Software Impacts 14 (2022) 100423

Contents lists available at ScienceDirect

Software Impacts

journal homepage: www.journals.elsevier.com/software-impacts

Original software publication

Machine learning framework to predict nonwoven material properties from fiber graph representations

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

Keywords:
Nonwoven fiber material
Manufacturing
Textile fabrics
Material property prediction
Graph representation
Tensile strength behavior

ABSTRACT

Nonwoven fiber materials are omnipresent in diverse applications including insulation, clothing and filtering. Simulation of material properties from production parameters is an industry goal but a challenging task. We developed a machine learning based approach to predict the tensile strength of nonwovens from fiber lay-down settings via a regression model. Here we present an open source framework implementing the following two-step approach: First, a graph generation algorithm constructs stochastic graphs, that resemble the adhered fiber structure of the nonwovens, given a parameter space. Secondly, our regression model, learned from ODE-simulation results, predicts the tensile strength for unseen parameter combinations.

Code metadata

Current code version	v1.0
Permanent link to code/repository used for this code version	https://github.com/SoftwareImpacts/SIMPAC-2022-144
Permanent link to reproducible capsule	https://codeocean.com/capsule/7514050/tree/v1
Legal code license	MIT License
Code versioning system used	github
Software code languages, tools, and services used	MATLAB, Python, bash
Compilation requirements, operating environments & dependencies	MATLAB 2021b, Python 3.8, matplotlib 3.5.2, numpy 1.22.3, pandas 1.4.2, scipy 1.8.1, networkx 2.6, pot 0.7.0, scikit-learn 1.0.2, tqdm 4.63.0, requests 2.27.1
Link to developer documentation/manual	https://github.com/pwelke/random-nonwoven-fibers/blob/main/README.md
Support email for questions	dario.antweiler@iais.fhg.de

Description

Predicting material properties from production parameters is desirable for many industrial settings including the production of nonwoven composites. These materials are characterized by a random fiber structure that is bonded using thermal, chemical or mechanical procedures. The application areas are diverse, including hygiene products, insulation materials, and fleece clothing [1]. One of the most important mechanical properties of nonwovens is their tensile strength, i.e., the amount of force that is needed to stretch a material sample in one dimension. The relationship between applied stress and resulting strain is typically represented by stress–strain curves. Within the industrial setting such curves can be produced by experimental measurements,

DOI of original article: https://doi.org/10.1016/j.mlwa.2022.100288.

The code (and data) in this article has been certified as Reproducible by Code Ocean: (https://codeocean.com/). More information on the Reproducibility Badge Initiative is available at https://www.elsevier.com/physical-sciences-and-engineering/computer-science/journals.

* Corresponding author at: Fraunhofer IAIS, Schloss Birlinghoven, Sankt Augustin, 53757, Germany.

E-mail addresses: dario.antweiler@iais.fhg.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.simpa.2022.100423
Received 11 July 2022; Received in revised form 5 September 2022; Accepted 8 September 2022

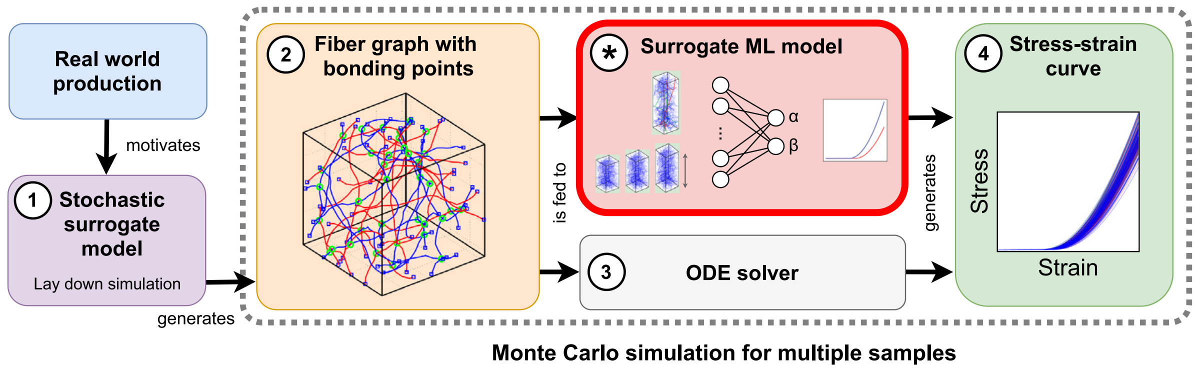

Fig. 1. Framework consisting of ① a surrogate model for the simulation of the fiber lay-down, ② the generation of a fiber graph, ③ a conventional solver for an ordinary differential equation describing the nonwovens' mechanical behavior under vertical load (ODE-solver), ④ and the final predicted stress–strain curve of the material. The presented machine learning approach ⊛ can reliably approximate the resulting curves via regression based on selected graph features while achieving a 1000× speedup.

but this approach is limited to few examples due to the physical effort that is required to conduct experimental tensile strength tests and the costs of stopping a running production for testing. This motivates simulating the stress–strain behavior through computational means. To be able to map a set of production parameters to the associated tensile strength behavior one requires (i) a model of the underlying production process and (ii) a model of the nonwovens' mechanical behavior. To bypass the associated simulation costs we introduce (iii) a machine learning approach. The individual components and their interactions are depicted in Fig. 1.

Fiber structure generation

For the production process (i) we consider the nonwoven airlay production incorporating a thermal bonding procedure, originally modeled in [2]. Therein the authors introduce a stochastic lay-down model, resulting in random virtual fiber structures that resemble the fiber structure of actual nonwovens, as well as a virtual bonding procedure, simulating the thermobonding. The stochastic lay-down model and the virtual bonding procedure are initiated by various tunable parameters involved in the airlay production process. Here, we consider an adaption of this approach, described in detail in [3], that additionally accounts for the nestling of the fibers to the characteristic fiber-ramp which builds up during production and a material composition of different (adhesive and non-adhesive) fiber types. The topology of the resulting virtually bonded fiber structure samples can be described using graphs, where nodes represent adhesive joints as well as fiber ends and edges represent fiber connections between them. These graphs, equipped with a fiber connection length for each edge and a spatial position for each node, are in the following referred to as fiber graphs and serve as basis for simulating/predicting the nonwovens' tensile strength behavior.

Tensile strength simulations

The employed tensile strength model-simulation framework (ii), originating from [3], recreates the elastic phase of the nonwovens' tensile strength behavior under vertical load. The suitability of the simulation results is discussed in [4]. Particularly, the model describes the mechanical behavior of the adhered fiber structure by capturing the interaction of the individual fiber connections, each equipped with a nonlinear material law, at network level. An additional regularization, ensuring the well-posedness of the model, allows to formulate this as large scale ordinary differential equation system (ODE). Hence, the tensile behavior of some sampled adhered fiber structure under vertical load can be simulated by straightforward numerical integration of the associated ODE. In the course of this procedure the relation of the sample strain to the reacting tractive forces can be traced. The resulting stress–strain curves describe the tensile strength behavior. Since production-like fiber structures contain many thousand fibers, the simulations are numerically demanding. The computational effort is multiplied further by a Monte-Carlo simulation framework that is required to infer the general material behavior. Although this model chain enables simulations in practice, cf. [5], it does not allow for advanced applications, such as virtual nonwoven material design, despite preliminary homogenization and parallelization.

Machine learning regression model

We approach the problem outlined above by introducing (iii) an interpretable machine learning regression model [4]. Subject of the predictions are the stress–strain curves associated to individual fiber graphs. As basis for the predictions serve features that are extracted from the respective fiber graphs. This comprises a selected set of standard graph features, such as the number of nodes and edges or the lengths of (weighted) shortest paths connecting the top to the bottom of the samples, as well as a set of stretch features. The stretch features are determined using a novel stretching algorithm [4] that is based on a reduced model of the nonwovens' tensile behavior. This already encodes a lot of information for predictions, especially with regard to predicting the sample elongation at which fibers straighten out and increased stress occurs. We emphasize that the resulting group of stretch features is particularly developed for the present purpose. The extracted features are used as input to a regression model, that relates them to the stress–strain curves associated to the individual fiber graphs. Stress–strain curves are approximated by the following constant-quadratic parametrization [3]:

$$T_p(\epsilon) = \begin{cases} 0, & \epsilon < \alpha \\ \beta(\epsilon-\alpha)^2, & \epsilon \geq \alpha, \end{cases} \quad \text{where } \boldsymbol{p} = (\alpha, \beta) \in \mathbb{R}_+^2 \tag{1}$$

with ϵ referring to the relative strain applied to the sample and $T_p : \mathbb{R}_+ \to \mathbb{R}_+$ describing the resulting reacting force, where $\boldsymbol{p} = (\alpha, \beta) \in \mathbb{R}_+^2$ parametrizes the curve. We opt for a simple linear regression model to allow interpretability of the results and the inspection of feature importances. During training, the model identifies optimal linear regression weights via Ordinary Least Squares (OLS) to relate the input features (cf. Table 1) to the output labels α, β respectively. Training data consists of pairs of fiber graphs and associated stress–strain curves that are generated according to the previously described model chain. To assess the quality of our predictions, we compare them to the original stress–strain curves produced by numerical integration of the ODE model. We obtain a coefficient of determination of $R^2 = 0.98$ using leave-one-out cross-validation while achieving a 1000× speedup in calculation time over the actual tensile strength simulations.

Table 1
Input features used in the proposed regression model.

Set	Symbols	Description
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	$\{\text{mean}, \text{median}, \text{sum}\}$ 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 $c \in \{1, 1.05, 1.1 \dots, 1.5\}$

Usage & impact

The code (provided on CodeOcean) is structured in a modular way according to the individual models listed above. The random fiber structure generator and the tensile strength simulation are written in Matlab. With regard to the regression model, these are used to generate the required training data. Note that we provide a dataset of appropriate size, as its generation is very time-consuming.[1] The implementation of the feature extraction as well as that of the regression model and its validation are written in Python.

The default for code execution is the shell script `run_ computation.sh` which generates exemplary data that is used to train and validate the machine learning regression model. Expected arguments are either “all”, “single” or “none”. If `run_computation.sh all` is executed, a completely new dataset is generated and used for training and validation. Initial production parameters are sampled from ranges defined in [4], which may be adjusted in the Matlab routine `runDataBaseGeneration.m`. It takes the arguments “NFullyLabeled”, “NSingleLabeled” as well as “NSamples”. Following [4] this allows to generate (“NFullyLabeled”) fully labeled data sets and (“NSingleLabeled”) single labeled data sets. For both data sets (“NSamples”) fiber graphs are generated where either all of them (fully labeled) or just one of them (single labeled) are equipped with an associated stress–strain curve. We note that labeling takes a considerable amount of computing time for which a paralellization is included. On execution of `run_computation.sh single`, a single data pair (consisting of a fiber graph and an associated stress–strain curve) is generated and added to the provided data set for training and validation. This is done using the Matlab routine `runSampleAndSimulate.m`, which accepts the arguments “NLabeledSamples” and “NUnlabeledSamples” that specify the number of fiber graphs sampled with and without associated stress–strain curve, respectively. Note that this option is provided to illustrate the generation of an additional data pair. The default for execution, however, is `run_computation.sh none` which uses a small precomputed dataset without generating any additional data pairs.

In all cases, the resulting dataset is processed using the python script `feature_generation.py`, which computes graph features that serve as explanatory variables for regression, and `ansatzfitting.py`, which labels the stress–strain curves for prediction. The labels of the stress–strain curves serve as dependent variables. In a last step, invoked by running `train_validate.py`, the obtained variables (explanatory/dependent) are used for training and validation of the regression model, the latter of which is performed using a Leave-One-Out Cross Validation across the sample production parameter combinations, cf., [4]. The resulting regression model is stored and can be used within the script `predict.py` to label newly generated fiber graphs.

The proposed simulation framework can be used for computer-aided process design and material optimization. Furthermore, it represents a proof of concept for the application of machine learning approaches to the prediction of nonwoven material properties. Since virtual material design has so far been limited by the high computational costs, this new machine learning model allows further research into more detailed process and product optimizations that have previously not been computationally feasible.

[1] Full dataset is available via https://github.com/pwelke/random-nonwoven-fibers/blob/main/code/download_data.py

Limitations & future work

The underlying model-simulation framework and thus the trained regression model recreate the elastic phase of the nonwovens tensile strength behavior. Possible extensions include the introduction of plastic effects such as fiber tearing as well as prediction modules for effects including twisting and bending of fibers. Additionally, we aim to extend our framework towards other nonwoven material properties including insulation, flow resistance or acoustic properties as well as different production processes.

Publications

Scholarly publications enabled by our software:

- Dario Antweiler, Marc Harmening, Nicole Marheineke, Andre Schmeißer, Raimund Wegener, Pascal Welke *Graph-Based Tensile Strength Approximation of Random Nonwoven Materials by Interpretable Regression*, Machine Learning with Applications, 2022, https://doi.org/10.1016/j.mlwa.2022.100288
- Marc Harmening, Nicole Marheineke, and Raimund Wegener. *Efficient graph-based tensile strength simulations of random fiber structures*. ZAMM - Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik, 2021, https://doi.org/10.1002/zamm.202000287

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, RTG 2126) at Trier University, funded by the German Research Foundation.

References

[1] Dipayan Das, Behnam Pourdeyhimi, Composite Nonwoven Materials: Structure, Properties and Applications, Elsevier, 2014.

[2] Simone Gramsch, Axel Klar, Günter Leugering, Nicole Marheineke, Christian Nessler, Christoph Strohmeyer, Raimund Wegener, Aerodynamic web forming: process simulation and material properties, J. Math. Ind. 6 (1) (2016).

[3] Marc Harmening, Nicole Marheineke, Raimund Wegener, Efficient graph-based tensile strength simulations of random fiber structures, ZAMM - J. Appl. Math. Mech./Z. Angew. Math. Mech. 101 (9) (2021) e202000287.

[4] Dario Antweiler, Marc Harmening, Nicole Marheineke, Andre Schmeißer, Raimund Wegener, Pascal Welke, Graph-based tensile strength approximation of random nonwoven materials by interpretable regression, Mach. Learn. Appl. 8 (2022) 100288.

[5] Raimund Wegener, Nicole Marheineke, Dietmar Hietel, Virtual production of filaments and fleeces, in: Currents in Industrial Mathematics, Springer, 2015, pp. 103–162.