Contents lists available at ScienceDirect

Machine Learning with Applications

journal homepage: www.elsevier.com/locate/mlwa

Graph-based tensile strength approximation of random nonwoven materials by interpretable regression

Dario Antweiler [a,b,*], Marc Harmening [c], Nicole Marheineke [c], Andre Schmeißer [b,d], Raimund Wegener [b,d], Pascal Welke [e]

[a] Fraunhofer IAIS, Schloss Birlinghoven, Sankt Augustin, 53757, Germany
[b] Fraunhofer Center for Machine Learning, Schloss Birlinghoven, Sankt Augustin, 53757, Germany
[c] Trier University, Universitaetsring 15, Trier, 54296, Germany
[d] Fraunhofer ITWM, Fraunhofer-Platz 1, Kaiserslautern, 67663, Germany
[e] University of Bonn, Friedrich-Hirzebruch-Allee 8, Bonn, 53115, Germany

ARTICLE INFO

MSC:
68T05
05C90

Keywords:
Nonwoven fiber material
Manufacturing
Textile fabrics
Material property prediction
Graph representation
Interpretable machine learning

ABSTRACT

Nonwoven materials consist of random fiber structures. They are essential to diverse application areas such as clothing, insulation and filtering. A long term goal in industry is the simulation-based optimization of material properties in dependence of the manufacturing parameters. Recent works developed a framework to predict tensile strength properties representing the fiber structure as a stochastic graph. In this paper we present an efficient machine learning approach using a regression model trained on features extracted from the graph, for which we develop a novel graph stretching algorithm. We demonstrate that applying our method to a practically relevant dataset yields similar prediction results as the original ODE approach ($R^2 = 0.98$), while achieving a significant speedup by up to three orders of magnitude. This opens the field to optimization, as Monte Carlo simulations accounting for the stochastic nature of nonwovens become easily accessible. Our model generalizes well to unseen parameter combinations. Additionally, our approach produces interpretable results by using a simple linear model for the regression task.

1. Introduction

Nonwoven materials are random fiber structures that are bonded by thermal, chemical, or mechanical means. The main advantage of such materials is their fast and cost-efficient production which makes them a suitable choice for various applications, including filter and insulating materials as well as medical and hygiene products such as face masks (Das & Pourdeyhimi, 2014). Aiming for a more sustainable production and an optimal material design with regard to desired fabric properties, there is a broad spectrum of research on predicting mechanical properties of nonwovens. Since experimental measurements are far too expensive, computer-based simulation approaches are used more frequently. But they involve a high computational effort (Wegener et al., 2015). Objective of this paper is to demonstrate the use of machine learning approaches for the prediction of nonwoven material properties. Therefore, we exemplarily focus on the tensile strength of airlay fabricated nonwovens (see Fig. 1(a)–(c)). In particular, we aim to predict the stress–strain curves commonly used to describe tensile strength behavior (Dowling, 2013). These curves indicate the relationship between strain and stress measured throughout the experimental tensile strength tests (see Fig. 1(d)). In production, the tensile strength behavior is of significant importance and can decide on the use of the materials in question, as fabrics with insufficient tensile strength stretch too easily or even tear rapidly.

Conventionally, determining the tensile strength associated with a setting of production parameters requires simulating the underlying production process and the resulting fiber structure's mechanical behavior. Here, we consider the following simulation framework, which is depicted in Fig. 2: For generating virtual production-like nonwoven samples we rely on Gramsch et al. (2016), who introduced a stochastic surrogate model for nonwoven airlay production ①. To predict the resulting fabric's tensile strength behavior we employ the model-simulation approach proposed by Harmening et al. (2020). Representing the fiber structure by a graph ② the approach is based on solving large-scale dynamical systems ③ depending on the graph topology. The simulation framework allows to map the involved production parameters to the resulting nonwovens' tensile strength ④ and forms the basis for inverse engineering. However, given the complex interactions of the many individual fibers involved, a single tensile

[*] Corresponding author at: Fraunhofer IAIS, Schloss Birlinghoven, Sankt Augustin, 53757, Germany.
E-mail addresses: dario.antweiler@iais.fraunhofer.de (D. Antweiler), harmening@uni-trier.de (M. Harmening), marheineke@uni-trier.de (N. Marheineke), andre.schmeisser@itwm.fhg.de (A. Schmeißer), wegener@itwm.fhg.de (R. Wegener), welke@cs.uni-bonn.de (P. Welke).

https://doi.org/10.1016/j.mlwa.2022.100288
Received 22 October 2021; Received in revised form 10 March 2022; Accepted 10 March 2022
Available online 16 March 2022

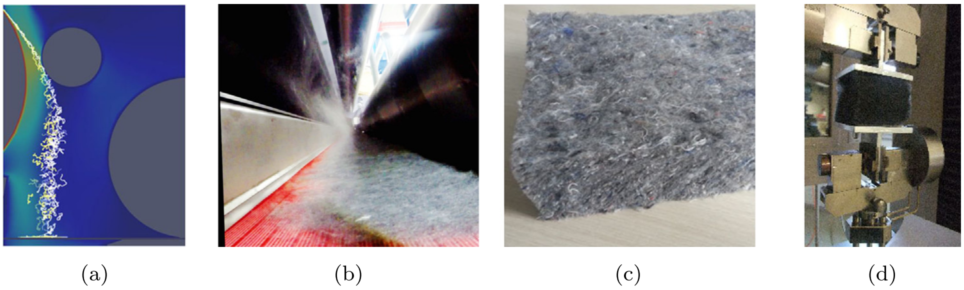

Fig. 1. Airlaid nonwoven materials' manufacturing and property testing: (a) simulated fiber dynamics and laydown in turbulent airflow, (b) fiber laydown zone, (c) final product, (d) tensile strength test for a nonwoven material sample.

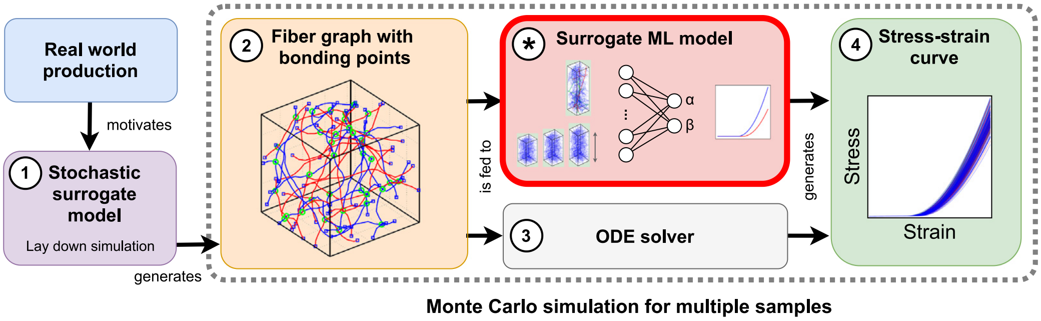

Fig. 2. Simulation framework using the stochastic fiber laydown model ① to generate a representative graph ② and applying the ODE-solver ③ to calculate the respective stress–strain curve ④. In this paper we aim to replace the computationally expensive ODE-solver by generating fiber graph features and using an interpretable regression model ⊛ to predict the stress–strain curve.

strength simulation is computationally expensive. The costs accumulate further due to Monte-Carlo simulations required to account for the randomness in the fiber structure generation. Repeated evaluation of the simulation framework, as required for virtual material design, is thus not possible in a reasonable time frame.

We want to avoid this trap of simulatability in principle and instead achieve efficient simulations in practice, which can contribute significantly to the design and optimization of nonwoven production processes. Hence, we introduce an efficient graph-based surrogate machine learning approach ⊛ that predicts the tensile strength of virtually generated fiber structure samples, bypassing the expensive tensile strength computations for individual samples after an initial training phase. It uses a regression model trained on features extracted from the graph, for which we develop a novel graph stretching algorithm. Our approach reduces the computation time by several orders of magnitude while yielding prediction results of similar quality. By exploiting insights into the process, we can constrain the machine learning approach to produce high-quality results with a simple regression model. Our feasibility study is performed on a 4-parametric process class and serves as a proof of concept. The underlying dataset covers a wide range of possible airlay scenarios. Our work is related to the research area of materials informatics and an example of informed machine learning (von Rüden et al., 2021). In the context of nonwoven material design, our approach is completely new.

We continue with an overview of previous works on tensile strength simulations of nonwovens and related machine learning approaches in Section 2. The underlying simulation framework consisting of the virtual fiber structure generation and the associated tensile strength determination is presented in Section 3. We discuss our novel surrogate machine learning approach in Section 4. Section 5 introduces the characteristics of the training data, describes our experiments, and discusses empirical findings, before we conclude in Section 6. The 3-part appendix provides details A) to our novel graph stretching algorithm including the analysis of correctness and runtime, B) to the model chain and parameters underlying the simulation framework and the process class under consideration, and C) to the quality of simulation and measurement data.

2. Related work

Our work belongs to the research areas of materials informatics and informed machine learning. In this section we comment on related work regarding the modeling of tensile strength behavior for nonwovens as well as machine learning approaches that describe the relationship between production parameters and material properties.

2.1. Tensile strength behavior of nonwoven materials

For virtual fiber structure generation a variety of approaches exist that mainly come from statistical analysis and stochastic geometry (Ohser & Mücklich, 2000; Schladitz et al., 2006) or three-dimensional volume imaging covering microscopy and X-ray tomography (Faessel et al., 2005; Ohser & Schladitz, 2009). However, realizing material structures from production parameters requires the simulation of the whole nonwovens' production process. Gramsch et al. (2016) established a simulation framework for an airlay fabricated nonwoven composite. To cope with the computational complexity that arises from several thousand airlaid fibers in a complex machine geometry, the authors introduce a chain of mathematical models coupled by means of parameter identification. The models cover a highly turbulent fiber suspension flow, a stochastic surrogate for the fiber laydown on a moving conveyor belt and a virtual bonding procedure imitating the incorporated thermobonding. The suitability of model hierarchies for the virtual generation of nonwovens and filaments is topic in Wegener et al. (2015).

A common procedure to predict material properties of nonwovens is to treat the material as a continuum which enables the use of efficient finite element methods (Demirci et al., 2011; Farukh et al., 2015). These approaches do not account for the individual fiber behavior. To incorporate the randomness in the material's microstructure they rely on knowledge of the statistical fiber orientation. Opposed to that, other approaches consider the material on the fiber-scale (Harmening et al., 2020; Kufner et al., 2018). Kufner et al. (2018) describe the material's structure as an elastic Cosserat network. However, since resolving the behavior of the individual fibers of beam type in a virtual material sample of industrial size is too complex, additional homogenization techniques are applied (Le Bris, 2010; Raina & Linder, 2014). Harmening et al. (2020) model the structure as truss of nonlinear elastic behavior and reduce the acting stress to the forces on the individual fiber connections that mainly determine the nonwovens' tensile strength behavior. A problem-tailored data reduction strategy and a singularly perturbed regularization approach make simulations with samples of industrial size possible. The approach particularly handles the problem-inherent multiscales (interplay of deterministic structural effects on macro-scale and random fiber distribution on micro-scale) and realizes the randomness by Monte-Carlo simulations.

2.2. Related machine learning approaches

We briefly discuss recent approaches to predict woven material properties from production parameters using machine learning. Early works used simple neural network architectures to predict the strength of yarn (Cheng & Adams, 1995) or worsted fabrics (Fan & Hunter, 1998). Both use a large number of measured input parameters and the neural network model prevents any interpretability of results. Abou-Nassif (2015) investigates both neural networks as well as linear regression to predict tensile strength of woven fabrics, but is limited to seven training samples. The approach in Ribeiro et al. (2020) deploys multiple regression models to predict different material properties of woven fabrics, but heavily relies on huge datasets and extensive manual feature selection by subject matter experts. Eltayib et al. (2016) use linear regression to predict tear strength of fabrics based on yarn count, yarn tenacity and fabric linear density. Due to the computational complexity of generating training data and their specialization to weaving features, these approaches cannot predict nonwoven fabric properties.

While much literature exists on woven materials, modeling approaches for nonwovens remain rare. Rahnama et al. (2013) use a feed-forward neural network based on a numerical propagation model to compare heat and moisture propagation through different nonwoven fabrics. Chen et al. (2007) integrates simple logical rules designed by domain experts into a neural network to predict elongation at break, but is limited to a single test example.

Aside from nonwoven fabric production, other works describe approaches to predict material properties from production parameters. We focus on those that try to integrate prior knowledge about the underlying physical mechanics into the data, the model architecture, or the loss term being optimized. Karpatne et al. (2017) integrate physical knowledge about feature dependencies as additional loss terms in a neural network. Lu et al. (2017) introduce an approach closest to our work by integrating knowledge about the underlying material mechanics as algebraic formulas into the machine learning approach. However, with a handcrafted neural network architecture to incorporate that knowledge, they do not account for interpretable results. We are not aware of any machine learning approaches predicting material properties for nonwoven fiber fabrics based on machine parameters. For recent surveys on the combination of machine learning and simulation approaches in a more general context, see von Rüden et al. (2021, 2020).

3. Problem statement

The airlay production process of nonwoven fiber materials typically uses different types of fiber material, such as synthetic fibers or reclaimed textile waste. The fibers exit from a rotating drum into a turbulent airflow and are blown onto a moving conveyor belt where they form a three-dimensional random nonwoven structure (cf. Fig. 1(a),(b)). During a heating treatment, the single fibers develop bonds and build the final product. The nonwoven material is characterized by the interplay of structural deterministic effects (such as the ramp-like material contour) on a macro-scale and randomness at the micro-scale (Fig. 1(c)). The focus of our study is to predict the behavior of a nonwoven material sample under strain (see Fig. 1(d) for a real-world measuring procedure). We are interested in the stress–strain curve that relates the relative strain in % of the original sample height to the reacting force of the material in Newton. The stress–strain curve is a highly relevant quality criterion for the produced material and an indicator for the tensile stress it can bear without tearing.

To predict the nonwovens' tensile strength from production parameters of a typical machine we build on the simulation framework of Harmening et al. (2020). It describes the nonwoven fiber structure as a parameter-dependent stochastic graph and computes its tensile strength behavior by solving a large-scale ordinary differential equation (ODE) on the graph. Fig. 2 illustrates the steps necessary to virtually map production parameters to the resulting material's tensile strength, our proposed changes are highlighted in ⊛. First in line is the generation of a virtual fiber structure ① with respect to a given production parameter combination. Fibers are drawn according to a stochastic laydown model and represented as piecewise linear Jordan curves in 3D space. The simulated curves are confined to a three-dimensional cuboid of fixed size to approximate a real-life punched material sample. Fiber endpoints together with virtual bonding points, where two or more curves meet each other closer than a fixed contact threshold, form a graph representation ②. The graph is reduced by removing all but the connected components joining the lower with the upper face of the cuboid. A tensile strength simulation based on an ordinary differential system on the graph ③ accurately predicts the stress–strain curve ④. Due to the random nature of the material, the fiber structure is represented by a stochastic graph. Thus, Monte-Carlo simulations are required to realize many graphs with the respective stress–strain curves and to predict the expected tensile stress behavior and its deviation for a given parameter combination. The graphs are embedded in a compact sample cuboid.

Note that the model chain underlying the simulation framework deals with 28 input parameters. After non-dimensionalization and under certain constraints on a process class, the number of practically relevant parameters can be reduced to four. Two of them influence the fiber deposit behavior in the stochastic laydown model, a third controls the amount of fibers used for the structure generation, and the remaining fourth parameter affects the sensitivity of the utilized bonding (graph topology). For details on model chain, parameters and the 4-parametric process class under consideration we refer to Appendix B.

In view of process design and material optimization the prescribed procedure by Harmening et al. (2020) suffers from its high computational cost. Whereas the virtual fiber structure generation is fast (few seconds per graph) and can be easily scaled up to a large number of samples, the ODE-solver is expensive. In fact, for one individual sample (one representative) a computation currently takes several hours up to multiple days. Thus, we aim for a machine learning model ⊛ to replace the ODE-solver by an efficient and interpretable approach.

4. Our surrogate model

We propose to learn a model on a given training dataset with varying parameter combinations to predict tensile strength behavior for unseen parameter combinations. We assume that the set of production parameters are known and adjustable, but the resulting effect on the tensile strength is unknown. In particular, as any given (graph) sample is just a representation of a stochastic process, we are interested not only in predictions for individual samples, but in the distribution of stress–strain curves for a given set of production parameters.

Our surrogate model approach uses the fiber graph generator and consists of two main parts. First, we approximate the observed stress–strain curve computed by the ODE-solver with a constant-quadratic function, which depends only on two parameters. Second, we employ a combination of nonlinear feature extraction based on the fiber graph

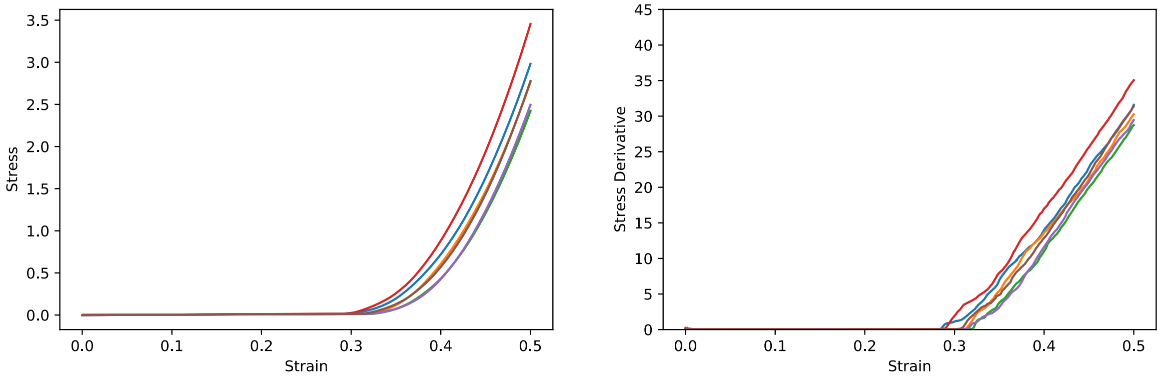

Fig. 3. Characteristic tensile strength behavior. Left: Example stress–strain curves obtained for a fixed parameter setting by ODE-approach. Right: Corresponding derivatives determined using central differences.

topology together with a simple, interpretable linear regression model to predict these two parameters. This allows the reconstruction of the tensile strength for a given sample. To this end, we learn the correspondence of parameters of the constant-quadratic function to topological features of the fiber graph.

4.1. Constant-quadratic approximation of stress–strain curves

We observe that the stress–strain curves obtained from the ODE-simulation for all parameter combinations follow a similar pattern: The curve is described by a constant part with near zero stress up to a threshold of applied strain, at which the curve establishes a quadratically increasing behavior. That the increase indeed exhibits a quadratic behavior becomes clear by examining the curve's derivative, which shows a piecewise linear behavior (cf. Fig. 3). The behavior is the result of more and more fibers coming under strain and thus contributing to the tensile behavior, neglecting plastic effects and fiber tearing. Moreover, the behavior allows to explain the observed stress–strain curves by means of a threshold of applied strain α, which indicates the beginning of the stress increase, and a coefficient β, which controls the quadratic behavior. This motivates the following constant-quadratic parametrization to approximately describe the relation between strain and stress for a given sample

$$T_{\alpha,\beta}(\epsilon) = \begin{cases} 0, & \epsilon < \alpha \\ \beta(\epsilon-\alpha)^2, & \epsilon \geq \alpha. \end{cases} \tag{1}$$

Here, ϵ refers to the relative strain applied to the sample and $T_{\alpha,\beta}$ describes the resulting reacting force, where $\alpha, \beta \in \mathbb{R}_{\geq 0}$ parametrize the curve. The replacement of the curve itself by an adequate estimation using only two parameters, enables a straight-forward machine learning modeling approach with the parameters α, β as our labels. To support the use of the constant-quadratic parametrization in this context, we provide the coefficient of determination ($R^2 = 0.99$) for the fitted curves on our dataset in Section 5.3.

4.2. Fiber graph feature extraction

Just as the physical structure of the real nonwoven material regulates the mechanical force which is needed to stretch the sample, the geometric arrangement of nodes (bonding points) and the topological structure of the fiber graph guides the ODE-simulation which results in the stress–strain curve. Hence, we select topological as well as geometric graph features, that presumably carry information about how easily the virtual fiber structure can be stretched, as features for our surrogate machine learning model. On the one hand, this includes basic information about the density and local structure of the graph, such as the number of nodes, edges and maximum degree. On the other hand, we include features that act as proxies for the ability to stretch, such as shortest paths connecting and minimum cuts separating the top and bottom layer of the graph. Lastly, we observe that each edge represents the connection between two bonding points along a fiber in the graph and its true length might differ from the Euclidean distance of its endpoints in physical space. Therefore, we calculate the difference between those values and aggregate them for the whole graph.

To define the features that we extract from such graphs, we clarify the informal introduction of the fiber graphs from above as follows: A *graph* $G = (V, E)$ consists of a set of nodes V and a set of edges E, each connecting exactly two nodes. In this work, we consider graphs that are embedded in $\mathbb{R}^3$, i.e., there exists a function $p : V \to \mathbb{R}^3$ mapping nodes to positions in space. For short, we will write $d(v, w)$ and $d(e)$ if $e = \{v, w\}$ is an edge to denote the Euclidean distance between $p(v)$ and $p(w)$. Furthermore, each edge e has a length $l(e)$. Graphs are assumed to be *undirected* and *loop-free*, i.e., edges are subsets of V with cardinality exactly two, but E can be a multiset. The set of neighbors of $v \in V$ is $\mathcal{N}(v) = \{w \in V \mid \{v, w\} \in E\}$. Let $\delta(v)$ be the number of edges incident to v (note that $\delta(v) \geq |\mathcal{N}(v)|$ in general). For a set of nodes $S \subseteq V$, we define $\mathcal{N}(S) = \bigcup_{v\in S} \mathcal{N}(v) \setminus S$. Finally, P_{uv} denotes a path in G from u to v and $\operatorname{len} P_{uv}$ is the number of edges in P_{uv}. We divide the nodes of the graph into three classes, nodes close to the lower and upper faces and nodes inside of the sample cuboid. The nodes near the lower and upper faces are of particular interest to us, to whose classes we refer to as V_l and V_u.

4.2.1. Graph features

In the "graph" feature set, we include the number of nodes $n = |V|$, number of edges $m = |E|$, maximum node degree $d_{\max} = \max_{v\in V} \delta(v)$, as well as the number of upper face nodes $|V_u|$ and lower face nodes $|V_l|$.

We assume that the interplay between positions of bonding points (nodes) and length of fibers (edges) connecting them, influences the final stress–strain curve of the simulated nonwoven. Hence, we encode this information in several features. In particular, we consider the overall sum of fiber lengths $L_{\text{fiber}} = \sum_{e\in E} l(e)$, as well as $L_1(P_1) = \operatorname{len} P_1$, the length of a shortest path (in terms of edge count) $P_1 = \operatorname{argmin}\left\{\operatorname{len}(P_{uv}) \mid u \in V_l, v \in V_u\right\}$, connecting the lower face V_l with the upper face V_u. Furthermore, we compute the sum of fiber lengths $L_2(P_2)$ along a shortest *weighted* path P_2 between V_l and V_u $\min_{P_{uv}}\left\{\sum_{e\in P_{uv}} l(e)\right\}$ with $u \in V_l, v \in V_u$ and the Euclidean length $L_3(P_2)$ of this weighted shortest path P_2. Finally, we consider the {mean, median, sum} of the differences between the Euclidean and fiber length distances over all edges $\{l(e) - d(e) \mid e \in E\}$ and the size of a minimum cut $C_{\min}$, i.e., an edge set with minimum cardinality that disconnects V_l from V_u when removed. For an illustration of these features see Fig. 4. Table 1 gives an overview of all "graph" features.

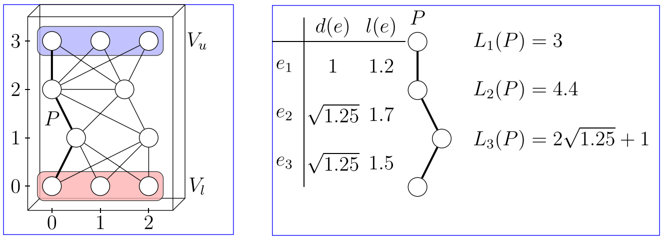

Fig. 4. Illustration of some graph features. The fiber graph on the left has $n = 10$ nodes, $m = 17$ edges. The sets V_u and V_l are colored in blue, and red, respectively. For the path P, we depict edge lengths $l(e)$, Euclidean distances between endpoints $d(e)$, as well as the three different path length variants L_1, L_2, L_3. A minimum cut C_{min} separates all nodes above value 1.5 from all nodes below that value, $|C_{min}| = 4$.

Table 1
Input features used in the proposed regression model.

Set	Symbols	Description
Param	–	Four production parameters characterizing the process class (cf. Appendix B)
Graph	n	Number of nodes
	m	Number of edges
	$d_{\max}$	Maximum node degree
	L_{fiber}	Sum of fiber lengths $\sum_{e \in E} l(e)$
	$\lvert V_u \rvert$	Number of upper face nodes
	$\lvert V_l \rvert$	Number of lower face nodes
	$L_1(P_1)$	Length of shortest path P_1 between V_l and V_u
	$L_2(P_2)$	Fiber lengths sum on weighted shortest path P_2
	$L_3(P_2)$	Euclidean length on weighted shortest path P_2
	D_1, D_2, D_3	$\{$mean, median, sum$\}$ of differences between Euclidean distance and fiber length
	$\lvert C_{\min} \rvert$	Size of minimum edge cut separating V_l and V_u
Stretch	$S_1^c, S_2^c, S_3^c, S_4^c, S_5^c$	{mean, std, median, max, sum} of stretching distance for $c \in \{1, 1.05, 1.1 \ldots, 1.5\}$

4.2.2. Stretching features

To complement the standard graph features, we propose a new one that is specifically tailored towards simulating the stretching of nonwovens. Recall that the process resulting in the stress–strain curves fixes the fibers at the bottom and top of a sample to two plates and pulls the two plates apart (cf. Fig. 1(d)). We ask how far this process (theoretically) can go without overstretching any fibers, i.e., how "far up" we can shift the bonding points without violating any individual length constraints while fixing the bonding points at the bottom in their positions.

In the initial state given by the graph generator, the Euclidean distance between two bonding points that are connected by a fiber should be at most the length of the fiber. We hence call a graph G with node positions p and edge lengths l a *valid instance* if

$$l(\{v, w\}) \geq \|p(v) - p(w)\|_2 = d(v, w) \;\; \forall \{v, w\} \in E \; . \tag{2}$$

We can reformulate our question as the maximization of the sum of height coordinates of the nodes in V_u while fixing the positions of the nodes in V_l and keeping the instance valid. To solve this problem efficiently, we make an additional simplifying assumption. We come up with a fast and simple algorithm to compute a lower bound on the maximum above. It allows the nodes of the fiber graph outside of V_l to move freely on the vertical (third) dimension, while fixing the coordinates in the horizontal plane (first and second dimension). Formally, we compute a solution to the following optimization problem, where $p(v)_i$ denotes the spatial coordinate to v, $i = 1, 2, 3$:

ZStretch

Given: a valid instance (G, p, l) and $V_l \subseteq V$

Maximize: $\sum_{v \in V \setminus V_l} p'(v)_3$ subject to

- $p'(v) = p(v) \; \forall v \in V_l$
- $p'(v)_1 = p(v)_1$ and $p'(v)_2 = p(v)_2 \; \forall v \in V$
- (G, p', l) is a valid instance

We note that this optimization problem ignores many properties of real-world nonwoven materials, such as the fact that fibers might be intertwined or cannot move freely around due to other reasons. However, it allows us to prove the following theorem. The corresponding algorithm (Algorithm 1) and proof of its correctness and runtime can be found in Appendix A.

Theorem 1. Given a valid instance (G, p, l) and a set $V_l \subseteq V$, the **ZStretch** problem can be solved in $O(|E| \log(|V|))$ time.

Given a valid instance (G, p, l), we use Algorithm 1 to compute the maximum stretch without violating any edge constraints. As a simple extension, we compute the maximum stretch *if we allow each fiber to be stretched to a multiple c of its length*. That is, we can investigate how the graph behaves if we consider edge weights $l'(e) = c\, l(e)$ for some $c > 1$. This is motivated by the physical (over-)stretching of fibers during tensile strength testing. Doing this for increasing values of c and a fixed graph results in nonlinear behavior of the average vertical positions of the nodes in V_u that we can use as additional features for our learning algorithm.

We use the mean, standard deviation, median, maximum and sum of the differences between original node positions and final positions of nodes in V_u after stretching along the vertical axis as features in our regression model. To add the possibility of 'over-stretching' we calculate these stretch features for multiple different length factors, namely $c \in \{1, 1.05, 1.1 \ldots, 1.5\}$ and use them as input to our model. We subsume these features in the "stretch" feature set, see Table 1 for all considered "stretch" features.

4.3. Summary of approach

Our proposed approach is illustrated in Fig. 5. Given a training set of production parameter combinations for the process class, we generate a set of fiber graph samples for each combination. With the ODE-solver we calculate stress–strain curves as our ground truth and input to our regression. We fit the constant-quadratic ansatz $T_{\alpha,\beta}$ (1) to each curve and extract the optimal values for α, β. For each graph, we calculate the graph and stretch features (cf. Table 1) as described in Section 4.2. Then, we fit linear regression models to express the relationship between the feature set and our targets α, β.

We use the resulting model as a surrogate for the ODE-solver (cf. ③ and ⊛ in Fig. 2). During inference, we predict a stress–strain curve given a fiber graph sample by calculating the graph and stretch features and determining α, β by means of our fitted model. We repeat this for multiple generated graph samples to account for the randomness of the nonwovens. We compute mean and standard deviation for the obtained set of curves.

5. Experiments

With the methodology described, this section specifies the evaluation of our approach. First, we report on the underlying dataset. Following, we characterize our experimental setting, report on our results, and close with a discussion on feature importance.

5.1. Dataset

We evaluate our approach on a dataset obtained from the fiber structure generator for the 4-parametric process class. The process class is motivated from the industrial setting in Gramsch et al. (2016) and covers practically relevant airlay scenarios. The used production parameter combinations ("param") are chosen randomly from a regime (fixed bounded domain) that result in reasonable fiber structures, cf., Appendices B and C. The computation of the associated stress–strain

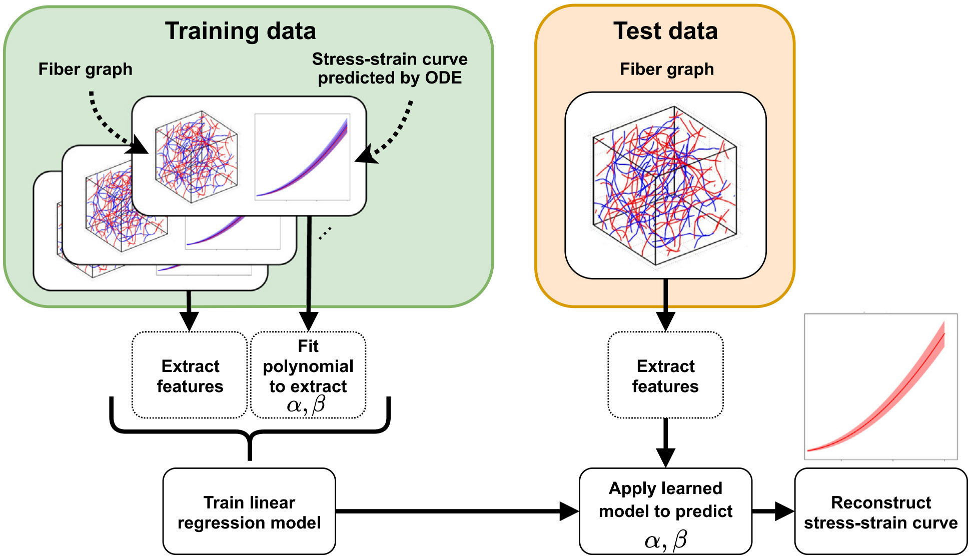
Training data
Fiber graph
Stress-strain curve predicted by ODE
Test data
Fiber graph
Extract features
Fit polynomial to extract α, β
Extract features
Train linear regression model
Apply learned model to predict α, β
Reconstruct stress-strain curve

Fig. 5. Regression pipeline for the prediction of the stress–strain curve from extracted fiber graph features.

Table 2
Composition of our dataset.

	Set 1	Set 2	Total
Graphs	6 × 25	37 × 25	1075
Stress–strain curves	6 × 25	37 × 1	187

curves requires solving large-scale dynamical systems tailored to the individual fiber structure samples. To account for the stiffness of the dynamical systems we employ an implicit Euler scheme with variable step size control. For solving the emerging nonlinear equation systems we use an exact Newton method employing analytical Jacobians and Armijo's line search. To warm start Newton's method we use a predictor determined over an explicit Euler scheme as initial guess.

While the graph generation and accompanying feature generation are fast, computing the stress–strain curves with the ODE-solver is very slow. For 43 parameter combinations, we generate 25 sample graphs each, totaling 1075 graphs. On average each graph contains 51.507 nodes (standard deviation ±2.182) and 198.744 edges (±29.996). We select six of our parameter combinations at random and compute the 25 stress–strain curves corresponding to the graphs using the ODE-solver (Set 1), while for all other combinations we compute only a single stress–strain curve for one of the corresponding samples (Set 2, cf. Table 2). This is due to the high computational costs of the ODE-solver. In fact, this solver typically requires between 24 and 28 h for a single instance, yielding multiple CPU weeks for our training dataset. Included in the dataset are hence only 187 supervised samples (6 × 25 samples + 37 × 1 sample) across 43 different parameter combinations. The graphs and the corresponding stress–strain curves act as ground truth examples for supervised learning. Given an unseen parameter combination, our goal is to predict the average behavior as well as a range of deviation of the resulting stress–strain curves as close as possible to the ground truth.

5.2. Experimental setup

We perform leave-one-out cross-validation across the parameter combinations. In each run, we separate the data into a training set containing the supervised samples of 42 parameter combinations and a test set containing the sample(s) of the remaining (single) parameter combination. To obtain training labels, we fit the constant-quadratic function $T_{\alpha,\beta}$ (1) onto the stress–strain curves inside the training set and use the best found parameters α, β as labels to train our model. During inference, the fitted model uses the features shown in Table 1 as input and predicts the parameters α, β, which are used to reconstruct the predicted stress–strain curve. For each parameter combination, we then compare predicted to ground truth curves.

Using parameter combinations with multiple supervised training samples (Set 1), we compare mean and standard deviation of the predicted and the ground truth curves respectively. For the single-sample supervised parameter combinations (Set 2) we take all 25 (mainly unlabeled) graph samples of that parameter combination and check how much the ground truth curve of the single labeled sample deviates from the mean of the predicted curves that our model produces. See Fig. 6 for examples of both cases.

To achieve a fair comparison, we first calculate the coefficient of determination R^2 and adjusted R^2 between the means of curves. We choose these measures as they provide a relative error between prediction and ground truth independent of the strain value. For example, given a single parameter combination from (test) Set 1 we predict a curve for each of the 25 graph samples with our model. We calculate the mean curve based on this set and compare it to the ground truth, i.e., the mean of all original curves. Given the variability of samples within the same parameter combination, this validation provides a robust estimation of model quality. While R^2 is a default evaluation score for regression tasks, we complement it with the adjusted R^2 score, which penalizes for larger numbers of selected attributes within a model via the formula

$$\overline{R^2} = 1 - (1 - R^2)\frac{n-1}{n-p-1}$$

where n is the number of data points and p is the number of features used by the model. That way, simpler and therefore more interpretable models are preferred.

Second, we perform an Optimal Transport (OT) optimization between the sets of curves embedded in $\mathbb{R}^d$, where $d = 1000$ is the number of base points at which the curve is sampled. In comparison to the median R^2 score, the OT score penalizes substantial differences between individual predicted and actual curves to a larger degree. With this additional score we can adequately assess the difference in distribution between prediction and ground truth curves. All experiments were done on a machine with 48 CPU cores (4x Intel(R) Xeon(R) Silver 4116 CPU @ 2.10 GHz) and 376 GB RAM running Debian 6.3.0 using Python 3.6.10.

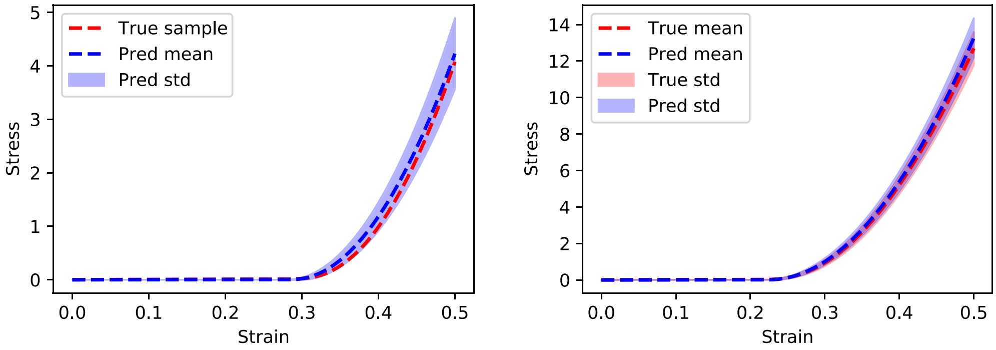

Fig. 6. Example results of predicted mean and standard deviation vs. ground truth. Left: a single ground truth curve vs. our predicted results. Right: mean and standard deviation for all available ground truth curves vs. our predicted results.

Table 3
Regression results on the synthetic dataset for baseline and our proposed approach. We take the ODE-solver results as ground truth for comparison and report Optimal Transport loss as well as median R^2 and adjusted R^2.

Feature set	Median R^2 ↑	Adj. R^2 ↑	OTLoss ↓
Baseline	0.39	–	–
Param	0.83	0.83	287.59
Stretch	0.97	0.96	111.77
Graph	0.97	0.97	99.65
Param + graph	0.97	0.97	82.62
Param + stretch	0.97	0.96	85.24
Param + graph + stretch	**0.98**	0.97	85.71
Graph + stretch	**0.98**	0.97	**71.44**
constant-quadratic approx.	**0.99985**	–	–

5.3. Results and discussion

Comparing our surrogate model to the ODE-simulation approach (cf., Fig. 2), we achieve a significant speedup. The time needed to compute a stress–strain curve for a sample generated by an unseen parameter combination is reduced by more than three orders of magnitude from 24–48 h per sample (ODE-simulation) to two minutes per sample (our approach). As both workflows can be executed in a parallelized fashion, we achieve a speedup of more than 1000×. This includes the generation of features and inference to test data.

We report the results for the 43-fold cross-validation over the parameter combinations in our dataset. Table 3 shows median coefficient of determination (R^2 and adjusted R^2) and OT loss for predicting stress–strain curves for a given parameter test set not part of training. As a baseline, we compute the mean of the fitted parameters α, β for all training examples and compare the reconstructed curve to the ground truth. We find that our model accurately predicts stress–strain curves for unseen parameter combinations and outperforms the baseline by a clear margin. Comparing different feature sets, the union of "graph" and "stretch" features achieves the best performance with a median coefficient of determination of $R^2 = 0.98$ and an OT loss of 71.44, calculated between mean predicted curves and the constant-quadratic approximation. This holds even compared to the case, where we add the production parameters "param" to the feature set. To our surprise, the "stretch" features calculated by Algorithm 1 already achieve good performance on their own, reaching $R^2 = 0.97$. This indicates that the topological and geometric structure of the fiber graph already encodes much of the behavior under vertical stretching. For comparison, we additionally evaluate a model trained solely on the (four) production parameters and find significantly worse performance. We explored the addition of lasso and ridge regularization to our regression model and found no significant change in results.

We investigate our approach by calculating feature importance for the trained linear regression model using the union of "graph" and "stretch" feature sets. Values are shown in Fig. 7. The larger the absolute value of the feature coefficient, the higher the impact on model predictions. We observe that high impact features differ between α and β prediction. Features from the "stretch" feature set display a large impact, particularly S_1 (mean) and S_5 (sum) and for larger values of length factor c. This indicates that the values calculated by Algorithm 1 already contain most of the information needed to predict tensile strength, while the "graph" features are only used for fine-tuning the result. As expected, as the number of edges m in the graph increases, the quadratic behavior in the stress–strain curve begins earlier (α is smaller) and the quadratic incline grows quicker (β is bigger). This is due to the fact that more fibers need to be stretched. The Euclidean length along a weighted shortest path $L_3(P_2)$ shows the opposite behavior. At a high value, the fibers along the shortest path have some room to be pulled apart in the tensile strength test without being stretched to their limit or beyond, and thus without contributing to the nonwoven's strength. Consequently, the feature value is positively related to α and negatively related to β. Overall, the coefficients are stable over different parameter combinations, indicating a robust model fitting. As seen in Table 3 our constant-quadratic function is a well-chosen approximation for the ground truth stress–strain curves.

It is remarkable that we achieve these powerful results with such a simple linear regression model. Presumably, this results from the integration of prior knowledge about the underlying physics via the fitting approximation and the calculated "graph" and "stretch" features.

6. Conclusion and future work

This work demonstrates the advantages of machine learning for tensile strength determination of nonwovens given as a network of bonded fibers in a stochastic graph structure. Our proposed regression model uses features extracted from the graph representation and generated by a novel graph stretching algorithm. We integrate prior knowledge of material behavior by fitting a constant-square approximation to the stress–strain curve. Compared to the sophisticated ODE-solution approach from literature, our model achieves speedups up to 1000-fold, while reliably obtaining stress–strain curves of comparable quality ($R^2 = 0.98$) for unknown combinations of production parameters. The dataset used for the 4-parametric production process class includes a variety of airlay scenarios. The simple, interpretable model allows experts to investigate the significance of the features. The speed increase achieved makes computer-aided process design and material optimization feasible, as it favors the use of simulation optimization methods required to cope with the stochastic nature of the presented simulation framework. The work can be considered as a feasibility study (proof of concept). In view of the modular nature of the underlying simulation framework, individual components (e.g., lay-down model or tensile strength model) can be easily replaced and the machine learning regression adapted accordingly. Even for more complex models, e.g., including effects such as bending and twisting of individual

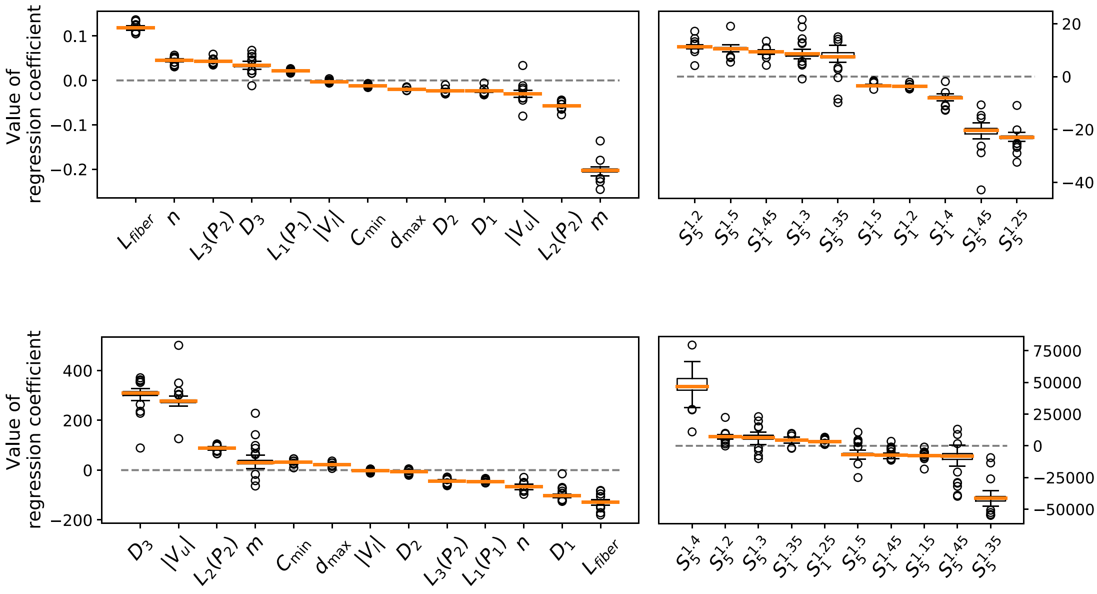

Fig. 7. Feature importance values for the linear regression models to predict α (top) and β (bottom). Left: Feature set "graph", right: "stretch". To reduce visual clutter, we display the five stretch features with biggest resp. smallest mean values.

fibers, this promises efficient predictions in a reasonable time frame. Equally important, the machine learning approach could also be used to predict other nonwoven properties such as insulation, flow resistance or acoustic properties, as well as material properties resulting from different production processes.

The data and code to train and evaluate our surrogate models is available at https://github.com/pwelke/random-nonwoven-fibers.

CRediT authorship contribution statement

Dario Antweiler: Conceptualization, Methodology, Software, Validation, Formal analysis, Writing – original draft, Writing – review & editing, Visualization, Project administration. **Marc Harmening:** Conceptualization, Methodology, Software, Validation, Formal analysis, Writing – original draft, Writing – review & editing, Visualization. **Nicole Marheineke:** Conceptualization, Methodology, Software, Validation, Formal analysis, Writing – original draft, Writing – review & editing, Visualization. **Andre Schmeißer:** Conceptualization, Methodology, Software, Validation, Formal analysis, Writing – original draft, Writing – review & editing, Visualization. **Raimund Wegener:** Conceptualization, Methodology, Software, Validation, Formal analysis, Writing – original draft, Writing – review & editing, Visualization. **Pascal Welke:** Conceptualization, Methodology, Software, Validation, Formal analysis, Writing – original draft, Writing – review & editing, Visualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The work of DA, AS and RW was conducted in Fraunhofer Cluster of Excellence Cognitive Internet Technologies. PW's research has partially been funded by the Federal Ministry of Education and Research of Germany as part of the Competence Center for Machine Learning ML2R (01|S18038C). MH and NM acknowledge the support of the Research Training Group Algorithmic Optimization (ALOP) at Trier University, funded by the German Research Foundation, Germany.

Appendix A. ZStretch problem and stretching algorithm

In this appendix we provide details on the computation of the stretch features for the fiber graphs. Recall that we call a connected graph $G = (V, E)$ with node positions $p : V \to \mathbb{R}^3$ and edge lengths $l : E \to \mathbb{R}_{\geq 0}$ a *valid instance* if

$$l(\{v,w\}) \geq \|p(v) - p(w)\|_2 = d\,(v,w) \;\; \forall\, \{v,w\} \in E,$$

where $d(\cdot,\cdot)$ denotes the Euclidean distance.

A.1. Problem formulation

Given a valid instance (G, p, l) with classes c, we call V_l and V_u the set of all nodes in the bottom and the top class, respectively. Ideally, to maximize the height of nodes in V_u, we would like to solve the following problem:

FreeStretch

Given: a valid instance (G, p, l) and $V_l \subseteq V$

Maximize: $\sum_{v \in V \setminus V_l} p'(v)_3$ s.t.

- $p'(v) = p(v)\ \forall v \in V_l$
- (G, p', l) is a valid instance

A solution to this problem would hence tell us, how far we can stretch a given virtual sample of a nonwoven without tearing any individual fibers if fibers cannot intertwine but move freely through each other. However, there are several technical problems with this formulation and we simplify it further to come up with a fast and simple algorithm to compute a lower bound on the maximum above.

ZStretch

Given: a valid instance (G, p, l) and $V_l \subseteq V$

Maximize: $\sum_{v \in V \setminus V_l} p'(v)_3$ s.t.

- $p'(v) = p(v)\ \forall v \in V_l$
- $p'(v)_1 = p(v)_1$ and $p'(v)_2 = p(v)_2\ \forall v \in V$
- (G, p', l) is a valid instance

As we have added the additional constraint that nodes may only move along the vertical axis, it immediately follows that the solution

Algorithm 1 Graph Stretching Algorithm.

```
Input: a valid instance $(G, p, l)$ and $V_l \neq \emptyset$

Output: a valid instance $(G, p', l)$ that maximizes the ZStretch objective

1: set $p'(v) = p(v)$ for all $v \in V$
2: set $V_\perp = V_l$ and $B = \mathcal{N}\left(V_\perp\right)$
3: for $v = \text{argmin}_{w \in B}\, maxMove(w, V_\perp)$ do
4:     pop $v$ from $B$
5:     $p'(v)_3 = maxMove(w, V_\perp) + p(v)_3$
6:     add $v$ to $V_\perp$
7:     $B = B \cup \mathcal{N}(v) \setminus V_\perp$
```

Algorithm 2 maxMove Subroutine.

```
Input: a vertex $v \in V$ and $V_\perp \subseteq V$.

Output: the largest $h$ s.t. $p'(v) = p(v) + (0,0,h)^T$ satisfies $d(p'(v), p(w)) \leq l(\{v, w\})$ for all $w \in \mathcal{N}(v) \cap V_\perp$

for all $\{v, w\} \in E$ do
    find the largest $h$ s.t.
    $p'(v) = p(v) + (0,0,h)^T$ satisfies
    $d(p'(v), p(w)) \leq l(v, w)$ for all $w \in \mathcal{N}(v) \cap V_\perp$
```

to the ZStretch problem is lower-bounding the FreeStretch problem. However, there is a fast algorithm that solves the ZStretch problem, which we will describe below.

A.2. Stretching algorithm

Algorithm 1 is very similar to Dijkstra's algorithm for weighted shortest paths. It iteratively "fixes" nodes in their maximal valid vertical coordinates and maintains an ever growing set of fixed nodes $V_\perp$. It starts by initially fixing the nodes in V_l and maintains a *border* B of nodes that are not yet fixed but connected to at least one fixed vertex. In each iteration, it selects a vertex v from B that can be moved by the least amount in the vertical coordinate (i.e., $p(v)_3$), while keeping the other coordinates fixed and while respecting the distance constraints given by edges from $V_\perp$ to v. Following this strategy allows us to prove that the algorithm maximizes the vertical values *of all nodes* under the given constraints in near linear runtime. This in turn implies that the algorithm maximizes the sum over all vertical values.

Theorem 1. *Given a valid instance (G, p, l) and a set $V_l \subseteq V$, Algorithm 1 solves the **ZStretch** problem in $O(|E| \log(|V|))$ time.*

We prove that Algorithm 1 is correct (Lemma 2) and runs in loglinear time (Lemma 3). The proof of Theorem 1 immediately follows from these two lemmas. To simplify the notation and appeal to intuition we will (without loss of generality) call a position shift in the vertical (third) dimension a move "up" or "down".

Lemma 2. *Given a valid instance (G, p, l) and a set $V_l \subseteq V$, Algorithm 1 maximizes $p'(v)_3$ for all $v \in V \setminus V_l$ s.t.*

1. $p'(v) = p(v)\ \forall v \in V_l$
2. $p'(v)_1 = p(v)_1$ *and* $p'(v)_2 = p(v)_2\ \forall v \in V$
3. (G, p', l) *is a valid instance*

Proof. We prove Lemma 2 by induction on the nodes in $V_\perp$, showing that the algorithm always maintains a valid maximized instance $G[V_\perp]$, i.e., the graph induced[1] by $V_\perp$ with updated p' is a solution to the **ZStretch** problem for $(G[V_\perp], p, l)$. Furthermore, we show that there always exists a valid instance (G, p'', l) with $p''|_{V_\perp} = p'|_{V_\perp}$, where $f|_X$ is the restriction of function f to the domain X. To this end, we note that at any time of the algorithm the sets $V_\perp$ and B are disjoint (cf. Lines 2 and 7). Furthermore, we never remove nodes from $V_\perp$. Finally, we change the z-coordinates of nodes only once, before adding them to $V_\perp$ (Line 5).

Initially, $V_\perp$ contains the nodes that must not be moved according to the objective of **ZStretch** (cf. Line 2). Hence, their coordinates will never be altered, as these nodes cannot be selected in Line 3 and the algorithm correctly maintains Condition 1. of Lemma 2. As we did not alter (G, p, l) we can assume that all conditions of Lemma 2 hold.

Now suppose we select some $v = \text{argmin}_{w \in B}\, maxMove(w, V_\perp)$. By definition, $maxMove$ respects the constraints $l(\{v, w\})$ of all edges $\{v, w\}$ that connect v to some $w \in V_\perp$, that is, for $\mathcal{N}(v) \cap V_\perp \subsetneq \mathcal{N}(v)$. By induction, we know that all $w \in V_\perp$ have maximal height. By definition, $h + p(v)_3$ hence maximizes the height of v without making $(G[V_\perp \cup \{v\}], p', l)$ invalid. Hence, $(G[V_\perp \cup \{v\}], p', l)$ fulfills all conditions of Lemma 2.

We will show that there exists a valid instance in which v is moved by $h = maxMove(w, V_\perp)$. More precisely, there exists a valid instance (G, p'', l) with $p''|_{V_\perp \cup \{v\}} = p'|_{V_\perp \cup \{v\}}$. We will show that

$$p''(w') = \begin{cases} p'(w') & \text{if } w' \in V_\perp \cup \{v\} \\ p(w') + (0,0,h)^T & \text{if } w' \in V \setminus (V_\perp \cup \{v\}) \end{cases}$$

results in a valid instance.

We have already seen that the constraints to any $w \in V_\perp$ are respected. Let now $w' \in W := V \setminus (V_\perp \cup \{v\})$. Recall that $h = \min_{w \in B} maxMove(w, V_\perp)$. Hence, we can move all nodes w' in B up by h without violating their constraints to nodes in $V_\perp$.

Now, for any $w' \in W \setminus B$, we have that w' has no edge to any vertex in $V_\perp$, hence, we can also move them up by h without violating any constraints to nodes in $V_\perp$. Furthermore, as (G, p, l) was a valid instance, we have that $l(\{w', w''\}) \geq d(p(w'), p(w'')) = d(p''(w'), p''(w''))$ for all $w', w'' \in W$. This results from the fact that we have fixed all but the vertical coordinates of nodes; shifting two nodes up by the same amount h does not change their relative distances to each other. Hence, there exists a valid instance (G, p'', l) where $p''(v) = p(v) + (0,0,h)^T$.

As G is connected, we will reach every vertex $v \in V$ at some point of the algorithm. The algorithm terminates if B is empty and has maximized the height of all nodes at that point. □

The second to last step of the above proof shows that we do not make our instance invalid by moving some v up at some point in the algorithm. Careless selection of the next vertex, however, might result in new positions p' that would "overstretch" certain edge constraints.

We now show that the algorithm can be implemented to run in near linear time. Note, however, that the pseudo-code given in the listing of Algorithm 1 shows the high level idea and needs to be modified to achieve the runtime claimed in the lemma below.

Lemma 3. *Algorithm 1 can be implemented to run in $O(|E| \log(|V|))$ time for a valid instance (G, p, l).*

Proof. Line 1 can be implemented in $O(|V|)$. To efficiently obtain the vertex from our border B which minimizes $maxMove(v, V_\perp)$, we use a heap for B that sorts by a key that is set to $maxMove(v, V_\perp)$. Hence, finding v minimizing $maxMove$ and popping it from the heap in Line 4 then requires $O(\log(|V|))$ time if the maximum number of elements in B is bounded by $|V|$ *or* $|E|$.

We note that for any $v \in B$, $maxMove(v, V_\perp)$ is always determined by an individual edge $e = \{v, w\}$ for some $w \in V_\perp$. This follows from

[1] The induced subgraph $G[X]$ of G is defined as $G[X] := (X, \{\{v, w\} \in E \mid v, w \in X\})$ for some $X \subseteq V$.

the definition of $maxMove$ as the largest h such that individual edge constraints are satisfied, and the fact that we only shift "up", while fixing the other dimensions. Hence, we can always find $maxMove(v, V_\perp)$ for any $v \in B$ by selecting the edge e that allows for the smallest shift of v, among all edges connecting $w \in V_\perp$ to $v \in B$. Thus, we can use a *queue over edges connecting* $V_\perp$ *to* B, ordered by the maximum move each edge allows. Line 2 can hence be implemented in $O(|E|\log(|V|))$. Node $v \in B$ minimizing $maxMove$ in Line 3 can now be found by popping an edge with the smallest key (i.e., smallest $maxMove$ value) repeatedly from the queue, until an edge is found that ends in some v that is not yet in $V_\perp$. Updating the height (Line 5) of v is then a constant time operation, as the required h is the key of the queue. Updating B in Line 7 can now be done in $O\left(|\mathcal{N}(v)| \log|V|\right)$, as we need to add or update the edges incident to v that connect to neighbors not yet in $V_\perp$.

As each vertex v is processed exactly once (cf. Line 6 and recall that G is assumed to be connected), we add each edge once to the queue and may update it once. Algorithm 1 hence runs in

$$O\left(|V| + |E|\log(|V|) + \sum_{v\in V} \left(\log(|V|) + |\mathcal{N}(v)| \log(|\mathcal{N}(v)|)\right) \right)$$

which is $O(|E|\log(|V|))$, by noting that $\sum_{v\in V} |\mathcal{N}(v)| = 2|E|$. □

Appendix B. Simulation framework and process class

This appendix provides details to the model chain underlying the simulation framework (Fig. 2) and the 4-parametric process class under consideration.

B.1. Model chain

A nonwoven material is the image of fibers deposited onto a moving conveyor belt. Consider a cubic reference material volume $\mathcal{V}_R$ over the nonwoven height H with base area w_R^2 and let T_R be the time needed to produce it. A deposited fiber of length L is identified with the lay-down time T and the planar coordinates (X, Y) of one of its end points. It contributes to $\mathcal{V}_R$, if $X - x_b(T) \in [-w_R/2, w_R/2]$ is satisfied, where x_b accounts for the motion of the conveyor belt. In the three-dimensional web a fiber is modeled in terms of the curve $\eta^{(X,Y,T)} : [0, L] \to \mathcal{V}_R$,

$$\begin{aligned}
\mathrm{d}\eta_s &= R(\eta_s \cdot e_x + x_b(T)) \cdot \tau_s \,\mathrm{d}s, \qquad \eta_0 = (X - x_b(T))e_x + Y e_y + r(X)e_z,\\
R(x) &= \frac{1}{\sqrt{1+r'(x)^2}} [I + (\sqrt{1+r'(x)^2} - 1)e_y \otimes e_y \\
&\qquad + r'(x)(e_z \otimes e_x - e_x \otimes e_z)],\\
r(x) &= H \int_{-\infty}^{x} g(\bar{x}) \,\mathrm{d}\bar{x}
\end{aligned}$$

with X g-distributed, $Y \sim \mathcal{U}([-w_R/2, w_R/2])$ and $T \sim \mathcal{U}([0, T_R])$ uniformly distributed — based on the stochastic Stratonovich differential system

$$\mathrm{d}\xi_s = \tau_s \,\mathrm{d}s, \qquad \mathrm{d}\tau_s = -\frac{1}{B+1}[\Pi_s(B) \cdot \nabla \Sigma(\xi_s) \,\mathrm{d}s + A\, \Pi_s(\sqrt{B}) \circ \mathrm{d}w_s]$$

with unit tensor I, projection $\Pi_s(x) = n_{1,s} \otimes n_{1,s} + x\, n_{2,s} \otimes n_{2,s}$ as well as $\xi_0 = 0$ and τ_0 uniformly distributed in the unit circle spanned by e_x and e_y. The stochastic lay-down model for position and orientation $((\xi, \tau) : [0, L] \to \mathbb{R}^3 \times \mathbb{S}^2)$ with unit sphere $\mathbb{S}^2 \subset \mathbb{R}^3$ describes the path of a deposited fiber onto the $e_x - e_y$ plane. In the modeling for the fiber tangent τ, the drift term prescribes the typical coiling behavior with the potential Σ, while the white noise term with the Wiener process $(w : [0, L] \to \mathbb{R}^3)$ and the amplitude A accounts for fluctuations in the lay-down process. Anisotropic behavior is indicated by the parameter $B \in [0, 1]$ with the local orthonormal triad $\{\tau, n_1, n_2\}$, $n_1 \in \mathrm{span}\{e_x, e_y\}$. The typical nestling behavior of the fiber on the ramp-like contour surface of the nonwoven is modeled by the curve η. The contour line r of the fiber material in machine direction is described by means of the joint probability density function g of the deposited material. A fiber end point lies on the associated contour surface and the fiber orientation is aligned to it due to the local rotation $R(x) \in SO(3)$.

We restrict our considerations to the embedded test material volume $\mathcal{V} \subset \mathcal{V}_R$ with smaller base w^2, $w = w_R - 2L$, to exclude lateral boundary effects. The random fiber web is consolidated by adhesive joints as a result of thermobonding. Let η_h denote the discretized fiber, i.e., set of discrete fiber points. An adhesive joint a to be formed between two fibers η_h and $\tilde{\eta}_\mathrm{h}$ is modeled as

$$\begin{aligned}
&a = \frac{1}{2}(q^\star + \tilde{q}^\star)\\
&\text{if} \quad \|q^\star - \tilde{q}^\star\|_2 < \kappa, \quad (q^\star, \tilde{q}^\star) = \operatorname*{argmin}_{(q,\tilde{q}) \in \eta_\mathrm{h} \times \tilde{\eta}_\mathrm{h}} \|q - \tilde{q}\|_2
\end{aligned}$$

with contact threshold $\kappa > 0$. The adhesive joint takes the place of the fiber points in contact in the respective fibers. As the minimizer might be not unique, we use the first minimizer found for practical reasons. Since the fibers lie rather straight, we assume at most one contact between each fiber pair. If more fibers are involved in a contact, the resulting adhesive joint is centered between the respective fiber points in contact. The resulting adhered fiber structure is considered as a connected graph $G = (V, E)$ with the nodes V representing adhesive joints as well as fiber ends and the edges E indicating fiber connections between them. The graph is supplemented by the node positions $p_0 : V \to \mathbb{R}^3$ and the edge-associated fiber lengths $l : E \to \mathbb{R}_{\geq 0}$.

The tensile strength test is modeled as differential system on the node positions $p : V \times [0, 1] \to \mathbb{R}^3$, initialized with $p(\cdot, 0) = p_0$,

$$\begin{aligned}
p(v, t) &= p_0(v), \qquad \forall v \in V_l, \qquad\qquad p(v, t) = p_0(v) + t\, h e_3, \quad \forall v \in V_u\\
\varepsilon\, \partial_t p(v, t) &= \sum_{e \in \delta(v)} f_e^v(t), \qquad\qquad\qquad\qquad \forall v \in V \setminus (V_l \cup V_u)\\
f_{e=\{v,v'\}}^v &= \frac{p(v') - p(v)}{d(e)} N\left(\frac{d(e) - l(e)}{l(e)}\right)
\end{aligned}$$

with $\delta(v) \subset E$ incident edges of node v. For fixed lower face V_l, the upper face V_u of the fiber structure is linearly shifted away in (vertical) e_3-direction (with maximal displacement $h > 0$). In the interior nodes of the graph the acting traction forces are balanced by a friction term with $\varepsilon > 0$. The force amplitude N depends on the relative strain of the fiber connection e with respect to its length $l(e)$, where $d(e)$ denotes the Euclidian distance. It reflects Hooke's law in the stretched state and is taken as zero in the unstretched state. The characterizing stress–strain relation for the fiber structure (with initial height H) is then given by $(\epsilon(t), T(p(\cdot, t)))$, $t \in [0, 1]$,

$$\epsilon(t) = \frac{h}{H} t, \qquad T(p(\cdot, t)) = -\sum_{v \in V_u} \sum_{e \in \delta(v)} f_e^v(t) \cdot e_3.$$

B.2. Parameters and process class

An airlaid nonwoven typically consists of two fiber types. Each is characterized by length L_f, line density $(\rho A)_f$, cross-sectional weighted elasticity modulus $(EA)_f$ and lay-down probability density g_f considered as normally distributed $g_f \sim \mathcal{N}(\mu_f, \sigma_f^2)$, $f = 1, 2$. The joint probability density is then $g = \beta_n g_1 + (1 - \beta_n) g_2$ with fiber number fraction β_n determined by mass fraction β. For technical reasons we use a compact support $supp(g) = [x_l, x_r]$. The production plant is characterized by conveyor belt width b and speed v_B as well as mass rate $\dot{m}$, the nonwoven sample is specified by height H and width w. Production time T_R, trace curve x_B and number of deposited fibers per type n_f, $f = 1, 2$ are resulting quantities. The laydown is parametrized regarding diffusion A, anisotropy B and bending potential Σ expressed by the three standard deviations σ_x, σ_y, σ_z in e_x, e_y, e_z-directions. The bonding considers fiber discretization length $\triangle s$ and contact threshold κ. The strength test is parametrized by adhesive thickness z for upper and lower structure faces, friction-associated regularization ε as well as traction function N with a regularization parameter δ. In total, there are 28 input parameters for the model chain, whereas the displacement

Table B.4
Characteristic dimensionless input parameters for model chain. Values for industrial airlay process, plant K12 with a mixture of solid (PES) and bi-component (PES/PET) fibers (cf. scenario in Gramsch et al., 2016 and measurement data in Appendix C). Referential values in SI units: $w = 1.0 \cdot 10^{-2}$ m, $v_B = 3.3 \cdot 10^{-2}$ m/s, $(EA)_1 = 1.0$ N.

Description	Symbol		Value
Fiber length	L_1/w,	L_2/L_1	5.5, 1.0
Fiber number	$\alpha_1 w^2/v_B$,	α_2/α_1	1150, 0.65
Elasticity modulus	$(EA)_2/(EA)_1$		1.0
Lay-down pdf mean	μ_1/w,	μ_2/w	0, 0
Lay-down pdf std	σ_1/w,	σ_2/σ_1	2.0, 1.0
Support joint lay-down pdf	x_l/σ_1,	x_r/σ_1	−5.0, 5.0
Nonwoven sample height	H/w		6.0
Bending potential (std)	σ_y/w, σ_x/σ_y, σ_z/σ_y		2.0, 0.75, 0.075
Diffusion	$A\sqrt{\sigma_y}$		$2.8 \cdot 10^{-2}$
Anisotropy	B		$3.0 \cdot 10^{-1}$
Fiber discretization	$\triangle s/w$		$3.7 \cdot 10^{-2}$
Contact threshold	κ/w		$2.6 \cdot 10^{-2}$ (calibrated)
Adhesive thickness at faces	z/w		$6.0 \cdot 10^{-2}$
Friction regularization	ε		$1 \cdot 10^{-7}$
Traction regularization	δ		$1 \cdot 10^{-4}$

Table B.5
4-parametric process class for machine learning approach in Section 5. Parameter ranges for dataset used in ML approach and corresponding values in industrial scenario (cf. Table B.4). The values of all other input parameters (ratios) are taken from Table B.4.

Symbol	Range	Industrial Value	Effect
$\hat{\alpha} = \alpha_1 w^2/v_B$	$[1000, 1515]$	1150	Amount of fibers
$\hat{\sigma} = \sigma_1/w$	$[1.0, 5.0]$	2.0	Laydown behavior
$\hat{\sigma}_y = \sigma_y/w$	$[1.0, 5.0]$	2.0	Laydown behavior
$\hat{\kappa} = \kappa/w$	$[2.8, 3.0] \cdot 10^{-2}$	$2.6 \cdot 10^{-2}$	Bonding

h in the strength test belongs to the output quantities, i.e., stress–strain relationship.

Since the parameters $(\rho A)_1$ $(\rho A)_2$, β, $\dot{m}$ and b only occur in the quantities $\alpha_1 = \beta\dot{m}/((\rho A)_1 L_1 b))$ and $\alpha_2 = (1-\beta)\dot{m}/((\rho A)_2 L_2 b)$ indicating the number of fibers for each type deposited per second and meter in cross direction on the conveyor belt, three parameters can be eliminated. Making the model problem dimensionless with nonwoven sample width w, conveyor belt speed v_B and elasticity modulus $(EA)_1$ reduces the set of input parameters by further three. The resulting dimensionless numbers are mainly formulated as ratios, see Table B.4. Note that the equation for the strength test was already considered in dimensionless form in order to incorporate the friction-associated (dimensionless) regularization parameter $\varepsilon \ll 1$ that ensures a unique solution. In this paper we focus on a 4-parametric process class. The process class is motivated from the industrial test setting in Gramsch et al. (2016) (airlay scenario with a mixture of solid (PES) and bi-component (PES/PET) fibers in plant K12), see values of dimensionless parameters (ratios) in Table B.4. For the process class we adopt the (industrial) values — except for $\hat{\alpha}$, $\hat{\sigma}$, $\hat{\sigma}_y$ and $\hat{\kappa}$, see Table B.5. These four parameters affect the fiber amount in the nonwoven (sample), the fiber laydown behavior as well as the bonding (i.e., fiber graph topology). By varying them in a certain regime, a broad variety of practically relevant airlay scenarios are covered. See Table B.5 for the parameter ranges underlying the dataset for our experiments in Section 5. Note that the larger chosen $\hat{\kappa}$ ensures a stronger bonding and hence a denser fiber structure than in the industrial test case.

Appendix C. Data by model and measurements

In the experimental real-world tensile strength test, the measured stress–strain curves for nonwoven material samples of fixed size from the same production setting show a large volatility, in particular for increasing strains $\epsilon > 0.2$. This observation is illustrated by three measured curves that correspond to the reliability zone within the measurement accuracy for the industrial test setting described in Gramsch et al. (2016) with sample base width $w_M = 1.0 \cdot 10^{-1}$ m and applied pre-force $T^\star = 1.0$ N, see Fig. C.8. Note that the fiber structure in this real-world experiment is loose (less bonded) to allow microstructure analysis.

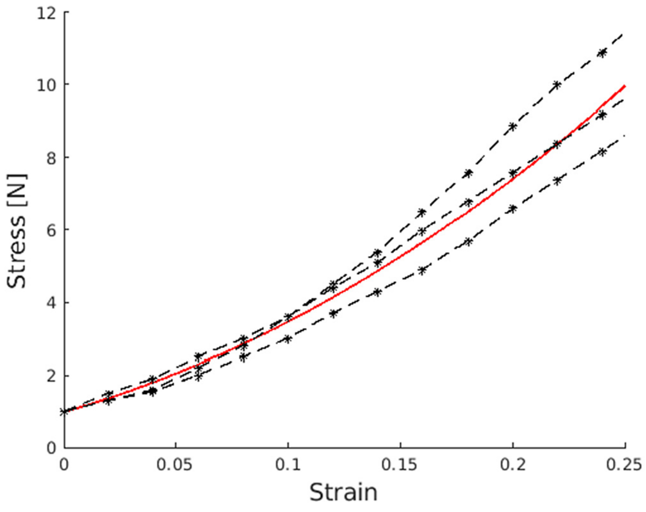

Fig. C.8. Stress–strain relations for industrial test setting in Gramsch et al. (2016) with sample base width $w_M = 1.0 \cdot 10^{-1}$ m and pre-force $T^\star = 1.0$ N. Measurements (dashed black lines) vs. our model (solid red line), cf. Table B.4.

Our model class of constant-quadratic approximations $T_{\alpha,\beta}$ (1) fits obviously very well. For comparability of our model with the measurements we have to account for pre-force (with associated pre-strain $\epsilon^\star$) and sample base width ratio. The respective (dimensionless) model stress-strain curve with respect to the pre-force $T^\star$ is given by

$$\hat{T}(\hat{\epsilon}) = \left(\frac{w_M}{w}\right)^2 \frac{1}{T^\star} T_{\alpha,\beta}(\epsilon)\Big|_{\epsilon=\epsilon^\star(\hat{\epsilon}+1)+\hat{\epsilon}} = (M_{\alpha,\beta}\,\hat{\epsilon}+1)^2 \tag{C.1}$$

$$\text{with} \quad M_{\alpha,\beta} = \frac{w_M}{w}\sqrt{\frac{\beta}{T^\star}}(\alpha+1)+1$$

where $\epsilon^\star = \alpha + \sqrt{T^\star/\beta}$. The quadratic curve is parametrized by the single term $M_{\alpha,\beta} > 0$ that depends on the parameters α, β of our model class (1).

The stress–strain relations (C.1) obtained from the simulation framework match the measured ones. However, note that the simulation results are affected by model parameters that are neither given nor can be identified in advance but must be calibrated, such as the topological contact threshold. While the calibration with the ODE-approach is tedious and computationally expensive, it turns out to be very easy to perform with the ML regression. For the industrial setting at hand with the model parameters of Table B.4, we find $\hat{\kappa} = 2.6 \cdot 10^{-2}$, implying $\alpha = 0.401$, $\beta = 0.297$ N and $M_{\alpha,\beta} = 8.632$ in mean (see Fig. C.8). Note that this value lies outside of the training dataset used in Section 5 (cf. Table B.5). The larger $\hat{\kappa}$ chosen for the training involves a denser fiber structure of higher realistic tensile strength.

To cope reliably with the nonwovens' random nature and dependence on production parameters, the execution of large (real-world) measurement series is much too expensive and time-consuming. Hence, we use the ODE-simulations as ground truth to train and investigate the performance of our machine learning approach. The dataset considered for the 4-parametric (production) process class reflects possible airlay scenarios within the industrial setting of Gramsch et al. (2016).

References

Abou-Nassif, G. A. (2015). Predicting the tensile and air permeability properties of woven fabrics using artificial neural network and linear regression models. *Journal of Textile Science & Engineering*, *05*(05), http://dx.doi.org/10.4172/2165-8064.1000209.

Chen, T., Li, L., Koehl, L., Vroman, P., & Zeng, X. (2007). A soft computing approach to model the structure–property relations of nonwoven fabrics. *Journal of Applied Polymer Science*, *103*(1), 442–450. http://dx.doi.org/10.1002/app.24909.

Cheng, L., & Adams, D. L. (1995). Yarn strength prediction using neural networks: Part I: Fiber properties and yarn strength relationship. *Textile Research Journal*, *65*(9), 495–500. http://dx.doi.org/10.1177/004051759506500901.

Das, D., & Pourdeyhimi, B. (2014). *Composite nonwoven materials: Structure, properties and applications*. Amsterdam: Elsevier Science, ISBN: 978-0-857-09775-0.

Demirci, E., Acar, M., Pourdeyhimi, B., & Silberschmidt, V. (2011). Finite element modelling of thermally bonded bicomponent fibre nonwovens: Tensile behaviour. *Computational Materials Science*, *50*(4), 1286–1291. http://dx.doi.org/10.1016/j.commatsci.2010.02.039.

Dowling, N. (2013). *Mechanical behavior of materials : engineering methods for deformation, fracture, and fatigue*. Boston: Pearson, ISBN: 978-0131395060.

Eltayib, H. E., Ali, A. H. M., & Ishag, I. A. (2016). The prediction of tear strength of plain weave fabric using linear regression models. *International Journal of Advanced Engineering Research and Science*, *3*(11), 151–154. http://dx.doi.org/10.22161/ijaers/3.11.25.

Faessel, M., Delisée, C., Bos, F., & Castéra, P. (2005). 3D modelling of random cellulosic fibrous networks based on X-ray tomography and image analysis. *Composites Science and Technology*, *65*(13), 1931–1940. http://dx.doi.org/10.1016/j.compscitech.2004.12.038.

Fan, J., & Hunter, L. (1998). A worsted fabric expert system: Part II: An artificial neural network model for predicting the properties of worsted fabrics. *Textile Research Journal*, *68*(10), 763–771. http://dx.doi.org/10.1177/004051759806801010.

Farukh, F., Demirci, E., Sabuncuoglu, B., Acar, M., Pourdeyhimi, B., & Silberschmidt, V. (2015). Mechanical analysis of bi-component-fibre nonwovens: Finite-element strategy. *Composites Part B: Engineering*, *68*, 327–335. http://dx.doi.org/10.1016/j.compositesb.2014.09.003.

Gramsch, S., Klar, A., Leugering, G., Marheineke, N., Nessler, C., Strohmeyer, C., & Wegener, R. (2016). Aerodynamic web forming: process simulation and material properties. *Journal of Mathematics in Industry*, *6*(1), 13. http://dx.doi.org/10.1186/s13362-016-0034-4.

Harmening, M., Marheineke, N., & Wegener, R. (2020). Efficient graph-based tensile strength simulations of random fiber structures. *Journal of Applied Mathematics and Mechanics*, *101*(9), http://dx.doi.org/10.1002/zamm.202000287.

Karpatne, A., Watkins, W., Read, J., & Kumar, V. (2017). Physics-guided neural networks (PGNN): An application in lake temperature modeling. arXiv e-prints, arXiv:1710.11431.

Kufner, T., Leugering, G., Semmler, J., Stingl, M., & Strohmeyer, C. (2018). Simulation and structural optimization of 3D Timoshenko beam networks. *ESAIM: Mathematical Modelling and Numerical Analysis*, *52*(6), 2409–2431. http://dx.doi.org/10.1051/m2an/2018065.

Le Bris, C. (2010). Some numerical approaches for weakly random homogenization. In *Numerical mathematics and advanced applications 2009* (pp. 29–45). Springer, http://dx.doi.org/10.1007/978-3-642-11795-4_3.

Lu, Y., Rajora, M., Zou, P., & Liang, S. Y. (2017). Physics-embedded machine learning: Case study with electrochemical micro-machining. *Machines*, *5*(1), 4. http://dx.doi.org/10.3390/machines5010004.

Ohser, J., & Mücklich, F. (2000). *Statistical analysis of microstructures in materials science*. Weinheim: John Wiley, ISBN: 978-0-471-97486-4.

Ohser, J., & Schladitz, K. (2009). *3D Images of materials structures: Processing and analysis*. Weinheim: Wiley-VCH, ISBN: 978-3-527-31203-0.

Rahnama, M., Semnani, D., & Zarrebini, M. (2013). Measurement of the moisture and heat transfer rate in light-weight nonwoven fabrics using an intelligent model. *Fibres and Textiles in Eastern Europe*, *21*, 89–94.

Raina, A., & Linder, C. (2014). A homogenization approach for nonwoven materials based on fiber undulations and reorientation. *Journal of the Mechanics and Physics of Solids*, *65*, 12–34. http://dx.doi.org/10.1016/j.jmps.2013.12.011.

Ribeiro, R., Pilastri, A., Moura, C., Rodrigues, F., Rocha, R., Morgado, J., & Cortez, P. (2020). Predicting physical properties of woven fabrics via automated machine learning and textile design and finishing features. In *Artificial intelligence applications and innovations* (pp. 244–255). Springer, ISBN: 978-3-030-49186-4.

von Rüden, L., Mayer, S., Beckh, K., Georgiev, B., Giesselbach, S., Heese, R., Kirsch, B., Walczak, M., Pfrommer, J., Pick, A., Ramamurthy, R., Garcke, J., Bauckhage, C., & Schuecker, J. (2021). Informed machine learning - A taxonomy and survey of integrating prior knowledge into learning systems. *IEEE Transactions on Knowledge and Data Engineering*, 1. http://dx.doi.org/10.1109/TKDE.2021.3079836.

von Rüden, L., Mayer, S., Sifa, R., Bauckhage, C., & Garcke, J. (2020). Combining machine learning and simulation to a hybrid modelling approach: Current and future directions. In *Lecture notes in computer science*: *vol. 12080*, *Advances in intelligent data analysis* (pp. 548–560). Springer, http://dx.doi.org/10.1007/978-3-030-44584-3_43.

Schladitz, K., Peters, S., Reinel-Bitzer, D., Wiegmann, A., & Ohser, J. (2006). Design of acoustic trim based on geometric modeling and flow simulation for non-woven. *Computational Materials Science*, *38*(1), 56–66. http://dx.doi.org/10.1016/j.commatsci.2006.01.018.

Wegener, R., Marheineke, N., & Hietel, D. (2015). Virtual production of filaments and fleeces. In *Currents in industrial mathematics* (pp. 103–162). Springer, http://dx.doi.org/10.1007/978-3-662-48258-2_6.