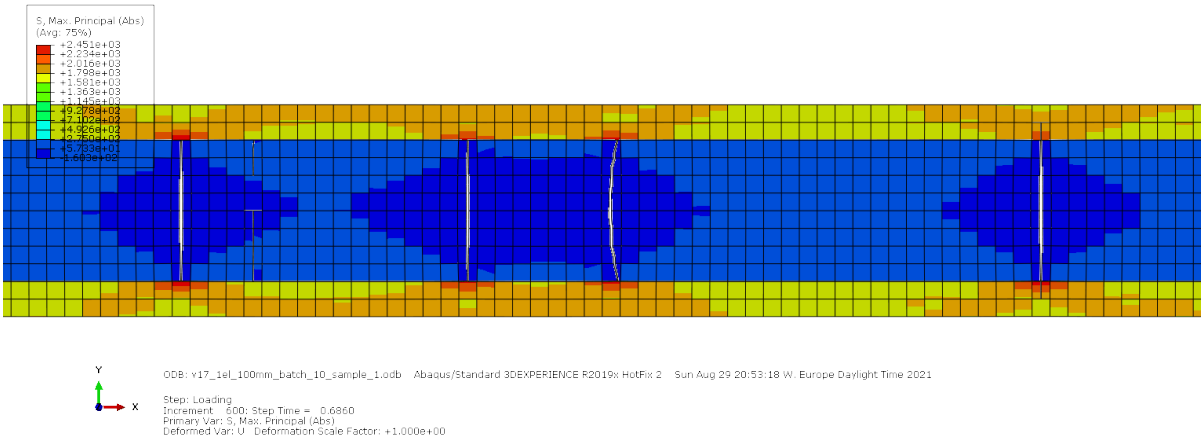

Figure B.6: Applied strain: 10290 $\mu\varepsilon$ — Increment: 600 — Step Time: 0.6860. Initiation of the curved crack.

C
Training Results

Table C.1: Training results from the initial configurations that were tested. The total loss is presented together with the top-5 accuracy of predicting the right location and the MSE of predicting the strain. Three models were selected based on a 'total score and are highlighted. The total score measures the overall performance, taking into account both the accuracy of predicting the strain and location. The total score was made up out of two elements: 1) the ranking based on lowest MSE, and 2) the ranking based on highest top-5 accuracy. A high ranking means a low score, so the models that performed best from an overall point of view have the lowest total score.

Reference no.	LSTM cell	Nodes	Dropout	Loss	MSE	Top-5 accuracy	Total score
1	1	2	0	4.5937	0.0234	0.0637	41
2	1	8	0	4.5867	0.0036	0.0722	11
3	1	16	0	4.5961	0.0043	0.0658	14
4	1	32	0.3	4.5654	0.0194	0.0849	25
5	1	64	0	4.6015	0.0052	0.0425	31
6	1	128	0.3	4.5494	0.019	0.0955	19
7	1	256	0.3	4.5413	0.0243	0.0807	35
8	2	2	0	4.6024	0.0086	0.0573	33
9	2	8	0	4.5928	0.0059	0.087	11
10	2	16	0	4.5991	0.0049	0.0913	7
11	2	32	0	4.5994	0.0045	0.0594	20
12	2	64	0	4.5903	0.0073	0.087	15
13	2	128	0.3	4.5934	0.0203	0.0594	39
14	2	256	0.3	4.5454	0.0217	0.0807	32
15	3	2	0	4.5988	0.0254	0.0679	39
16	3	8	0	4.6047	0.0098	0.0637	29
17	3	16	0	4.6005	0.0073	0.0488	34
18	3	32	0	4.6016	0.0063	0.051	29
19	3	64	0	4.6045	0.0097	0.0467	39
20	3	128	0.3	4.5959	0.0206	0.0616	38
21	3	256	0	4.5742	0.0063	0.0807	16
22	5	2	0	4.6026	0.0046	0.0467	27
23	5	8	0	4.6048	0.0058	0.0403	33
24	5	16	0	4.6105	0.0102	0.0318	45
25	5	32	0	4.5931	0.0075	0.0743	22
26	5	64	0.3	4.6121	0.019	0.0637	33
27	5	128	0.3	4.6145	0.0208	0.0403	50
28	5	256	0.3	4.6114	0.0208	0.051	45

Bibliography

[1] H. Wang, Z. Xu, H. Fujita, and S. Liu, "Towards felicitous decision making: An overview on challenges and trends of Big Data," *Information Sciences*, vol. 367-368, pp. 747–765, 11 2016.

[2] K. Smarsly, K. Dragos, and J. Wiggenbrock, "Machine learning techniques for structural health monitoring," in *8th European Workshop On Structural Health Monitoring (EWSHM 2016)*, (Bilbao, Spain), 7 2016.

[3] M. Danish, S. Taghipour, and C.-G. Lee, "Optimal RUL Estimation: A State-of-Art Digital Twin Application," tech. rep., University of Toronto, 2020.

[4] P. K. Majumdar, M. F. Haider, and K. Reifsnider, "Multi-physics response of structural composites and framework for modeling using material geometry," in *54th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 2013.

[5] S. R. Yeratapally, P. E. Leser, J. D. Hochhalter, W. P. Leser, and T. J. Ruggles, "A digital twin feasibility study (Part I): Non-deterministic predictions of fatigue life in aluminum alloy 7075-T651 using a microstructure-based multi-scale model," *Engineering Fracture Mechanics*, vol. 228, 1 2020.

[6] J. D. Hochhalter, W. P. Leser, J. A. Newman, E. H. Glaessgen, V. K. Gupta, V. Yamakov, S. R. Cornell, S. A. Willard, and G. Heber, "Coupling Damage-Sensing Particles to the Digitial Twin Concept," tech. rep., NASA Langley Research Center, Hampton, Virginia, 2014.

[7] P. M. Karve, Y. Guo, B. Kapusuzoglu, S. Mahadevan, and M. A. Haile, "Digital twin approach for damage-tolerant mission planning under uncertainty," *Engineering Fracture Mechanics*, vol. 225, 2 2020.

[8] P. E. Leser, J. E. Warner, W. P. Leser, G. F. Bomarito, J. A. Newman, and J. D. Hochhalter, "A digital twin feasibility study (Part II): Non-deterministic predictions of fatigue life using in-situ diagnostics and prognostics," *Engineering Fracture Mechanics*, vol. 229, 4 2020.

[9] B. R. Seshadri and T. Krishnamurthy, "Structural health management of damaged aircraft structures using the digital twin concept," in *25th AIAA/AHS Adaptive Structures Conference, 2017*, American Institute of Aeronautics and Astronautics Inc, AIAA, 2017.

[10] M. Grieves and J. Vickers, "Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems," in *Transdisciplinary Perspectives on Complex Systems: New Findings and Approaches*, pp. 85–113, Springer International Publishing, 1 2016.

[11] E. H. Glaessgen and D. S. Stargel, "The Digital Twin Paradigm for Future NASA and U.S. Air Force Vehicles," tech. rep., American Institute of Aeronautics and Astronautics, 2012.

[12] D. Jones, C. Snider, A. Nassehi, J. Yon, and B. Hicks, "Characterising the Digital Twin: A systematic literature review," *CIRP Journal of Manufacturing Science and Technology*, vol. 29, pp. 36–52, 5 2020.

[13] F. Tao, J. Cheng, Q. Qi, M. Zhang, H. Zhang, and F. Sui, "Digital twin-driven product design, manufacturing and service with big data," *International Journal of Advanced Manufacturing Technology*, vol. 94, pp. 3563–3576, 2 2018.

[14] R. Tharma, R. Winter, and M. Eigner, "An approach for the implementation of the digital twin in the automotive wiring harness field," in *Proceedings of International Design Conference, DESIGN*, vol. 6, pp. 3023–3032, Faculty of Mechanical Engineering and Naval Architecture, 2018.

[15] D. Wagg, K. Worden, R. Barthorpe, and P. Gardner, "Digital Twins: State-of-The-Art Future Directions for Modelling and Simulation in Engineering Dynamics Applications," *ASCE-ASME J Risk and Uncert in Engrg Sys Part B Mech Engrg*, 3 2020.

[16] E. J. Tuegel, A. R. Ingraffea, T. G. Eason, and S. M. Spottswood, "Reengineering aircraft structural life prediction using a digital twin," *International Journal of Aerospace Engineering*, 2011.

[17] K. Reifsnider and P. Majumdar, "Multiphysics stimulated simulation digital twin methods for fleet management," in *54th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 2013.

[18] C. Li, S. MahaDeVan, Y. Ling, S. Choze, and L. Wang, "Dynamic Bayesian network for aircraft wing health monitoring digital twin," *AIAA Journal*, vol. 55, no. 3, pp. 930–941, 2017.

[19] R. Ghanem, C. Soize, L. Mehrez, and V. Aitharaju, "Probabilistic learning and updating of a digital twin for composite material systems," *International Journal for Numerical Methods in Engineering*, 2020.

[20] Y. Ye, Q. Yang, F. Yang, Y. Huo, and S. Meng, "Digital twin for the structural health management of reusable spacecraft: A case study," *Engineering Fracture Mechanics*, vol. 234, 7 2020.

[21] M. Liao, G. Renaud, and Y. Bombardier, "Airframe digital twin technology adaptability assessment and technology demonstration," *Engineering Fracture Mechanics*, vol. 225, 2 2020.

[22] A. Gilchrist, "Introduction to the Industrial Internet," in *Industry 4.0*, ch. 1, pp. 1–12, Apress, 2016.

[23] Z. Wang, "Digital Twin Technology," in *Industry 4.0 - Impact on Intelligent Logistics and Manufacturing*, IntechOpen, 3 2020.

[24] A. Califano, N. Chandarana, L. Grassia, A. D'Amore, and C. Soutis, "Damage Detection in Composites By Artificial Neural Networks Trained By Using in Situ Distributed Strains," *Applied Composite Materials*, vol. 27, pp. 657–671, 10 2020.

[25] R. W. Batterman, "The Tyranny of Scales," tech. rep., Department of Philosophy, University of Pittsburgh, 2011.

[26] R. Talreja, "Multiscale failure assessment of composite laminates," in *Modeling Damage, Fatigue and Failure of Composite Materials*, pp. 349–355, Elsevier Inc., 2016.

[27] C. Kassapoglou, *Modeling the Effect of Damage in Composite Structures*. John Wiley & Sons, Ltd, 2015.

[28] M. Kashtalyan and C. Soutis, "Analysis of composite laminates with intra- and interlaminar damage," 2005.

[29] H. Pakdel and B. Mohammadi, "Characteristic damage state of symmetric laminates subject to uniaxial monotonic-fatigue loading," *Engineering Fracture Mechanics*, vol. 199, pp. 86–100, 8 2018.

[30] J. A. Glud, J. M. Dulieu-Barton, O. T. Thomsen, and L. C. Overgaard, "Fatigue damage evolution in GFRP laminates with constrained off-axis plies," *Composites Part A: Applied Science and Manufacturing*, vol. 95, pp. 359–369, 4 2017.

[31] X. Li, M. Saeedifar, R. Benedictus, and D. Zarouchas, "Damage accumulation analysis of cfrp cross-ply laminates under different tensile loading rates," *Composites Part C: Open Access*, vol. 1, p. 100005, 8 2020.

[32] J. A. Nairn, "Exact and variational theorems for fracture mechanics of composites with residual stresses, traction-loaded cracks, and imperfect interfaces," *International Journal of Fracture*, vol. 105, no. 3, 2000.

[33] E. Correa, F. París, and V. Mantič, "Fiber-matrix debonding in composite materials: Transverse loading," in *Modeling Damage, Fatigue and Failure of Composite Materials*, pp. 97–116, Elsevier Inc., 2016.

[34] E. Graciani, V. Mantič, F. París, and J. Varna, "Fiber-matrix debonding in composite materials: Axial loading," in *Modeling Damage, Fatigue and Failure of Composite Materials*, pp. 117–141, Elsevier Inc., 2016.

[35] S. Zhandarov and E. Mäder, "Characterization of fiber/matrix interface strength: Applicability of different tests, approaches and parameters," *Composites Science and Technology*, vol. 65, pp. 149–160, 1 2005.

[36] A. Forghani, M. Shahbazi, N. Zobeiry, A. Poursartip, and R. Vaziri, "An overview of continuum damage models used to simulate intralaminar failure mechanisms in advanced composite materials," in *Numerical Modelling of Failure in Advanced Composite Materials*, pp. 151–173, Elsevier Inc., 8 2015.

[37] M. J. M. J. Hinton, A. S. Kaddour, and P. D. P. D. Soden, *Failure criteria in fibre reinforced polymer composites : the World-Wide Failure Exercise*. Elsevier, 2004.

[38] K. A. Kalteremidou, M. Hajikazemi, W. Van Paepegem, D. Van Hemelrijck, and L. Pyl, "Effect of multiaxiality, stacking sequence and number of off-axis layers on the mechanical response and damage sequence of carbon/epoxy composite laminates under static loading," *Composites Science and Technology*, vol. 190, 4 2020.

[39] S. T. Pinho, *Modelling failure of laminated composites using physically-based failure models*. PhD thesis, University of London, 2005.

[40] L. Zubillaga, A. Turon, J. Renart, J. Costa, and P. Linde, "An experimental study on matrix crack induced delamination in composite laminates," *Composite Structures*, vol. 127, pp. 10–17, 9 2015.

[41] T. E. Tay, "Characterization and analysis of delamination fracture in composites: An overview of developments from 1990 to 2001," 1 2003.

[42] B. L. Bak, C. Sarrado, A. Turon, and J. Costa, "Delamination under fatigue loads in composite laminates: A review on the observed phenomenology and computational methods," 11 2014.

[43] E. A. Lindgren, J. C. Aldrin, D. H. Mollenhauer, and M. D. Flores, "Nondestructive evaluation for damage tolerance life management of composite structures," in *Lecture Notes in Mechanical Engineering*, pp. 1054–1064, Springer, 2020.

[44] P. A. Carraro, E. Novello, M. Quaresimin, and M. Zappalorto, "Delamination onset in symmetric cross-ply laminates under static loads: Theory, numerics and experiments," *Composite Structures*, vol. 176, pp. 420–432, 9 2017.

[45] J. A. Nairn and S. Hu, "The initiation and growth of delaminations induced by matrix microcracks in

laminated composites," *International Journal of Fracture*, vol. 57, pp. 1–24, 9 1992.

[46] M. R. Garnich and V. M. Akula, "Review of degradation models for progressive failure analysis of fiber reinforced polymer composites," 1 2009.

[47] E. J. Barbero, *Finite element analysis of composite materials using Abaqus*. CRC Press, 2013.

[48] B. Y. Chen and T. E. Tay, "Progressive damage in fibre-reinforced composites: Towards more accurate and efficient computational modelling and analysis," in *The Structural Integrity of Carbon Fiber Composites: Fifty Years of Progress and Achievement of the Science, Development, and Applications*, pp. 391–423, Springer International Publishing, 1 2016.

[49] A. De Luca and F. Caputo, "A review on analytical failure criteria for composite materials," *AIMS Materials Science*, vol. 4, no. 5, pp. 1165–1185, 2017.

[50] L. Kachanov, "Time of the Rupture Process under Creep Conditions," *Izvestiia Akademii Nauk SSSR, Otdelenie Teckhnicheskikh Nauk*, vol. 8, pp. 26–31, 1958.

[51] Robotnov Y.N., "Creep rupture in applied mechanics," in *Proceedings of the 12th International Congress on Applied Mechanics*, pp. 342–349, 1968.

[52] P. F. Liu and J. Y. Zheng, "Recent developments on damage modeling and finite element analysis for composite laminates: A review," *Materials and Design*, vol. 31, pp. 3825–3834, 9 2010.

[53] P. Maimí, P. P. Camanho, J. A. Mayugo, and C. G. Dávila, "A continuum damage model for composite laminates: Part I - Constitutive model," *Mechanics of Materials*, vol. 39, pp. 897–908, 10 2007.

[54] A. Matzenmiller, J. Lubliner, and R. L. Taylor, "A constitutive model for anisotropic damage in fiber-composites," *Mechanics of Materials*, vol. 20, pp. 125–152, 1995.

[55] X. F. Hu, B. Y. Chen, M. Tirvaudey, V. B. Tan, and T. E. Tay, "Integrated XFEM-CE analysis of delamination migration in multi-directional composite laminates," *Composites Part A: Applied Science and Manufacturing*, vol. 90, pp. 161–173, 11 2016.

[56] A. Griffith, "The Phenomena of Rupture and Flow in Solids," *Philosophical Transactions of the Royal Society of London, A*, vol. 221, pp. 163–198, 1921.

[57] D. S. Dugdale, "YIELDING OF STEEL SHEETS CONTAINING SLITS," *J. Mech. Phys. Solids*, vol. 8, pp. 100–104, 1960.

[58] G. I. Barenblatt, "The Mathematical Theory of Equilibrium Cracks in Brittle Fracture," *Advances in Applied Mechanics*, vol. 7, pp. 55–129, 1962.

[59] Z. Hashin, "Finite thermoelastic fracture criterion with application to laminate cracking analysis," *J. Mech. Phys. Solids*, vol. 44, no. I, pp. 1129–145, 1996.

[60] K. Friedrich, *Application of fracture mechanics to composite materials*. Elsevier, 1989.

[61] H. Pakdel and B. Mohammadi, "Progressive matrix cracking master curves of mid and outer off-axis plies in CFRP laminates," *Composite Structures*, vol. 188, pp. 497–502, 3 2018.

[62] W. Xu, Z. Z. Guo, Y. Yu, and X. J. Zhang, "Novel methods for measuring the mode I and mixed modes I/II interlaminar fracture toughnesses of composite," in *Lecture Notes in Mechanical Engineering*, pp. 461–476, Springer, 2020.

[63] J. Schijve, "Stress Intensity Factor of Cracks," in *Fatigue of Structures and Materials*, ch. 5, pp. 105–139, Springer, 2 ed., 2009.

[64] S. W. Tsai and E. M. Wu, "A General Theory of Strength for Anisotropic Materials," *J. Composite Mateirals*, vol. 5, pp. 58–80, 1971.

[65] R. M. Christensen, "Failure criteria for fiber composite materials, the astonishing sixty year search, definitive usable results," 9 2019.

[66] Z. Hashin and A. Rotem, "A fatigue failure criterion for fiber reinforced materials," tech. rep., Israel Institute of Technology, Technion, 3 1973.

[67] Z. Hashin, "Fatigue Failure Criteria for Unidirectional Fiber Composites," *Journal of Applied Mechanics*, vol. 48, pp. 846–852, 1981.

[68] A. Kaddour and M. Hinton, "Progress in failure criteria for polymer matrix composites: A view from the first World-Wide Failure Exercise (WWFE)," in *Failure Mechanisms in Polymer Matrix Composites*, pp. 3–25, Elsevier, 2010.

[69] C. Kassapoglou, "Stiffness & strength of composite structural elements," in *Polymer Composites in the Aerospace Industry*, pp. 147–193, Elsevier, 2020.

[70] A. Puck and H. Schürmann, "Failure analysis of FRP laminates by means of physically based phenomenological models," *Composites Science and Technology*, vol. 62, pp. 1633–1662, 2002.

[71] A. Puck, J. Kopp, and M. Knops, "Guidelines for the determination of the parameters in Puck's action plane strength criterion," *Composites Science and Technology*, vol. 62, pp. 371–378, 2002.

[72] R. G. Cuntze, "The predictive capability of failure mode concept-based strength criteria for multidirectional laminates-part B," *Composites Science and Technology*, vol. 64, no. 3-4, pp. 487–516, 2004.

[73] P. P. Camanho, "Failure Criteria for FRP Laminates in Plane Stress," tech. rep., NASA, 2003.

[74] P. P. Camanho, C. G. Dávila, S. T. Pinho, L. Iannucci, and P. Robinson, "Prediction of in situ strengths and matrix cracking in composites under transverse tension and in-plane shear," *Composites Part A: Applied Science and Manufacturing*, vol. 37, 2 2006.

[75] S. T. Pinho, C. G. Dávila, P. P. Camanho, L. Iannucci, and P. Robinson, "Failure Models and Criteria for FRP Under In-Plane or Three-Dimensional Stress States Including Shear Non-Linearity," tech. rep., NASA Langley Research Center, Hampton, Virginia, 2005.

[76] S. T. Pinho, R. Darvizeh, P. Robinson, C. Schuecker, and P. P. Camanho, "Material and structural response of polymer-matrix fibre-reinforced composites," *Journal of Composite Materials*, vol. 46, pp. 2313–2341, 9 2012.

[77] R. M. Christensen, "The world wide failure exercise II examination of results," *Journal of Reinforced Plastics and Composites*, vol. 32, pp. 1668–1672, 11 2013.

[78] M. Fakoor and S. Mohammad Navid Ghoreishi, "Experimental and numerical investigation of progressive damage in composite laminates based on continuum damage mechanics," *Polymer Testing*, vol. 70, pp. 533–543, 9 2018.

[79] T. E. Tay, G. Liu, V. B. Tan, X. S. Sun, and D. C. Pham, "Progressive failure analysis of composites," *Journal of Composite Materials*, vol. 42, pp. 1921–1966, 9 2008.

[80] J. Chen, E. Morozov, and K. Shankar, "A combined elastoplastic damage model for progressive failure analysis of composite materials and structures," *Composite Structures*, vol. 94, 12 2012.

[81] P. Camanho, P. Maimí, and C. Dávila, "Prediction of size effects in notched laminates using continuum damage mechanics," *Composites Science and Technology*, vol. 67, 10 2007.

[82] J. D. Lee, "THREE DIMENSIONAL FINITE ELEMENT ANALYSIS OF DAMAGE ACCUMULATION IN COMPOSITE LAMINATE," *Computers \& Struclures*, vol. 15, no. 3, pp. 335–350, 1982.

[83] C. T. McCarthy, M. A. McCarthy, and V. P. Lawlor, "Progressive damage analysis of multi-bolt composite joints with variable bolt-hole clearances," *Composites Part B: Engineering*, vol. 36, pp. 290–305, 6 2005.

[84] P. P. Camanho and F. L. Matthews, "A Progressive Damage Model for Mechanically Fastened Joints in Composite Laminates," *Journal of Composite Materials*, vol. 33, 12 1999.

[85] C. Zhang, N. Li, W. Wang, W. K. Binienda, and H. Fang, "Progressive damage simulation of triaxially braided composite using a 3D meso-scale finite element model," *Composite Structures*, vol. 125, pp. 104–116, 7 2015.

[86] T. Peng, Y. Liu, A. Saxena, and K. Goebel, "In-situ fatigue life prognosis for composite laminates based on stiffness degradation," *Composite Structures*, vol. 132, pp. 155–165, 11 2015.

[87] D. Cristiani, C. Sbarufatti, and M. Giglio, "Damage diagnosis and prognosis in composite double cantilever beam coupons by particle filtering and surrogate modelling," *Structural Health Monitoring*, pp. 1–21, 10 2020.

[88] Mucha, "Application of Artificial Neural Networks in Hybrid Simulation," *Applied Sciences*, vol. 9, 10 2019.

[89] X. Lu, M. Ridha, B. Y. Chen, V. B. Tan, and T. E. Tay, "On cohesive element parameters and delamination modelling," *Engineering Fracture Mechanics*, vol. 206, pp. 278–296, 2 2019.

[90] S. Pinho, L. Iannucci, and P. Robinson, "Physically based failure models and criteria for laminated fibre-reinforced composites with emphasis on fibre kinking. Part II: FE implementation," *Composites Part A: Applied Science and Manufacturing*, vol. 37, 5 2006.

[91] F. P. Van Der Meer and L. J. Sluys, "Continuum models for the analysis of progressive failure in composite laminates," *Journal of Composite Materials*, vol. 43, pp. 2131–2156, 9 2009.

[92] B. Chen, "Material nonlinearity." Lecture slides, 2019.

[93] R. Krueger, "Virtual crack closure technique: History, approach, and applications," *Applied Mechanics Reviews*, vol. 57, pp. 109–143, 1 2004.

[94] M. A. Eder and X. Chen, "FASTIGUE: A computationally efficient approach for simulating discrete fatigue crack growth in large-scale structures," *Engineering Fracture Mechanics*, vol. 233, 6 2020.

[95] T. Belytschko and T. Black, "Elastic crack growth in finite elements with minimal remeshing," *International Journal for Numerical Methods in Engineering*, vol. 45, pp. 601–620, 6 1999.

[96] L. Bouhala, Q. Shao, Y. Koutsawa, A. Younes, P. Núñez, A. Makradi, and S. Belouettar, "An XFEM crack-tip enrichment for a crack terminating at a bi-material interface," *Engineering Fracture Mechanics*, vol. 102, pp. 51–64, 4 2013.

[97] T. E. Tay, X. S. Sun, and V. B. Tan, "Recent efforts toward modeling interactions of matrix cracks and

delaminations: An integrated XFEM-CE approach," *Advanced Composite Materials*, vol. 23, pp. 391–408, 10 2014.

[98] T. P. Fries and T. Belytschko, "The extended/generalized finite element method: An overview of the method and its applications," *International Journal for Numerical Methods in Engineering*, vol. 84, pp. 253–304, 10 2010.

[99] N. A. Petrov, L. Gorbatikh, and S. V. Lomov, "A parametric study assessing performance of eXtended Finite Element Method in application to the cracking process in cross-ply composite laminates," *Composite Structures*, vol. 187, pp. 489–497, 3 2018.

[100] A. Hansbo and P. Hansbo, "A finite element method for the simulation of strong and weak discontinuities in solid mechanics," *Computer Methods in Applied Mechanics and Engineering*, vol. 193, pp. 3523–3540, 8 2004.

[101] F. P. Van Der Meer, L. J. Sluys, S. R. Hallett, and M. R. Wisnom, "Computational modeling of complex failure mechanisms in laminates," *Journal of Composite Materials*, vol. 46, pp. 603–623, 3 2012.

[102] T. Okabe and N. Takeda, "Estimation of strength distribution for a fiber embedded in a single-fiber composite: experiments and statistical simulation based on the elasto-plastic shear-lag approach," *Composites Science and Technology*, vol. 61, 9 2001.

[103] Z. Sun, I. M. Daniel, and J. J. Luo, "Modeling of fatigue damage in a polymer matrix composite," *Materials Science and Engineering A*, vol. 361, pp. 302–311, 11 2003.

[104] W. Lian and W. Yao, "Fatigue life prediction of composite laminates by FEA simulation method," *International Journal of Fatigue*, vol. 32, pp. 123–133, 1 2010.

[105] M. Naderi and A. R. Maligno, "Finite element simulation of fatigue life prediction in carbon/epoxy laminates," *Journal of Composite Materials*, vol. 47, pp. 475–484, 2 2013.

[106] F. P. Van Der Meer and C. G. Dávila, "Cohesive modeling of transverse cracking in laminates under in-plane loading with a single layer of elements per ply," *International Journal of Solids and Structures*, vol. 50, pp. 3308–3318, 10 2013.

[107] D. J. Lekou and T. P. Philippidis, "Mechanical property variability in FRP laminates and its effect on failure prediction," *Composites Part B: Engineering*, vol. 39, pp. 1247–1256, 10 2008.

[108] J. W. Jin, B. W. Jeon, C. W. Choi, and K. W. Kang, "Multi-scale probabilistic analysis for the mechanical properties of plain weave carbon/epoxy composites using the homogenization technique," *Applied Sciences (Switzerland)*, vol. 10, 9 2020.

[109] S. Zhang, C. Zhang, and X. Chen, "Effect of statistical correlation between ply mechanical properties on reliability of fibre reinforced plastic composite structures," *Journal of Composite Materials*, vol. 49, 9 2015.

[110] L. Wu, V.-D. Nguyen, L. Adam, and L. Noels, "An inverse micro-mechanical analysis toward the stochastic homogenization of nonlinear random composites," *Computer Methods in Applied Mechanics and Engineering*, vol. 348, 5 2019.

[111] J. R. Martinez and P. L. Bishay, "On the stochastic first-ply failure analysis of laminated composite plates under in-plane tensile loading," *Composites Part C: Open Access*, vol. 4, 3 2021.

[112] Y. Bao and H. Li, "Machine learning paradigm for structural health monitoring," *Structural Health Monitoring*, vol. 20, pp. 1353–1372, 7 2021.

[113] H. Sun, H. V. Burton, and H. Huang, "Machine learning applications for building structural design and performance assessment: State-of-the-art review," *Journal of Building Engineering*, vol. 33, 1 2021.

[114] Aurélien Géron, "Processing Sequences Using RNNs and CNNs," in *Hands-on Scikit-Learn for Machine Learning Applications*, pp. 497–523, Berkeley, CA: O'Reilly Media, Inc„ 2 ed., 2019.

[115] N.-H. Kim, D. An, and J.-H. Choi, "Data-Driven Prognostics," in *Prognostics and Health Management of Engineering Systems*, ch. 5, pp. 179–241, Cham: Springer International Publishing, 2017.

[116] K. Goebel, B. Saha, and A. Saxena, "A comparison of three data-driven techniques for prognostics," in *62nd Meeting of the Society for Machinery Failure Prevention Technology*, (Virginia Beach), 2008.

[117] T. H. Loutas, A. Panopoulou, D. Roulias, and V. Kostopoulos, "Intelligent health monitoring of aerospace composite structures based on dynamic strain measurements," *Expert Systems with Applications*, vol. 39, pp. 8412–8422, 7 2012.

[118] M. He, Y. Wang, K. Ram Ramakrishnan, and Z. Zhang, "A comparison of machine learning algorithms for assessment of delamination in fiber-reinforced polymer composite beams," *Structural Health Monitoring*, vol. 20, pp. 1997–2012, 7 2021.

[119] S. P. Curram and J. Mingers, "Neural Networks, Decision Tree Induction and Discriminant Analysis: an Empirical Comparison," *Journal of the Operational Research Society*, vol. 45, no. 4, pp. 440–450, 1994.

[120] G. K. Tso and K. K. Yau, "Predicting electricity energy consumption: A comparison of regression analysis,

decision tree and neural networks," *Energy*, vol. 32, 9 2007.

[121] Y. Bengio, "Learning Deep Architectures for AI," *Foundations and Trends® in Machine Learning*, vol. 2, no. 1, 2009.

[122] K. Greff, R. K. Srivastava, J. Koutnik, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A Search Space Odyssey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, pp. 2222–2232, 10 2017.

[123] C.-H. Wu, C.-C. Lu, Y.-F. Ma, and R.-S. Lu, "A New Forecasting Framework for Bitcoin Price with LSTM," in *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, IEEE, 11 2018.

[124] A. M. Abdelhameed, H. G. Daoud, and M. Bayoumi, "Deep Convolutional Bidirectional LSTM Recurrent Neural Network for Epileptic Seizure Detection," in *2018 16th IEEE International New Circuits and Systems Conference (NEWCAS)*, IEEE, 6 2018.

[125] W. Mucha, W. Kuś, J. C. Viana, and J. P. Nunes, "Operational load monitoring of a composite panel using artificial neural networks," *Sensors (Switzerland)*, vol. 20, 5 2020.

[126] X. Li, J. Kupski, S. Teixeira De Freitas, R. Benedictus, and D. Zarouchas, "Unfolding the early fatigue damage process for CFRP cross-ply laminates," *International Journal of Fatigue*, vol. 140, 11 2020.

[127] J. Kupski, S. Teixeira de Freitas, D. Zarouchas, P. Camanho, and R. Benedictus, "Composite layup effect on the failure mechanism of single lap bonded joints," *Composite Structures*, vol. 217, 6 2019.

[128] M. Smith, *ABAQUS/Standard User's Manual, Version 6.9*. Dassault Systèmes Simulia Corp, 2019.

[129] G. Van Rossum and F. L. Drake Jr, *Python tutorial*. Centrum voor Wiskunde en Informatica Amsterdam, The Netherlands, 1995.

[130] F. P. van der Meer, "Mesolevel Modeling of Failure in Composite Laminates: Constitutive, Kinematic and Algorithmic Aspects," *Archives of Computational Methods in Engineering*, vol. 19, 9 2012.

[131] Y. Wang, R. Jia, and F. Liu, "A ply-by-ply discretized 2D FEA approach with the integrated XFEM-CE strategy for predicting multiple failures in laminated composite structures," *CMES - Computer Modeling in Engineering and Sciences*, vol. 120, no. 1, pp. 215–234, 2019.

[132] Zhen-Zhong Du, "eXtended Finite Element Method (XFEM) in Abaqus."

[133] P. Maimí, P. P. Camanho, J. A. Mayugo, and C. G. Dávila, "A continuum damage model for composite laminates: Part II - Computational implementation and validation," *Mechanics of Materials*, vol. 39, pp. 909–919, 10 2007.

[134] J. Paśnik, S. Samborski, and J. Rzeczkowski, "Application of the CZM Technique to Delamination Analysis of Coupled Laminate Beams," *IOP Conference Series: Materials Science and Engineering*, vol. 416, 10 2018.

[135] A. Turon, C. Dávila, P. Camanho, and J. Costa, "An engineering solution for mesh size effects in the simulation of delamination using cohesive zone models," *Engineering Fracture Mechanics*, vol. 74, 7 2007.

[136] A. P. Duarte, A. Díaz Sáez, and N. Silvestre, "Comparative study between XFEM and Hashin damage criterion applied to failure of composites," *Thin-Walled Structures*, vol. 115, pp. 277–288, 6 2017.

[137] S. R. Hallett, B. G. Green, W. G. Jiang, and M. R. Wisnom, "An experimental and numerical investigation into the damage mechanisms in notched composites," *Composites Part A: Applied Science and Manufacturing*, vol. 40, pp. 613–624, 5 2009.

[138] N. G. Perogamvros and G. N. Lampeas, "Experimental and numerical investigation of AS4/8552 interlaminar shear strength under impact loading conditions," *Journal of Composite Materials*, vol. 50, pp. 2669–2685, 8 2016.

[139] F. van der Meer and L. Sluys, "Mesh-independent modeling of both distributed and discrete matrix cracking in interaction with delamination in composites," *Engineering Fracture Mechanics*, vol. 77, 3 2010.

[140] D. Kumar, R. Roy, J. H. Kweon, and J. h. Choi, "Numerical Modeling of Combined Matrix Cracking and Delamination in Composite Laminates Using Cohesive Elements," *Applied Composite Materials*, vol. 23, pp. 397–419, 6 2016.

[141] G. Alfano, "On the influence of the shape of the interface law on the application of cohesive-zone models," *Composites Science and Technology*, vol. 66, 5 2006.

[142] S. Sriramula and M. K. Chryssanthopoulos, "An experimental characterisation of spatial variability in GFRP composite panels," *Structural Safety*, vol. 42, 5 2013.

[143] D. Motamedi, M. Takaffoli, and A. S. Milani, "Nonlinear XFEM modeling of mode ii delamination in PPS/-glass unidirectional composites with uncertain fracture properties," *Materials*, vol. 13, 8 2020.

[144] F. Chollet, "Keras." https://github.com/fchollet/keras, 2015.

[145] Y. Gao, H. Yao, H. Wei, and Y. Liu, "Physics-based deep learning for probabilistic fracture analysis of composite materials," in *AIAA Scitech 2020 Forum*, vol. 1 PartF, American Institute of Aeronautics and Astronautics Inc, AIAA, 2020.

[146] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *3rd International Conference*

on Learning Representations, ICLR 2015, 12 2014.

[147] L. N. Smith, "Cyclical Learning Rates for Training Neural Networks," tech. rep., U.S. Naval Research Laboratory, Code 5514, Washington, D.C., 6 2015.

[148] S. Salman and X. Liu, "Overfitting Mechanism and Avoidance in Deep Neural Networks," tech. rep., Department of ComputerScience, Florida State University, Tallahassee, Florida, 1 2019.

[149] H. You, C. Li, P. Xu, Y. Fu, Y. Wang, X. Chen, R. G. Baraniuk, Z. Wang, and Y. Lin, "Drawing early-bird tickets: Towards more efficient training of deep networks," in *8th International Conference on Learning Representations, ICLR 2020*, (Addis Ababa, Ethiopia), 9 2019.

[150] E. Hüllermeier and W. Waegeman, "Aleatoric and epistemic uncertainty in machine learning: an introduction to concepts and methods," *Machine Learning*, vol. 110, pp. 457–506, 3 2021.

[151] Reddy J.N., "A generalization of two-dimensional theories of laminated composite plates," *Communications in applied numerical methods*, vol. 3, pp. 173–180, 1987.

[152] C. Kassapoglou, *Design and Analysis of Composite Structures*. John Wiley & Sons, Ltd, 2010.